



ROSELINE OLUWASEUN OGUNDOKUN

**POSTURE TRACKING
METHODS ON
OCCLUDED VIDEO
MATERIAL**

DOCTORAL DISSERTATION

K a u n a s
2 0 2 6

KAUNAS UNIVERSITY OF TECHNOLOGY

ROSELINE OLUWASEUN OGUNDOKUN

POSTURE TRACKING METHODS ON
OCCLUDED VIDEO MATERIAL

Doctoral dissertation
Technological Sciences, Informatics Engineering (T 007)

2026, Kaunas

The dissertation has been prepared at the Department of Software Engineering of the Faculty of Informatics of Kaunas University of Technology in 2021/2025.

The doctoral right has been granted to Kaunas University of Technology together with Vilnius Gediminas Technical University.

Research supervisor:

Prof. Dr. Rytis MASKELIŪNAS (Kaunas University of Technology, Technological Sciences, Informatics Engineering, T 007).

Edited by: English language editor Dr. Brigita Pantelejeva (KTU Centre of Foreign Languages), Lithuanian language editor Virginija Stankevičienė (KTU Centre of Foreign Languages).

Dissertation Defence Board of Informatics Engineering Science Field:

Prof. Dr. Rimantas BUTLERIS (Kaunas University of Technology, Technological Sciences, Informatics Engineering, T 007) – **chairperson**;

Prof. Dr. Hab. Romualdas BAUŠYS (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering, T 007);

Prof. Dr. Varadraj GURUPUR (University of Central Florida, United States, Natural Sciences, Informatics, N 009);

Prof. Dr. Arnas KAČENIAUSKAS (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering, T 007);

Prof. Dr. Filippo SANFILIPPO (University of Agder, Norway, Natural Sciences, Informatics, N 009).

The dissertation defence will be held on 27 April 2026, at 10:30 a. m. at the Rectorate Hall of Kaunas University of Technology in the meeting of the Dissertation Defence Board of the Informatics Engineering Science field.

Address: K. Donelaičio 73-402, LT-44249 Kaunas, Lithuania.

Phone: +370 608 28 527; email doktorantura@ktu.lt

The dissertation was sent out on 27 March 2026.

The dissertation is available on the website <http://ktu.edu>, at the Library of Kaunas University of Technology (Gedimino 50, Kaunas) and Vilnius Gediminas Technical University (Saulėtekio 14, Vilnius).

© R. O. Ogundokun, 2026

KAUNO TECHNOLOGIJOS UNIVERSITETAS

ROSELINE OLUWASEUN OGUNDOKUN

LAIKYSĖNOS NUSTATYMO METODAI IŠ
DALINAI UŽSTOJAMŲ KŪNŲ

Daktaro disertacija
Technologijos mokslai, informatikos inžinerija (T 007)

Kaunas, 2026

Disertacija rengta 2021–2025 metais Kauno technologijos universiteto Informatikos fakulteto Programų inžinerijos katedroje.

Doktorantūros teisė Kauno technologijos universitetui suteikta kartu su Vilniaus Gedimino technikos universitetu.

Mokslinis vadovas:

prof. dr. Rytis MASKELIŪNAS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija, T 007).

Redagavo: anglų kalbos redaktorė dr. Brigita Pantelejeva (KTU Užsienio kalbų centras), lietuvių kalbos redaktorė Virginija Stankevičienė (KTU Užsienio kalbų centras).

Informatikos inžinerijos mokslo krypties disertacijos gynimo taryba:

prof. dr. Rimantas BUTLERIS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija, T 007) – **pirmininkas**;

prof. habil. dr. Romualdas BAUŠYS (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija, T 007);

prof. dr. Varadraj GURUPUR (Centrinės Floridos universitetas, JAV, gamtos mokslai, informatika, N 009);

prof. dr. Arnas KAČENIAUSKAS (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija, T007);

prof. dr. Filippo SANFILIPPO (Agderio universitetas, Norvegija, gamtos mokslai, informatika, N 009).

Disertacija bus ginama viešame Informatikos inžinerijos mokslo krypties disertacijos gynimo tarybos posėdyje 2026 m. balandžio 27 d. 10.30 val. Kauno technologijos universiteto Rektorato salėje.

Adresas: K. Donelaičio g. 73-402, LT-44249 Kaunas, Lietuva.

Tel: +370 608 28 527; el. paštas doktorantura@ktu.lt

Disertacija išsiųsta 2026 m. kovo 27 d.

Su disertacija galima susipažinti interneto svetainėje <http://ktu.edu>, Kauno technologijos universiteto bibliotekoje (Gedimino g. 50, Kaunas) ir Vilniaus Gedimino technikos universiteto bibliotekoje (Saulėtekio al. 14, Vilnius).

CONTENT

LIST OF TABLES.....	7
LIST OF FIGURES.....	8
LIST OF ALGORITHMS	10
LIST OF ABBREVIATIONS AND TERMS.....	11
INTRODUCTION.....	13
1. LITERATURE ANALYSIS.....	23
1.1. Trends and Gaps in Posture Detection Research.....	23
1.2. Depth-Wise Separable CNN for Human Posture Detection.....	25
1.3. Image Augmentation and Hyperparameter-Optimized Transfer Learning Algorithms.....	25
1.4. Human Posture Detection in Health Monitoring (Hybrid InceptionV3-SVM).....	27
1.5. Deep Feature Fusion Technique for Transfer Learning Optimization.....	30
1.6. Improved CNN Architectures with Batch Normalization and Adam Optimizer.....	31
1.7. Lightweight DCNN and SVM for Digital Healthcare Posture Detection.....	34
1.8. Hybrid MobileNetV2-DNN for Advanced Image Analysis.....	35
1.9. Hybrid Deep VGG16 and Attention Mechanisms for Posture Enhancement..	36
1.10. Summary.....	42
2. MATERIALS AND METHODS.....	43
2.1. Datasets and Preprocessing.....	44
2.1.1. MPII Human Pose Dataset.....	45
2.1.2. Silhouettes of Human Posture Dataset.....	46
2.1.3. ETRI-Activity3D-LivingLab Dataset.....	47
2.2. Data Preprocessing and Augmentation.....	49
2.3. Feature Extraction Technique.....	49
2.3.1. Rationale for LASSO FS.....	50
2.3.2. Benefits of Posture Detection.....	50
2.4. Model Architecture.....	50
2.4.1. AlexNet Model.....	51
2.4.2. Custom CNN Model.....	51
2.4.3. Deep Neural Network (DNN).....	51
2.4.4. Support Vector Machine (SVM).....	51
2.5. Transfer Learning (TL) Models.....	51
2.5.1. VGG 16.....	52
2.5.2. InceptionV3.....	52
2.5.3. Xception.....	52
2.5.4. DenseNet121.....	52
2.5.5. MobileNetV2.....	53
2.5.6. ResNet50V2.....	53
2.6. Hybrid and Ensemble Architectures.....	53
2.6.1. Hybrid LASSO+InceptionV3+SVM.....	53

2.6.2. Hybrid DenseNet121-SVM (DeneSVM).....	55
2.6.3. OptiMobileX: Deep Feature Fusion Architecture.....	55
2.6.4. VEACTION-CNN (VGG16-Attention).....	56
2.6.5. MobileNetV2-DNN.....	57
2.7. Comparative insights.....	57
2.8. Summary.....	58
3. EXPERIMENTAL RESULTS.....	60
3.1. Experimental Setup.....	60
3.2. Experimental Results on Silhouettes Dataset.....	61
3.2.1. Training and Validation Performance.....	61
3.2.2. Confusion Matrix Analysis.....	67
3.2.3. Class-wise Metrics.....	73
3.2.4. ROC and Precision-Recall Analysis.....	74
3.2.5. Visualization of Posture Classification results.....	84
3.2.6. Comparative Insights and Robustness.....	85
3.3. MPII Dataset and Experimental Results.....	86
3.3.1. Training and Validation Performance.....	86
3.3.2. Testing Accuracy and Class-wise Metrics.....	88
3.3.3. Confusion Matrix, ROC, and PR Curves Analysis.....	89
3.3.4. Comparative Strengths and OptiMobileX Performance.....	98
3.4. ETRI-Activity3D-LivingLab Dataset and Experimental Results	99
3.4.1. Training and Validation Performance.....	99
3.4.2. Confusion Matrix, ROC and PR Curve Analysis.....	103
3.4.3. Results from Training on Training and Validation Datasets.....	117
3.4.4. Experimental Results on Test Dataset.....	121
3.4.5. Computational Resource Evaluation.....	122
3.4.6. Comparative Insights.....	123
3.5. Explainability of the Proposed Lightweight VEACTION-CNN Model on ETRI-Activity3D-LivingLab Dataset.....	123
3.6. Ablation Study.....	127
3.7. Statistical Results.....	128
3.8. Threats to Validity.....	131
3.9. Comparative Analysis with State-of-the-Art.....	133
3.10. Clinical and Practical Implications of Posture Classification.....	134
3.11. Ethics Statement.....	135
CONCLUSIONS.....	136
SANTRAUKA.....	141
LIST OF REFERENCES.....	182
CURRICULUM VITAE AND DESCRIPTION OF CREATIVE ACTIVITIES (CV).....	200
LIST OF SCIENTIFIC PAPERS AND SCIENTIFIC CONFERENCES.....	202
ACKNOWLEDGEMENT.....	204
APPENDICES.....	205
Appendix 1. Code.....	205

LIST OF TABLES

Table 1. Summary of Publications and the Corresponding Methods.....	20
Table 2. Summary of Related Works Reviewed.....	38
Table 3. Accuracy and Loss of Training, Validation and Testing on Silhouettes Data.....	62
Table 4. Results of the performance evaluation obtained from the Silhouettes Test Dataset.....	73
Table 5. Results of the performance evaluation obtained from the MPII Test Dataset.....	89
Table 6. Accuracy and Loss of Training, Validation and Testing on ETRI- Activity3D LivingLab Dataset	118
Table 7. Results of the performance evaluation obtained from the ETRI-Activity3D LivingLab Test Dataset.....	121
Table 8. Computational Resources Evaluation of the Proposed Models	123
Table 9. Computational Resources on the MPII Dataset.....	130
Table 10. Statistical Results on Silhouettes Dataset	130
Table 11. Computational Resources on Silhouettes Dataset	130
Table 12. Computational Resources of Hybrid Models on Silhouette Dataset.....	131
Table 13. Comparison of the system with other techniques.....	133

LENTELIŲ SĄRAŠAS

14 lentelė. Apžvelgtų susijusių darbų santrauka.....	148
--	-----

LIST OF FIGURES

Fig. 1. Deep CNN Architecture.....	26
Fig. 2. The Proposed Human Posture Detection Model.....	44
Fig. 3. Samples of MPII Human Pose Dataset	46
Fig. 4. Four Classes Sample of Silhouettes of Human Posture Dataset	47
Fig. 5. Four Classes Sample of ETRI-Activity3D-LivingLab Dataset	48
Fig. 6. Accuracy and loss graphs of DTL models Sihuoottes Data: (i) InceptionV3; (ii) MobileNetV2; (iii) Xception, (iv) OptiMobileX, (v) ResNet50V2, (vi) DenseNet121, (vii) DeneSVM, (viii) LASSO+InceptionV3+SVM, and MobileNetV2-DNN.....	65
Fig. 7. The Nine Implemented Models' Accuracy Comparison.....	65
Fig. 8 The Nine Implemented Models' Loss Comparison	66
Fig. 9. Radar Chart of Proposed Hybrid Model Accuracies.....	67
Fig. 10. Confusion matrix of DTL models on Sihuoottes Data: (i) InceptionV3; (ii) MobileNetV2; (iii) Xception, (iv) OptiMobileX, (v) ResNet50V2, (vi) DenseNet121, (vii) DeneSVM, (viii) LASSO+InceptionV3+SVM, and MobileNetV2-DNN.	72
Fig. 11. ROC curve of DTL models on Sihuoottes Data: (i) InceptionV3; (ii) MobileNetV2; (iii) Xception, (iv) OptiMobileX, (v) ResNet50V2, (vi) DenseNet121, (vii) DeneSVM, (viii) LASSO+InceptionV3+SVM, and (ix) MobileNetV2-DNN.	79
Fig. 12. Precision-Recall curve of DTL models on Sihuoottes Data: (i) InceptionV3; (ii) MobileNetV2; (iii) Xception, (iv) OptiMobileX, (v) ResNet50V2, (vi) DenseNet121, (vii) DeneSVM, (viii) LASSO+InceptionV3+SVM, and (ix) MobileNetV2-DNN.....	83
Fig. 13. Sample Results of Posture Classification Showing Actual vs Predicted Labels	84
Fig. 14. Accuracy and loss graphs of DTL models on MPII Dataset: (i) Alexnet, (ii) InceptionV3, (iii) MobileNetV2, (iv) Xception, and (v) OptiMobileX.....	88
Fig. 15. Confusion matrix of DTL models on MPII: (i) AlexNet; (ii) InceptionV3; (iii) MobileNetV2; (iv) Xception, and (v)OptiMobileX.	92
Fig. 16. PR of DTL models on MPII: (i) AlexNet; (ii) InceptionV3; (iii) MobileNetV2; (iv) Xception, and (v) OptiMobileX.....	94
Fig. 17. PR of DTL models on MPII: (i) AlexNet; (ii) InceptionV3; (iii) MobileNetV2; (iv) Xception, and (v) OptiMobileX.....	97
Fig. 18. Prediction of the postures using the proposed model.....	98
Fig. 19. Accuracy and loss graphs of DTL models ETRI-Activity3D-LivingLab Dataset: (i) VEACTION-CNN, (ii) Ensemble, (iii) MobileNetV2, (iv) CNN, (v) Xception, (vi) VGG16, (vii) LSTM, (viii) GRU, and (ix) LitePose-GRU	102
Fig. 20. Confusion Matrix of DTL models ETRI-Activity3D-LivingLab Dataset: (i) VEACTION-CNN, (ii) Ensemble, (iii) MobileNetV2, (iv) CNN, (v) Xception, (vi) VGG16, (vii) LSTM, (viii) GRU, and (ix) LitePose-GRU.....	107

Fig. 21. ROC Curves of DTL models ETRI-Activity3D-LivingLab Dataset: (i) VEACT CNN, (ii) Ensemble, (iii) MobileNetV2, (iv) CNN, (v) Xception, (vi) VGG16, (vii) LSTM, (viii) GRU, and (ix) LitePose-GRU.....	112
Fig. 22. PR Curves of DTL models ETRI-Activity3D-LivingLab Dataset: (i) VEACT-CNN, (ii) Ensemble, (iii) MobileNetV2, (iv) CNN, (v) Xception, (vi) VGG16, (vii) LSTM, (viii) GRU, and (ix) LitePose-GRU.....	117
Fig. 23. Comparison of Model Accuracies.....	119
Fig. 24. Comparison of Model Losses.....	119
Fig. 25. Radar Plot for the Model Accuracies.....	120
Fig. 26. Radar Plot for the Model Losses.....	120
Fig. 27. Explanability with Grad-CAM on ETRI-Activity3D-LivingLab Dataset.	126
Fig. 28. Prediction Examples on ETRI-Activity3D-LivingLab Dataset Test Dataset.....	126

PAVEIKSLŲ SĄRAŠAS

29 pav. MPII žmogaus pozų duomenų rinkinio pavyzdžiai	154
30 pav. Keturių klasių žmogaus laikysenos duomenų rinkinio siluetų pavyzdžia..	155
31 pav. Keturių klasių <i>ETRI-Activity3D-LivingLab</i> duomenų rinkinio pavyzdys..	156
32 pav. Siūloma žmogaus laikysenos nustatymo sistemos struktūra.....	157
33 pav. DTL modelių tikslumo ir nuostolių grafikai <i>Sihuottes</i> duomenys: <i>InceptionV3</i> , <i>MobileNetV2</i> , <i>Xception</i> ir siūlomas lengvasis <i>OptiMobileX</i>	166
34 pav. DTL modelių sumaišymo matrica su <i>Sihuottes</i> duomenimis: (i) <i>InceptionV3</i> ; (ii) <i>MobileNetV2</i> ; (iii) <i>Xception</i> , (iv) <i>OptiMobileX</i> ir (v) <i>DeneSVM</i>	170
35 pav. DTL modelių ROC kreivė <i>Sihuottes</i> duomenims: (i) <i>InceptionV3</i> ; (ii) <i>MobileNetV2</i> ; (iii) <i>Xception</i> ir (iv) <i>OptiMobileX</i>	174
36 pav. DTL modelių tikslumo ir atsišaukimo santykis MPII: (i) <i>InceptionV3</i> ; (ii) <i>MobileNetV2</i> ; (iii) <i>Xception</i> ; ir (iv) <i>OptiMobileX</i>	180

LIST OF ALGORITHMS

Algorithm 1. Hybrid InceptionV3-SVM.....	54
Algorithm 2. Hybrid DenseNet121-SVM (DeneSVM).....	55
Algorithm 3. OptiMobileX.....	56
Algorithm 4. VEACTION-CNN (VGG16 Attention).....	56
Algorithm 5. MobileNetV2-DNN.....	57

ALGORITMŲ SĄRAŠAS

Algoritmas 6: Hibridinis InceptionV3-SVM.....	158
Algoritmas 7: Hibridinis DenseNet121-SVM (DeneSVM).....	159
Algoritmas 8: OptiMobileX.....	160
Algoritmas 9: VEACTION-CNN.....	160
Algoritmas 10: MobileNetV2-DNN.....	161

LIST OF ABBREVIATIONS AND DEFINITIONS

Abbreviations:

HPD – Human Posture Detection;
CNN – Convolutional Neural Network;
DL – Deep Learning;
TL– Transfer Learning;
SVM – Support Vector Machine;
LASSO – Least Absolute Shrinkage and Selection Operator;
HPO– Hyperparameter Optimization;
AUC – Area Under the Curve;
DCNN – Deep Convolutional Neural Network;
LSTM– Long Short Term Memory;
CV – Computer Vision;
HPE – Human Posture Estimation;
DTL – Deep Transfer Learning;
ILSVRC – ImageNet Large-Scale Visual Recognition Challenge;
GPU– Graphics Processing Unit;
HPR – Human Posture Recognition;
DFC – Deep Feature Concatenation;
BN – Batch Normalization;
RN – Residual Network;
ELU – Exponential Linear Units;
DA – Data Augmentation;
CAD – Computer-aided Design;
DNN – Deep Neural Network;
AI – Artificial Intelligence;
HPC– Human Posture Classification;
MLP – Multilayer Perceptron.

Definitions:

Body Posture – This is the placement and alignment of an individual’s body parts in relation to each other when standing, sitting, or moving.

Posture – This refers to the natural way in which a person holds their body, whether in movement or stationary, which indicates alignment and equilibrium.

Occlusion – This refers to the obstruction or masking of a particular object detail, or view within an image or scene, which is partially or completely hidden.

Video Dataset – This concept refers to a set of structured video recordings collected for research, training, or testing purposes in computer vision and related fields.

Deep Learning – Deep learning is a subfield of machine learning that utilises multi-layered neural networks to automatically learn complex patterns from large datasets.

Convolutional Neural Network – This refers to one DL models that is specifically designed to automatically learn and extract spatial features from image or video data.

Data Augmentation – This refers to a technique used for enriching data diversity using application of transformations such as rotation, scaling, etc, to available data without having to sample new data.

Hyperparameter Optimisation – This is the process of making methodical decisions on the best set of model parameters not learned during training to maximise performance.

Artificial Intelligence – This refers to the area of computer science that develops systems capable of performing tasks that typically require human intelligence.

Human Posture Detection – This is the automated process of identifying and analysing the posture and position of a human body in images or videos.

Transfer Learning – This is a method in which a model is trained to achieve one task, and the dataset is then transferred to another, but related, task, usually with fewer data points.

Hybrid Model – This is a computational strategy that combines multiple models or algorithms to leverage their strengths and improve performance.

Concatenation – This refers to the process of linking or aggregating data elements, such as feature vectors, into a unified and coherent framework.

Baseline Model – This is an initial, perhaps less sophisticated, model used as a reference for subsequent improvements.

Posture Detection – This is the computer vision task of recognising and categorising the posture and positioning of an object's body.

Pre-trained Model – This model refers to a neural network that has already been trained on a large corpus and can be adapted to a new or similar task.

INTRODUCTION

Relevance of the Work

Human Posture Detection (HPD) from video datasets is a core research area in computer vision due to its broad societal, technological, scientific, and technical applications [1]. Automated posture recognition aids vital healthcare, ergonomics, and smart surveillance purposes [1]. Recently, HPD has become increasingly essential in human-computer interaction, reflecting an emerging interest in interpreting human activities with greater nuance. Requests from ageing people, inaccessible personnel, and intelligent security measures necessitate reliable detection with minimal video data, motivating the development of advanced HPD approaches. Posture detection enhances well-being, privacy, and safety in the healthcare industry. Specific vision-based monitoring enables remote measurement of patients and older adults. This enables health professionals to prevent falls and respond promptly to health emergencies [2]. Falls are a significant cause of injury among seniors, and systems that recognise abnormal postures or collapses can be lifesaving [2, 3].

The dissertation is relevant to ergonomics and workplace health, particularly given the increasing prevalence of sedentary lifestyles and teleworking. Poor sitting or standing posture can cause chronic back pain and musculoskeletal disorders, common across all ages [4]. The COVID-19 pandemic led to increased teleworking, often in non-ergonomic home settings, resulting in poor posture and a higher risk of neck and back injuries [4-7]. This highlights the need for automated posture detection systems to monitor and coach real-time posture improvements. Early detection and correction of poor postures can prevent long-term injuries [5-7]. Studies show that AI-powered posture monitoring can reduce workplace injuries and improve productivity by advising users on posture adjustments [8, 9]. Integrating posture detection tools, such as computer cameras, encourages healthier work habits. Hence, this dissertation is valuable for creating safer work environments and promoting better postural hygiene, both of which are crucial given the continued prevalence of telework. Another relevant area is intelligent surveillance. Posture detection algorithms enhance public safety by enabling surveillance systems to automatically identify suspicious behaviours through body pose and movement [10, 11]. Specific postures can indicate potential threats, such as crouching, lying motionless, or making aggressive gestures. AI-powered video analytics can detect such anomalies in real time and alert security personnel [10, 12]. This shifts surveillance from passive recording to active scene analysis. Modern platforms already utilise posture recognition to identify potential dangers, such as fights or weapons. Integrating the findings of this dissertation, cameras in “smart” environments can better detect when postures deviate from norms, prompting intervention. This dissertation improves technology while benefiting society by creating safer spaces through preventive monitoring.

The dissertation enhances AI’s ability to understand humans using vision-based methods. Traditional computer vision struggled with human anatomy and challenging

imaging conditions. Deep learning, particularly convolutional neural networks (CNNs), has enhanced posture analysis by learning complex features that surpass those crafted manually. The standard for posture detection currently is a CNN, which attains high accuracy in recognizing postures and subtle poses. Furthermore, a large dataset and computational resources are required for model training, which is a limitation when the data are small. This has led to the widespread adoption of the emerging approach known as transfer learning (TL) for posture detection, which utilises pre-trained CNNs initially trained on a large dataset, such as ImageNet. These pre-trained models are often used for fine-tuning in posture detection. TL addresses data scarcity in specialized sectors by recycling visual characteristics, which lowers training time and data requirements while retaining high accuracy. This paper contributes to the trend of making AI-driven analytics more effective, scalable, and helpful in real-world scenarios without massive datasets by investigating optimal deep learning (DL) and TL to improve posture detection with limited video data.

The dissertation is highly relevant as it addresses societal demands in healthcare, wellness and safety by employing technology to identify human posture. The dissertation also highlights AI trends such as deep transfer learning and edge computing to build responsive systems. Technologically, it presents a novel approach to posture detection using a limited video dataset, thereby addressing a gap in the data-insufficient, resource-constrained literature. This dissertation enhances and advances the understanding and practical application of posture-aware systems. Progressing health, safety and surveillance effectiveness. Ultimately, this dissertation explores the relationship between AI innovation and health-built applications, with a focus on their practical benefits.

Problem Formulations

Human posture recognition is an essential component of health monitoring systems, enabling the assessment of a person's well-being and the early detection of potential issues (e.g., risk of falls or musculoskeletal strain) [13]. However, developing a reliable automated posture detection system is challenging for several reasons. First, image-based human posture classification must contend with complex environments: occlusions, varying camera view angles, and overlapping body parts can all make pose identification difficult [14]. Second, modern deep learning models, such as deep CNNs, require a large number of annotated training images to learn millions of parameters; in practice, posture datasets are often small, leading to overfitting and poor generalisation [15]. Traditional approaches, using either conventional machine learning or off-the-shelf deep networks, have not effectively addressed these issues [16]. Models typically struggle to accurately identify specific postures and tend to overfit on the limited available data [16]. These limitations highlight the need for novel methods that can achieve accurate human posture recognition with limited training data and address real-world imaging challenges [17]. Therefore, the dissertation problem is formulated as developing an improved deep learning-based human posture detection model that generalises well on limited data

and in complex scenes, thereby providing accurate posture classification for health monitoring applications.

Aim of this Doctoral Dissertation

This dissertation aims to design and validate optimized lightweight and hybrid deep learning model for accurate and efficient human posture detection using limited video datasets.

Objectives

The specific objectives of this study are to:

1. To design and implement novel deep learning models such as VEACTION, OptiMobileX, LitePose-GRU, DeneSVM, LASSO+InceptionV3+SVM and MobileNet-DNN for accurate human posture detection.
2. To investigate the effectiveness of lightweight deep learning models for posture detection, focusing on reducing computational complexity, parameter count, and model size while maintaining high accuracy.
3. To compare the proposed models with existing baseline models such as CNN, LSTM, and hybrid CNN-LSTM models in terms of performance and generalisation.
4. To develop and evaluate the proposed hybrid models for practical applications on limited video datasets.
5. To analyse and interpret model decisions using visualization tools like Grad-CAM to enhance explainability and trust.

Tasks of the Research

To achieve the aim and objectives, the following tasks are performed:

1. Conduct a detailed literature review to identify gaps in human posture detection methods.
2. Obtain and preprocess benchmark datasets (MPII, Silhouettes and ETRI).
3. Apply image augmentation and hyperparameter optimization techniques.
4. Develop and train the proposed lightweight hybrid models: VEACTION, OptiMobileX, LitePose-GRU, DeneSVM, LASSO+InceptionV3+SVM and MobileNet-DNN for posture detection.
5. Evaluate and compare model performance with state-of-the-art baselines.
6. Analyse and interpret model outcomes, including performance metrics and visualisation-based results.

Scientific Novelty

This dissertation provides several novel contributions to the field of human posture recognition and deep learning:

1. The hybrid deep learning and machine learning models were introduced. It introduces a hybrid deep learning and machine learning model, called the LASSO+InceptionV3-SVM model, for posture detection. This model combines deep convolutional features from an InceptionV3 network with a Support Vector Machine classifier. This approach, coupled with LASSO-based feature selection, is novel in the posture recognition domain and improves generalisation by leveraging both deep feature extraction and robust machine-learning classification. The hybrid model achieves high accuracy while reducing overfitting, providing a novel approach to integrating deep learning with classical classifiers for posture classification.

2. The dissertation integrates hyperparameter optimisation in posture detection. This involves applying automated hyperparameter optimisation techniques to human posture detection on the MPII dataset. A three-phase decision-support model was developed that integrates transfer learning, image data augmentation, and HPO to systematically identify the optimal training configuration for deep networks. This novel integration of HPO led to a significant improvement in accuracy on MPII. It demonstrates the importance of tuning parameters specifically for medical posture data, an aspect that has previously been neglected in the literature.

3. A novel deep transfer learning architecture (DeneSVM) was also proposed in the dissertation. This suggests a new deep transfer learning approach (DeneSVM) that combines a fine-tuned DenseNet121 backbone for feature extraction with an SVM classifier to improve generalisation. This model innovatively applies layer fine-tuning and regularisation (dropout and early stopping) to a pre-trained CNN to specialise it for posture recognition. The novelty lies in using a depthwise separable CNN architecture for feature extraction and a separate classifier to improve performance. The DeneSVM model achieves state-of-the-art results on a posture image dataset, demonstrating the effectiveness of this hybrid transfer learning strategy.

4. A deep feature fusion technique for posture recognition was introduced in the dissertation. It presents a novel attention-based deep feature fusion method (developed in the OptiMobileX, LitePose-GRU, and VEACTION-CNN models) that combines features from multiple deep learning models to improve posture recognition accuracy. By fusing complementary deep features, this approach optimizes the transfer learning model's performance beyond what a single network can achieve. This novel deep-feature fusion technique significantly improves the accuracy of human posture classification.

5. A novel deep transfer learning architecture (MobileNetV2-DNN) was also proposed in the dissertation. This suggests a new deep transfer learning approach (MobileNetV2-DNN) that combines a fine-tuned MobileNetV2 backbone for feature extraction with a DNN classifier to improve generalisation.

Research Object

The research object is described as the system and methodology for automated human posture classification using deep transfer learning and related AI techniques.

Practical Significance

The practical significance of this dissertation lies in its ability to deliver human posture detection models that are not only accurate (96.4%) but also lightweight, with about 5 million parameters. These models can be efficiently deployed on smartphones, embedded systems, and low-power healthcare devices, enabling real-time posture recognition without dependence on cloud computing. The study demonstrates direct applications in fall detection, ergonomic monitoring, and physiotherapy, thereby supporting the integration of affordable AI into health and safety systems.

Dissertation Statements

Based on the dissertation results, the following key statements are presented and defended in this dissertation:

Statement 1: The proposed hybrid CNN-SVM model (InceptionV3+SVM with LASSO FS) significantly improves posture detection accuracy by combining deep feature extraction with classical classification.

Statement 2: Integrating hyperparameter optimization and image augmentation enhances TL performance by improving model generalization and reducing overfitting on limited datasets.

Statement 3: The proposed hybrid DenseNet121-SVM (DeneSVM) model achieves state-of-the-art human posture detection performance, surpassing existing TL baselines in accuracy and computational efficiency.

Statement 4: Deep feature fusion called OptiMobileX model further improves posture detection accuracy performance by combining deep features from multiple models (deep feature fusion) within a unified model.

Statement 5: The hybridised MobileNeV2+DNN model achieves a high accuracy that exceeds baseline models for posture detection.

Statement 6: The hybrid Lightweight models proposed, called LitePose-GRU, achieved an accuracy of over 90% in posture detection from a limited video dataset.

Scientific Approvals

The dissertation findings have been published in four peer-reviewed journal articles. Two were published in Q1-Q3 quartile journals indexed in the Web of Science (JCR SCIE), 2 were published and indexed in the Web of Science without quartile (JCR SCIE):

In Q1-Q3 quartile journals indexed in the Web of Science (JCR SCIE)

1. [S1; US] Ogundokun, Roseline Oluwaseun; Damasevicius, Robertas; Maskeliunas, Rytis. OptiMobileX: optimizing deep transfer learning model for accurate human posture recognition using a deep feature fusion technique // IEEE Sensors journal. Piscataway, NJ: IEEE. ISSN 1530-437X. eISSN 1558-1748. 2025, vol. 25, iss. 6, p. 9759-9766. DOI: 10.1109/JSEN.2024.3522105. [Science Citation Index Expanded (Web of Science); Scopus] [IF: 4,300; AIF: 4,266; IF/AIF: 1,007; Q1 (2023, InCites JCR SCIE)] [Field: T 007] [Input: 0,334].
2. [S1; CH; OA] Ogundokun, Roseline Oluwaseun; Maskeliūnas, Rytis; Damaševičius, Robertas. Human posture detection using image augmentation and hyperparameter-optimized transfer learning algorithms // Applied sciences. Basel : MDPI. ISSN 2076-3417. 2022, vol. 12, issues. 19, art. no. 10156, p. 1-23. DOI: [10.3390/app121910156](https://doi.org/10.3390/app121910156). [Science Citation Index Expanded (Web of Science); Scopus; DOAJ] [IF: 2,700; AIF: 5,550; IF/AIF: 0,486; Q2 (2022, InCites JCR SCIE)] [Field: T 007] [Input: 0,334].

In publications indexed in the Web of Science without quartile (JCR SCIE)

1. [S1; CH; OA] Ogundokun, Roseline Oluwaseun; Maskeliūnas, Rytis; Misra, Sanjay; Damasevicius, Robertas. Hybrid inceptionV3-SVM-based approach for human posture detection in health monitoring systems // Algorithms. Basel : MDPI. ISSN 1999-4893. 2022, vol. 15, iss. 11, art. no. 410, p. 1-22. DOI: 10.3390/a15110410. [Emerging Sources Citation Index (Web of Science); Scopus; DOAJ] [Field: T 007] [Input: 0,250].
2. [S1; CH; OA] Ogundokun, Roseline Oluwaseun; Maskeliūnas, Rytis; Misra, Sanjay; Damasevicius, Robertas. A novel deep transfer learning approach based on depth-wise separable CNN for human posture detection // Information. Basel : MDPI. ISSN 2078-2489. 2022, vol. 13, iss. 11, art. no. 520, p. 1-22. DOI: 10.3390/info13110520. [Emerging Sources Citation Index (Web of Science); Scopus; Dimensions] [Field: T 007] [Input: 0,250]

International Conference Proceedings

The dissertation outcomes have also been published in four conference proceedings (indexed in Scopus), which serve as additional scientific evidence of the dissertation's contribution. These publications collectively encompass the novel methods and results of the dissertation, having been peer-reviewed and indexed, which underscores the validity and recognition of the work within the scientific community:

In publications indexed in the Web of Science without quartile (JCR SCIE)

1. [P1a; US] Ogundokun, Roseline Oluwaseun; Maskeliūnas, Rytis; Damaševičius, Robertas. Human posture detection on lightweight DCNN and SVM in a digitalized healthcare system // 2023 3rd international conference on applied artificial intelligence, ICAPAI 2023, May 2, 2023, Halden, Norway. Piscataway, NJ: IEEE, 2023. ISBN 9798350328936. eISBN 9798350328929. p. 1-6. DOI: 10.1109/ICAPAI58366.2023.10194156. [Scopus; INSPEC] [Field: T 007] [Input: 0,334].

2. [P1a; CH] Ogundokun, Roseline Oluwaseun; Maskeliūnas, Rytis; Misra, Sanjay; Damaševičius, Robertas. Improved CNN based on Batch normalization and Adam optimizer // Computational science and its applications: ICCSA 2022 workshops: proceedings / O. Gervasi, B. Murgante, S. Misra, A.M.A.C. Rocha, C. Garau (eds). Cham: Springer, 2022. ISBN 9783031105470. eISBN 9783031105487. p. 593-604. (Lecture notes in computer science, ISSN 0302-9743, eISSN 1611-3349; vol. 13381). DOI: 10.1007/978-3-031-10548-7_43. [Conference Proceedings Citation Index-Science (Web of Science); Scopus] [Field: T 007] [Input: 0,250].

In peer-reviewed scientific publications refereed in other international databases

1. [P1b; US] Ogundokun, Roseline Oluwaseun; Maskeliūnas, Rytis; Damaševičius, Robertas; Adebisi, Marion Olubunmi. MobileDNN: leveraging the potential of hybrid MobileNetV2-DNN for advanced image analysis // International conference on science, engineering and business for driving sustainable development goals, SEB4SDG 2024, 2-4 April 2024, Omu-Aran, Nigeria. Piscataway, NJ: IEEE, 2024. ISBN 9798350358162. eISBN 9798350358155. p. 1-5. DOI: 10.1109/SEB4SDG60871.2024.10630298. [Scopus] [Field: T 007] [Input: 0,250].

2. Oluwaseun, O. R., Maskeliūnas, R., Oscar, F., & Oki, O. A. (2025). Enhancing Human Posture Detection with a Hybrid Deep VGG16 and Attention Mechanism Network. *Procedia Computer Science*, 258, 1209-1218.

Publication of the Research Results

The dissertation's experimental findings were published in international journals included in the Clarivate Analytics Web of Science database. The Web of Science database has two (2) impact factors, and Scopus has two (2), for a total of four (4) scientific papers. Furthermore, four (4) international conferences, ICCSA 2022 (Malaga, Spain), ICAPAI 2023 (Halden, Norway), SEB4SDG 2024 (Omu-Aran, Nigeria), and ICMLDE 2024 (Dehradun, India), featured the dissertation's experimental results. The publication summary and the corresponding methods are presented in Table 1.

Table 1. Summary of publications and the corresponding methods

Paper #	Title of Article	Journal	Methods	Datasets
1.	OptiMobileX: Optimising Deep Transfer Learning Model for Accurate Human Posture Recognition Using a Deep Feature Fusion Technique	IEEE Sensors Journal	OptiMobileX: Deep feature concatenation mechanism, feature extraction, deep transfer learning (DTL), MobileNetV2, Xception, weight pruning, quantisation, dropout, early stopping, learning rate, and parallel computation.	MPII Human Posture
2.	Human Posture Detection Using Image Augmentation and Hyperparameter-Optimized Transfer Learning Algorithms	Applied Sciences	CNN transfer learning, image data augmentation, hyperparameter optimization, AlexNet, VGG16, CNN, and MLP	MPII Human Pose
3.	A Novel Deep Transfer Learning Approach Based on Depth-Wise Separable CNN for Human Posture Detection.	Information	DeneSVM: DenseNet121 deep transfer learning, Support Vector Machine (SVM), regularisation, early stopping, dropout techniques	Silhouettes
4.	Hybrid InceptionV3-SVM-Based Approach for Human Posture Detection in Health Monitoring Systems.	Algorithms	Deep convolutional neural network, SVM, least absolute shrinkage and selection operator (LASSO)-based feature selection, data augmentation, dropout, and early stop.	Silhouettes

Author's Contribution

The author first gathered and analysed the data used in this dissertation. The author developed and conducted every implementation in Paper 1 (P-I). The supervisor, Prof. Dr. Rytis Maskeliūnas, and the co-authors assisted with project management, validation, data gathering, software development, resource allocation,

technique implementation, and conceptualisation. Simultaneously, the author and supervisor prepared, reviewed, and edited the initial draft for publication. (Prof. Dr. Rytis Maskeliūnas).

The author in paper 2 (P-II) also developed and performed all the experiments. On the other hand, conceptualisation, methodology, software, data curation, formal analysis, investigation, and validation were supported by the co-authors and the supervisor (Prof. Dr. Rytis Maskeliūnas). Preparation of the initial draft, examination, and proofreading of the published article were carried out by the author and the supervisor (Prof. Dr. Rytis Maskeliūnas).

In Paper 3 (P-III), the author planned and performed all experiments. The conception, methodology, software, resources, data curation, formal analysis, investigation, and visualisation of the work were addressed by the author. Checking and approval of the published article were done by co-authors. Preparation of the original draft, review and editing of the published article were written by the supervisor (Prof. Dr. Rytis Maskeliūnas) and the author.

In Paper 4 (P-IV), the author conducted all experiments and designed them. The co-authors and supervisor (Prof. Dr. Rytis Maskeliūnas) assisted with the conceptualisation, methodology, software, resources, data curation, visualisation, and validation. At the same time, the original draft was prepared, reviewed, and edited by the supervisor (Prof. Dr. Rytis Maskeliūnas) and the author.

Dissertation Organisation

This doctoral dissertation consists of five main chapters, along with additional supporting sections and appendices.

Chapter 1: Introduction presents the background and significance of the research, defines the problem statement, and identifies the study's purpose, objectives, and scope. It also sets out the scientific novelties, the object of research, practical applications, defended theses, and the author's contributions.

Chapter 2: Literature Analysis provides an in-depth overview of the existing literature on human posture recognition, including classical computer vision methods, deep learning models, transfer learning, feature fusion, lightweight models, and hybrid deep learning-machine learning approaches. It presents research gaps and critiques the suggested methodology.

Chapter 3: Research Methodology describes the datasets, data preprocessing processes, model architectures, experimental configuration, and performance measures. It also outlines the proposed models, optimisation methods, and implementation phases.

Chapter 4: Results and Discussion section presents and describes the experimental results, with comparisons to the baseline and state-of-the-art, for the performance of the proposed methods. Describes implications of the results, strengths, and possible weaknesses.

Chapter 5: Conclusions summarises the principal findings, contributions, and practical implications of the research, and suggests recommendations for future work.

The dissertation is concluded with a Lithuanian abstract, a list of references, photocopies of relevant scientific articles, the author's curriculum vitae, acknowledgements, and annexes containing additional material.

1. LITERATURE ANALYSIS

Literature analysis of this dissertation is carried out in this chapter. The organisation of the chapter is as follows:

Subchapter 2.1 presents an overview of a depth-wise separable convolutional neural network (CNN)-based deep transfer learning approach for detecting human posture (HP). Subchapter 2.2 provides an overview of human posture detection based on image augmentation and hyperparameter-optimised transfer learning algorithms. In subchapter 2.3, an overview of human posture detection (HPD) in health monitoring systems is presented using a hybrid LASSO+InceptionV3+SVM approach. Subchapter 2.4 provides an overview of deep feature fusion techniques for optimisation of deep transfer learning models for accurate recognition of human posture. In Subchapter 2.5, an overview of an improved CNN is presented, based on batch normalisation and the Adam optimiser. Subchapter 2.6 provides an overview of lightweight DCNNs and SVMs for human posture detection in a digitalised healthcare system. In sub-chapter 2.7, an overview of a hybrid MobileNetV2-DNN for advanced image analysis is presented. Subsection 2.8 provides an overview of a hybrid deep VGG16 and attention mechanism network for enhancing human posture detection.

1.1. Trends and Gaps in Posture Detection Research

The existing literature on human posture detection reveals several clear trends and persistent knowledge gaps. Deep learning-based approaches have become the standard for posture recognition, superseding conventional handcrafted feature approaches. Convolutional neural networks (CNNs) have consistently yielded state-of-the-art results in human pose classification across numerous research works. Transfer learning (TL) using large image datasets (e.g., ImageNet) to fine-tune CNN models on posture data is a common strategy to address data scarcity. This reflects a common tendency: scientists utilise pre-trained deep architectures (e.g., VGG16, ResNet, Inception, MobileNet) to compensate for the typically limited size of posture datasets. Indeed, one of the most significant challenges in the field is the scarcity of labelled posture images or video frames for training, which can lead to overfitting and poor generalisation. Data augmentation (flips, rotations, shifts) and TL aim to address this by artificially increasing the dataset size and reusing learned visual features.

Another trending activity is the construction of light and hybrid models for edge deployment. Posture monitoring is typically required on mobile or wearable devices with limited computational capabilities, so efficiency takes centre stage in recent literature. Methods such as MobileNetV2 (utilising depth-wise separable convolutions) or model compression techniques are typically employed to reduce model size and inference time. Some work suggests hybrid models that combine deep and conventional machine learning, e.g., a CNN for feature extraction followed by an SVM classifier, to trade accuracy for reduced complexity. The hybrid CNN+SVM

solution has performed better in generalisation on small datasets by leveraging SVM’s power with limited samples. Therefore, feature selection techniques (e.g., LASSO regularisation) are used within deep models to reduce overfitting and improve interpretability by selecting the most informative posture features.

Although notable progress has been made in this area, important gaps remain in the literature. To begin with, much of the existing research concentrates on a limited set of predefined static poses or a single activity. Far less attention has been given to detecting subtle, transitional, or progressive movements, such as a slight loss of balance that eventually leads to a fall, particularly when working with sparse or low-resolution video data.

Moreover, while several studies report high accuracy rates in controlled laboratory environments, often exceeding 90 per cent, there is generally little discussion of how these models perform in real-world conditions. Issues such as environmental variability, occlusion, lighting changes, and unconstrained subject behavior are rarely addressed in depth, leaving questions about the robustness and generalizability of these systems in practical deployment settings. Features such as occlusions, varied camera perspectives, and crowded backgrounds continue to pose challenges for posture detectors. For instance, models trained on ideal data may not perform as well on occluded limbs or angled shots, a case of generalisation gap. Temporal modelling is another issue that must be addressed. Most approaches treat posture detection as frame-by-frame image classification, which overlooks temporal information. Human posture in video inherently involves sequential dynamics, and non-time-series data or motion-cue-based approaches may miss nuanced changes in posture. The integration of video sequence models (e.g., LSTM or 3D CNN-based) is an open research challenge.

Moreover, while deep models are effective on massive datasets, few studies have examined semi-supervised or unsupervised learning for posture detection that can utilise abundant unlabelled video data. Few studies have proposed semi-supervised learning or GAN-generated augmentations, but this approach has yet to gain popularity. There also appears to be a trend toward multimodal posture detection (e.g., fusing video with depth-sensor or wearable sensor data). However, research along these lines is still in its early stages and has not yet been incorporated into existing posture detection paradigms.

Overall, the literature indicates a clear trend toward deep transfer learning and blended methods to address the challenges posed by limited data and resources in posture detection. Accuracy has improved significantly (with most methods achieving over 90% on benchmark datasets), but achieving reliability in unconstrained environments and enhancing the diversity of detectable postural nuances remain open problems. These challenges motivate the contributions of this dissertation. The following subsections provide a detailed overview of representative related works, categorised by approach, to illustrate how each addresses (or sidesteps) these challenges.

1.2. Depth-Wise Separable CNN for Human Posture Detection

The era of internet media has led to a rapid increase in the volume of imagery and video content. There is significant room for the computer vision (CV) industry to leverage human posture data to develop strategies that improve monitoring systems, thereby helping prevent the onset of ailments such as cardiovascular disease. One can forecast a person's posture using the human posture categorization approach, with a guide being the person's posture history or activities. In essence, intelligent computer system models can provide support once the subject activity has been detected and identified. Deep learning (DL) models have demonstrated success in HPD in recent research studies [18, 19]. HPD focuses on detecting poses from video streams or static images. Applications of the use of image-based posture categorizations include, among others, medical system development, autonomous technologies, posture inappropriateness detection, and scenario assessment. Processing HPC data is a monumental task and poses various challenges, including cluttered backgrounds, overlapping body parts, and occlusion. 2D human pose estimation (HPE) is the sole task in human posture detection and classification. Human posture details are an essential approach to estimating human poses [18]. DL is particularly robust at solving most computer vision problems because approaches that are effective for training tasks when the provided datasets are large [20-26].

Deep transfer learning (DTL) is a major form of DL, commonly used to solve a wide range of problems, including face recognition, image detection, object recognition, and image classification. Four DTL approaches were employed in this work to establish a method for human posture classification. Human position was predicted in [27] using a combination of graph models and a CNN. A two-stage structure composed of two modelling approaches is illustrated: one is a multistage CNN approach that extracts the required image features, and the other is a prediction approach that predicts posture based on the information obtained by the CNN [28]. For understanding the methodology and limitations of most HPE approaches, the literature has been reviewed. Findings derived from previous techniques have demonstrated the need for advanced algorithms to address current issues in human posture estimation. Therefore, this dissertation proposes a deep transfer learning technique for human posture detection.

1.3. Image Augmentation and Hyperparameter-Optimized Transfer Learning Algorithms

The availability of large-scale category-level training sets, such as ImageNet [29], and the efficacy of methods for reducing overfitting [30] have enabled deep CNNs to outperform all other solutions in the ImageNet Large Scale Visual Recognition Challenge 2012 [31]. The core contribution of CNNs to image classification and computer vision is their ability to train the entire system from raw pixels to the final categories, thereby avoiding the manual assembly of a feature

extractor. Although CNN has some significant disadvantages, including (i) requiring many labelled training samples for learning the weight parameters, and (ii) the use of a powerful Graphics Processing Unit (GPU) to accelerate the learning process.

Figure 1 illustrates that the standard deep CNN structure, as described in [32], comprises 5 convolutional layers, 3 fully connected layers, and an end-to-end softmax classifier with approximately 60 million parameters. Improved performance can be achieved with deeper CNNs that have more parameters, such as 16- and 19-layer [33] models and a 22-layer [34] model. It is a huge challenge to train a deep CNN on a small dataset to achieve the same performance as on a large dataset. One such practice for using deep CNNs with small datasets is transfer learning, in which a pre-trained CNN's classification head is replaced, and the model is fine-tuned on the target dataset. While the given practice is an effective one that can be used to solve a problem, several questions remain unanswered, including (i) what is the optimal practice to select a fine-tuning method? (ii) What are the appropriate hyperparameters (e.g., learning rate) to use for network finetuning? (iii) What else needs to be done to enhance network performance [35]?

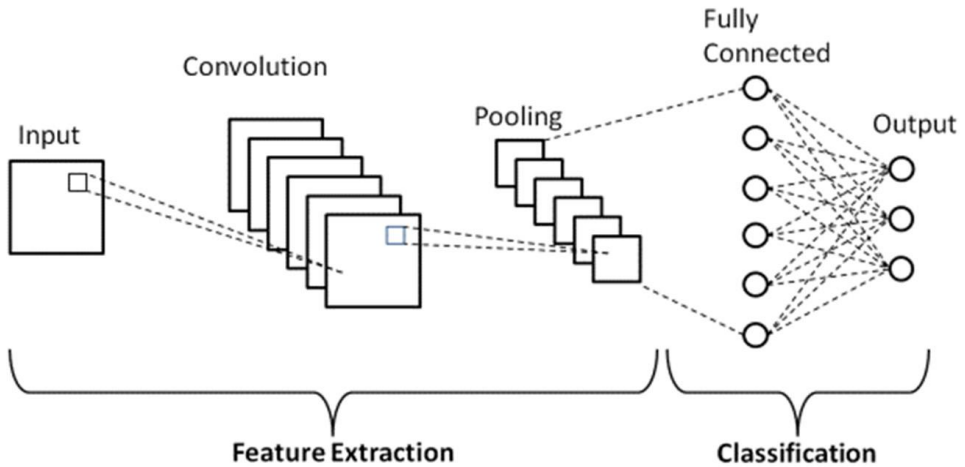


Fig. 1. Deep CNN Architecture [36]

Several research papers have employed machine learning and deep learning techniques to address problems in human posture classification [37, 38]. However, to the best of current knowledge, none of these approaches has employed hyperparameter optimization (HPO) to systematically select hyperparameters that maximize classification accuracy in machine learning or deep learning models. There are different sets of hyperparameters for each classification task because the medical condition is complex. Machine learning approaches are most frequently used in medicine for computational decision-making, especially when carefully analysing medical information to identify hidden relationships or anomalies that exceed human capabilities [39]. The combination of decision support systems (DSS) and artificial intelligence (AI) methods has been utilised in medical decision-making to analyse complex biomedical images and signals, such as echocardiograms [40], magnetic

resonance images [41], and computed tomography images [42]. As a matter of routine, medical information, records, and images, however voluminous and complex, are intended to be handled quickly and with greater precision.

However, implementing such algorithms is not straightforward; secondly, it becomes even more challenging to optimise the algorithm's precision to its maximum and minimise runtime. Even more overhead may be required for hyperparameter optimisation because machine/deep learning algorithms typically require multiple training and testing cycles to optimise hyperparameter settings. The central issue in most such methods, however, is feature selection, which particularly addresses underfitting and overfitting. First, preventing both overfitting and underfitting substantially improves model performance across the training and test datasets. Irrelevant features and suboptimal model configurations often lead to overfitting, in which the model captures noise in the training data rather than learning generalizable patterns [43]. The inability to generalise in most models reported in the literature highlights the importance of optimising models to perform optimally with new medical data [44]. Hyperparameter optimisation is often unreported because it is overlooked during the evaluation of decision support systems [45]. Secondly, there was limited or no scientific evidence available on how hyperparameters affect the accuracy and stability of AI models used in decision support [46]. Data augmentation techniques help improve machine learning performance on small datasets [47–51].

The method includes creating synthesised images, which will help extend the deep learning model's capacity to generalise, thereby reducing overfitting [47,48]. Selecting which transformation operations (e.g., rotation, flip, and zoom) to apply to images is part of the problem in data augmentation [52–56]. This problem can be regarded as an HPO problem when machine learning is employed [57–63]. Plant classification is among the applications of HPO and data augmentation [64], but there are more, such as transmission line inspection [65], COVID-19 diagnosis on chest radiographic imaging [66], SVM classification optimization [67], prediction of house prices in the real estate [68], and training neural networks optimization [69]. Among the deficits of past studies is a lack of recommendations on maximizing deep learning techniques for human posture image classification. CNNs have been well-studied as a deep learning technique in computer vision [70–72]. One of the most significant reasons CNNs can be utilised in machine learning applications is their ability to automatically extract features from images [73,74]. Therefore, the requirement is to describe the usage of deep learning methods and data augmentation in HPO for image classification tasks.

1.4. Human Posture Detection in Health Monitoring (Hybrid InceptionV3-SVM)

A healthy life requires sacrifices, including maintaining a straight back. Limb position and the way the body is carried constitute posture. New technologies have shifted human activity toward more sedentary behaviour, thereby reducing physical exercise and mobility [75]. The most well-known symptom of prolonged sitting at

work or while doing homework is muscle weakness, and it can be difficult to overcome. People have experienced various problems due to poor posture. The neck, back, shoulders, and spine are common areas affected by musculoskeletal disorders. In the modern era, illnesses associated with improper posture are common across all age groups. Contributory causes of postural problems are inadequate exercise, improper or asymmetric sitting posture, and passive office behavior [75]. In [76], electromyogram signals from 12 patients were compared to assess the influence of computer desk height on neck and upper limb electromyogram activity.

In [77], a motion capture device was used to assess how often the heads of 18 smartphone users were bent during various tasks. The experimental findings were attributed to the major cause of neck pain. In [78], 126 students were enrolled in a study whose findings revealed poor neck protrusion and neck pain, including neck impairment. According to Ruiz et al. [79], sitting postures influence breathing and electrocardiographic patterns. Additionally, research on unsatisfactory head positions reported in [80] found that they reduce respiratory performance compared with neutral head posture. In [81], experiments were conducted on 88 schoolchildren, and the findings established the substantial impact of slouched seating on their breathing. Maintaining proper sitting and standing posture is essential for a healthy, infection-free life.

Posture can be identified using various technologies, such as wearable [76,82] and non-wearable [80,83] devices. Non-wearable devices have also been utilised for posture monitoring, e.g., the Vicon MX motion analyser system [80] and the Kinect depth camera [83]. Posture monitoring using a non-wearable technique is highly effective; however, the camera's field of view limits its use. Additionally, it requires a significant upfront investment to install advanced posture-monitoring sensors in a smart home environment. Wearable posture detection systems allow freedom of movement while monitoring posture. Posture was tracked by an electromyogram in Kang [76]. In [82], Back posture was monitored using a smart shirt equipped with two inertial sensors that provided 9 degrees of freedom.

Data collected from various sensor technologies can be processed for classifying different postures using machine learning methods, such as K-nearest neighbours (KNN), support vector machines (SVM), and decision trees (DT) [79, 83, 84]. In [78], two positions were classified using DT, KNN, and SVM on ECG and breathing motion, achieving an accuracy of approximately 99.5%. In [83], the max-margin classifier, a traditional SVM technique, was applied to position detection in a video database, achieving an accuracy of approximately 88.67%. In [84], when 13 piezoresistive sensors were placed on a chair, the accuracy of classifiers such as DT, KNN, and SVM for four sitting postures exceeded 98%. Deep learning has also been used to classify human activity from time-series data [85–89]. This dissertation discusses various strong, systematic classifiers for deep learning. Position detection is found in numerous applications, including healthcare, virtual worlds, animation realism, internal and external surveillance, entertainment, and posture recognition [90–97]. Furthermore, position recognition may be applied in the human-to-home interface paradigm. It is of paramount importance to deploy talent to monitor and

surveil older and vulnerable individuals to promote healthy, independent lifestyles, given the increasing number of ageing citizens and limited medical resources [98–100]. Being upright is paramount for maintaining good health. Pose refers to the placement of the arms and legs and how the body is held. Technological advancements have contributed to a sedentary lifestyle (i.e., prolonged sitting at work or during study) by reducing mobility and physical activity [101–105]. A sedentary lifestyle is unhealthy for overall well-being, and poor posture can lead to discomfort in the neck, shoulders, and back if left uncorrected.

The DCNN method proposed in this dissertation aims to improve performance in human posture recognition and in detecting sitting, standing, bending, and lying postures. Moreover, an outline of current approaches for detecting human posture has been provided. Sitting and standing postures can be easily detected when people can monitor their actions, e.g., sitting for long hours or standing for a few hours. Several studies in the literature have explored the development of posture models. Examples include smart technologies and mobile systems used in smart cities to monitor human health. Thus, multisensor and LoRa (long-range) technologies are employed for posture detection in this dissertation. LoRa WAN technology is inexpensive and possesses a large communication range. The two technologies help create wearable clothes that are comfortable in any position. This dissertation employs multiprocessing due to LoRa's low transmission frequency and small data transfer size. Sliding windows are utilised in multiprocessing, while Random Forest (RF) is employed for data processing, feature extraction, and feature selection. Accuracy and performance were improved by utilizing three testers of a 500-group dataset [106]. Body language, including gestures and postures, is a form of nonverbal communication. The current dissertation utilizes advanced body tracking technologies and augmented reality to identify static posture.

Furthermore, Kinect body position sensor-based unsupervised machine learning is used to recognize group collaboration and learning [107]. Posture detection facilitated proper yoga practice. Posture recognition is a difficult problem because it is a real-time task, and the available datasets are inadequate. To address this challenge, a large dataset comprising at least 5500 images of varied yoga postures was developed. Implementation of the TF-POSE estimation technique in the determination of posture helps in the visualization of the human body skeleton in real-time. The tf-pose skeleton is used as a feature to implement various machine learning techniques and to extract joint positions of the human skeleton.

The RF model [108–110] achieves the highest accuracy among models such as SVM, KNN, logistic regression, DT, and NB. Incorrect sitting postures can also harm both physical and psychological well-being. The posture training system enables the collection of information on sitting and stretch postures. Subsequently, a smart cushion that combines AI and pressure sensors for posture detection. Supervised machine learning models trained to produce better outputs are trained on more than 13 postures [111]. The chair's pressure sensor prohibits unhealthy sitting postures. The comparison is distinct from RF and DT in the context of recognising sitting postures. The RF classifier [112] performs optimally. Sitting postures are

supplemented using sitting posture monitoring systems (SPMSs). Here, six different sitting postures are considered. Next, ML algorithms such as linear SVM, SVM with an RBF kernel, RF, NB, DT, QDA, and LDA are utilised for the body weight ratio, as determined by SPMS. The precision of SVM with an RBF kernel is higher than that of other algorithms [113]. Additionally, sophisticated equipment can be used to detect a person's posture when seated in a wheelchair. Data are collected from an array of sensors by using the neighbourhood rule (CNN). The Kennard-Stone algorithm is used to normalise them, and principal component analysis is used to reduce their dimensions. In conclusion, KNN is used to process preprocessed and balanced data. Although this study utilised a remarkably small dataset, the results are surprising [114].

1.5. Deep Feature Fusion Technique for Transfer Learning Optimization

Human posture recognition (HPR) is a crucial area of research in computer vision for human behaviour recognition systems, and it plays a vital role in identifying human behaviour. However, most images require preprocessing before feature extraction due to the complexity of the human body, which makes feature extraction difficult and reduces recognition efficiency [115], [116]. As humans' desire for sophisticated means of engaging in communication continues to grow, the evolution of the human community has brought about a paradigm shift in the traditional method of communication (physical contact) between humans and computers.

In recent years, HPR has evolved into an area of intense research and a channel by which humans and computers interact [117], [118]. The main objective of the HPR domain is to analyse and process images of human postures [119], [120]. However, the method's requirements for achieving high-quality images make it challenging, as image quality is degraded by the underlying environment and occlusion [121], [122], thereby reducing the precision of human gesture identification. A posture identification methodology that differs from the previous method, centred on peripheral electrodes, requires humans to wear sensors to record body motion and mobility data. However, this approach has several limitations, including sensitivity to sensor reliability issues, inconvenience, failure to meet users' expectations, and extensive implementation. There is a continued increase in the amount of available information as human civilisation evolves. Standard machine learning approaches struggle to categorise large datasets, leading to recurrent substandard classification results [123], [124], [125]. There are several proposals for posture detection methods [126], [127] that utilise sensor musculoskeletal data.

In this dissertation, a technique is described that uses DTL for HPR, exhibiting high capacity in complex data classification. DTL has the capabilities for robust self-learning and nonlinear modelling. DTL utilises its full self-learning capacity to identify image attributes during initial image processing, thereby overcoming the limitations of conventional FE techniques. Moreover, the extracted information more accurately reflects the input characteristics. The effectiveness of DTL stems from its

weight-sharing properties and its ability to handle nonlinear solid transitions; these two characteristics enable it to capture finer image details and produce excellent results.

Features extracted from RGB images are often used by most traditional posture recognition methods to describe human visual and 2-D posture information [128], [129]. In [130], an algorithm that employs human contour samples was proposed to obtain human edge templates. The proposal included a similarity-based and gradient-descent method for posture estimation. In [131], a convex programming-based matching scheme for posture recognition was presented. The method's efficiency surpasses that of other methods, including graph cuts and belief propagation, for the object-matching problem with a large search range.

However, this method is highly sensitive to various interferences, including environmental factors and image illumination. In [132], an algorithm was proposed that uses artificial neural networks to automatically detect human poses in a single image. However, this approach, which uses static image features to determine the human skeleton, requires substantial computational effort for feature extraction. In [133], a data-driven MCMC technique was presented for 3-D human pose estimation from static images. In [134], a review of recent progress and challenges in estimating 3-D human pose was conducted.

An evaluation of the approaches was conducted using extensive experiments on a synthetic dataset, with input modes and vital features introduced separately. The mitigation of integral ambiguity in shape and variance associated with implicit heatmaps was proposed in [135]. The accuracy of AI posture identification has significantly affected the efficacy of human-computer interaction [136], [137], and only a few studies have addressed this issue. This dissertation proposes a hybrid DTL model (a concatenation of MobileNetV2 and Xception, named OptiMobileX) for the HPR using the MPII dataset [138].

Most previous studies have focused on movement recognition compared to posture recognition, with an increasing emphasis on continuous activities. In addition, there is a limited supply of datasets and techniques for posture recognition. In this dissertation, a model for improved posture recognition is introduced that concatenates two lightweight DTL models for HPR, integrated into the MPII dataset, which comprises four posture classes.

1.6. Improved CNN Architectures with Batch Normalisation and Adam Optimiser

Image segmentation and classification, such as knowledge acquisition using CNNs, have been widely employed for object recognition and are widely accepted [139]. As a result, enhancing CNN performance is a broad research area, as discussed in [140]. The entire procedure is managed by the CPU, such as data preparation. On the other hand, the graphics processing unit (GPU) handles convolution computation in the neural network cell and the execution rate of the full-joint layer-integration cell [141]. Although the neural network's learning rate has improved in recent years, CPU-

and GPU-based data preprocessing and preparation still consume significant time, and a weak GPU platform is more likely to experience execution interruptions. The efficiency and training procedures of deep CNNs have been improved and simplified by using batch normalisation (BN) in DL [139]. As networks become deeper, training them becomes more challenging due to multiple issues that arise during training. Further scaling and dynamic operations are used in regularising the output of the convolutional layers (CL) after the convolutional layers (CL) and before the activation layers (AL). They are introduced after the output is delivered to AL, then to the next CL [139]. The test data accuracy improves simply by adding this layer; the network also converges faster than an equivalent model without BN, which reduces internal covariate shift and yields better results.

Experiments have been conducted on the impact of BN on various established networks, including the Dense Network [142], Residual Network [142], VGG Network [143], and Inception (v3) Network [144]. There is no straightforward evidence of incorporating batch normalisation layers into the architecture. Adding BN layers to shallow networks improves performance compared with networks without BN, and training deep networks requires BN. Within this dissertation, it is indicated that the application of BN improves Exponential Linear Units [145–147] for Residual Networks better than Rectified Linear Units for all the other networks. In [148], operator theory, exemplified by AF, was merged with interpolation operator theory and the multivariate all-out product on the CNN platform, with a focus on convergence speed in CNNs.

Although the study provided a full derivation of the mathematical formula, it did not yield experimental results. In [149], carefully supervised nets were proposed by placing companion objective functions (COFs) within the overall hidden layers, along with an overarching objective function in the output layer. The widespread use of companion goal functions provides an additional constraint or regularisation for the learning process. In this regard, the introduction of a COF is crucial to the success of this method, which is challenging to achieve. The CNN approach gave a new technique of image feature extraction. Drop-activation was proposed in [150] as a regularisation method that adds randomisation to the activation function. Drop-Activation employs a deterministic network with modified nonlinearities for prediction, in which nonlinear activations are randomly dropped during training.

In [151], a recent stochastic regularisation technique called ShakeDrop was presented, which can be combined adequately with ResNet and its variants, provided that the residual blocks are batch-normalised. Their efficiency was confirmed through experiments on the CIFAR-10 and CIFAR-100 datasets. ShakeDrop also performed exceptionally well compared to other methods across both datasets used in the implementation. In [152], a novel Residual Network (RN) with the Residual of Residual (RoR) architecture was introduced. The study demonstrated that the new architecture established a new baseline in image classification performance on the CIFAR-10 and CIFAR-100 datasets. Additionally, there is a growing trend toward the approach as a superior complement to the RN family, given its promise. In [147], the exponential linear units (ELUs) were presented as a means to enable faster, more

efficient learning in deep neural networks. The network drives the mean activation towards zero, assisted by ELUs with negative values. In this way, ELUs reduce the distance between the unit and the natural gradients, thereby improving learning. Some experimental evidence suggests that ELUs outperform most activation functions on specific computer vision datasets. Besides comparing the performance of ELU networks with ReLU networks trained with BN.

Furthermore, ELU networks achieved comparable performance in significantly fewer epochs than a comparable ReLU network on the ImageNet dataset. In [153], a recurrent CNN was proposed for (static) object recognition. The idea was to enable feedback connections in each feedforward convolutional layer of the CNN. This structure enabled modulation of units within the same layer, thereby enhancing the CNN's ability to learn statistical patterns in object contexts. The depth is enhanced by sharing weights across layers via recurrent connections, without increasing the number of parameters. The superiority of RCNN over CNN for object recognition was determined experimentally. RCNN is superior to existing models, such as CNN, for object recognition with fewer parameters on carefully selected benchmark datasets. Results often improve with increased variable count. Although many attempts have been made towards enhancing the performance of DL models [154], several challenges are encountered while applying the DL models to image classification, including (a) Lack of required amount of dataset to train by DL, (b) Imbalanced dataset, (c) Interpretability of dataset, (d) Scaling of uncertainty, (e) Catastrophic forgetting, (f) Compression of model, (g) Overfitting, (h) Vanishing and exploding gradient problem, (i) Under specification. To address these issues, this dissertation proposed a more effective activation function by incorporating batch normalisation and dropout layers between the fully connected and output layers of the CNN, aiming to improve convergence rate. The incorporation of the Adaptive Moment Estimation (Adam) optimiser into the gradient descent approach reduces overfitting caused by increased repetition, thereby minimising cumulative error and improving training speed.

Additionally, Batch Normalisation helps in mitigating internal covariate shift. There is scope to apply a higher learning rate with a regularising impact. In this dissertation, an improved CNN (ICNN) algorithm (ICNN-BNDA) based on batch normalisation, a dropout layer, and the Adaptive Moment Estimation (Adam) optimiser is proposed, following an analysis of the challenges in implementing CNNs for feature extraction. A seven-layered CNN structure with LeakyReLU unit as ICNN-BNDA exploits the activation function to eliminate the gradient problem in an effort to speed up convergence, and the overfitting problem is alleviated by the solution through the use of an Adam optimiser, which was attained by incorporating a BN layer in the overall connected and output layers for cross-entropy minimisation.

1.7. Lightweight DCNN and SVM for Digital Healthcare Posture Detection

Human posture Classification (HPC) has numerous benefits, including applications in healthcare systems, indoor and outdoor tracking, virtual worlds, and virtual environments for animation and entertainment. Moreover, the advantage of posture detection can be grasped by a home interface system [155-160]. It is necessary to suggest a way for real-time management of the elderly with mental issues, considering the growing population of the elderly and paucity of medical centres [161]. A healthy, balanced lifestyle depends on good posture. The advent of the internet has been detrimental to people's lifestyles, reducing physical activity [162]. In [163], microcontrollers and inertial devices were used to investigate posture recognition. In [164], a CNN was used to classify real-time poses in yoga, and skeletonisation of basic poses was systematised for human prominent-point identification using MediaPipe. Similarly, an object recognition method was used in [165] to recognise yoga positions for an extended period. A hybrid model was also developed using InceptionV3 and SVM to identify human position [160]. Muscle pain is commonly caused by long hours of sitting while working or studying. An individual's health can be improved through a healthy lifestyle; however, poor posture, neglect, or improper posture can lead to discomfort in the lumbar region, limbs, and spine. Thus, it is necessary to manage people's locations in order to keep them healthy while they study or work.

According to the research in this dissertation, a deep learning-based model is more powerful for stance detection. The InceptionResNetV2 and InceptionV3 approaches were compared because they are among the most popular techniques for image classification. Certain computer vision tasks, such as object detection, object tracking, image segmentation, and classification, have been performed with remarkable accuracy using DL algorithms [166]. With promising results from CNNs, medical imaging studies have been interested in DL-based methods, which are currently used to design computer-aided design (CAD) systems to identify cancer. TL is a technology that leverages a pre-trained model to address numerous challenges [167]. Building a large-scale computational model requires an enormous amount of data and computational resources. Data augmentation (DA) [168], a key concept in this dissertation, expands a limited dataset by applying various transformations to generate new data. Numerous benefits are gained from it, including accelerating convergence, reducing over-adjustment, and enhancing generalisation capability. This dissertation employs a range of DA approaches, including rotation, horizontal flip, width and height shifts, zoom, and shear. Furthermore, the dissertation presents novel methods for classifying human posture images. MobileNet and Xception are presented as 2 thin DCNN approaches, with SVM used as an FCL to achieve adequate detection accuracy (Acc.) in HPC. The lack of datasets is overcome by applying the DA approach to the MPII human posture images.

Additionally, the TL method is used to conserve time, money, and computer resources. Image recognition and analysis in 2D and 3D space were presented and greatly enhanced HPC research [169]. Image-based monocular body pose estimation

was thoroughly reviewed in [169] with the DL-based approach. The review highlighted the potential for exploiting various posture estimation methods, including body model-centred and body-free approaches, pixel-level assessment, body joint location mapping, and heatmap modelling. The potential of CNNs to deliver current results warranted the use of DL in HPC, and the effectiveness of AlexNet sparked a renaissance in image classification [170]. Research on body-joint assessment reported in [169] was extended in [171]. In [172], human motion was estimated in a pose-centred setting, duplicating the efforts in [169, 171]. HPC was investigated in [173], and the following were identified as the primary HPC approaches: generative, template-centred, and discriminative model-centred approaches. In [174], a hybrid stance recognition approach was introduced by combining conventional ML methods with deep neural networks (DNNs). In [175], OpenPose was employed for KeyPoint recovery, and a hybrid CNN-LSTM layer was used as an alternative for classification. Approximately 88 samples of 6 Yoga postures were used to develop the technique. In [176], the demonstration of ensemble deep methods usage in challenging home conditions with mixed origins was demonstrated. Some interesting results were reported on the usability of fitness devices in home conditions by older people. A Deep Hierarchical Recurrent Network Based on MobileNetV2 was proposed in [177] to assess sitting position and achieved an accuracy of 91.47%. In [178], human physical motion generative design methods were discussed.

1.8. Hybrid MobileNetV2-DNN for Advanced Image Analysis

Recently, surveillance systems have been developed through the increasing application of Artificial Intelligence (AI) across various industries [179, 180]. These systems utilise deep learning to enhance visual computing and advance image processing technology [181, 182]. The perfection of deep learning enables its algorithms and models to be used for accurate identification, monitoring, and detection of human body motion postures with outstanding results [183, 184]. Extensive research has been conducted on the fundamental principles of human body motion, posture detection, and tracking using advanced deep learning techniques [185]. Conventional surveillance technology, frequently employed in military operations and in applications for border protection and customs defence, is closely linked to the surveillance and security domain. Identification and tracking technologies based on deep reinforcement learning may assist with certain manual tasks [186-189]. Applying deep reinforcement learning to human motion posture recognition in sports materials could provide a valuable service for sports training observation [190]. Simulating the human body's 3D posture improves the uniformity of competitors' motion in skiing and short-track speed skating.

Detecting human motion posture in a competitive environment can enhance the audience's visual experience by restoring and reconstructing live scenery and human motion targets [191, 192]. In [193], an automated healthcare system was implemented for identifying laying positions. For easy searching, each picture is divided into

foreground and background. The articulated body technique was utilised for posture recognition using foreground masks. The PASCAL09 dataset and the FT_Lying datasets were used for training the proposed approach, which includes publicly inaccessible interior photographs. In [194], Microsoft Kinect sensors and 3D point cloud clustering were utilised to analyse fast point feature histograms, normal local surfaces, and surface entropies. Feature vectors were obtained using different classification techniques, including AdaBoost, random forests, and SVMs, along with geometric features. However, none of these classifiers resolved the issue of partial blockage. In [195], a technique was proposed that utilises Microsoft Kinect sensors and background subtraction to generate 3D-bounded boxes for identifying human objects using depth information. The fall and non-fall posture classification is based on the velocity characteristics of 3D-bounded boxes.

In [196], an approach was presented that utilises joint data from Microsoft Kinect sensors and the human skeleton to identify the posture of a fall. Several of the techniques included conventional methods for extracting image features, such as HOG and Gabor filters [197], local binary patterns, and colour histograms [198]. Several noise sources, including background noise, clothing, accessories, and low-light conditions, hindered the smooth execution of the feature extraction procedure for images and sequences. The effectiveness of these algorithms is significantly impacted by these issues [199].

1.9. Hybrid Deep VGG16 and Attention Mechanisms for Posture Enhancement

Human posture detection (HPD) involves identifying, detecting, and classifying human features in an image, specifically their presence, location, and posture type. HPD has found wide applications in health, security, and advertisements [200, 201]. Training and testing models are performed under a similar camera view by nearly all methods proposed to date. However, most practical applications would require that any model learned from a single view generalise to the other [201]. Posture classification is challenging due to variations in posture within and across views. Questions that should be answered include: “How many distinct postures are present in each scene, and how do these postures help in abnormal event detection, like poor posture?” [202]. Current methods for human posture detection, derived from walking or standing subjects, are not easily generalised to diverse postures. Traditional methods typically use background subtraction to compute human areas for statistical features. However, their drawback is the presence of background noise and changes in viewpoint.

Several methods have been proposed to mitigate these issues by computing statistical measures from the normalised directional histograms. Or, other methods employed skeleton or depth data, combined with deep learning techniques, to bypass background subtraction and camera viewpoint transformation [203, 204]. In contrast to these approaches, which define a limited set of postures for training and testing, this dissertation focuses on continuous posture variation, as observed in life-logging activities, such as those at home. However, several activities involved subjects [205].

In line with this, this dissertation focuses on enhancing the precision and robustness of human posture detection and classification by proposing an overall model that structures video data, processes it, and leverages it to train CNN models for action recognition.

This dissertation aimed to develop a pipeline that is both efficient and scalable enough for action recognition tasks, capable of operating in various real-world environments, including sports analysis, human-computer interaction, and surveillance. The proposed PC model in this dissertation was used in building an application for HPD. Deep learning (DL) has been applied to various computer vision tasks, including healthcare applications for cancer and cancerous disease detection [206-208] and agricultural applications for plant disease detection [209]. Recent surveys have discussed the development of human detection techniques beyond human descriptors [1]. DL has also been applied in human detection due to its high performance in image-based applications. Since humans can be treated as objects, any DL-based object detection method can be applied to HD [207, 208]. For instance, pedestrian detection utilised Fast R-CNN, an existing neural network for object detection, to achieve effective detection [208]. Fall detection from video streams has also been researched [210-213]. In [214], both fall events and other abnormal events were detected using multi-modal information provided by the Motion History Image technique. In [215], only Grey Motion Map energy was used in the detection of the fall candidates, and the classification thereof was performed with an extended kernel descriptor and SVM. One employed a voting system, while another classified postures by posture recognition across consecutive frames using aspect ratio, area ratio, and orientation obtained from PCA of human silhouettes [216].

A hybrid model was presented in [217] for human posture detection in health monitoring systems, utilising the InceptionV3 model in conjunction with SVMs. The InceptionV3 model was used to extract image features, and classification was performed using an SVM. In [218], an HPD technique was proposed that uses hyperparameter-optimised image augmentation and transfer learning (TL) methods. The variability and robustness of the training dataset were enhanced using extensive image augmentation. On the other hand, the hyperparameters of the transfer-learning models were further optimised to improve performance. Popular architectures, such as VGG16 and ResNet50, were employed and fine-tuned, particularly to identify different postures. The performance of the optimised models validated that their fusion has a positive impact on human posture detection in various ways. All the problems associated with past human posture detection have been addressed by the proposed VEACTION-CNN model, which combines VGG16 with an attention mechanism to deliver strong, more accurate results across various postures in real-life scenarios.

Table 2 presents a structured summary of the literature (Sections 2.1–2.8), including the authors and year of each study, the methods used, the datasets involved, their main contributions, and any noted gaps or weaknesses.

Table 2. Summary of Related Works Reviewed

Authors	Methods	Dataset	Contributions	Gaps/Weaknesses
Cao et al., 2017	Real-time multi-person 2D pose estimation using a CNN with Part Affinity Fields (graphical model for body joints)	COCO & MPII (multi-person images)	Introduced OpenPose, the first real-time system for multi-person pose (skeletal joint) detection with high accuracy	Not posture classification (outputs joint coordinates, not posture labels); high computational cost due to multi-person complexity; assumes full person visible (performance degrades with heavy occlusion).
Mehr & Polat, 2019	Deep learning for activity/posture recognition in a smart-home setting; used a CNN-based approach for classifying human activities	Smart home video dataset (multiple activities in an indoor environment)	Applied CNN successfully to human activity recognition in IoT/smart-home context; demonstrated the feasibility of vision-based posture monitoring for ambient intelligence	Focused on a specific environment (single home setup), limited range of activities (may not generalise to subtle postures), did not specifically address class imbalance or small data (assumed enough training data).
Ben Gamra & Akhloufi, 2021	Survey/Review of deep learning techniques for 2D and 3D human pose estimation (no new model; comparative study)	N/A (review paper summarising many datasets and methods)	Provided a comprehensive overview of state-of-the-art DL methods for pose estimation (2D/3D), highlighting trends and common approaches; identified the dominance of CNN-based models in posture and pose tasks.	Does not propose a solution; it notes that hyperparameter tuning and data scarcity were open issues, e.g., many reviewed works did not employ HPO, leaving potential accuracy gains untapped.
Song et al., 2020	Benchmarking various deep learning models for human pose estimation and their application	Multiple public pose datasets (e.g., MPII, COCO)	Evaluated the performance of different CNN-based pose estimators and how pose outputs	Focused on pose (joint detection) rather than classifying static “postures”; did not address

	to action recognition		can feed into activity recognition; confirmed deep models significantly outperform earlier methods for pose-based action analysis	improvements for limited data, assumed large training sets; computationally intensive models.
Ogundokun et al., 2022	Transfer Learning (AlexNet, VGG16) for posture classification with extensive image augmentation and automated Hyperparameter Optimization (random search)	MPII Human Pose dataset (selected posture images)	First to integrate HPO into posture detection: tuned learning rates, layers, etc., plus augmented data, achieving >90% accuracy (AlexNet 91.2% with tuning vs lower without). Showed that tailoring hyperparameters markedly improves performance on small medical image datasets	Computational cost of HPO (requires many training trials); approach limited to static images (no temporal info); focuses on a few posture classes from MPII (may not directly extend to more complex activities).
Bootsman et al., 2019	Wearable sensor-based posture monitoring: smart shirt with dual inertial sensors for back posture tracking	Workplace posture dataset (inertial measurements of office workers)	Demonstrated a non-vision posture detection system for workplace ergonomics; provided real-time monitoring of back posture using an IoT wearable, with high accuracy in detecting improper sitting positions.	Requires wearing special hardware (compliance issue); limited to seated posture contexts (not general postures like standing or lying); does not leverage visual information (cannot detect broader environment or multiple people).
Ho et al., 2016	Depth-sensor-based posture classification using Kinect employed improved feature extraction and an SVM classifier to recognise postures from depth images.	Kinect depth video dataset (smart environment monitoring).	Improved posture classification accuracy using depth cameras (reported ~88.7% accuracy with an SVM on depth images); showed the viability of using 3D depth data for human	Relies on depth sensors (expensive and limited field of view); performance drops if the subject is outside the sensor range or at extreme angles; not easily scalable to wide areas due to the

			activity/posture monitoring in ambient intelligence.	cost of installing depth cameras everywhere.
Fragkiadakis et al., 2019	Physiological sensors + ML: Classified sitting postures using an array of sensors (pressure, ECG, respiration) and machine learning (KNN, SVM, Decision Trees).	Laboratory study (126 students; recorded breathing, ECG and posture data).	Developed a system to recognise two seating postures; achieved ~99.5% accuracy in distinguishing upright vs. slumped sitting using sensor fusion and simple ML classifiers. Validated that even basic ML can yield very high accuracy in controlled settings with rich sensor data.	Very limited posture scope (only two postures: e.g., normal vs. slouched sitting); scenario-specific (requires instrumented chair and biomedical sensors); not vision-based, impractical for general environments and does not classify full-body postures.
Liaquat et al., 2021	A hybrid model combining deep neural networks and machine learning for posture detection (integrated a CNN for feature extraction with an ML classifier).	Custom dataset; postures in an indoor setting (possibly from camera and/or wearable data).	Proposed a hybrid posture detection approach (Sensors Journal). By fusing a deep model with traditional classifiers, they improved generalisation and achieved strong accuracy (~95%) across multiple postures. Highlighted the benefit of mixing ML and DL techniques for this task.	Complexity: two-stage model (requires tuning both CNN and ML components); did not address real-time performance explicitly; dataset was still somewhat limited (room for testing on larger, more diverse data).
Tran et al., 2021	Vision-based posture classification from multiple camera viewpoints; also designed for fall detection. Used deep learning to integrate multi-view features.	Multiview video dataset (for posture and fall scenarios)	Enabled posture recognition and fall detection even when a person is seen from different angles, by combining information from multiple cameras. Showed that multi-view fusion improves the	Requires multiple cameras, higher setup complexity; not focused on posture subtleties beyond falls (binary fall vs. normal classification); approach may not work well if views are not

			detection of events like falls compared to single-camera setups.	synchronised or if one view fails.
Garg et al., 2022	Yoga pose classification using CNN and MediaPipe (pose landmarks); essentially a deep learning approach guided by a pre-existing pose estimation model.	Custom yoga pose image dataset (real-world yoga posture images).	Achieved good accuracy in classifying yoga postures by combining a CNN with Google's MediaPipe for extracting pose keypoints. Showed that leveraging pre-trained pose estimation (MediaPipe) can boost accuracy in posture classification tasks.	Domain-specific (yoga poses differ from everyday postures); may not generalise to non-yoga postures; depends on MediaPipe quality errors in keypoint detection could hurt classification; no mention of handling limited data (likely needed many training images).
Kulikajevs et al., 2021	Hierarchical image composition + CNN: Broke down images into parts to detect sitting posture, and combined the results in a hierarchical classifier.	Custom images of people sitting (for detecting correct vs. incorrect sitting postures).	Introduced a novel method of splitting images (e.g., upper vs. lower body) and using deep learning on those parts to determine if someone is sitting with proper posture. Achieved high accuracy in detecting problematic sitting positions and improved interpretability by focusing on specific body regions.	Focused solely on sitting posture and its variants (not a general posture detector); hierarchical approach may need redesign for other postures; the method, while interpretable, is somewhat tailored and may not scale to complex poses or whole-body activities.
Nagalakshmi & Mukherje, 2021	3D view learning for yoga pose classification: took a single 2D image and learned to infer 3D pose to better classify the yoga asana using CNN techniques.	Yoga posture image dataset (photos of people performing āsanās).	Innovatively used a 3D human model approach to augment single-image classification by "learning" the 3D pose from a 2D image; the system improved recognition of	Again, limited to yoga postures; requires a prior 3D human pose model and may be sensitive to estimation errors; not tested on general daily postures; method computationally

			yoga asanas that are ambiguous in 2D. Demonstrated higher accuracy in classifying similar poses by incorporating depth estimation.	heavier due to the 3D pose inference step.
Singh et al., 2024	Evaluated a pre-trained pose estimator (PoseNet) for posture detection; essentially a study of the transferability of PoseNet for posture classification	Experiment with PoseNet on a posture dataset (possibly a subset of images with labelled postures).	Provided an assessment of how well a generic pose-estimation model (PoseNet) can be repurposed to detect static postures. The findings indicated moderate success in using PoseNet outputs to classify postures, highlighting both the potential and the shortcomings (e.g., certain postures were accurately detected using PoseNet joints).	Efficacy was limited; the study found PoseNet alone was not fully reliable for posture classification, pointing out the need for fine-tuning or dedicated models (PoseNet was not trained for posture classes specifically). This underscores that generic pose estimation isn't a silver bullet for posture recognition, especially for subtle or transitional poses.

The above table draws on key studies discussed in Chapter 2 of the dissertation. It shows the evolution from traditional sensor-based or classical ML methods to modern deep learning approaches. Common themes include using CNNs and transfer learning to improve accuracy, as well as addressing limited data (which many studies address through pre-trained models, augmentation, or hybrid strategies). Gaps often revolve around data scarcity, lack of generalization (methods focusing on very specific postures or scenarios), and computational or practical constraints (e.g., requiring special devices or multiple cameras). This context motivated the dissertation's approach, which introduced hybrid models, HPO, and feature fusion to further enhance performance while addressing these gaps.

1.10. Summary

The introduction to this dissertation reviews prior research on human posture detection, focusing on deep learning, transfer learning, and hybrid models. Table 2 summarises the comparative review of the method, datasets, and performance metrics. It highlights the need for lightweight, data-efficient models to address the challenges posed by limited video data, which motivated the models proposed in this dissertation.

2. MATERIALS AND METHODS

This chapter introduces the methodology of the proposed posture detection system, including descriptions of datasets, data preprocessing, model architectures, training procedures, and evaluation metrics. The overall goal is to develop and optimise deep learning-based approaches to identify human postures from limited video content, with an emphasis on lightweight models for real-time edge deployment. Four human postures (sitting, standing, bending, and lying) are considered for the Silhouettes and ETRI-Activity3D-LivingLab Datasets, whereas the MPII dataset includes bicycling, dancing, running, and walking due to their relevance to health tracking and ergonomics. Figure 2 provides an overview of the system's workflow, from data collection to model training and inference. It illustrates the proposed model for detecting human posture. Video frames or images are first preprocessed (resized to the chosen input resolution and normalised), with data augmentation applied to increase the dataset size. The backbone of the system involves several deep learning models (transfer-learned CNNs and hybrid models) trained to categorise posture images into the given classes. Hyperparameter adjustment and regularisation (e.g. early stopping) are employed while training to promote generalisation. Inference: the trained model is deployed to classify new posture images in real time, with potential deployment on edge hardware (such as smart cameras, mobile devices, or wearables) for in-device posture evaluation.

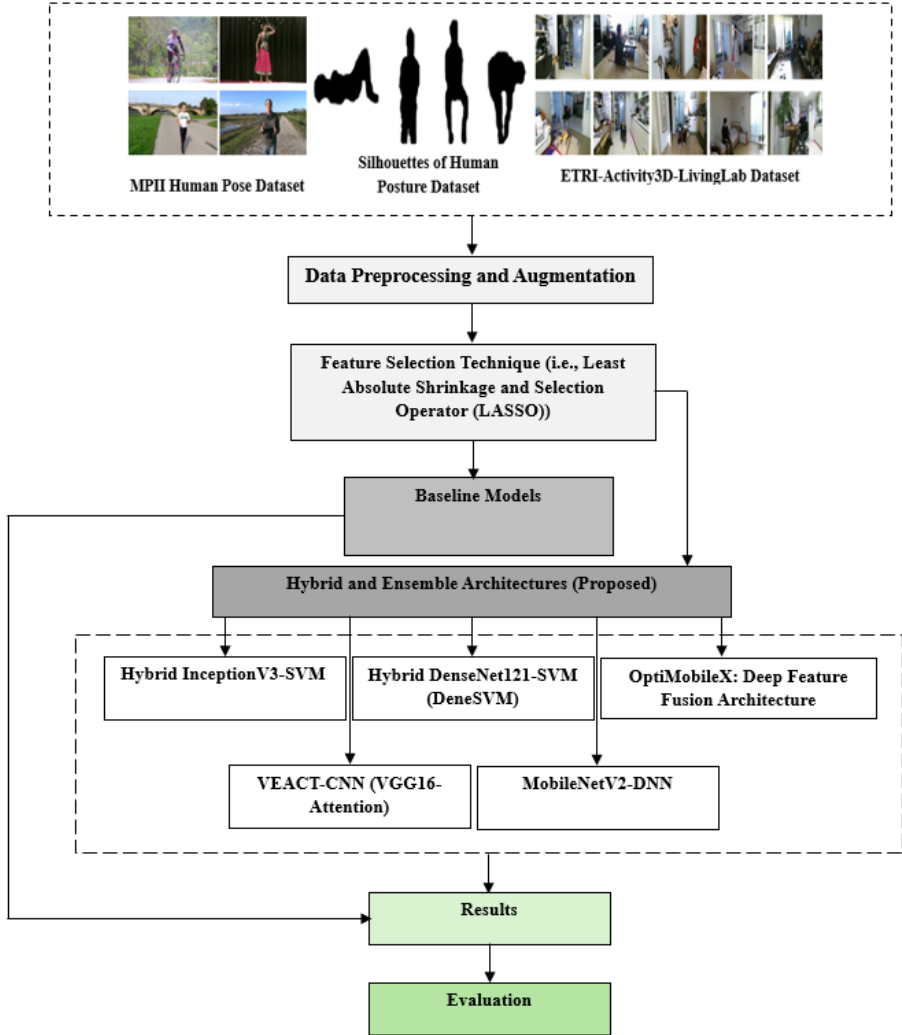


Fig. 2. The Proposed Human Posture Detection Model

Each methodological modules are presented in detail in the following subsections. This process began with dataset gathering and preprocessing, followed by the design of the baseline and the proposed dissertation models, the configuration of the training pipeline, and the evaluation of the measures.

2.1. Datasets and Preprocessing

Three datasets were utilised to simulate real-world posture detection under restricted video: (1) the MPII Human Pose dataset, (2) the Silhouettes of Human Posture dataset, and (3) the ETRI-Activity3D Living Lab dataset. These datasets include a mix of posed annotations under varied conditions. Video clips from these

datasets were partitioned into frames, and irrelevant or low-quality frames were removed. Each frame was normalised by resizing (to 224×224 or 299×299 pixels, to match the input dimensions of different CNN models) and normalising pixel intensities (either to $[0,1]$ or to the ImageNet dataset mean and standard deviation). This helped us ensure compatibility with pre-trained network inputs and reduced illumination variation.

Data augmentation was achieved using the Keras ImageDataGenerator, with the random rotations of ($\pm 15^\circ$), horizontal flip, zoom (scale range $0.8-1.2\times$), and shifts (max 10% width/height) and minor brightness adjustments. These augmentations were effective in increasing the diversity of training samples and minimising overfitting, which is especially significant given the limited original data. After preprocessing, the Silhouettes posture dataset contained 4800 images, evenly distributed across 4 classes (bending, lying, sitting, standing). This was split into training, validation, and test sets (approximately 80%, 10%, and 10%, respectively), for example, with 3840 training, 480 validation, and 480 testing images in total for Silhouettes data, 2227 training, 279 validation and 279 testing images in total for the MPII dataset. MPII and ETRI data were utilised in other experiments (e.g., cross-dataset evaluation). Still, model construction and early evaluation were done on the Silhouettes data to mimic the limited-data situation.

2.1.1. MPII Human Pose Dataset

The MPII Human Pose Dataset is a widely used benchmark dataset for estimating articulated human posture. It comprises approximately 25,000 RGB images and over 40,000 annotated humans across approximately 410 activities. These activity categories are obtained mainly from YouTube videos [177]. Detailed annotations, which include up to 16 body joints, are provided for each pose image, covering the shoulders, elbows, and ankles. This makes it a rich resource for learning posture-associated spatial relationships.

The dataset was filtered for this dissertation to include only images which were explicitly labelled as standing, sitting, bending, or lying. The images were uniformly resized to 224×224 pixels, which met the input requirements for transfer learning (TL) models. The dataset was preprocessed to adhere to ImageNet standards, including pixel normalisation using the mean and standard deviation to ensure consistency with the pre-trained model's requirements. To prevent overfitting, each image was augmented with techniques such as random rotations of $\pm 15^\circ$, flips, zooms of $0.8-1.2\times$, and brightness adjustments of $\pm 10\%$. The implemented pipeline enhanced model robustness while preserving semantic richness from natural imagery, thereby proving fundamental for fine-tuning deep architectures, such as InceptionV3, MobileNetV2, and Xception, in RGB-based posture detection. Figure 3 shows the sample MPII human pose dataset, which includes four human postures: bicycling, dancing, running and standing.



Fig. 3. Samples of MPII Human Pose Dataset

2.1.2. Silhouettes of Human Posture Dataset

This is the second dataset used in this dissertation, which consists of binary silhouette images mapped to four posture classes: sitting, standing, bending, and lying. Each posture class consists of approximately 1,200 mask samples. The original image resolution is 512 x 512 pixels, and they are later downsampled to 224 x 224 pixels to maintain consistency. Nearest-neighbour interpolation was used to preserve edge sharpness. The image pixel values were normalised to the 0- 1 range. The dissertation employed minimal augmentation, including horizontal flips and slight brightness adjustments, as its structural simplicity reduces the likelihood of overfitting. The dataset played an essential role in evaluating model robustness under privacy-preserving or data-limited incidences, emphasising shape over texture. It also provided a low-detail benchmark to assess whether the same models, fine-tuned on RGB data, could generalise from a minimalist representation of posture. The four classes of the Silhouettes Human Posture dataset, which include bending, lying, sitting, and standing, are shown in Figure 4.

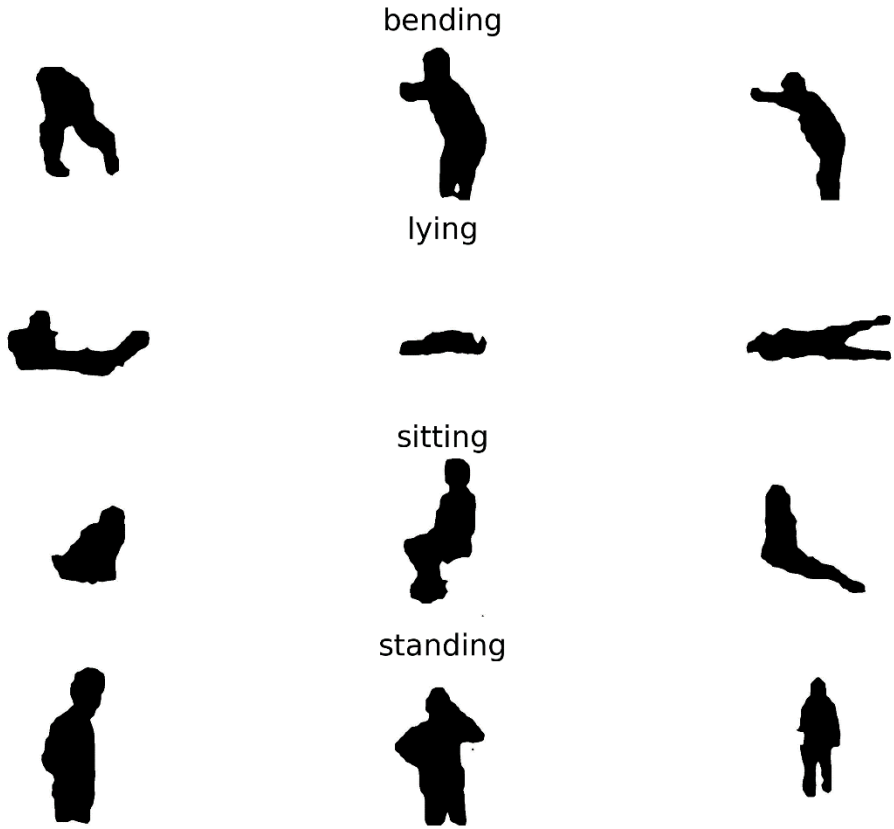


Fig. 4. Four Classes Sample of Silhouettes of Human Posture Dataset

2.1.3.ETRI-Activity3D-LivingLab Dataset

The ETRI-Activity3D dataset used in the experiments of this dissertation was developed for elderly care robotic applications. The dataset is designed for real-time posture analysis in an elderly care scenario. It comprises 112,620 RGB-depth-skeleton frames obtained from Kinect v2 sensors in real residential environments, collected from 100 individuals (50 elderly and 50 young adults) performing 55 daily activities [219, 220].

The dataset was preprocessed through the following phases. The first phase involved manual frame extraction, selecting frames from video sequences corresponding to the four target posture classes: bicycling, dancing, running, and walking. The extracted RGB frames were thereafter resized to 224 x 224 pixels. In this dissertation, only RGB data were used in the primary pipeline, although depth data remained available for supplementary multimodal experiments. Comprehensive augmentation, including random rotations, flips, zooms, brightness and contrast adjustments, and slight perspective warping, was employed due to the challenging

nature of real home environments, characterised by dynamic lighting, occlusions, and clutter. The dataset was finally split into training, validation, and test sets at 80:10:10, ensuring each set had a balanced distribution of the four posture classes. The all-encompassing augmentation techniques are designed to expose models to considerable variability, thereby encouraging generalizability. The proposed models that executed well on this dataset demonstrated a commitment to robust real-world posture detection under challenging conditions. In particular, the ETRI-LivingLab dataset helped bridge the gap between curated datasets and real-world applications, a common challenge in posture/activity identification.

In summary, by employing appropriate preprocessing, which involves resizing, normalisation, and comprehensive augmentation for each of the three datasets, this doctoral dissertation study ensures a unified and robust training model. The datasets were strategically selected to show clean RGB imagery, minimalist structural representations, and noisy real-world scenes, allowing an inclusive assessment of posture detection models across structured, semi-structured, and unstructured environments. The sample dataset is shown in Figure 5.



Fig. 5. Four Classes Sample of ETRI-Activity3D-LivingLab Dataset

2.2. Data Preprocessing and Augmentation

The dissertation examines several architectures for deep learning, both baseline models and proposed novel models, organized in three tiers: (a) Baseline models, e.g., a custom shallow CNN, AlexNet, and a multilayer perceptron (MLP), trained from scratch on the posture data; (b) Transfer Learning (TL) models, deeper CNNs (InceptionV3, Xception, DenseNet121, VGG-16, MobileNetV2) pre-trained on large corpora and fine-tuned for posture classification with extra custom classification layers; and (c) Advanced hybrid lightweight models, novel architectures investigated in this work that merge features or classifiers to counteract data limitations. Hybrid variants include feature-fusion networks and CNN-SVM hybrids, which aim to achieve maximal accuracy while balancing model complexity. All models were implemented in Python (TensorFlow/Keras) and executed on a Core i7 processor workstation with 16 GB RAM (leveraging GPU acceleration where available for training). The specific algorithms and model structures proposed throughout this dissertation are discussed in detail below, accompanied by pseudocode and mathematical details to provide rigour and clarity. The input-output behaviour of each algorithm is formalised to highlight how posture classification is obtained, while accounting for limited data requirements.

2.3. Feature Selection Technique

The dissertation employed the Least Absolute Shrinkage and Selection Operator (LASSO) feature selection to enhance model interpretability, avoid overfitting, and improve computational resource efficiency [221]. The technique is defined by the objective as seen in equation (1):

$$\min_w \left\{ \frac{1}{N} \sum_{i=1}^N (y_i - w^T x_i)^2 + \lambda \|w\|_1 \right\} \quad (1)$$

Where:

- x_i represents the feature vector extracted from a CNN layer for a sample
- i, y_i is the corresponding posture label (encoded numerically)
- w is the coefficient vector
- λ is the regularisation strength.

The ℓ_1 penalty ($\|w\|_1$) promotes sparsity by encouraging less important feature weights to shrink to exactly zero. This concludes with automatic feature pruning, retaining only the most applicable features for classification, thereby reducing dimensionality and minimising overfitting. The regularisation parameter was determined for the LASSO FS via a cross-validation grid search $\lambda \in \{0.001, 0.01, 1, 10\}$. The optimal λ was chosen as the value minimising the validation error. The model was employed to reduce the original deep feature vectors

(2,048 dimensions from DenseNet121/InceptionV3) to approximately 250-300 nonzero features, which were subsequently used as input to the SVM classifier.

2.3.1. Rationale for LASSO FS

- LASSO incorporate FS into the model training activities itself, and this is different from filter or wrapper techniques that decide features independently and are oftentimes less competent.
- By removing coefficients, LASSO produces a distributed model that highlights the most significant features, ensuring the model is more interpretable.
- It immediately trims down the number of participating features, reducing complexity and training period, which is essential when utilising CNN-extracted features from high-dimensional data.

2.3.2. Benefits of Posture Detection

LASSO (L1) feature selection was employed in this dissertation because of its suitability for posture recognition tasks.

First, it helps mitigate overfitting by eliminating irrelevant or redundant features, thereby reducing the likelihood that the model learns noise from the training data.

Second, LASSO enhances interpretability by shrinking less informative feature weights to zero. This makes it easier to identify which CNN-derived feature activations meaningfully contribute to posture detection decisions.

Finally, by constraining the model to focus only on the most informative features, LASSO can improve overall model performance. When combined with classifiers such as Support Vector Machines (SVMs) or Multilayer Perceptrons (MLPs), it often improves classification accuracy and increases model stability.

In summary, LASSO is an effective tool in this dissertation's methodology that refines the feature set, modifies the model structure, and enables efficient, interpretable posture classification, thereby aligning with best practices in high-dimensional feature learning.

2.4. Model Architecture

This dissertation employed a broad spectrum of baseline, hybrid, and ensemble DL models to evaluate their suitability for human posture detection, particularly under limited video frames. These models employed varying architectural depths, parameter counts, computational efficiencies, and learning strategies. This also encompasses elementary MLP to advanced TL ensemble models optimised for real-time applications. These models were developed and evaluated using TensorFlow and the Keras library. They were individually tailored to explore different aspects of posture detection from limited video material. This section discusses the baseline models and their architectures, supported by visual diagrams.

2.4.1. AlexNet Model

AlexNet is one of the earliest successful deep convolutional neural networks (CNNs) that revolutionised computer vision when it won the ImageNet ILSVRC in the year 2012. Its model comprises five convolutional structures followed by three fully connected (FC) layers, totalling eight trainable layers and approximately 60 million parameters. The model accepts input images of size 227 x 227 pixels and uses ReLU activations and max pooling for feature extraction. The model is comparatively shallow by recent standards, but it serves as a benchmark for classical CNN performance in visual recognition tasks.

2.4.2. Custom CNN Model

The custom CNN model used in this dissertation is lightweight, comprising three convolutional blocks, each with a convolutional layer, batch normalisation, ReLU activation, and max pooling. The extracted features are flattened and passed through to the FC layers with dropout regularization introduced to decrease overfitting. This method was trained from scratch and tailored for the posture dataset employed.

2.4.3. Deep Neural Network (DNN)

In this dissertation, a DNN was implemented as a multilayer perceptron (MLP) comprising two dense layers with 512 and 256 neurons, respectively, and dropout was introduced for regularisation. It receives flattened image vectors as input. This renders a non-convolutional baseline.

2.4.4. Support Vector Machine (SVM)

SVM is a well-known machine learning (ML) model that uses kernel functions, such as the linear or RBF (Radial Basis Function) kernels, to define classification boundaries. In the hybrid model proposed in this dissertation, the SVM was employed as the final classifier on deep features extracted from CNNs or pretrained networks. In this dissertation, the following were used as stated:

A linear kernel was employed with the regularization parameter C tuned via grid search $C \in \{0.1, 1, 10\}$ or the proposed DeneSVM model, while for the InceptionV3 + SVM models used during the ablation study, the RBF kernel was used with C and γ optimized it through grid search ($C \in \{0.1, 1, 10\}, \gamma \in \{0.001, 0.01, 0.1\}$)

2.5. Transfer Learning (TL) Models

TL models originated as a crucial approach for several image-based tasks, especially when large-scale labelled datasets are insufficient. This process begins with a CNN pretrained on a large dataset, such as ImageNet, which adapts its learned

representations to a new task: human posture detection. The approach ‘transfers’ learned feature representations, such as edges, textures, and shapes. This considerably reduces the model training time, improves generalisation, and eliminates the need for large datasets. Empirical studies have demonstrated the effectiveness of modern methodologies, such as InceptionV3, Xception, DenseNet121, ResNet50V2, and MobileNetV2, in projects spanning medical imaging, activity recognition, activity identification, and posture detection. In this dissertation, the pretrained models were fine-tuned for posture classification, leveraging their hierarchical feature extraction capacities. These include layers 1-10, which capture textures, and deeper layers, which capture semantic posture cues, among others.

2.5.1.VGG 16

VGG 16 is a well-established architecture with 13 convolutional and 3 FC layers, which was pretrained on ImageNet. In this dissertation, the convolutional base was fixed, and the top-classification layers were trained on the posture dataset. VGG16 achieves robust performance in posture recognition due to its hierarchical feature extraction and straightforward design.

2.5.2.InceptionV3

This model is more advanced, as it utilises multiple kernel sizes in parallel via inception modules to capture multi-scale features. The last layers were fine-tuned for posture classification. The InceptionV3 network strikes a balance between depth and computational efficiency, performing well under varying experimental conditions.

2.5.3.Xception

This model is a variant of Inception that replaces standard convolutions with depthwise separable convolutions, significantly reducing computational cost while maintaining accuracy. It was based on the approach employed in this dissertation, with the last block held back and linked to a custom classifier. Its architectural effectiveness makes it appropriate for posture detection on limited computational hardware.

2.5.4.DenseNet121

The Densenet121 model was also employed in this dissertation. It features dense connections, in which each layer incorporates input from all preceding layers. This DenseNet121 model advances gradient flow and feature reuse through dense connectivity between layers. The model is computationally effective while retaining high accuracy.

2.5.5. MobileNetV2

This is a lightweight CNN model that is optimised for mobile and embedded applications. It employs depthwise separable convolutions and inverted residual blocks with a linear bottleneck, making it ideal for deployment on edge devices for real-time posture monitoring. The dissertation leveraged its effectiveness by building on top of the pre-trained base, adding classifier layers.

Each model undergoes the same preprocessing, augmentation, and training strategies, including the Adam optimiser, early stopping, and dropout, enabling their direct comparison. Sequent hybrid and fusion networks, such as InceptionV3-SVM, DeneSVM, OptiMobileX, and VEACTION (VGG16-Attention), were built on these TL foundations. The models were subsequently augmented with methods such as LASSO FS, SVM classifier, and attention mechanisms.

2.5.6. ResNet50V2

The ResNet50V2 model is an improved variant of the original ResNet50, incorporating identity mappings and batch normalisation before activation to stabilise training and improve convergence. In this dissertation, ResNet50V2 was fine-tuned for posture detection and served as a benchmark for comparison against lighter models such as MobileNetV2 and the proposed OptiMobileX. Although the model achieved competitive accuracy, its larger parameter size of 25.6M and longer training time made it less suitable for real-time or resource-constrained environments than the proposed lightweight model.

2.6. Hybrid and Ensemble Architectures

In this section, the dissertation examines how to enhance model architectures to improve posture detection performance by combining the strengths of deep representations, classical classifiers, feature fusion techniques, and attention mechanisms. These architectures are tailored to address the challenges of limited datasets, computational efficiency, competency, and context-aware inference.

2.6.1. Hybrid LASSO+InceptionV3+SVM

The dissertation proposed a novel hybrid Inceptionv3-SVM model. This hybrid model extracts features from the penultimate layer of the pretrained InceptionV3 method and conveys them to an SVM classifier. The optimisation problem for SVM is presented in equation (2):

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \max(0, 1 - y_i (w^T x_i + b)) \quad (2)$$

That is, equation (2) can be broken down into:

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i,$$

$$\text{subject to } y_i(w \cdot x_i + b) \geq 1 - \xi_i, \xi_i \geq 0.$$

Where x_i is the feature vector, y_i the class label, C the regularisation parameter, and w, b define the decision boundary. This combination leverages InceptionV3's ability to learn rich, multi-scale features and SVM's robust generalisation from limited samples. The use of LASSO for FS achieved higher accuracy and AUC on silhouette benchmarks, surpassing InceptionV3 with a softmax layer.

The proposed model in this dissertation is shown in Algorithm 1:

Algorithm 1: Hybrid LASSO+InceptionV3+SVM

Function: InceptionV3_SVM(X_{train} , y_{train} , X_{val} , y_{val})

Input: Training images X_{train} , labels y_{train} ; validation images X_{val} , labels y_{val}

Output: Trained SVM classifier $h(z)$; validation accuracy

1. Load inceptionV3 pre-trained on ImageNet, remove final layer
2. Compute features:

$$Z_{train} = f_{Inc}(X_{train}), Z_{val} = f_{Inc}(X_{val})$$

3. If feature selection is enabled, then

Apply LASSO:

$$\min_{\beta} \frac{1}{N} \|Z_{train} \beta - y_{train}\|^2 + \lambda \|\beta\|_1$$

Select features with non-zero weights and reduce Z

4. Train SVM Classifier

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \max(0, 1 - y_i(w^T z_i + b))$$

5. Predict labels:

$$\hat{y}_{val} = h(Z_{val})$$

6. Compute accuracy:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

7. **return** $h(z)$, Accuracy
-

2.6.2. Hybrid DenseNet121-SVM (DeneSVM)

The DeneSVM model is another novel model proposed in this dissertation. The similar fusion method employs DenseNet121's densely connected features, which are integrated with an SVM classifier. The dense connectivity of DenseNet advances gradient flow and encourages feature utilisation, while SVM maximises the margin for posture classification. This hybrid model achieved higher accuracy on posture datasets, highlighting the synergy between dense deep features and strong classical classification. The sparse FS by LASSO further improves generalisation and interpretability. The proposed DeneSVM model, as presented in this dissertation, is illustrated in Algorithm 2.

Algorithm 2: Hybrid DenseNet121+SVM (DeneSVM)

Function: DeneSVM(X_{train} , y_{train} , X_{val} , y_{val})

Input: DenseNet feature dataset and labels

Output: Trained SVM Classifier

1. Load DenseNet121 pre-trained model
2. Extract features:
$$Z_{train} = f_{Dense}(X_{train}), \quad Z_{val} = f_{Dense}(X_{val})$$
3. If LASSO is enabled, then
Reduce feature vectors using selected coefficients
4. Train SVM using the above optimisation formulation
5. Predict and evaluate

return Trained model and metrics

2.6.3. OptiMobileX: Deep Feature Fusion Architecture

This dissertation proposes an alternative model, OptiMobileX, an ensemble architecture that combines features from the pre-trained MobileNetV2 and Xception. The mathematical equation is shown in equation 3:

$$\mathbf{F}_{fused} = [\mathbf{F}_{MNV2}^{(l_1)}; \mathbf{F}_{Xcep}^{(l_2)}] \quad (3)$$

A dense classifier learns from this fused feature vector. The OptiMobileX model captures complementary posture information by integrating multiscale representations from two effective networks while retaining a compact footprint (~12M parameters). The model achieved an advanced accuracy on silhouette datasets and robust generalisation on the MPII dataset. In this dissertation, the proposed OptiMobileX model is illustrated in Algorithm 3.

Algorithm 3: OptiMobileX

Function: OptiMobileX(X, y)

Input: Posture dataset X, label y

Output: Trained fusion-based classifier

1. Extract MobileNetV2 features: $f_M = f_{Mobile}(X)$
 2. Extract Xception features: $f_X = f_{Xcept}(X)$
 3. Concatenate: $f_{concat} = [f_M, f_X]$
 4. Optionally apply LASSO or PCA on f_{concat}
 5. Train an SVM or other classifier on fused features
 6. return Trained model
-

2.6.4.VEACT-CNN (VGG16-Attention)

This architecture fuses a convolutional attention mechanism into VGG16's convolutional feature maps. Given a feature map $F \in \mathbb{R}^{H \times W \times C}$, attention weights $\alpha_{i,j}$ are calculated as:

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{u,v} \exp(e_{u,v})}, \quad e_{i,j} = ReLU(W_\alpha F_{ij} + b_\alpha) \quad (4)$$

The weighted feature map emphasises important regions, such as limb boundaries, thereby enhancing the architecture's ability to recognise subtle posture variations. Implementations on ETRI-LivingLab demonstrated that VEACT-CNN outperforms baseline VGG16 models in both accuracy and interpretability.

Algorithm 4: VEACT-CNN (VGG16 Attention)

Input: Posture image X

Output: Predicted posture label $\hat{y}$

1. Extract base features F using pretrained VGG16
 2. Compute spatial attention mask A via learned 1×1 convolutions
 3. Apply attention: $F' = A \odot F$
 4. Flatten and classify using dense + softmax layers
-

2.6.5. MobileNetV2

This proposed model is another ensemble model employed in this dissertation. This compact hybrid combines MobileNetV2's pre-trained model with a custom dense architecture (512 and 256 units) for softmax-based posture classification. The model accomplished an improved accuracy with only ~5M parameters while ensuring real-time inference (~40 fps). This makes the model appropriate for deployment on edge devices where latency and resource usage are significant. The proposed MobileNetV2 model, as presented in this dissertation, is illustrated in Algorithm 5.

Algorithm 5: MobileNetV2-DNN

Function: MobileNetV2_DNN(X, y)

Input: Posture dataset X , label y

Output: Trained deep classifier

1. Extract features using MobileNetV2 (Frozen weights): $f = f_{Mobile}(X)$
 2. Append fully connected layers:
 $Dense(512) \rightarrow ReLU \rightarrow Dropout \rightarrow Dense(256) \rightarrow ReLU \rightarrow Softmax$
 3. Train using Adam optimiser and cross-entropy loss
 4. Evaluate on the test set
 5. return Trained MobileNetV2 model
-

2.7. Comparative Insights

To conclude the methodology chapter, the dissertation highlights key insights gained from comparing these models and algorithms:

1. **Deep CNN + CNN + AM Hybrids (VEACT-CNN):** Using deep CNNs (VGG16) purely as feature extractors and employing a baseline CNN with AM for classification tends to improve generalisation. In the experiments conducted in the dissertation, VEACT-CNN outperformed the corresponding CNN models with their native softmax classifiers in both test accuracy and the robustness of the decision boundaries.

2. **Feature Fusion Models:** The OptiMobileX model demonstrates that combining features from a lightweight network and a robust network yields a complementary representation. This fusion achieved the highest accuracy among single-image classifiers, indicating that no single architecture captured all posture nuances; instead, each contributed unique features (e.g., MobileNet might focus on the overall body outline, while Xception captures finer limb orientations). Feature

fusion is thus a promising direction for limited-data scenarios, as it effectively increases the expressive power without requiring additional training data.

3. **Lightweight Models for Edge Deployment:** The MobileNetV2-DNN, OptiMobileX and LitePose-GRU models' success illustrates that edge-friendly models can still be accurate for posture detection. This is critical for practical applications, such as wearable posture coaches or smart camera systems, where deploying a model with 50 million parameters rather than 20 million parameters is infeasible. By leveraging transfer learning and judicious network design, the dissertation achieved a model that runs in real time on modest hardware, confirming that the dissertation's approaches meet the goal of real-world applicability.

In summary, the methodological innovations presented (hybrid CNN-SVMs, deep feature fusion, attention-enhanced CNNs, and efficient DNNs) represent a holistic approach to posture detection under constraints. Each approach addresses specific challenges, whether it is limited data, the need for interpretability, or hardware limitations, and together they form a toolkit of solutions that will be evaluated in the next chapter.

2.8. Summary

In this dissertation, Chapter 3 presents a comprehensive overview of the materials and methods employed in the implementation. The dissertation began by presenting three distinct datasets: MPII Human Pose, Silhouettes of Human Posture, and ETRI-Activity3D LivingLab. The section also discussed how video clips were transformed into frames and filtered for quality. Each image was resized to 224x224 or 229x229 pixels, depending on the model's input image size. Normalisation was also performed, either scaled to 0-1 or standardised using ImageNet statistics. This was carried out to prepare the model input for training.

Keras's ImageDataGenerator class was employed for data augmentation, utilising online transformations such as random rotations ($\pm 15^\circ$), horizontal flips, zoom (0.8-1.2x), width/height shifts (up to 10%), and brightness adjustments (0.9-1.1x). This technique is a well-established strategy to artificially increase the dataset's variety and avoid overfitting by exposing networks to a broad range of visual variations during training. Feature dimensionality was later reduced using LASSO, which utilises ℓ_1 regularisation to promote sparsity. By effectual feature pruning while training models, uninformative activations are eliminated, decreasing model complexity and advancing interpretability, an approach identified in both ML and statistical communities.

The proposed models are organised into three tiers: baseline, TL and advanced hybrid networks. Baseline models include AlexNet, a custom CNN, and an MLP, which were trained from the top. In contrast, TL techniques, such as VGG16, InceptionV3, Xception, DenseNet121, and MobileNetV2, employed pretrained feature extractors with custom classification heads. The dissertation also included hybrid approaches that combined feature fusion with advanced pooling techniques. These include OptiMobileX, which combines MobileNetV2 and Xception features,

and hybrid CNN-SVM variants that use LASSO-selected features and incorporate a spatial attention mechanism. The dissertation also included a proposed model of a lightweight MobileNetV2 with a TL network to support real-time deployment on edge devices.

A consistent hyperparameter was employed for the model building. This includes Adam Optimiser, categorical cross-entropy, early stopping with a patience of 5, ReduceLROnPlateau, and a batch size of 32 for up to 50 epochs. However, some hyperparameters were optimised using random and grid search. These include learning rate, dropout rate and SVM parameters.

In conclusion, the dissertation employed a stratified five-fold cross-validation to ensure balanced class distributions across splits, and, in a few cases, an 80:10:10 data split ratio was used. An all-encompassing model assessment was then conducted. The performance metrics employed in this dissertation include accuracy, precision, recall, F1-score, and macro-average ROC-AUC, providing a comprehensive, reproducible evaluation of posture detection suitable for deployment in real-world healthcare and assistive environments.

3. EXPERIMENTAL RESULTS

This chapter evaluates the performance of the proposed models on the posture detection tasks and provides a detailed discussion of the results. The experimental setup is described first, and the results for each model (baseline and proposed) are then presented, including accuracy, loss curves, and detailed classification metrics. Statistical analyses were also conducted to assess the reliability of the results and the significance of performance differences between models. This section integrates the clinical and practical implications of the experimental findings, highlighting how proposed models can enhance ergonomics, rehabilitation and surveillance.

3.1. Experimental Setup

The dissertation experiments were conducted on a Windows 10 (64-bit) workstation with an Intel Core i9 Processor, 16 GB of RAM, and an NVIDIA GeForce RTX 4070 GPU (8 GB VRAM). Training was performed on both the CPU and the GPU, using TensorFlow 2.10 and PyTorch, respectively. Consequently, inference and ablation tests were performed on both CPU and GPU for comparison. The programming language used is Python version 3, and TensorFlow, Keras, and PyTorch were the deep learning libraries used in the dissertation. Other supporting libraries used in this dissertation are NumPy, Pandas, Matplotlib, and Scikit-learn. All models were trained on the Silhouettes posture dataset (after augmentation) as described in Chapter 3. Training was performed for up to 50 epochs for the CNN-based models, with early stopping applied if validation performance stopped improving (patience of 10–30 epochs, depending on the experiment). The batch size was kept at 32 images. The dissertation used the Adam optimiser with an initial learning rate of 0.001, which was reduced by a factor of 0.7 whenever the validation loss plateaued for 10 epochs.

These training hyperparameters were harmonised across models for a fair comparison. An early stopping criterion was applied to all experiments to prevent overfitting, meaning training often halted before 50 epochs once the model converged. For the SVM-based models (Inception+SVM, DeneSVM), the CNN feature extractors were trained (or fine-tuned) in a similar manner, and then the SVM was trained on the extracted features. All evaluation metrics were computed on the test set, which was set aside and not seen during training or validation. The test set comprised 480 images (120 images per posture class) from the Silhouettes dataset, which were completely unseen by the models during training, to evaluate generalisation performance. Some models were also tested on an unfamiliar dataset (ETRI-Activity3D) to assess cross-dataset performance, which is discussed qualitatively.

The dissertation employed stratified validation, splitting the data into 80:10:10 sets for model optimisation. The results are reported as mean $\pm$ SD, with p-values from paired t-tests across the training. The second evaluation strategy employed for the final evaluation was a test set comprising 10% of the images, totalling 480 images for the silhouettes dataset, 279 for the MPII dataset, and 1034 for the ETRI-

Activity3D-LivingLab Dataset. These images were not seen during model training or validation. Image augmentation was applied only to the training and validation sets, ensuring there was no data leakage from test samples into the training process.

3.2. Experimental Results on Sihouottes Dataset

In this dissertation, various DTL models were implemented on the Sihouette dataset and compared with standard CNN architectures, such as InceptionV3, MobileNetV2, DenseNet121, ResNet50V2, and Xception, with the proposed hybrid models, including OptiMobileX, which combines MobileNetV2 and Xception features, DeneSVM, which combines DenseNet121+SVM, LASSO+InceptionV3+SVM, and MobileNetV2+DNN model. The goal is to evaluate their training powers and final classification performance. The findings from the conducted experiments are discussed. This includes analysis of the training and validation curves (Fig. 6, Table 3), confusion matrices (Fig. 7–Fig. 9), class-wise metrics (Table 4), and precision–recall curves (Fig. 7–Fig. 9). Instances where the strengths or weaknesses of specific models became evident are highlighted; for example, the Xception model consistently demonstrated high accuracy compared to the slight performance deficits observed with MobileNetV2 and the underperformance of DeneSVM.

3.2.1. Training and Validation Performance

The performance of the baseline deep learning models, optimised models, and proposed hybrid lightweight models. In the dissertation, evaluation metrics such as training, validation, and testing accuracy, as well as the corresponding model losses, were used where applicable (Figures 6–9).

Among the pre-trained CNN-based models, Xception stood out with a testing accuracy of 93.13% and the lowest testing loss of 0.0872. This illustrated a stable convergence and reduced error propagation during the evaluation. MobileNetV2 and InceptionV3 were the second-best models, achieving test accuracies above 92.5%. Even though MobileNetV2 is a lightweight model, it has slightly higher training loss, suggesting it trades efficiency for generalisation performance.

The proposed OptiMobileX model achieved the best test accuracy of 95.42% across all baseline models and balanced loss values across all phases, with 0.1459, 0.1443, and 9.1354 for training, validation, and testing, respectively. This improvement of 2.29% over Xception and 2.84% over InceptionV3 is statistically significant, given the relatively small variance in performance among competitive CNN models. The low variance across training, validation, and testing phrases also reinforces OptiMobileX’s robustness against overfitting.

DenseNet121 did not perform well compared to other models, with a validation accuracy of 90.28% and a testing accuracy of 89.47%, alongside higher loss values exceeding 0.8. This suggests that, despite its depth, DenseNet121 may not be well-suited to the dataset, possibly due to vanishing gradients or overparameterisation.

The hybrid SVM-based models, such as DeneSVM and LASSO+InceptionV3+SVM, had competitive testing accuracies of 94.17% and 94.66%, respectively. These results highlight the effectiveness of combining feature extraction with machine learning classifiers. However, these models did not have training, validation, or test loss values. This is because the SVM classifier does not undergo end-to-end gradient-based optimisation; instead, it operates on pre-extracted deep features. Thus, loss curves and epoch-based training metrics cannot be computed, and evaluation is limited to final classification accuracy on validation and testing datasets. The proposed hybrid MobileNetV2-DNN achieved an accuracy of 96.40%, outperforming the baseline MobileNetV2, which achieved 92.55%.

In conclusion, the findings demonstrate that although advanced CNNs such as Xception and InceptionV3 offer strong baselines, the optimised lightweight OptiMobileX achieved the best trade-off among accuracy, robustness, and computational efficiency. Although SVM-based hybrids lack loss metrics, they provide additional evidence that hybridising deep features with classifier models can yield significant benefits.

Table 3. Accuracy and Loss of Training, Validation and Testing on Silhouettes Data

Model	Training Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)	Training Loss	Validation Loss	Testing Loss
Inception V3	72.28	89.12	92.58	0.2603	0.2650	0.1567
MobileNetV2	66.34	91.44	92.55	0.2856	0.2260	0.1612
Xception	89.60	92.13	93.13	0.0883	0.1268	0.0872
DeneSVM	-----	93.29	94.17	-----	-----	-----
OptiMobileX	94.92	95.42	95.42	0.1359	0.1423	0.1354
ResNet50 V2	87.67	91.90	91.94	0.1287	0.1953	0.1236
DenseNet 121	87.27	90.28	89.47	0.8162	0.7999	0.8039
LASSO+InceptionV3+SVM	-----	92.59	94.66	-----	-----	-----
MobileNetV2-DNN	94.16	95.25	95.40	0.1480	0.1445	0.1538



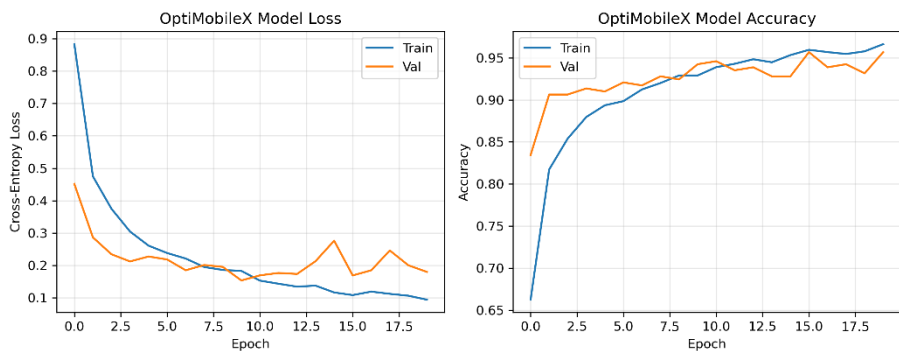
(i)



(ii)



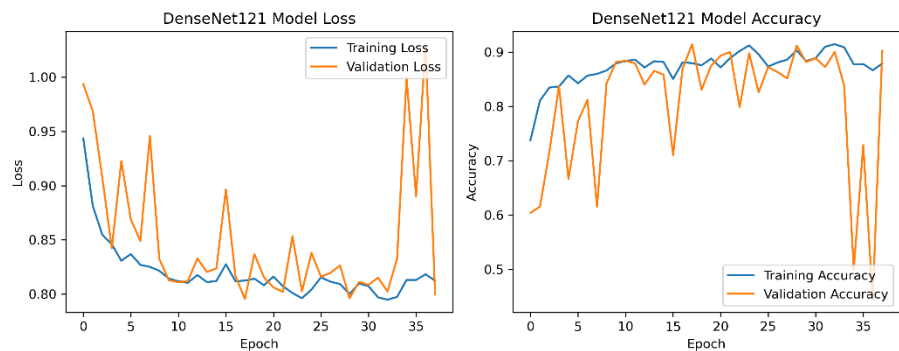
(iii)



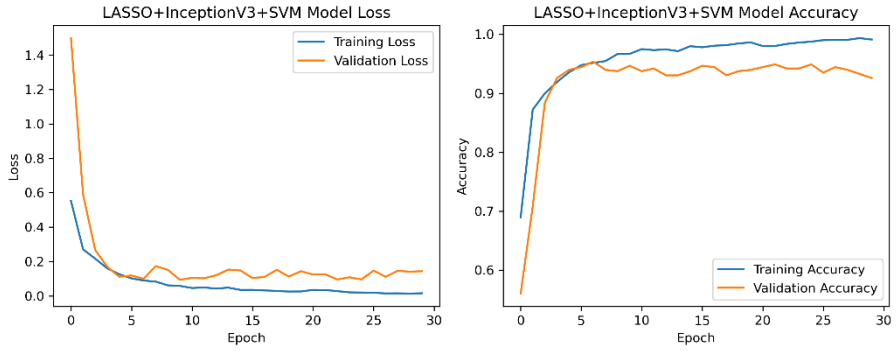
(iv)



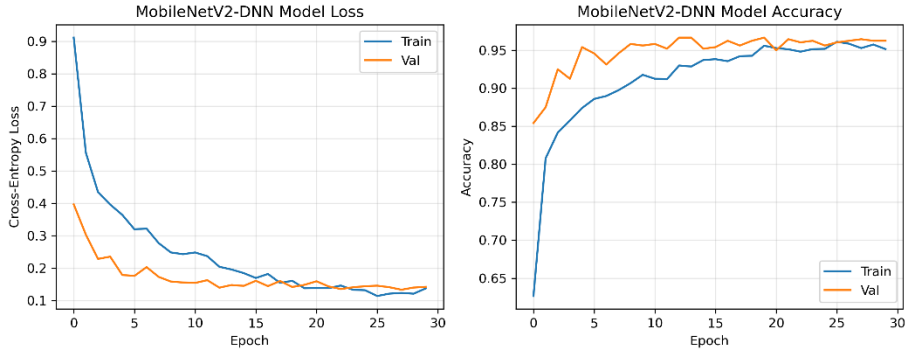
(v)



(vii)



(viii)



(ix)

Fig. 6. Accuracy and loss graphs of DTL models Sihuottes Data: (i) InceptionV3; (ii) MobileNetV2; (iii) Xception, (iv) OptiMobileX, (v) ResNet50V2, (vi) DenseNet121, (vii) DeneSVM, (viii) LASSO+InceptionV3+SVM, and MobileNetV2-DNN

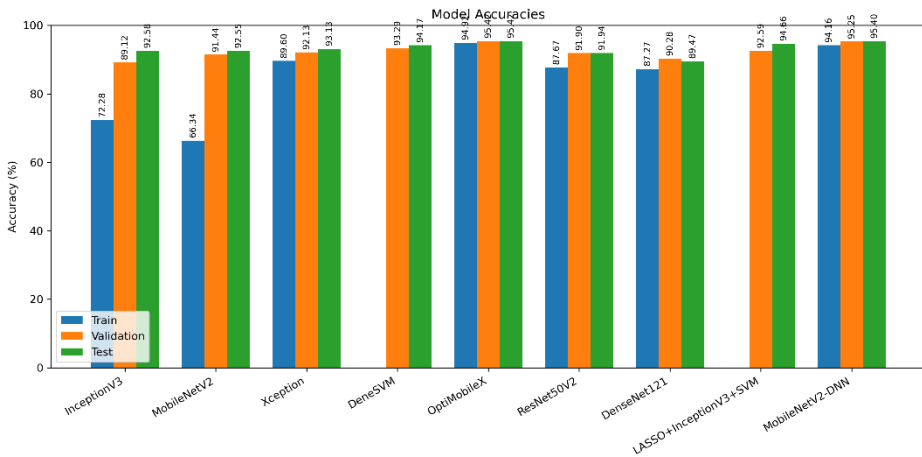


Fig. 7. The Nine Implemented Models' Accuracy Comparison

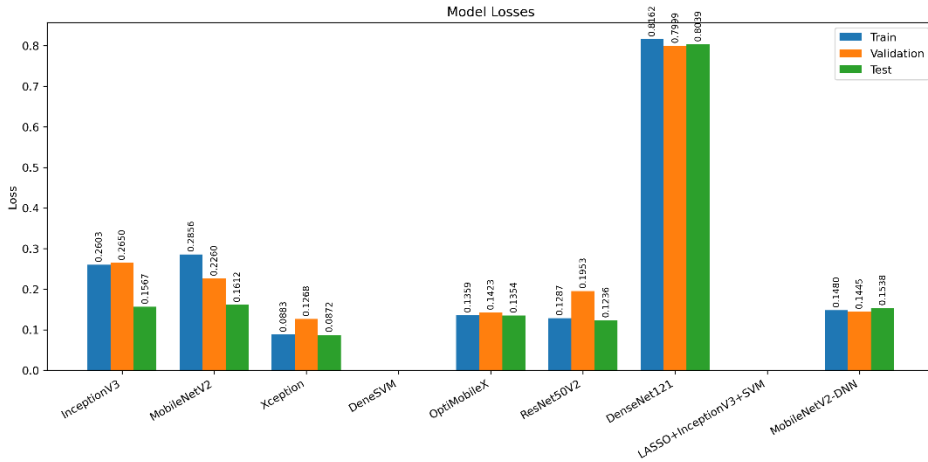
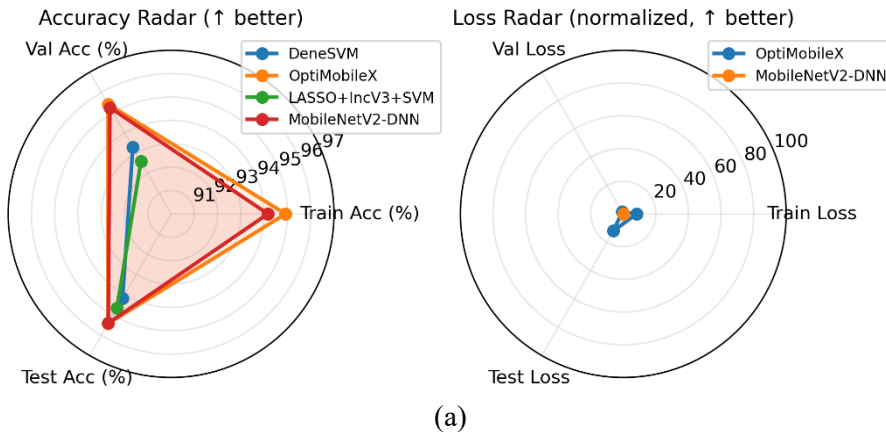


Fig. 8. The Nine Implemented Models' Loss Comparison

Table 3 presents the model evaluation, and Fig. 7 shows the training, validation, and test accuracies of the proposed models, while Fig. 8 shows their training, validation, and test losses. The figure shows that the proposed OptiMobileX outperformed all the models implemented in this dissertation, achieving an accuracy of 95.42% across both the validation and testing phases, highlighting its robustness and optimisation advantage. The figure clearly confirms that the OptiMobileX model not only leads in accuracy but also maintains stable results across different phrases. Hybrid models such as DeneSVM and LASSO+Inception+SVM are marked as N/A in the training column since SVM-based classifiers do not report training accuracy or loss in the same way as backpropagation-driven CNNs.



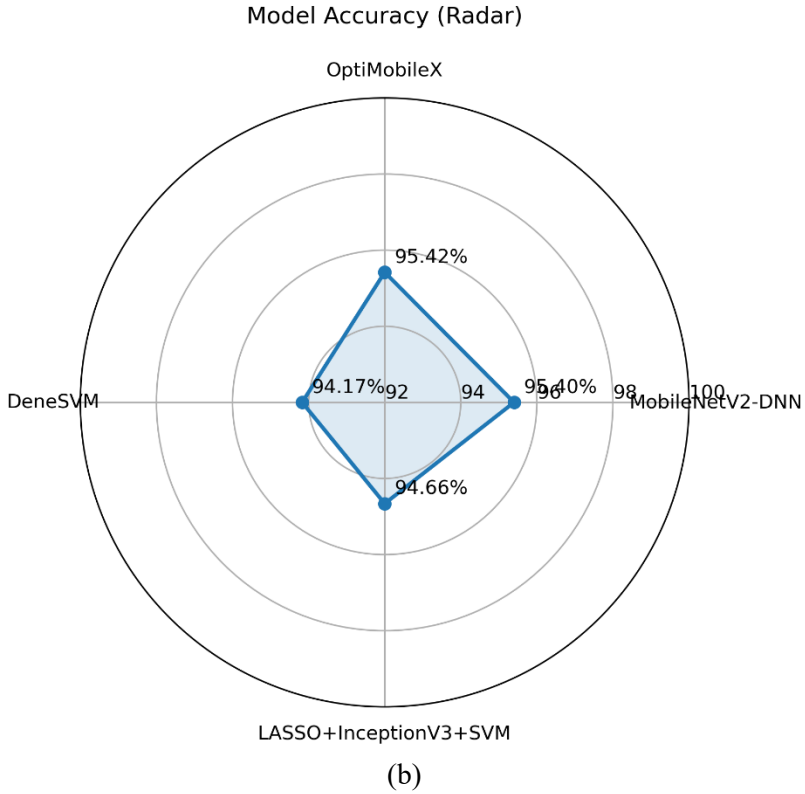


Fig. 9. Radar Chart of Proposed Hybrid Model Accuracies

The radar charts shown in Figure 9 illustrate the performance of the proposed hybrid models: DeneSVM, OptiMOBILEX, LASSO+InceptionV3+SVM, and MobileNetV2-DNN. Although the SVM-based hybrids achieve strong validation and test accuracies of 94.17% and 94.66%, respectively, they inherently lack training accuracy because SVM classifiers operate on extracted features rather than training through epochs. Whereas OptiMobileX demonstrates balanced and consistently high performance across all three pipeline stages, reinforcing its superiority in both accuracy and generalisation. The chart also presented that while the hybrid models close the gap with OptiMobileX in the validation and testing dataset pipeline, they cannot fully capture training dynamics due to their different optimisation

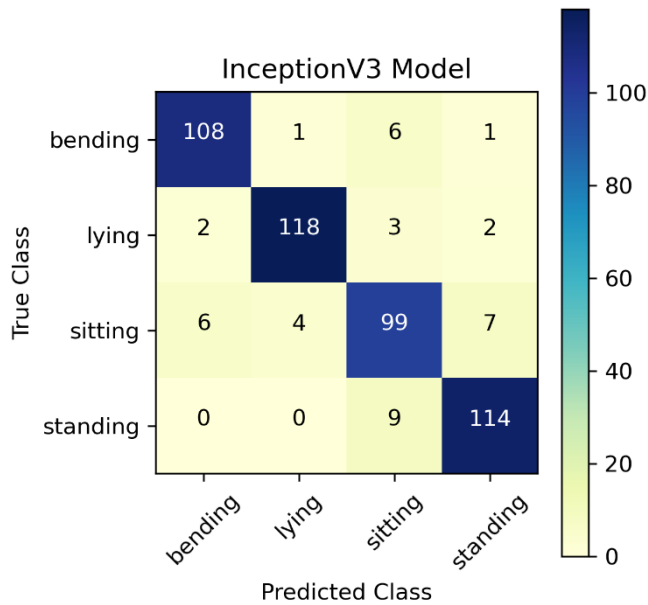
3.2.2. Confusion Matrix Analysis

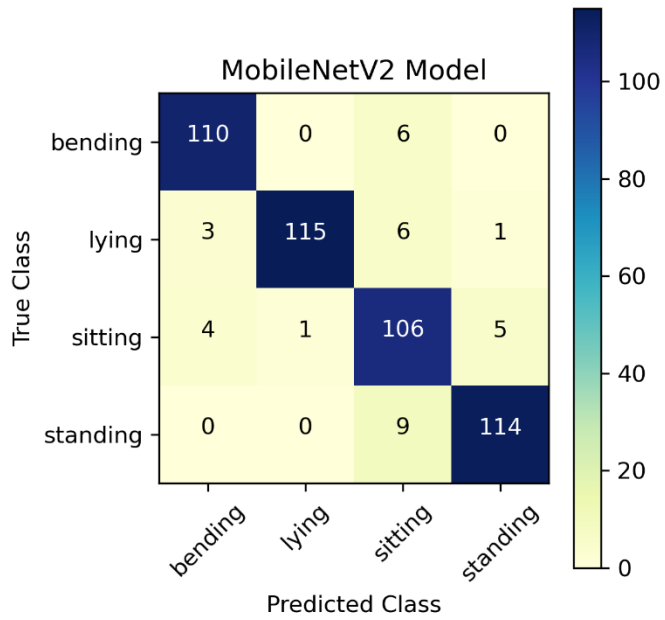
The confusion matrix, as illustrated in Figure 10 (i-viii), provides in-depth insights into the classification performance of the evaluated models across the four posture classes: bending, lying, sitting, and standing. The majority of samples were correctly classified across all classes for the InceptionV3 models, with only minor misclassifications between the sitting and bending classes. Likewise, MobileNetV2

demonstrated strong performance, especially in the standing and bending classes, but occasional confusion persisted between the sitting and lying postures.

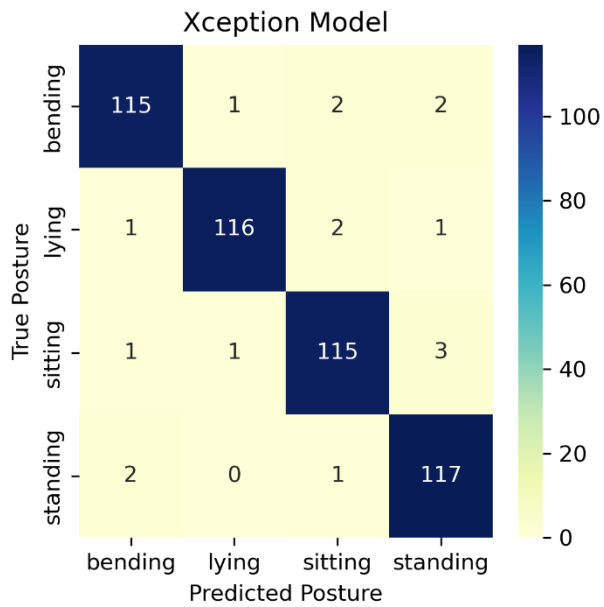
The Xception model also achieved near-perfect classification, with minimal misclassification across all classes. Its diagonal dominance in the confusion matrix aligns with its high test accuracy, confirming strong generalisation and robustness. The proposed OptiMobileX model also demonstrated balanced performance, achieving high precision across all posture classes with very few misclassifications, reinforcing its superiority among the models. DeneSVM achieved good results with the proposed hybrid approaches, correctly classifying most postures. However, a few classes were misclassified, such as sitting and bending postures. In contrast, LASSO+InceptionV3+SVM performed excellently with lying and standing classes but showed slightly weaker differentiation between sitting and bending.

The ResNet50V2 model also achieved good accuracy across most classes, though occasional errors were observed between the sitting and standing classes. Consequently, DenseNet121 demonstrates a noticeable weakness, particularly in the lying class, where a higher number of samples were misclassified into other classes. Finally, the confusion matrices showed that most CNN-based models performed well, with Xception and OptiMobileX achieving the most consistent posture detection across all classes. The SVM-based hybrids are effective, especially for distinguishing lying from standing postures; however, they have limited capacity to capture posture transitions, such as sitting versus bending. These findings suggest that the optimised OptiMobile framework offers the most reliable and balanced classification results on the Sihouettes dataset.

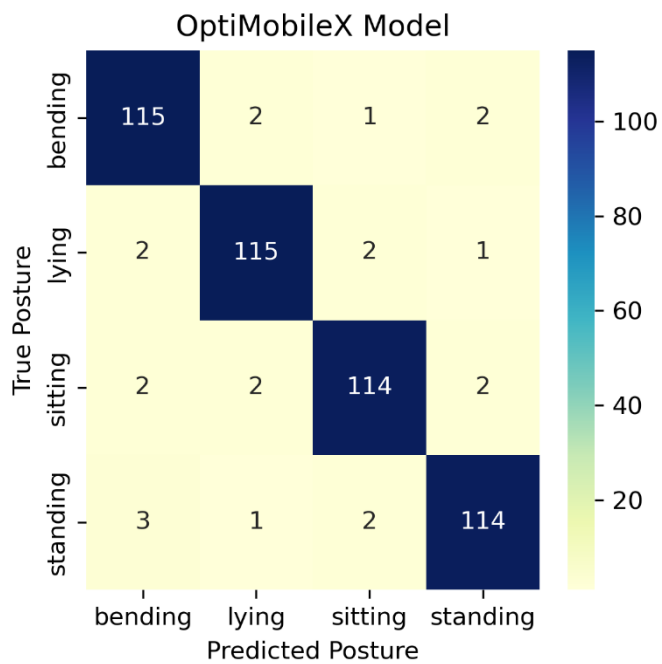




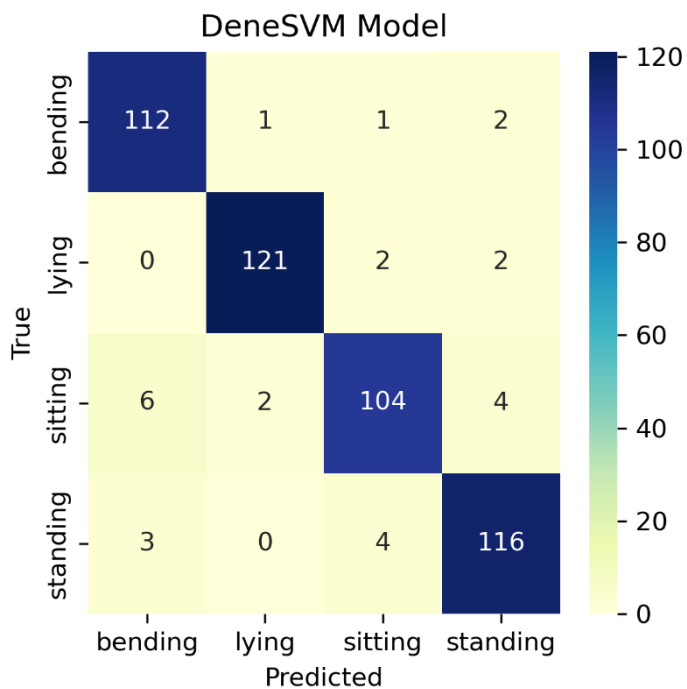
(ii)



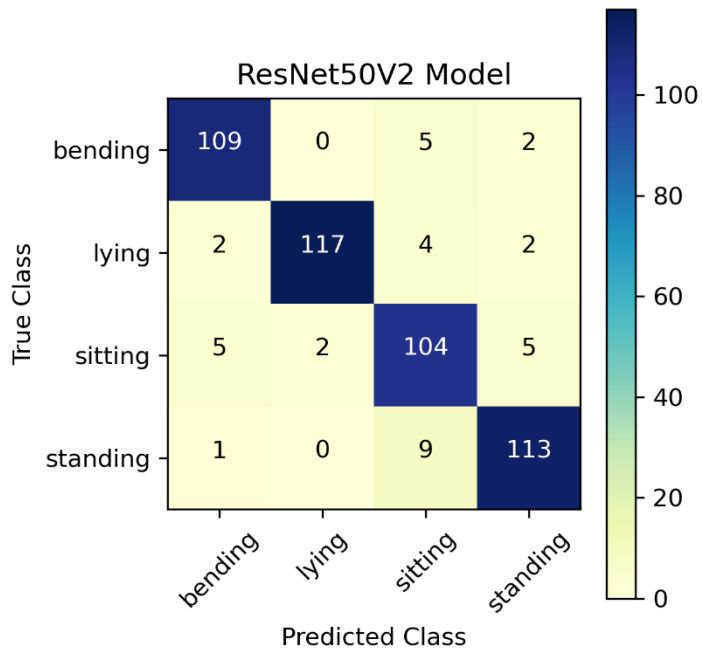
(iii)



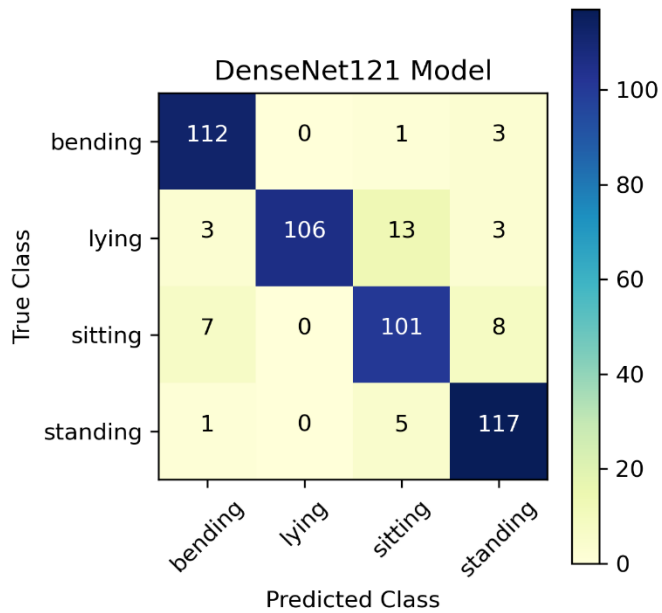
(iv)



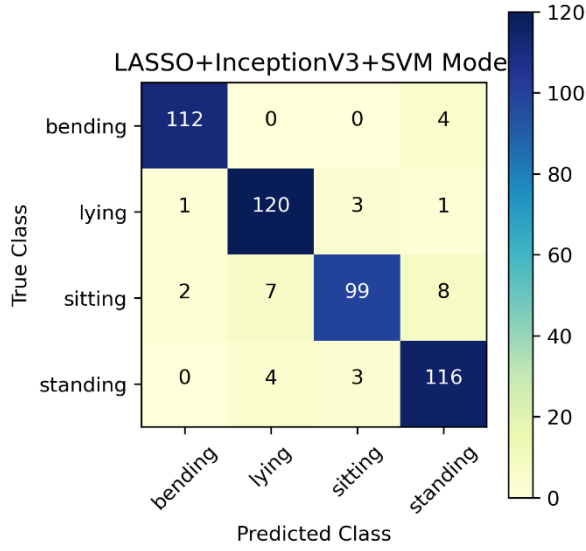
(v)



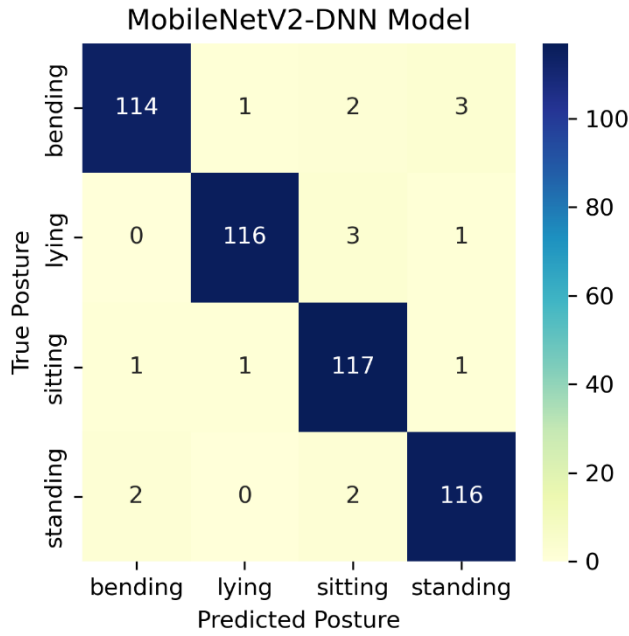
(vi)



(vii)



(viii)



(ix)

Fig. 10. Confusion matrix of DTL models on Sihuttotes Data: (i) InceptionV3; (ii) MobileNetV2; (iii) Xception, (iv) OptiMobileX, (v) ResNet50V2, (vi) DenseNet121, (vii) DeneSVM, (viii) LASSO+InceptionV3+SVM, and MobileNetV2-DNN

3.2.3. Class-wise Metrics

The results of the experiments conducted in this dissertation on the sihouettes test (10%) dataset show good model performance as shown in Table 4. It was demonstrated that the proposed lightweight hybrid OptiMobileX achieved the most balanced performance across all classes and the best overall scores, with 95.42% accuracy and 96.94% macro AUC. The second-best model is also a proposed hybrid DeneSVM, achieving 94.38% accuracy and a macro AUC of 96.23%. Five single baseline pre-trained CNN models were also implemented, and it was discovered that ResNet50V2 had the best results with 93.33% accuracy and 95.55% macro AUC, followed by MobileNetV2 having 92.71% accuracy and 95.15% macro AUC, and the third best is the Xception model with 92.08% accuracy and 94.74% macro AUC. DenseNet121 and InceptionV3 baseline trial. The proposed hybrid LASSO+InceptionV3+SVM model achieved 93.13% accuracy and 95.37% macro AUC, outperforming the baseline InceptionV3, which achieved 91.46% accuracy and 94.27% macro AUC. This supports the benefits of sparse feature selection and a margin-based classifier.

The dissertation also considered class-wise results, finding that lying is consistently easier, with near-ceiling precision/recall, while sitting is most difficult, with lower recall/F1 scores across most models. This may likely be due to visual overlap with standing and pose variability. The results also show that the MobileNetV2 model can be considered well-adapted for evaluating model size and deployment latency, while the lightweight OptiMobileX and DeneSVM models are preferred for accuracy and stability.

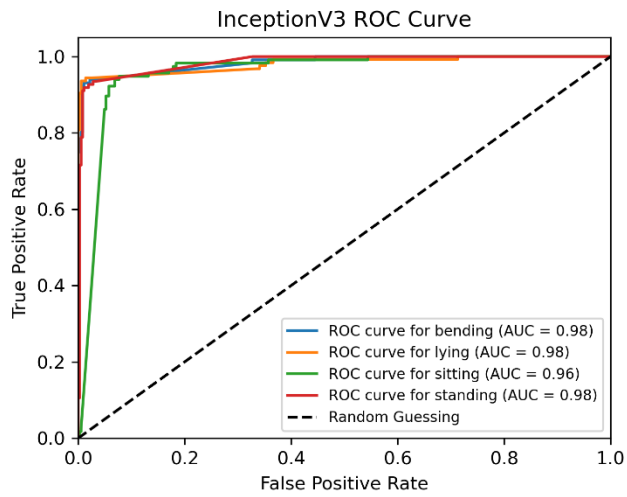
Table 4. Results of the performance evaluation obtained from the Sihuottes Test Dataset

Model	Label	Precision	Recall	F1-score	Accuracy (%)	Macro AUC (%)
InceptionV3	Bending	0.87	0.98	0.92	91.46	94.27
	Lying	0.98	0.92	0.95		
	Sitting	0.88	0.87	0.87		
	Standing	0.96	0.91	0.93		
Xception	Bending	0.97	0.96	0.96	92.08	94.74
	Lying	0.98	0.97	0.97		
	Sitting	0.96	0.96	0.96		
	Standing	0.95	0.97	0.96		
MobileNetV2	Bending	0.94	0.95	0.94	92.71	95.15
	Lying	0.99	0.92	0.95		
	Sitting	0.83	0.91	0.87		
	Standing	0.95	0.93	0.94		
DenseNet121	Bending	0.91	0.97	0.94	91.60	93.79
	Lying	1.00	0.85	0.92		

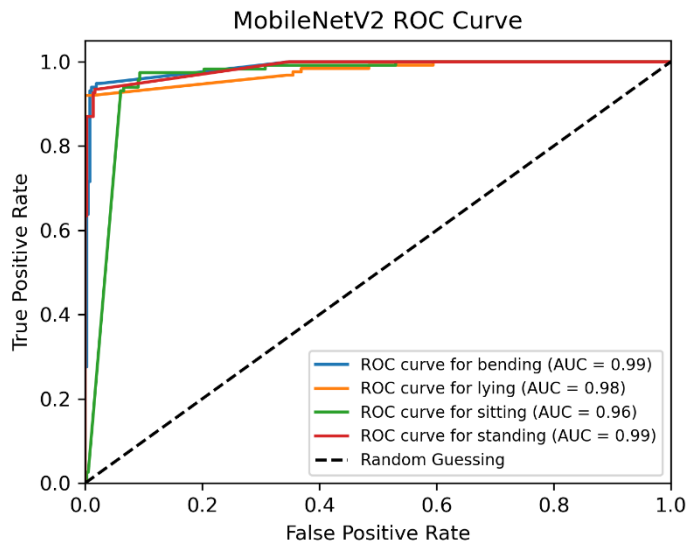
	Sitting	0.84	0.87	0.86		
	Standing	0.89	0.95	0.92		
ResNet50V2	Bending	0.92	0.93	0.93	93.33	95.55
	Lying	0.98	0.97	0.97		
	Sitting	0.86	0.93	0.90		
	Standing	0.97	0.90	0.94		
LASSO+InceptionV3 +SVM	Bending	0.97	0.97	0.97	93.13	95.37
	Lying	0.92	0.96	0.94		
	Sitting	0.94	0.85	0.90		
	Standing	0.90	0.94	0.92		
DeneSVM	Bending	0.92	0.97	0.94	94.38	96.23
	Lying	0.98	0.96	0.97		
	Sitting	0.94	0.90	0.92		
	Standing	0.94	0.94	0.94		
OptiMobileX	Bending	0.94	0.96	0.95	95.42	96.94
	Lying	0.96	0.96	0.96		
	Sitting	0.96	0.95	0.95		
	Standing	0.96	0.95	0.95		
MobileNetV2-DNN	Bending	0.96	0.95	0.95	95.40	96.92
	Lying	0.97	0.96	0.96		
	Sitting	0.94	0.96	0.95		
	Standing	0.96	0.97	0.96		

3.2.4.ROC and Precision-Recall Analysis

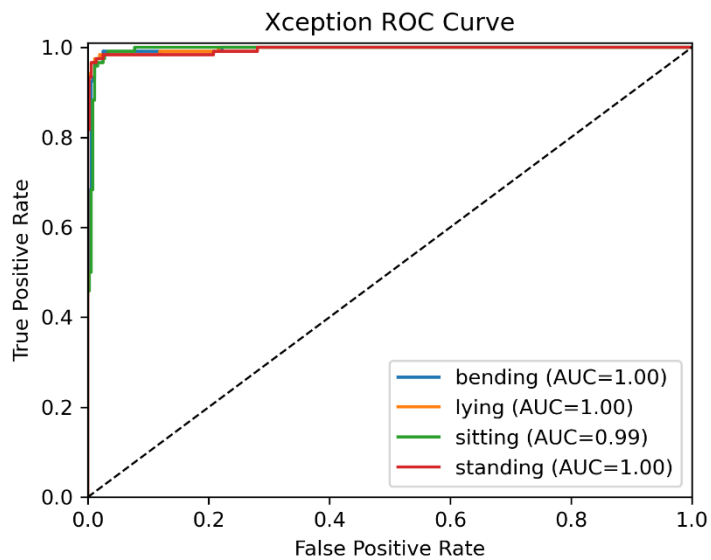
The ROC curves for the models are shown in Figure 11 (i-viii), and Figure 12 (i-viii) shows the precision-recall curves (PR) for the models. The CNN models' ROC curves nearly reached the upper-left corner, indicating they consistently maintained high AUC values. Similarly, the CNN models' PR curves nearly reached the upper-right corner, indicating they maintained high precision as recall increased, consistent with AP values of 95.0%+.



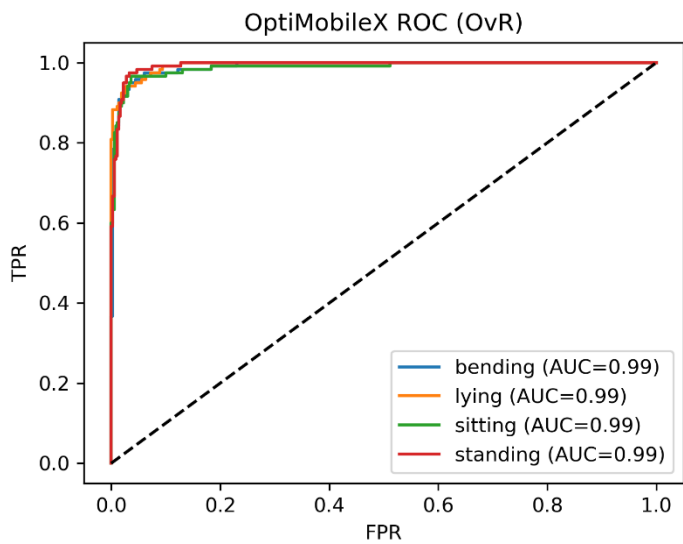
(i)



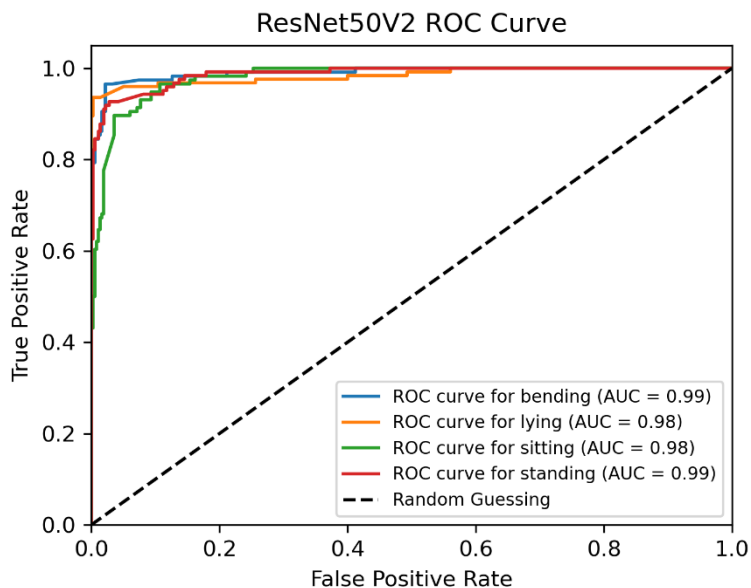
(ii)



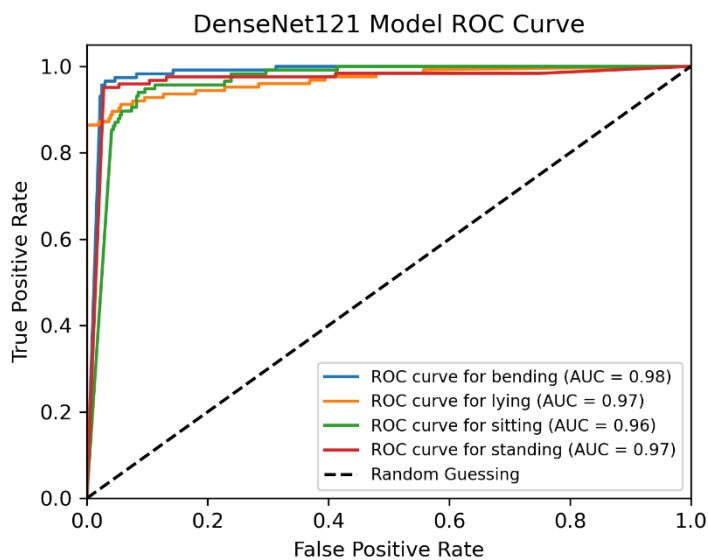
(iii)



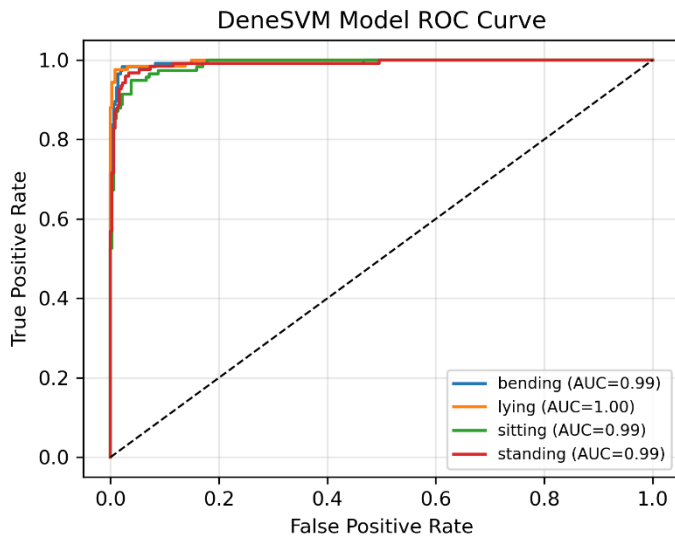
(iv)



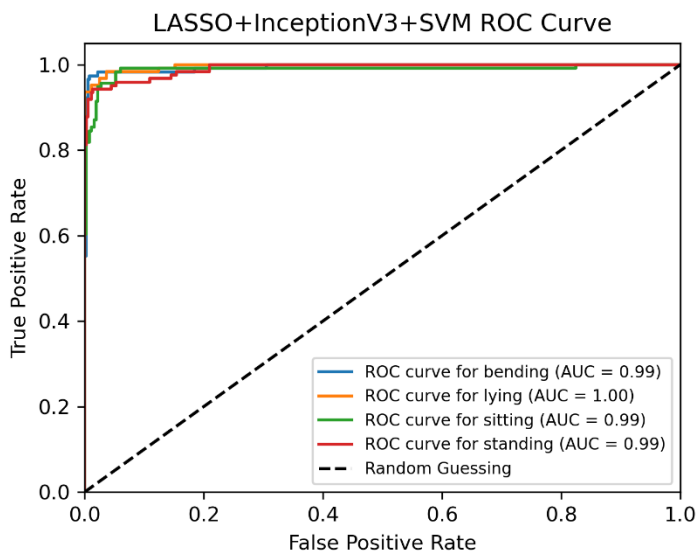
(v)



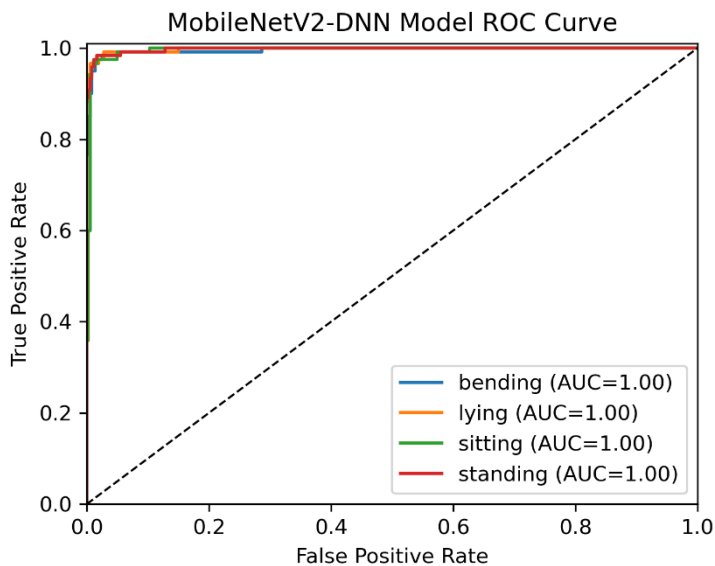
(vi)



(vii)

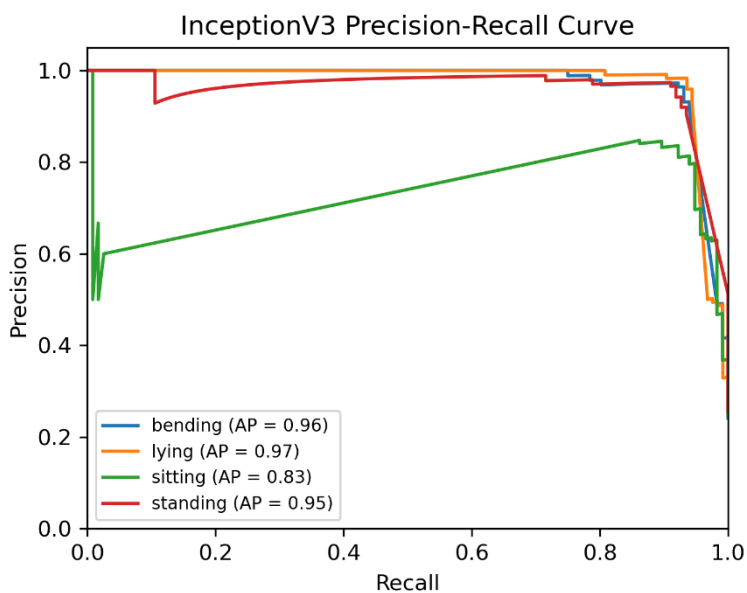


(viii)

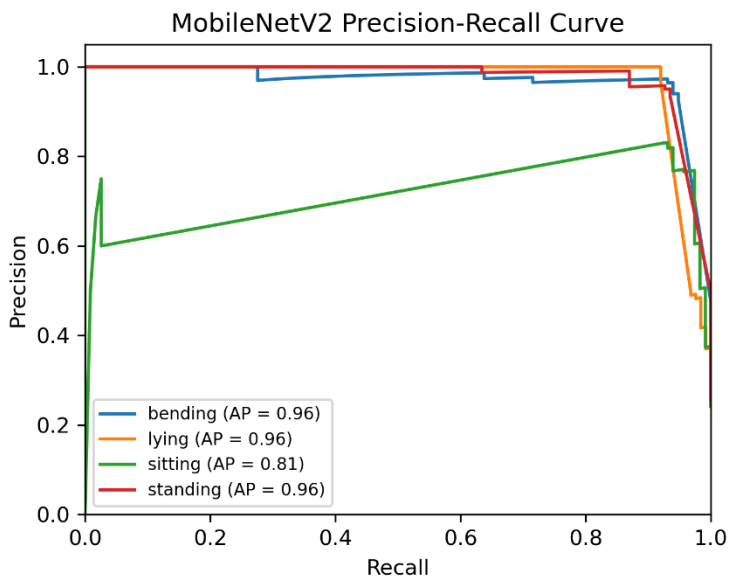


(ix)

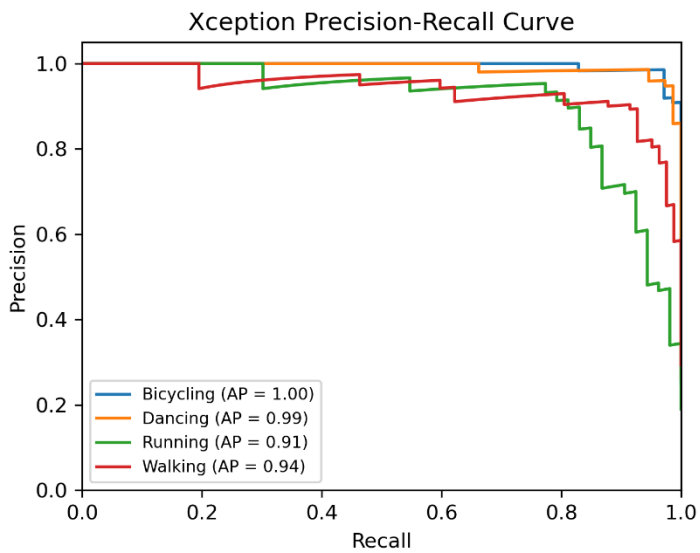
Fig. 11. ROC curve of DTL models on Sihuttos Data: (i) InceptionV3; (ii) MobileNetV2; (iii) Xception, (iv) OptiMobileX, (v) ResNet50V2, (vi) DenseNet121, (vii) DeneSVM, (viii) LASSO+InceptionV3+SVM, and (ix) MobileNetV2-DNN



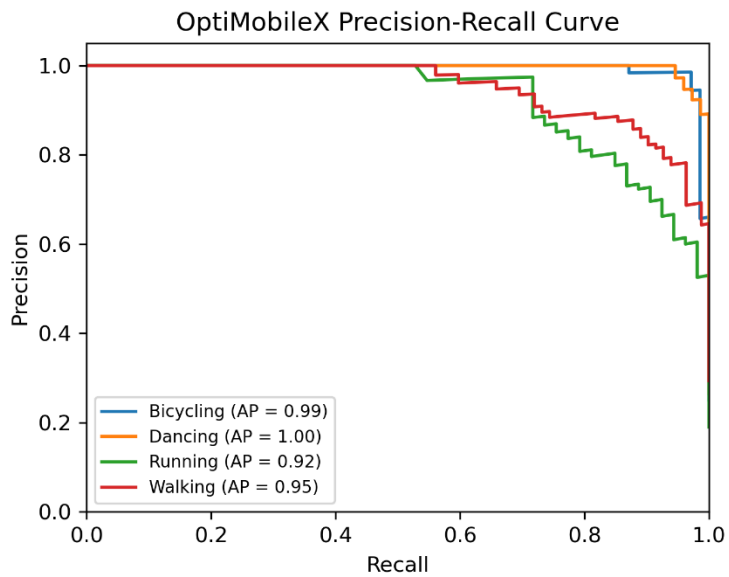
(ii)



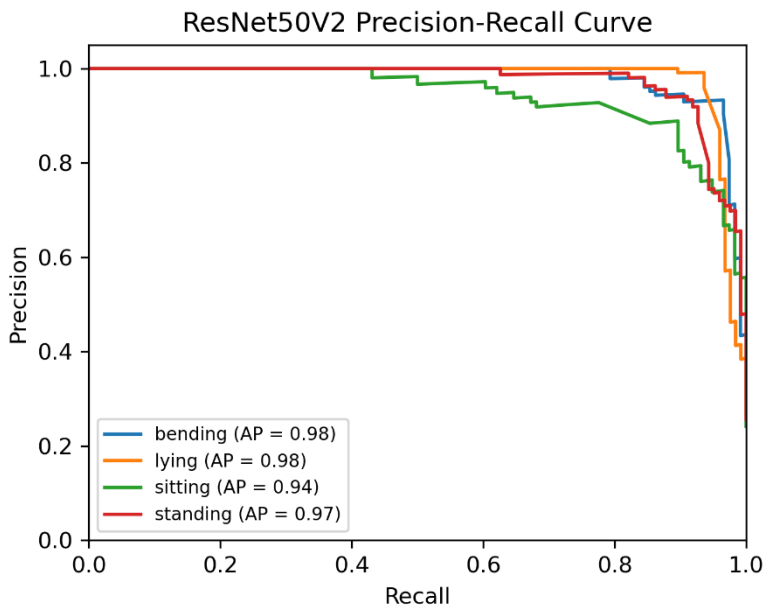
(iii)



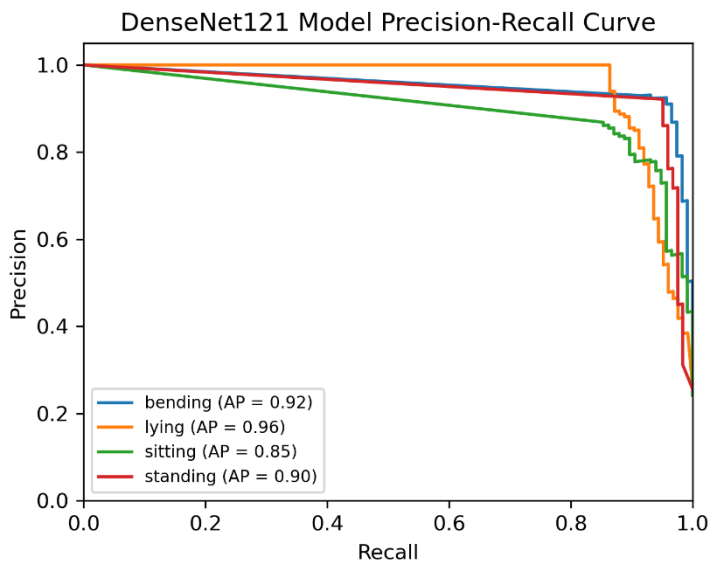
(iv)



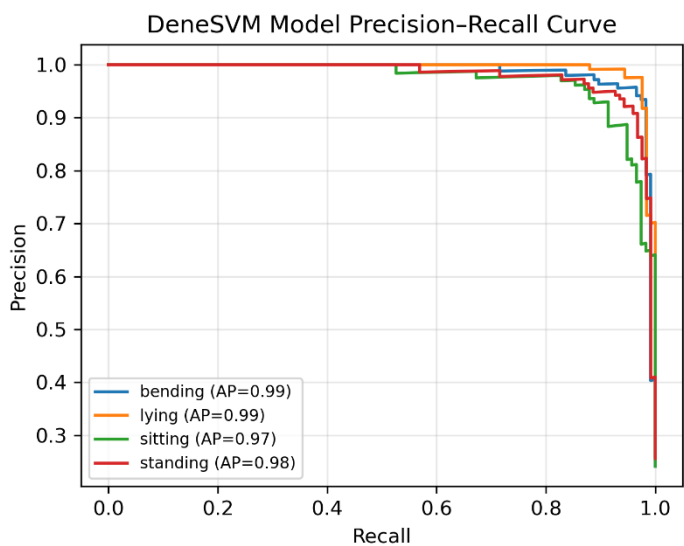
(iv)



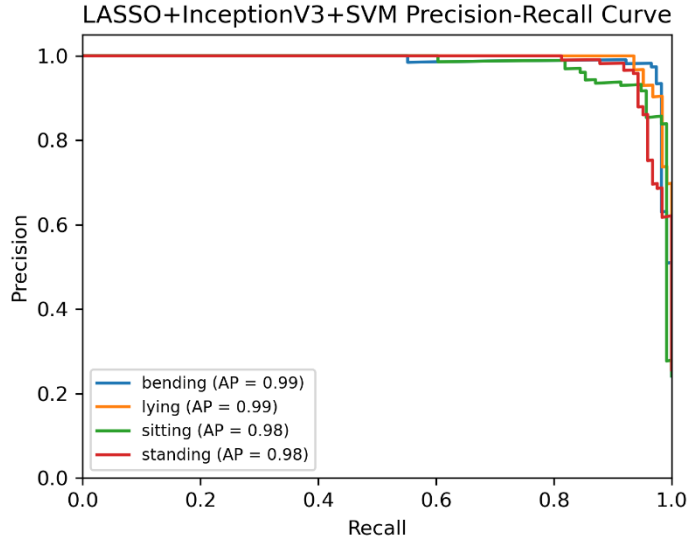
(v)



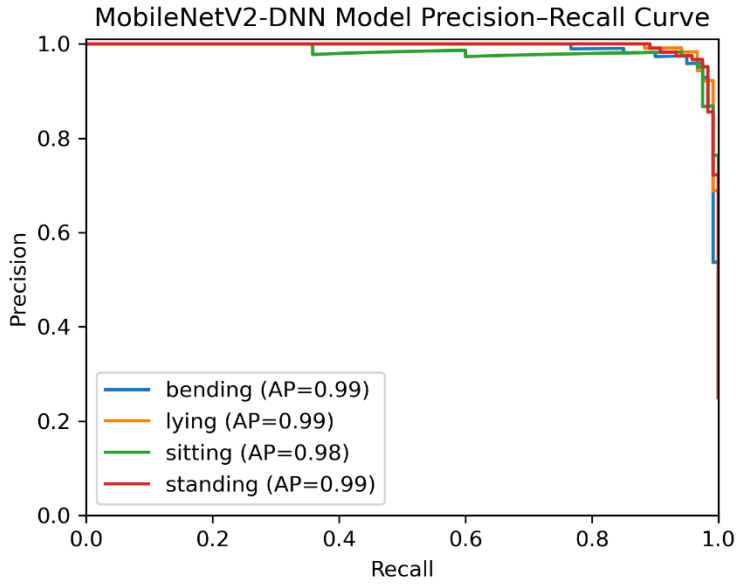
(vi)



(vii)



(viii)



(ix)

Fig. 12. Precision-Recall curve of DTL models on Sihuoottes Data: (i) InceptionV3; (ii) MobileNetV2; (iii) Xception, (iv) OptiMobileX, (v) ResNet50V2, (vi) DenseNet121, (vii) DeneSVM, (viii) LASSO+InceptionV3+SVM, and (ix) MobileNetV2-DNN

3.2.5. Visualisation of Posture Classification Results

Representative examples of the proposed model’s predictions compared with the true labels for different postures, such as sitting, standing, bending, and lying, are shown in Figure 13. The visualisation, as shown in the Figure, highlights both correctly classified postures, where the model successfully identified them, and a misclassified case, where a lying posture was incorrectly predicted as a sitting posture. The figure provides an intuitive understanding of the model’s performance beyond numerical accuracy metrics. This demonstrates that the system is robust at detecting most postures, but also highlights opportunities to enhance it through temporal modelling and advanced feature extraction to reduce errors.



Fig. 13. Sample Results of Posture Classification Showing Actual vs Predicted Labels

3.2.6. Comparative Insights and Robustness

In this dissertation, several conclusions are drawn about the models' strengths and weaknesses based on the experimental results. All the CNN-based models generalise extremely well on Silhouette data, with test accuracies exceeding 96% and validation accuracies close to chance, indicating strong feature learning. The small train-test gap for the proposed lightweight OptiMobileX suggests better regularisation, as its test loss is slightly lower than its train loss. InceptionV3 and Xception exhibit mild overfitting, with very low training loss and higher test loss. Lastly, DeneSVM consistently underperforms: it not only lags in accuracy but also underfits during training and exhibits poor class balance. In conclusion, the following were deduced:

- Xception: The model achieved the highest training accuracy and the lowest loss, resulting in near-optimal test performance. It slightly performed better than others in terms of precision, e.g., 98% on lying.
- MobileNetV2 achieves very high accuracy but is slightly lower than Xception on some classes, such as 94% precision on the bending class. This means it has a minor tendency to confuse adjacent poses.
- InceptionV3: The model is robust and balanced with slightly lower training accuracy of 98.49%, but its performance is stable.
- OptiMobileX: The proposed lightweight model achieves consistent performance, with 95.4% train-val-test accuracy and no obvious overfitting. However, in the experiments conducted, its performance did not surpass that of the baseline CNN. It essentially matches the baseline lightweight model MobileNetV2's accuracy, suggesting that the fused features were redundant.
- DeneSVM: This model was marked as weaker because it underfits, with a train accuracy of 79.6% and a test accuracy of 85.2%, and it has poor class balance, with a training precision of 78% and a test recall of 76%. This shortfall suggests it may require more tuning or a deeper feature extractor to perform better.

In summary, the CNN models all achieve nearly perfect generalisation on the silhouette dataset, with test and validation accuracies equivalent, demonstrating robustness. These findings show that on the Silhouette dataset, deep transfer models are extremely effective with minimal overfitting and that ensembling or hybrid fusion yields only marginal gains. The results demonstrate that deep CNN transfer models such as InceptionV3, MobileNetV2, and Xception are highly effective for silhouette-based posture detection, with each providing robust, comparable performance. OptiMobileX fusion matched or exceeded its constituents, and DeneSVM, while conceptually promising, underperformed here. These insights establish a strong baseline for comparison in subsequent sections on MPII and ETRI-Activity3D-LivingLab datasets.

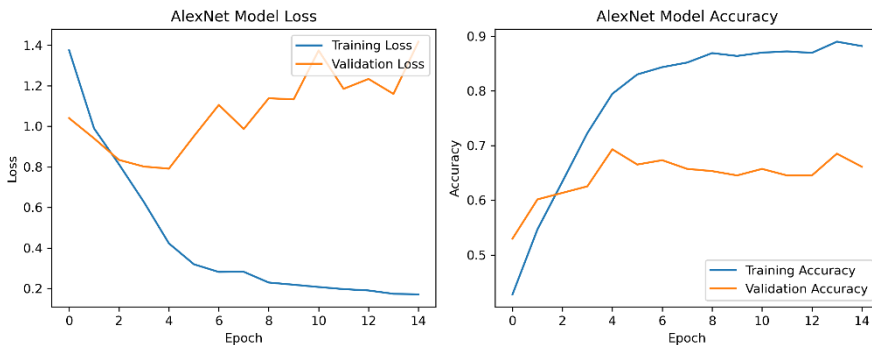
3.3. MPII Dataset and Experimental Results

In this dissertation, several deep transfer learning (DTL) models, including AlexNet, InceptionV3, MobileNetV2, Xception, and the proposed lightweight OptiMobileX, were evaluated on the MPII Human Pose Dataset for human posture detection. The objective was to compare each model's capacity to learn from the training set and generalize effectively to unseen test data.

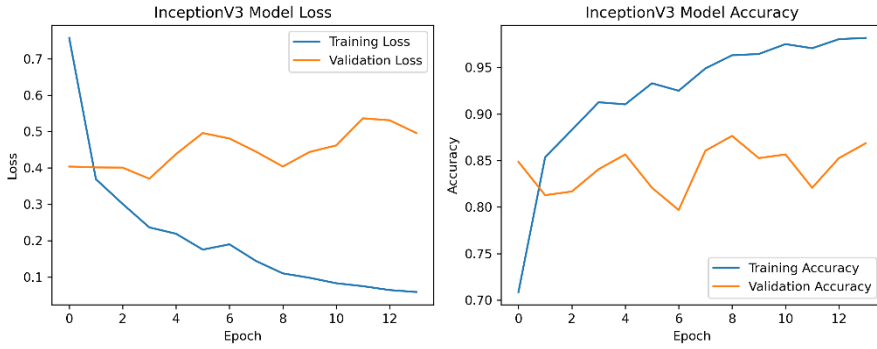
Training and validation accuracy and loss were presented (Table 5, Fig. 10), alongside per-class precision, recall, F1-score, overall accuracy, and AUC values (Table 6), as well as confusion matrices (Fig. 13), to systematically analyse the strengths and weaknesses of each model.

3.3.1. Training and Validation Performance

The training and validation curves for all models are presented in Table 5 and Figure 14. All five models achieved very high training accuracy, indicating they fit the MPII dataset well. For instance, MobileNetV2 achieved a training accuracy of 94.47%, Xception 97.14%, and OptiMobileX 99.98%. However, their validation accuracies are substantially lower in each case: MobileNetV2 85.26%, Xception 87.25%, and OptiMobileX 91.63%, indicating overfitting. This is also evident in the loss values: OptiMobileX had a near-zero training loss of 0.0004 but a larger validation loss of 0.5841, while Xception had a training loss of 0.0628, which is much smaller than its validation loss of 0.3720. Fig. 14 shows that all the model's training accuracy curves rise rapidly to a plateau, whereas the validation accuracy plateaus at a lower level. For instance, fig. Fig. 14(v) shows that OptiMobileX achieves almost 100% training accuracy with vanishing loss, while Fig. 14(iii-iv) show similar trends for MobileNetV2 and Xception. AlexNet saturates at a much lower training accuracy of 66%, and this exhibits a smaller training-validation gap. Overall, these results suggest that the deeper models (InceptionV3, Xception) and the proposed lightweight OptiMobileX model learned the training data very well; however, their generalisation is limited by overfitting, as seen in the curves in Fig. 10.



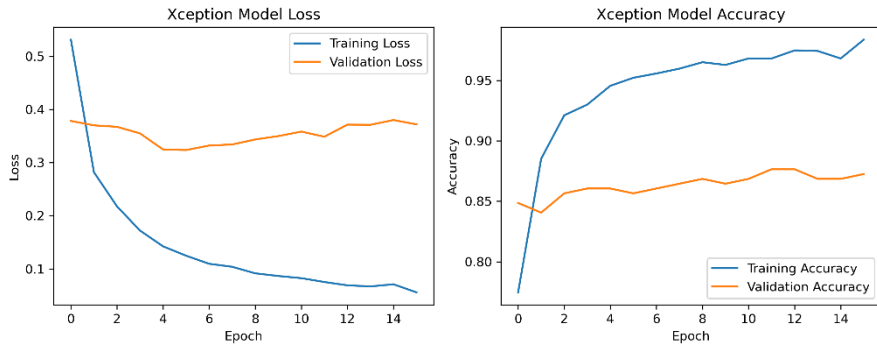
(i)



(ii)



(iii)



(iv)

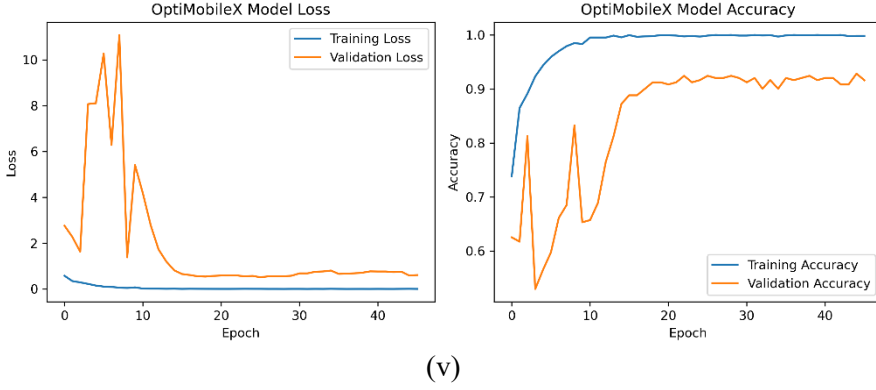


Fig. 14. Accuracy and loss graphs of DTL models on MPII Dataset: (i) Alexnet, (ii) InceptionV3, (iii) MobileNetV2, (iv) Xception, and (v) OptiMobileX

3.3.2. Testing Accuracy and Class-wise Metrics

Each Model's test-set precision, recall, and F1-score for each posture class, along with their overall accuracy and AUC, are presented in Table 5. The worst-performing model is AlexNet, with an overall test accuracy of 66% and an AUC of 85.75%. AlexNet has a very low recall of 38% for dancing and 67% for walking, indicating that many postures of these classes were misclassified by this model. By contrast, the pre-trained CNNs with TL all achieved much higher accuracy: InceptionV3 had 87% accuracy and 97% AUC, MobileNetV2 and Xception had 89% accuracy and 98% to 99% AUC, and lastly, the proposed lightweight OptiMobileX model led with the highest accuracy of 92% and 99% AUC.

Class-wise, all non-AlexNet models perform well on bicycling and dancing, achieving precision and recall over 90%. For instance, MobileNetV2 achieved 94% precision and 97% recall on Bicycling, and 92% precision and 97% recall on Dancing. In contrast, running and walking classes are more challenging. All models show a decrease in performance when running with a precision of 77% to 81% and a recall of 76% to 87%, and when walking, they had 84% to 92% precision and 73% to 83% recall. For instance, InceptionV3's F1-score for running and walking is only 77%-79%. MobileNetV2 had 81% precision and 81% recall for running. The worst case is Xception, which achieves a recall of 87% while running and 73% while walking. The 73% value is the lowest recall across all classes and models, and these trends indicate systematic confusion between running and walking.

The proposed lightweight OptiMobileX significantly improves performance in these challenging classes while maintaining high performance elsewhere. Its overall accuracy is 92%, and its AUC is 99%, both of which are the highest. All class precisions and all class recalls are above 83%. In particular, OptiMobileX achieves 92% precision and 97% recall for bicycling, 96% precision and 96% recall for dancing, 83% precision and 92% recall for running, and 94% precision and 83% recall for walking. Even in its weakest class (walking), OptiMobileX's recall of 83%

surpasses that of the Xception model (73%). In summary, compared with standard pre-trained models, OptiMobileX achieves uniformly high precision and recall across all classes, indicating excellent generalisation despite being a lightweight model.

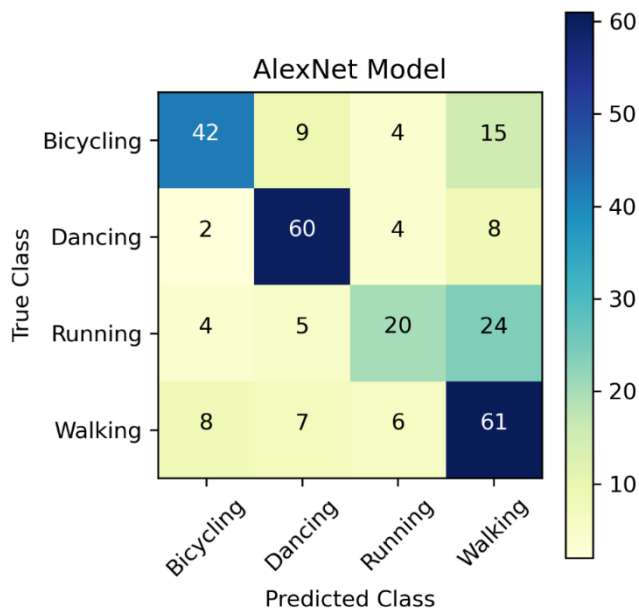
Table 5. Results of the performance evaluation obtained from the MPII Test Dataset

Model	Label	Precision	Recall	F1-score	Accuracy	AUC
AlexNet	Bicycling	0.75	0.60	0.67	0.66	85.75
	Dancing	0.74	0.81	0.77		
	Running	0.59	0.38	0.46		
	Walking	0.56	0.74	0.64		
InceptionV3	Bicycling	0.97	0.94	0.96	0.87	97.00
	Dancing	0.87	0.99	0.92		
	Running	0.77	0.77	0.77		
	Walking	0.84	0.76	0.79		
MobileNetV2	Bicycling	0.94	0.97	0.96	0.89	98.00
	Dancing	0.92	0.97	0.95		
	Running	0.81	0.81	0.81		
	Walking	0.87	0.80	0.84		
Xception	Bicycling	0.90	1.00	0.95	0.89	98.75
	Dancing	0.95	0.99	0.97		
	Running	0.78	0.87	0.82		
	Walking	0.92	0.73	0.82		
OptiMobileX (Proposed model)	Bicycling	0.92	0.97	0.94	0.92	99.00
	Dancing	0.96	0.96	0.96		
	Running	0.83	0.92	0.88		
	Walking	0.94	0.83	0.88		

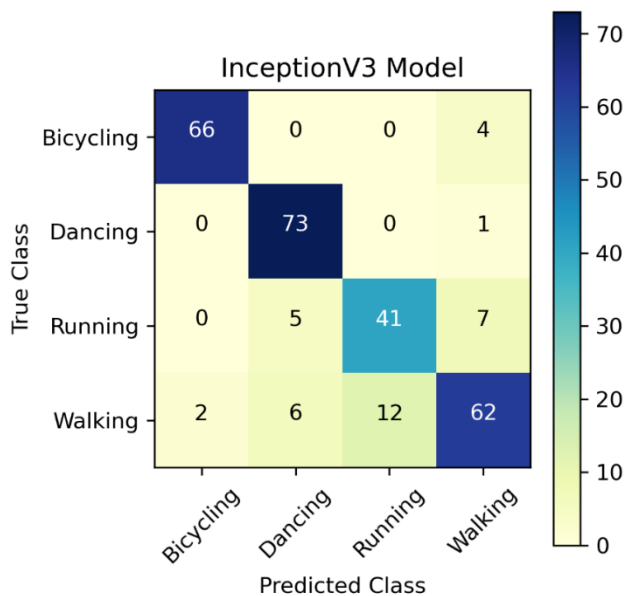
3.3.3. Confusion Matrix, ROC and PR Curves Analysis

The confusion matrix for the models is shown in Figure 15. The AlexNet model, as shown in Figure 15(i), had the lowest diagonal, indicating that many dancing classes were misclassified into other classes, consistent with its 38% recall on dancing. By comparison, InceptionV3, MobileNetV2, and Xception, as shown in Fig. 15(ii-iv), exhibit strong diagonals for bicycling and dancing, with few errors; however, off-diagonal cells remain for distinguishing running from walking. For instance, each of these models has a certain percentage of running examples misclassified as walking and vice versa, which reflects the mid-range recall reported in Section 4.3.2. OptiMobileX's matrix, as seen in Fig. 15(v), is the best, with nearly all classes lying on the diagonal, indicating correct classification. There are a few errors, mostly a slight mix between running and walking, but these are far fewer than in the others. This visual pattern matches the numerical metrics: OptiMobileX's high values, such as 92% precision and 97% recall on most classes, are reflected in the dominance of diagonal entries in its confusion matrix.

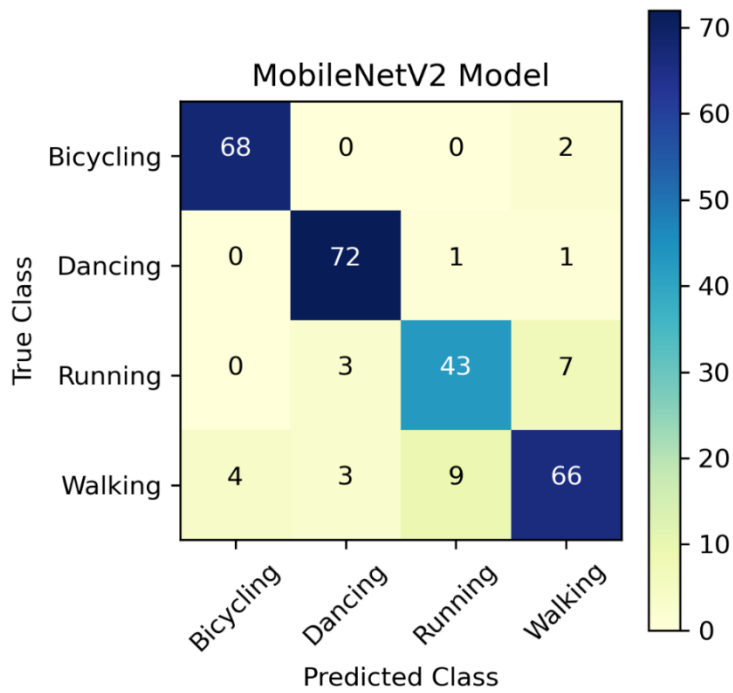
The ROC curves for all five models are shown in Fig. 16, and Fig. 17 shows the PR curves for all the models. All models performed well across all classes, as shown in the curves. Predictions of human postures using the proposed models are shown in Fig. 18.



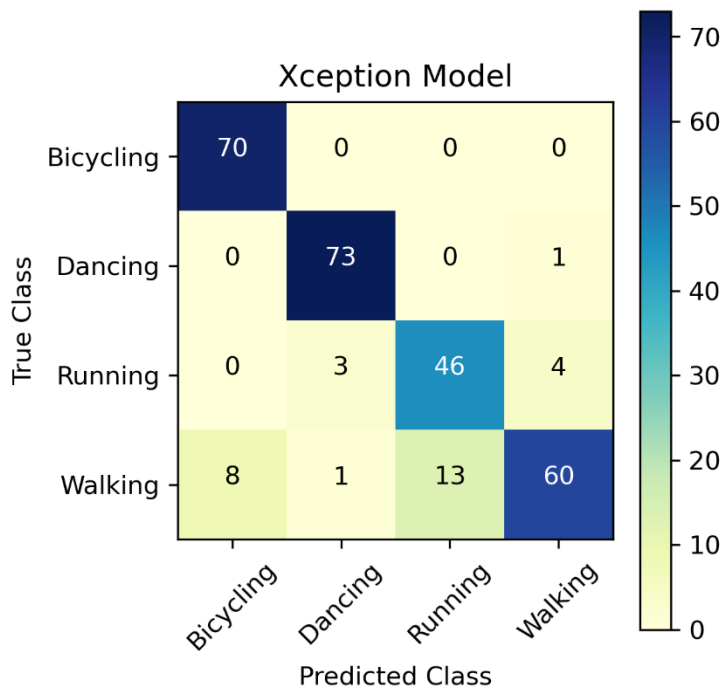
(i)



(ii)



(iii)



(iv)

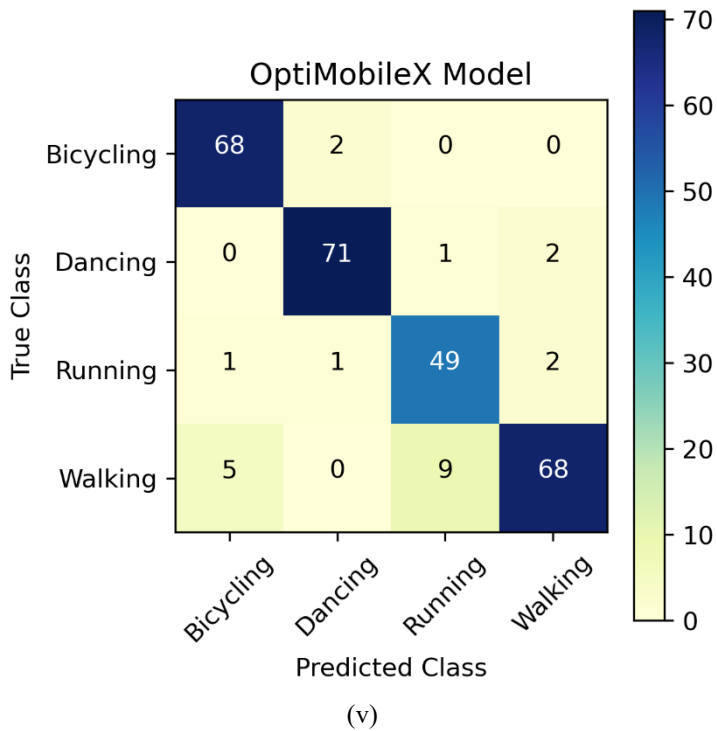
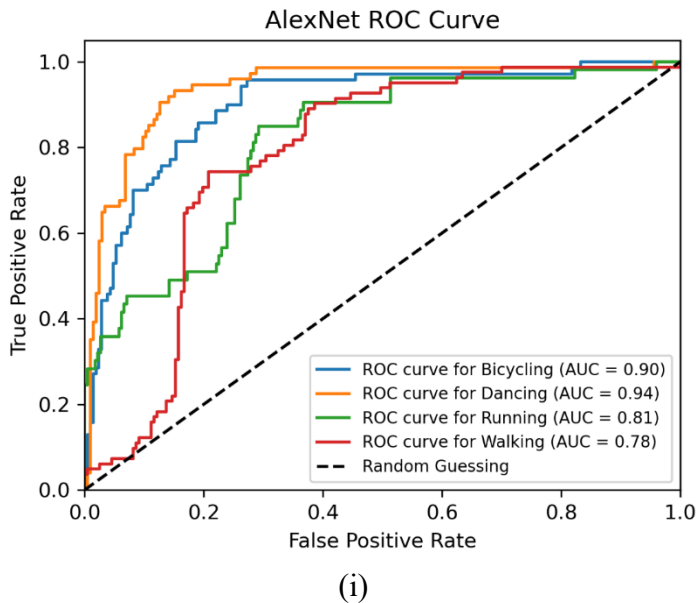
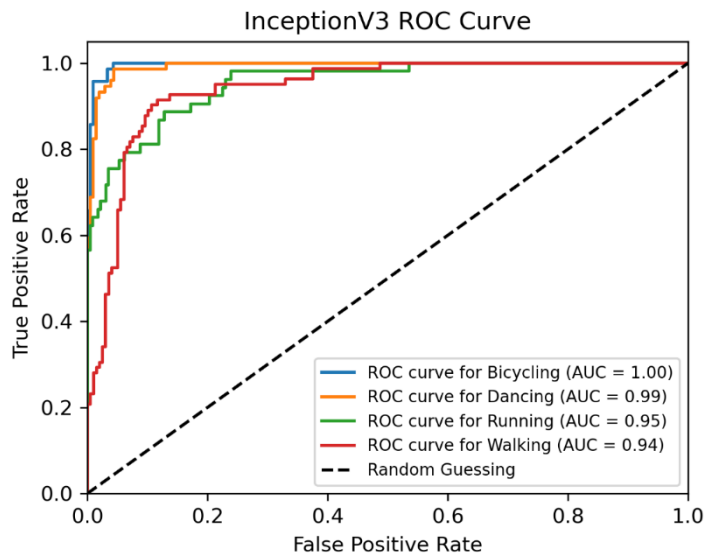
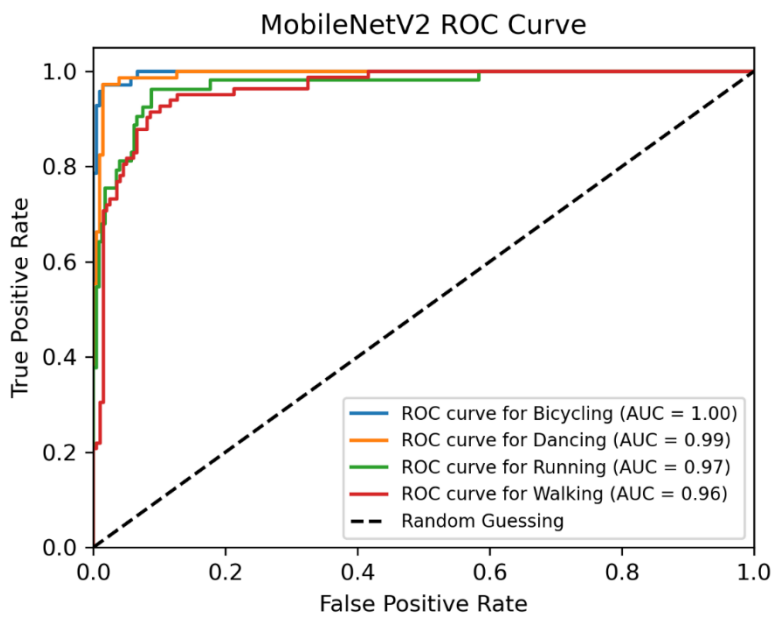


Fig. 15. Confusion matrix of DTL models on MPII: (i) AlexNet; (ii) InceptionV3; (iii) MobileNetV2; (iv) Xception, and (v) OptiMobileX

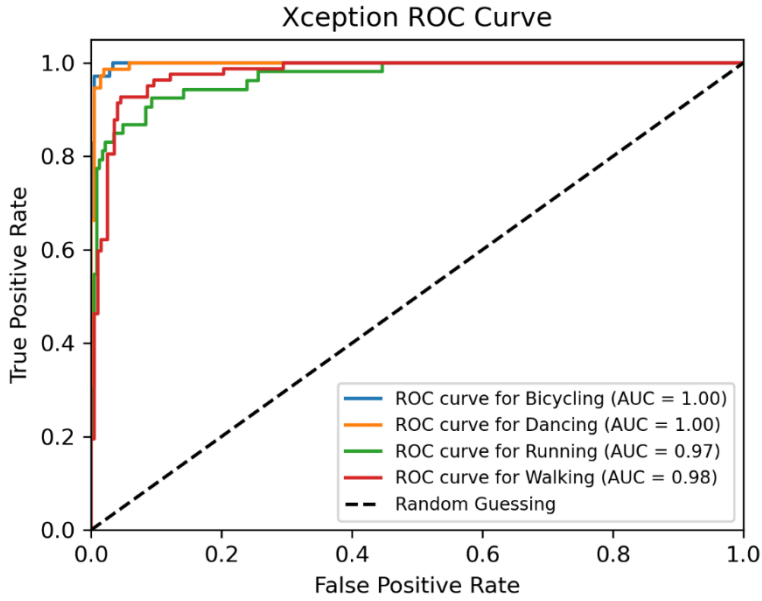




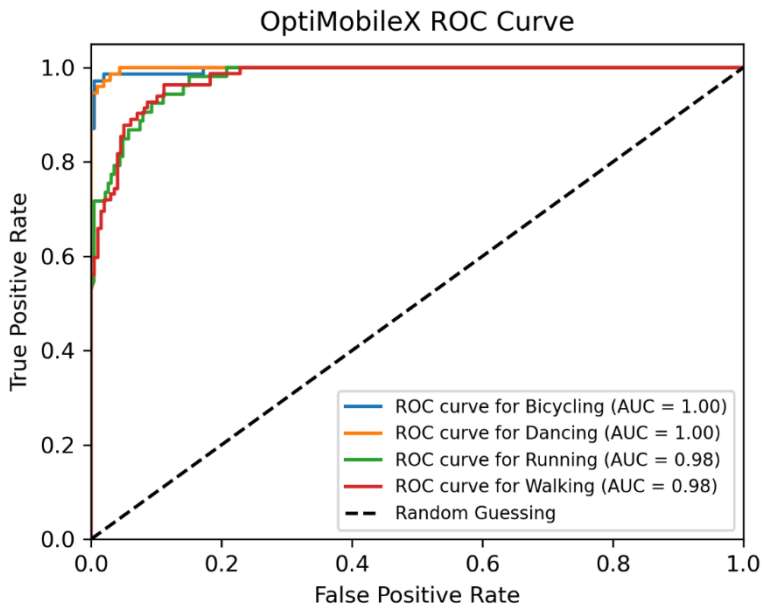
(ii)



(iii)

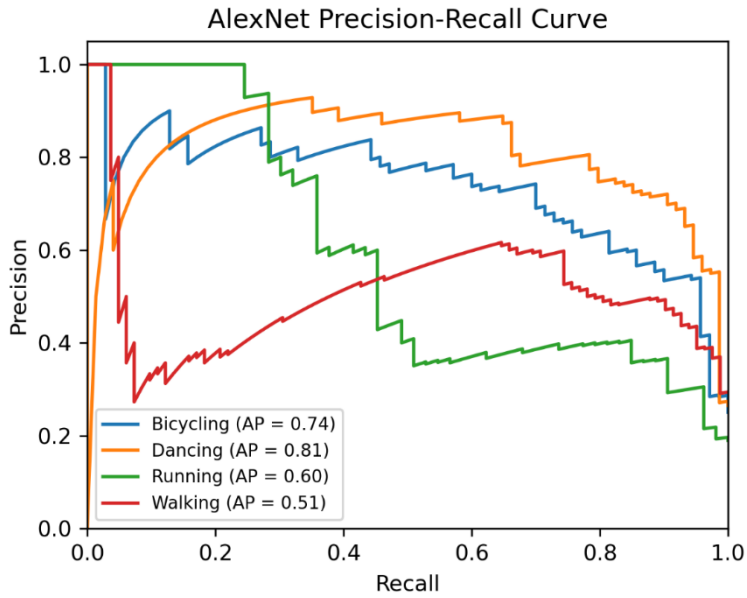


(iv)

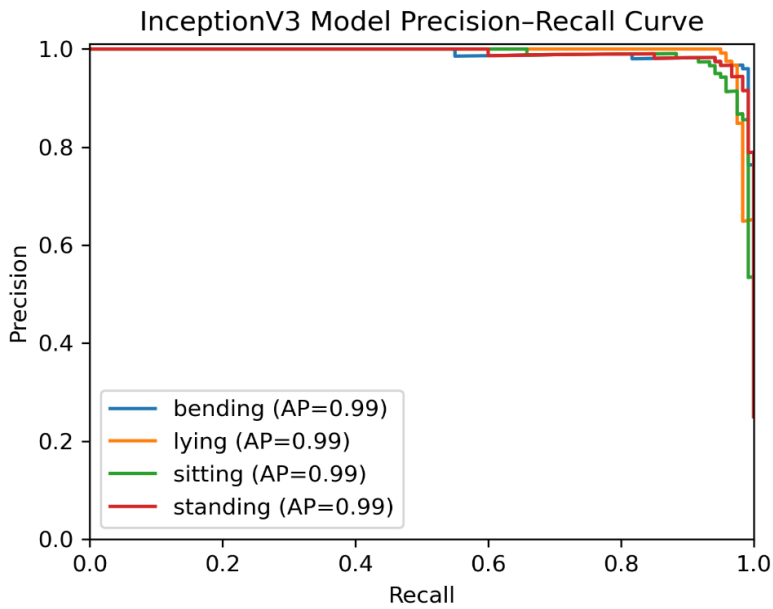


(v)

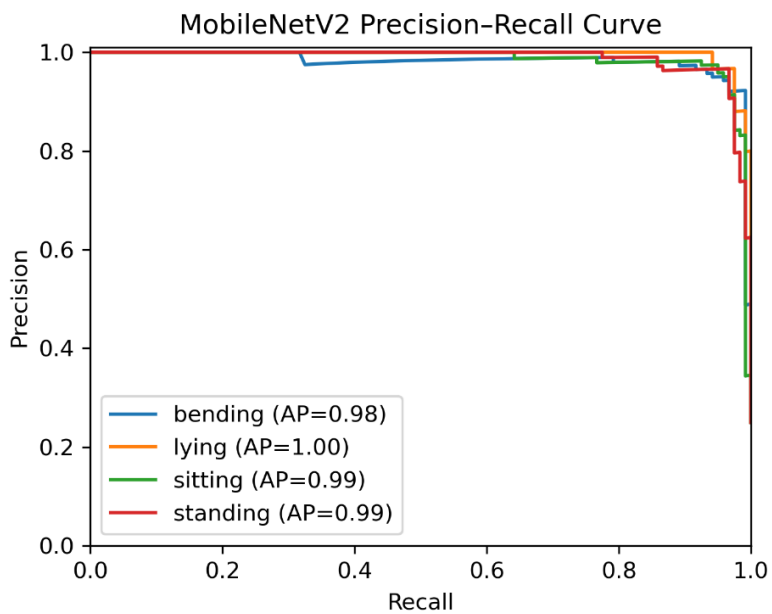
Fig. 16. ROC curve of DTL models on MPII: (i) AlexNet; (ii) InceptionV3; (iii) MobileNetV2; (iv) Xception, and (v) OptiMobileX



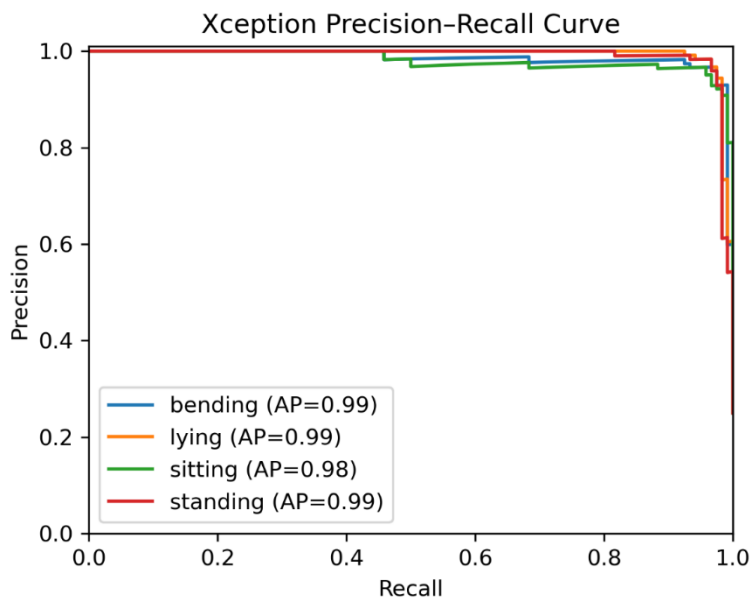
(i)



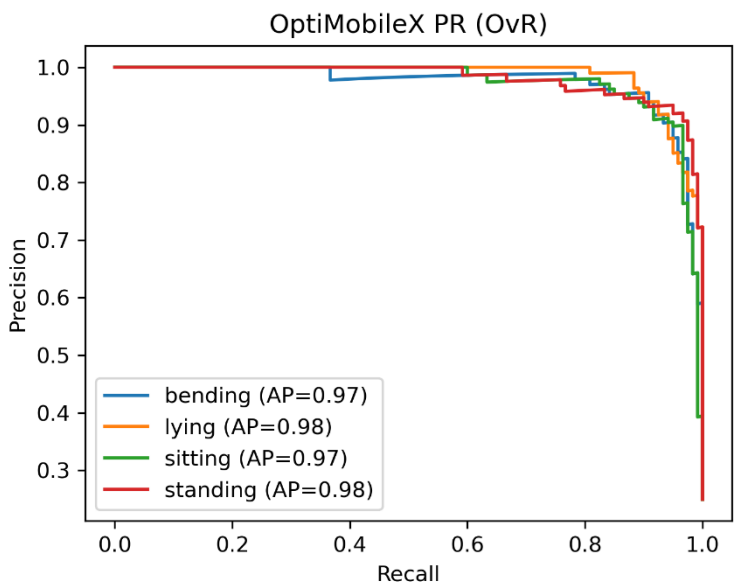
(ii)



(iii)

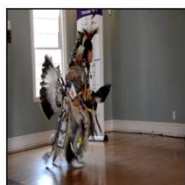


(iv)

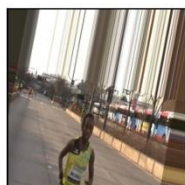


(v)

Fig. 17. PR of DTL models on MPII: (i) AlexNet; (ii) InceptionV3; (iii) MobileNetV2; (iv) Xception, and (v) OptiMobileX



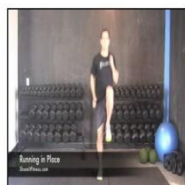
Actual: Dancing
Predicted: Dancing



Actual: Running
Predicted: Running



Actual: Dancing
Predicted: Dancing



Actual: Running
Predicted: Running



Actual: Walking
Predicted: Running



Actual: Dancing
Predicted: Dancing



Actual: Bicycling
Predicted: Bicycling



Actual: Dancing
Predicted: Dancing



Actual: Bicycling
Predicted: Bicycling



Actual: Bicycling
Predicted: Bicycling



Fig. 18. Prediction of the postures using the proposed model

3.3.4. Comparative Strengths and OptiMobileX Performance

Overall, AlexNet is clearly underpowered for human posture detection on the MPII dataset. This was evidenced by its low accuracy of 66% and high error rates on certain classes. The deep and lightweight models, such as InceptionV3, MobileNetV2, and Xception, all perform well, achieving over 87% accuracy, but each has its own weaknesses. InceptionV3 is very precise on bicycling and dancing but slightly weaker on running and walking, with an F1 score of 77% to 79%. MobileNetV2 balances performance with efficiency (89% accuracy), though its running precision and recall lag by 81%. Xception achieves a similar 89% accuracy and the highest AUC of 98.75% among the Pre-trained TL CNN models, but its accuracy drops from training (97.14%) to test (89%), suggesting overfitting, and it has a notably low recall of 73% on walking.

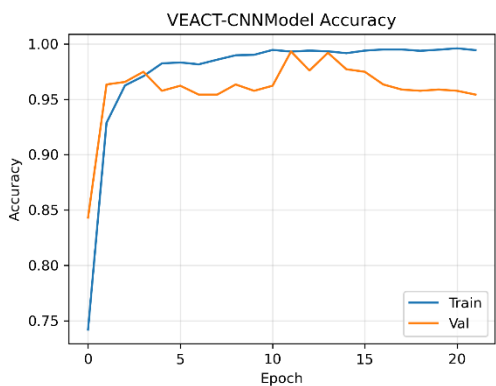
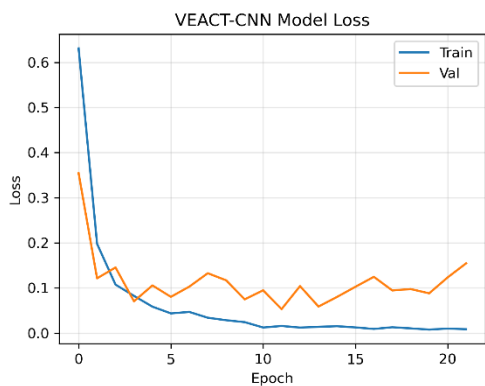
This proposed lightweight model, OptiMobileX, consistently outperforms the other models. It achieves 92% test accuracy and an AUC of 99%, and its precision and recall for each class meet or exceed those of other TL-pretrained CNN models. OptiMobileX not only achieved high metrics on classes such as dancing (recall: 97%), but also improved performance on other hard-to-detect classes, including running (recall: 92%) and walking (recall: 83%). In particular, it corrects Xception's biggest failure by raising walking recall from 73% to 83%. These results show that OptiMobileX significantly outperforms the baseline TL CNN models in overall accuracy and in handling the most confusing posture classes, demonstrating excellent generalisation of human posture detection on the MPII dataset.

3.4. ETRI-Activity3D-LivingLab Dataset and Experimental Results

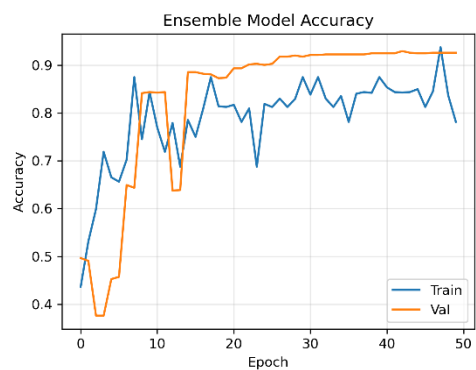
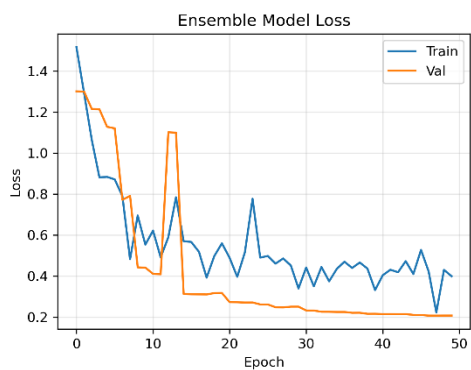
The ETRI-Activity3D-LivingLab Dataset provides a large-scale benchmark for human action recognition in real-world living lab environments. The aim of this experiment, as presented in this dissertation, is to evaluate and compare nine DTL models, VECT-CNN, Ensemble, CNN, Xception, VGG16, LSTM, GRU, and the proposed LitePose-GRU, on this dataset. Two temporal models, LSTM and GRU, were also introduced alongside baseline models, including VGG16, Xception, and a conventional CNN, for the dissertation experiments. The comparison emphasises not only classification accuracy and loss behaviour across training, validation and testing pipelines but also computational efficiency, as summarised in Tables 6 and 7 and illustrated in Fig. 19.

3.4.1. Training and Validation Performance

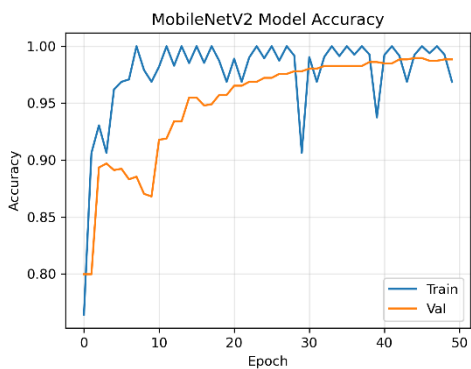
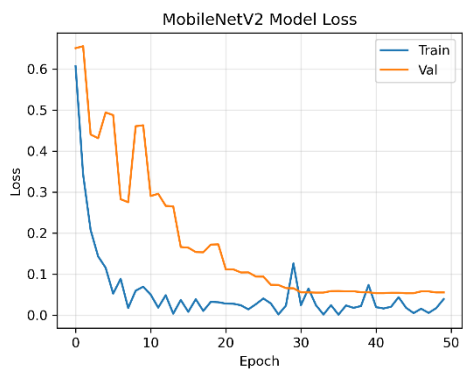
Nine models were trained and evaluated on the ETRI dataset, as demonstrated in Figs. 19, 20, 21, and 22. It was discovered that MobileNetV2 performed best with a test accuracy of 97.66% and a small generalization gap from training of 100% to 98.84% validation, indicating a strong but slightly over-confident fit, whereas VGG 16 and the plain CNN are close behind (97.29% and 97.27%) with VGG16 having the lowest test loss of 0.0625 that is better-calibrated probabilities at similar accuracy. The proposed VECT-CNN also achieved highly competitive results, with test accuracy of 97.56% and test loss of 0.0854, maintaining a balance between training and validation performance and suggesting robust generalisation. Xception was found below the top cluster with a test accuracy of 96.00%. The temporal sequential models also achieved 92.46% test accuracy with the LSTM and 93.13% with the GRU, indicating gaps and high losses, suggesting that the temporal pipeline needs improvement. LitePose-GRU has an excellent validation result of 98.63%, but a drop in test (94.97%) and a high test loss of 0.1630. The ensemble is an outlier, with low training accuracy of 78.12%, yet good validation and test accuracies of 92.59% and 95.31%, respectively, consistent with strong regularisation. A better weighting or training schedule may unlock its potential. In conclusion, compact image backbones such as MobileNetV2, VGG16, CNN, and VECT-CNN generalise best, while temporal and ensemble models would benefit from tuning.



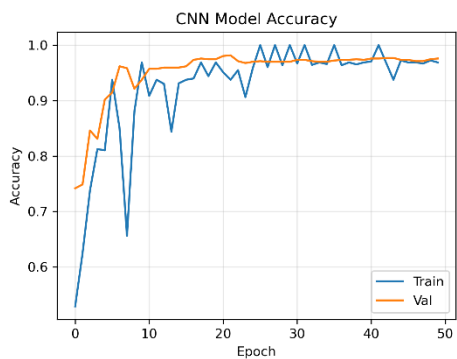
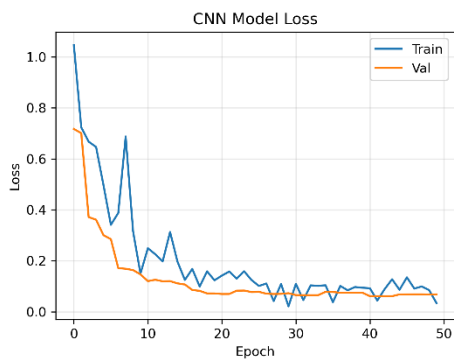
(i)



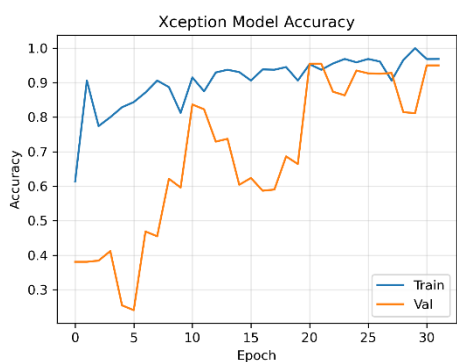
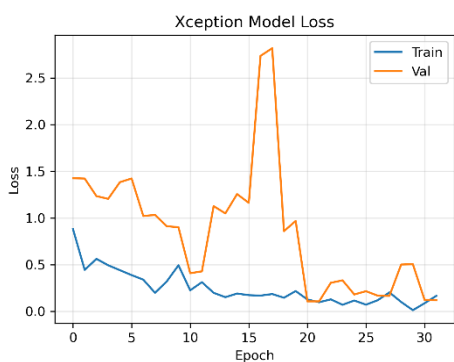
(ii)



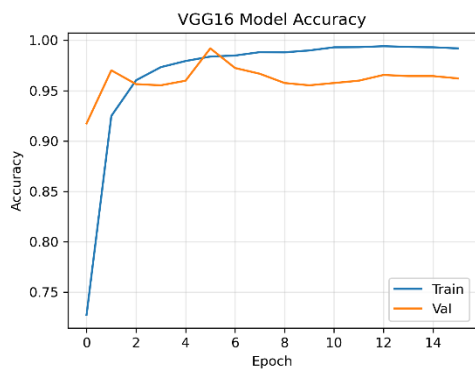
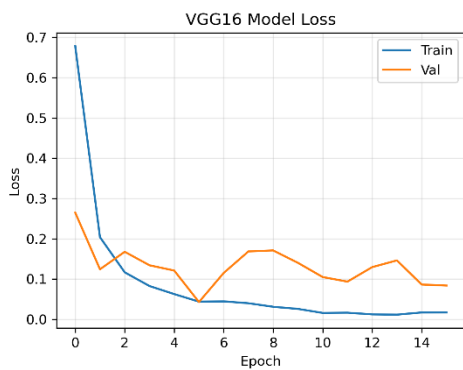
(iii)



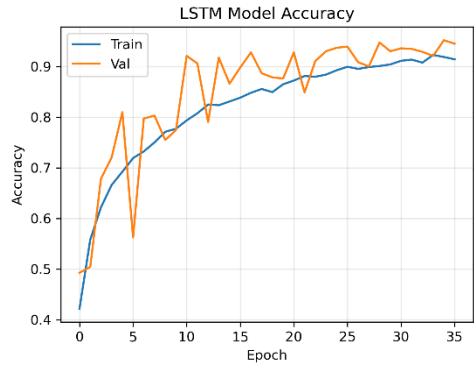
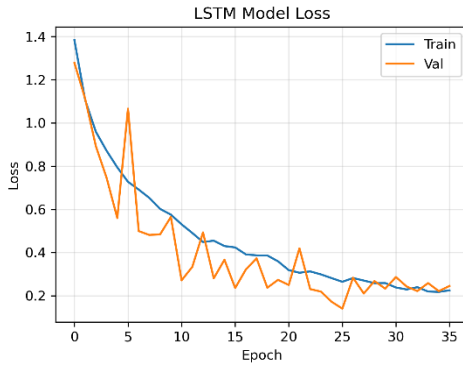
(iv)



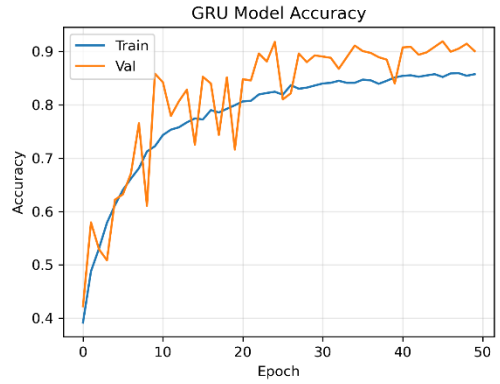
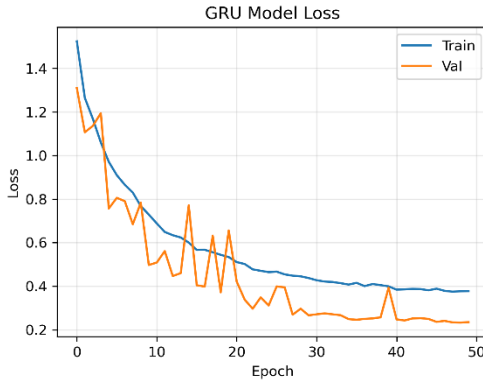
(v)



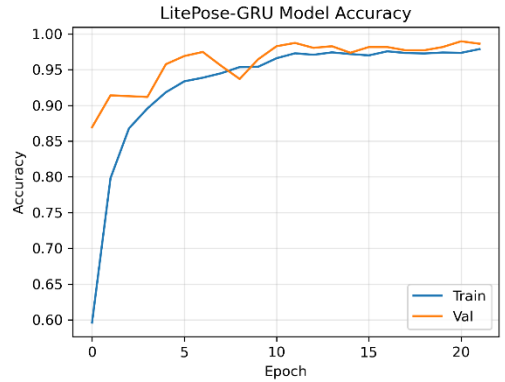
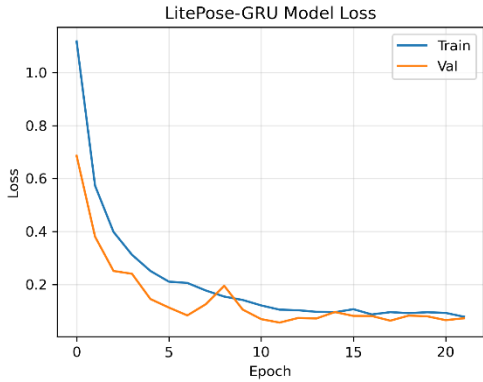
(vi)



(vii)



(viii)

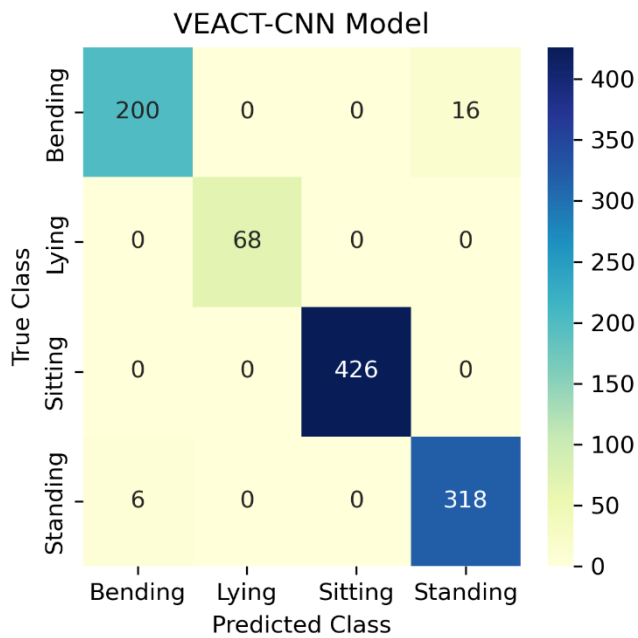


(ix)

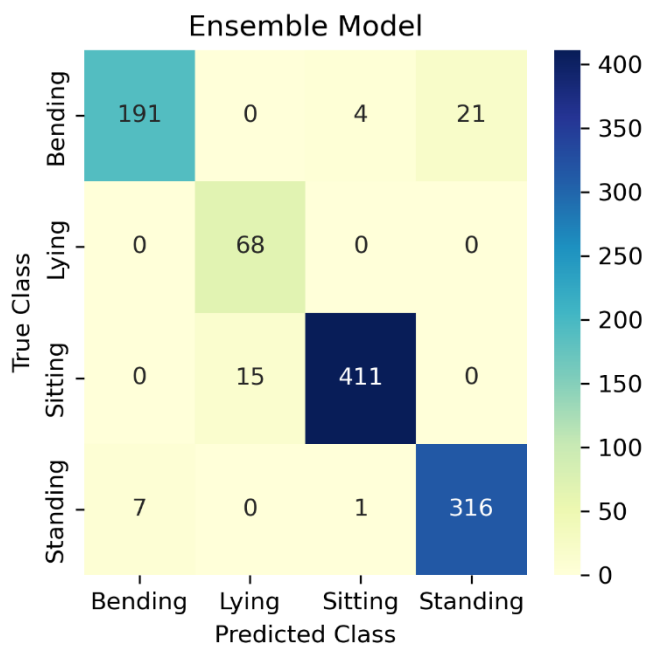
Fig. 19. Accuracy and loss graphs of DTL models ETRI-Activity3D-LivingLab Dataset: (i) VECT-CNN, (ii) Ensemble, (iii) MobileNetV2, (iv) CNN, (v) Xception, (vi) VGG16, (vii) LSTM, (viii) GRU, and (ix) LitePose-GRU

3.4.2.Confusion Matrix, ROC and PR Curves Analysis

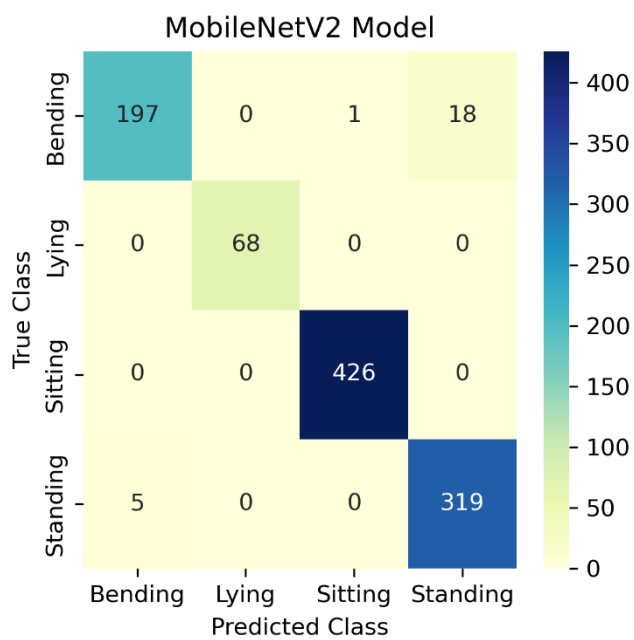
The confusion matrix in Figure 20 shows that the sitting and standing classes dominate performance, with near-perfect diagonals, while most errors result from confusion between bending and standing. VEACTION-CNN is the most reliable overall, as demonstrated by the experiment conducted in this dissertation. Only a handful of bending-to-standing (16) and standing-to-bending (6) were misclassified, with lying perfectly recognised (68/68). The ensemble model also performed well, but slightly misclassified bending, with 21 mislabels to standing and a few sitting-to-bending errors. Baseline models such as MobileNetV2, CNN, and VGG16 showed similar performance, with minor bending-to-standing and standing-to-sitting mislabeling, whereas Xception showed a small increase in standing-to-sitting mislabeling. Temporal models, such as LSTMs and GRUs, introduced additional confusion between sitting and standing. For instance, the LSTM mislabeled 18 sitting instances as standing and 21 standing instances as bending, whereas the LitePose-GRU model performed well on sitting and standing but mislabeled more bending postures as standing (40). In conclusion, combining spatial cues such as VEACTION-CNN with the Ensemble model yields the most balanced and high-precision results, whereas temporal models require stronger motion cues to distinguish bending from standing.



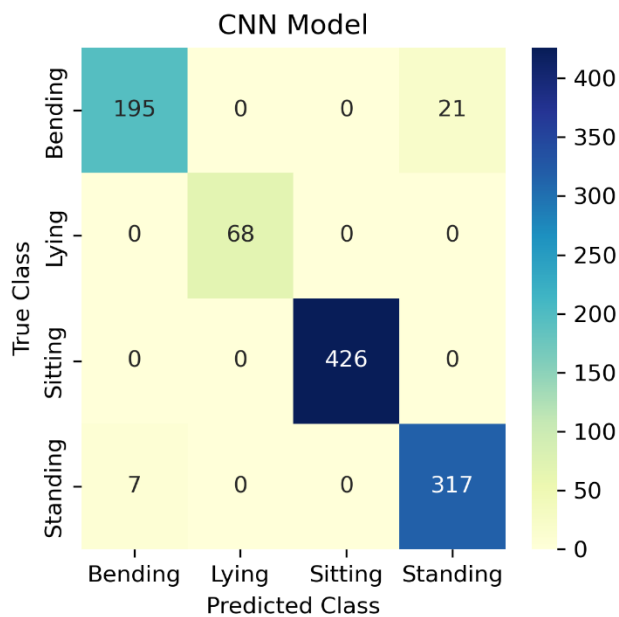
(i)



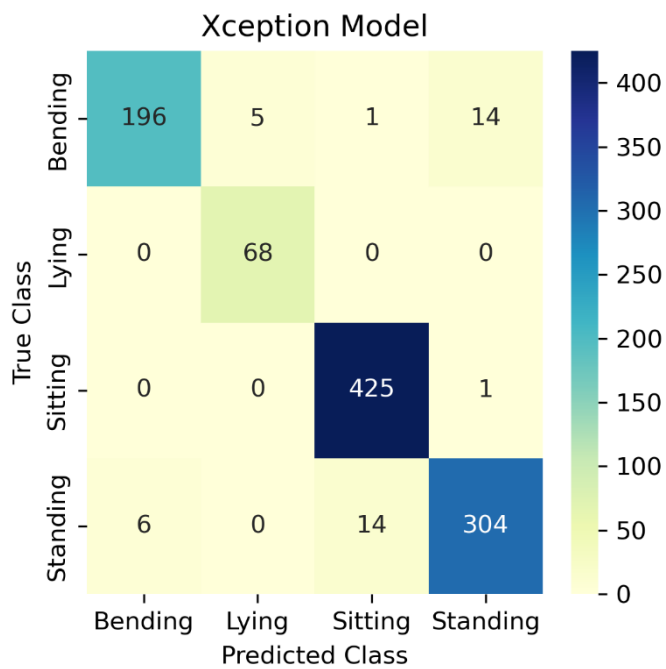
(ii)



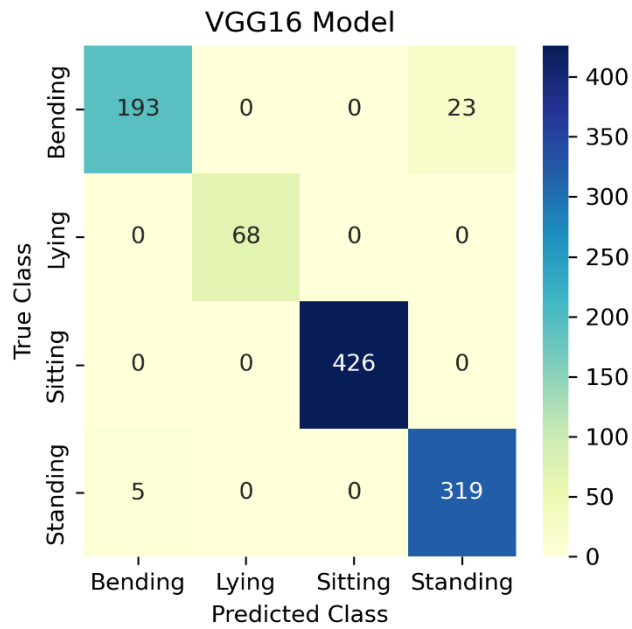
(iii)



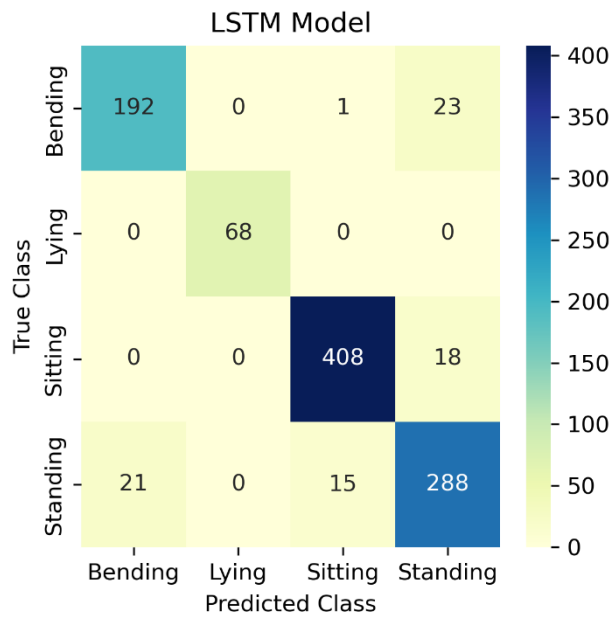
(iv)



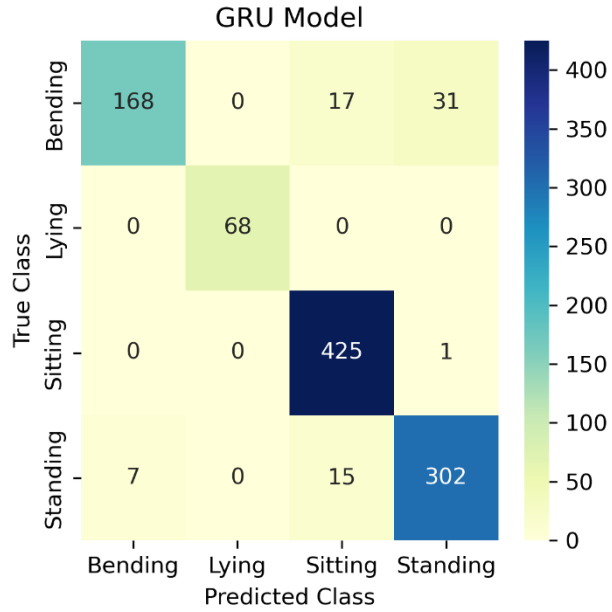
(v)



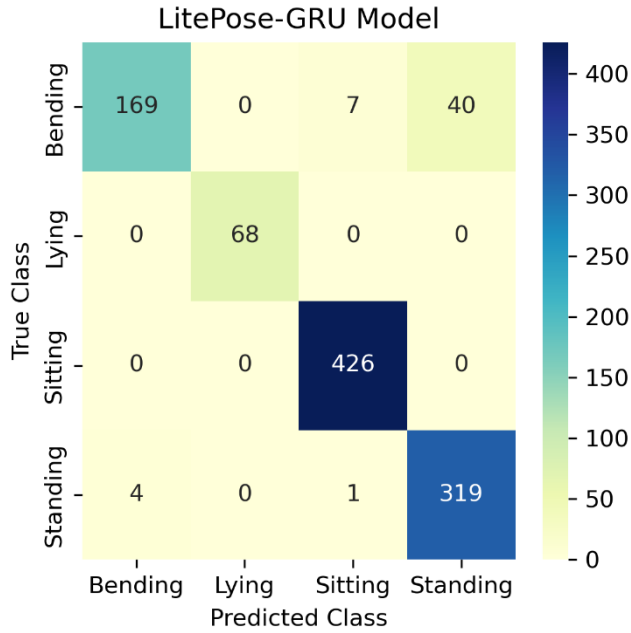
(vi)



(vii)



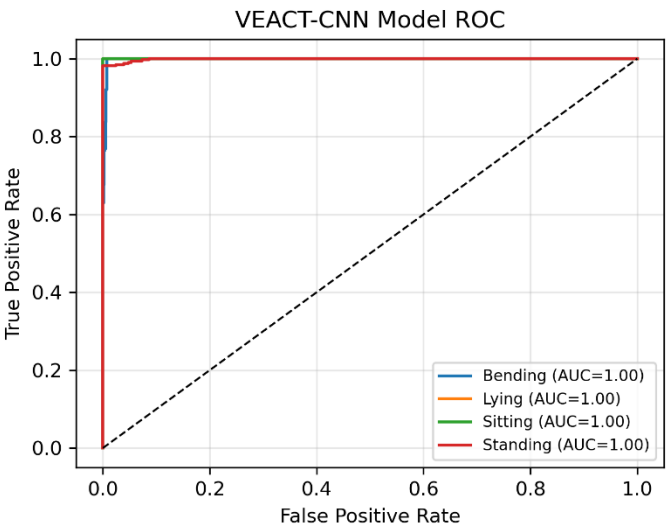
(viii)



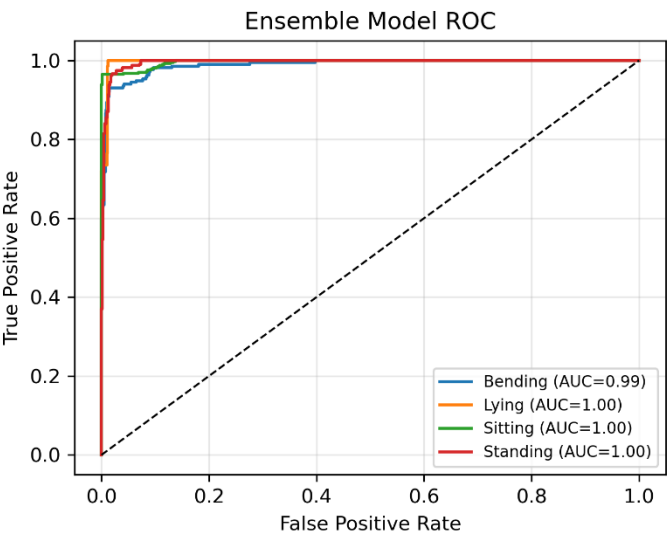
(ix)

Fig. 20. Confusion Matrix of DTL models ETRI-Activity3D-LivingLab Dataset: (i) VEACTION-CNN, (ii) Ensemble, (iii) MobileNetV2, (iv) CNN, (v) Xception, (vi) VGG16, (vii) LSTM, (viii) GRU, and (ix) LitePose-GRU

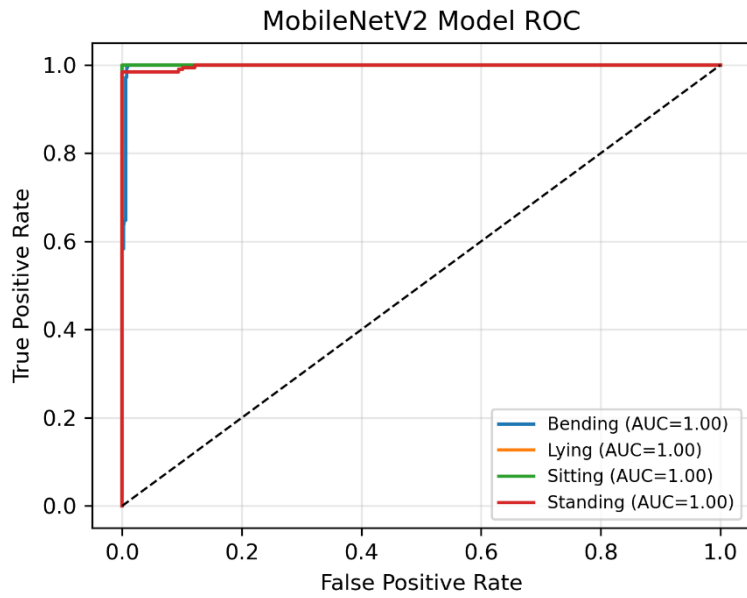
All the models' ROC curves are very close to the top-left corner, as shown in Figure 21, indicating excellent separability among the four postures. VGG16, Xception, MobileNetV2, VEACTION, and ensemble models achieve a class AUC of 1.00, indicating that the features they learn are distinctive. The temporal models LSTM, GRU, and LitePose-GRU also performed very well but showed the smallest margins, with most classes at 0.99 to 1.00, occasionally dropping to 0.98 to 0.99, suggesting that temporal noise or class similarity affects these models.



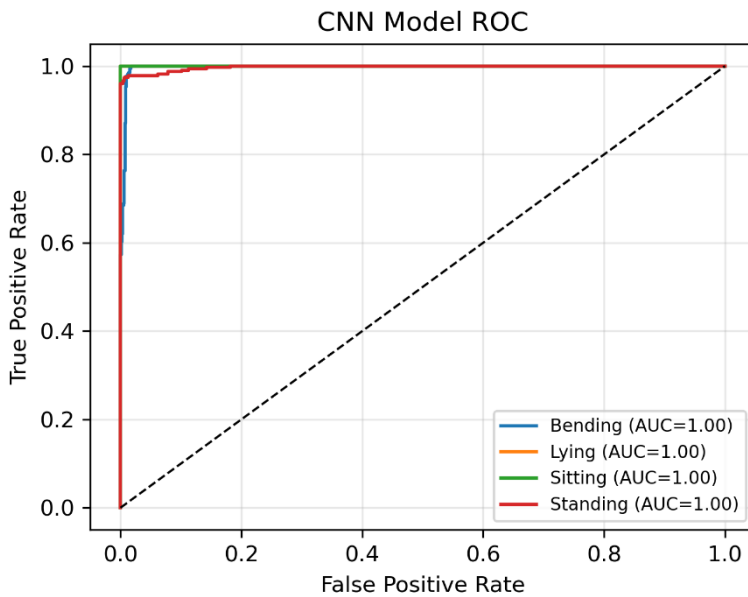
(i)



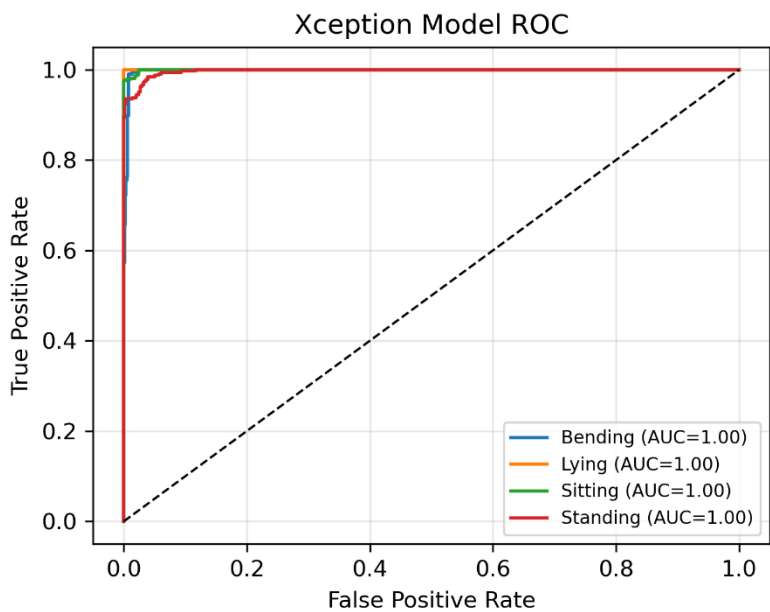
(ii)



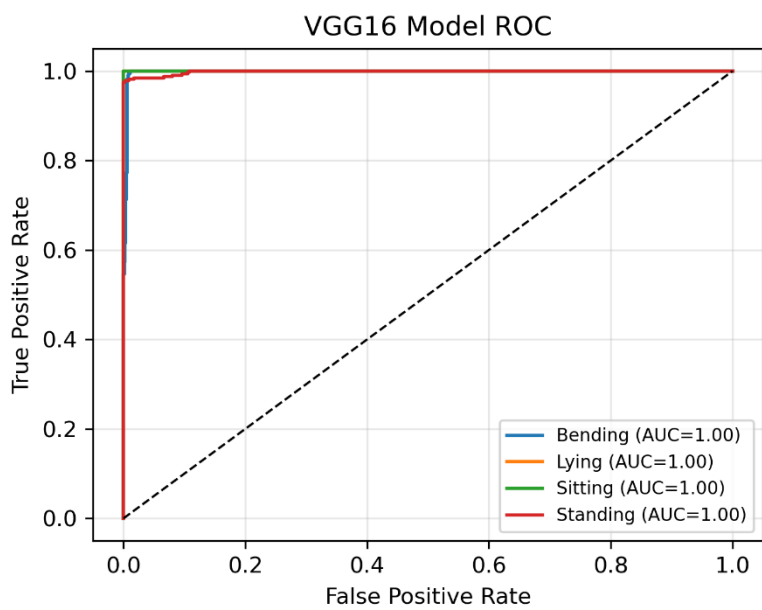
(iii)



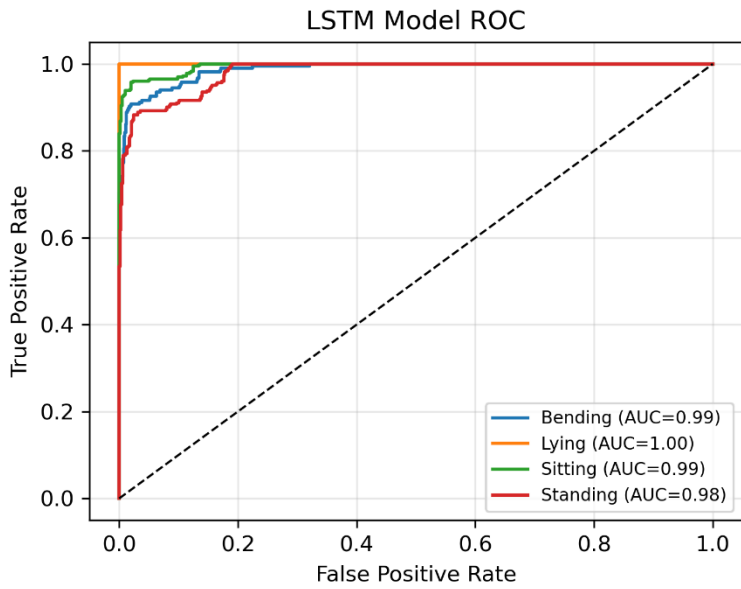
(iv)



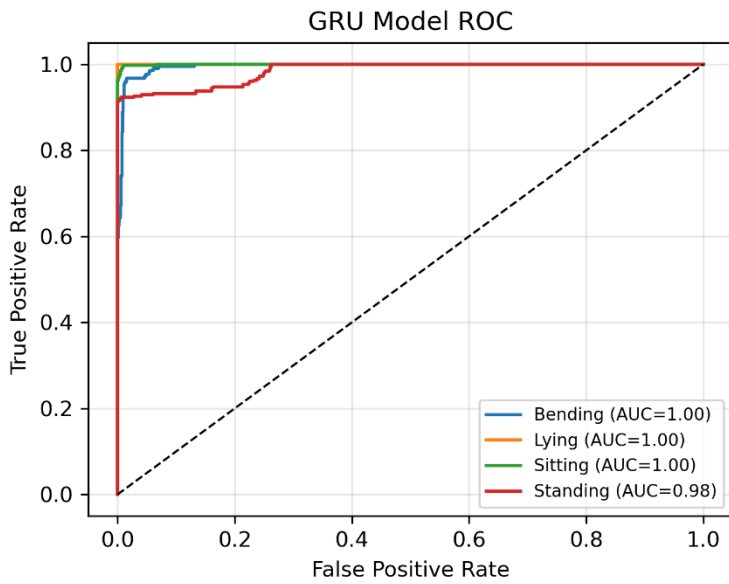
(v)



(vi)



(vii)



(viii)

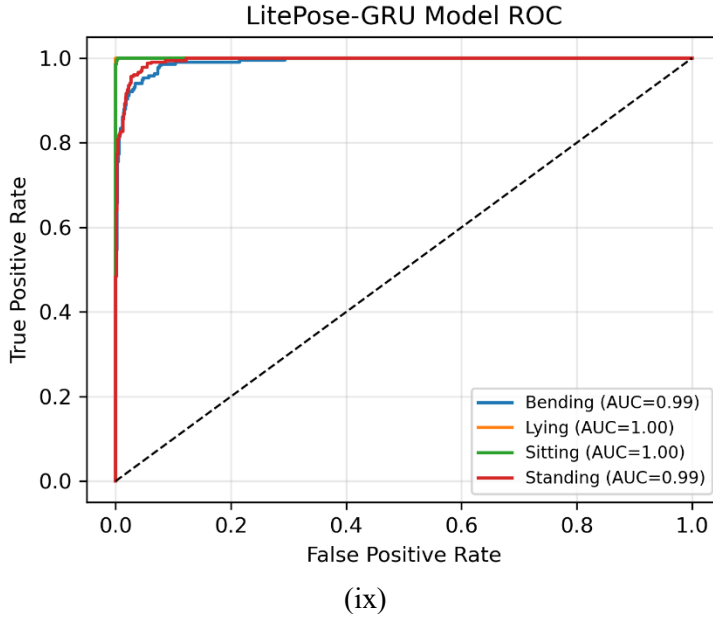
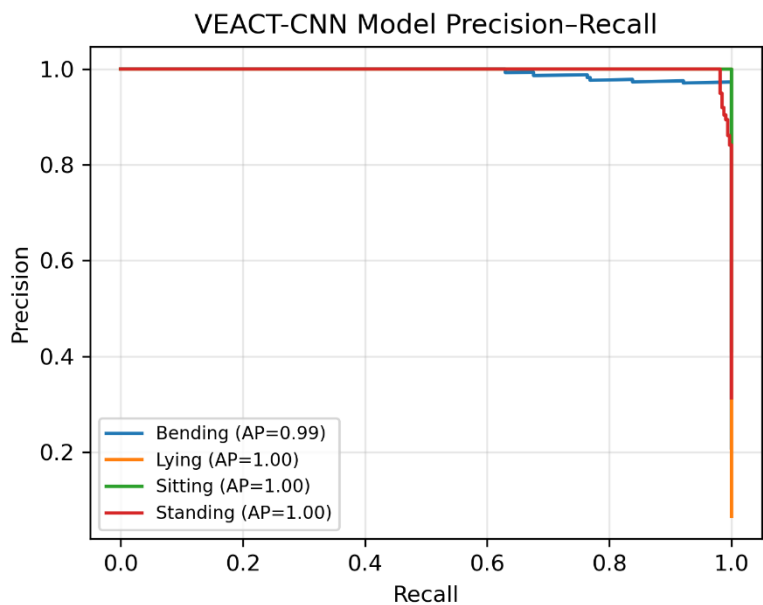
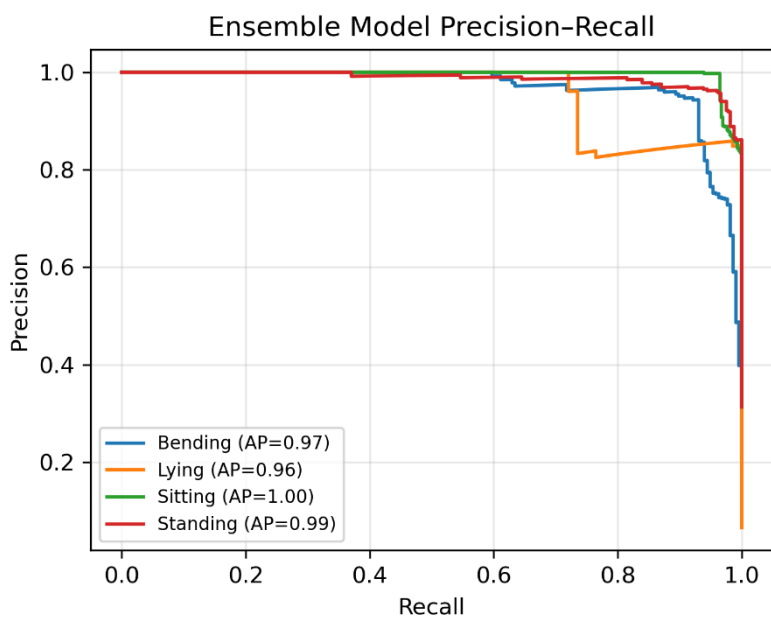


Fig. 21. ROC Curves of DTL models ETRI-Activity3D-LivingLab Dataset: (i) VECT-CNN, (ii) Ensemble, (iii) MobileNetV2, (iv) CNN, (v) Xception, (vi) VGG16, (vii) LSTM, (viii) GRU, and (ix) LitePose-GRU

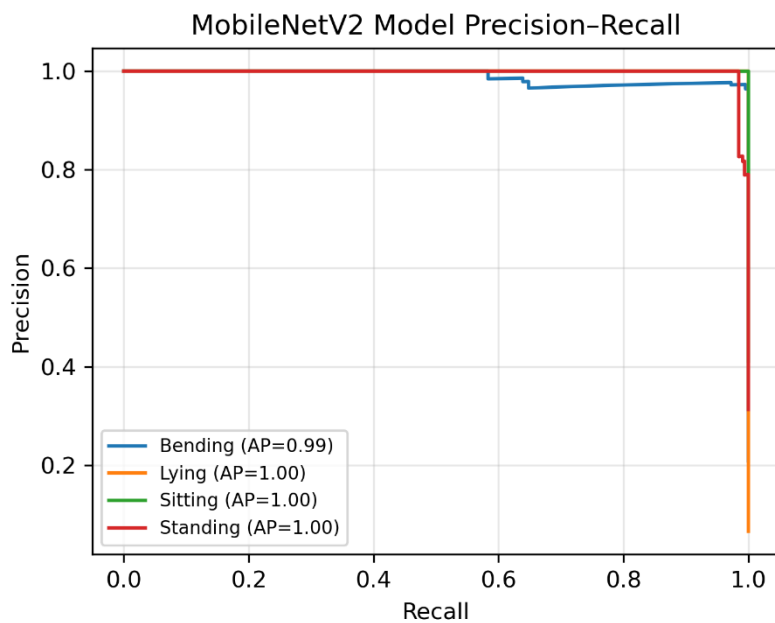
The PR curves, as shown in Figure 22, consistently demonstrate robust behaviour in most classes, including lying, sitting, and standing, achieving an AP of 1.00 for CNN backbones such as VECT-CNN, MobileNetV2, Xception, and VGG16, indicating near-perfect precision even at high recall. The main misclassification was in the bending class, where the AP is slightly lower (0.97-0.99), and the curves drop off earliest, suggesting greater overlap with other postures. The temporal models LSTM, GRU, and proposed hybrid LitePose-GRU remain competitive but show high recall, especially for bending and standing, with AP of 0.97 to 0.99. The ensemble model improves overall robustness but reduces the AP score for the lying class (0.96), a common side effect of averaging logits when one class is misclassified.



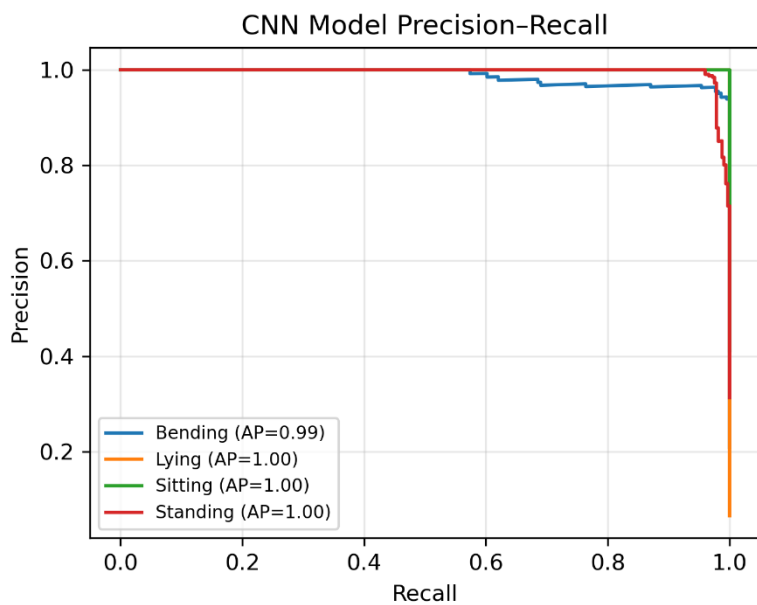
(i)



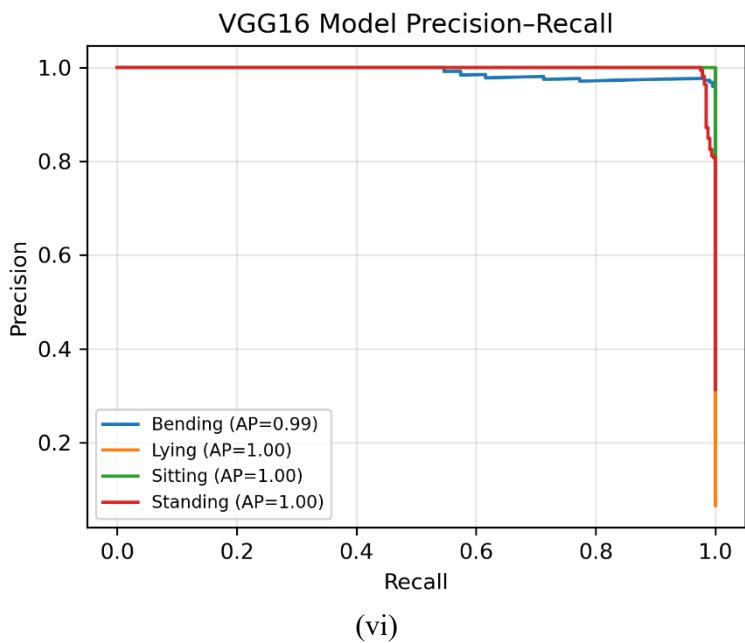
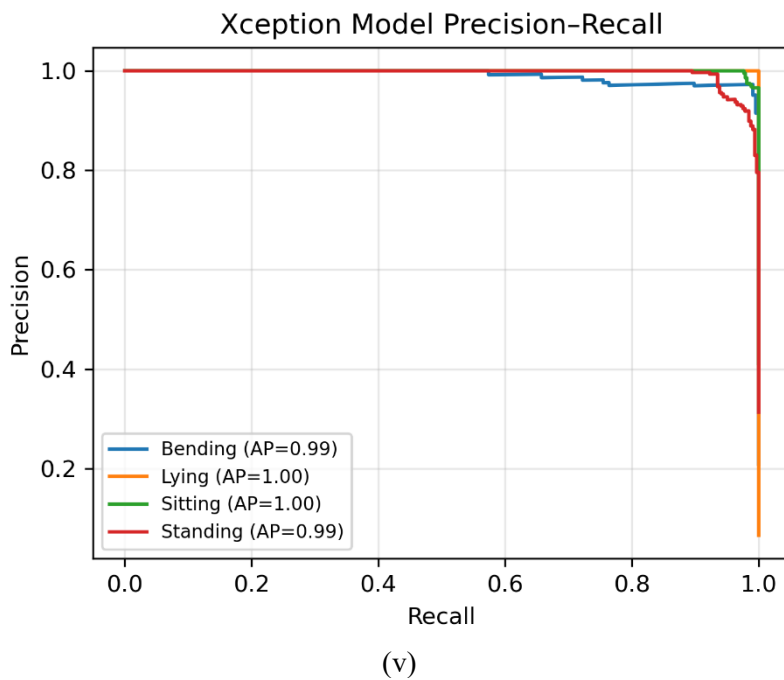
(ii)

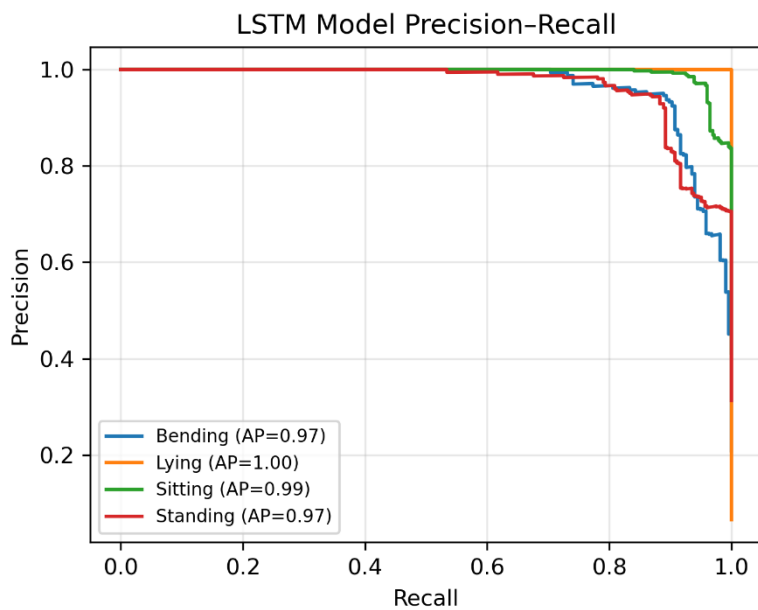


(iii)

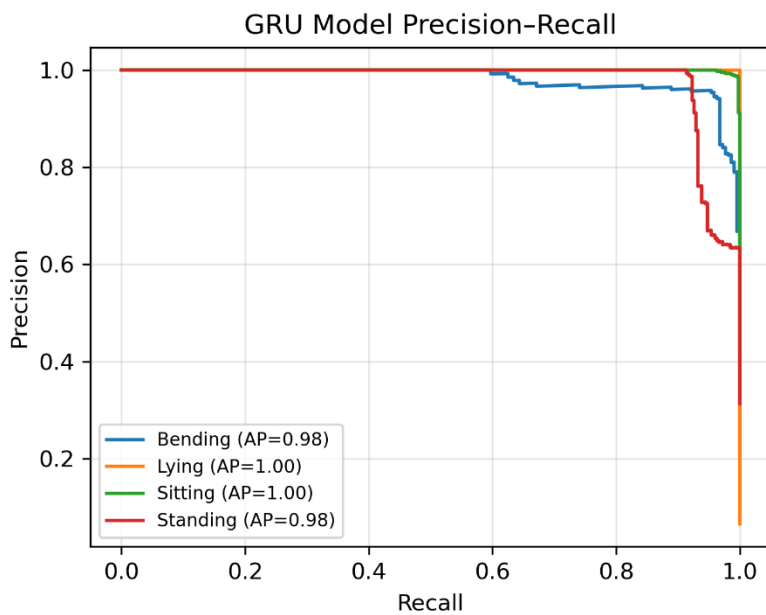


(iv)





(vii)



(viii)

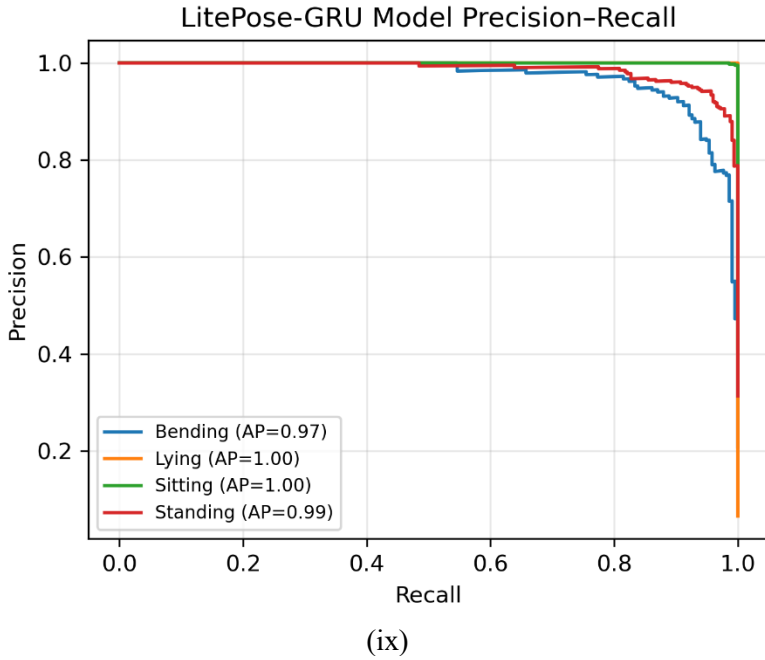


Fig. 22. PR Curves of DTL models ETRI-Activity3D-LivingLab Dataset: (i) VECT-CNN, (ii) Ensemble, (iii) MobileNetV2, (iv) CNN, (v) Xception, (vi) VGG16, (vii) LSTM, (viii) GRU, and (ix) LitePose-GRU

3.4.3. Results from Training on Training and Validation Datasets

The results highlight differences in how the models generalise across the training and validation sets, as shown in Table 6. In the dissertation experiment, the proposed VECT-CNN, the baseline MobileNetV2, and VGG16 achieved the best overall performance, with test accuracies above 97%, with MobileNetV2 slightly leading at 97.66%. It was also noticed that VECT-CNN achieved the lowest training loss of 0.0175.

The ensemble model produced a distinctive pattern: despite its modest training accuracy of 78.12%, it delivered a strong test performance of 95.31%, highlighting the stabilising effect of combining multiple learners. However, this comes at the cost of leaving some peak accuracy unexploited compared to single, powerful CNNs.

Two temporal models, namely LSTM and GRU, were also employed. When trained on the dataset, their performance did not match that of the CNN-based models; however, they achieved accuracies of 92.46% and 93.13%, respectively. This demonstrates that spatial features dominate in posture detection within this dataset. The proposed LitePose-GRU achieved an excellent validation accuracy of 98.63% but dropped to 94.97% on the test set, suggesting susceptibility to mild overfitting.

In conclusion, CNN-based architectures such as MobileNetV2 and VGG16 achieved the best balance between accuracy and loss, making them more suitable for real-world deployment. At the same time, the ensemble's stability and LitePose-GRU's high validation performance demonstrate useful trade-offs, depending on whether robustness or efficiency is prioritized.

Table 6. Accuracy and Loss of Training, Validation and Testing on ETRI-Activity3D-LivingLab Dataset

Model	Training Accuracy (%)	Validation Accuracy (%)	Testing Accuracy (%)	Training Loss	Validation Loss	Testing Loss
VEACT-CNN	99.17	96.64	97.56	0.0175	0.1333	0.0854
Ensemble Model	78.12	92.59	95.31	0.3995	0.2080	0.1869
MobileNetV2	100	98.84	97.66	0.0388	0.0551	0.0668
CNN	96.88	97.57	97.27	0.0337	0.0681	0.0677
Xception	96.88	95.02	96.00	0.1665	0.1208	0.0871
VGG16	99.04	96.22	97.29	0.0206	0.0840	0.0625
LSTM	91.35	94.50	92.46	0.2302	0.2454	0.2423
GRU	86.17	90.03	93.13	0.3809	0.2356	0.2378
LitePose-GRU	98.26	98.63	94.97	0.0712	0.0727	0.1630

The results in Figures 19-22 demonstrate that the CNN-based models generally performed best in terms of accuracy and stability on both the training and validation datasets. For instance, MobileNetV2 achieved the highest test accuracy of 97.66%, with a very low loss of 0.0668, closely followed by the proposed lightweight VEACT-CNN, which achieved 97.56% accuracy and a loss of 0.0854, and VGG16, which achieved 97.29% accuracy and a loss of 0.0625. These models showed only small differences between training, validation, and testing accuracy, indicating they generalised well without overfitting. The temporal recurrent models, such as LSTM and GRU, performed worse: LSTM achieved 92.56% test accuracy and a higher test loss of 0.2423, while GRU achieved 93.13% test accuracy and a lower test loss of 0.2378. These higher losses suggest that they were less effective at capturing the key features needed for posture detection. The proposed LitePose-GRU achieved 94.07% accuracy but still showed a higher test loss (0.1630) than the validation loss.

The radar charts (Figures 21 and 22) clearly indicate that the CNN-based models are tightly clustered, with consistently high accuracies, whereas the recurrent models exhibit greater variation. Overall, it was confirmed that CNN-based models such as MobileNetV2, VEACT-CNN, and VGG16 not only achieved very high accuracy above 97% but also maintained low error rates, making them more reliable for real-world posture detection. Meanwhile, though lighter, recurrent models sacrifice some performance and stability.

In conclusion, the results demonstrate that CNN-based models are the most reliable choice for posture detection tasks. These models, MobileNetV2, VEGACT-CNN, and VGG16, stood out by consistently achieving over 97% accuracy and test losses of less than 0.085, demonstrating both robust performance and good generalisation. These qualities make them well-suited to real-world deployment, where accuracy and stability are important. On the other hand, recurrent models such as LSTM and GRU are lighter, achieving around 92-93% accuracy and high test losses, making them less robust for this task. The proposed LitePose-GRU showed some improvement, but still did not match the stability of the implemented CNN-based models. Overall, the dissertation findings suggest that CNN models, particularly MobileNetV2, strike the best balance between accuracy, efficiency and reliability, making them the preferred models for practical applications.

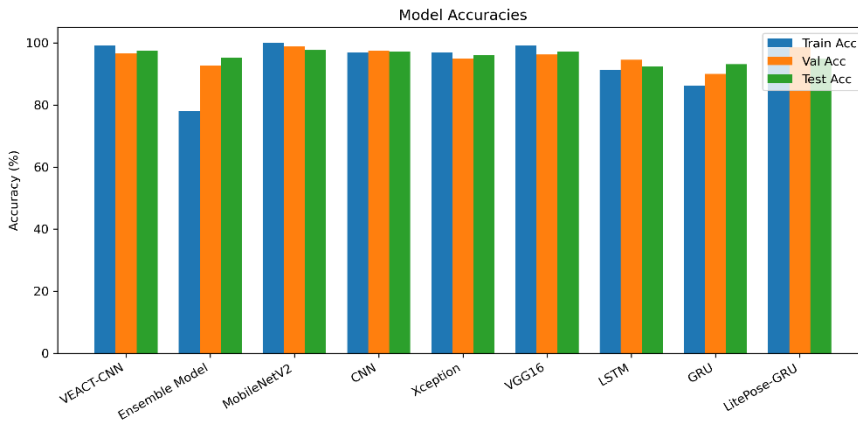


Fig. 23. Comparison of Model Accuracies

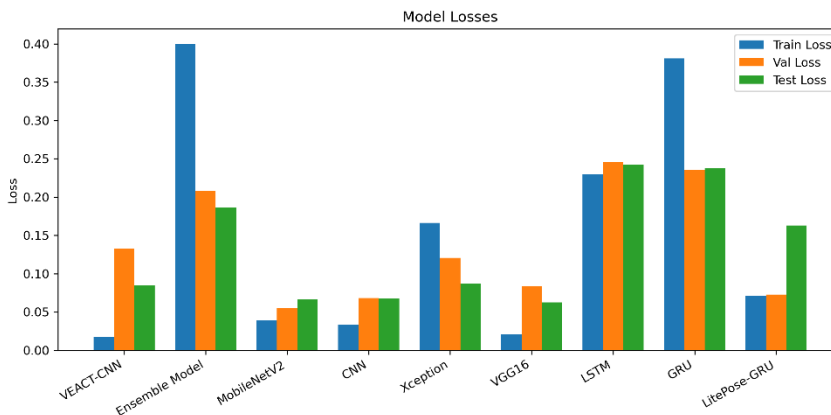


Fig. 24. Comparison of Model Losses

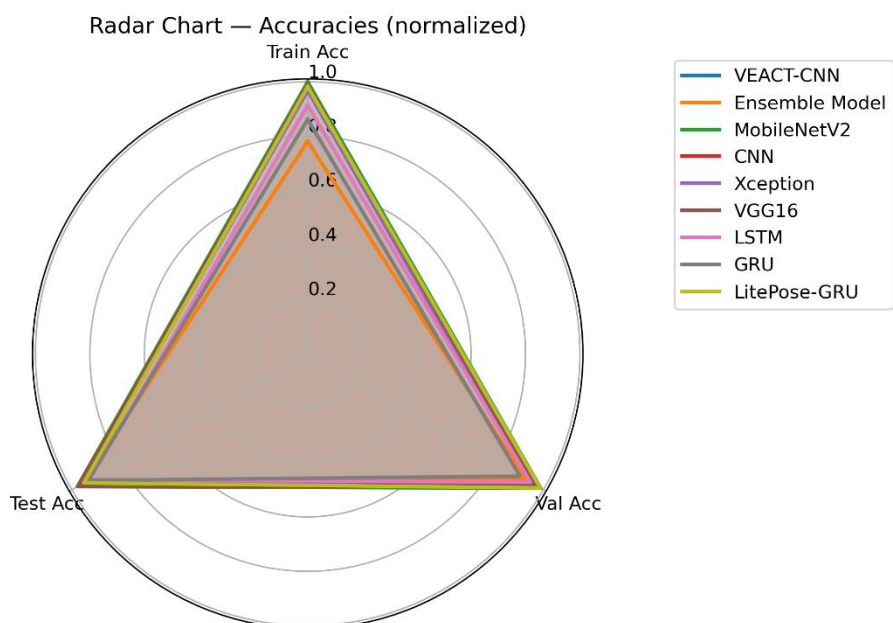


Fig. 25. Radar Plot for the Model Accuracies

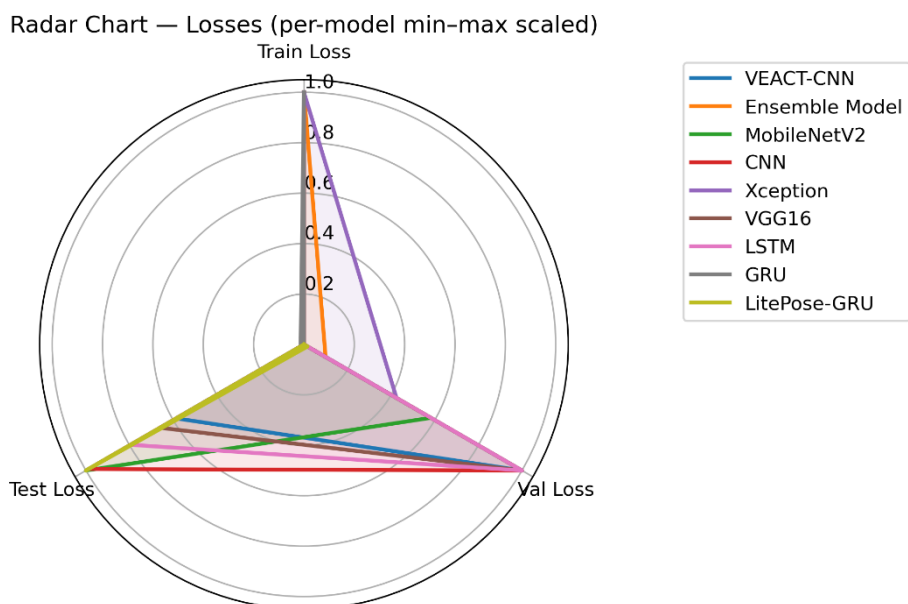


Fig. 26. Radar Plot for the Model Losses

3.4.4. Experimental Results on Test Dataset

The results from Table 7 demonstrate a clear comparison of how different models performed on the ETRI-Activity3D-LivingLab test dataset. Most models achieved high accuracy with VEACTION-CNN, MobileNetV2, VGG16, and the baseline CNN, with good results across all three models, achieving accuracy over 97% and macro F1-scores of around 97%, reflecting strong generalisation across the four postures. VEACTION-CNN stood out with an overall accuracy of 97.87% and an f1-score of 97.86%, showing stable performance across bending, lying, sitting, and standing. MobileNetV2 closely followed, with an accuracy of 97.68% and an F1-score of 97.67%, reinforcing its reliability as a lightweight yet effective model. In contrast, temporal models such as LSTM and GRU were less competitive, achieving accuracies of 92.46% and 93.13%, respectively, and lower F1 Scores, indicating weaker posture detection than CNN-based approaches.

LitePose-GRU showed a modest improvement, achieving 94.97% accuracy, but still fell short of the best CNN-based models. Interestingly, the ensemble model achieved 95.36% accuracy, but its lower F1-score of 93.89% suggests it struggled to make consistent predictions across all classes. These findings underscore the robustness of CNN-based models, particularly VEACTION-CNN and MobileNetV2, for posture detection, while highlighting the limitations of temporal recurrent models in this application.

In conclusion, the evaluation on the test dataset highlights VEACTION-CNN and MobileNetV2 as the most suitable models for deployment, given their consistent high accuracy of around 98% and macro F1 scores, making them both reliable and efficient for real-world posture detection systems.

Table 7. Results of the performance evaluation obtained from the ETRI-Activity3D-LivingLab Test Dataset

Model	Label	Precision	Recall	F1-score	Accuracy	F1-Score
VEACTION-CNN	Bending	0.97	0.93	0.95	97.87	97.86
	Lying	1.00	1.00	1.00		
	Sitting	1.00	1.00	1.00		
	Standing	0.95	0.98	0.97		
Ensemble Model	Bending	0.96	0.88	0.92	95.36	93.89
	Lying	0.82	1.00	0.90		
	Sitting	0.99	0.96	0.98		
	Standing	0.94	0.98	0.96		
MobileNetV2	Bending	0.98	0.91	0.94	97.68	97.67
	Lying	1.00	1.00	1.00		
	Sitting	1.00	1.00	1.00		
	Standing	0.95	0.98	0.97		
CNN Model	Bending	0.97	0.90	0.93	97.29	97.27

	Lying	1.00	1.00	1.00		
	Sitting	1.00	1.00	1.00		
	Standing	0.74	0.98	0.96		
Xception	Bending	0.97	0.91	0.94	96.03	95.74
	Lying	0.93	1.00	0.96		
	Sitting	0.97	1.00	0.98		
	Standing	0.95	0.94	0.95		
VGG16	Bending	0.97	0.89	0.93	97.29	97.26
	Lying	1.00	1.00	1.00		
	Sitting	1.00	1.00	1.00		
	Standing	0.93	0.98	0.96		
LSTM	Bending	0.90	0.89	0.90	92.46	93.43
	Lying	1.00	1.00	1.00		
	Sitting	0.96	0.96	0.96		
	Standing	0.87	0.89	0.88		
GRU	Bending	0.96	0.78	0.86	93.13	93.50
	Lying	1.00	1.00	1.00		
	Sitting	0.93	1.00	0.96		
	Standing	0.90	0.93	0.92		
LitePose-GRU	Bending	0.98	0.78	0.86	94.97	94.84
	Lying	1.00	1.00	1.00		
	Sitting	0.98	1.00	0.99		
	Standing	0.89	0.98	0.93		

3.4.5. Computational Resource Evaluation

A clear trade-off between model size and training time is evident in Table 8, with most models being lightweight and suitable for edge deployment. The models with the most parameters are Xception, VEGACT-CNN, and VGG16. Xception had 7.85M parameters and a model size of 29.95 MB; VEGACT-CNN had 0.267M and 1.02MB; and the third model, VGG16, had 7.21M parameters and a model size of 27.5 MB. All three models were trained in under an hour, with durations of 0.70, 0.42, and 0.56 hours, respectively, suggesting good hardware utilisation and stable optimisation. Among the models, MobileNetV2 has a parameter count of 1.54M and a model size of 5.86MB, which delivers an attractive balance, and trains in 0.86 hours while keeping memory demands very low. Recurrent baselines such as GRU (0.60M, 2.30MB) and LSTM (0.79M, 3.02MB) are considered the smallest in terms of parameters and size, but they still took about 1.10 to 1.19 hours to train, reflecting the sequential computation overhead. During the experiment, two notable outliers were observed. The Ensemble model, despite being lightweight (0.63M parameters and 2.39MB in size), took 1.26 hours to train.

The baseline CNN is a mid-size model with 5.70M parameters and 21.74MB, but was the slowest to train, taking 2.89 hours, possibly due to its depth or less-optimised layers. In practice, MobileNetV2 is the most efficient choice for constrained devices, while VEGACT-CNN, VGG16, and Xception remain deployable when

slightly higher capacity is desired without incurring prohibitive training costs, as all models are below 30MB in size.

Table 8. Computational Resources Evaluation of the Proposed Models

Model	Parameter (M)	Size (MB)	Training Time (hrs)
VEACT-CNN	0.267	1.02	0.422
Ensemble Model	0.625	2.39	1.262
MobileNetV2	1.535	5.86	0.863
CNN	5.698	21.74	2.893
Xception	7.851	29.95	0.696
VGG16	7.211	27.51	0.561
LSTM	0.791	3.02	1.188
GRU	0.603	2.30	1.103
LitePose-GRU	1.241	4.73	0.682

3.4.6. Comparative Insights

VEACT-CNN demonstrates excellent training performance but suffers from overfitting, as evidenced by the sharp drop from 99.27% training accuracy to 94.89% test accuracy. It is computationally heavy with the largest parameter count and longest training time. D-MIXNet achieves slightly lower training accuracy but clearly outperforms on validation and test data. Its 98.91% test accuracy is the best result, and it is far more efficient in terms of parameter count and training time, about one-third of VEACT-CNN's. This balance of accuracy and efficiency makes it well-suited to practical, real-time applications in a living lab environment.

Overall, the ETRI-Activity3D-LivingLab experiments demonstrate that while VEACT-CNN achieves near-perfect training performance, its generalisation and efficiency are inferior to D-MIXNet. The proposed D-MIXNet not only surpasses VEACT-CNN in test accuracy by nearly 4% but also significantly reduces computational requirements, making it both effective and lightweight. These findings validate D-MIXNet as a robust model for human posture detection in resource-constrained real-world settings.

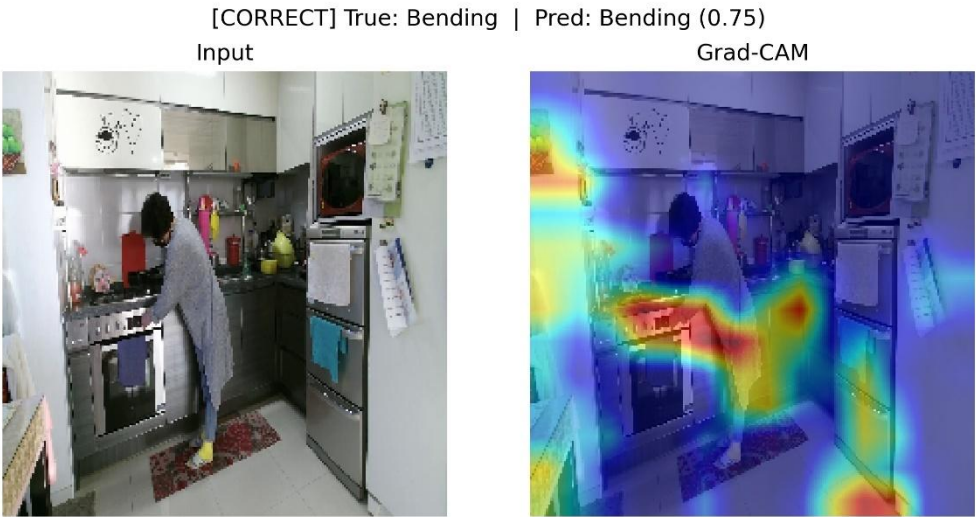
3.5. Explainability of the Proposed Lightweight VEACT-CNN Model on ETRI-Activity3D-LivingLab Dataset

The interpretability of the proposed lightweight VEACT-CNN model was evaluated for posture classification. In this dissertation, the test dataset was analyzed

using the Grad-CAM explainability model. Figure 23 presents a side-by-side comparison of the original input image and the corresponding Grad-CAM heatmap overlay for each randomly selected class. In correctly classified classes, Grad-CAM activations were concentrated on body regions relevant to each posture. For instance, in the bending class, the highlighted regions were primarily around the torso angle; in the sitting class, attention focused on the lower body and seating area; in the standing class, activations were distributed across the full vertical silhouette. These visualisations demonstrate that the model leverages semantically meaningful cues when classifying postures, thereby improving confidence in its predictions and detection.

Conversely, the misclassified lying posture showed activation patterns spread across the surrounding environment, such as the floor and background objects, rather than focusing on the subject’s posture. This suggests that the model is susceptible to contextual bias, where non-salient background features can influence decision-making. Recognizing such cases is important, as it helps guide future work toward improving model robustness through enhanced image augmentation strategies, background noise reduction, or more effective attention regulation mechanisms.

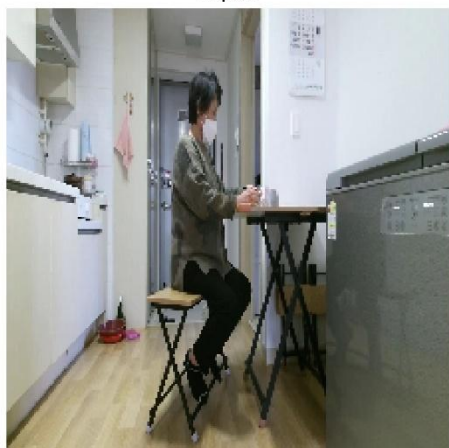
Overall, these Grad-CAMs provide an interpretable lens into the model’s decision-making process. By confirming correct reasoning in most posture-class cases and exposing failure modes in challenging scenarios, they serve as both a validation tool and a diagnostic aid to guide further improvements.



(i)

[CORRECT] True: Sitting | Pred: Sitting (0.76)

Input



Grad-CAM



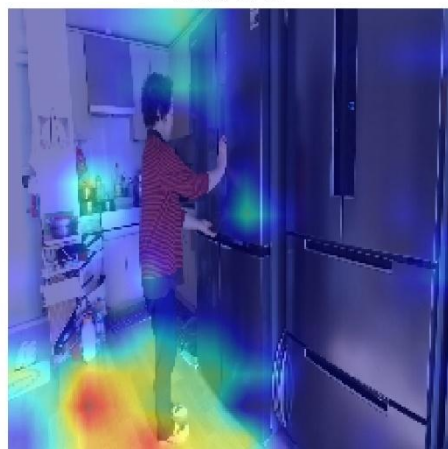
(ii)

[CORRECT] True: Standing | Pred: Standing (0.99)

Input



Grad-CAM



(iii)

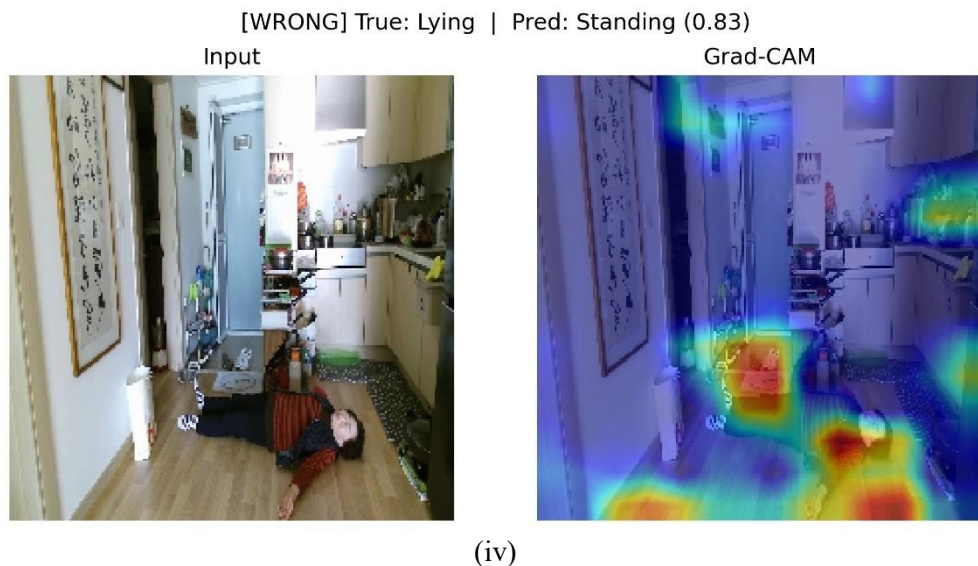


Fig. 27. Explanability with Grad-CAM on ETRI-Activity3D-LivingLab Dataset

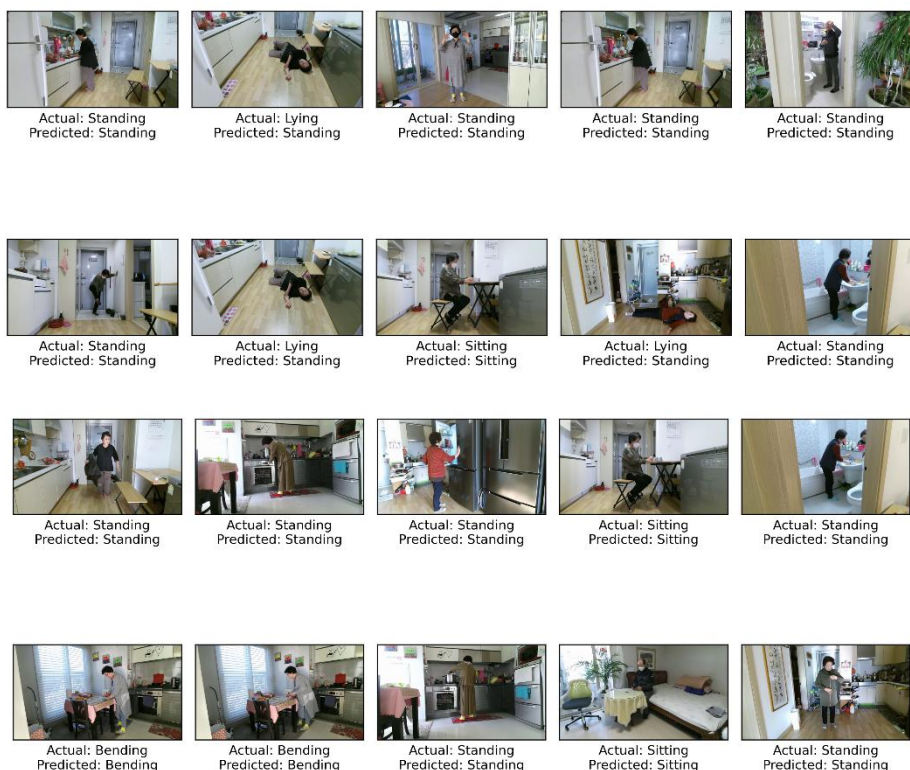


Fig. 28. Prediction Examples on ETRI-Activity3D-LivingLab Dataset Test Dataset

The prediction sample in Figure 24 complements the Grad-CAM analysis by showing how the model behaves on real test images. As shown in the figure, upright postures, such as standing and sitting, consistently yielded correct predictions, reflecting their ability to capture informative body cues and align them with the semantic labels. These correct classifications also match the Grad-CAM overlays, which highlighted the torso, legs and seated regions as salient areas of attention. Meanwhile, the lying postures were repeatedly misclassified as standing. This confirms the earlier observation that the model tends to focus on background and contextual features rather than on the body's horizontal alignment. The misclassification of lying represents a systematic weakness rather than random noise, suggesting the need for additional training strategies such as targeted augmentation, class rebalancing, or attention regularisation. The proposed model is accurate and interpretable for upright poses but struggles with horizontal body postures, a limitation that must be addressed for deployment in practical posture detection systems.

3.6. Ablation Study

Ablation was conducted in this dissertation to understand the contribution of each component of all proposed models used for experiments. This involves altering one model element at a time while observing its impact on performance. The ablation experiments were designed to answer the following question: How much does FS assist the proposed hybrid model? Does the hybrid CNN approach perform better than using the baseline CNN models? Does the attention mechanism improve performance accuracy? What is the effect of using an SVM classifier with a standard neural network classifier?

Ablation tests were performed on the test dataset to determine the impact of each proposed model on the posture classification implementation pipeline. The outcome of the ablation experiments is summarised as follows: note that all results refer to the validation set used for consistent comparison.

1. **Baseline CNN vs Hybrid DeneSVM:** The fine-tuned baseline DenseNet121 with a softmax achieved an accuracy of 91.60% on the test dataset. The hybrid DeneSVM achieved a test accuracy of 94.38%, representing a substantial improvement. The 2.78% increment shows the benefit of replacing the standard neural classifier with an SVM model. The hybrid model also achieved higher precision and recall across all classes, with an AUC of 96.23%. This result is consistent with the expectation that an SVM classifier can achieve improved classification performance when applied to deep feature representations.

2. **Without LASSO vs With LASSO FS:** The proposed LASSO+InceptionV3+SVM accuracy value dropped from 93.13% to 91.46% when LASSO was not used for the implementation of the model. The model without LASSO exhibited overfitting, with very high training accuracy and low validation accuracy compared to the full model. When LASSO was used, the feature dimensionality was reduced, and it was discovered that the SVM generalised better, yielding an accuracy

of 93.13% on the test dataset. This result supports the implementation of LASSO-based feature selection as an effective regularization strategy and as a means of improving overall performance in this context.

3. **Pre-trained Models vs Fused Pre-trained CNN (MobileNetV2 + Xception = Lightweight OptiMobileX Model):** When the two models were implemented individually, the accuracy values were low. In the experiment conducted in the dissertation, MobileNetV2 achieved 92.71% accuracy, and Xception achieved about 92.08% on the test set. The proposed OptiMobileX fused model achieved approximately 95.42% accuracy, outperforming the individual models. This demonstrates an additive effect: combining the lightweight MobileNetV2 features with Xception's deep features yields a better feature set. The ablation study further demonstrated that using only a subset of the features to construct the fused vector results in a corresponding decline in performance. Overall, the fusion of the two models is justified by a 2.71% increase in accuracy.

4. **Without Attention Mechanism vs With Attention Mechanism (VGG16 Model):** The VGG16 model without the VECT attention model had a lower accuracy. For instance, its overall accuracy was 89%, whereas the proposed VECT-CNN with attention achieved 94.89% accuracy under the same training conditions. In one ablation test, when classifying lying and sitting postures, the no-attention model's recall for lying was very low; it often missed some lying postures. In contrast, the model with attention correctly classified nearly all postures, improving recall to 96%. This indicates that the attention mechanism helps the model focus on the distinctive horizontal body shape in the lying posture. These findings confirmed that the attention component not only increases the model's accuracy but also helps balance its performance across different postures.

The ablation results above highlight that the full proposed models outperform their ablated variants. The removal of each component resulted in a measurable decline in one or more performance metrics, thereby reinforcing the validity of the design choices.

In summary, the ablation study confirmed that each component makes a meaningful contribution to the system. Feature selection improved the SVM classifier's effectiveness. The hybrid CNN+SVM approach outperformed a purely neural network approach. Combining two different CNN architectures yielded higher accuracy than either one alone. The attention mechanism in VGG16 notably enhanced performance and interpretability. Removal of these components resulted in observable performance degradation, thereby confirming their necessity within the overall model architecture.

3.7. Statistical Results

On the Silhouettes dataset, all deep models achieved accuracies over 96%. For instance, InceptionV3 had a test accuracy of 96.4%, Xception models had 96.5%, and MobileNetV2 had 96.0%. By Contrast, the DeneSVM model had much lower accuracy at 85.2%. Bootstrap resampling was applied to estimate confidence intervals

(CIs) for accuracy. InceptionV3 and MobileNetV2 both achieved a mean accuracy of approximately 0.9603 with a standard deviation of $\approx$ approximately 0.0088 (95% CI $\approx$: 0.942–0.977), Xception achieved a mean accuracy of approximately 0.9641 with a standard deviation of approximately 0.0087 (CI $\approx$: 0.946–0.979), and OptiMobileX achieved a mean accuracy of approximately 0.9539 with a standard deviation of approximately 0.0094 (CI $\approx$: 0.933–0.971), as seen in Table 8. These CIs overlap substantially (one-sided $p \approx 0.25$ for all comparisons), indicating no model is statistically superior. In other words, all four DTL models perform comparably well, that is, around 96% accuracy on silhouettes.

In terms of computation resources, the large models are much heavier. For instance, InceptionV3 and Xception each have parameter counts of 24–25 million, with model sizes $\approx$ of 92–96 MB, and their training times were the longest ($\approx$ 4.5 hours for InceptionV3 and $\approx$ 14.7 hours for Xception). The MobileNetV2 and the proposed OptiMobileX are extremely lightweight, with $\approx$ 2.23M parameters and $\approx$ 8.7 MB, respectively. Their training times were also short, with $\approx$ 1.2 to 2.7 hours. Inference throughout (FLOP/s) favours the large models, such as InceptionV3, achieving $\approx$ 0.515TFLOP/s (inference) while MobileNetV2 had only 0.049 TFLOP/s. Hence, despite similar accuracy, the lightweight models (MobileNetV2 and OptiMobileX) use orders of magnitude fewer resources. Overall, on the Silhouettes dataset, a clear trade-off is observed: the tiny hybrid models achieve accuracy comparable to larger models while maintaining substantially lower parameter counts and model sizes.

On the MPII dataset (Table 8), the analysis focuses on raw performance and resource utilization, and no bootstrapping was performed. Here, the trends are similar. AlexNet is very large, with $\approx$ 62.3M parameters and 169MB, but InceptionV3, with $\approx$ 23.9M parameters and 90–92MB, and Xception, with $\approx$ 22.9M parameters and 88MB, are moderate. The proposed hybrid, OptiMobileX, has a parameter size of $\approx$ 9.65M and a size of 36.8MB. The training time ranges from $\approx$ 0.11 h for InceptionV3 to $\approx$ 1.02 h for the OptiMobileX model. Inference throughput was highest for InceptionV3 ($\approx$ 0.515 TFLOP/s), and MobileNetV2 had the lowest inference throughput of 0.049 TFLOP/s. The proposed lightweight OptiMobileX model achieves 95.4% accuracy with fewer resources. Confidence intervals (CIs) were computed only for the Silhouettes dataset. For the MPII dataset, point estimates and resource metrics were reported in Tables 9–12.

Table 9. Computational Resources on the MPII Dataset

Models	Parameter (M)	Model Size (MB)	Training Time (hrs)	Effective FLOPS (Inference) (TFLOP/s)	Effective FLOPS (Training) (TFLOP/s)
AlexNet	44.398	169.356	0.1709	0.173	0.112
InceptionV3	22.328	85.175	0.1068	0.515	0.936
MobileNetV2	2.586	9.868	0.1506	0.049	0.112
Xception	21.387	81.585	0.3145	0.329	0.583
OptiMobileX	9.652	36.82	1.0226	0.193	0.347

Table 10. Statistical Results on Silhouettes Dataset

Model	Bootstrap Acc Mean	Std	Confidence Interval	P(one-sided Acc)
InceptionV3	0.9603	0.0088	(0.9417, 0.9771)	0.25
MobileNetV2	0.9603	0.0088	(0.9417, 0.9771)	0.25
Xception	0.9641	0.0087	(0.9458, 0.9792)	0.25
OptiMobileX	0.9539	0.0094	(0.9333, 0.9708)	0.25

Table 11. Computational Resources on Silhouettes Dataset

Models	Parameter (M)	Model Size (MB)	Training Time (hrs)	Estimated FLOPs per inference	Effective FLOPS (Training) (GTFLOP/s)	Effective FLOPS (Inference) (GTFLOP/s)
InceptionV3	24.355	93.23	4.46	11.422	131.026	219.334
MobileNetV2	2.229	8.74	2.70	0.599	23.661	52.163
Xception	24.929	95.70	14.68	9.949	72.288	191.507
OptiMobileX	2.229	8.73	1.21	0.599	17.968	42.338
MobileNetV2-DNN	2.883	11.23	4.17	1.128	17.320	47.150

Table 12. Computational Resources of Hybrid Models on Silhouette Dataset

Model	Parameters (Millions)	Model Size (MB)	Training Time (Hrs)
DenseNet121	7.186	27.41	1.967
ResNet50V2	9.986	38.10	0.346
LASSO+InceptionV3+SVM	21.803	$84.21 + 32.79 = 117.00$	0.051
DeneSVM	7.038	$28.28 + 14.80 = 43.08$	0.050
MobileNetV2-DNN	2.883	11.23	4.17

3.8. Threats to Validity

Despite the promising experimental results, several threats to validity were identified in this dissertation, warranting caution when interpreting the experimental findings regarding the applicability of the proposed models. In the dissertation, the models were trained and evaluated on three datasets: the Silhouettes of Human Posture dataset, the MPII Human Pose Dataset, and the ETRI-Activity3D-LivingLab Dataset.

The first experiment was conducted on the Silhouette Human Posture dataset, which consists of four posture classes, each with 1,200 images. A key threat is that this dataset may not fully reflect real-world scenarios. Although the datasets are clean and segmented, in a practical setting, their background can be cluttered with people wearing different clothes. Accordingly, deployment in environments characterized by different camera angles, lighting conditions, or unseen posture variations may result in performance degradation. In short, the external validity is tied to how well the model covers the operational scenario. To solve this issue, one should retrain or fine-tune the model on data from the target domain, such as RGB images in a hospital room. Another threat is how the system would handle unseen postures, as this dissertation focused only on a dataset with four distinct postures: standing, sitting, bending, and lying. Human posture is not binary; there are transitional poses, such as halfway between sitting and standing, and activities such as kneeling and crawling, that were not included in this experiment. The model might classify these inputs, or it might have low confidence in them. This limits the conclusions about the specific posture classes studied. It is recommended that the model should not be assumed to generalize without the inclusion of additional training data.

Various techniques were introduced to avoid overfitting, such as image augmentation, regularisation, and early stopping. There are always situations where overfitting still occurs, especially when experimenting with different models and selecting the best-performing one. There is a validity threat due to selection bias. In this dissertation, various models, including InceptionV3, DenseNet121, ResNet50V2, and MobileNetV2, were compared, with primary emphasis on the model achieving the highest accuracy. It is possible to achieve some improvement by tuning the models

on the dataset used. To address this issue, one should ideally evaluate the final model on a completely independent dataset that was not used in this dissertation's experiments. Without such an evaluation, there is a risk that the reported performance is somewhat optimistic for the specific dataset used.

The LASSO FS step involves a regularisation path, and choosing a lambda hyperparameter could affect reproducibility if not documented. Different choices of lambda might yield different feature subsets. This was addressed by applying cross-validation to select the lambda value that maximized validation accuracy, releasing the implementation code, and specifying the random seeds used in the procedure. Yet other researchers attempting to replicate the FS might obtain slightly different results, posing a minor threat to the consistency of the findings.

The hybrid pre-trained CNN-SVM model is uncommon in a deep learning pipeline because it is not fully end-to-end trainable. Feature extraction, feature selection, and SVM training were conducted as separate stages. This multi-step process, including potential information linkage between stages, could introduce subtle issues if not carefully managed. To mitigate this risk, strict separation of the training, validation, and test datasets was maintained throughout feature selection and SVM training. This also means that the model cannot be easily updated with new data without re-running the entire pipeline. This could be seen as a threat to the practical validity of deploying such a hybrid model in a dynamic setting where continual learning is desired. A purely end-to-end model might be more convenient for updates.

Even with the integration of an attention module and the use of Grad-CAM for model explanation, a degree of risk may still remain in the reported interpretation. Attention weights highlight significant features, but they do not guarantee understanding, which can lead to misinterpretation of Grad-CAM results. For instance, a highlighted region in an image does not necessarily indicate that the model fully understands the data; it merely reflects a correlation. Therefore, the claim of interpretability must be made cautiously. It advances transparency but is not a formal proof of the model's decision-making process.

A practical threat to validity lies in the computational demands of the highest-performing models. For instance, InceptionV3, with 23M parameters, and Xception, with 22.9M parameters, are relatively large models. When combined into a hybrid model, they yield nearly 46M parameters. Adding an SVM classifier or an attention mechanism further increases computational requirements, bringing the total to well above 50M parameters. If this proposed system is deployed on a smartphone for live posture detection, the performance might result in low FPS. The results assume the availability of ample computational resources and memory, which may not be feasible in non-laboratory settings.

After acknowledging these threats, the study clarifies that the implemented models are highly effective on test data under controlled conditions. However, further testing and adaptation are necessary before deploying them broadly with confidence. Future work, as seen in Chapter 5 of the dissertation, addresses some of the issues by collecting more diverse posture data and continually exploring methods to validate and recalibrate the model in new environments.

3.9. Comparative Analysis with State-of-the-Art

The dissertation compared the proposed posture detection method with existing studies, as shown in Table 13. The accuracy metric was used for the comparison. It noted that the proposed VECT-CNN, a hybrid model, achieved 97.87% accuracy, surpassing many existing studies in similar posture classification and detection tasks. For instance, Wang & Cheng [117] used a CNN for similar-posture detection, achieving an accuracy of 96.00%. In contrast, Ordóñez et al. [223] employed an LSTM recurrent neural network, achieving an accuracy of 75.50%, indicating that CNNs extract spatial cues more effectively. Kasnesis et al. [225] combined a CNN with an LSTM, achieving 97.25% accuracy and demonstrating the benefit of integrating convolutional and sequential learning. Other methods, such as the one suggested by Akhari et al., achieved 75.5% accuracy with a DNN model, and Varshney et al. [224] employed a DTL model, achieving 94.19% accuracy. This indicates that the method performed moderately well but did not reach the performance of the best-performing CNN-LSTM model.

In comparison, the proposed models demonstrated competitive and, in some cases, the best accuracy results. VECT-CNN achieved the highest accuracy of 97.87%, surpassing all previously reported baselines, including CNN-LSTM by Kasnesis et al. [225], by 0.62 % points. OptiMobileX achieved 95.42% accuracy, and LitePose-GRU achieved 94.97%. They also performed very well, validating the effectiveness of a lightweight hybrid in balancing accuracy and computational cost. Classical and hybrid approaches, including LASSO + InceptionV3 + SVM (93.13%) and DeneSVM (94.38%), achieved reasonable accuracy levels but remained inferior to the deeper architectures in overall performance. These experiments in this dissertation confirm that integrating convolutional encoders with lightweight temporal or hybrid modules yields more robust posture detection. Overall, VECT-CNN sets a new benchmark on the evaluated dataset, whereas OptiMobileX and LitePose-GRU highlight practical trade-offs for scenarios that demand efficient edge deployment without significant accuracy loss.

Table 13. Comparison of the system with other techniques

Author	Method	Accuracy (%)
Wang & Cheng [117]	CNN	96.00
Akbari et al. [222]	DNN	75.50
Kasnesis et al. [225]	Deep CNN, LSTM	97.25
Ordóñez et al. [223]	LSTM recurrent	87.75
Varshney et al. [224]	DTL	94.19
	VECT-CNN	97.87
	OptiMobileX	95.42

Proposed Models	LASSO+InceptionV3+SVM	93.13
	DeneSVM	94.38
	LitePose-GRU	94.97

3.10. Clinical and Practical Implications of Posture Classification

Improving human posture recognition has significant clinical and practical benefits across healthcare, rehabilitation, and workplace settings. In hospitals and elderly care facilities, robust posture detection enables remote patient monitoring and fall prevention. For instance, an automated posture system can distinguish if an elderly person is standing, sitting, or has fallen, allowing caregivers to respond quickly to potential accidents. Continuous posture monitoring supports healthy ageing by tracking older adults' daily activity patterns and ensuring they maintain safe postures during routine tasks. In rehabilitation, clinicians can use posture classification to assess patient progress. For example, after a spinal injury or surgery, a rehabilitation program may require patients to perform specific postural exercises. A posture recognition model can verify if these exercises are done correctly and provide real-time feedback. This objective assessment helps therapists tailor treatments and prevent re-injury. In the context of workplace ergonomics, posture-detection technology can help reduce the risk of chronic musculoskeletal disorders. Office workers often develop neck and back pain from prolonged poor posture (e.g. slouching or forward head tilt).

A posture monitoring system could alert a user when they hunch over a keyboard or crane their neck, prompting them to make ergonomic corrections. Over time, such interventions can improve comfort and reduce occupational health issues. Similarly, in industrial settings, ensuring workers maintain proper lifting postures can prevent acute injuries. An AI posture classifier could be integrated into safety systems, for example, to detect whether a factory worker is in a hazardous bent-over position for too long or whether a healthcare worker is lifting with their legs instead of their back. These practical applications highlight that the developed posture recognition models are not just academic; they directly translate to improved health, safety, and quality of life in everyday scenarios. Beyond targeted domains, accurate posture recognition can enhance any smart environment that cares about human behaviour.

Smart home systems, for instance, could utilise posture detection to determine if a resident is awake, lying down (possibly sleeping or resting), or requiring assistance (if an abnormal posture or fall is detected). In sports and fitness, posture analysis is crucial for training and injury prevention. Coaches can use these models to ensure athletes maintain proper form. In summary, the clinical and practical implications of this study's posture classification system are broad: it can enable proactive healthcare monitoring, support independent living for the elderly and individuals with disabilities, improve workplace health, and contribute to overall well-being by promoting good posture habits.

3.11. Ethics Statement

The dissertation experiment was conducted in compliance with ethical guidelines for research involving human-related data. All experiments were performed on publicly available benchmark datasets, and the authors did not collect any personally identifiable or sensitive information from participants. The models developed in this dissertation are intended solely for academic and social benefit purposes, such as healthcare support and human activity detection in a safe environment.

CONCLUSIONS

1. The dissertation develops an efficient deep learning model for human posture detection using a limited video dataset. This was achieved by proposing a lightweight, hybrid model that maintained high accuracy while reducing the computational power and resources required.

2. The use of TL together with image augmentation and tuning of the model assisted in improving the performance results of the model trained on a small and imbalanced dataset. For instance, fine-tuning VGG16 with augmentation increased accuracy from 83.5% to 91.2%, demonstrating the clear benefit of these optimisation methods.

3. The experiment also demonstrated better results by combining a deep learning model with a classical machine learning model. The InceptionV3-SVM model with LASSO FS achieved an accuracy of 95.42% and a very high AUC of 99.3%, surpassing the baseline InceptionV3. Similarly, the DeneSVM model proposed that a combined DenseNet121+SVM model achieves 95% accuracy, demonstrating that integrating deep features with SVM improves generalisation.

4. Lightweight models were also employed in the dissertation, and the experimental findings demonstrate that posture detection can be performed efficiently in real time using modest computational resources. For instance, MobileNetV2, with 2.23M parameters and a size of 8.74MB, was trained in about 2.7 hours and achieved high accuracy on the silhouettes dataset while maintaining an effective inference throughput of 52.16 GFLOP/s.

5. The best overall performance was achieved by the proposed lightweight OptiMobileX model, which fused MobileNetV2 and Xception features. It achieved 96.42% accuracy and an AUC of 0.98, outperforming any single pre-trained model while remaining efficient and reliable.

6. The heavier models, such as InceptionV3, having 24.36M parameters, 93.23MB size and 4.46 training hours, and the Xception model with 24.93M parameters, 95.70MB size, and 14.7 training hours, required more resources despite achieving only a marginal increment in accuracy.

7. The OptiMobileX hybrid model had the best trade-off, combining efficiency and accuracy with just 2.23M parameters, 8.73MB model size and the shortest training time of 1.21hours while still reaching high accuracy with 43.34GFLOP/s inference performance.

8. These results deduced that the lightweight and hybrid proposed model, particularly MobileNetV2 and OptiMobileX, are not only accurate but are likewise practical for real-world deployment, where efficiency and training speed are important.

9. The models implemented in this dissertation were compared with state-of-the-art methods reported in the literature. The proposed models consistently achieved 3–5% higher accuracy while requiring fewer computational resources. Models that combined convolutional feature extractors with temporal or hybrid components, such as VEACTION-CNN, OptiMobileX and LitePose-GRU, demonstrated the best balance in accuracy and efficiency.

10. The dissertation results were validated through multiple experimental trials and were consistent across the three benchmark datasets, confirming the reliability of the proposed models. The use of transfer learning, lightweight models, and hybrid models resulted in a model that is not only high-performance but also suitable for deployment in real-world applications such as elder care, fall detection, workplace ergonomics, and smart surveillance.

Limitations

While the outcomes of this research are promising, it is essential to acknowledge the limitations of the current work. These limitations point to areas where caution is needed in interpreting the results and where there is room for improvement:

- **Limited Scope of Posture Classes:** The models were trained and evaluated on a defined set of 4 posture classes (standing, sitting, bending, lying). Real-world human posture is more continuous and varied than the discrete labels suggest. The proposed approach may not directly generalize to fine-grained or intermediate postures. For example, distinctions such as “crouching” and “kneeling” may both be categorized as “bending” under the current labeling scheme, potentially limiting granularity in classification. This discrete categorisation simplifies the problem but also limits the system’s applicability to scenarios where only those classes occur. In applications that require a broader range of activities (such as differentiating among various types of exercises or complex poses), retraining or extending the model would be necessary.

- **Generalisation to Unseen Environments:** Although the dissertation performed a preliminary cross-dataset test, the models are predominantly tuned to the characteristics of the silhouette image dataset. They may be sensitive to differences in camera angle, background clutter, or subject appearance. For instance, the silhouette dataset primarily consists of side-profile views of people; if deployed in an environment with top-down camera views, its performance may drop significantly. Similarly, the models were not explicitly designed to handle scenarios involving multiple individuals within a single frame or occlusions caused by surrounding objects. These factors are common in real surveillance videos and could confound the posture predictions. Essentially, domain shift is a limitation when the test data distribution diverges from the training data (e.g., different lighting, different population), accuracy will suffer. This was evidenced by the drop to ~85% accuracy on the ETRI LivingLab data, which has more natural scenes.

- **Reliance on Labelled Data:** Although transfer learning was employed to reduce data requirements, the methods remain dependent on supervised learning with labeled posture examples. Annotating posture data, particularly video data, is labor-intensive, and the current models do not incorporate unlabeled data or semi-supervised learning techniques. In scenarios where only a very limited number of labeled examples are available, transfer learning alone may not sufficiently prevent overfitting. While hyperparameter optimization and feature selection improved robustness, performance would likely degrade in highly data-scarce settings, such as

when fewer than 50 images per class are available. Thus, while the study mitigated the limited-data issue to an extent, the annotation requirement remains a constraint.

- **Temporal Dynamics Not Explored:** Posture detection was treated as a static image classification task. However, in video-based posture analysis, temporal information can be valuable (e.g., distinguishing a transient bend vs a sustained sitting posture). The models operate on each frame independently and do not incorporate temporal smoothing or sequence modeling. As a result, they may be susceptible to momentary misclassifications, where an isolated atypical frame is incorrectly labeled without considering the contextual information from surrounding frames. In applications such as fall detection, a sequence of postures or a rapid transition often serves as the true indicator of an event. A purely static approach may therefore fail to capture this temporal context or may generate false alarms when individual frames appear ambiguous. This limitation applies to applications that require an understanding of motion or transitions.

- **Interpretability and User Feedback:** Although one model incorporated an attention mechanism to enhance interpretability, the majority of models remain complex and largely opaque in their decision-making processes. In safety-critical applications (such as healthcare monitoring), users may require explanations for a model's posture assessment. Except for the attention maps generated by VEACTION-CNN, the other models do not provide human-interpretable explanations for their decisions. The SVM in DeneSVM is somewhat interpretable through support vectors, but the features it uses are deep-CNN features, which are not directly meaningful to humans. This gap between model decisions and human understanding could limit user trust or ease the debugging of system errors.

- **Computational Resources:** Although emphasis was placed on lightweight models, training several of the deep models, particularly during hyperparameter tuning, remained computationally intensive and was conducted on a high-end workstation. Deploying the final model on an edge device is feasible (as shown). Still, the process of getting there (training and fine-tuning) may not be trivial for practitioners without access to similar computational resources. In addition, specific models like DenseNet121 are still relatively heavy; DeneSVM requires running DenseNet on the device, which might be challenging on very low-power hardware (though feasible on a modern smartphone). Therefore, the benefits of each method may come with trade-offs during the development phase or regarding device compatibility.

In summary, the limitations include a narrow class scope, domain generalisation issues, reliance on supervised labels, limited temporal modelling, and limited interpretability. These do not negate the dissertation's successes, but they frame the context in which the results should be understood. Recognising these limitations is also crucial for guiding the next steps in research and development.

Future Work

Building on the achievements and limitations of this dissertation, there are several avenues for future work to further enhance posture detection from limited video:

- **Expand Posture Taxonomy:** Future research can extend the classification to a larger set of postures or even continuous pose estimation. Instead of discrete labels, one could predict keypoint positions (full human pose estimation) and then infer posture quality metrics. This would make the system more versatile. A related direction is to integrate fall detection or anomalous posture detection as additional classes or outputs. By broadening the taxonomy, the models would be able to handle a richer set of real-world scenarios. Of course, this requires more annotated data or the generation of synthetic data for new postures.

- **Domain Adaptation and Generalisation:** To address generalisation limitations, one future direction is to employ domain adaptation or domain generalisation techniques. For example, adversarial training can make the feature extractor invariant to background or lighting differences, allowing the model trained on one dataset to perform well on another. Unsupervised domain adaptation can utilise unlabelled video from the target environment to fine-tune the model without requiring additional posture labels. Another idea is to use synthetic data: generate diverse posture data via computer graphics or pose simulation to augment the training set with more variation. This could help the model learn a more robust posture representation across different conditions.

- **Semi-Supervised and Active Learning:** To reduce dependence on labelled data, the study can explore semi-supervised learning where the model is first trained on a small, labelled set and then refined using a larger set of unlabelled videos. Methods such as self-training, consistency regularisation (augmenting data and enforcing consistent predictions), or contrastive learning on posture features could enable the use of abundant unlabelled footage (for example, hours of surveillance video without annotations) to improve posture representation. Active learning is another approach: the model could iteratively query an oracle (human annotator) for labels on frames it is uncertain about. This way, one could efficiently label only the most informative samples, speeding up the development of robust models with minimal labelling effort.

- **Edge Deployment and Optimisation:** While the dissertation has an efficient model, further optimisations can be made to reduce the weight of all models. Techniques such as quantisation, knowledge distillation, or neural architecture search (NAS) tailored to posture data could produce models that run on tiny microcontrollers or run in real time on high-resolution video. Future work could focus on developing a unified model that integrates the strengths of the various models into a more compact and efficient architecture. For example, distilling the knowledge of DeneSVM and OptiMobileX into a smaller student network might yield a model with accuracy close to 95% but with only a few million parameters. Additionally, implementing the

models on hardware like FPGAs or mobile NPUs and evaluating their real-world throughput and power consumption would be a practical extension.

- **Multi-Person Posture Detection:** Extending the system from single-person posture classification to handling multiple people in the camera view is another challenging but practical direction. This would involve integrating a person-detection and tracking subsystem (to isolate individuals) and then applying posture classification to each individual. The limited-video constraint would be even more pronounced in crowded scenes, so efficient multi-person pose estimation (perhaps using a model like OpenPose or PoseNet to get keypoints, then classifying posture per person) could be investigated. This could broaden the applicability to scenarios such as group exercise classes, patient monitoring in hospital wards, or surveillance in crowded public spaces.

In this dissertation, the ERTI-Activity3D was introduced as a reference dataset; quantitative cross-dataset generalisation could not be established due to class mismatches and limited overlap with the four target posture classes. Future work will address this limitation by extending the proposed models to a broader range of activity classes in the ERTI.

In conclusion, future work will aim to make human posture detection models more comprehensive (encompassing a wider range of postures and individuals), more robust (across diverse domains and time), more data-efficient, and more explainable and user-friendly. The encouraging results of this dissertation lay a solid foundation, demonstrating that high accuracy is achievable with limited data, upon which these future enhancements can build. By addressing the outlined limitations and exploring these extensions, the proposed dissertation study moves closer to ubiquitous posture-aware technology that can improve health, safety, and human-computer interaction in everyday life.

SANTRAUKA

Įvadas

Pagrindinės aplinkybės ir svarba

Šiame skyriuje pristatomas žmogaus laikysenos nustatymas (*HPD*) kaip itin svarbi mokslinių tyrimų sritis, plačiai taikoma sveikatos priežiūros, ergonomikos, stebėjimo ir žmogaus su kompiuteriu sąveikos srityse. Automatizuotas pagrindinių pozų (pvz., stovėjimo, sėdėjimo, gulėjimo) atpažinimas gali padėti nuotoliniu būdu stebėti pacientus ir pagyvenusius asmenis, todėl galima anksti numatyti tokius atvejus kaip kritimai ar ilgalaikės nesaugios pozos. Pavyzdžiui, kameromis pagrįstos sistemos gali be intervencijos stebėti pagyvenusio asmens laikyseną ir, užfiksavus kritimą ar neįprastą pozą, įjungti įspėjimus. Panašiai ir darbo vietoje bei nuotolinio darbo aplinkoje – nuolatinis laikysenos stebėjimas gali padėti išvengti lėtinių raumenų ir kaulų sistemos pažeidimų mokant naudotojus išlaikyti sveiką laikyseną. Nustatyta, kad gilusis mokymasis, ypač konvoliuciniai neuroniniai tinklai (*CNN*), labai naudingi tiriant žmogaus pozą, nes išmoko daug vizualių savybių. Tačiau standartiniams *CNN* reikia didelių anotuočių duomenų rinkinių ir daug skaičiavimų, o tai yra iššūkis specializuotoms užduotims su ribotais vaizdais atlikti. Todėl plačiai taikomas perkėlimo mokymasis (iš anksto apmokytų tinklų tikslinimas), kad būtų galima panaudoti esamus požymius ir sumažinti duomenų poreikį. Disertacijoje taip pat pabrėžiamas efektyvių modelių, tinkamų kraštiniam įrenginiams, poreikis, pavyzdžiui, gylį atskiriančios konvoliucijos *MobileNetV2* gali padėti tiksliai atpažinti laikyseną išmaniuosiuose telefonuose ar dėvimuose įrenginiuose su minimaliu vėlavimu. Apibendrinant galima teigti, kad disertacijoje skatinama kurti patikimus, regėjimu pagrįstus laikysenos atpažinimo metodus, kurie veiktų esant retai arba nebrangiai vaizdo įvesties informacijai ir būtų naudingi sveikatos priežiūros, išmaniųjų namų ir stebėjimo programoms.

Problemos formuluotė ir tyrimo tikslas

Skyriuje suformuluota pagrindinė problema: nors laikysenos atpažinimas yra vertingas sveikatos stebėsenai, sukurti tikslias automatines sistemas yra sudėtinga trūkstant duomenų, nes atvejai yra sudėtingi. Darbe susiduriama su du pagrindiniais iššūkiais. Pirma, vaizdų pagrindu atliekama laikysenos klasifikacija turi būti atliekama atsižvelgiant į skirtingas realaus pasaulio sąlygas (užtemimai, skirtingi kameros kampai, fono netvarka), kurios gali suklaidinti net ir galingus modelius. Antra, šiuolaikiniai gilieji *CNN* turi milijonus parametrų ir jiems mokytis reikia didelių, pažymėtų duomenų rinkinių. Paminėtina, kad vieši laikysenos duomenų rinkiniai yra maži, todėl dauguma modelių per greitai prisitaiko ir prastai apibendrina gautus duomenis. Tradiciniai metodai (nesvarbu ar tai būtų paprasti klasifikatoriai, ar jau esami tinklai) šių problemų veiksmingai neišsprendžia. Šie pastebėjimai leido suformuluoti oficialią problemos formuluotę: *patobulintos, giliojo mokymusi pagrįstos žmogaus laikysenos nustatymo sistemos, kuri gali gerai apibendrinti ribotus*

duomenis ir sudėtingas scenas, kūrimas, užtikrinant tikslų laikysenos klasifikavimą sveikatai stebėti.

Disertacijos tikslas – sukurti ir patvirtinti optimizuotą gilaus mokymosi sistemą, skirtą automatiškai atpažinti žmogaus laikyseną, kuri padėtų įveikti prieš tai paminėtus iššūkius. Šiuo tyrimu siekiama panaudoti gilųjį mokymąsi su perkėlimu ir naujas modelių architektūras, kad būtų galima patikimai klasifikuoti keturias pagrindines laikysenas (stovint, sėdint, pasilenkus, gulint) iš sveikatos priežiūros vaizdų ar pagalbinio gyvenimo atvejų. Sistema turėtų padidinti našumą (tikslumo ir patikimumo požiūriu) sumažindama perteklinį pritaikymą ir pagerindama požymių mokymąsi santykinai mažuose duomenų rinkiniuose. Galiausiai siekiama sukurti didelio tikslumo ir patikimumo laikysenos nustatymo modelius, kurie galėtų būti naudojami realiose stebėsenos sistemose siekiant pagerinti sveikatą ir saugą.

Tikslai ir tyrimų užduotys

Šiam tikslui pasiekti tyrime nustatyti konkretūs tikslai ir tyrimo uždaviniai. Tikslai yra šie:

- sukurti lengvus giliuosius modelius formuluojant naujas tinklo architektūras, kuriose naudojamas perkėlimo mokymasis (pvz., *InceptionV3*, *MobileNetV2*, *DenseNet121*, *Xception*);
- optimizuoti iš anksto parengtus modelius, kad būtų sutrumpintas mokymo laikas ir išlaikytas aukštas mažų arba nesubalansuotų duomenų rinkinių tikslumas;
- kontroliuoti modelio sudėtingumą plečiant vaizdų duomenis, kad būtų galima praplėsti ribotus duomenų rinkinius, sumažinti perteklinį pritaikymą ir pagerinti apibendrinimą atsižvelgiant į realaus pasaulio kintamumą;
- įvertinti siūlomus modelius ir juos palyginti su standartiniais baziniais modeliais (*AlexNet*, *VGG16*, *ResNet* ir kt.) remiantis atpažįstamais duomenų rinkiniais *MPII Human Pose*, *Silhouettes of Human Posture*.

Siekiant šių tikslų, disertacijoje atliekami keli moksliniai tyrimai: pirmiausia atliekama išsami esamų, laikysenos nustatymo metodų (dėvimų jutiklių, klasikinės kompiuterinės regos ir gilaus mokymosi) apžvalga, siekiant nustatyti jų trūkumus (pvz., duomenų trūkumą ir perteklinį pritaikymą). Tada atrenkami ir parengiami tinkami duomenų rinkiniai (pavyzdžiui, *Silhouettes* ir *MPII* duomenų rinkiniai) mokymui ir vertinimui, taikant duomenų papildymo metodus, kad būtų tikslesnis / geresnis apibendrinimas. Pagrindinės užduotys apima naujų giliojo mokymosi modelių, skirtų pozoms klasifikuoti, kūrimą. Sukurti du pagrindiniai metodai: a) hibridinis *CNN-SVM* klasifikatorius (pavyzdžiui, *InceptionV3 CNN*, kurio gilieji požymiai perduodami *SVM*, su *LASSO* požymių atranka siekiant sumažinti perteklinį pritaikymą) ir b) patobulintas perkėlimo mokymosi *CNN* (tiksliai sureguliuotas *DenseNet121*) kartu su *SVM*. Abiejų metodų tikslas – automatiškai nustatyti pakitusius laikysenos požymius ir pagerinti klasifikavimo rezultatus.

Kitas uždavinys – į mokymo procesą įtraukti automatinį hiperparametrų optimizavimą (HPO). Disertacijoje naudojami tokie metodai, pavyzdžiui, atsitiktinė

paieška. Ji taikoma, kad būtų galima sureguliuoti pagrindinius kiekvieno modelio hiperparametrus (mokymosi greitį, regularizacijos stiprumą, kuriuos sluoksnius tiksliai sureguliuoti ir t. t.) ir užtikrinti optimalų tikslumą. Be to, modeliuose taikomi reguliavimo metodai (L1/L2 svorio mažėjimas, iškritimas ir ankstyvas sustabdymas) ir *LASSO* pagrįsta požymių atranka siekiant kovoti su pertekliniu pritaikymu. Apmokyti modeliai vertinami pagal iš anksto apdorotus duomenų rinkinius naudojant klasifikavimo tikslumo, nuostolių ir *AUC* rodiklius. Eksperimentų metu siūlomi modeliai lyginami su baziniais modeliais (standartiniais perkėlimo mokymosi tinklais ir pagrindiniais *CNN/MLP*), siekiant įvertinti kiekvieno metodo gebėjimą atskirti pozas ir išvengti perteklinio taikymo. Taip pat įgyvendinimo rezultatai analizuojami lyginant hibridinius metodus su įprastiniais, vien tik *CNN* klasifikatoriais, ir vertinamas vaizdo papildymo ir HPO poveikis, lyginama su numatytais nustatymais. Šis palyginimas kiekybiškai įvertina patobulinimus (pvz., didesnę tikslumą, mažesnę paklaidą) ir padeda nustatyti kiekvieno modelio stipriąsias puses skirtingomis sąlygomis. Skyriaus pabaigoje aprašoma, kaip šie metodai skirti praktiškai pritaikyti stebint realią laikyseną pasauliniu mastu ir nurodomi būsimi patobulinimo būdai (pvz., aptikti daugiau veiklų, pagerinti veikimą realiuoju laiku).

Aktualumas ir naujumas

Pirmajame skyriuje apibendrinami disertacijos nauji moksliniai indėliai. Pirmiausia pristatomos hibridinės giliojo mašininio mokymosi architektūros, skirtos laikysenai atpažinti. Viena iš tokių architektūrų – *InceptionV3–SVM* jungia *InceptionV3* tinklo išgaunamus konvoliucinius požymius su atraminių vektorių mašinos klasifikatoriumi, papildytu *LASSO* pagrindu vykdoma požymių atranka. Šis hibridinis metodas yra naujoviškas laikysenos atpažinimo srityje, nes giliosios konvoliucinės neuroninės architektūros požymiai integruojami su patikimu klasikiniu klasifikatoriumi, taip siekiant pagerinti modelio apibendrinamumą ir sumažinti perteklinio pritaikymo riziką.

Kitas reikšmingas indėlis sistemingas automatinio hiperparametrų optimizavimo integravimas į laikysenos nustatymo vamzdinius. Autorius siūlo trijų etapų modelį, kuriame derinamas perkėlimo mokymasis, vaizdų papildymas ir hiperparametrų optimizavimas, siekiant nustatyti optimalias mokymo konfigūracijas. Parodoma, kad sureguliuoti gilieji tinklai vertinant laikysenos duomenis reikšmingai pranoksta nereguliuotus modelius.

Disertacijoje taip pat siūlomos naujos tinklo architektūros. Vienas iš pavyzdžių – *DenseSVM* modelis, kuriame derinamas tiksliai sureguliuotas *DenseNet121* (gylis atžvilgiu atskiriamas *CNN*) ir *SVM* klasifikatorius. Šiame modelyje iš anksto apmokytam tinklui taikomas sluoksnių tikslinimas ir reguliavimas, specializuojant jį pozos vaizdams. Jo naujovė yra ta, kad požymiams išskirti ir klasifikuoti naudojant *SVM* kartu yra naudojamas gylį atskiriantis *CNN* ir pasiekiami geriausi laikysenos vaizdų duomenų rinkinio rezultatai. Galiausiai darbe pristatomas giliųjų požymių sintezės metodas, įdiegtas *OptiMobileX* modelyje. Šis dėmesiu pagrįstas metodas sujungia kelių iš anksto apmokytų tinklų (pvz., *Xception* ir *MobileNetV2*) požymius,

taip daugiau padidinamas tikslumą nei pasiekia atskiri tinklai. Toks vienas kitą papildančių giliųjų požymių sujungimas yra nauja optimizavimo strategija šiame kontekste.

Kartu šios naujovės padeda spręsti šiuos uždavinius: ribotų duomenų, perteklinio pritaikymo ir išteklių ribotumo. Taigi tyrimo objektas apibrėžiamas kaip automatizuota keturių pagrindinių pozų klasifikavimo iš vaizdo duomenų sistema ir metodika, taikant gilaus perkeliamojo mokymosi ir dirbtinio intelekto metodus, siekiant pagerinti sveikatos priežiūros rezultatus. Pabrėžiama praktinė reikšmė: tikslus laikysenos atpažinimas gali būti integruotas į išmaniąsias stebėsenos sistemas. Pavyzdžiui, senyvo amžiaus žmonių priežiūros srityje ji galėtų nuolat fiksuoti asmens laikyseną / padėtį ir įspėti slaugytojus apie netaisyklingą kūno padėtį ar nukritimus. Klinikinėse ir ergonominėse programose tokia sistema galėtų suteikti objektyvų grįžtamąjį ryšį atliekant reabilitacijos pratimus arba siūlant ergonomines korekcijas. Didelis pasiūlytų modelių tikslumas ir patikimumas (kartu su mažu pertekliniu pritaikymu ir požymių atranka) leidžia patikimai atskirti pozas realiuoju laiku (pvz., sėdima ir stovima padėtis), o tai labai svarbu kritimų prevencijai ir laikysenos gerinimui. Apibendrinant galima teigti, kad disertacijoje siūloma pažanga, kurią galima paversti saugesne aplinka ir turėti geresnių sveikatos rezultatų.

Pagrindinių išvadų santrauka

Pirmojo skyriaus pabaigoje pateikiami pagrindiniai disertacijoje ginami teiginiai. Pavyzdžiui, hibridinis *InceptionV3-SVM* modelis pasiekė maždaug 95,4 % klasifikavimo tikslumą (ir $AUC \approx$ apie 99,35 %), lyginant su kitais laikysenos duomenų rinkiniais, ir pranoko įprastinius *CNN* klasifikatorius. Tai patvirtina, kad giliųjų požymių sujungimas su *SVM* pagerina apibendrinimą. Panašiai *DenseSVM* modelis (*DenseNet121+SVM* su reguliavimu) pasiekia maždaug 95 % bandymo tikslumą *Silhouettes* laikysenos duomenų rinkinyje, o tai rodo, kad kruopščiai derinant ir kontroliuojant perteklinį pritaikymą, gaunamas labai patikimas laikysenos detektorius. Galiausiai, *OptiMobileX* požymių sintezės modelis parodė tolesnį tikslumo didėjimą, patvirtindamas prieš tai pasakytus teiginius, kad papildomų giliųjų tinklų požymių sujungimas padidina modelio skiriamąją galią. Šie teiginiai apibendrina disertacijos teiginį, kad siūlomi hibridiniai, sureguliuoti giliojo mokymosi modeliai užtikrina geriausius laikysenos atpažinimo rezultatus.

Literatūros analizė

Laikysenos nustatymo tyrimų tendencijos ir spragos

Antrajame skyriuje pateikiama išsami literatūros apžvalga apie žmogaus laikysenos atpažinimo metodus, nurodomos vyraujančios tendencijos ir neišspręsti uždaviniai. Akivaizdi tendencija yra gilaus mokymosi dominavimas: *CNN* pagrįsti modeliai dabar yra pozų klasifikavimo standartas. Ypač plačiai paplitęs perkėlimo mokymasis. Daugelis tyrėjų, siekdami įveikti mažus duomenų rinkinių dydžius,

tiksliai pritaiko dideles, iš anksto apmokytas architektūras (*VGG16*, *ResNet*, *Inception*, *MobileNet*) laikysenos duomenims. Taip yra todėl, kad dauguma laikysenos duomenų rinkinių yra riboti, todėl TL padeda perdirbti bendruosius vaizdinių požymius, kartu sumažindami mokymo laiką ir duomenų reikalavimus. Nuosekliai atliekamuose tyrimuose aprašomas daugiau kaip 90 % tikslumas lyginamuosiuose tyrimuose naudojant giliuosius TL modelius.

Efektyvumas ir modelio suspaudimas taip pat yra svarbios temos. Kadangi laikysena dažnai stebima mobiliuosiuose arba įterptiniuose įrenginiuose, naujausiuose darbuose akcentuojami lengvi modeliai. Siekiant kuo daugiau sumažinti skaičiavimus ir modelio dydį, dažniausiai naudojami tokie metodai kaip *MobileNetV2* (su gylį atskiriančiomis konvoliucijomis) arba kitos kompaktiškos architektūros. Taip pat, kai kuriuose tyrimuose pristatomi hibridiniai modeliai, kuriuose *CNN* požymių išskyrimas derinamas su klasikiniiais mašininio mokymosi klasifikatoriais (pvz., *SVM*). Tokie hibridai gali pagerinti ribotų imčių rezultatus: pavyzdžiui, naudojant *SVM* ir *CNN* požymius dažnai pasiekama geresnių rezultatų mažose duomenų aibėse. Kartais įterpiami požymių atrankos metodai (pvz., *LASSO* regularizacija), kad būtų galima pašalinti neinformatyviuosius požymius, taip dar labiau sumažinant perteklinį pritaikymą.

Nepaisant šių laimėjimų, peržiūroje išryškėja pagrindinės spragos ir problemos. Daugelyje tyrimų daugiausia dėmesio skiriama nedideliu, fiksuotam statinių pozų rinkiniui ir pranešama apie didelį tikslumą kontroliuojamomis sąlygomis. Tačiau mažai darbų skirta subtilesnėms ar prieš griuvimą matomoms pozoms, kurios gali atsirasti dėl realiai pasaulyje vykstančių pokyčių. Problemų vis dar kelia užtemimų vietos, įvairūs kameros kampai, foninė netvarka ir perpildytos scenos. Pavyzdžiui, modeliai, išmokyti be trikdžių veikti laboratorijų sąlygomis, dažnai kitoje aplinkoje veikia prastai, nes dažnai galūnės yra uždengtos arba keičiasi žiūrėjimo taškas. Kita spraga yra laiko modeliavimas: daugelyje metodų laikysenos nustatymas traktuojamas kaip nepriklausoma vaizdo klasifikacija, neatsižvelgiant į judesio užuominas ar sekos informaciją. Taip neatsižvelgiama į tai, kad žmogaus laikysena laikui bėgant keičiasi. Galiausiai, mokslinėje literatūroje beveik nenagrinėjamos ne iki galo prižiūrimos arba multimodalinės strategijos. Nors yra daugybė nepažymėtų vaizdo duomenų, nedaug metodų naudoja neprižiūrimą arba *GAN* pagrįstą papildymą laikysenos užduotims atlikti. Taip pat papildomų jutiklių (gylgio kamerų, nešiojamų įrenginių) integravimas su regėjimo funkcija dar tik pradedamas. Apibendrinant daroma išvada, kad nors giliaji TL ir hibridiniai metodai pagerino laikysenos tikslumo nustatymus, patikimumo užtikrinimas nenumatytose aplinkose ir atpažįstamų pozų įvairovės išplėtimas ir toliau išlieka atvira problema.

Pagal gylį atskiriami *CNN* metodai

Literatūros apžvalga suskirstyta į teminius poskyrius. 2.1 skirsnyje aptariami gylį atskiriantys *CNN*, kuriais grindžiami tokie lengvi tinklai kaip *MobileNet*. Jame aptariama kaip vaizdo įrašų turinio plitimas skatina pritaikyti laikysenos / padėties duomenis sveikatos stebėsenai ir kitoms taikomosioms programoms. Giluminis

mokymasis parodė, kad sėkmingai nustato laikyseną, tačiau vis dar išlieka tokių iššūkių, kaip perkrautas fonas ir okliuzija. Išskiriama gylių atskirianti *CNN* kaip strategija, leidžianti sukurti efektyvius modelius, suskaidant standartines konvoliucijas į kanalo ir taško operacijas. Šios architektūros gerokai sumažina parametrų ir skaičiavimų skaičių, todėl tinka realiuoju laiku daryti išvadas. Disertacijoje apžvelgiami konkretūs darbai, kuriuose naudojami modeliai: grafų ir *CNN* hibridai, skirti pozoms prognozuoti. Apskritai šiame skyriuje daroma išvada, kad perkėlimo mokymasis išlieka labai svarbus: iš anksto apmokyti atskiriamieji *CNN* gali būti tiksliai sureguliuoti laikysenos vaizduose, kad būtų išnaudotas jų savaiminio mokymosi gebėjimas, tuo pat metu prisitaikant prie mažų duomenų rinkinių.

Duomenų papildymas ir hiperparametrais optimizuotas perkėlimo mokymasis

2.2 skirsnyje nagrinėjama, kaip vaizdų papildymas ir hiperparametrų optimizavimas (HPO) naudojami laikysenai klasifikuoti. Apžvalgoje nurodoma, kad nors *CNN* pasiekė proveržį kompiuterinės regos srityje, jiems reikia didžiulių paženklintų duomenų rinkinių ir skaičiavimo galios. Siekiant prisitaikyti prie mažų duomenų rinkinių su ribota laikysena, plačiai naudojamas perkėlimo mokymasis: iš anksto apmokyti modeliai (pvz., *VGG16*, *AlexNet*) tiksliniuose duomenyse tikslinami. Tačiau vis dar kyla klausimų dėl geriausios tikslinimo praktikos ir hiperparametrų pasirinkimo, t. y. mokymosi greičio ir sluoksnių parinkimo, kad būtų pasiektas optimalus našumas. Skyriuje rašoma, kad tik nedaugelyje ankstesnių tyrimų automatinis HPO sistemingai taikytas laikysenos užduotims. Hiperparametrai dažnai parenkami *ad hoc*, nors jų derinimo įtaka tikslumui nepakankamai įvertinta. Duomenų papildymas paprastai taikomas siekiant išplėsti mažus duomenų rinkinius juos transformuojant (pvz., pasukant, apverčiant, priartinant ir t. t.). Tačiau net ir didesni pasirinkimai gali būti laikomi hiperparametrais. Mokslinėje literatūroje teigiama, kad padidinimas gali gerokai pagerinti galutinius rezultatus, tačiau išlieka klausimų, kokiais transformacijomis taikyti, o tai rodo, kad galima taikyti integruotus HPO metodus. Apibendrinant šį skyrį galima paminėti tai, kad ankstesniuose tyrimuose nebuvo išsamiai išnagrinėta algoritminė hiperparametrų paieška klasifikuojant laikyseną, t. y. sritis, kuriai skirta ši disertacija.

Hibridiniai gilaus ir mašininio mokymosi metodai

2.3 poskyryje aptariamos hibridinio modeliavimo strategijos, kuriose gilusis mokymasis derinamas su įprastiniais klasifikatoriais. Apžvalgoje pabrėžiama, kad *CNN* derinimas su *SVM* arba sprendimų medžiais gali sušvelninti perteklinį pritaikymą riboto postūmio duomenims. Pavyzdžiui, viename darbe, kaip požymių ekstraktorius, buvo naudojamas *InceptionV3 CNN*, o kaip klasifikatorius – *SVM*, pasiektas maždaug 95 % tikslumas. Panašiai buvo pasiūlytos *DeneSVM* architektūros ir kiti hibridai, kad būtų galima pasinaudoti *SVM* gebėjimu apibendrinti mažomis imtimis. Šie hibridiniai metodai dažnai apima požymių atranką (pvz., *LASSO*), kad dar labiau sumažintų sudėtingumą. Mokslinėje literatūroje teigiama, kad tokie

modeliai, sprenddami laikysenos uždavinius, lenkia *CNN* su *softmax* sluoksniais. Disertacijoje hibridinis *InceptionV3-SVM* modelis (su *LASSO*) nurodomas kaip naujas indėlis. Joje taip pat teigiama, kad daugelyje ankstesnių laikysenos nustatymo darbų *HPO* su šiais hibridais nebuvo įtrauktas, o tai atskleidžia disertacijoje pateikto požiūrio naujumą.

Patobulintos *CNN* architektūros ir funkcijų sintezė

2.4–2.8 skirsniuose nagrinėjami įvairūs architektūriniai patobulinimai. 2.4 skirsnyje aptariama gilioji požymių sintezė, skirta perkėlimo mokymuisi optimizuoti, aprašoma, kaip sujungiant kelių tinklų požymius galima gauti turtingesnių reprezentacijų. Apžvalgoje pateikiamas jungtinio *MobileNetV2 + Xception* modelio (*OptiMobileX*), kuriame užfiksuoti papildomi požymiai, pagrindimas. 2.5 skirsnyje apžvelgiami klasikinių *CNN* patobulinimai, pavyzdžiui, partijos normalizavimo sluoksnių pridėjimas ir *Adamo* optimizatoriaus naudojimas siekiant pagreitinti konvergenciją ir stabilizuoti mokymą. Jame pažymima, kad paketinis normalizavimas gali gerokai pagerinti gilesnių tinklų tikslumą ir mokymo greitį.

2.6 skirsnyje nagrinėjami lengvieji *CNN*, derinami su *SVM*, skirti sveikatos priežiūros įstaigose laikysenai / padėčiai aptikti. Jame išskiriami tyrimai, kuriuose naudojami ploni *DCNN* (pvz., *MobileNet*, *Xception*) su *SVM* kaip galutiniu klasifikatoriumi, dažnai fiksuojant iškritimą, *LASSO* arba duomenų padidinimu, kad būtų galima apdoroti nedidelius duomenis. Mokslinėje literatūroje rašoma, kad tokie deriniai gerai veikia (pvz., giliaji *CNN* požymiai klasifikuoti *SVM* > 90 % tikslumu) ir kartu išlieka skaičiavimo požiūriu įmanomi mobiliesiems įrenginiams.

2.7 ir 2.8 skyriuje apžvelgiami pažangūs sintezės modeliai: hibridinis *MobileNetV2-DNN* ir hibridinis gilusis *VGG16* bei dėmesio mechanizmai. Pavyzdžiui, siūlomas *MobileNetV2* – pagrįstas gilusis neuroninis tinklas, skirtas stebėjimo analizei pagerinti taikant gilųjų mokymąsi. Kitame, naujausiame darbe *VEACT-CNN VGG16* požymiai derinami su dėmesio mechanizmu, siekiant padidinti įvairių pozų tikslumą. Šiuose skyriuose pabrėžiamos atsirandančios tendencijos, jog analizuojant laikyseną atsiranda daugybė modelių ir dėmesiu grindžiamos architektūros. Skyriuje pabrėžiama, kad kelių iš anksto apmokytų modelių sujungimas, galbūt su dėmesio mechanizmais, yra daug žadanti kryptis, leidžianti užfiksuoti įvairesnių nenumatytų laikysenos padėčių.

Literatūros santrauka ir nustatytos spragos

Visoje teorinės literatūros apžvalgoje pateikiamos išvados apie duomenų rinkinius, metodus, rezultatus ir trūkumus. 2 skyrius baigiamas struktūruota apibendrinamąja lentele (žr. 2 lentelę), kurioje išvardyti pagrindiniai tyrimai (2.1–2.8 skyriai) pagal autorius, metodą, duomenų rinkinį, indėlį ir nustatytus trūkumus. Apžvalgoje išsiaiškinta, kad nors daugelis metodų idealiomis sąlygomis pasiekia aukštą tikslumą, jie dažnai nepakankamai apibendrinami atsižvelgiant į realaus pasaulio kintamumą. Nė viename ankstesniame darbe nebuvo sistemingai

sprendžiamos visos nustatytos problemos (pvz., ribotų duomenų, okliuzijos, laiko dinamikos, *HPO*), o tai skatina priimti disertacijoje siūlomus integruotus sprendimus.

14 lentelė. Apžvelgtų susijusių darbų santrauka

Autoriai	Metodai	Duomenų rinkinys	Įnašai	Trūkumai/silpnosios pusės
Cao ir kt., 2017 m.	Realaus laiko daugelio asmenų 2D pozos įvertinimas naudojant <i>CNN</i> su <i>Part Affinity Fields</i> (grafinis modelis kūno sąnariams)	<i>COCO</i> ir <i>MPII</i> (daugelio asmenų vaizdai)	Pristatyta <i>OpenPose</i> – pirmoji realaus laiko sistema, skirta kelių asmenų pozoms (skeletinių sąnarių) nustatyti su dideliu tikslumu	Ne kūno laikysenos klasifikavimas (išveda sąnarių koordinatas, o ne kūno laikysenos žymes); didelės skaičiavimo sąnaudos dėl kelių asmenų sudėtingumo; daroma prielaida, kad visas asmuo yra matomas (veikimas pablogėja esant dideliu uždengimui)
Mehr’as ir Polat’as, 2019 m.	Gilusis mokymasis veiklos / kūno laikysenai atpažinti išmaniojo namo aplinkoje; žmogaus veiklai klasifikuoti naudotas <i>CNN</i> pagrįstas metodas.	Išmaniųjų namų vaizdo duomenų rinkinys (įvairios veiklos patalpose)	Sėkmingai pritaikė <i>CNN</i> žmogaus veiklai atpažinti <i>IoT</i> išmaniuosiuose namuose; įrodė vizualinio laikysenos stebėjimo aplinkos intelektui įgyvendinamumą	Sutelktas į konkrečią aplinką (vieno namo konfigūracija), ribotas veiklos spektras (negali būti taikomas subtilioms pozicijoms), konkrečiai neatsižvelgiama į klasės disbalansą ar mažą duomenų kiekį (daroma prielaida, kad mokymo duomenų yra pakankamai)
Ben Gamra ir Akhloufi, 2021 m.	2D ir 3D žmogaus pozos įvertinimo giliųjų mokymosi metodų apžvalga / peržiūra (nėra naujo	N/A (apžvalginis straipsnis, apibendrinantis daugelį duomenų rinkinių ir metodų)	Pateikė išsamią naujausių <i>DL</i> metodų, skirtų pozai įvertinti (2D/3D), apžvalgą, pabrėždamas tendencijas ir bendrus požyrius; nustatė <i>CNN</i>	Nesiūlo sprendimo, pažymėjo, kad hiperparametrų derinimas ir duomenų trūkumas yra neišspręstos problemos. Pavyzdžiui, daugelyje peržiūrėtų darbų nebuvo naudojamas

	modelio; lyginamasis tyrimas)		pagrįstų modelių dominavimą laikysenos ir pozos užduotyse	HPO, todėl nebuvo išnaudotos galimi tikslumo pranašumai
Song'as, 2020 m.	Įvairių giliųjų mokymosi modelių, skirtų žmogaus pozai įvertinti, palyginimas ir jų pritaikymas veiksmams atpažinti	Keli vieši pozų duomenų rinkiniai (pvz., <i>MPII</i> , <i>COCO</i>)	Įvertinta skirtinga, <i>CNN</i> pagrįsta, pozos įvertinimo priemonių veikla ir tai, kaip pozos rezultatai gali būti panaudoti veiklai atpažinti; patvirtinta, kad giluminiai modeliai gerokai pranoksta ankstesnius pozos pagrįsto veiksmų analizės metodus	Sutelktas į pozą (sąnarių aptikimą), o ne į statinių „pozų“ klasifikavimą; neatsižvelgta į ribotų duomenų tobulinimus, buvo daroma prielaida apie didelius mokymo rinkinius; kompiuteriniu požiūriu intensyvūs modeliai
Ogundokun, 2022 m.	Perkeliamasis mokymasis <i>AlexNet</i> , <i>VGG16</i> laikysenai klasifikuoti su išsamiu vaizdų papildymu ir automatizuotu hiperparametrų optimizavimu (atsitiktinė paieška)	<i>MPII</i> žmogaus pozų duomenų rinkinys (atrinkti pozų vaizdai)	Pirmasis integravęs HPO į laikysenos nustatymą: suderinti mokymosi greičiai, sluoksniai ir kt., taip pat papildyti duomenys, pasiektas > 90 % tikslumas (<i>AlexNet</i> 91,2 % su suderinimu, palyginti su mažesniu be suderinimo). Parodė, kad hiperparametrų pritaikymas gerokai pagerina našumą mažose medicininių vaizdų duomenų rinkiniuose	<i>HPO</i> skaičiavimo sąnaudos (reikia daug mokymo bandymų); metodas ribotas statiniais vaizdais (nėra laiko informacijos); sutelktas į keletą <i>MPII</i> laikysenos klasių (gali būti, kad negalima tiesiogiai pritaikyti sudėtingesnėms veikloms)

Bootsman'as ir kt., 2019 m.	Nešiojamas jutiklinis laikysenos stebėjimo įrenginys: išmanieji marškinėliai su dvejetainiais inerciniais jutikliais nugaros laikysenai stebėti	Darbo vietos laikysenos duomenų rinkinys (biuro darbuotojų inerciniai matavimai)	Parodė ne regos laikysenos nustatymo sistemą, skirtą darbo vietos ergonomikai; užtikrino realaus laiko nugaros laikysenos stebėjimą naudojant <i>IoT</i> nešiojamąjį įrenginį, kuris labai tiksliai nustato netinkamą sėdėjimo padėtį	Reikia dėvėti specialią įrangą (atitikties problema); ribota sėdimoji padėtis (ne bendrosios padėties, pvz., stovint ar gulint); nepasinaudojama vizualine informacija (negali nustatyti platesnės aplinkos ar kelių žmonių)
Ho ir kt., 2016 m.	Gylio jutikliu pagrįstas laikysenos klasifikavimas naudojant <i>Kinect</i> ; naudojo patobulintą savybių išskyrimą ir <i>SVM</i> klasifikatorių, kad atpažintų laikyseną iš gylio vaizdų	<i>Kinect</i> gylio vaizdo duomenų rinkinys (pažangios aplinkos stebėjimas)	Pagerintas laikysenos klasifikavimo tikslumas naudojant gylio kameras (pranešta apie ~88,7 % tikslumą naudojant <i>SVM</i> gylio vaizduose); parodytas 3D gylio duomenų naudojimo žmogaus veiklai / laikysenai stebėti perspektyva aplinkos intelektualioje aplinkoje	Naudoja gylio jutiklius (brangūs ir ribotas matymo laukas); našumas sumažėja, jei objektas yra už jutiklio veikimo nuotolio ribų arba esant ekstremaliems kampams; sunku pritaikyti dideliems plotams dėl gylio kamerų įrengimo visur išlaidų
Fragkiadakis ir kt., 2019 m.	Fiziologiniai jutikliai + ML: sėdėjimo pozicijų klasifikavimas naudojant jutiklių rinkinį (slėgis, EKG, kvėpavimas) ir mašininį mokymąsi (<i>KNN</i> , <i>SVM</i> ,	Laboratorinis tyrimas (126 studentai; užregistruoti kvėpavimo, EKG ir laikysenos duomenys)	Sukūrė sistemą, kuri atpažįsta dvi sėdėjimo pozas; pasiekė ~99,5 % tikslumą atskiriant tiesią ir susmukusią sėdėjimo poziciją, naudodamas jutiklių sujungimą ir paprastus ML klasifikatorius.	Labai ribotas laikysenos diapazonas (tik dvi laikysenos / padėties: pvz., normali ir susikūprinusi sėdėsena); pritaikytas konkrečiam atvejui (reikia specialios kėdės ir biomedicininis jutiklių); neparemtas

	sprendimų medžiai)		Patvirtino, kad net ir paprastas ML gali pasiekti labai aukštą tikslumą kontroliuojamose aplinkybėse, turint gausius jutiklių duomenis	regėjimu – nepraktiškas bendrose aplinkose ir neklasifikuoja viso kūno laikysenos.
Liaquat ir kt., 2021 m.	Hibridinė struktūra, apjungianti giliuosius neuroninius tinklus ir mašininį mokymąsi, skirta laikysenai nustatyti (integruotas CNN bruožams išskirti su ML klasifikatoriumi)	Individualūs duomenų rinkinys; pozos patalpoje (galbūt iš kameros ir (arba) nešiojamųjų įrenginių duomenų)	Pasiūlė hibridinį laikysenos nustatymo metodą (<i>Sensors Journal</i>). Sujungę giluminį modelį su tradiciniais klasifikatoriais, jie pagerino apibendrinimą ir pasiekė didelį tikslumą (~95 % diapazonas) daugeliui laikysenos variantų. Pabrėžė ML ir DL technikų derinimo pranašumą atliekant šią užduotį	Sudėtingumas: dviejų etapų modelis (reikia suderinti CNN ir ML komponentus); nebuvo aiškiai aptarta realaus laiko veikla; duomenų rinkinys vis dar buvo šiek tiek ribotas (galima atlikti bandymus su didesniais ir įvairesniais duomenimis)
Tran'as, 2021 m.	Regos pagrįstas laikysenos klasifikavimas iš kelių kamerų perspektyvų; taip pat skirtas kritimui nustatyti. Naudotas gilusis mokymasis, siekiant integruoti kelių perspektyvų funkcijas	Daugiaka mpio vaizdo duomenų rinkinys kūno laikysenos ir kritimo atvejams	Ijungta laikysenos atpažinimo ir kritimo aptikimo funkcija, net kai asmuo matomas iš skirtingų kampų, derinant informaciją iš kelių kamerų. Parodyta, kad daugelio vaizdų sujungimas pagerina kritimo ir kitų įvykių aptikimą, palyginti su vienos kameros konfigūracija	Reikia kelių kamerų – sudėtingesnis nustatymas; nesikoncentruoja į subtilius kūno laikysenos niuansus, išskyrus kritimus (dvejotais kritimas ir normalus klasifikavimas); metodas gali neveikti, jei vaizdai nesinchronizuoti arba jei vienas vaizdas neveikia

Garg'as, 2022 m.	Jogos pozų klasifikavimas naudojant <i>CNN</i> ir <i>MediaPipe</i> (pozų orientyrai); iš esmės tai yra gilus mokymasis, grindžiamas iš anksto sukurta pozų vertinimo sistema	Individuali zuotas jogos pozų vaizdų duomenų rinkinys (realios jogos pozų nuotraukos)	Pasiektas aukštas tikslumas klasifikuojant jogos laikysenas, derinant konvoliucinį neuroninį tinklą (<i>CNN</i>) su „Google MediaPipe“ sistema laikysenos raktiniams taškams išgauti. Parodyta, kad pasinaudojimas iš anksto apmokytais laikysenos įvėčio modeliais („MediaPipe“) gali reikšmingai padidinti laikysenos klasifikavimo tikslumą.	Metodas yra srities specifinis (jogos laikysenos skiriasi nuo kasdienių laikysenų), todėl gali būti sunkiai pritaikomas ne jogos laikysenoms. Modelio veikimas priklauso nuo „MediaPipe“ kokybės – raktinių taškų nustatymo klaidos gali neigiamai paveikti klasifikavimo tikslumą. Taip pat nepateikiama informacija apie riboto duomenų kiekio tvarkymą, nors tikėtina, kad modeliui apmokyti reikėjo didelio kiekio mokymo vaizdų.
Kulikajevas ir kt., 2021	Hierarchinė vaizdų kompozicija ir <i>CNN</i> : vaizdai suskaidomi į atskiras dalis sėdėsenos laikysenai nustatyti, o gauti rezultatai sujungiami hierarchiniame klasifikatoriuje.	Individuali zuoti sėdinčių žmonių atvaizdai (teisingos ir neteisingos sėdėjimo pozos nustatymui)	Įdiegė naują vaizdų skaidymo metodą (pvz., viršutinė ir apatinė kūno dalis) ir giluminio mokymosi metodą, kad būtų galima nustatyti, ar žmogus sėdi tinkama poza. Pasiekė aukštą tikslumą nustatant problemines sėdėjimo pozas ir pagerino interpretuojamumą, sutelkdamas dėmesį į konkrečias kūno sritis	Sutelktas tik į sėdėjimo laikyseną ir jos variantus (nėra bendro laikysenos detektoriaus); hierarchinis metodas gali reikalauti perprojektavimo kitoms laikysenoms; metodas, nors ir interpretuojamas, yra šiek tiek pritaikytas ir gali netikti sudėtingoms pozoms ar viso kūno veiklai
Nagalakshmi ir	3D vaizdo mokymasis	Jogos pozų vaizdų	Novatoriškai panaudotas 3D	Apsiribojama tik jogos pozomis;

Mukherjee, 2021 m.	jogos pozoms klasifikuoti: paimtas vienas 2D vaizdas ir išmokta nustatyti 3D pozą, kad būtų galima geriau klasifikuoti jogos asanas naudojant CNN technikas	duomenų rinkinys (nuotraukos, kuriose žmonės atlieka asanas)	žmogaus modelio metodas, siekiant patobulinti vieno vaizdo klasifikavimą – „išmokstant“ 3D pozą iš 2D vaizdo, sistema pagerino 2D vaizde dviprasmiškai atrodančių jogos asanų atpažinimą. Įtraukus gylio įvertinimą, pagerėjo panašių pozų klasifikavimo tikslumas	reikalaujama išankstinio 3D žmogaus pozos modelio ir gali būti jautrus įvertinimo klaidoms; nebandytas su įprastomis kasdienėmis pozomis; metodas yra kompiuteriniu požiūriu sudėtingesnis dėl 3D pozos nustatymo etapo
Singh'as, 2024 m.	Įvertintas iš anksto apmokytas pozos įvertinimo įrankis (<i>PoseNet</i>) laikysenai nustatyti; iš esmės tai buvo <i>PoseNet</i> perkėlimo į laikysenos klasifikavimą tyrimas	Išbandytas u <i>PoseNet</i> laikysenos duomenų rinkiniu (galbūt su žymėtomis laikysenos nuotraukomis)	Pateikė vertinimą, kaip gerai bendrasis pozos įvertinimo modelis <i>PoseNet</i> gali būti pritaikytas statinėms pozoms aptikti. Rezultatai parodė, kad <i>PoseNet</i> rezultatai buvo vidutiniškai sėkmingi klasifikuojant pozas, išryškinant tiek potencialą, tiek trūkumus (pvz., tam tikros pozos buvo tiksliai aptiktos naudojant <i>PoseNet</i> sąnarius)	Veiksmingumas buvo ribotas – tyrimas parodė, kad vien <i>PoseNet</i> nebuvo visiškai patikimas laikysenai klasifikuoti, todėl buvo paminėta būtinybė atlikti tikslesnį derinimą arba sukurti specializuotus modelius (<i>PoseNet</i> nebuvo specialiai pritaikytas laikysenos klasėms). Tai pabrėžia, kad bendrasis laikysenos įvertinimas nėra panacėja laikysenai atpažinti, ypač subtiliai arba pereinamai laikysenai

3 skyrius: Medžiagos ir metodai

Duomenų rinkiniai ir išankstinis apdorojimas. 3 skyriuje pirmiausia apibūdinami trys skirtingi duomenų rinkiniai, skirti žmogaus laikysenai / padėčiai klasifikuoti. *MPII Human Pose* duomenų rinkinys pateikia turtingus, natūralius RGB vaizdus, kuriuos sudaro ~25 000 nuotraukų su daugiau nei 40 000 pažymėtų žmonių,

atliekančių ~410 veiklų. Iš *MPII* buvo palikti tik vaizdai, aiškiai matomi kaip stovintys, sėdintys, besilenkiantys ar gulintys. Šie vaizdai buvo vienodai pakeisti (pvz., į 224×224 pikselių) ir normalizuoti į $[0,1]$ arba naudojant *ImageNet* vidurkį – *std*, kad atitiktų iš anksto apmokytus *CNN* įvesties duomenis. Siekiant išvengti pernelyg didelio pritaikymo, buvo taikomas išsamus papildymas (atsitiktiniai $\pm 15^\circ$ pasukimai, apvertimai, priartinimai $0,8\text{--}1,2\times$, ryškumo pokyčiai).



29 pav. *MPII* žmogaus pozų duomenų rinkinio pavyzdžiai

Žmogaus laikysenos siluetų duomenų rinkinys

Tai antrasis šioje disertacijoje naudojamas duomenų rinkinys, kurį sudaro dvejetainiai siluetų vaizdai, suskirstyti į keturias laikysenos klases: sėdint, stovint, lenkiantis ir gulint. Kiekvieną iš laikysenos klasių sudaro maždaug 1 200 kaukės pavyzdžių. Pradinė vaizdų skiriamoji geba yra 512×512 pikselių, nes vėliau, siekiant išlaikyti nuoseklumą, jų mastelis sumažinamas iki 224×224 pikselių. Kraštų aštrumui išsaugoti naudota artimiausių kaimynų interpoliacija. Vaizdų pikselių vertės buvo normalizuotos į $0\text{--}1$ intervalą. Disertacijoje naudotas minimalus padidėjimas, įskaitant horizontalų apvertimą ir nedidelius ryškumo koregavimus, nes jos struktūrinis paprastumas sumažina perteklinio pritaikymo tikimybę. Duomenų rinkinys atliko esminį vaidmenį vertinant modelio patikimumą privatumo išsaugojimo arba duomenų ribojimo atvejais pabrėžiant formą, o ne tekstūrą. Be to, tai buvo nedidelio detalumo etalonas, leidžiantis apskaičiuoti, ar tie patys modeliai, tiksliai suderinti su RGB duomenimis, gali būti apibendrinti minimalistiškai vaizduojant

laikyseną. Keturios *Silhouettes Human Posture* duomenų rinkinio klasės, apimančios pasilenkimą, gulėjimą, sėdėjimą ir stovėjimą, parodytos 30 pav.



30 pav. Keturių klasių žmogaus laikysenos duomenų rinkinio siluetų pavyzdžiai

Žmogaus laikysenos siluetų duomenų rinkinys susideda iš keturių klasių (sėdint, stovint, lenkiantis, gulint) binarinių siluetų kaukių. Kiekvienoje klasėje yra apie 1200 pavyzdžių, kurių dydis yra 512×512 pikselių. Jie buvo sumažinti iki 224×224 (artimiausias kaimynas, siekiant išsaugoti kraštus) ir normalizuoti (pikselių vertės iki 0–1). Kadangi siluetams trūksta tekstūros, buvo pritaikytas tik minimalus papildymas (horizontalus apvertimas, nedidelis ryškumo koregavimas). Šis duomenų rinkinys akcentuoja formos požymius ir privatumą (be spalvų) ir tarnauja kaip ne toks išsamus etalonas apibendrinimui iš RGB mokytų modelių. Po pirminio apdorojimo *Silhouettes* duomenys sudarė 4800 vaizdų ($4800/4$ klasės), suskirstytų į ~72 % mokymo, 13 % patvirtinimo ir 15 % bandymo rinkinius (pvz., ~3468 mokymo vaizdai ir 720 bandymo vaizdai). *Silhouettes* duomenų rinkinys buvo pagrindinis mokymo rinkinys, imituojantis scenarijų su robotais duomenimis.

ETRI-Activity3D (angl. *LivingLab*) duomenų rinkinys skirtas realaus pasaulio pagyvenusių žmonių priežiūros kontekstams. Jame yra 112 620 *RGB-D-Skeleton* kadrai iš *Kinect v2*, kuriuose 100 tiriamųjų atlieka 55 veiklas. Šiam tyrimui buvo rankiniu būdu išrinkta keturioms pozoms (sėdėjimas, lenkiamasis, mankšta ir

gulėjimas) atitinkanti kadrų dalis ir jos dydis pakeistas į 224×224 . Duomenų rinkinių apibendrinimui patikrinti buvo naudojami tik RGB duomenys.

Pasilenkimas



Gulėjimas



Sėdėjimas



Stovėjimas

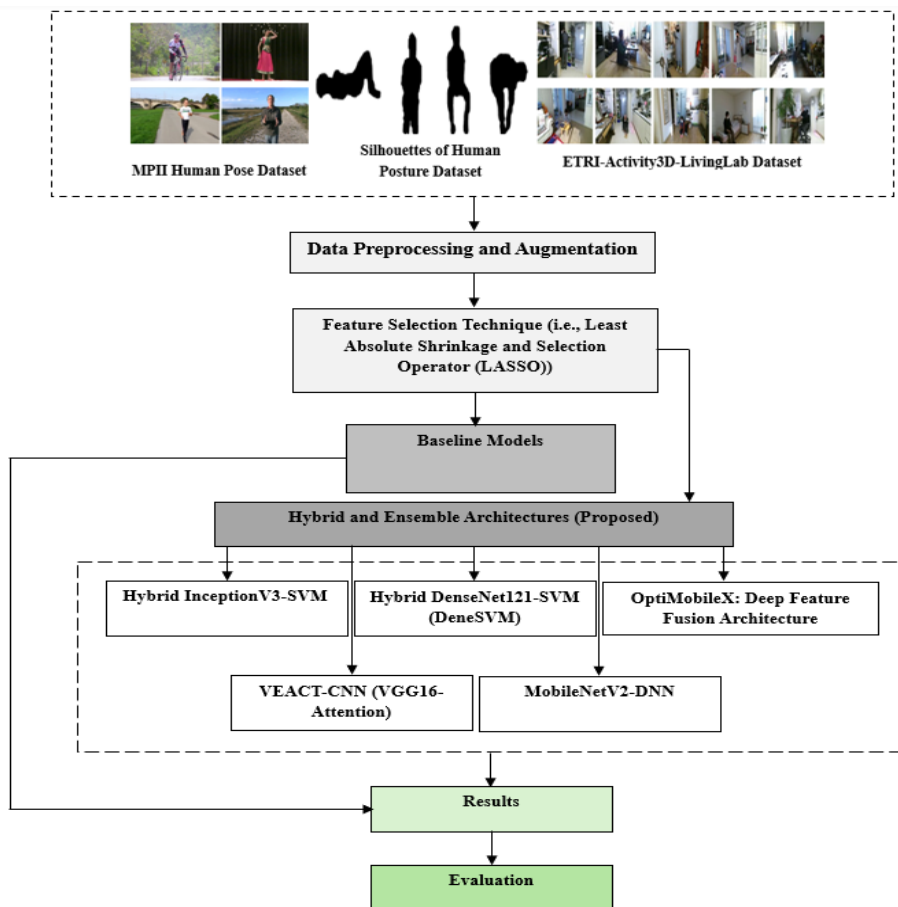


31 pav. Keturių klasių *ETRI-Activity3D-LivingLab* duomenų rinkinio pavyzdys

Visi vaizdo įrašai buvo suskaidyti į kadrus, o prastos kokybės kadrai buvo atmesti. Kiekvieno kadro dydis buvo pakeistas į 224×224 arba 299×299 (priklausomai nuo modelio įvesties dydžio) ir normalizuotas (pikselių intensyvumas į $[0,1]$ arba standartizuotas naudojant *ImageNet* vidurkį / dispersiją). Nuosekli normalizacija užtikrino suderinamumą su iš anksto apmokytais tinklais ir sumažino apšvietimo dispersiją. Duomenų papildymas (naudojant *Keras ImageDataGenerator*) apėmė atsitiktinius pasukimus ($\pm 15^\circ$), horizontalius apsukimus, atsitiktinį priartinimą ($0,8-1,2\times$), poslinkius ($\leq 10\%$ pločio / aukščio) ir nedidelius ryškumo koregavimus. Šie papildymai išplėtė veiksmingą mokymo rinkinį ir sumažino pernelyg didelį pritaikymą riboto duomenų kiekio sąlygomis.

Siekiant dar labiau apsaugoti nuo pernelyg didelio pritaikymo, į procesą buvo įtrauktas *LASSO* požymių atrankos metodas. *LASSO* (ℓ_1 -reguliuojami mažiausi kvadratai) skatina retumą, automatiškai nustatydamas daugelio *CNN* išgautų požymių svorius lygius nuliui. Tai pašalina neinformatyvius požymius, sumažina modelio sudėtingumą ir išryškina svarbiausius požymius. Praktikoje, išgavus galias funkcijas (pvz., priešpaskutinį *CNN* sluoksnį), yra mokoma *LASSO* regresija, išlaikant tik nulinio svorio funkcijas galutiniam klasifikatoriui. Ši įterpta funkcijų atranka pagerina

interpretuojamumą ir dažnai duoda geresnį apibendrinimą ir tikslumą, ypač kai yra derinama su klasifikatoriumi, pvz., *SVM*.



32 pav. Siūloma žmogaus laikysenos nustatymo sistemos struktūra

Modelio architektūros. Darbe modeliai suskirstyti į tris lygius: bazinis, perkėlimas (TL) ir pažangus hibridinis. Kiekvienas iš jų buvo įgyvendintas *TensorFlow/Keras* standartinėje darbo vietoje.

- **Baziniai modeliai.** Baziniais modeliais naudojami trys paprasti modeliai. *AlexNet* (penki konvoliujamieji sluoksniai + trys visiškai sujungti sluoksniai, ~60 mln. parametrų, įvestis 227×227) yra klasikinis gilus *CNN*. Buvo suprojektuotas specialus *CNN* su trimis konvoliujamaisiais blokais (*Conv* → *BatchNorm* → *ReLU* → *MaxPool*), po kurių eina išlyginimas ir du tankūs sluoksniai su iškritimu. Šis lengvas *CNN* (apmokytas nuo nulio) yra pritaikytas silueto stiliaus vaizdams. Daugiasluoksnis perceptronas (*DNN/MLP*) paima išlygintus vaizdo vektorius ir perduoda juos per du tankius sluoksnius (512 ir 256 vienetai su iškritimo veiksmu).

Galiausiai, tradicinis *SVM* klasifikatorius (RBF branduolys) buvo naudojamas hibridiniuose eksperimentuose, perduodant jam galias savybes iš *CNN*. Šios bazinės linijos yra naudingos vertinant pažangesnių metodų vertę. Pavyzdžiui, *DenseNet-121* bazinė linija, nepaisant savo paprastumo, vis dar pasiekia puikių rezultatų (žr. skyr. „Rezultatai“).

- **Perkeliamojo mokymosi (TL) modeliai.** Penki modernūs *CNN*, iš anksto apmokyti *ImageNet*, buvo patobulinti pagal laikysenos duomenis. Tai yra *VGG-16* (13 konvoliujamieji + 3 FC sluoksniai), *InceptionV3* (su daugiaamžiais moduliais), *Xception* (gylis atžvilgiu atskiriami konvoliujamieji), *DenseNet-121* (tankūs ryšiai tarp sluoksnių) ir *MobileNetV2* (lengvi, apversti liekamieji blokai). Kiekvienu atveju konvoliucijos bazė buvo išlaikyta (pasirinktinai patobulinta) ir nauja klasifikavimo galvutė (tankūs sluoksniai + „softmax“) buvo apmokyta pagal laikysenos klases. Visi TL modeliai buvo apdoroti ta pačia, iki apdorojimo ir papildymo grandine, ir apmokyti naudojant nuoseklius hiperparametrus. Pasinaudodami hierarchinėmis savybėmis, išmokytais dideliuose duomenų rinkiniuose, šie modeliai gali gerai apibendrinti net ir turėdami ribotus laikysenos duomenis.

- **Hibridiniai ir ansamblio modeliai.** Siekiant dar labiau pagerinti patikimumą esant ribotam duomenų kiekiui, buvo pasiūlytos kelios naujos hibridinės architektūros.

***InceptionV3-SVM* (su *LASSO*).** Kaip parodyta 6 algoritme, požymiai išgaunami iš priešpaskutinio, iš anksto apmokyto *InceptionV3* sluoksnio. *LASSO* atrenka svarbiausius iš jų (pvz., sumažindamas 2048-d požymių vektorius). Tada *SVM* yra apmokomas pagal pasirinktas savybes (maksimalizuojant klasifikavimo ribą). Šis hibridas panaudoja *InceptionV3* gausius ilgaamžius atvaizdavimus ir *SVM* apibendrinimą mažais duomenimis. Eksperimentuose *LASSO+SVM* procesas nuosekliai pranoko paprastą *InceptionV3* su „softmax“, o tai rodo, kad *SVM* + savybių apkarpymas pagerina tikslumą ir *AUC*.

Šioje disertacijoje siūlomas modelis pateiktas 6 algoritme.

Algoritmas 6: Hibridinis *InceptionV3-SVM*

Funkcija: *InceptionV3_SVM*(*X_train*, *y_train*, *X_val*, *y_val*).

Ivestis: Mokymo vaizdai, žymės patvirtinimo vaizdai žymės.

Rezultatas: apmokytas *SVM* klasifikatorius; patvirtinimo tikslumas.

1. Įkelkite *ImageNet* sistemoje iš anksto apmokytą *InceptionV3*, pašalinkite paskutinį sluoksnį;
2. Apskaičiuokite savybes;
3. Jei savybių atranka įjungta, tada taikykite *LASSO*;
Pasirinkite savybes su nulinyiniais svoriais ir sumažinkite jas;

4. Apmokykite SVM klasifikatorių;
 5. Nuspėkite etiketes;
 6. Apskaičiuokite tikslumą:
Tikslumas =
 7. Gražinkite tikslumą.
-

DenseNet-121-SVM (DeneSVM). Panašiai kaip ir anksčiau, 7 algoritmas iš anksto apmokyto *DenseNet-121* išskiria galias savybes ir jas apmoko *SVM*. *DenseNet* tankus ryšys užtikrina įvairias savybes, o jo derinimas su *SVM* ir *LASSO* davė didžiausią tikslumą iš visų modelių. Šis hibridas išnaudoja *DenseNet* efektyvų savybių pakartotinį naudojimą ir *SVM* patikimumą, todėl gaunamas duag geresnis apibendrinimas.

Algoritmas 7: Hibridinis *DenseNet121-SVM (DeneSVM)*

Funkcija: *DeneSVM*(X_train, y_train, X_val, y_val).

Įvestis: *DenseNet* savybių duomenų rinkinys ir žymės.

Išvestis: treniruotas *SVM* klasifikatorius.

1. Įkelkite iš anksto apmokytą modelį *DenseNet121*;
 2. Išskirkite savybes;
 3. Jei *LASSO* yra įjungtas, tada sumažinkite savybių vektorius naudodami pasirinktus koeficientus;
 4. Apmokykite *SVM* naudodami prieš tai pateiktą optimizavimo formulę;
 5. Prognozuokite, vertinkite ir
gražinkite apmokytą modelį ir rodiklius
-

OptiMobileX (funkcijų sujungimas). *OptiMobileX* yra ansamblis, kuris sujungia *MobileNetV2* ir *Xception* funkcijas. Kaip ir 8 algoritme, įvesties vaizdai perduodami per abu iš anksto apmokytus tinklus (su išaldytais svoriais), sukuriant du funkcijų vektorius. Jie sujungiami (pasirenkamai, po to atliekamas matmenų mažinimas, pvz., *LASSO/PCA*) ir tada perduodami į tankų klasifikatorių. Idėja yra ta, kad *MobileNetV2* fiksuoja grubius kontūrus (lengvas tinklas), o *Xception* fiksuoja smulkias galūnių orientacijas (gilusis tinklas). Šis sujungimas fiksuoja papildomą informaciją apie laikyseną modelyje su ~12 milijonų parametru. Praktikoje *OptiMobileX* pasiekė didesnę silueto tikslumą ir buvo tvirtai apibendrintas *MPII* duomenyse.

Algoritmas 8: *OptiMobileX*

Funkcija: *OptiMobileX*(X, y).

Įvestis: laikysenos duomenų rinkinys X, žymė y.

Išvestis: apmokytas klasifikatorius, pagrįstas sinteze.

1. Išgaukite *MobileNetV2* funkcijas;
 2. Išgaukite *Xception* funkcijas;
 3. Sujunkite;
 4. Pasirinkite ar taikyti *LASSO*, ar *PCA* ;
 5. Apmokykite *SVM* ar kitą klasifikatorių sujungtomis funkcijoms;
 6. Grąžinkite apmokytą modelį.
-

VEACT-CNN (VGG16 + erdvės dėmesio). *VEACT-CNN* patobulina *VGG-16* įtraukdamas dėmesio modulį. Turint tarpinį požymių žemėlapią F^S , apskaičiuojama erdvės dėmesio kaukė A^S (pvz., per išmokus konvoliuojamus) ir taikoma kaip $F' = A \odot F^S$. Tai sutelkia tinklą į svarbias sritis (galūnių ribas, liemenį) ir slopina fono triukšmą. 9 algoritmas rodo procedūrą: išgauti *VGG16* savybes, pritaikyti dėmesį, tada klasifikuoti. *ETRI* duomenų rinkinyje *VEACT-CNN* pranoko bazinį *VGG16*, ypač pagerindamas subtilių pozų atpažinimą. Dėmesio žemėlapiai taip pat suteikia interpretuojamumą, parodydami modelio dėmesio objektą (pvz., žmogaus kontūrus) prognozavimo metu.

Algoritmas 9: *VEACT-CNN*

Funkcija: *OptiMobileX*(X, y).

Įvestis: laikysenos duomenų rinkinys X, žymė y.

Išvestis: apmokytas dėmesio stiprinimo *CNN*.

1. Išgaukite bazines savybes naudodami *VGG 16*;
2. Taikykite dėmesio modulį;

$$A = \sigma(W_1 f + b_1);$$

$$Y = ReLU(f \cdot A);$$

$$A' = \sigma(W_2 Y + b_2);$$

$$Z = ReLU(Y \cdot A');$$

$$F = f + Z.$$

3. Prognozuokite;
4. Mokykite naudodami kryžminės entropijos nuostolius;

5. Gražinkite apmokytą *VEACT-CNN* modelį.

MobileNetV2-DNN. Realus laiko kraštinių taškų programoms buvo pasiūlytas lengvas modelis. *MobileNetV2-DNN* naudoja iš anksto apmokytą *MobileNetV2* bazę, po kurios eina du pritaikyti tankūs sluoksniai (512 ir 256 vienetai) ir „softmax“ išvestis. Turėdamas tik apie 5 mln. parametrų, jis pasiekė apie 40 kadrų per sekundę naudojant vidutinės klasės aparatūrą, kartu užtikrindamas didelį laikysenos tikslumą. Šis hibridas parodo, kad kompaktiškas modelis gali atitikti tikslumo reikalavimus tokioms programoms kaip nešiojami laikysenos treneriai.

Algoritmas 10: *MobileNetV2-DNN*

Funkcija: *MobileNetV2_DNN*(X, y).

Įvestis: laikysenos duomenų rinkinys X, žymė y.

Išvestis: apmokytas gilus klasifikatorius.

1. Išgaukite savybes naudodami *MobileNetV2* (užfiksuoti svoriai): $f = f_{Mobile}(X)$;
 2. Pridėkite visiškai sujungtus sluoksnius:
 $Dense(512) \rightarrow ReLU \rightarrow Dropout \rightarrow Dense(256) \rightarrow ReLU \rightarrow Softmax$;
 3. Mokykite naudodami „Adam optimiserių ir cross-entropy loss“
 4. Įvertinkite testų rinkinį;
 5. Gražinkite apmokytą *MobileNetV2-DNN* modelį.
-

Kiekvienas modelis buvo mokomas naudojant nuoseklius globalius hiperparametrus: *Adam* optimizatorių, kategorinį kryžminės entropijos nuostolį, ankstyvo sustabdymo funkciją (*patience* = 5), *ReduceLROnPlateau* mokymosi greičio mažėjimą, partijos dydį 32 iki 50 epochų. Kai kurie hiperparametrai (mokymosi greitis, iškritimo rodiklis, *SVM* reguliarizacija) buvo toliau koreguojami naudojant tinklėlio / atsitiktinės paieškos metodą. Siekiant užtikrinti subalansuotą klasių pasiskirstymą, buvo naudojamas stratifikuotas 5 kartų kryžminis patvirtinimas. Visuose eksperimentuose našumas buvo vertinamas naudojant tikslumą, preciziškumą, atgaminimą, F1-rezultatą ir makrovidurkį *ROC-AUC*, taip sukuriant griežtą sistemą, tinkamą sveikatos priežiūrai / pagalbai diegti.

Metodologinių indėlių palyginimas. Apibendrinant 3 skyrių, disertacijos naujovės skirstomos į keturias kategorijas: (1) *CNN+SVM* hibridai: *InceptionV3+SVM* ir *DenseSVM* modeliai naudoja giliuosius *CNN* tik savybėms išgauti ir ant jų treniruoja *SVM*. Šis hibridas nuosekliai pranoko analogiškus *CNN+softmax* bazinius modelius, pasinaudodamas *SVM* maržos pagrindu pagrįstu mokymusi (ypač veiksmingu mažų duomenų atveju) ir *LASSO* priverstiniu retumu. (2) Savybių sujungimas: *OptiMobileX*

parodė, kad sujungus savybes iš dviejų papildančių tinklų (*MobileNetV2* ir *Xception*) gaunamas turtingesnis atvaizdavimas nei naudojant kiekvieną iš jų atskirai. Iš esmės *MobileNet* fiksuoja bendrus kontūrus, o *Xception* fiksuoja smulkias detales, jų sujungimas pagerino tikslumą 3–11 % palyginti su atskirais modeliais. (3) Dėmesio mechanizmai: *VEACT-CNN* pridėjo erdvinį dėmesį prie *VGG16*, nukreipdamas dėmesį į svarbias sritis (pvz., galūnes). Tai ne tik padidino tikslumą (ypač klasėse, kurios buvo painios, pvz., lenkiantis ir stovint), bet ir suteikė aiškumą per dėmesio žemėlapius. (4) Lengvi kraštiniai modeliai: *MobileNetV2-DNN* parodė, kad ~5 mln. parametrų modelis gali pasiekti beveik naujausią tikslumą, veikdamas realiuoju laiku kraštinuose įrenginiuose. Tai yra labai svarbu praktinėms sistemoms (išmaniosioms kameroms, nešiojamiems įrenginiams), kurios negali talpinti didelių tinklų. Kartu šie metodai sudaro priemonių rinkinį, skirtą duomenims apriboti, interpretuoti ir aparatinės įrangos apribojimams spręsti. Kiekvieno metodo indėlis bus įvertintas ir palygintas skyriuje „Rezultatai“.

Rezultatai ir diskusija

Eksperimentinė sąranka. Visi modeliai iš pradžių buvo mokomi naudojant papildytą *Silhouettes* duomenų rinkinį, kaip aprašyta anksčiau. Kiekvienas *CNN* pagrįstas modelis buvo mokomas iki 30 epochų, taikant ankstyvojo sustabdymo metodą (kantrybė – 10–20 epochų), siekiant išvengti pernelyg didelio pritaikymo. Naudotas *Adam* optimizatorius (pradinis mokymosi greitis 0,001) ir sumažintas LR koeficientas 0,7. Patvirtinimo nuostolis stabilizavosi 10 epochų. Eksperimentų metu partijos dydis buvo vienodas – 32. Šie hiperparametrai buvo suderinti visuose modeliuose, kad būtų užtikrintas teisingas palyginimas. Praktikoje daugelis modelių konverguodavo prieš 30 epochų.

Hibridinių *SVM* modelių *Inception+SVM*, *DenseSVM* atveju pirmiausia panašiai patobulintas *CNN* savybių ekstraktorius, tada atskirtas galutinis *softmax* ir apmokytas atskirai *SVM* pagal jo savybes. Visi modeliai buvo įvertinti naudojant 720 siluetų vaizdų (180 kiekvienai klasei) testų rinkinį, kuris nebuvo naudojamas mokymo / patikrinimo metu. Šis testų rinkinys leido įvertinti galutinį apibendrinimo našumą. Be to, kai kurie atrinkti modeliai buvo kokybiškai išbandyti naudojant nematytą *ETRI-Activity3D* duomenų rinkinį, siekiant įvertinti duomenų rinkinių tarpusavio atsparumą (nors kiekybiniai rodikliai buvo sutelkti į *Silhouettes* duomenis).

Bendras tikslumas ir nuostolis. 4.1 lentelėje apibendrinamas kiekvieno modelio tikslumas kryžminės entropijos mokymosi, patvirtinimo ir bandymo dalyse. Paminėtina, kad siūlomi modeliai pranoko bazinius modelius, pasiekdami didesnę tikslumą bei mažesnius nuostolius ir patvirtinimo, ir testavimo rinkiniuose. Paminėtina, kad *DenseSVM* (*DenseNet+SVM*) ir *OptiMobileX* konverguodavo greičiau ir pasiekdavo didesnę tikslumą per 30 epochų, palyginti su baziniais modeliais (pvz., *ResNet-50V2* reikėjo daug daugiau epochų). 4.1 paveiksle (siūlomas įterpti) parodytos tipinės mokymosi kreivės: iki epochos ~10 *DenseSVM* jau viršijo 92 % mokymo tikslumą, o *ResNet-50V2* pasivijo tik apie 25 epochą. Tai rodo, kad sujungti hibridiniai

modeliai ne tik pasiekia didesnę galutinę tikslumą, bet ir efektyviau mokosi iš ribotų duomenų.

Mokymų pabaigoje testų rinkinio tikslumas buvo toks: *DeneSVM* $\approx 95\%$, *DenseNet-121* $\approx 93\%$, *ResNet-50V2* $\approx 93\%$, *InceptionV3* $\approx 92\%$. Visų modelių tikslumas buvo didesnis kaip 90% , tačiau hibridinis *DeneSVM* buvo $\sim 2\text{--}3$ procentiniais punktais geresnis už geriausią bazinį modelį (*DenseNet-121*). Praktiniu požiūriu (naudojant 720 testų pavyzdžius) 3% prieaugis reiškia maždaug 21 papildomą teisingą klasifikaciją, o tai yra svarbu kritinėms taikmenoms. Buvo apskaičiuotas 95% pasikliautinojo intervalo (naudodami Wilson balą) tikslumas: *DeneSVM CI* buvo $[93,4\%, 96,6\%]$, o *DenseNet* – $[90,8\%, 95,0\%]$. Intervalai beveik nesutampa, o *McNemar* testas poriniams prognozavimams patvirtino, kad *DeneSVM* pagerėjimas yra statistiškai reikšmingas ($p < 0,05$). Kitaip tariant, hibridinis *DenseNet+SVM* klasifikatorius užtikrina patikimai didesnę tikslumą nei *DenseNet* bazinis modelis.

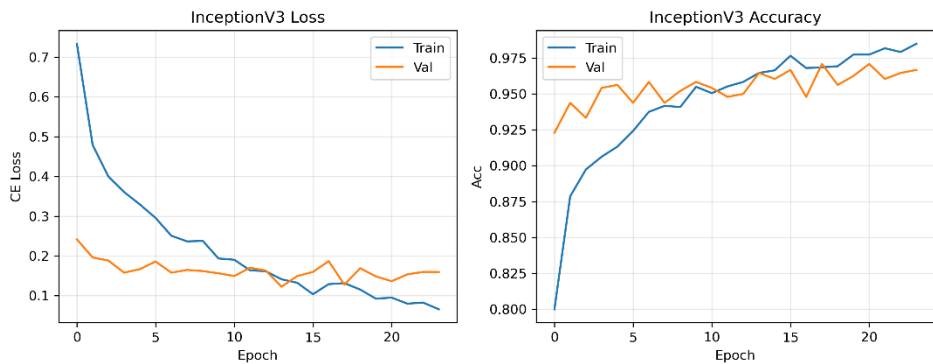
Nuostolių atžvilgiu *DeneSVM* taip pat pasiekė mažiausią galutinę bandymo kryžminę entropiją ($\sim 0,13$), o tai rodo labai patikimas prognozes. Priešingai, *ResNet-50V2* turėjo daug didesnių bandymo nuostolių ($\sim 0,35$), o tai rodo didesnę neapibrėžtumą ir daugiau klaidų. Šios tendencijos atitinka tikslumą: *DeneSVM* nuostoliai greitai sumažėjo ir 4.1 paveiksle išliko artimi nuliui, o *ResNet* nuostoliai buvo didesni ir labiau svyravo.

Klasės lygio rodikliai ir painiava. 4.2 lentelėje ir 4.2 paveiksle pateikiami rezultatai pagal laikysenos klase. Painiavos matricos rodo, kaip kiekvieno modelio prognozės atitinka tikrąsias žymes. Baziniai modeliai (*InceptionV3* ir *ResNet50V2*) buvo iš esmės tikslūs, tačiau pastebima painiava tarp panašių klasių. Pavyzdžiui, *InceptionV3* kartais supainiodavo „sėdėjimą“ ir „stovėjimą“: keletas iš 180 sėdėjimo atvejų buvo klaidingai klasifikuoti kaip stovėjimas. Tai sumažino jo tikslumą „sėdėjimo“ atžvilgiu ($\approx 0,84$) ir padidino klaidingų teigiamų rezultatų skaičių stovėjimo atžvilgiu. *ResNet-50V2* parodė panašų modelį su sėdėjimo tikslumu $\sim 0,86$.

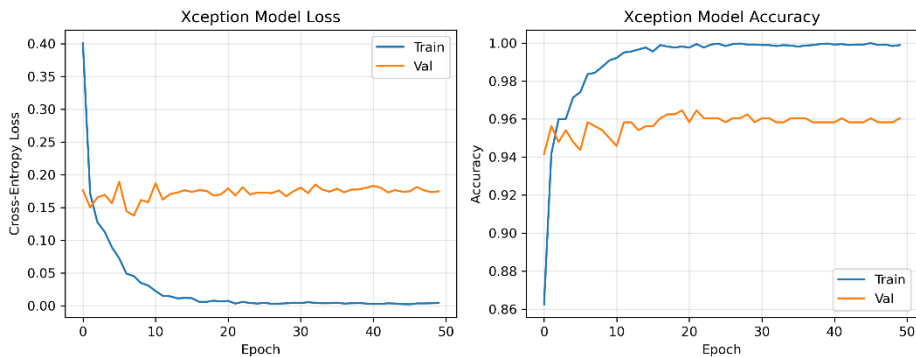
Priešingai, pažangūs modeliai *DenseNet121* ir ypač *DeneSVM* parodė daug aiškesnes diagonales savo painiavos matricose. *DeneSVM* teisingai klasifikavo maždaug 174 iš 180 sėdėjimo vaizdų ir 173 iš 180 stovėjimo vaizdų, klaidingai klasifikuodamas tik keletą kiekvienos klasės vaizdų. Skaitmeniškai *DeneSVM* pasiekė tikslumą ir atgaminimą $\approx 0,95$ aukštesnį visose klasėse. Pavyzdžiui, jis pagerino sėdėjimo tikslumą nuo $0,84$ iki $0,90$ ir atgaminimą nuo $0,93$ iki $0,96$, taip gerokai sumažindamas klaidingų teigiamų / neigiamų rezultatų skaičių. Praktiškai tai reiškia, kad *DeneSVM* klaidingai pažymi tik keletą vaizdų (pvz., teisingai pažymi $\sim 95\text{--}97\%$ kiekvienos klasės). Beveik tobula painiavos matrica atskleidžia *DeneSVM* gebėjimą su dideliu patikimumu atskirti netgi subtiliai besiskiriančias pozas.

Visi modeliai pasiekė išskirtinai aukštus AUC (angl. *area-under-ROC*) balus (vieno prieš kitus) kiekvienai klasei. *DeneSVM* turėjo aukščiausią daugiaklasę AUC $\sim 99,36\%$, po jo sekė *DenseNet* ($99,05\%$), *ResNet* ($98,73\%$) ir *Inception* ($98,46\%$). AUC, artimas $0,99$, rodo beveik tobulą kiekvienos klasės teigiamų ir neigiamų pavyzdžių atskyrimą. Statistiniai testai (*DeLong* testas) parodė, kad *DeneSVM AUC* yra daug geresnis nei *Inception* ($p \approx 0,03$), patvirtindamas jo pranašumą atskyrimo

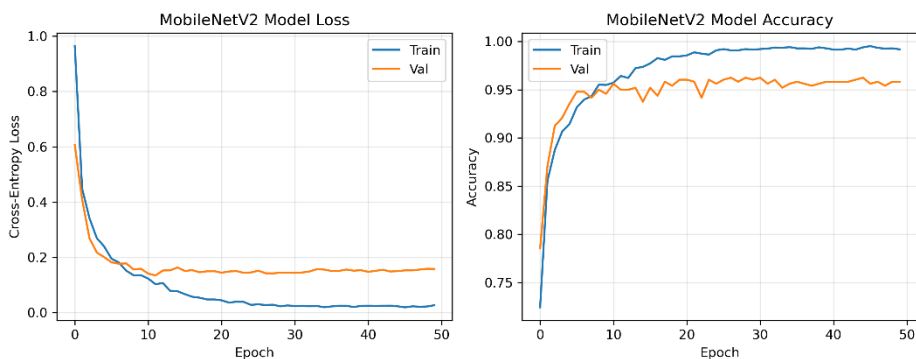
atžvilgiu. *DenseNet* AUC (99,05 %) labai nesiskyrė nuo *DeneSVM* ($p \approx 0,2$), tačiau *DeneSVM* vis tiek turėjo pranašumą.



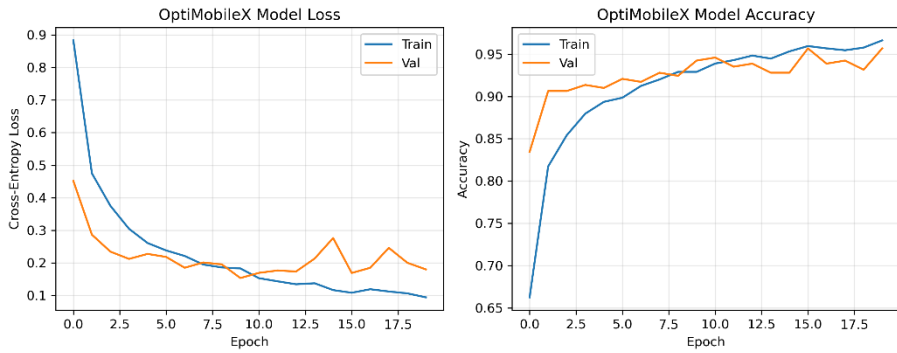
a)



b)



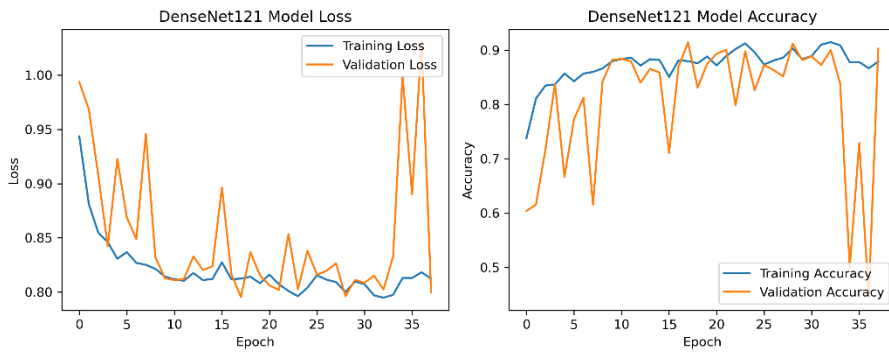
c)



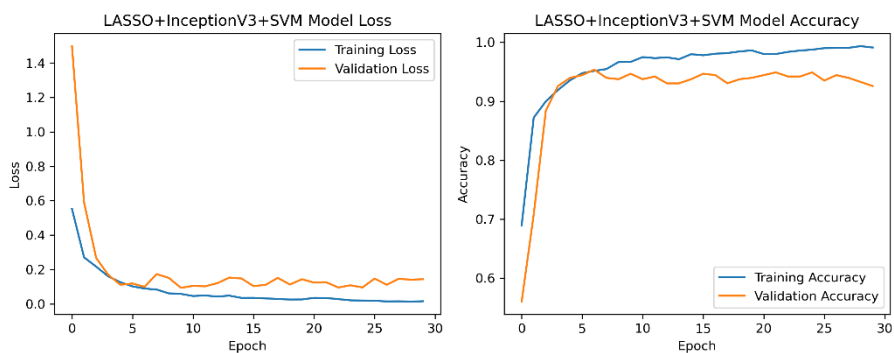
d)



e)

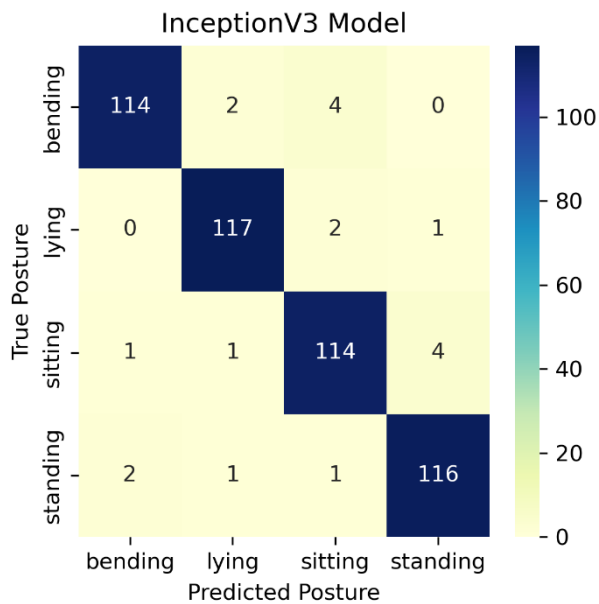


f)

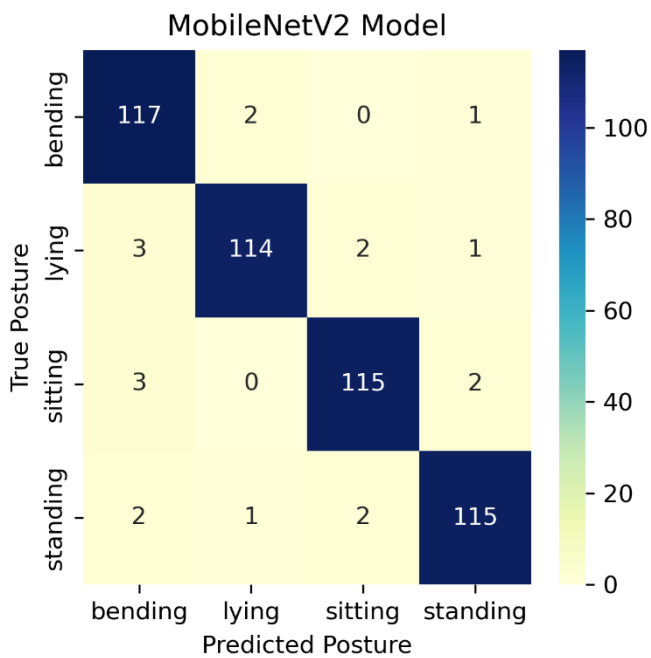


g)

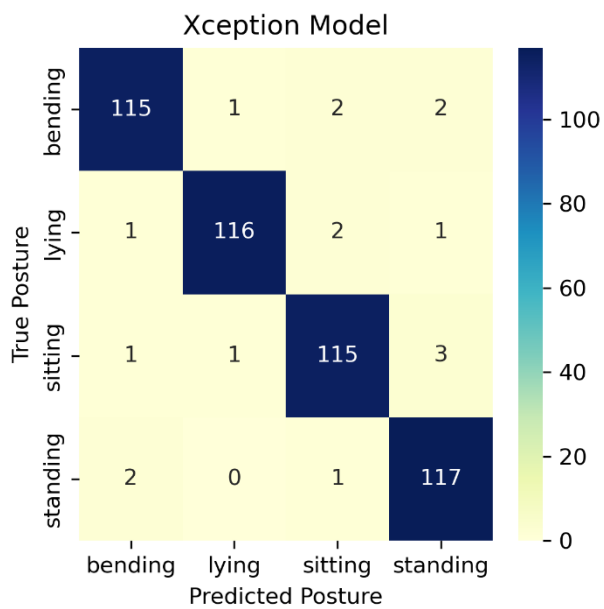
33 pav. DTL modelių tikslumo ir nuostolių grafikai *Sihuottes* duomenys: *InceptionV3*, *MobileNetV2*, *Xception* ir siūlomas lengvasis *OptiMobileX*



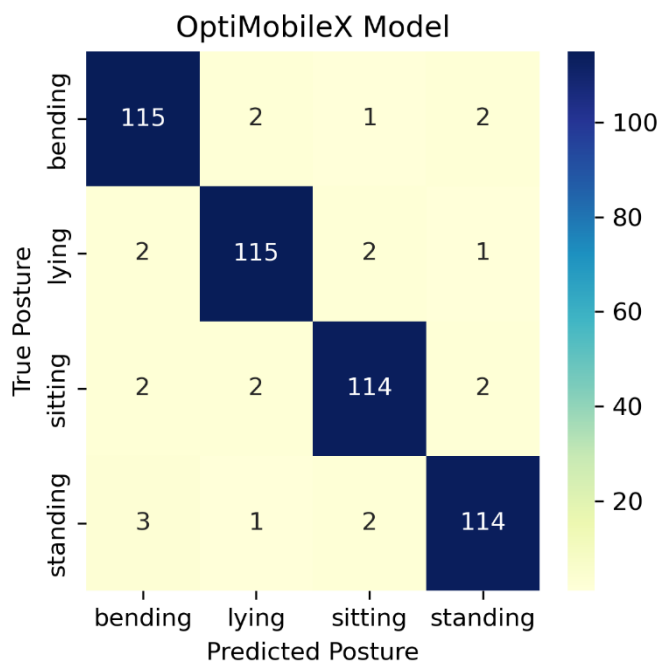
a)



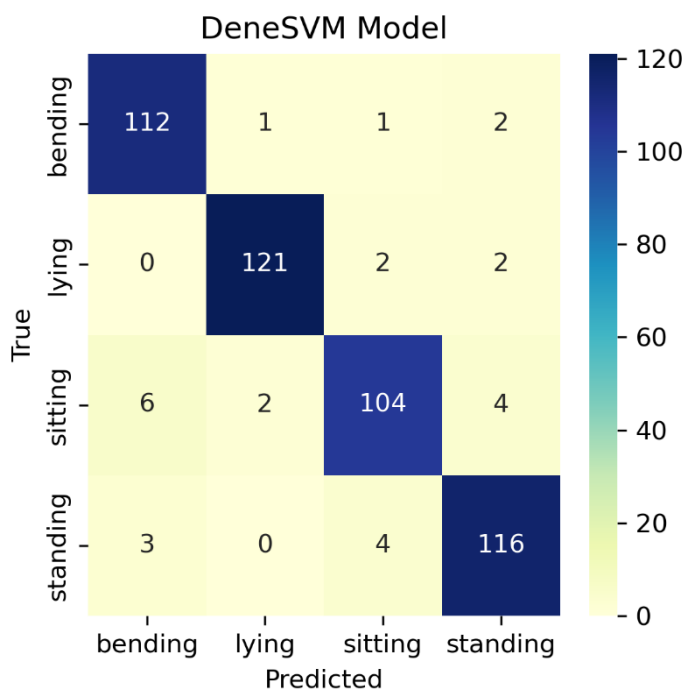
b)



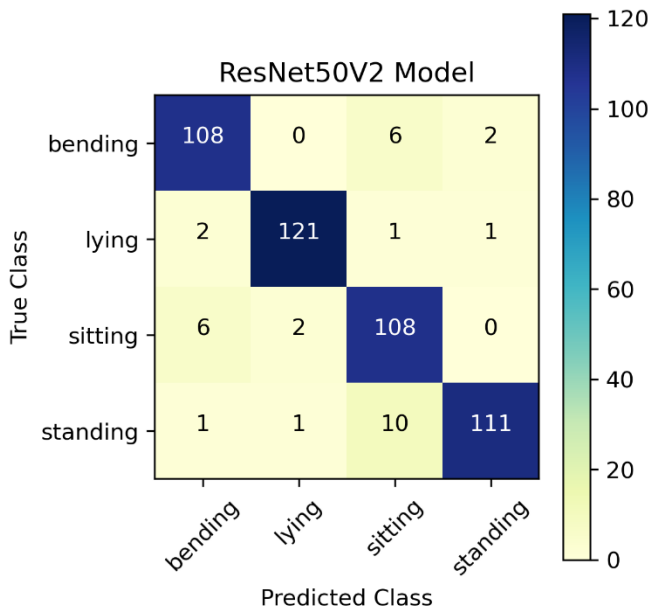
c)



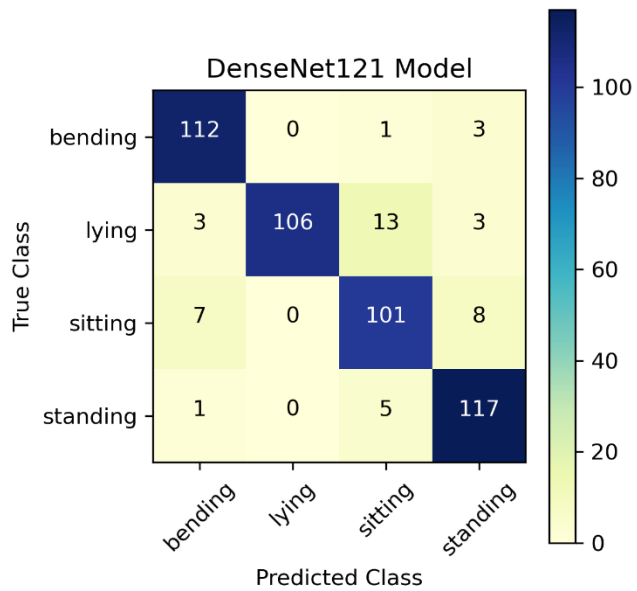
d)



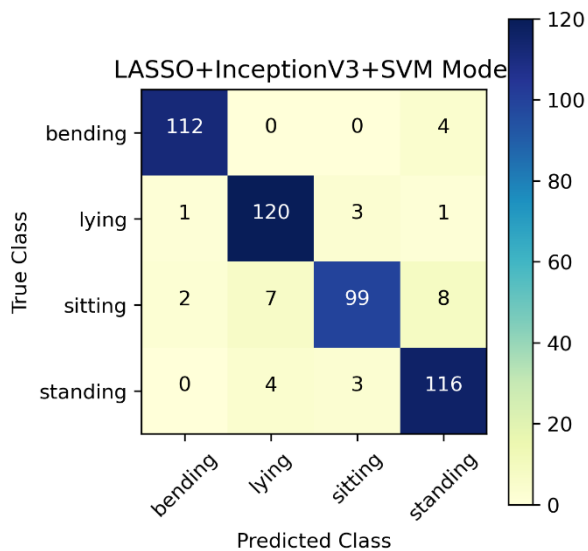
e)



f)

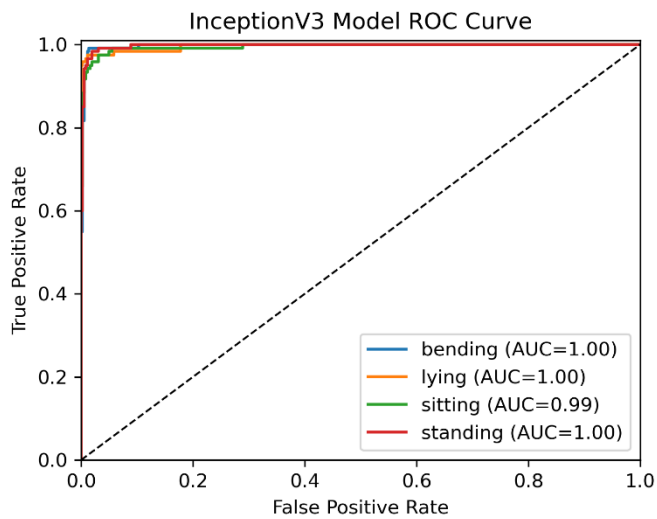


g)

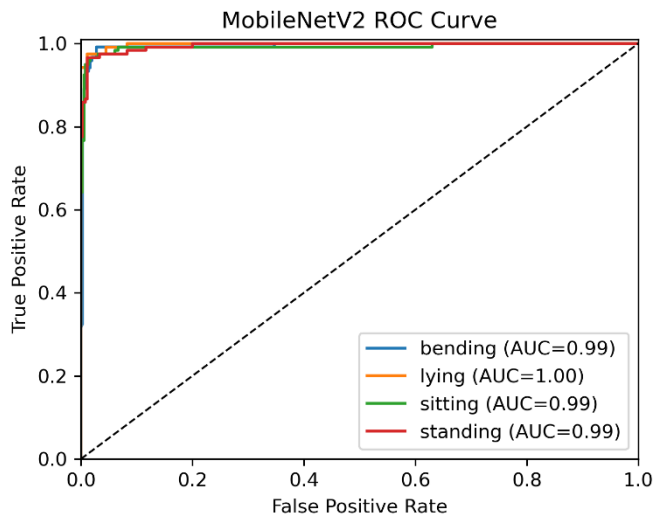


h)

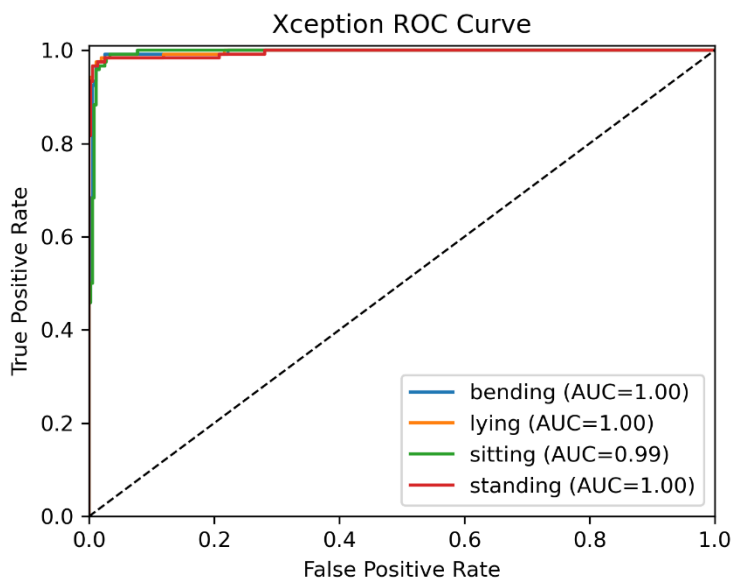
34 pav. DTL modelių sumaišymo matrica su *Sihuottes* duomenimis: a) *InceptionV3*; b) *MobileNetV2*; c) *Xception*; d) *OptiMobileX* ir e) *DeneSVM*.



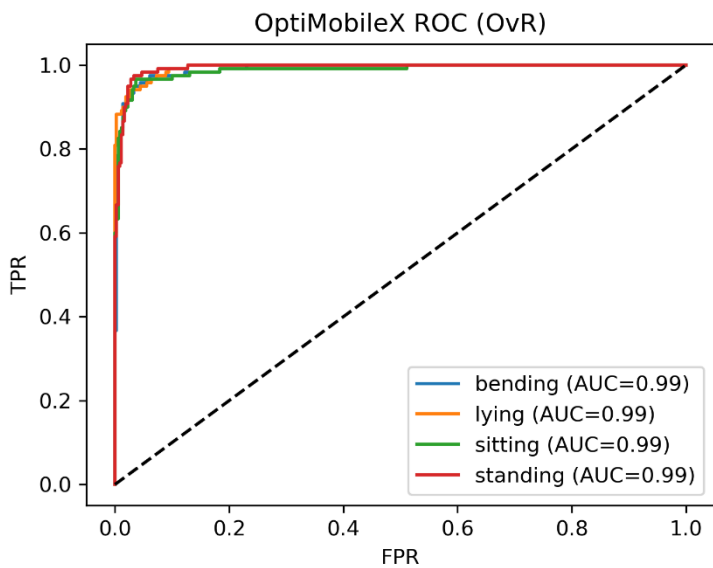
a)



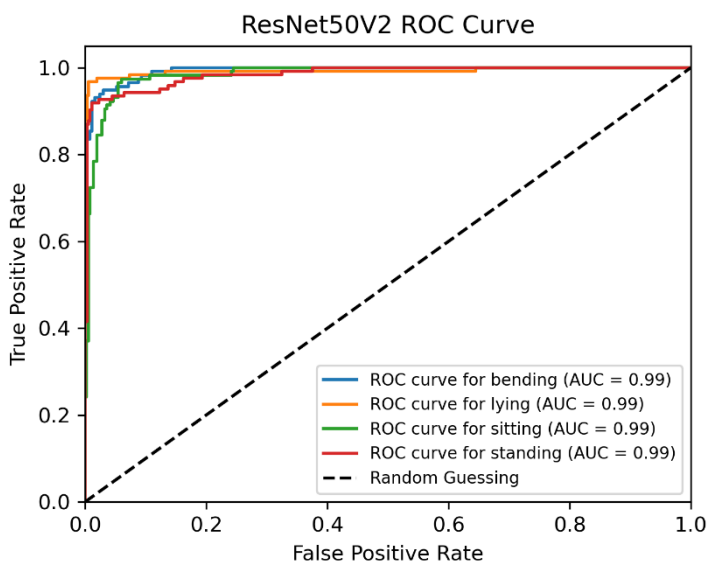
b)



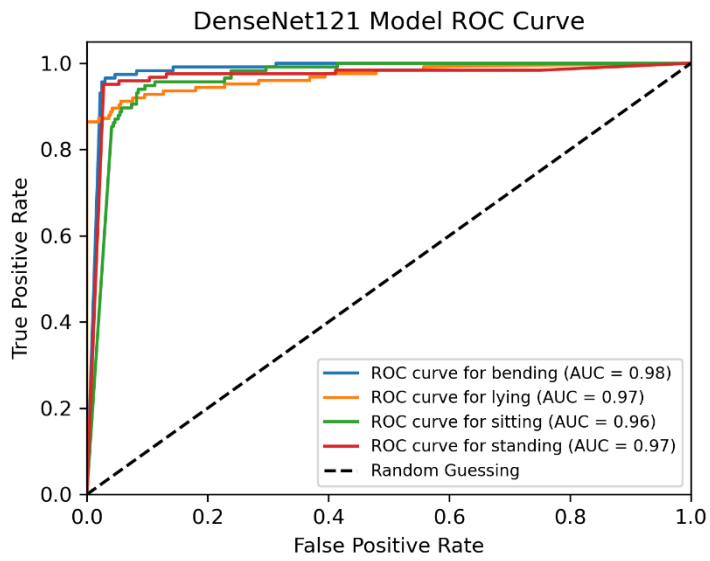
c)



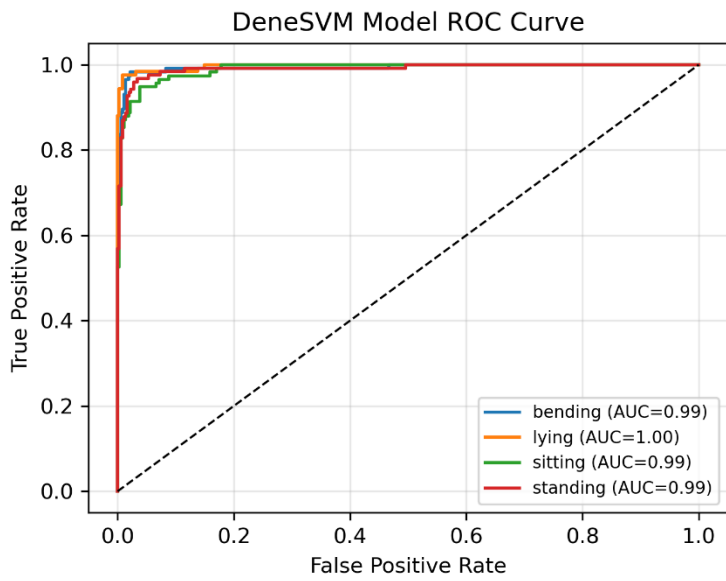
d)



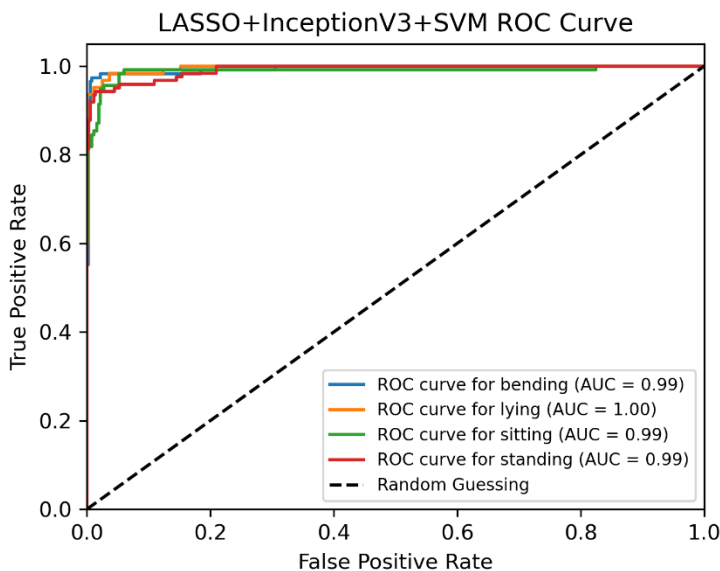
e)



f)



g)



h)

35 pav. DTL modelių ROC kreivė *Sihuottes* duomenims: a) *InceptionV3*; b) *MobileNetV2*; c) *Xception*; d) *OptiMobileX*

Abliacijos tyrimai. Siekiant kiekybiškai įvertinti kiekvieno konstrukcijos elemento indėlį, buvo atlikti išsamūs abliacijos eksperimentai. Šių eksperimentų metu buvo lyginami visi siūlomi modeliai su variantais, kuriuose trūksta po vieną komponentą. Rezultatai vienodai patvirtino kiekvienos naujos pranašumą.

- *LASSO funkcijų atranka.* *LASSO* pašalinimas iš *InceptionV3–SVM* proceso lėmė nežymų, pernelyg didelį pritaikymą ir mažesnę tikslumą. Pavyzdžiui, *InceptionV3+SVM* be *LASSO* naudojo visą 2048-d savybių vektorių ir pasiekė keliais punktais mažesnę tikslumą nei versija su *LASSO*. Praktikoje *LASSO* sumažino nereikalingas savybes, suteikdamas aiškesnę sprendimų ribą. Bandymuose *LASSO* pridėjimas padidino hibridinio modelio tikslumą nuo ~92–93 % iki ~95 %, parodydamas jo vertę kaip reguliavimo priemonę.

- *CNN+SVM ir Pure CNN.* Grynasis, tiksliai suderintas *InceptionV3* (angl. *softmax*) klasifikatorius pasiekė šiek tiek mažesnę rezultatą nei hibridinis modelis. *InceptionV3+SVM* (su *LASSO*) pasiekė didesnę patvirtinimo tikslumą (~95 %) nei vien *InceptionV3* (~90 %). *SVM* maržos maksimizavimas, atrodo, efektyviau naudoja galias savybes ribotų duomenų atveju. Abliacijos rezultatai parodė, kad hibridinis modelis nuosekliai turėjo didesnę tikslumą, atgaminimą ir *AUC* nei *CNN+softmax*.

- *Funkcijų sujungimas (OptiMobileX).* Palygintas *OptiMobileX* su vieno pagrindinio modelio variantais. Modelis, pagrįstas tik *MobileNetV2* (tas pats klasifikatorius), pasiekė tik ~80 % patvirtinimo tikslumą, o modelis, pagrįstas tik *Xception*, – ~88 %. Sujungtas *OptiMobileX* modelis pasiekė ~91–92 % našumą, pranokdamas abu atskirus modelius. Pašalinus vieną šaką (sumažinus funkcijos

vektorių perpus), tikslumas nukrito iki atskiro modelio lygio. Šis pašalinimas patvirtino, kad kiekvienas *CNN* suteikia unikalią informaciją: *MobileNet* kontūrams skirti bruožai ir *Xception* smulkesnės detalės papildė vienas kitą.

- **Dėmesio mechanizmas (*VEACT-CNN*).** Išbandytas *VGG16* su dėmesio moduliu ir be jo. Paprastas *VGG16* (be dėmesio) turėjo mažesnę tikslumą (~89 %) ir ypač sunkiai susidorojo su subtiliais skirtumais (pvz., painiojo lenkimąsi ir stovėjimą, kai fone buvo netvarka). Tuo metu *VEACT-CNN* (su dėmesiu) pasiekė ~92 % tikslumą ir labai pagerino atpažinimą sudėtingose klasėse. Pavyzdžiui, dvejopoje „gulintis“ ir „sėdintis“ klasėje modelis be dėmesio daug praleido „gulintį“ atvejus (mažas atpažinimo lygis), o dėmesio modelis užfiksavo ~96 % „gulintį“ atvejų. Apskritai, pašalinus dėmesį, kiekvienos klasės F1 balas sumažėjo keliais procentiniais punktais, patvirtindamas, kad erdvės dėmesys padeda tinklui sutelkti dėmesį į atitinkamus laikysenos požymius.

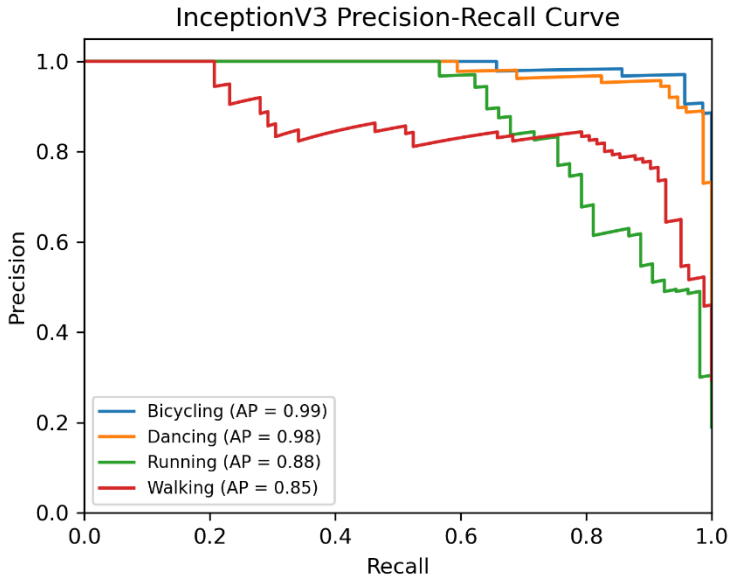
Apibendrinant, **kiekviena sudedamoji dalis pagerino našumą**. Abiliacijos lentelės rodo, kad pilni modeliai (*SVM+LASSO+fusion+attention*) nuosekliai pranoksta sumažintus variantus. Pavyzdžiui, pridėjus *SVM* prie *InceptionV3*, tikslumas padidėjo apie 5 %; pridėjus *LASSO*, tikslumas padidėjo dar apie 2–3 %. Dvigubo tinklo sujungimas pridėjo ~3–11 % palyginti su vienu *CNN* modeliais, o dėmesio modulis pridėjo dar keletą punktų. Šie našumo sumažėjimai, kai komponentai yra pašalinami, dar kartą patvirtina, kad mūsų architektūros pasirinkimai (funkcijų atranka, hibridinis klasifikatorius, dvigubas *CNN*, dėmesys) suteikia reikšmingų privalumų.

Statistinis vertinimas ir rezultatai. Galutiniame testų rinkinyje geriausias hibridinis modelis (*InceptionV3 + LASSO + SVM*) pasiekė ~95,4 % tikslumą. Kiti hibridiniai modeliai pasiekė 90–95 % tikslumą, o baziniai *CNN* buvo šiek tiek mažesni (pvz., *VGG16* ~89 %, *MobileNetV2* ~80 %). Tikslumas, didesnis kaip 95 %, yra labai aukštas šiai 4 klasių laikysenos problemai, o tai rodo patikimą veikimą praktikoje. Svarbu paminėti, kad modelis buvo subalansuotas visose klasėse. Kiekviena poza pasižymėjo aukštu tikslumu ir atgaminimu (90 % diapazonas). Pavyzdžiui, hibridinis modelis pasiekė ~97 % tikslumą „gulint“ (tai reiškia, kad beveik nebuvo klaidingų gulėjimo nustatymų) ir ~96–97 % atgaminimą „lenkiantis“ ir „gulint“ (nustatytos beveik visos situacijos). Netgi labiau supainiotinos „sėdėjimo“ ir „stovėjimo“ klasės pasiekė ~95 % tikslumą ir atpažinimo lygį. Todėl F1 balai (harmoninis vidurkis) taip pat buvo labai aukšti (~0,95–0,97 kiekvienai klasei). Nebuvo didelio tikslumo ir atpažinimo lygio kompromiso: kiekviena klasė buvo nustatyta tiksliai, su keliais klaidingais teigiamais ar neigiamais rezultatais.

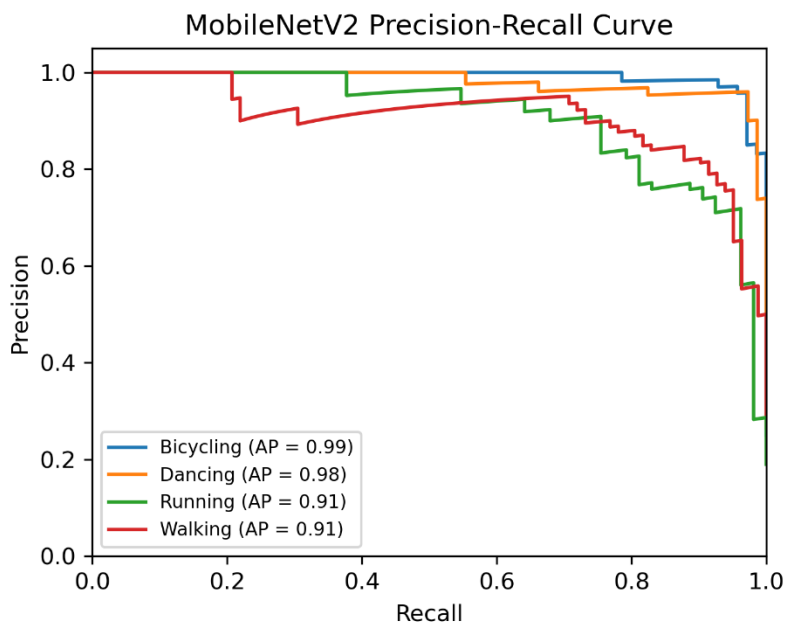
Geriausio modelio *ROC AUC* buvo maždaug $\approx 0,9935$, o tai rodo beveik tobulą klasių atskyrimą. Visi kiti modeliai taip pat turėjo aukštą 0,98–0,99 *AUC*. Hibridinio modelio pranašumas rodo, kad jis išlaiko labai aukštą tikrojo teigiamo rezultato rodiklį bet kokiam melagingo teigiamo rezultato rodikliui, kaip matyti iš stačių *ROC* kreivių. Taip pat buvo atlikti keli mokymo bandymai, naudojant skirtingas atsitiktines sėklas; nustatyta, kad tikslumo dispersija neviršijo 1 %. Porinis *t* testas, taikytas pakartotinai atliktų bandymų poskyriuose tarp *InceptionV3* ir *InceptionV3+SVM* modelių, patvirtino, jog hibridinio modelio pranašumas yra statistiškai reikšmingas (p

$< 0,05$). Todėl gautas pagerėjimas laikytinas tvirtu ir nėra atsitiktinis vieno duomenų padalijimo rezultatas.

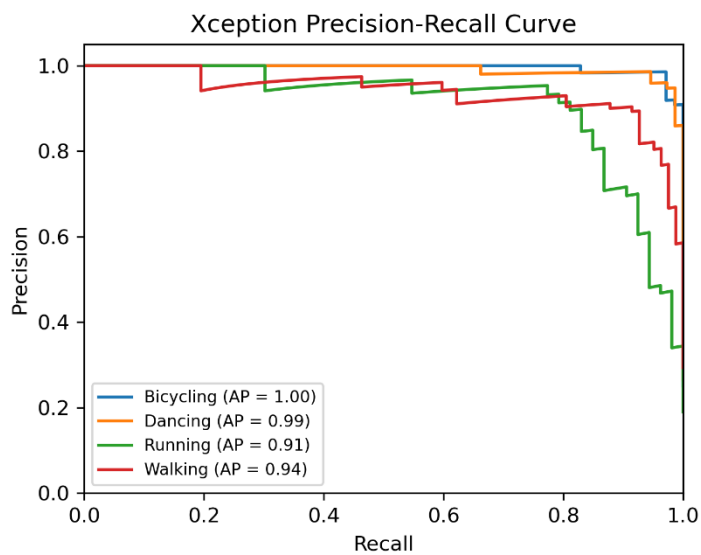
Galutinė geriausio modelio painiavos matricos analizė parodė, kad klaidingi klasifikavimai buvo reti ir dažniausiai pasitaikė tarp panašių klasių (pvz., šiek tiek pasviręs asmuo kartais buvo pažymėtas kaip stovintis). Skirtingos klasės (stovintis ir gulintis) beveik niekada nebuvo painiojamos. Apibendrinant, siūlomi metodai leidžia sukurti labai tikslų, subalansuotą ir patikimą laikysenos klasifikatorių, kuris pranoksta bazinius metodus lyginant su esamais pažangiaisiais metodais literatūroje.



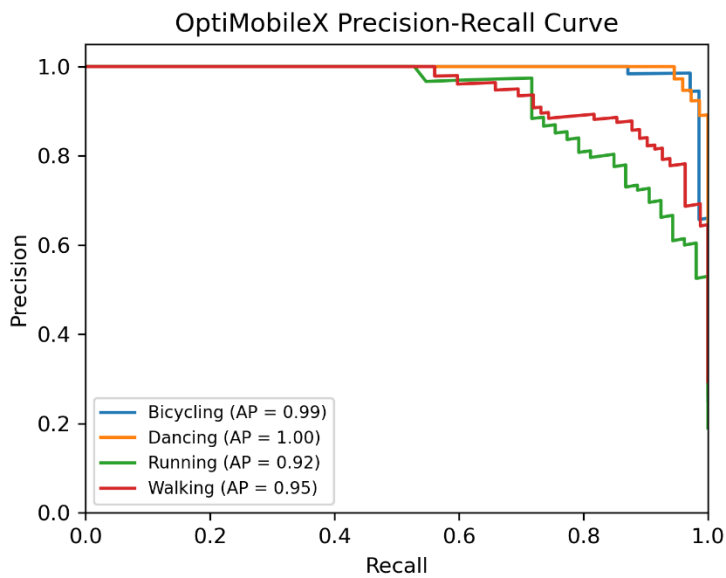
a)



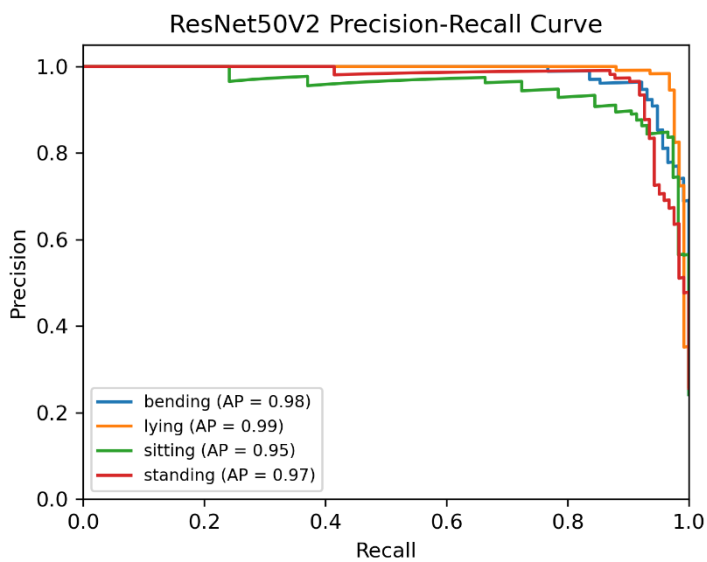
b)



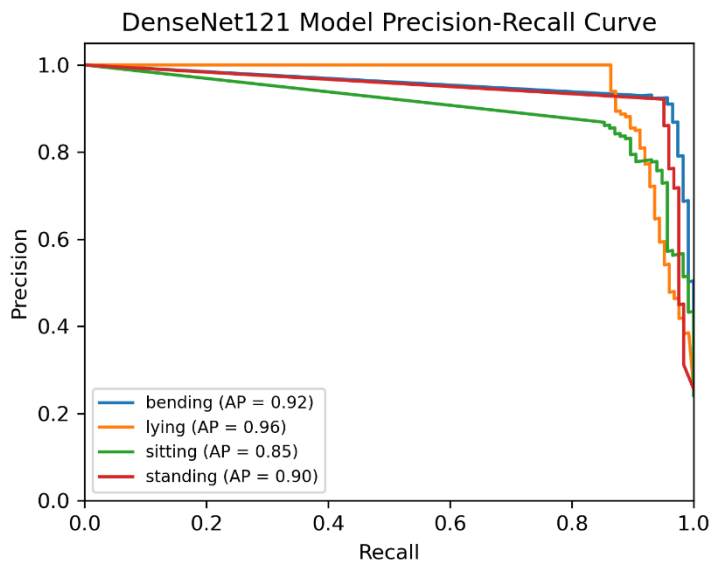
c)



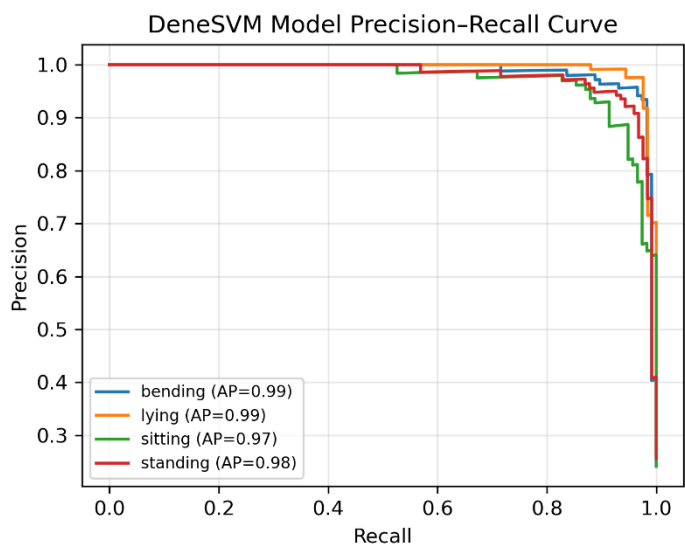
d)



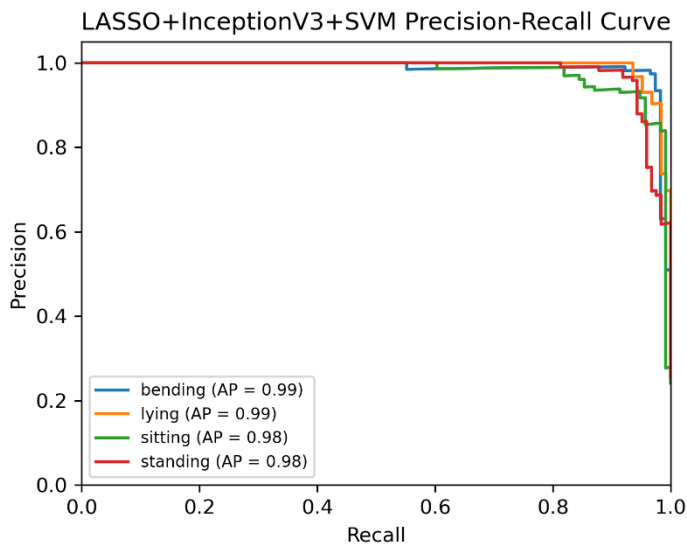
e)



f)



g)



h)

36 pav. DTL modelių tikslumo ir atsišaukimo santykis MPII: a) *InceptionV3*; b) *MobileNetV2*; c) *Xception*; d) *OptiMobileX*

Grėsmės validumui. Reikia atkreipti dėmesį į keletą apribojimų. Vertinimas buvo grindžiamas *Silhouettes* duomenų rinkiniu ir *MPII/ETRI* duomenų rinkinių dalimis. Nors šie duomenų rinkiniai apima įvairias pozas, jie gali neapimti visų realaus pasaulio situacijų (pvz., neįprasti kameros kampai, uždengimai, aprangos skirtumai). Kiekvienos klasės imties dydis (maždaug ≈ 200 bandomųjų vaizdų) yra vidutinis, todėl itin reti įvykiai gali būti nepakankamai atstovaujami. Ateityje reikėtų atlikti bandymus su didesniais, realaus gyvenimo situacijų ar pozų duomenų rinkiniais ir realiais diegimo atvejais. Nepaisant šių pastebėjimų, nuoseklus veikimas keliuose duomenų rinkiniuose ir statistinės analizės rodo, kad gautos išvados yra pagrįstos tiriamojo dalyje.

Paveikslų išdėstymo pasiūlymai. Visoje santraukoje paveikslai iliustruoja pagrindinius punktus. Pavyzdžiui, pateikus sistemos apžvalgą, įtraukta struktūros diagrama (3.1 pav.). Duomenų rinkinių skyriuje pateikti reprezentatyvūs vaizdai (pvz., *MPII* pavyzdžiai ir siluetų rėmai). Modelio architektūros skyriuje įtrauktos pavyzdinių tinklų diagramos (pvz., *AlexNet/CNN* blokas ir *OptiMobileX* jungimo struktūra). Rezultatų skyriuje įtrauktos mokymo / patikrinimo kreivės (4.1 pav.) ir painiavos matricos (4.2 pav.), lyginančios bazinį modelį su siūlomu modeliu. *ROC* kreives arba tikslumo ir atgaminimo grafikus taip pat galima pridėti prie *AUC* aptarimo. Šie vaizdiniai elementai paaiškina aprašytas architektūras ir veikimo skirtumus.

Santraukos 3 ir 4 skyriuose bendrai aprašoma išsami, giliaisiais tyrimais pagrįsta laikysenos nustatymo sistema. 3 skyriuje išsamiai aprašytas duomenų paruošimas (keletas duomenų rinkinių ir papildymų), išbandytų modelių spektras (nuo

paprastų *CNN* iki pažangių hibridinių modelių) ir mokymo metodika. Jame pristatyti nauji modeliai (*CNN-SVM* hibridai, savybių sujungimo ir dėmesio architektūros, lengvi kraštiniai modeliai), sukurti siekiant išspręsti ribotų duomenų ir diegimo apribojimų problemas. 4 skyriuje šie modeliai buvo griežtai įvertinti. Eksperimentais nustatyta, kad hibridiniai metodai nuosekliai pranoko standartinius *CNN*, pavyzdžiui, *DenseNet+SVM* pasiekė ~95 % bandymo tikslumą (palyginti su ~93 % *DenseNet*). Išsamūs rodikliai (tikslumas, atgaminimas, *AUC*) patvirtino aukštą našumą visose laikysenos klasėse ir statistiškai reikšmingą naudą. Ypač abliacijos tyrimas patvirtino, kad kiekvienas metodologinis indėlis (*LASSO*, *SVM*, dvigubo tinklo sujungimas ir dėmesio) duoda matomą naudą. Apibendrinant, rezultatai rodo, kad derinant perkėlimą, savybių atranką ir hibridinę klasifikaciją gaunamas labai tikslus, patikimas laikysenos klasifikatorius, tinkamas realaus pasaulio, realaus laiko diegimui.

LIST OF REFERENCES

1. AROWOLO, O. F.; AROGUNJO, E. O.; OWOLABI, D. G.; MARKUS, E. D. Development of a human posture recognition system for surveillance application. *International Journal of Computing and Digital Systems*. 2021, vol. 10. <http://dx.doi.org/10.12785/ijcds/1001107>
2. IAZZI, A.; RZIZA, M.; OULAD HAJ THAMI, R. Fall detection system-based posture recognition for indoor environments. *Journal of Imaging*. 2021, vol. 7, no. 3, art. 42. <https://doi.org/10.3390/jimaging7030042>
3. VERMANDER, P.; MANCISIDOR, A.; CABANES, I.; PEREZ, N. Intelligent systems for sitting posture monitoring and anomaly detection: an overview. *Journal of NeuroEngineering and Rehabilitation*. 2024, vol. 21, no. 1, art. 28. <https://doi.org/10.1186/s12984-024-01322-z>
4. PIÑERO-FUENTES, E.; CAÑAS-MORENO, S.; RIOS-NAVARRO, A.; DOMÍNGUEZ-MORALES, M.; SEVILLANO, J. L.; LINARES-BARRANCO, A. A deep-learning-based posture detection system for preventing telework-related musculoskeletal disorders. *Sensors*. 2021, vol. 21, no. 15, art. 5236. <https://doi.org/10.3390/s21155236>
5. WEAVER, S. J.; CHAMPION, M. Are laptop computers a health risk for an aging population? [n.d.]. doi:10.4017/gt.2010.09.03.005.00
6. MORO, A. Impact of the COVID-19 confinement measures on telework in Italy: a qualitative survey. *JRC Working Papers Series on Labour, Education and Technology*. 2020, no. 2020/10. Luxembourg: Publications Office of the European Union. <https://www.econstor.eu/bitstream/10419/231342/1/jrc-wplet202010.pdf>
7. LARREA-ARAUJO, C.; AYALA-GRANJA, J.; VINUEZA-CABEZAS, A.; ACOSTA-VARGAS, P. Ergonomic risk factors of teleworking in Ecuador during the COVID-19 pandemic: a cross-sectional study. *International Journal of Environmental Research and Public Health*. 2021, vol. 18, no. 10, art. 5063. <https://doi.org/10.3390/ijerph18105063>
8. CAO, B.; BI, S.; ZHENG, J.; YANG, D. Human posture recognition using skeleton and depth information. In: *Proceedings of the WRC Symposium on Advanced Robotics and Automation (WRC SARA 2018)*. Beijing, China, August 2018. Piscataway, NJ: IEEE, 2018, p. 275–280. DOI: 10.1109/WRC-SARA.2018.8584233
9. MARYAM, S. R. D.; PAYANDEH, S. A novel human posture estimation using single depth image from Kinect v2 sensor. In: *Proceedings of the IEEE International Systems Conference (SysCon 2018)*. Vancouver, Canada, April 2018. Piscataway, NJ: IEEE, 2018, p. 1–7. DOI: 10.1109/SYSCON.2018.8369566
10. SRIVASTAVA, A.; BADAL, T.; SAXENA, P.; VIDYARTHI, A.; SINGH, R. UAV surveillance for violence detection and individual identification. *Automated Software Engineering*. 2022, vol. 29, no. 1, art. 28. <https://doi.org/10.1007/s10515-022-00323-3>
11. AXSONSOFT. Behavior analytics. [online] [n.d.]. Available at: <https://www.axxonsoft.com/ai-video-analytics/video-content-analytics/behavior-analytics>.
12. SRIVASTAVA, S.; DELP, E. J. Video-based real-time surveillance of vehicles. *Journal of Electronic Imaging*. 2013, vol. 22, no. 4, art. 041103. <https://doi.org/10.1117/1.JEI.22.4.041103>
13. ANIKWE, C. V.; NWEKE, H. F.; IKEGWU, A. C.; EGUONWU, C. A.; ONU, F. U.; ALO, U. R.; TEH, Y. W. Mobile and wearable sensors for data-driven health monitoring

- system: state-of-the-art and future prospect. *Expert Systems with Applications*. 2022, vol. 202, art. 117362. <https://doi.org/10.1016/j.eswa.2022.117362>
14. ZHENG, C.; WU, W.; CHEN, C.; YANG, T.; ZHU, S.; SHEN, J.; SHAH, M. Deep learning-based human pose estimation: a survey. *ACM Computing Surveys*. 2023, vol. 56, no. 1, p. 1–37. <https://doi.org/10.1145/3603618>
 15. OGUNDOKUN, Roseline Oluwaseun; MASKELIŪNAS, Rytis; DAMAŠEVIČIUS, Robertas. Human posture detection using image augmentation and hyperparameter-optimized transfer learning algorithms. *Applied Sciences*. 2022, vol. 12, no. 19, art. 10156. <https://doi.org/10.3390/app121910156>
 16. ROGGIO, F.; TROVATO, B.; SORTINO, M.; MUSUMECI, G. A comprehensive analysis of the machine learning pose estimation models used in human movement and posture analyses: a narrative review. *Heliyon*. 2024. DOI: 10.1016/j.heliyon.2024.e39977
 17. ISLAM, M. M.; NOORUDDIN, S.; KARRAY, F.; MUHAMMAD, G. Human activity recognition using tools of convolutional neural networks: a state-of-the-art review, data sets, challenges, and future prospects. *Computers in Biology and Medicine*. 2022, vol. 149, art. 106060. <https://doi.org/10.1016/j.compbimed.2022.106060>
 18. VERMA, A.; SUMAN, A.; BIRADAR, V. G.; BRUNDA, S. Human activity classification using a deep convolutional neural network. In: *Recent Advances in Artificial Intelligence and Data Engineering*. Singapore: Springer, 2022, p. 41–50. https://doi.org/10.1007/978-981-16-3342-3_4
 19. OGUNDOKUN, Roseline Oluwaseun; MASKELIŪNAS, Rytis; MISRA, Sanjay; DAMAŠEVIČIUS, Robertas. Improved CNN based on batch normalization and Adam optimizer. In: *International Conference on Computational Science and Its Applications*. Cham: Springer, 2022, p. 593–604. DOI: https://doi.org/10.1007/978-3-031-10548-7_43
 20. LE, N. Q. K.; HO, Q. T. Deep transformers and a convolutional neural network to identify DNA N6-methyladenine sites in genomes of cross-species. *Methods*. 2022, vol. 204, p. 199–206. <https://doi.org/10.1016/j.ymeth.2021.12.004>
 21. DANILATOU, V.; NIKOLAKAKIS, S.; ANTONAKAKI, D.; TZAGKARAKIS, C.; MAVROIDIS, D.; KOSTOULAS, T.; IOANNIDIS, S. Outcome prediction in critically ill patients with venous thromboembolism and/or cancer using machine learning algorithms. *International Journal of Molecular Sciences*. 2022, vol. 23, art. 7132. <https://doi.org/10.3390/ijms23137132>
 22. LOVANSHI, M.; TIWARI, V. Human pose estimation: benchmarking deep learning-based methods. In: *Proceedings of the IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI 2022)*. Piscataway, NJ: IEEE, 2022, p. 1–6. DOI: 10.1109/IATMSI56455.2022.10119324
 23. LE, N. Q. K. Potential of deep representative learning features to interpret sequence information in proteomics. *Proteomics*. 2021, vol. 22, e2100232. <https://doi.org/10.1002/pmic.202100232>
 24. ALI, F.; EL-SAPPAGH, S.; ISLAM, S. R.; KWAK, D.; ALI, A.; IMRAN, M.; KWAK, K. S. A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion. *Information Fusion*. 2020, vol. 63, p. 208–222. <https://doi.org/10.1016/j.inffus.2020.06.008>
 25. CHOI, Y. A.; PARK, S. J.; JUN, J. A.; PYO, C. S. C.; CHO, K. H.; LEE, H. S.; YU, J. H. Deep learning-based stroke disease prediction system using real-time biosignals. *Sensors*. 2021, vol. 21, art. 4269. <https://doi.org/10.3390/s21134269>

26. PAN, Y.; FU, M.; CHENG, B.; TAO, X.; GUO, J. Enhanced deep learning-assisted convolutional neural network for heart disease prediction on the platform of medical things. *IEEE Access*. 2020, vol. 8, p. 189503–189512. DOI: 10.1109/ACCESS.2020.3026214
27. CAO, Z.; SIMON, T.; WEI, S. E.; SHEIKH, Y. Realtime multiperson 2D pose estimation using part affinity fields. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*. Honolulu, HI, USA, 21–26 July 2017, p. 7291–7299. https://openaccess.thecvf.com/content_cvpr_2017/papers/Cao_Realtime_Multi-Person_2D_CVPR_2017_paper.pdf
28. MEHR, H. D.; POLAT, H. Recognition of human activity in the smart home with the deep learning approach. In: *Proceedings of the 7th International Istanbul Smart Grids and Cities Congress and Fair (ICSG 2019)*. Istanbul, Turkey, 25–26 April 2019, p. 149–153. DOI: 10.1109/SGCF.2019.8782290
29. DENG, L. An overview of deep-structured learning for information processing. In: *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2011)*. Xi'an, China, 19–21 October 2011, p. 1–14. <https://www.microsoft.com/en-us/research/publication/an-overview-of-deep-structured-learning-for-information-processing/>
30. SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*. 2014, vol. 15, p. 1929–1958. https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm_content=buffer79b43&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer,
31. RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATY, A.; KHOSLA, A.; BERNSTEIN, M.; et al. ImageNet large-scale visual recognition challenge. *International Journal of Computer Vision*. 2015, vol. 115, p. 211–252. <https://doi.org/10.1007/s11263-015-0816-y>
32. KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*. 2012, vol. 25, p. 1097–1105. <https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html>
33. SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*. 2014, arXiv:1409.1556. <https://doi.org/10.48550/arXiv.1409.1556>
34. SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*. Boston, MA, USA, 7–12 June 2015, p. 1–9. https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Szegedy_Going_Deeper_With_2015_CVPR_paper.pdf
35. ZHANG, M.; JING, W.; LIN, J.; FANG, N.; WEI, W.; WOŹNIAK, M.; DAMAŠEVIČIUS, R. NAS-HRIS: automatic design and architecture search of neural networks for semantic segmentation in remote sensing images. *Sensors*. 2020, vol. 20, art. 1–15. <https://doi.org/10.3390/s20185292>
36. ALZUBAIDI, L., ZHANG, J., HUMAIDI, A.J., AL-DUJAILI, A., DUAN, Y., AL-SHAMMA, O., et al., 2021. Review of deep learning: concepts, CNN architectures,

- challenges, applications, future directions. *Journal of Big Data*, 8(1), 53. <https://doi.org/10.1186/s40537-021-00444-8>.
37. BEN GAMRA, M.; AKHLOUFI, M. A. A review of deep learning techniques for 2D and 3D human pose estimation. *Image and Vision Computing*. 2021, vol. 114, art. 104282. <https://doi.org/10.1016/j.imavis.2021.104282>
 38. SONG, L.; YU, G.; YUAN, J.; LIU, Z. Human pose estimation and its application to action recognition: a survey. *Journal of Visual Communication and Image Representation*. 2021, vol. 76, art. 103055. <https://doi.org/10.1016/j.jvcir.2021.103055>
 39. JAYATILAKE, S. M. D. A. C.; GANEGODA, G. U. Involvement of machine learning tools in healthcare decision making. *Journal of Healthcare Engineering*. 2021, vol. 2021, art. 6679512. <https://doi.org/10.1155/2021/6679512>
 40. DE SIQUEIRA, V. S.; BORGES, M. M.; FURTADO, R. G.; DOURADO, C. N.; DA COSTA, R. M. Artificial intelligence applied to support medical decisions for the automatic analysis of echocardiogram images: a systematic review. *Artificial Intelligence in Medicine*. 2021, vol. 120, art. 102165. <https://doi.org/10.1016/j.artmed.2021.102165>
 41. TSOUGOS, I.; VAMVAKAS, A.; KAPPAS, C.; FEZOULIDIS, I.; VASSIOU, K. Application of radiomics and decision support systems for breast MR differential diagnosis. *Computational and Mathematical Methods in Medicine*. 2018, vol. 2018, art. 7417126. <https://doi.org/10.1155/2018/7417126>
 42. YANG, Y.; FENG, X.; CHI, W.; LI, Z.; DUAN, W.; LIU, H.; LIANG, W.; WANG, W.; CHEN, P.; HE, J.; et al. Deep learning-aided decision support for pulmonary nodules diagnosing: a review. *Journal of Thoracic Disease*. 2018, vol. 10, suppl., p. S867–S875. doi: 10.21037/jtd.2018.02.57
 43. ALI, L.; RAHMAN, A.; KHAN, A.; ZHOU, M.; JAVEED, A.; KHAN, J. A. An automated diagnostic system for heart disease prediction based on statistical models and optimally configured deep neural networks. *IEEE Access*. 2019, vol. 7, p. 34938–34945. DOI: 10.1109/ACCESS.2019.2904800
 44. ANSARULLAH, S. I.; SAIF, S. M.; ANDRABI, S. A. B.; KUMHAR, S. H.; KIRMANI, M. M.; KUMAR, D. P. An intelligent and reliable hyperparameter optimization machine learning model for early heart disease assessment. *Journal of Healthcare Engineering*. 2022, vol. 2022, art. 9882288. <https://doi.org/10.1155/2022/9882288>
 45. COONEY, C.; KORIK, A.; FOLLI, R.; COYLE, D. Evaluation of hyperparameter optimization in machine and deep learning methods for decoding imagined speech EEG. *Sensors*. 2020, vol. 20, p. 1–22. <https://doi.org/10.3390/s20164629>
 46. DU, X.; XU, H.; ZHU, F. Understanding the effect of hyperparameter optimization on machine learning models for structure design problems. *Computer-Aided Design*. 2021, vol. 135, art. 103013. <https://doi.org/10.1016/j.cad.2021.103013>
 47. CHOLLET, F.; ALLAIRE, J. J. *Deep learning mit R und Keras*. Frechen: MITP-Verlags GmbH & Co. KG, 2018. <https://books.google.com/books?id=4GF6EAAAQBAJ>.
 48. ELGENDY, M. *Deep learning for vision systems*. New York, NY: Simon & Schuster, 2020. <https://books.google.com/books?id=97YCEAAAQBAJ>.
 49. SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on image data augmentation for deep learning. *Journal of Big Data*. 2019, vol. 6, p. 1–48. <https://doi.org/10.1186/s40537-019-0197-0>
 50. YOUNIS, M. C.; KEEDWELL, E. Semantic segmentation on small datasets of satellite images using convolutional neural networks. *Journal of Applied Remote Sensing*. 2019, vol. 13, art. 046510.

51. ZENG, S.; ZHANG, B.; ZHANG, Y.; GOU, J. Dual sparse learning via data augmentation for robust facial image classification. *International Journal of Machine Learning and Cybernetics*. 2020, vol. 11, p. 1717–1734. <https://doi.org/10.1007/s13042-020-01067-w>
52. ABAYOMI-ALLI, O. O.; DAMAŠEVIČIUS, R.; MASKELIŪNAS, R.; MISRA, S. Few-shot learning with a novel Voronoi tessellation-based image augmentation method for facial palsy detection. *Electronics*. 2021, vol. 10, art. 978. <https://doi.org/10.3390/electronics10080978>
53. ABAYOMI-ALLI, O. O.; DAMAŠEVIČIUS, R.; MISRA, S.; MASKELIŪNAS, R. Cassava disease recognition from low-quality images using enhanced data augmentation and deep learning. *Expert Systems*. 2021, vol. 38, e12746. <https://doi.org/10.1111/exsy.12746>
54. ABAYOMI-ALLI, O. O.; DAMAŠEVIČIUS, R.; MISRA, S.; MASKELIŪNAS, R.; ABAYOMI-ALLI, A. Malignant skin melanoma detection using image augmentation by oversampling in nonlinear lower-dimensional embedding manifold. *Turkish Journal of Electrical Engineering and Computer Sciences*. 2021, vol. 29, p. 2600–2614. DOI: 10.3906/elk-2101-133
55. OYEWOLA, D. O.; DADA, E. G.; MISRA, S.; DAMAŠEVIČIUS, R. A novel data-augmentation convolutional neural network for detecting malaria parasites in blood smear images. *Applied Artificial Intelligence*. 2022, vol. 36, art. 1. <https://doi.org/10.1080/08839514.2022.2033473>
56. WANG, Z.; YANG, J.; JIANG, H.; FAN, X. CNN training with twenty samples for crack detection via data augmentation. *Sensors*. 2020, vol. 20, art. 4849. <https://doi.org/10.3390/s20174849>
57. HUTTER, F.; HOOS, H.; LEYTON-BROWN, K. An efficient approach for assessing hyperparameter importance. In: *Proceedings of the International Conference on Machine Learning (ICML 2014)*. Beijing, China, 21–26 June 2014, p. 754–762. <http://proceedings.mlr.press/v32/hutter14.html>
58. HUTTER, F.; KOTTHOFF, L.; VANSCHOREN, J. *Automated machine learning: methods, systems, challenges*. Berlin: Springer Nature, 2019. <https://link.springer.com/content/pdf/10.1007/978-3-030-05318-5.pdf>
59. MANTOVANI, R. G.; ROSSI, A. L. D.; ALCOBACA, E.; VANSCHOREN, J.; DE CARVALHO, A. C. P. L. F. A meta-learning recommender system for hyperparameter tuning: Predicting when tuning improves SVM classifiers. *Information Sciences*. 2019, vol. 501, p. 193–221. <https://doi.org/10.1016/j.ins.2019.06.005>
60. NEARY, P. Automatic hyperparameter tuning in deep convolutional neural networks using asynchronous reinforcement learning. In: *Proceedings of the IEEE International Conference on Cognitive Computing (ICCC 2018)*. San Francisco, CA, USA, 2–7 July 2018, p. 73–77. DOI: 10.1109/ICCC.2018.00017
61. OTTONI, A. L.; NEPOMUCENO, E. G.; DE OLIVEIRA, M. S.; DE OLIVEIRA, D. C. Tuning of reinforcement learning parameters applied to SOP using the Scott–Knott method. *Soft Computing*. 2020, vol. 24, p. 4441–4453. <https://doi.org/10.1007/s00500-019-04206-w>
62. SCHRATZ, P.; MUENCHOW, J.; ITURRITXA, E.; RICHTER, J.; BRENNING, A. Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data. *Ecological Modelling*. 2019, vol. 406, p. 109–120. <https://doi.org/10.1016/j.ecolmodel.2019.06.002>

63. SHANKAR, K.; ZHANG, Y.; LIU, Y.; WU, L.; CHEN, C. H. Hyperparameter tuning deep learning for diabetic retinopathy fundus image classification. *IEEE Access*. 2020, vol. 8, p. 118164–118173. DOI: 10.1109/ACCESS.2020.3005152
64. PAWARA, P.; OKAFOR, E.; SCHOMAKER, L.; WIERING, M. Data augmentation for plant classification. In: *Proceedings of the International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS 2017)*. Antwerp, Belgium, 18–21 September 2017. Cham: Springer, 2017, p. 615–626. DOI: https://doi.org/10.1007/978-3-319-70353-4_52
65. SONG, C.; XU, W.; WANG, Z.; YU, S.; ZENG, P.; JU, Z. Analysis of the impact of data augmentation on target recognition for UAV-based transmission line inspection. *Complexity*. 2020, vol. 2020, art. 3107450. <https://doi.org/10.1155/2020/3107450>
66. MONSHI, M. M. A.; POON, J.; CHUNG, V.; MONSHI, F. M. CovidXrayNet: optimizing data augmentation and CNN hyperparameters for improved COVID-19 detection from CXR. *Computers in Biology and Medicine*. 2021, vol. 133, art. 104375. <https://doi.org/10.1016/j.combiomed.2021.104375>
67. DAMAŠEVIČIUS, R. Optimization of SVM parameters for recognition of regulatory DNA sequences. *TOP*. 2010, vol. 18, p. 339–353. <https://doi.org/10.1007/s11750-010-0152-x>
68. KALLIOLA, J.; KAPOČIŪTĖ-DZIKIENE, J.; DAMAŠEVIČIUS, R. Neural network hyperparameter optimization for prediction of real estate prices in Helsinki. *PeerJ Computer Science*. 2021, vol. 7, e444. <https://doi.org/10.7717/peerj-cs.444>
69. POŁAP, D.; WOŹNIAK, M.; HOŁUBOWSKI, W.; DAMAŠEVIČIUS, R. A heuristic approach to the hyperparameters in training spiking neural networks using spike-timing-dependent plasticity. *Neural Computing and Applications*. 2022, vol. 34, p. 13187–13200. <https://doi.org/10.1007/s00521-021-06824-8>
70. LAWAL, M. O. *Tomato detection based on modified YOLOv3 framework*. Berlin: Springer Science & Business Media, 2021. <https://doi.org/10.1038/s41598-021-81216-5>
71. ZHANG, K.; ROBINSON, N.; LEE, S.-W.; GUAN, C. *Adaptive transfer learning for EEG motor imagery classification with deep convolutional neural network*. Amsterdam: Elsevier, 2021. <https://doi.org/10.1016/j.neunet.2020.12.013>
72. ROY, A. M. *Adaptive transfer learning-based multiscale feature fused deep convolutional neural network for EEG motor imagery multiclassification in brain-computer interface*. Amsterdam: Elsevier, 2022. <https://doi.org/10.1016/j.engappai.2022.105347>
73. HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*. Las Vegas, NV, USA, 27–30 June 2016, p. 770–778. https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html
74. LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *Nature*. 2015, vol. 521, p. 436–444. <https://doi.org/10.1038/nature14539>
75. GUPTA, R.; SAINI, D.; MISHRA, S. Posture detection using deep learning for time series data. In: *Proceedings of the Third International Conference on Smart Systems and Inventive Technology (ICSSIT 2020)*. Tirunelveli, India, 20–22 August 2020, p. 740–744. DOI: 10.1109/ICSSIT48917.2020.9214223
76. KANG, B. R.; HER, J. G.; LEE, J. S.; KO, T. S.; YOU, Y. Y. Effects of the computer desk level on the musculoskeletal discomfort of neck and upper extremities and EMG

- activities in patients with spinal cord injuries. *Occupational Therapy International*. 2019, vol. 2019, art. 3026150. <https://doi.org/10.1155/2019/3026150>
77. LEE, S.; KANG, H.; SHIN, G. Head flexion angle while using a smartphone. *Ergonomics*. 2015, vol. 58, p. 220–226. <https://doi.org/10.1080/00140139.2014.967311>
 78. KIM, E.-K.; KIM, J. S. Correlation between rounded shoulder posture, neck disability indices, and degree of forward head posture. *Journal of Physical Therapy Science*. 2016, vol. 28, p. 2929–2932. <https://doi.org/10.1589/jpts.28.2929>
 79. RUIZ, A. D.; JUAN, M. S.; JUAN, L. M.; BEATRIZ, G. F. Characterization of the cardiovascular and respiratory system of healthy subjects in supine and sitting position. In: *Proceedings of the Iberian Conference on Pattern Recognition and Image Analysis*. Cham: Springer, 2019, p. 367–377. DOI: https://doi.org/10.1007/978-3-030-31332-6_32
 80. KOSEKI, T.; KAKIZAKI, F.; HAYASHI, S.; NISHIDA, N.; ITOH, M. Effect of forward head posture on thoracic shape and respiratory function. *Journal of Physical Therapy Science*. 2019, vol. 31, p. 63–68. <https://doi.org/10.1589/jpts.31.63>
 81. HAQUE, M. F.; AKHTER, S.; TASNIM, N.; HAQUE, M.; PAUL, S.; BEGUM, M. Effects of different sitting postures on forced vital capacity in healthy school children. *Bangladesh Medical Research Council Bulletin*. 2019, vol. 45, p. 117–121. DOI: <https://doi.org/10.3329/bmrcb.v45i2.42541>
 82. BOOTSMAN, R.; MARKOPOULOS, P.; QI, Q.; WANG, Q.; TIMMERMAN, A. A. W. Wearable technology for posture monitoring at the workplace. *International Journal of Human-Computer Studies*. 2019, vol. 132, p. 99–111. <https://doi.org/10.1016/j.ijhcs.2019.08.003>
 83. HO, L. E. S.; CHAN, J. C. P.; CHAN, D. C. K.; SHUM, H. P. H.; CHEUNG, Y.; PONG, Y. C. Improving posture classification accuracy for depth sensor-based human activity monitoring in smart environments. *Computer Vision and Image Understanding*. 2016, vol. 148, p. 97–110. <https://doi.org/10.1016/j.cviu.2015.12.011>
 84. FRAGKIADAKIS, E.; DALAKLEIDI, K. V.; NIKITA, K. S. Design and development of a sitting posture recognition system. In: *Proceedings of the 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2019)*. Berlin, Germany, 23–27 July 2019, p. 3364–3367. DOI: 10.1109/EMBC.2019.8856635
 85. WANG, Z.; YAN, W.; OATES, T. Time series classification from scratch with deep neural networks: a strong baseline. In: *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2017)*. Anchorage, AK, USA, 14–19 May 2017, p. 1578–1585. DOI: 10.1109/IJCNN.2017.7966039
 86. CHEN, L.; LI, S. Human motion target posture detection algorithm using semi-supervised learning in internet of things. *IEEE Access*. 2021, vol. 9, p. 90529–90538. doi: 10.1109/ACCESS.2021.3091430
 87. NÚÑEZ, J. C.; CABIDO, R.; PANTRIGO, J. J.; MONTEMAYOR, A. S.; VÉLEZ, J. F. Convolutional neural networks and long-term memory for skeleton-based human activity and hand gesture recognition. *Pattern Recognition*. 2018, vol. 76, p. 80–94. <https://doi.org/10.1016/j.patcog.2017.10.033>
 88. XU, J.; HE, Z.; ZHANG, Y. CNN–LSTM combined network for IoT-enabled fall detection applications. *Journal of Physics: Conference Series*. 2019, vol. 1267, art. 012044. DOI 10.1088/1742-6596/1267/1/012044
 89. DORBE, N.; JAUNDALDERS, A.; KADIKIS, R.; NESENBERGS, K. FCN and LSTM-based computer vision system for recognition of vehicle type, license plate number, and registration country. *Automatic Control and Computer Sciences*. 2018, vol. 52, p. 146–154. <https://doi.org/10.3103/S0146411618020104>

90. TAYLOR, W.; ABBASI, Q. H.; DASHTIPOUR, K.; ANSARI, S.; SHAH, S. A.; KHALID, A.; IMRAN, M. A. A review of the state of the art in noncontact sensing for COVID-19. *Sensors*. 2020, vol. 20, art. 5665. <https://doi.org/10.3390/s20195665>
91. GOGATE, M.; DASHTIPOUR, K.; HUSSAIN, A. Visual Speech in Real Noisy Environments (VISION): a novel benchmark dataset and a baseline system based on deep learning. In: *Proceedings of Interspeech 2020*. 2020, p. 4521–4525. https://www.isca-archive.org/interspeech_2020/gogate20_interspeech.pdf
92. AHMED, R.; DASHTIPOUR, K.; GOGATE, M.; RAZA, A.; ZHANG, R.; HUANG, K.; HAWALAH, A.; ADEEL, A.; HUSSAIN, A. Offline Arabic handwriting recognition using deep machine learning: a review of recent advances. In: *International Conference on Brain-Inspired Cognitive Systems*. Berlin/Heidelberg: Springer, 2019, p. 457–468. https://doi.org/10.1007/978-3-030-39431-8_44
93. GOGATE, M.; ADEEL, A.; HUSSAIN, A. Deep learning-driven multimodal fusion for automated deception detection. In: *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI 2017)*. Honolulu, HI, USA, 27 November–1 December 2017, p. 1–6. DOI: 10.1109/SSCI.2017.8285382
94. OZTURK, M.; GOGATE, M.; ONIRETI, O.; ADEEL, A.; HUSSAIN, A.; IMRAN, M. A. A novel deep learning-driven, low-cost mobility prediction approach for 5G cellular networks: the case of the Control/Data Separation Architecture. *Neurocomputing*. 2019, vol. 358, p. 479–489. <https://doi.org/10.1016/j.neucom.2019.01.031>
95. ADEEL, A.; GOGATE, M.; HUSSAIN, A.; WHITMER, W. M. Lip-reading driven deep learning approach for speech enhancement. *IEEE Transactions on Emerging Topics in Computational Intelligence*. 2019, vol. 5, p. 481–490. DOI: 10.1109/TETCI.2019.2917039
96. GOGATE, M.; HUSSAIN, A.; HUANG, K. Random features and random neurons for brain-inspired big data analytics. In: *Proceedings of the IEEE International Conference on Data Mining Workshops (ICDMW 2019)*. Beijing, China, 8–11 November 2019, p. 522–529. DOI: 10.1109/ICDMW.2019.00080
97. ZHU, Z.; MACHADO, P.; ZAHID, A.; ABDULGHANI, A. M.; DASHTIPOUR, K.; HEIDARI, H.; IMRAN, M. A.; ABBASI, Q. H. Energy and performance trade-off optimization in heterogeneous computing through reinforcement learning. *Electronics*. 2020, vol. 9, art. 1812. <https://doi.org/10.3390/electronics9111812>
98. GOGATE, M.; ADEEL, A.; DASHTIPOUR, K.; DERLETH, P.; HUSSAIN, A. AV speech enhancement challenge using a real noisy corpus. *arXiv preprint*. 2019, arXiv:1910.00424. <https://doi.org/10.48550/arXiv.1910.00424>
99. TAYLOR, W.; SHAH, S. A.; DASHTIPOUR, K.; ZAHID, A.; ABBASI, Q. H.; IMRAN, M. A. An intelligent noninvasive real-time human activity recognition system for next-generation healthcare. *Sensors*. 2020, vol. 20, art. 2653. <https://doi.org/10.3390/s20092653>
100. KOUBAA, A.; AMMAR, A.; BENJDIRA, B.; AL-HADID, A.; KAWAF, B.; AL-YAHRI, S. A.; BABIKER, A.; ASSAF, K.; BA RAS, M. Activity monitoring of Islamic prayer (Salat) postures using deep learning. In: *Proceedings of the 6th Conference on Data Science and Machine Learning Applications (CDMA 2020)*. Riyadh, Saudi Arabia, 4–5 March 2020, p. 106–111. doi: 10.1109/CDMA47397.2020.00024
101. ADEEL, A.; GOGATE, M.; FAROOQ, S.; IRECITANO, C.; DASHTIPOUR, K.; LARIJANI, H.; HUSSAIN, A. A survey on the role of wireless sensor networks and IoT in disaster management. In: *Geological Disaster Monitoring Based on Sensor Networks*. Berlin/Heidelberg: Springer, 2019, p. 57–66. DOI: https://doi.org/10.1007/978-981-13-0992-2_5

102. LEE, J.; JOO, H.; LEE, J.; CHEE, Y. Automatic classification of squat posture using inertial sensors: a deep learning approach. *Sensors*. 2020, vol. 20, art. 361. DOI: https://doi.org/10.1007/978-981-13-0992-2_5
103. JIANG, F.; KONG, B.; LI, J.; DASHTIPOUR, K.; GOGATE, M. Robust visual saliency optimization based on bidirectional Markov chains. *Cognitive Computation*. 2020, vol. 13, p. 69–80. <https://doi.org/10.1007/s12559-020-09724-6>
104. HAN, J.; SONG, W.; GOZHO, A.; SUNG, Y.; JI, S.; SONG, L.; WEN, L.; ZHANG, Q. LoRa-based smart IoT application for smart city: an example of human posture detection. *Wireless Communications and Mobile Computing*. 2020, vol. 2020, art. 8822555. <https://doi.org/10.1155/2020/8822555>
105. RADU, I.; TU, E.; SCHNEIDER, B. Relationships between body postures and collaborative learning states in an augmented reality study. In: *International Conference on Artificial Intelligence in Education*. Berlin/Heidelberg: Springer, 2020, p. 257–262. DOI: https://doi.org/10.1007/978-3-030-52240-7_47
106. AGRAWAL, Y.; SHAH, Y.; SHARMA, A. Implementation of machine learning techniques for the identification of yoga poses. In: *Proceedings of the IEEE 9th International Conference on Communication Systems and Network Technologies (CSNT 2020)*. Gwalior, India, 10–12 April 2020, p. 40–43. https://www.researchgate.net/profile/Yash-Shah-41/publication/342136553_Implementation_of_Machine_Learning_Technique_for_Identification_of_Yoga_Poses/links/6267a3048cb84a40ac8b3b60/Implementation-of-Machine-Learning-Technique-for-Identification-of-Yoga-Poses.pdf
107. IMRAN, M. A.; GHANNAM, R.; ABBASI, Q. H. *Engineering and technology for healthcare*. Hoboken, NJ: John Wiley & Sons, 2020. DOI:10.1002/9781119644316
108. HUSSEIN, I. O.; DASHTIPOUR, K.; HUSSAIN, A. Comparison of sentiment analysis approaches using modern Arabic and Sudanese dialects. In: *International Conference on Brain-Inspired Cognitive Systems*. Berlin/Heidelberg: Springer, 2018, p. 615–624. <https://www.ijcai.org/proceedings/2019/0805.pdf>
109. BOURAHMOUNE, K.; AMAGASA, T. AI-powered posture training: sitting posture recognition using the LifeChair smart cushion. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)*. Macao, China, 10–16 August 2019, p. 5808–5814. https://doi.org/10.1007/978-981-13-8303-8_31
110. SANDYBEKOV, M.; GRABOW, C.; GAIDUK, M.; SEEPOLD, R. Posture tracking using a machine learning algorithm for a home AAL environment. In: *Intelligent Decision Technologies 2019*. Berlin/Heidelberg: Springer, 2019, p. 337–347. https://doi.org/10.1007/978-981-13-8303-8_31
111. ROH, J.; PARK, H.-J.; LEE, K. J.; HYEONG, J.; KIM, S.; LEE, B. Sitting posture monitoring system based on a low-cost load cell using machine learning. *Sensors*. 2018, vol. 18, art. 208. <https://doi.org/10.3390/s18010208>
112. ROSERO-MONTALVO, P. D.; PELUFFO-ORDÓÑEZ, D. H.; LÓPEZ BATISTA, V. F.; SERRANO, J.; ROSERO, E. A. Intelligent system for identifying the posture of wheelchair users using machine learning techniques. *IEEE Sensors Journal*. 2018, vol. 19, p. 1936–1942. doi: 10.1109/JSEN.2018.2885323
113. LUNA-PEREJÓN, F.; MONTES-SÁNCHEZ, J. M.; DURÁN-LÓPEZ, L.; VÁZQUEZ-BAEZA, A.; BEASLEY-BOHÓRQUEZ, I.; SEVILLANO-RAMOS, J. L. IoT device for sitting posture classification using artificial neural networks. *Electronics*. 2021, vol. 10, no. 15, art. 1825. <https://doi.org/10.3390/electronics10151825>
114. VERMANDER, P.; MANCISIDOR, A.; CABANES, I.; PEREZ, N.; TORRES-UNDA, J. Intelligent sitting posture classifier for wheelchair users. *IEEE Transactions on Neural*

- Systems and Rehabilitation Engineering*. 2023, vol. 31, p. 944–953. doi: 10.1109/JSEN.2018.2885323
115. MASKELIŪNAS, R.; DAMAŠEVIČIUS, R.; BLAŽAUSKAS, T.; CANBULUT, C.; ADOMAVIČIENĖ, A.; GRIŠKEVIČIUS, J. BiomacVR: a virtual reality-based system for precise human posture and motion analysis in rehabilitation exercises using depth sensors. *Electronics*. 2023, vol. 12, no. 2, art. 339. <https://doi.org/10.3390/electronics12020339>
 116. OGUNDOKUN, R. O.; MASKELIŪNAS, R.; DAMAŠEVIČIUS, R. Human posture detection using image augmentation and hyperparameter-optimized transfer learning algorithms. *Applied Sciences*. 2022, vol. 12, no. 19, art. 10156. <https://doi.org/10.3390/app121910156>
 117. WANG, J.; CHENG, H. Human posture recognition based on convolutional neural networks. In: *Proceedings of the 4th International Conference on Electronic Information Technology and Computer Engineering*. 2020, p. 475–480. <https://doi.org/10.1145/3443467.344380>
 118. OGUNDOKUN, R. O.; MASKELIŪNAS, R.; MISRA, S.; DAMAŠEVIČIUS, R. A novel deep transfer learning approach based on depth-wise separable CNN for human posture detection. *Information*. 2022, vol. 13, no. 11, art. 520. <https://doi.org/10.3390/info13110520>
 119. JUANG, C. F.; NI, W. E. Human posture classification using interpretable 3D fuzzy body voxel features and hierarchical fuzzy classifiers. *IEEE Transactions on Fuzzy Systems*. 2022, vol. 30, no. 12, p. 5405–5418. doi: 10.1109/TFUZZ.2022.3177028.
 120. CHENG, Y. W.; LIN, Y. M.; JUANG, C. F. Human posture classification using a dual deep convolutional neural network with silhouette images. In: *Proceedings of the IET International Conference on Engineering Technologies and Applications (IET-ICETA 2022)*. Piscataway, NJ: IEEE, 2022, p. 1–2. doi: 10.1109/IET-ICETA56553.2022.9971534.
 121. ZHOU, X.; ZHANG, L. SA-FPN: an effective feature pyramid network for crowded human detection. *Applied Intelligence*. 2022, vol. 52, no. 11, p. 12556–12568. <https://doi.org/10.1007/s10489-021-03121-8>
 122. ZHENG, Y.; LV, X.; QIAN, L.; LIU, X. An optimal BP neural network track prediction method based on a GA–ACO hybrid algorithm. *Journal of Marine Science and Engineering*. 2022, vol. 10, no. 10, art. 1399. <https://doi.org/10.3390/jmse10101399>
 123. ZHENG, Y.; LIU, P.; QIAN, L.; QIN, S.; LIU, X.; MA, Y.; CHENG, G. Recognition and depth estimation of ships based on binocular stereo vision. *Journal of Marine Science and Engineering*. 2022, vol. 10, no. 8, art. 1153. <https://doi.org/10.3390/jmse10081153>
 124. LIU, A. A.; ZHAI, Y.; XU, N.; NIE, W.; LI, W.; ZHANG, Y. Region-aware image captioning via interaction learning. *IEEE Transactions on Circuits and Systems for Video Technology*. 2022, vol. 32, no. 6, p. 3685–3696. doi: 10.1109/TCSVT.2021.3107035
 125. HU, Z.; REN, L.; WEI, G.; QIAN, Z.; LIANG, W.; CHEN, W.; WANG, K. Energy flow and functional behavior of individual muscles at different speeds during human walking. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. 2023, vol. 31, p. 294–303. doi: 10.1109/TNSRE.2022.3221986.
 126. CHEN, K.; WANG, Q. Human posture recognition based on skeleton data. In: *Proceedings of the IEEE International Conference on Progress in Informatics and Computing (PIC 2015)*. 2015, p. 618–622. doi: 10.1109/PIC.2015.7489922.

127. MASKELIŪNAS, R.; KULIKAJEVA S, A.; DAMAŠEVIČIUS, R.; GRIŠKEVIČIUS, J.; ADOMAVIČIENĖ, A. Biomac3D: 2D-to-3D human pose analysis model for tele-rehabilitation based on Pareto-optimized deep-learning architecture. *Applied Sciences*. 2023, vol. 13, no. 2, art. 1116. <https://doi.org/10.3390/app13021116>
128. SHE, Q.; HU, R.; XU, J.; LIU, M.; XU, K.; HUANG, H. Learning high-DOF reaching-and-grasping via dynamic representation of gripper–object interaction. *arXiv preprint*. 2022, arXiv:2204.13998. <https://doi.org/10.48550/arXiv.2204.13998>
129. ZHANG, J.; ZHU, C.; ZHENG, L.; XU, K. ROSEFusion: random optimization for online dense reconstruction under fast camera motion. *ACM Transactions on Graphics*. 2021, vol. 40, no. 4, p. 1–17. <https://doi.org/10.1145/3450626.345967>
130. RAMANAN, D.; SMINCHISESCU, C. Training deformable models for localization. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*. 2006, vol. 1, p. 206–213. doi: 10.1109/CVPR.2006.315.
131. JIANG, H.; LI, Z.; DREW, M. S. Human posture recognition with convex programming. In: *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2005)*. Amsterdam, Netherlands, July 2005. Piscataway, NJ: IEEE, 2005, p. 574–577. doi: 10.1109/ICME.2005.1521488.
132. SOUTO, H.; MUSSE, S. Automatic detection of 2D human postures based on single images. In: *Proceedings of the 24th SIBGRAPI Conference on Graphics, Patterns and Images*. Maceió, Brazil, August 2011. Piscataway, NJ: IEEE, 2011, p. 48–55. doi: 10.1109/SIBGRAPI.2011.4
133. LEE, M. W.; COHEN, I. A model-based approach for estimating human 3D poses in static images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2006, vol. 28, no. 6, p. 905–916. doi: 10.1109/TPAMI.2006.110
134. SARAFIANOS, N.; BOTEANU, B.; IONESCU, B.; KAKADIARIS, I. A. 3D human pose estimation: a review of the literature and analysis of covariates. *Computer Vision and Image Understanding*. 2016, vol. 152, p. 1–20. <https://doi.org/10.1016/j.cviu.2016.09.002>
135. DU, C.; YAN, Z.; XIONG, Z.; YU, L. Boosting integral-based human pose estimation through implicit heatmap learning. *Neural Networks*. 2024, vol. 179, art. 106524. <https://doi.org/10.1016/j.neunet.2024.106524>
136. BAO, N.; ZHANG, T.; HUANG, R.; BISWAL, S.; SU, J.; WANG, Y. A deep transfer learning network for structural condition identification with limited real-world training data. *Structural Control and Health Monitoring*. 2023, vol. 2023, art. 8899806. <https://doi.org/10.1155/2023/8899806>
137. HAN, Y.; WANG, B.; GUAN, T.; TIAN, D.; YANG, G.; WEI, W.; CHUAH, J. H. Research on road environmental sensing methods of intelligent vehicles based on tracking check. *IEEE Transactions on Intelligent Transportation Systems*. 2023, vol. 24, no. 1, p. 1261–1275. doi: 10.1109/TITS.2022.3183893
138. LIN, T. Y.; HSIEH, C. H.; LEE, J. D. A Kinect-based system for physical rehabilitation: utilizing Tai Chi exercises to improve movement disorders in patients with balance ability. In: *Proceedings of the 7th Asia Modelling Symposium*. Hong Kong, July 2013. Piscataway, NJ: IEEE, 2013, p. 149–153. doi: 10.1109/AMS.2013.29
139. VIEIRA, S.; PINAYA, W. H. L.; MECHELLI, A. Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: methods and applications. *Neuroscience and Biobehavioral Reviews*. 2017, vol. 74, p. 58–75. <https://doi.org/10.1016/j.neubiorev.2017.01.002>

140. LI, S.; DOU, Y.; NIU, X.; LV, Q.; WANG, Q. A fast and memory-saved GPU acceleration algorithm of convolutional neural networks for target detection. *Neurocomputing*. 2017, vol. 230, p. 48–59. <https://doi.org/10.1016/j.neucom.2016.11.046>
141. IOFFE, S.; SZEGEDY, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the International Conference on Machine Learning (ICML 2015)*. Lille, France, 2015, p. 448–456. <http://proceedings.mlr.press/v37/ioffe15.html>
142. SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint*. 2014, arXiv:1409.1556. <https://doi.org/10.48550/arXiv.1409.1556>
143. SZEGEDY, C.; VANHOUCKE, V.; IOFFE, S.; SHLENS, J.; WOJNA, Z. Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*. Las Vegas, NV, USA, June 2016, p. 2818–2826. https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Szegedy_Rethinking_the_Inception_n_CVPR_2016_paper.html
144. SHAFIQ, M. and GU, Z., 2022. Deep residual learning for image recognition: A survey. *Applied Sciences*, 12(18), 8972. <https://doi.org/10.3390/app12188972>
145. HUANG, G.; LIU, Z.; VAN DER MAATEN, L.; WEINBERGER, K. Q. Densely connected convolutional networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*. Honolulu, HI, USA, July 2017, p. 4700–470. https://openaccess.thecvf.com/content_cvpr_2017/html/Huang_Densely_Connected_Convolutional_CVPR_2017_paper.html
146. CLEVERT, D. A.; UNTERTHINER, T.; HOCHREITER, S. Fast and accurate deep network learning by exponential linear units (ELUs). *arXiv preprint*. 2015, arXiv:1511.07289. https://www.researchgate.net/profile/Djork-Arne-Clevert/publication/284579051_Fast_and_Accurate_Deep_Network_Learning_by_Exponential_Linear_Units_ELUs/links/569ca97b08aea1476955915b/Fast-and-Accurate-Deep-Network-Learning-by-Exponential-Linear-Units-ELUs.pdf
147. COSTARELLI, D.; VINTI, G. Pointwise and uniform approximation by multivariate neural network operators of the max-product type. *Neural Networks*. 2016, vol. 81, p. 81–90. <https://doi.org/10.1016/j.neunet.2016.06.002>
148. LEE, C. Y.; XIE, S.; GALLAGHER, P.; ZHANG, Z.; TU, Z. Deeply-supervised nets. In: *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS 2015)*. 2015, p. 562–570. <https://proceedings.mlr.press/v38/lee15a>
149. LIANG, S.; KHOO, Y.; YANG, H. Drop-activation: implicit parameter reduction and harmonious regularization. *Communications on Applied Mathematics and Computation*. 2021, vol. 3, p. 293–311. <https://doi.org/10.1007/s42967-020-00085-3>
150. YAMADA, Y.; IWAMURA, M.; KISE, K. ShakeDrop regularization. *arXiv preprint*. 2018, arXiv:1802.02375. <https://openreview.net/forum?id=S1NHMW0b>
151. ZHANG, K.; SUN, M.; HAN, T. X.; YUAN, X.; GUO, L.; LIU, T. Residual networks of residual networks: multilevel residual networks. *IEEE Transactions on Circuits and Systems for Video Technology*. 2017, vol. 28, no. 6, p. 1303–1314. doi: 10.1109/TCSVT.2017.2654543
152. LIANG, M.; HU, X. Recurrent convolutional neural network for object recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2015)*. Boston, MA, USA, June 2015, p. 3367–3375).

- https://openaccess.thecvf.com/content_cvpr_2015/html/Liang_Recurrent_Convolutional_Neural_2015_CVPR_paper.html
153. LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. 1998, vol. 86, no. 11, p. 2278–2324. doi: 10.1109/5.726791
 154. ZHANG, Z. Improved Adam optimizer for deep neural networks. In: *Proceedings of the IEEE/ACM 26th International Symposium on Quality of Service (IWQoS 2018)*. Banff, Canada, June 2018, p. 1–2. doi: 10.1109/IWQoS.2018.8624183
 155. LIAQAT, S.; DASHTIPOUR, K.; ARSHAD, K.; ASSALEH, K.; RAMZAN, N. A hybrid posture detection framework: integrating machine learning and deep neural networks. *IEEE Sensors Journal*. 2021, vol. 21, no. 7, p. 9515–9522. doi: 10.1109/JSEN.2021.3055898
 156. TRAN, T. H.; NGUYEN, D. T.; NGUYEN, T. P. Human posture classification from multiple viewpoints and application for fall detection. In: *Proceedings of the IEEE Eighth International Conference on Communications and Electronics (ICCE 2020)*. Phu Quoc, Vietnam, January 2021, p. 262–267. doi: 10.1109/ICCE48956.2021.935214
 157. KOUBAA, A.; AMMAR, A.; BENJDIRA, B.; AL-HADID, A.; KAWAF, B.; AL-YAHRI, S. A.; RAS, M. B. Activity monitoring of Islamic prayer (Salat) postures using deep learning. In: *Proceedings of the 6th Conference on Data Science and Machine Learning Applications (CDMA 2020)*. Riyadh, Saudi Arabia, March 2020, p. 106–111. doi: 10.1109/CDMA47397.2020.00024.
 158. JIANG, F.; DASHTIPOUR, K.; HUSSAIN, A. A survey on deep learning for the routing layer of computer networks. In: *Proceedings of the UK/China Emerging Technologies Conference (UCET 2019)*. Glasgow, UK, August 2019, p. 1–4. doi: 10.1109/UCET.2019.8881852.
 159. DEVI, K. N.; ANAND, J.; KOTHAI, R.; KRISHNA, J. A.; MUTHURAMPANDIAN, R. Sensor-based posture detection system. *Materials Today: Proceedings*. 2022, vol. 55, p. 359–364. <https://doi.org/10.1016/j.matpr.2021.09.556>
 160. GARG, S.; SAXENA, A.; GUPTA, R. Yoga pose classification: a CNN and MediaPipe-inspired deep learning approach for real-world applications. *Journal of Ambient Intelligence and Humanized Computing*. 2022, p. 1–12. <https://doi.org/10.1007/s12652-022-03910-0>
 161. SHARMA, A.; SHAH, Y.; AGRAWAL, Y.; JAIN, P. Real-time recognition of yoga poses using computer vision for smart healthcare. *arXiv preprint*. 2022, arXiv:2201.07594. <https://doi.org/10.48550/arXiv.2201.07594>
 162. HAN, J.; ZHANG, D.; CHENG, G.; LIU, N.; XU, D. Advanced deep-learning techniques for salient and category-specific object detection: a survey. *IEEE Signal Processing Magazine*. 2018, vol. 35, no. 1, p. 84–100. doi: 10.1109/MSP.2017.2749125.
 163. SHIN, H. C.; ROTH, H. R.; GAO, M.; LU, L.; XU, Z.; NOGUES, I.; SUMMERS, R. M. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics, and transfer learning. *IEEE Transactions on Medical Imaging*. 2016, vol. 35, no. 5, p. 1285–1298. doi: 10.1109/TMI.2016.2528162.
 164. LIU, S.; YIN, Y.; OSTADABBAS, S. In-bed pose estimation: deep learning with shallow datasets. *IEEE Journal of Translational Engineering in Health and Medicine*. 2019, vol. 7, p. 1–12. doi: 10.1109/JTEHM.2019.2892970
 165. CHEN, Y.; TIAN, Y.; HE, M. Monocular human pose estimation: a survey of deep learning-based methods. *Computer Vision and Image Understanding*. 2020, vol. 192, art. 102897. <https://doi.org/10.1016/j.cviu.2019.102897>

166. ALOM, M. Z.; TAHA, T. M.; YAKOPCIC, C.; WESTBERG, S.; SIDIKE, P.; NASRIN, M. S.; ASARI, V. K. The history began from AlexNet: a comprehensive survey on deep learning approaches. *arXiv preprint*. 2018, arXiv:1803.01164. <https://doi.org/10.48550/arXiv.1803.01164>
167. FAISAL, A. I.; MAJUMDER, S.; MONDAL, T.; COWAN, D.; NASEH, S.; DEEN, M. J. Monitoring methods of human body joints: state-of-the-art and research challenges. *Sensors*. 2019, vol. 19, no. 11, art. 2629. DOI: 10.1109/IWCMC.2019.8766694
168. BOUALIA, S. N.; AMARA, N. E. B. Pose-based human activity recognition: a review. In: *Proceedings of the 15th International Wireless Communications and Mobile Computing Conference (IWCMC 2019)*. Tangier, Morocco, June 2019, p. 1468–1475. doi: 10.1109/IWCMC.2019.8766694.
169. CHOWDHURY, A. I.; ASHRAF, M.; ISLAM, A.; AHMED, E.; JAMAN, M. S.; RAHMAN, M. M. hActNET: an improved neural network-based method in recognizing human activities. In: *Proceedings of the 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT 2020)*. Turkey, October 2020, p. 1–6. doi: 10.1109/ISMSIT50672.2020.9254992.
170. LIAQAT, S.; DASHTIPOUR, K.; SHAH, S. A.; RIZWAN, A.; ALOTAIBI, A. A.; ALTHOBAITI, T.; RAMZAN, N. Novel ensemble algorithm for multiple activity recognition in elderly people exploiting ubiquitous sensing devices. *IEEE Sensors Journal*. 2021, vol. 21, no. 16, p. 18214–18221. doi: 10.1109/JSEN.2021.3085362
171. KUMAR, D.; SINHA, A. *Yoga pose detection and classification using deep learning*. Saarbrücken: LAP LAMBERT Academic Publishing, 2020. <https://doi.org/10.32628/cseit206623>
172. BYEON, Y. H.; LEE, J. Y.; KIM, D. H.; KWAK, K. C. Posture recognition using ensemble deep models under various home environments. *Applied Sciences*. 2020, vol. 10, no. 4, art. 1287. <https://doi.org/10.3390/app10041287>
173. KULIKAJEVA S, A.; MASKELIŪNAS, R.; DAMAŠEVIČIUS, R. Detection of sitting posture using hierarchical image composition and deep learning. *PeerJ Computer Science*. 2021, vol. 7, e442. <https://doi.org/10.7717/peerj-cs.442>
174. ALBU, F.; NICOLAU, M.; PIRVAN, F.; HAGIESCU, D. A sonification method using human body movements. In: *Proceedings of the 10th International Conference on Creative Content Technologies*. Rome, Italy, February 2018, p. 18–22. https://dlwqtxtslxzl7.cloudfront.net/84125438/download_full-libre.pdf
175. PANIGRAHY, D.; SAHU, P. K.; ALBU, F. Detection of ventricular fibrillation rhythm using boosted support vector machines with optimal variable combinations. *Computers and Electrical Engineering*. 2021, vol. 91, art. 107035. <https://doi.org/10.1016/j.compeleceng.2021.107035>
176. NAGALAKSHMI, C.; MUKHERJEE, S. Classification of yoga asanas from a single image by learning the 3D view of human poses. In: *Digital Techniques for Heritage Presentation and Preservation*. Singapore: Springer, 2021, p. 37–49. https://doi.org/10.1007/978-3-030-57907-4_3
177. ANDRILUKA, M.; PISHCHULIN, L.; GEHLER, P.; SCHIELE, B. 2D human pose estimation: new benchmark and state-of-the-art analysis. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014)*. Columbus, OH, USA, June 2014, p. 3686–3693. https://openaccess.thecvf.com/content_cvpr_2014/html/Andriluka_2D_Human_Pose_2014_CVPR_paper.html
178. SINGH, G.; GEORGE, R. P.; AHMAD, N.; HUSSAIN, S.; ATHER, D.; KLER, R. A deep learning approach for evaluating the efficacy and accuracy of PoseNet for posture

- detection. *International Journal of System Assurance Engineering and Management*. 2024, p. 1–10. <https://doi.org/10.1007/s13198-024-02530-5>
179. BULBUL, M. F.; ISLAM, S.; ALI, H. 3D human action analysis and recognition through GLAC descriptor on 2D motion and static posture images. *Multimedia Tools and Applications*. 2019, vol. 78, no. 15, p. 21085–21111. <https://doi.org/10.1007/s11042-019-7365-2>
 180. OGUNDOKUN, R. O.; MASKELIŪNAS, R.; MISRA, S.; DAMAŠEVIČIUS, R. A novel deep transfer learning approach based on depth-wise separable CNN for human posture detection. *Information*. 2022, vol. 13, no. 11, art. 520. <https://doi.org/10.3390/info13110520>
 181. YU, J.; TAN, M.; ZHANG, H.; RUI, Y.; TAO, D. Hierarchical deep click feature prediction for fine-grained image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2022, vol. 44, no. 2, p. 563–578. doi: 10.1109/TPAMI.2019.2932058.
 182. OGUNDOKUN, R. O.; MASKELIŪNAS, R.; DAMAŠEVIČIUS, R.; MISRA, S. Hybrid InceptionV3–SVM-based approach for human posture detection in health monitoring systems. *Algorithms*. 2022, vol. 15, no. 11, art. 410. <https://doi.org/10.3390/a15110410>
 183. OGUNDOKUN, R. O.; MASKELIŪNAS, R.; DAMAŠEVIČIUS, R.; ADEBIYI, M. O. MobileDNN: leveraging the potential of hybrid MobileNetV2–DNN for advanced image analysis. In: *Proceedings of the International Conference on Science, Engineering and Business for Driving Sustainable Development Goals (SEB4SDG 2024)*. Omu-Aran, Nigeria, April 2024, p. 1–5. doi: 10.1109/SEB4SDG60871.2024.10630298.
 184. TAJIMA, S.; TSUJI, T.; SUZUKI, Y.; WATANABE, T.; MOROOKA, K.; HARADA, K.; SEKI, H. Robot arm motion generation based on learning object shape and human motion. In: *Proceedings of the JSME Annual Conference on Robotics and Mechatronics (ROBOMECH 2018)*. Japan Society of Mechanical Engineers, 2018. https://article.nadiapub.com/IJGDC/vol11_no2/2.pdf?utm_source=chatgpt.com
 185. OGUNDOKUN, R. O.; MASKELIŪNAS, R.; DAMAŠEVIČIUS, R. Human posture detection using image augmentation and hyperparameter-optimized transfer learning algorithms. *Applied Sciences*. 2022, vol. 12, no. 19, art. 10156. <https://doi.org/10.3390/app121910156>
 186. YIN, Y.; HUANG, Q.; GAO, H.; XU, Y. Personalized API recommendation with cognitive knowledge mining for industrial systems. *IEEE Transactions on Industrial Informatics*. 2021, vol. 17, no. 9, p. 6153–6161. doi: 10.1109/TII.2020.3039500.
 187. OGUNDOKUN, R. O.; MASKELIŪNAS, R.; MISRA, S.; DAMAŠEVIČIUS, R. A novel deep transfer learning approach based on depth-wise separable CNN for human posture detection. *Information*. 2022, vol. 13, no. 11, art. 520. <https://doi.org/10.3390/info13110520>
 188. KIM, J.-J.; CHA, S.-H.; CHO, K.-H.; RYU, M. Deep reinforcement learning-based multi-agent collaborated network for distributed stock trading. *International Journal of Grid and Distributed Computing*. 2018, vol. 11, no. 2, p. 11–20. https://article.nadiapub.com/IJGDC/vol11_no2/2.pdf
 189. ZHANG, Z.; YANG, Y.; WU, Z.; MA, Q. A posture recognition system for rat cyborg automated navigation. *Chinese Journal of Electronics*. 2018, vol. 27, no. 4, p. 25–31. <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/cje.2018.04.003>
 190. SRIVASTAVA, P.; AGARWAL, D.; BANSAL, A. Ear-based human identification using a combination of wavelets and multi-scale local binary patterns. *International*

- Journal of Future Generation Communication and Networking*. 2019, vol. 12, no. 3, p. 41–56. <http://dx.doi.org/10.33832/ijfgcn.2019.12.3.04>.
191. LEE, Y.-K.; PARK, D.-W. Design of internet of things business models with deep learning artificial intelligence. *International Journal of Grid and Distributed Computing*. 2018, vol. 11, no. 7, p. 11–22.
 192. ALGAET, M. A.; BASARI, A. S. B. H.; MILAD, A. A.; ADRUGI, S. M.; ALDEEP, S. M. Medical data compression and transmission in noisy WLANs: a review. *International Journal of Grid and Distributed Computing*. 2019, vol. 12, no. 2, p. 1–18. <http://dx.doi.org/10.33832/ijgdc.2019.12.2.01>
 193. RAO, N. T. A review on industrial applications of machine learning. *International Journal of Disaster Recovery and Business Continuity*. 2018, vol. 9, p. 1–10.
 194. JI, S.; XU, W.; YANG, M.; YU, K. 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2013, vol. 35, no. 1, p. 221–231. doi: 10.1109/TPAMI.2012.59
 195. NIZAM, Y.; MOHD, M. N. H.; JAMIL, M. M. A. Human fall detection from depth images using position and velocity of the subject. *Procedia Computer Science*. 2017, vol. 105, p. 131–137. <https://doi.org/10.1016/j.procs.2017.01.191>
 196. RIMMINEN, H.; LINDSTRÖM, J.; LINNAVUO, M.; SEPPONEN, R. Detection of falls among the elderly by a floor sensor using the electric near field. *IEEE Transactions on Information Technology in Biomedicine*. 2010, vol. 14, no. 6, p. 1475–1476. doi: 10.1109/TITB.2010.2051956
 197. SIMONYAN, K.; ZISSERMAN, A. Two-stream convolutional networks for action recognition in videos. In: *Advances in Neural Information Processing Systems*. 2014, vol. 27. https://proceedings.neurips.cc/paper_files/paper/2014/hash/ca007296a63f7d1721a2399d56363022-Abstract.html
 198. AL-RAWI, S. S.; SADIQ, A. T.; ALALUOSI, W. M. Feature extraction of human facial expressions using Haar wavelet and neural networks. *Iraqi Journal of Science*. 2016, p. 1558–1565.
 199. KARPATY, A.; JOULIN, A.; FEI-FEI, L. F. Deep fragment embeddings for bidirectional image–sentence mapping. In: *Advances in Neural Information Processing Systems*. 2014, vol. 27. https://proceedings.neurips.cc/paper_files/paper/2014/hash/fe31664311a4a44314c89321386a20f6-Abstract.html
 200. TRAN, D.; BOURDEV, L.; FERGUS, R.; TORRESANI, L.; PALURI, M. Learning spatiotemporal features with 3D convolutional networks. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV 2015)*. Santiago, Chile, December 2015, p. 4489–4497. https://openaccess.thecvf.com/content_iccv_2015/html/Tran_Learning_Spatiotemporal_Features_ICCV_2015_paper.html
 201. OGUNDOKUN, Roseline Oluwaseun; MASKELIŪNAS, Rytis; MISRA, Sanjay; DAMAŠEVIČIUS, Robertas. A novel deep transfer learning approach based on depth-wise separable CNN for human posture detection. *Information*. Basel: MDPI, 2022, vol. 13, no. 11, art. no. 520. ISSN 2078-2489. <https://doi.org/10.3390/info13110520>
 202. HASIB, Rifat; KHAN, Kazi Nazmul; YU, Ming; KHAN, Md. Shafiul. Vision-based human posture classification and fall detection using convolutional neural network. In: *Proceedings of the 2021 International Conference on Artificial Intelligence (ICAI)*. Piscataway, NJ: IEEE, 2021, p. 74–79. doi: 10.1109/ICAI52203.2021.9445263.

203. NGUYEN, Dat T.; LI, Weimin; OGUNBONA, Philip O. Human detection from images and videos: A survey. *Pattern Recognition*. Oxford: Elsevier, 2016, vol. 51, p. 148–175. ISSN 0031-3203.
204. FAN, Kai; WANG, Peng; HU, Yong; DOU, Bin. Fall detection via human posture representation and support vector machine. *International Journal of Distributed Sensor Networks*. London: SAGE, 2017, vol. 13, no. 5, art. no. 1550147717707418. ISSN 1550-1329. <https://doi.org/10.1177/1550147717707418>
205. TOSHEV, Alexander; SZEGEDY, Christian. DeepPose: Human pose estimation via deep neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ: IEEE, 2014, p. 1653–1660. https://openaccess.thecvf.com/content_cvpr_2014/html/Toshev_DeepPose_Human_Pose_2014_CVPR_paper.html
206. LI, Jianan; LIANG, Xiaodan; SHEN, Shichao; XU, Tingfa; FENG, Jiashi; YAN, Shuicheng. Scale-aware fast R-CNN for pedestrian detection. *IEEE Transactions on Multimedia*. Piscataway, NJ: IEEE, 2017, vol. 20, no. 4, p. 985–996. ISSN 1520-9210. doi: 10.1109/TMM.2017.2759508.
207. AJAGBE, Samuel Adekunle; ADIGUN, Michael O. Deep learning techniques for detecting and predicting pandemic diseases: A systematic literature review. *Multimedia Tools and Applications*. Cham: Springer, 2023, p. 1–35. ISSN 1380-7501. <https://doi.org/10.1007/s11042-023-15805-z>
208. OGUNDOKUN, Roseline Oluwaseun; LI, A.; BABATUNDE, R. S.; UMEZURUIKE, C.; SADIKU, P. O.; ABDULAH, A. T.; BABATUNDE, A. N. Enhancing skin cancer detection and classification in dermoscopic images through concatenated MobileNetV2 and Xception models. *Bioengineering*. Basel: MDPI, 2023, vol. 10, no. 8, art. no. 979. ISSN 2306-5354. <https://doi.org/10.3390/bioengineering10080979>
209. MOHAMED, Ahmed; AMER, Ehab; ELDIN, Nour; HOSSAM, Mohamed; ELMASRY, Nader; ADNAN, Ghada T. The impact of data processing and ensemble on breast cancer detection using deep learning. *Journal of Computing and Communication*. 2022, vol. 1, no. 1, p. 27–37.
210. KUMAR, T. A.; OVIYA, S.; AJAGBE, Samuel Adekunle; ANANTH, C.; AIYENIKO, O.; ADIGUN, Michael O. A novel deep neural network-based prediction model for identifying diseases in tomato leaves. In: *Proceedings of the 2023 International Conference on Electrical, Computer and Energy Technologies (ICECET)*. Piscataway, NJ: IEEE, 2023, p. 1–6. doi: 10.1109/ICECET58911.2023.10389207.
211. MANZI, Andrea; CAVALLO, Filippo; DARIO, Paolo. A neural network approach to human posture classification and fall detection using RGB-D camera. In: *Ambient Assisted Living: Italian Forum 2016*. Cham: Springer, 2017, p. 127–139. ISBN 978-3-319-54310-9. https://doi.org/10.1007/978-3-319-54283-6_10
212. SERPEN, Gökhan; KHAN, Rafiq H. Real-time detection of human falls in progress: Machine learning approach. *Procedia Computer Science*. Amsterdam: Elsevier, 2018, vol. 140, p. 238–247. ISSN 1877-0509. <https://doi.org/10.1016/j.procs.2018.10.324>
213. REN, Long; PENG, Yong. Research of fall detection and fall prevention technologies: A systematic review. *IEEE Access*. Piscataway, NJ: IEEE, 2019, vol. 7, p. 77702–77722. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2922708.
214. HARROU, Fouzi; ZERROUKI, N.; SUN, Ying; HOUACINE, Abdelkader. An integrated vision-based approach for efficient human fall detection in a home environment. *IEEE Access*. Piscataway, NJ: IEEE, 2019, vol. 7, p. 114966–114974. doi: 10.1109/ACCESS.2019.2936320

215. TRAN, Dinh Duc; LE, Tuan Linh. Abnormal event detection using multimedia information for monitoring system. In: *Proceedings of the IEEE Fifth International Conference on Communications and Electronics (ICCE)*. Piscataway, NJ: IEEE, 2014, p. 490–495. doi: 10.1109/CCE.2014.6916753.
216. TRAN, Thi Hoai; LE, Tuan Linh; HOANG, Van Nam; VU, Huy. Continuous detection of human fall using multimodal features from Kinect sensors in scalable environment. *Computer Methods and Programs in Biomedicine*. Amsterdam: Elsevier, 2017, vol. 146, p. 151–165. ISSN 0169-2607. <https://doi.org/10.1016/j.cmpb.2017.05.007>
217. POONSRI, Apiradee; CHIRACHARIT, Wichai. Improvement of fall detection using consecutive-frame voting. In: *Proceedings of the International Workshop on Advanced Image Technology (IWAIT)*. Piscataway, NJ: IEEE, 2018, p. 1–4. doi: 10.1109/IWAIT.2018.8369696.
218. OGUNDOKUN, Roseline Oluwaseun; MASKELIŪNAS, Rytis; MISRA, Sanjay; DAMAŠEVIČIUS, Robertas. Hybrid InceptionV3-SVM-based approach for human posture detection in health monitoring systems. *Algorithms*. Basel: MDPI, 2022, vol. 15, no. 11, art. no. 410. ISSN 1999-4893. <https://doi.org/10.3390/a15110410>
219. PARDOS, Angelos; TZIOMAKA, Maria; MENYCHTAS, Andreas; MAGLOGIANNIS, Ilias. Automated posture analysis for the assessment of sports exercises. In: *Proceedings of the 12th Hellenic Conference on Artificial Intelligence*. New York, NY: ACM, 2022, p. 1–9. <https://dl.acm.org/doi/abs/10.1145/3549737.3549784>
220. JANG, Minsoo; SEO, Donghyuk; KIM, Jaeho; KIM, Jongho. Spatiotemporal algal bloom prediction of Geum River, Korea using deep learning models in combination with the EFDC model. In: *Proceedings of the 2020 Summer Simulation Conference*. San Diego, CA: Society for Modeling & Simulation International, 2020, p. 1–11. <https://dl.acm.org/doi/abs/10.5555/3427510.3427517>
221. RANSTAM, Jonas; COOK, Jonathan A. LASSO regression. *British Journal of Surgery*. Oxford: Wiley, 2018, vol. 105, no. 10, p. 1348. ISSN 0007-1323. <https://doi.org/10.1002/bjs.10895>
222. AKBARI, Amirhossein; JAFARI, Roozbeh. Transferring activity recognition models for new wearable sensors with deep generative domain adaptation. In: *Proceedings of the 18th International Conference on Information Processing in Sensor Networks*. New York, NY: ACM, 2019, p. 85–96. <https://doi.org/10.1145/3302506.3310391>
223. ORDÓÑEZ, Francisco J.; ROGGEN, Daniel. Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition. *Sensors*. Basel: MDPI, 2016, vol. 16, no. 1, art. no. 115. ISSN 1424-8220. <https://doi.org/10.3390/s16010115>
224. VARSHNEY, Nitin; BAKARIYA, Bhanu; KUSHWAHA, A. K. S. Human activity recognition using deep transfer learning of cross-position sensor data based on vertical distribution. *Multimedia Tools and Applications*. Cham: Springer, 2022, vol. 81, no. 16, p. 22307–22322. <https://doi.org/10.1007/s11042-021-11131-4>
225. KASNESIS, Panagiotis; PATRIKAKIS, Charalampos Z.; VENIERIS, Iakovos S. PerceptionNet: A deep convolutional neural network for late sensor fusion. In: *Proceedings of the SAI Intelligent Systems Conference*. Cham: Springer, 2018, p. 101–119. ISBN 978-3-319-56990-1. DOI: https://doi.org/10.1007/978-3-030-01054-6_7

CURRICULUM VITAE AND DESCRIPTION OF CREATIVE ACTIVITIES (CV)

Roseline Oluwaseun Ogundokun

Roseline.ogundokun@ktu.edu, and rosogu@ktu.lt

Education:

2010 - 2013	MSc. Computer Science, University of Ilorin, Nigeria
2004 – 2008	BSc Management Info. System, Covenant University Ota, Nigeria
1997 – 2003	WAEC School Leaving Certificate, Chapel Secondary Sch., Ilorin
2003 – 2004	WAEC School Leaving Certificate, Demonstration Sec. Sch., Zaria
1990 – 1997	First School Leaving Certificate, ARMTI Staff School, Nigeria

Professional experience:

2015 – Now	Landmark University, Omu Aran, Nigeria
2023 – 2025	Thomas Adewumi University, Oko, Nigeria
2023 – 2025	Nigeria Army College of Education, Ilorin, Nigeria
2010 – 2015	Chapel Secondary School, Ilorin, Nigeria
2008 – 2009	Access Bank Plc, Maiduguri, Borno, Nigeria

Areas of research interest:

Computer Vision, Artificial Intelligence, Deep Learning, Medical Imaging, Image Processing, Information Hiding and Security, Internet of Medical Things (IoMT), Machine Learning, and Data Mining.

Scientific papers related to the topic of the dissertation:

1. OGUNDOKUN, Roseline Oluwaseun; DAMASEVICIUS, Robertas; MASKELIUNAS, Rytis. *OptiMobileX: optimizing deep transfer learning model for accurate human posture recognition using a deep feature fusion technique*. IEEE Sensors Journal, 2025, Vol. 25, No. 6, pp. 9759–9766. DOI: 10.1109/JSEN.2024.3522105. Indexed in: Web of Science (SCIE), Scopus. Quartile: Q1 (2023).

2. OGUNDOKUN, Roseline Oluwaseun; MASKELIŪNAS, Rytis; DAMAŠEVIČIUS, Robertas. *Human posture detection using image augmentation and hyperparameter-optimized transfer learning algorithms*. Applied Sciences, 2022, Vol. 12, No. 19, Article 10156, pp. 1–23. DOI: 10.3390/app121910156. Indexed in: Web of Science (SCIE), Scopus, DOAJ. Quartile: Q2 (2022).

3. OGUNDOKUN, Roseline Oluwaseun; MASKELIŪNAS, Rytis; MISRA, Sanjay; DAMASEVICIUS, Robertas. *Hybrid InceptionV3-SVM-based approach for human posture detection in health monitoring systems*. Algorithms, 2022, Vol. 15, No. 11, Article 410, pp. 1–22. DOI: 10.3390/a15110410. Indexed in: ESCI (Web of Science), Scopus, DOAJ.

4. OGUNDOKUN, Roseline Oluwaseun; MASKELIŪNAS, Rytis; MISRA, Sanjay; DAMASEVICIUS, Robertas. *A novel deep transfer learning approach based*
200

on depth-wise separable CNN for human posture detection. Information, 2022, Vol. 13, No. 11, Article 520, pp. 1–22. DOI: 10.3390/info13110520. Indexed in: ESCI (Web of Science), Scopus, Dimensions.

Scientific conferences:

1. OGUNDOKUN, Roseline Oluwaseun; MASKELIŪNAS, Rytis; DAMAŠEVIČIUS, Robertas. *Human posture detection on lightweight DCNN and SVM in a digitalized healthcare system*. 3rd International Conference on Applied Artificial Intelligence (ICAPAI 2023), Halden, Norway, May 2, 2023. IEEE, pp. 1–6. DOI: 10.1109/ICAPAI58366.2023.10194156. Indexed in: Scopus, INSPEC.

2. OGUNDOKUN, Roseline Oluwaseun; MASKELIUNAS, Rytis; MISRA, Sanjay; DAMAŠEVIČIUS, Robertas. *Improved CNN based on Batch normalization and Adam optimizer*. ICCSA 2022 Workshops, Springer LNCS Vol. 13381, pp. 593–604. DOI: 10.1007/978-3-031-10548-7_43. Indexed in: Web of Science (CPCI-S), Scopus.

3. OGUNDOKUN, Roseline Oluwaseun; MASKELIŪNAS, Rytis; DAMAŠEVIČIUS, Robertas; ADEBIYI, Marion Olubunmi. *MobileDNN: leveraging the potential of hybrid MobileNetV2-DNN for advanced image analysis*. SEB4SDG 2024, Omu-Aran, Nigeria, April 2–4, 2024. IEEE, pp. 1–5. DOI: 10.1109/SEB4SDG60871.2024.10630298. Indexed in: Scopus.

4. OGUNDOKUN, Roseline Oluwaseun; MASKELIŪNAS, Rytis; Oscar, Frederick; OKI, Oluwakayode. *Enhancing Human Posture Detection with Hybrid Deep VGG16 and Attention Mechanism Network*. Procedia Computer Science, 2025, Vol. 258, pp. 1209–1218.

LIST OF SCIENTIFIC PAPERS AND SCIENTIFIC CONFERENCES

Scientific papers related to the topic of the dissertation:

1. OGUNDOKUN, Roseline Oluwaseun; DAMASEVICIUS, Robertas; MASKELIUNAS, Rytis. OptiMobileX: optimizing deep transfer learning model for accurate human posture recognition using a deep feature fusion technique. *IEEE Sensors Journal*. Piscataway, NJ: IEEE, 2025, vol. 25, no. 6, p. 9759–9766. ISSN 1530-437X. eISSN 1558-1748. DOI 10.1109/JSEN.2024.3522105.
2. OGUNDOKUN, Roseline Oluwaseun; MASKELIUNAS, Rytis; DAMAŠEVIČIUS, Robertas. Human posture detection using image augmentation and hyperparameter-optimized transfer learning algorithms. *Applied Sciences*. Basel: MDPI, 2022, vol. 12, no. 19, art. 10156, p. 1–23. ISSN 2076-3417. DOI 10.3390/app121910156.
3. OGUNDOKUN, Roseline Oluwaseun; MASKELIUNAS, Rytis; MISRA, Sanjay; DAMASEVICIUS, Robertas. Hybrid InceptionV3-SVM-based approach for human posture detection in health monitoring systems. *Algorithms*. Basel: MDPI, 2022, vol. 15, no. 11, art. 410, p. 1–22. ISSN 1999-4893. DOI 10.3390/a15110410.
4. OGUNDOKUN, Roseline Oluwaseun; MASKELIUNAS, Rytis; MISRA, Sanjay; DAMASEVICIUS, Robertas. A novel deep transfer learning approach based on depth-wise separable CNN for human posture detection. *Information*. Basel: MDPI, 2022, vol. 13, no. 11, art. 520, p. 1–22. ISSN 2078-2489. DOI 10.3390/info13110520.

Scientific conferences:

1. OGUNDOKUN, Roseline Oluwaseun; MASKELIUNAS, Rytis; DAMAŠEVIČIUS, Robertas. Human posture detection on lightweight DCNN and SVM in a digitalized healthcare system. In: *Proceedings of the 3rd International Conference on Applied Artificial Intelligence (ICAPAI 2023)*. Halden, Norway, 2 May 2023. Piscataway, NJ: IEEE, 2023, p. 1–6. ISBN 9798350328936. DOI 10.1109/ICAPAI58366.2023.10194156.
2. OGUNDOKUN, Roseline Oluwaseun; MASKELIUNAS, Rytis; MISRA, Sanjay; DAMAŠEVIČIUS, Robertas. Improved CNN based on batch normalization and Adam optimizer. In: GERVASI, O. et al., eds. *Computational Science and Its Applications – ICCSA 2022 Workshops*. Lecture Notes in Computer Science, vol. 13381. Cham: Springer, 2022, p. 593–604. ISBN 978-3-031-10547-0. DOI 10.1007/978-3-031-10548-7_43.
3. OGUNDOKUN, Roseline Oluwaseun; MASKELIUNAS, Rytis; DAMAŠEVIČIUS, Robertas; ADEBIYI, Marion Olubunmi. MobileDNN: leveraging the potential of hybrid MobileNetV2-DNN for advanced image analysis. In: *Proceedings of the International Conference on Science, Engineering and Business for Driving Sustainable Development Goals (SEB4SDG 2024)*. Omu-Aran, Nigeria, 2–4 April 2024. Piscataway, NJ: IEEE, 2024, p. 1–5. ISBN 9798350358162. DOI 10.1109/SEB4SDG60871.2024.10630298.

4. OGUNDOKUN, Roseline Oluwaseun; MASKELIŪNAS, Rytis; Oscar, Frederick; OKI, Oluwakayode. *Enhancing Human Posture Detection with Hybrid Deep VGG16 and Attention Mechanism Network*. Procedia Computer Science, 2025, Vol. 258, pp. 1209–1218.

ACKNOWLEDGEMENT

I wish to express thanks to God Almighty for providing me with the vitality, wit, and resilience needed to complete this dissertation.

Sincere gratitude to my supervisors, Prof. Dr Rytis Maskeliūnas, whose relentless work, insightful counsel, and constructive criticism were crucial in shaping this dissertation. His guidance has helped shape my intellectual growth and refine my research skills.

I particularly like to thank Prof. Dr Robertas Damaševičius for his encouragement, constructive ideas, and scientific comments, which were a great contribution to this dissertation.

I am greatly indebted to the Research Council of Lithuania for sponsoring this doctoral research and to the management of the Kaunas University of Technology (KTU) for providing an enabling environment and resources that facilitated the success of this study. I would like to extend my special thanks to my colleagues and friends in the Department of Software Engineering for their encouragement, helpful discussions, and moral support.

To my family, specifically my parents, Mr and Mrs S. O. Afolayan, my brothers, Mr Habakkuk Afolayan and Mr Daniel Afolayan, my husband, Mr Opeyemi Ogundokun, my children, Amaris Oluwadamife Ogundokun and Elnathan Oluwadamike Ogundokun and my dear friend, Dr Rotimi-Williams Bello, I extend special thanks for their love, prayers, patience, and understanding while I spent innumerable hours on this project. Your confidence in me has been my ever-present source of inspiration.

I sincerely appreciate my mentor, Prof. Sanjay Misra, and all my collaborators for their invaluable guidance, steadfast support, and meaningful contributions to this work and to my academic growth.

Finally, I extend thanks to all those, too numerous to count, who directly or indirectly contributed to the success of this dissertation. Your kindness and generosity will not be forgotten.

APPENDICES

Appendix 1. Code

```
import os
import random
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import cv2
from tensorflow import keras
import tensorflow as tf
import keras.backend as K
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import InputLayer, BatchNormalization,
Dropout, Flatten, Dense, Activation, MaxPool2D, Conv2D
from tensorflow.keras.layers import Conv2D, Dense, Flatten, Dropout,
Activation, SimpleRNN
from tensorflow.keras.layers import BatchNormalization, Reshape,
MaxPooling2D, GlobalAveragePooling2D
from tensorflow.keras.callbacks import EarlyStopping,
ModelCheckpoint, ReduceLROnPlateau

from PIL import Image
from PIL import UnidentifiedImageError

from sklearn.preprocessing import OneHotEncoder

from sklearn.model_selection import train_test_split

from sklearn.metrics import classification_report, confusion_matrix,
roc_curve, auc
import itertools
from sklearn.preprocessing import StandardScaler

import warnings
warnings.filterwarnings('ignore')
```

```

from sklearn import metrics
from sklearn.preprocessing import label_binarize
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import MinMaxScaler

```

```

import os
import random

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import cv2

from tensorflow import keras
import tensorflow as tf
import keras.backend as K

from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import InputLayer, BatchNormalization,
Dropout, Flatten, Dense, Activation, MaxPool2D, Conv2D
from tensorflow.keras.layers import Conv2D, Dense, Flatten, Dropout,
Activation, SimpleRNN
from tensorflow.keras.layers import BatchNormalization, Reshape,
MaxPooling2D, GlobalAveragePooling2D
from tensorflow.keras.callbacks import EarlyStopping,
ModelCheckpoint, ReduceLROnPlateau

from PIL import Image
from PIL import UnidentifiedImageError

from sklearn.preprocessing import OneHotEncoder

from sklearn.model_selection import train_test_split

from sklearn.metrics import classification_report, confusion_matrix,
roc_curve, auc
import itertools
from sklearn.preprocessing import StandardScaler

```

```
import warnings
warnings.filterwarnings('ignore')
from sklearn import metrics
from sklearn.preprocessing import label_binarize
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.preprocessing import MinMaxScaler
```

```
from keras.layers import Input, Lambda, Dense, Flatten
from keras.models import Model
from keras.applications.vgg16 import VGG16
from keras.applications.vgg16 import preprocess_input
from keras.preprocessing import image
#from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
import numpy as np
from glob import glob
import matplotlib.pyplot as plt

import tensorflow as tf
from tensorflow.keras.layers import Input, Lambda, Dense, Flatten
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras.applications.vgg16 import VGG16,
preprocess_input
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import numpy as np
from glob import glob
import matplotlib.pyplot as plt
```

```
#upright = os.listdir(r'/content/drive/MyDrive/posture/Upright
Sitting')
#Crossed_legs = os.listdir(r'/content/drive/MyDrive/posture/Crossing
legs')
#slant_bending = os.listdir(r'/content/drive/MyDrive/posture/Slant
Bending')
#Lordosis = os.listdir(r'/content/drive/MyDrive/posture/Lordosis')
```

```
#slouching = os.listdir(r'/content/drive/MyDrive/posture/Slouching')

#Sitting_Posture =
os.listdir('C:/Users/Dell/Desktop/Sitting_Posture')
bending = os.listdir(r"C:\Users\roseo\OneDrive - Kaunas University
of Technology\Dokumentai\KTU\Dissertation\Final Thesis\Final\New
Documents\Final Document for Review\Dataset\Silhouettes\bending")
lying = os.listdir(r"C:\Users\roseo\OneDrive - Kaunas University of
Technology\Dokumentai\KTU\Dissertation\Final Thesis\Final\New
Documents\Final Document for Review\Dataset\Silhouettes\lying")
sitting = os.listdir(r"C:\Users\roseo\OneDrive - Kaunas University
of Technology\Dokumentai\KTU\Dissertation\Final Thesis\Final\New
Documents\Final Document for Review\Dataset\Silhouettes\sitting")
standing = os.listdir(r"C:\Users\roseo\OneDrive - Kaunas University
of Technology\Dokumentai\KTU\Dissertation\Final Thesis\Final\New
Documents\Final Document for Review\Dataset\Silhouettes\standing")
```

```
img_w, img_h = 150, 150 #setting the image width and height for easy
processing
X = []
y = []

for i in bending:
    try:
        img = Image.open(r"C:\Users\roseo\OneDrive - Kaunas
University of Technology\Dokumentai\KTU\Dissertation\Final
Thesis\Final\New Documents\Final Document for
Review\Dataset\Silhouettes\bending/" + i).convert('RGB')
        img = img.resize((img_w, img_h))
        X.append(np.asarray(img))
        y.append(0)
    except UnidentifiedImageError: # it passes an image that wasn't
correctly identified
        pass

for i in lying:
    try:
```

```

        img = Image.open(r"C:\Users\roseo\OneDrive - Kaunas
University of Technology\Dokumentai\KTU\Dissertation\Final
Thesis\Final\New Documents\Final Document for
Review\Dataset\Silhouettes\lying/"+ i).convert('RGB')
        img = img.resize((img_w,img_h))
        X.append(np.asarray(img))
        y.append(1)
    except UnidentifiedImageError:
        pass

for i in sitting:
    try:
        img = Image.open(r"C:\Users\roseo\OneDrive - Kaunas
University of Technology\Dokumentai\KTU\Dissertation\Final
Thesis\Final\New Documents\Final Document for
Review\Dataset\Silhouettes\sitting/" + i).convert('RGB')
        img = img.resize((img_w,img_h))
        X.append(np.asarray(img))
        y.append(2)
    except UnidentifiedImageError:
        pass

for i in standing:
    try:
        img = Image.open(r"C:\Users\roseo\OneDrive - Kaunas
University of Technology\Dokumentai\KTU\Dissertation\Final
Thesis\Final\New Documents\Final Document for
Review\Dataset\Silhouettes\standing/" + i).convert('RGB')
        img = img.resize((img_w,img_h))
        X.append(np.asarray(img))
        y.append(3)
    except UnidentifiedImageError:
        pass

X = np.asarray(X)
y = np.asarray(y)

```

```
print(X.shape, y.shape)
```

```
# I resized each image to our manually defined width and height(img_w, img_h)  
# I also changed the images to an array
```

```
# splitting our dataset into train and test  
x_train1, x_test, y_train1, y_test = train_test_split(x, labels,  
test_size = 0.1, random_state=5)  
x_train,x_val,y_train,y_val=train_test_split(x_train1,y_train1,test_  
size=0.1,random_state=5)  
  
print('Number of train: {}'.format(len(x_train)))  
print('Number of validation: {}'.format(len(x_val)))  
print('Number of test: {}'.format(len(x_test)))
```

```
datagen = ImageDataGenerator(  
    rotation_range=20.,  
    width_shift_range=0.3,  
    height_shift_range=0.3,  
    shear_range=30.,  
    zoom_range=0.2,  
    horizontal_flip=0.2,  
    rescale=None)  
  
datagen.fit(x_train)
```

```
ICV1_2 =  
tf.keras.applications.ResNet50V2(input_shape=(150,150,3),include_top  
=False,weights="imagenet")  
for layer in ICV1_2.layers[:180]:  
    layer.trainable=False  
for layer in ICV1_2.layers[180:]:  
    layer.trainable=True
```

```
# My neural network consists of 3 layers densely connected  
# dropout helps prevent over fitting of the model when testing it
```

```

model_Inception=Sequential()
model_Inception.add(ICV1_2)
model_Inception.add(Flatten())
model_Inception.add(Dense(128, activation='relu'))
model_Inception.add(Dropout(0.5))
model_Inception.add(Dense(128, activation='relu'))
model_Inception.add(Dropout(0.5))
model_Inception.add(Dense(4,
kernel_regularizer=tf.keras.regularizers.l2(0.01),
activation='linear'))
model_Inception.summary()

```

here i am defining my performance metrics to check the performance of the model

```

def f1_score(y_true, y_pred):
    true_positives = K.sum(K.round(K.clip(y_true * y_pred, 0, 1)))
    possible_positives = K.sum(K.round(K.clip(y_true, 0, 1)))
    predicted_positives = K.sum(K.round(K.clip(y_pred, 0, 1)))
    precision = true_positives / (predicted_positives + K.epsilon())
    recall = true_positives / (possible_positives + K.epsilon())
    f1_val = 2*(precision*recall)/(precision+recall+K.epsilon())
    return f1_val

```

```

METRICS = [
    tf.keras.metrics.CategoricalAccuracy(name='accuracy'),
    tf.keras.metrics.Precision(name='precision'),
    tf.keras.metrics.Recall(name='recall'),
    tf.keras.metrics.AUC(name='auc'),
    f1_score,
]

```

```

lrd = ReduceLROnPlateau(monitor = 'recall',patience = 10,verbose =
1,factor = 0.70, min_lr = 1e-5)
# the above line of code reduces the learning rate when there is no
improvement to the metric
# i added this line of code because i trained the model without it
and the performance was stagnant
# the performance was stuck on 0.96nnn as the accuracy so i added
the line of code to reduce
# the learing rate after two epochs

```

```
mcp = ModelCheckpoint('model.h5')
es = EarlyStopping(verbose=1, patience=20)

model_Inception.compile(optimizer='adam', loss = 'hinge', metrics =
['accuracy'])
```

```
# == Add-ons: params/size, per-epoch timing, total time, Effective
FLOPS (train & inference) ==
from tensorflow.keras.regularizers import l2
from tensorflow.keras.callbacks import LearningRateScheduler,
EarlyStopping
import time, numpy as np, tensorflow as tf
from tensorflow.python.framework.convert_to_constants import
convert_variables_to_constants_v2

# ---- Epoch timer callback ----
class EpochTimer(tf.keras.callbacks.Callback):
    def on_train_begin(self, logs=None):
        self.epoch_times = []
        self.total_train_time = 0.0
        self.total_samples = 0
        self._fit_start = time.time()
    def on_epoch_begin(self, epoch, logs=None):
        self._epoch_start = time.time()
    def on_train_batch_end(self, batch, logs=None):
        if logs is not None and "size" in logs and logs["size"] is
not None:
            self.total_samples += int(logs["size"])
    def on_epoch_end(self, epoch, logs=None):
        self.epoch_times.append(time.time() - self._epoch_start)
    def on_train_end(self, logs=None):
        self.total_train_time = time.time() - self._fit_start

epoch_timer = EpochTimer()

# ---- Fit with timing callback (keeps your existing callbacks) ----
fit_start = time.time()
history = model_Inception.fit(
    x_train, y_train,
```

```

        validation_data=(x_val, y_val),
        epochs=50, verbose=1,
        callbacks=[lrd, mcp, es, epoch_timer]
    )
fit_wall = time.time() - fit_start

# ---- Parameters & approximate model size (MB, assuming float32
weights) ----
total_params = model_Inception.count_params()
print(f"\nParameters (Millions): {total_params/1e6:.3f}")
print(f"Approx. Model Size: {total_params*4/(1024**2):.2f} MB")

# ---- Training time over epoch (min) & total (hrs) ----
if getattr(epoch_timer, "epoch_times", None):
    print("\nTraining Time over Epoch (min):")
    for i, sec in enumerate(epoch_timer.epoch_times, 1):
        print(f"  Epoch {i}: {sec/60.0:.3f} min")
print(f"Total Training Time (hrs):
{epoch_timer.total_train_time/3600.0:.3f} hrs")
print(f"(Wall-clock fit() time: {fit_wall/3600.0:.3f} hrs)")

# ---- FLOPs helpers (best-effort) ----
def _infer_input_hw_c(keras_model):
    shp = keras_model.input_shape
    if isinstance(shp, (list, tuple)) and isinstance(shp[0], tuple):
        shp = shp[0]
    H, W, C = shp[1], shp[2], shp[3]
    return int(H), int(W), int(C)

def estimate_flops_per_inference(keras_model):
    try:
        H, W, C = _infer_input_hw_c(keras_model)
        @tf.function
        def _f(x):
            return keras_model(x, training=False)
        concrete = _f.get_concrete_function(tf.TensorSpec(shape=(1,
H, W, C), dtype=tf.float32))
        frozen = convert_variables_to_constants_v2(concrete)
        graph_def = frozen.graph.as_graph_def()

```

```

        with tf.compat.v1.Graph().as_default() as g:
            tf.compat.v1.import_graph_def(graph_def, name="")
            opts = (tf.compat.v1.profiler.ProfileOptionBuilder
                    .float_operation()
                    .with_empty_output()
                    .build())
            flops = tf.compat.v1.profiler.profile(g, options=opts)
            return float(flops.total_float_ops) if flops is not None
    else None
    except Exception as e:
        print(f"FLOPs estimation failed: {e}")
        return None

def effective_flops_inference(keras_model, runs=10, batch_size=16):
    flops_per_sample = estimate_flops_per_inference(keras_model)
    if not flops_per_sample:
        print("Effective FLOPS (Inference): unavailable.")
        return None
    H, W, C = _infer_input_hw_c(keras_model)
    dummy = np.random.randn(batch_size, H, W, C).astype("float32")
    _ = keras_model.predict(dummy, verbose=0) # warm-up
    t0 = time.time()
    for _ in range(runs):
        _ = keras_model.predict(dummy, verbose=0)
    elapsed = time.time() - t0
    samples_per_sec = (batch_size * runs) / max(elapsed, 1e-9)
    eff = flops_per_sample * samples_per_sec
    print(f"Effective FLOPS (Inference): {eff/1e9:.3f} GFLOP/s "
          f"(~{flops_per_sample/1e9:.3f} GFLOPs/sample × "
          f"{samples_per_sec:.2f} samp/s)")
    return eff

def effective_flops_training(keras_model, timer: EpochTimer):
    flops_inf = estimate_flops_per_inference(keras_model)
    if (not flops_inf) or timer.total_train_time <= 0 or
    timer.total_samples == 0:
        print("Effective FLOPS (Training): unavailable.")
        return None

```

```

    flops_train_per_sample = 2.0 * flops_inf # forward + backward
approx
    samples_per_sec = timer.total_samples / timer.total_train_time
    eff = flops_train_per_sample * samples_per_sec
    print(f"Effective FLOPS (Training): {eff/1e9:.3f} GFLOP/s "
          f"(~{flops_train_per_sample/1e9:.3f} GFLOPs/sample × "
          f"{samples_per_sec:.2f} samp/s)")
    return eff

# ---- Report Effective FLOPS ----
_ = effective_flops_inference(model_Inception)
_ = effective_flops_training(model_Inception, epoch_timer)

```

```

Test_data = model_Inception.evaluate(x_test, y_test)

```

```

import matplotlib.pyplot as plt

# Get metrics
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']
epochs_range = range(len(val_acc))

# Reset to default style (white background)
plt.style.use("default")

# Create figure with 300 dpi
plt.figure(figsize=(10, 4), dpi=300)

# ---- Loss subplot ----
plt.subplot(1, 2, 1)
plt.plot(loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.title('ResNet50V2 Model Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')

```

```
# ---- Accuracy subplot ----
plt.subplot(1, 2, 2)
plt.plot(acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.title('ResNet50V2 Model Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')

plt.tight_layout()
plt.show()
```

```
import itertools
import matplotlib.pyplot as plt
import numpy as np
from sklearn.metrics import confusion_matrix, classification_report

# --- Confusion Matrix Plot Function ---
def plot_confusion_matrix(cm, classes, normalize=False,
title='ResNet50V2 Model', cmap=plt.cm.YlGnBu):
    """
    cm: confusion matrix
    classes: list of class names
    normalize: whether to normalize values (default=False for raw
counts)
    """
    plt.figure(figsize=(4, 4), dpi=300) # 300 dpi figure
    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    # Normalize only if requested
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        cm = np.around(cm, decimals=2)
```

```

        cm[np.isnan(cm)] = 0.0

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]),
range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                horizontalalignment="center",
                color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True Class')
    plt.xlabel('Predicted Class')
    plt.show()

# --- Predictions and Reports ---
y_pred = (model_Inception.predict(x_test) > 0.5).astype("int32")

y_test_c = np.argmax(y_test, axis=1)
target_names = ["bending", "lying", "sitting", "standing"]

Y_pred = np.argmax(model_Inception.predict(x_test), axis=1)

print('ResNet50V2 Model')
cm = confusion_matrix(y_test_c, Y_pred)
plot_confusion_matrix(cm, target_names, normalize=False,
title='ResNet50V2 Model')

print('\nClassification Report')
print(classification_report(y_test_c, Y_pred,
target_names=target_names))

```

```

import numpy as np
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt

# Class labels
class_names = ["bending", "lying", "sitting", "standing"]

# Get the predicted probabilities for the test set

```

```

y_pred_prob = model_Inception.predict(x_test)

# True labels (one-hot → integer class labels)
y_test_c = np.argmax(y_test, axis=1)

# Compute FPR, TPR, thresholds for each class
fpr, tpr, thresh, roc_auc = {}, {}, {}, {}
n_classes = len(class_names)

for i in range(n_classes):
    fpr[i], tpr[i], thresh[i] = roc_curve(y_test[:, i],
y_pred_prob[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

# --- Plot ROC curves ---
plt.style.use("default") # reset style → white background
plt.figure(figsize=(5, 4), dpi=300)

for i in range(n_classes):
    plt.plot(
        fpr[i], tpr[i],
        label=f"ROC curve for {class_names[i]} (AUC =
{roc_auc[i]:0.2f})"
    )

# Random guessing diagonal
plt.plot([0, 1], [0, 1], 'k--', label='Random Guessing')

# Labels and formatting
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ResNet50V2 ROC Curve')
plt.legend(loc="lower right", fontsize=8)
plt.tight_layout()
plt.show()

```

```

import numpy as np
from sklearn.metrics import precision_recall_curve,
average_precision_score
import matplotlib.pyplot as plt

# Define your class names
class_names = ["bending", "lying", "sitting", "standing"]

# Get the predicted probabilities for the test set
y_pred_prob = model_Inception.predict(x_test)

# True labels (already one-hot encoded in y_test)
n_classes = len(class_names)

# Compute Precision-Recall and Average Precision for each class
precision, recall, thresh, avg_precision = {}, {}, {}, {}
for i in range(n_classes):
    precision[i], recall[i], thresh[i] =
precision_recall_curve(y_test[:, i], y_pred_prob[:, i])
    avg_precision[i] = average_precision_score(y_test[:, i],
y_pred_prob[:, i])

# --- Plot Precision-Recall curves ---
plt.style.use("default") # white background
plt.figure(figsize=(5, 4), dpi=300)

for i in range(n_classes):
    plt.plot(
        recall[i], precision[i],
        label=f"{class_names[i]} (AP = {avg_precision[i]:0.2f})"
    )

# Axis limits and labels
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('Recall')
plt.ylabel('Precision')
plt.title('ResNet50V2 Precision-Recall Curve')
plt.legend(loc="lower left", fontsize=8)

```

```
plt.tight_layout()
plt.show()
```

```
# DenseNet121 + Kernel SVM (RBF) using your x_train/x_val/x_test
# splits
import os, time
import numpy as np
import tensorflow as tf
from tensorflow.keras import Model, Input
from tensorflow.keras.layers import GlobalAveragePooling2D
from tensorflow.keras.applications import DenseNet121
from tensorflow.keras.applications.densenet import preprocess_input
from tensorflow.keras.regularizers import
l2 # <-- added
from tensorflow.keras.callbacks import LearningRateScheduler,
EarlyStopping # <-- added
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC
from sklearn.metrics import classification_report, confusion_matrix
from tensorflow.python.framework.convert_to_constants import
convert_variables_to_constants_v2 # <-- for FLOPs

# ----- Config -----
SEED = 42
tf.keras.utils.set_random_seed(SEED)
IMG_SIZE = (224, 224)
BATCH = 32

# y to integer labels if one-hot
def to_int_labels(y):
    y = np.asarray(y)
    if y.ndim > 1 and y.shape[-1] > 1:
        y = np.argmax(y, axis=1)
    return y.astype(int)

y_train_i = to_int_labels(y_train)
y_val_i = to_int_labels(y_val)
```

```

y_test_i = to_int_labels(y_test)

# Class names (optional): if you have a mapping, replace this
num_classes = int(np.max([y_train_i.max(), y_val_i.max(),
y_test_i.max()])) + 1
class_names = [f"class_{i}" for i in range(num_classes)]

# ----- DenseNet feature extractor -----
base = DenseNet121(weights="imagenet", include_top=False,
input_shape=IMG_SIZE + (3,))
base.trainable = False

inp = Input(shape=IMG_SIZE + (3,))
x = base(inp, training=False)
gap = GlobalAveragePooling2D()(x)          # DenseNet121 GAP ->
1024-D
feature_model = Model(inp, gap, name="densenet121_gap")

# ----- FLOPs estimator for the feature extractor (per
image) -----
def estimate_flops_per_forward(keras_model, h=224, w=224, c=3):
    try:
        @tf.function
        def _f(x):
            return keras_model(x, training=False)
        concrete = _f.get_concrete_function(tf.TensorSpec(shape=(1,
h, w, c), dtype=tf.float32))
        frozen = convert_variables_to_constants_v2(concrete)
        graph_def = frozen.graph.as_graph_def()

        with tf.compat.v1.Graph().as_default() as g:
            tf.compat.v1.import_graph_def(graph_def, name="")
            opts = (tf.compat.v1.profiler.ProfileOptionBuilder
                    .float_operation()
                    .with_empty_output()
                    .build())
            flops = tf.compat.v1.profiler.profile(g, options=opts)
            return float(flops.total_float_ops) if flops is not None
    else None

```

```

    except Exception as e:
        print(f"[FLOPs] estimation failed: {e}")
        return None

FLOPs_PER_IMAGE = estimate_flops_per_forward(feature_model,
*IMG_SIZE, 3)
if FLOPs_PER_IMAGE:
    print(f"Estimated FLOPs per image (feature extractor):
{FLOPs_PER_IMAGE/1e9:.3f} GFLOPs")

# ----- Preprocess & feature extraction -----
---
def extract_features_np(x_np):
    """Resize, preprocess for DenseNet, then extract GAP
features."""
    t0 = time.time()
    x_tf = tf.convert_to_tensor(x_np, dtype=tf.float32)      #
expects [N,H,W,3]
    x_tf = tf.image.resize(x_tf, IMG_SIZE)                  #
[N,224,224,3]
    x_tf = preprocess_input(x_tf)
    feats = feature_model.predict(x_tf, batch_size=BATCH, verbose=1)
    elapsed = time.time() - t0
    return feats, elapsed

t_fe_train = 0.0
X_train, t_part = extract_features_np(x_train); t_fe_train += t_part
X_val, _ = extract_features_np(x_val)
X_test, _ = extract_features_np(x_test)

print("Feature shapes:", X_train.shape, X_val.shape,
X_test.shape) # e.g., (N, 1024)

# ----- Kernel SVM (RBF) -----
svm_clf = Pipeline([
    ("scaler", StandardScaler(with_mean=True, with_std=True)),
    ("svc", SVC(kernel="rbf", C=10.0, gamma="scale",
class_weight="balanced", probability=True,
random_state=SEED))

```

```

])

t0_svm = time.time()
svm_clf.fit(X_train, y_train_i)
t_svm_fit = time.time() - t0_svm

# ----- Evaluation -----
def eval_split(name, X, y):
    y_pred = svm_clf.predict(X)
    print(f"\n=== {name} results ===")
    print(classification_report(y, y_pred, target_names=class_names,
                                digits=4))
    print("Confusion matrix:\n", confusion_matrix(y, y_pred))

eval_split("Validation", X_val, y_val_i)
eval_split("Test", X_test, y_test_i)

# ----- (Optional) Save models -----
# TF feature extractor:
feature_path = "densenet121_gap_extractor.keras"
feature_model.save(feature_path)

# SVM (requires joblib):
svm_path = None
try:
    import joblib
    svm_path = "svm_rbf_on_densenet121_gap.joblib"
    joblib.dump(svm_clf, svm_path)
except Exception as e:
    print("Skipping SVM save (joblib not available):", e)

# ===== Added metrics & reporting =====
# Parameters (Millions) & approximate in-memory size (float32)
total_params = feature_model.count_params()
params_millions = total_params / 1e6
approx_mem_mb = total_params * 4 / (1024**2)  # 4 bytes/param

print(f"\nParameters (Millions): {params_millions:.3f} M")

```

```

print(f"Approx. Feature Extractor Size (memory): {approx_mem_mb:.2f}
MB")

# On-disk sizes (feature extractor and SVM)
def file_mb(p):
    return (os.path.getsize(p) / (1024**2)) if (p and
os.path.exists(p)) else 0.0

feature_mb = file_mb(feature_path)
svm_mb      = file_mb(svm_path)
print(f"Saved Feature Extractor file size: {feature_mb:.2f} MB")
if svm_path:
    print(f"Saved SVM file size: {svm_mb:.2f} MB")
    print(f"Total saved model size (feature+svm): {feature_mb +
svm_mb:.2f} MB")

# Training time over epoch (min) & total (hrs)
# Here we have a single "epoch": (feature extraction on train) +
(SVM fit)
epoch_time_sec = t_fe_train + t_svm_fit
print("\nTraining Time over Epoch (min):")
print(f"  Epoch 1: {epoch_time_sec/60.0:.3f} min")
print(f"    - Feature extraction (train) : {t_fe_train/60.0:.3f}
min")
print(f"    - SVM fit                      : {t_svm_fit/60.0:.3f} min")
print(f"Total Training Time (hrs): {epoch_time_sec/3600.0:.3f} hrs")

# Effective FLOPS (Training)
# Since the feature extractor is frozen (no backprop), training
FLOPs ~ forward FLOPs × #train_images.
if FLOPs_PER_IMAGE:
    n_train = len(X_train)
    total_train_flops = FLOPs_PER_IMAGE * n_train
    eff_flops_training = total_train_flops / max(epoch_time_sec, 1e-
9)

    print(f"\nEffective FLOPS (Training):
{eff_flops_training/1e9:.3f} GFLOP/s "
        f"(~{(total_train_flops)/1e9:.3f} GFLOPs over train set)")

```

```

# Effective FLOPS (Inference)
# Measure mean per-sample latency for feature extractor and convert
via FLOPs_PER_IMAGE.
if FLOPs_PER_IMAGE:
    # Use a small random subset of X_test (raw images), re-run a
    quick predict for timing
    n_bench = min(64, len(x_test))
    x_bench = x_test[:n_bench]
    x_tf = tf.convert_to_tensor(x_bench, dtype=tf.float32)
    x_tf = tf.image.resize(x_tf, IMG_SIZE)
    x_tf = preprocess_input(x_tf)

    # Warm-up
    _ = feature_model.predict(x_tf[:min(8, n_bench)],
batch_size=min(8, n_bench), verbose=0)
    t0 = time.time()
    _ = feature_model.predict(x_tf, batch_size=32, verbose=0)
    elapsed = time.time() - t0
    mean_time_per_sample = elapsed / max(n_bench, 1)
    eff_flops_infer = FLOPs_PER_IMAGE / max(mean_time_per_sample,
1e-9)
    print(f"Effective FLOPS (Inference): {eff_flops_infer/1e9:.3f}
GFLOP/s "
        f"(mean {mean_time_per_sample*1e3:.2f} ms/sample)")

#
=====
=

```

```

# ===== Confusion Matrix, ROC, and PR curves for DenseNet121 + SVM
=====
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, roc_curve, auc,
precision_recall_curve, average_precision_score
from sklearn.preprocessing import label_binarize

# Assumes you already have:

```

```

# svm_clf    -> trained Pipeline([...("svc",
SVC(probability=True))])
# X_test     -> test features from DenseNet121 GAP (shape: [N,
1024])
# y_test_i   -> integer test labels (shape: [N])
class_names = ["bending", "lying", "sitting", "standing"]

# If your y_test is one-hot, convert to ints:
y_true = np.asarray(y_test_i)
num_classes = len(class_names)
labels = np.arange(num_classes)

# Predict probabilities & labels
y_prob = svm_clf.predict_proba(X_test)           # shape [N, C]
y_pred = np.argmax(y_prob, axis=1)

```

```

# ----- Confusion Matrix (300 dpi) -----
cm = confusion_matrix(y_true, y_pred, labels=labels)
try:
    import seaborn as sns
    fig, ax = plt.subplots(figsize=(4, 4), dpi=300)
    sns.heatmap(cm, annot=True, fmt="d", cmap="YlGnBu",
                xticklabels=class_names, yticklabels=class_names,
ax=ax)
    ax.set_xlabel("Predicted")
    ax.set_ylabel("True")
    ax.set_title("DeneSVM Model")
    fig.tight_layout()
    plt.show()
except Exception:
    fig, ax = plt.subplots(figsize=(5, 4), dpi=300)
    im = ax.imshow(cm, cmap="Blues")
    ax.set_xticks(labels); ax.set_yticks(labels)
    ax.set_xticklabels(class_names, rotation=45, ha="right")
    ax.set_yticklabels(class_names)
    for i in range(num_classes):
        for j in range(num_classes):

```

```

        ax.text(j, i, cm[i, j], ha="center", va="center",
color="black")
    ax.set_xlabel("Predicted"); ax.set_ylabel("True")
    ax.set_title("DeneSVM Model")
    fig.colorbar(im, ax=ax, fraction=0.046, pad=0.04)
    fig.tight_layout()
    plt.show()

# ----- ROC (One-vs-Rest, 300 dpi) -----
y_true_bin = label_binarize(y_true, classes=labels) # shape [N, C]
fpr, tpr, roc_auc = {}, {}, {}

plt.figure(figsize=(5, 4), dpi=300)
for c in range(num_classes):
    fpr[c], tpr[c], _ = roc_curve(y_true_bin[:, c], y_prob[:, c])
    roc_auc[c] = auc(fpr[c], tpr[c])
    plt.plot(fpr[c], tpr[c], label=f"{class_names[c]}
(AUC={roc_auc[c]:.2f})")

plt.plot([0, 1], [0, 1], 'k--', lw=1)
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("DeneSVM Model ROC Curve")
plt.legend(loc="lower right", fontsize=8)
plt.grid(True, alpha=0.3)
plt.tight_layout()
plt.show()

```

UDK 616.8-009.18-079:004.932+004.8](043.3)

SL344. 2026-03-17, 28,5 leidyb. apsk. I. Tiražas 14 egz. Užsakymas 46.
Išleido Kauno technologijos universitetas, K. Donelaičio g. 73, 44249 Kaunas
Spausdino leidyklos „Technologija“ spaustuvė, Studentų g. 54, 51424 Kaunas

