Kaunas University of Technology

Faculty of Informatics

Person re-identification algorithms in video analysis

Master's Final Degree Project

Jokūbas Šakinis

Project author

Prof. Dr. Andrius Kriščiūnas

Supervisor

Kaunas, 2026

Kaunas University of Technology

Faculty of Informatics

Person re-identification algorithms in video analysis

Master's Final Degree Project

Artificial Intelligence in Computer Science (6211BX007)

Jokūbas Šakinis

Project author

Prof. Dr. Andrius Kriščiūnas

Supervisor

Doc. Dr. Liudas Motiejūnas

Reviewer

Kaunas, 2026

Kaunas University of Technology

Faculty of Informatics

Jokūbas Šakinis

Person re-identification algorithms in video analysis

Declaration of Academic Integrity

I confirm the following:

1. I have prepared the final degree project independently and honestly without any violations of the copyrights or other rights of others, following the provisions of the Law on Copyrights and Related Rights of the Republic of Lithuania, the Regulations on the Management and Transfer of Intellectual Property of Kaunas University of Technology (hereinafter – University) and the ethical requirements stipulated by the Code of Academic Ethics of the University;

2. All the data and research results provided in the final degree project are correct and obtained legally; none of the parts of this project are plagiarised from any printed or electronic sources; all the quotations and references provided in the text of the final degree project are indicated in the list of references;

3. I have not paid anyone any monetary funds for the final degree project or the parts thereof unless required by the law;

4. I understand that in the case of any discovery of the fact of dishonesty or violation of any rights of others, the academic penalties will be imposed on me under the procedure applied at the University; I will be expelled from the University and my final degree project can be submitted to the Office of the Ombudsperson for Academic Ethics and Procedures in the examination of a possible violation of academic ethics.

Jokūbas Šakinis

Confirmed electronically

Šakinis, Jokūbas. Person re-identification algorithms in video analysis. Master's Final Degree Project / Supervisor Prof. Dr. Andrius Kriščiūnas; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Computer Science, Informatics (B01)

Keywords: Artificial intelligence, person re-identification, person re-id, sequence encoder, TSCL.

Kaunas, 2026. 57p.

Summary

In this research two different deep-learning person re-identification models are presented with their implementation, trained on three benchmark datasets: IUST_PersonReID, Market-1501, MARS. Market-1501 and MARS datasets were split according to the official data split for results to be comparable to existing research and an unofficial custom cross-camera split without junk to display best-case scenario how models would perform using clean data.

First implementation trained on IUST_PersonReID dataset uses a GooGleNet Inception v1 backbone with weighted contrastive loss consisting of Gaussian-scored positive pair, margin-based negative pairs, identity-agnostic attention.

Second implementation was trained on Market-1501 and MARS datasets separately producing two models using a DINOv2 feature extraction backbone with a combined loss consisting of AMSoftmax, circle loss, batch-hard triplet loss and a novel suggested temporal stripe consistency loss consisting of intra-frame and inter-frame components making sure horizontal stripe features are consistent across multiple frames across the tracklet.

In this paper the evaluation results are reported as CMC curves, Rank-1 curve which show how Rank-1 is influenced by input/gallery sequence lengths, Rank-1/mAP tables without re-ranking to display out-of-the-box performance, with re-ranking to show optimal performance and lastly actual queries to showcase where model excels and struggles with predictions.

Šakinis, Jokūbas. Žmonių pakartotinio identifikavimo algoritmai vaizdo analizėje. Magistro baigiamasis projektas / Vadovas Prof. Dr. Andrius Kriščiūnas; Kauno technologijos universitetas, informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos mokslai, Informatika (B01)

Reikšminiai žodžiai: Dirbtinis intelektas, žmonių pakartotinis identifikavimas, žmonių re-id, sekų enkoderiai, TSCL.

Kaunas, 2026, 57p.

Santrauka

Šiame tyrime pristatomi du skirtingi giliuoju mokymusi pagrįsti asmenų pakartotinio atpažinimo (re-identifikavimo) modeliai kartu su jų realizacija, apmokyti naudojant tris etaloninius duomenų rinkinius: IUST_PersonReID, Market-1501 ir MARS. Market-1501 ir MARS duomenų rinkiniai buvo padalyti pagal oficialų duomenų padalijimą, kad rezultatus būtų galima palyginti su esamais tyrimais, taip pat pagal neoficialų pasirinktinį padalijimą tarp kamerų be nereikalingų („junk“) duomenų, siekiant parodyti geriausią įmanomą scenarijų, kaip modeliai veiktų naudodami švarius duomenis.

Pirmoji realizacija, apmokyta naudojant IUST_PersonReID duomenų rinkinį, naudoja GoogLeNet Inception v1 pagrindą (backbone) su svertine kontrastine praradimo funkcija, kurią sudaro pagal Gauso pasiskirstymą įvertintos teigiamos poros, riba pagrįstos neigiamos poros ir nuo tapatybės nepriklausomas dėmesys (attention).

Antroji realizacija buvo apmokyta atskirai naudojant Market-1501 ir MARS duomenų rinkinius, taip sukuriant du modelius. Realizacijoje naudojamas DINOv2 požymių išskyrimo pagrindas su jungtine praradimo funkcija, kurią sudaro AMSoftmax, apskritimo praradimas (circle loss), partijos sunkiausių pavyzdžių trijulių praradimas (batch-hard triplet loss) ir naujai siūlomas nuo laiko priklausančių juostų nuoseklumo praradimas (temporal stripe consistency loss). Pastarąjį sudaro vidinės kadro ir tarpkadrinės dalys, užtikrinančios, kad horizontalių juostų požymiai išliktų nuoseklūs per kelis pilnos sekos (tracklet) kadrus.

Šiame darbe vertinimo rezultatai pateikiami kaip CMC kreivės; Rank-1 kreivė, rodanti, kaip Rank-1 priklauso nuo įvesties / galerijos sekų ilgio; Rank-1/mAP lentelės be pakartotinio reitingavimo (re-ranking), rodančios veikimą „iš karto“; lentelės su pakartotiniu reitingavimu, rodančios optimalų veikimą; ir galiausiai konkrečios užklausos, atskleidžiančios, kur modelis prognozuoja gerai, o kur susiduria su sunkumais.

Table of contents

List of figures

List of tables

List of abbreviations and terms

CNN – Convolutional Neural Network

CUDA - Compute Unified Device Architecture

PCB - Part-based convolutional baseline

Re-Id – re-identification

PIF – Pose Invariant Feature

LDF – Local Descriptive Feature

GDPR – general data protection regulation

MARS – Motion Analysis and Re-identification Set

Tracklet – an image sequence of a person

LR – learning rate

TSCL – temporal stripe consistency loss

mAP — mean Average Precision

CMC — Cumulative Matching Characteristic

AMSoftmax — Additive Margin Softmax

AQE — Average Query Expansion

ViT — Visual Transformer

GRU — Gated Recurrent Unit

PID — Person Identity

OOM — Out Of Memory

OSM — Online Soft Mining

CAA — Class Aware Attention

SOTA — State Of The Art

GPU — Graphics Processing Unit

Rank-1 — Highest scoring accuracy

Introduction

Project novelty and relevance

Quickly finding criminals who escaped the crime scene or missing people is extremely important when time is of the essence. Manually combing through visual data when human resources are limited, taking hours and sometimes even days to find the person of interest could mean life or death to that person or someone else. By finding the most efficient method in person re-id we can quickly comb through the massive amounts of camera feed and locate the person of interest in the shortest time possible.

Analysis of existing methods and tools has shown the current problem in correctly re-identifying a person from an image or a video feed is the variance in pose and view and finding key points in an image such as body parts. Another main issue is the time that it takes to process said data when models become too complex. If possible, the research will suggest a hybrid solution for peak performance in accuracy and inference time.

Aim and objectives

The goal of this research is to find the combination of video-based person re-identification components which would achieve the best performance across identification accuracy and image processing time in person re-identification. To achieve this goal, the following objectives were formulated:

1. Review existing person re-identification methods and datasets – single image and tracklet datasets, single-image and video baselines and select components which seem to have a positive impact on person re-identification model training while considering limitations of the training and testing environment.
2. Implement the selected components into a person re-identification pipeline of data loading, data sampling, data augmentation and model training.
3. Evaluate the trained model using the chosen Rank-1 and mAP (mean average precision) metrics on custom and official Market-1501, MARS evaluation data splits, re-rank results, display how input sequence length affects Rank-1 and mAP metrics, check average query expansion results with varying k1 and lambda values and display query results with Rank-5.
4. Review the experiment results and suggest either the best performing or a novel hybrid solution which would outperform other solutions by being less complex and/or more accurate.

Document structure

This document focuses on the problem of person re-identification systems, models, methods and problems. In the first analysis chapter there will be a comprehensive analysis of existing technologies, architectures, methods and models while answering questions in each chapter being: What is it? How does it work? How is it relevant to person re-id task? In Methodology and Experimentation design backbone models will be presented together with the losses used in model training, evaluation results and conclusions that we draw from evaluation on official evaluation data splits of MARS and Market-1501 datasets. Finally, overall conclusions at the end of the paper summarizing impact of data, models, losses and possible future improvements.

1. Person re-identification algorithms in video analysis

In this chapter several person re-identification techniques will be analyzed to find the best fitting one for this research. In this chapter there will be an analysis on feature extraction methods, models commonly used in object re-identification tasks, main challenges in person re-identification tasks and ways to overcome said challenges. Lastly, the chapter will outline key points about the most optimal ways to solve the person re-identification problem presented as a summary at the very end.

1.1. Convolutional Neural Networks

What is convolution? Let us start by convolving a matrix with one single convolution kernel. Suppose the input image is 3 × 4 and the convolution kernel size is 2 × 2, as illustrated in Fig. 1 [1]

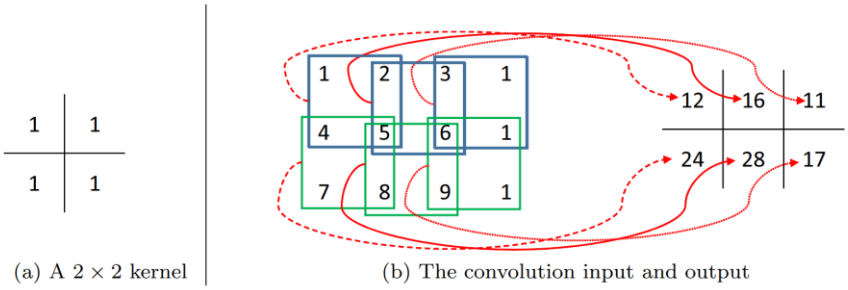

Fig. 1. Illustration of the convolution operation. [1]

By overlapping the kernel, we calculate the products and the total sum for the first window in a manner 1 × 1 + 1 × 4 + 1 × 2 + 1 × 5 = 12, 1 x 2 + 1 x 5 + 1 x 3 + 1 x 6 = 16, 1 x 3 + 1 x 6 + 1 x 1 + 1 x 1 = 11. In the same manner the sliding window is applied to the entire input to calculate the product.

Using convolution, we can apply different filters to exaggerate the selected features. The most common filters include edge detection Fig. 2, sharpening Fig. 3, smoothing Fig. 4 and embossing Fig. 5 meant to exaggerate the selected features in the given data. [2]

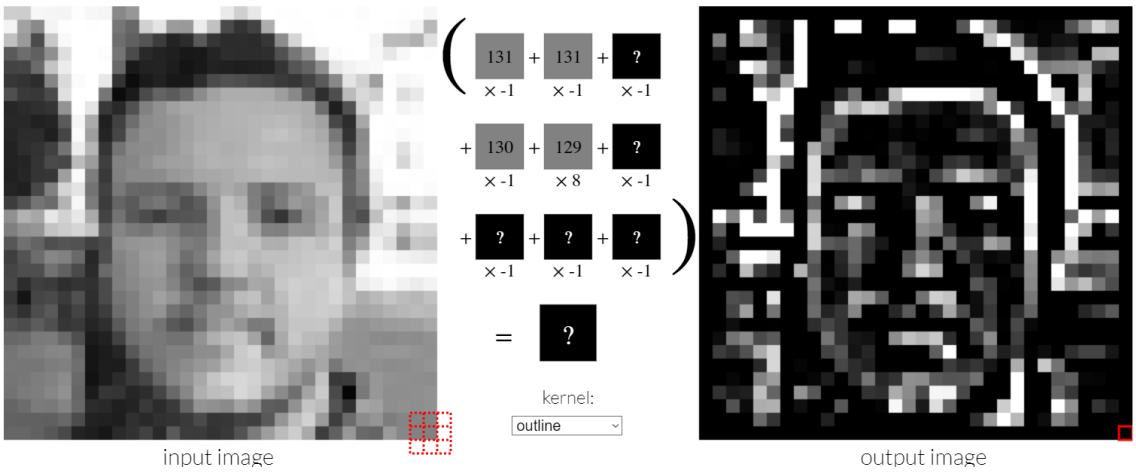

Fig. 2. Image with edge detection/outline filter applied [3]

As seen on Fig. 2, an edge detection filter is used to find the object edges in an image commonly used in image segmentation, object detection and computer vision.

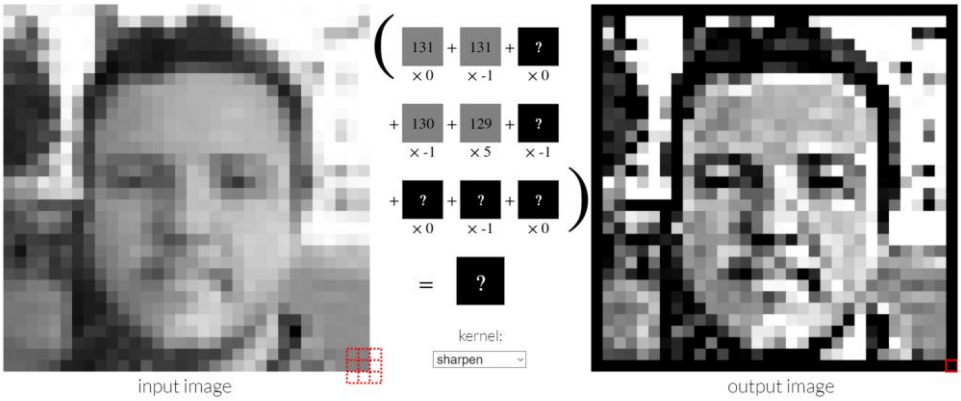

Fig. 3. Image with sharpening filter applied [3]

As seen on Fig. 3, a sharpening filter is used to enhance fine details in an image commonly used in improving clarity of an image.

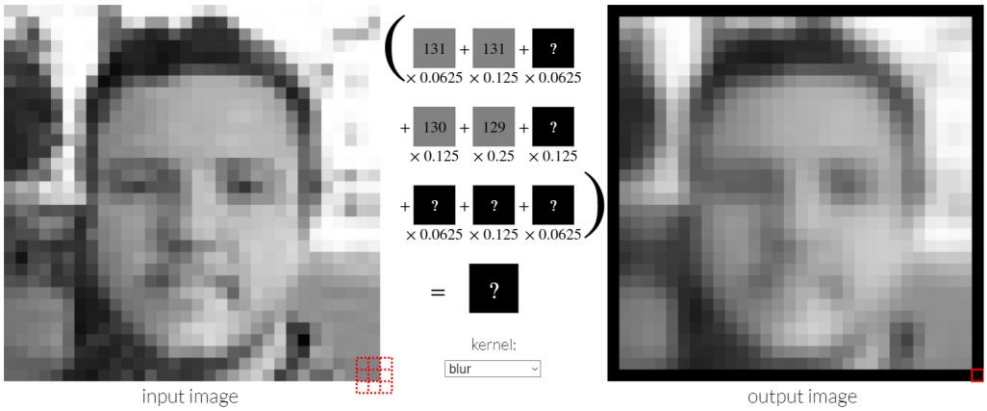

Fig. 4. Image with blurring/averaging filter applied [3]

As seen on Fig. 4, a blurring filter is used to reduce noise in an image commonly used for background smoothing, image preprocessing for object detection and reducing noise.

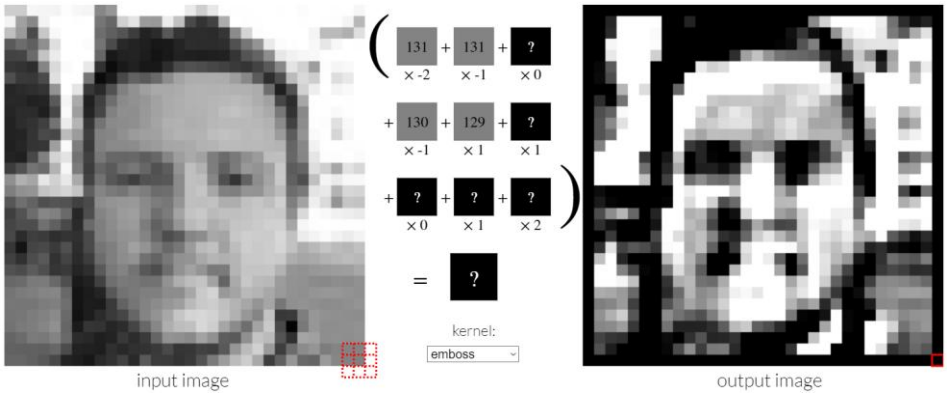

Fig. 5. Image with embossing filter applied [3]

As seen on Fig. 5, an embossing filter is used to create a 3D effect on an image commonly used for texture analysis and object feature enhancement.

Convolutional neural networks take advantage of these selected features, such as edges and patterns, by applying different kernels and learning from the calculated feature maps during training. Each training epoch seeks to optimize the model by minimizing the selected loss metric

In the task of person re-identification, we could use CNNs to extract discriminative features such as body shape, motion and clothing patterns. Using specialized kernels ensures that the model learns to identify and differentiate people across frames under different circumstances. The ability to be used with spatiotemporal data is key in the task of person re-identification, for this reason CNNs will be further explored.

1.2. Part-based CNN (PCB)

Part-based convolutional baseline is a model architecture designed for person re-identification. PCB uses the convolutional features and partitions them into separate parts for further analysis. It uses refined part pooling to fix partition errors, along with a weakly-supervised training technique called induced training. PCB utilizes multiple partitioning methods, such as pose estimation, human parsing and uniform partitioning for the exploration of features and the improvement of accuracy and efficiency in the task of person re-identification.

PCB works by modifying a backbone like ResNet50 to extract part-based features. It works by transforming input image to a tensor and dividing it into parts represented as column vectors, see Fig. 6 (f). These vectors are then processed by convolutional layers and classifiers for identity prediction. Refined part pooling refines the feature partitioning by reassigning pixels based on similarity, this way improving consistency across multiple different parts. During training, model joins the refined features into descriptors which help identify individuals.

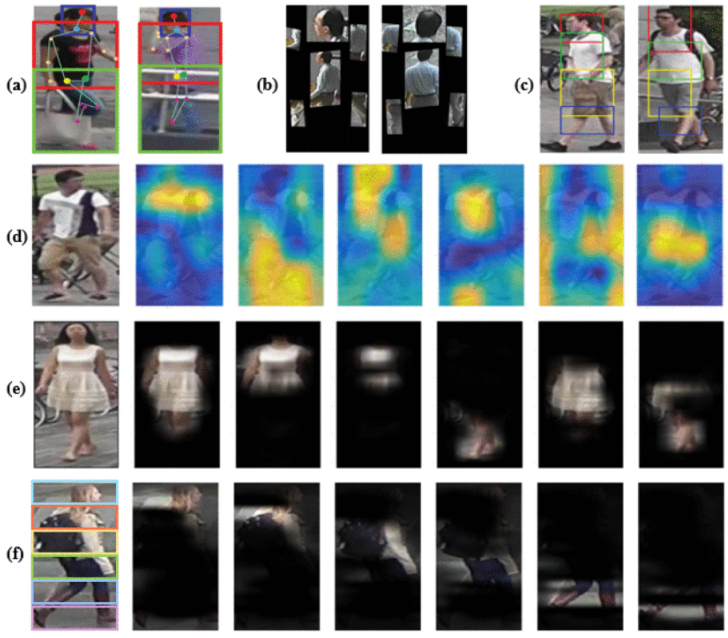

Fig. 6. Different partitioning strategies for people re-identification. (a) GLAD, (b) PDC, (c) DPL, (d) Hydra-plus, (e) PAR, (f) PCB + RPP [4]

The PCB is very useful in extracting spatial details necessary in identifying similarly looking people across different frames. The integration of part-based features and error correction ensures that the model works efficiently while limiting faults. Instead of using other models for body parts

segmentation, it divides an image into horizontal boxes which are then individually analyzed. This way the computational requirements are lower and as provided in the article reach state-of-the-art accuracy performance [4]. Weakly-supervised training allows the model to achieve relatively high accuracy without requiring detailed pixel-wise labels, making it more adaptable to real-world scenarios where there's a lack of labeled data.

In the task of person re-identification PCB raises the accuracy of the model by isolating features meant for each body part like the head, torso, arms, legs and shoes. Without the requirement of massive computing resources, it's much more efficient to adapt a simpler approach to the person re-id solution. Using the refined part pooling method minimizes inconsistency during human parsing. Considering the ever evolving need to accommodate higher loads of processing power it's crucial to minimize the need for it, for this specific reason PCB approach will be further explored.

1.3. Pose-Guided ReID

Pose-Guided ReID focuses on re-identifying people based on their pose using CNN's. It is a Part-Guided Representation consisting of Pose-invariant feature as well as Local Descriptive Feature to improve person re-identification. Part-guided Representation relies on pose extraction only introduced in the model training stage. The proposed method demonstrates competitive accuracy and efficiency over multiple datasets in the task of person re-id, however because it learns deep feature representations the model is more difficult to fine-tune. Because the expensive pose estimation and part segmentation is only applied during the training stage, but not required for the re-id, it allows for faster inference [5].

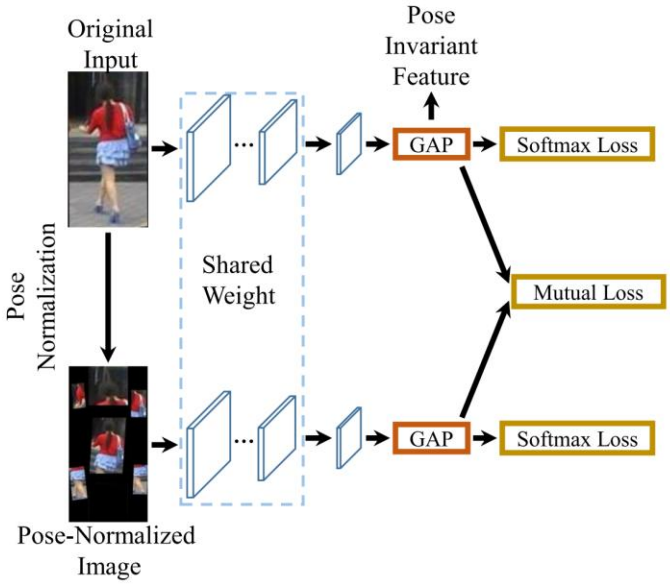

Fig. 7. PIF architecture [5]

Pose Invariant Feature is extracted by a deep model from a global image, general structure is displayed in Fig. 7. To gain the ability of pose invariance, PIF is enforced to approximate a pose invariant representation extracted from a pose normalized image. Pose normalization is achieved by first detecting deformable human parts, e.g., limbs and trunk, then normalizing those parts to fixed orientation and size with 2D affine transformations. [5]

PIF is directly learned from global images based on Eq. (1). Pose-invariant visual features are extracted to guide the learning of PIF. [5]

$$O(f) = \min\left(\frac{1}{N_P}\sum_{i,j,a} D\left(f_i^{(a)}, f_j^{(a)}\right) - \frac{1}{N_N}\sum_{i,j,a\neq b} D\left(f_i^{(a)}, f_j^{(b)}\right)\right) \quad (1)$$

Pose-invariant visual features are extracted to guide the learning of PIF. The updated training objective for PIF learning can be conceptually formulated based on Eq (2)[5]

$$O_{PIF} = O(pf) + min\sum_{i,a} D\left(pf_i^{(a)}, \overline{pf_i^{(a)}}\right) \quad (2)$$

Where $\overline{pf}$ denotes the pose-invariant feature extracted from the pose-normalized image. Compared with the original formulation in Eq. (1), Eq. (2) provides explicit supervision for pose invariance. The PIF can be extracted by performing pose normalization. Each rigid body part is detected and then a corresponding affine transformation is imposed to eliminate the pose variations caused by body movements. [5]

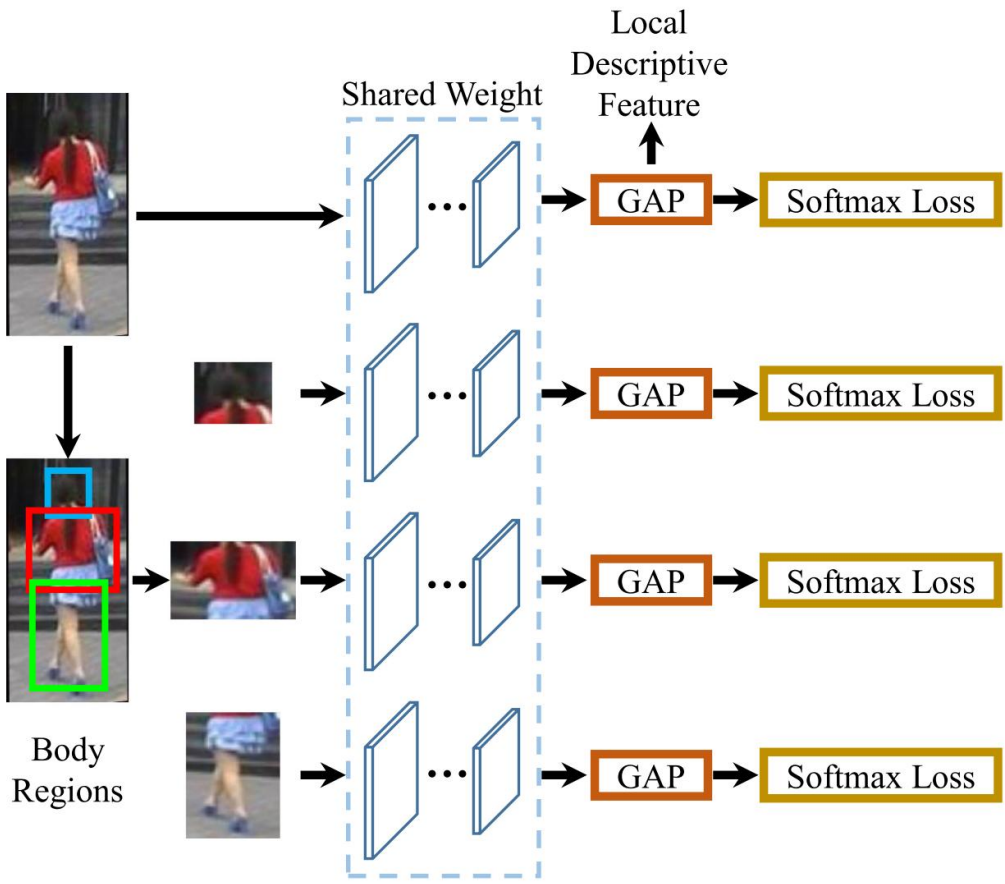

Fig. 8. LDF architecture [5]

LDF is Local Descriptive Feature used for extracting a person's body region consisting of a head, torso, arms and legs, an example is shown in Fig. 8. Body region detection is estimated to be using existing pose estimation algorithms. Located body regions are cropped by bounding boxes located by key points. The LDF is made discriminative to foreground and insensitive to backgrounds and learned by optimizing the following objective as such[5]

$$\min\left(\frac{1}{N'_P}\sum_{i,j,a} D\left(lf_i^{(a)}, \overline{lf_j^{(a)}}\right) - \frac{1}{N'_N}\sum_{i,j,a\neq b} D\left(lf_i^{(a)}, \overline{lf_j^{(b)}}\right)\right) \quad (3)$$

, where $\overline{lf}$ .denotes the feature extracted from a foreground person region. N'_P and N'_N are the numbers of positive and negative image pairs, respectively. The goal is to optimize LDF to make it discriminative to different body regions using similar idea of PIF. The extractor is trained on different body regions to learn regional features for person re-id. [5]

Using LDF and PIF we can move the difficult to process operations to the training stage, however doing so will make the model harder to tune since it will be a deep model. PIF is trained using softmax loss and mutual loss with pose-normalized images to ensure model robustness to pose variations. LDF is extracted using a CNN trained on regional body parts and co-trained with multiple classification tasks to enhance discriminativeness for body regions. The proposed method suggests pose normalization through rigid part detection, affine transformations and spatial transformer networks. Finally, PIF and LDF are both combined for final re-id using Euclidian distance to ensure robustness and discrimination.

In the task of person re-identification, we could use the Pose-Guided re-id approach to move heavy processing tasks to the model training stage, this way saving time while performing inference later on. By doing so the main issue becomes model tunability so it's important to weigh if it's worth sacrificing accuracy for the need of less computational power and latency.

1.4. Temporal Attention Mechanisms

Temporal attention mechanisms are used in sequential models like RNNs, transformers and temporal CNNs to focus on the most important parts of the data. Its usefulness comes from being able to efficiently deal with any sequential data such as videos, audio files, time-series data such as stocks, etc. [6][7][8]

Temporal attention mechanisms work by assigning weights of importance on sequential elements, allowing the model to focus on the most relevant information while processing the given data, see Fig. 9 [7]. An attention mask is generated in time and used to select frames of importance. This is key in cases where not all frames in a video are of equal importance. E.g. Occlusion, which partially hides the person of interest, pose variations which make it harder to detect a person or cluttered backgrounds which can be misinterpreted as people. By providing soft pixel-level attention the model can remember where the person was and quickly find it within a hard region-level after the occlusion has been cleared.

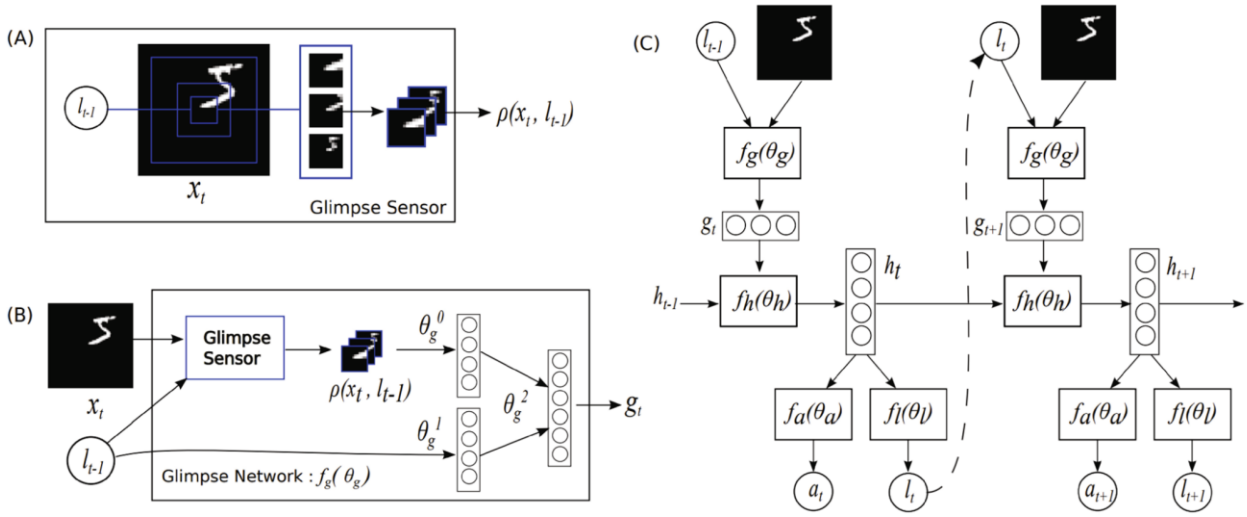

Fig. 9. Attention process in RAM [6]

Attention process in RAM is displayed in Fig. 9, where Recurrent attention model. (A) A glimpse sensor takes image and center coordinates as input and outputs multiple resolution patches. (B) A glimpse network includes a glimpse sensor, taking image and center coordinates as input and outputting a feature vector. (C) The entire network recurrently uses a glimpse network, outputting the predicted result as well as the next center coordinates.

Glimpse network meant to extract useful information is expressed as

$$g_t = f_{image}(X) \cdot f_{loc}(l_t) \tag{4}$$

where $f_{image}(X)$ and $f_{loc}(l_t)$ are non-linear functions which both output vectors having the same dimension, and multiplication denotes element-wise product, used for fusing information from two branches. [9]

The temporal attention mechanism is usually implemented using soft attention, where weights are calculated as probabilities over the sequence, and hard attention, which discreetly focuses on specific time steps. Furthermore, the mechanisms use self-attention in models like transformers, which let all the elements in a sequence be captured efficiently. Couple of examples of channel attention mechanisms are shown in Fig. 10

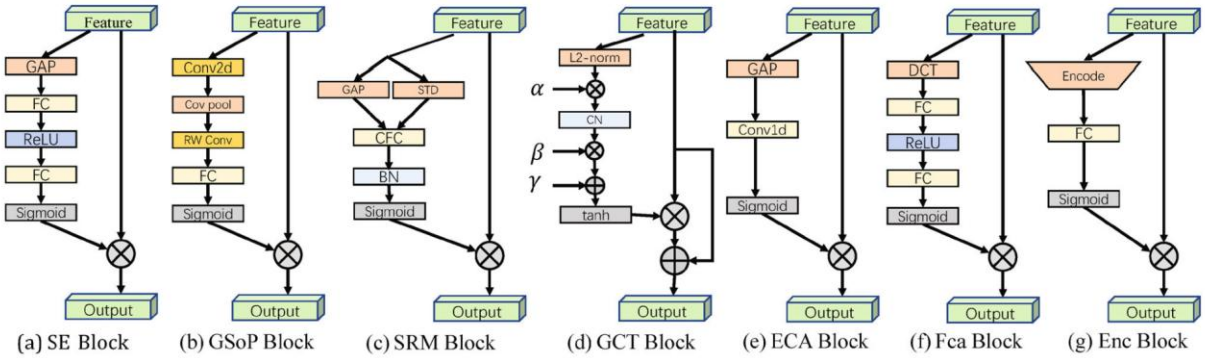

Fig. 10. Various channel attention mechanisms. [6]

In Fig. 10 we can see examples of various channel attention mechanisms such as: GAP = global average pooling, GMP = global max pooling, FC = fully-connected layer, Cov pool = Covariance pooling, RW Conv = row-wise convolution, CFC = channel-wise fully connected, CN = channel normalization, DCT = discrete cosine transform.

An existing solution suggests using an Attention Pyramid for person re-identification. Attention pyramid works by exploiting the attention regions in a multi-scale manner because human attention varies with different scales. The attention pyramid seeks to imitate the process of human visual perception which tends to notice the foreground person over the cluttered background and further focus on the specific color of the shirt with close observation. The attention pyramid works by a "split-attend-merge-stack" principle. The features are split into multiple local parts and then the corresponding attentions are learned from them. Following that the local attentions are merged and stacked with the residual connections in an attention pyramid, see Fig. 11. [8]

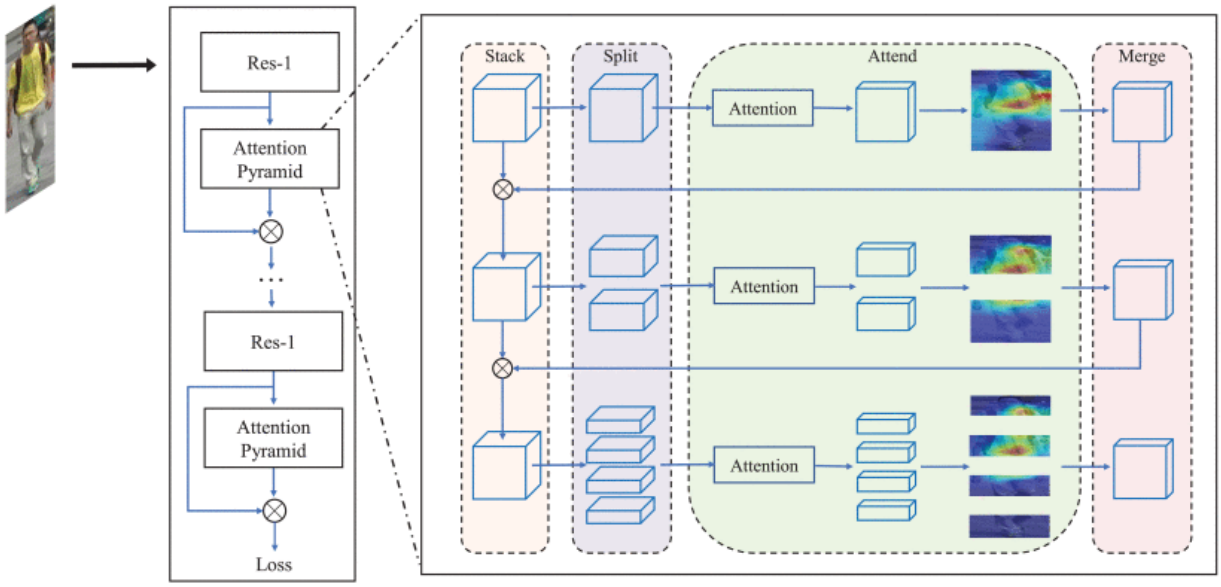

Fig. 11. "split-attend-merge-stack" principle attention pyramid model [8]

In the task of person re-id we could focus on key frames where the person is most visible, and the confidence is highest to then assign large weights for soft pixel-level attention. By doing so the model can quickly re-identify the same person by focusing on the area of interest which will most likely contain the person of interest. This way, occlusion will have less of an impact on overall model performance. The main problem in this solution is learning to assign the optimal weights and model tunability. This could be mitigated by dynamically assigning weights and weight decay.

1.5. Transformer-based Models

Out of the two analyzed types of models being non-deep learning and deep learning, transformer-based model is the latter. Transformers are best for working with sequence-to-sequence tasks and excel at capturing dependencies between data. They rely on a self-attention mechanism that processes input sequences in parallel, rather than sequentially compared to RNNs. This methodology allows them to handle larger datasets more efficiently.

Some of the better-known transformer-based models are BERT (Bidirectional Encoder Representations from Transformers) used to natural language processing tasks. GPT (Generative

pretrained Transformer) model which generates sequences by predicting the next token which is well known by many people nowadays thanks to ChatGPT. TransReID used in re-identification which uses learnable positional embeddings, Part-Aware transformers that combine part-based representations using transformer attention to enhance model's ability to re-id people based on key body regions. The general transformer architecture can be seen in Fig. 12.

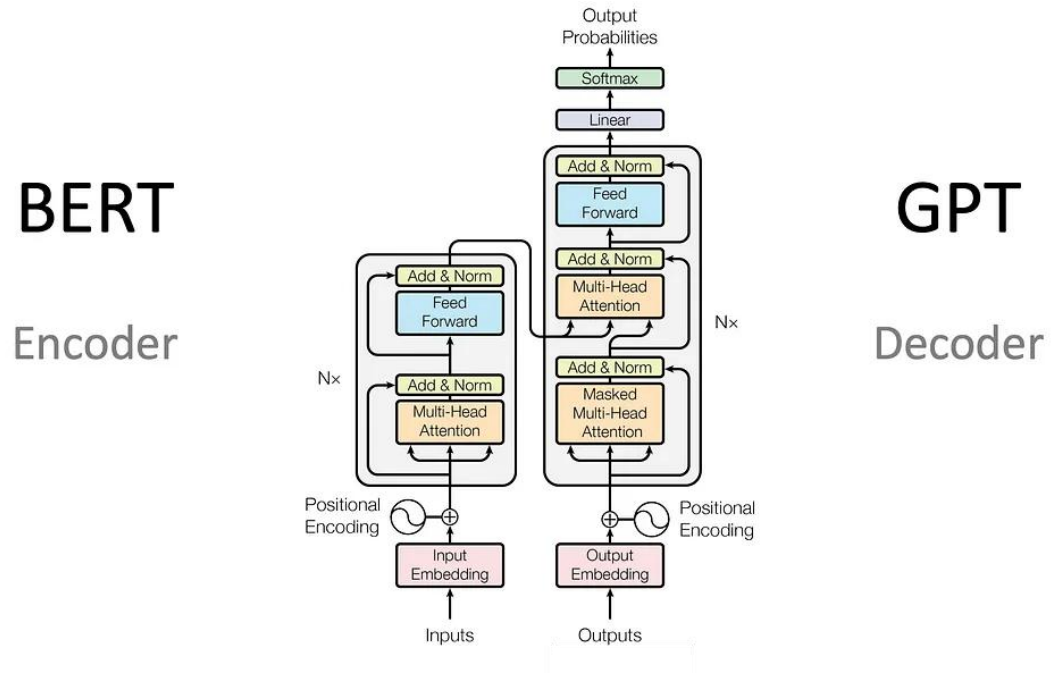

Fig. 12. Transformer architecture [10]

There exists pure transformer-based object re-id framework named TransReID. The implementation suggests using the jigsaw patch module (JPM) to rearrange the patch embeddings via shift and patch shuffle operations to generate robust features with improved discrimination ability and more diversified coverage. Side information embeddings (SIE) mitigate feature bias towards view variations by plugging in learnable embeddings meant to incorporate non-visual clues. This method achieves state-of-the-art performance on both person and vehicle reID benchmarks.

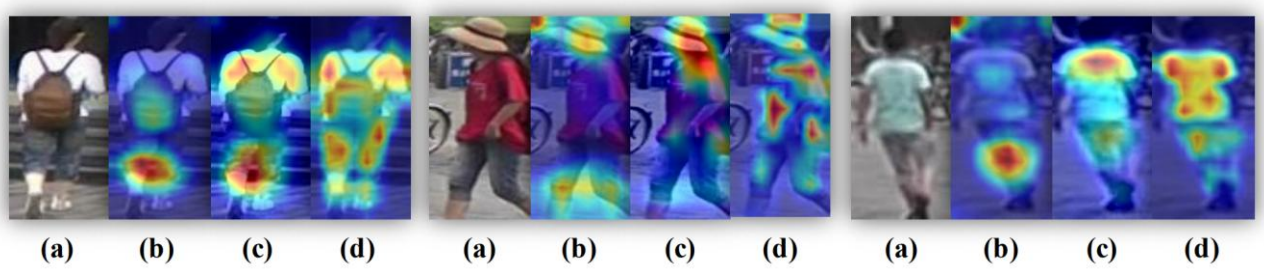

Fig. 13. Grad-CAM visualization of attention maps [11]

The Fig. 13 showcases Grad-CAM attention map visualizations: (a) Original images, (b) CNN-based methods, (c) CNN+attention methods, (d) Transformer-based methods which captures global context information and more discriminative parts. As seen in the Fig. 13 we can clearly see the superiority of a transformer-based method thanks to its ability to capture a greater amount of a person compared

to CNN-based methods and CNN+attention methods. Since view and pose variations are essentially the same as the model, the pure transformer-based solution offers a fix to both problems.

1.6. Market-1501 dataset

The Market-1501 dataset [12] is one of the largest public benchmark datasets created for person re-identification. Its 1501 identities, hence the name Market-1501, captured by six cameras along its 32668 pedestrian image bounding-boxes obtained using the Deformable Part Models pedestrian detector. On average each person has 3.6 images from each viewpoint. Dataset has been split into two parts: 750 identities used for training and the other 751 identities are used for testing. The dataset used deformable part-based model object detection framework after which the dataset was manually cleaned up to exclude false positives. See Fig. 14

- Identities: 1501
- Data count: 32668 images
- Data format: Image-based
- Camera count: 6 cameras
- Used for: Image to image re-id
- Data example: bounding_box_train/0002_c1s1_000451_03.jpg

In this example "bounding_box_train" denoted the training subset, "0002" is persons identity, "c1" means camera 1,"s1" means sequence 1 within the identity, "000451" says it's the 451st frame of the video and "03" means it's the 3rd bounding box.

Note that due to further research it has been noted that the mentioned dataset has been created without prior consent of the tracked individuals and therefore is questionable whether this dataset is usable in the scope of this research. The dataset does not have clearly documented consent of captured individuals and could face legal action for being problematic regarding EU's GDPR for processing personal data and may be removed from public access in the future.

Fig. 14. Market-1501 dataset example [12]

1.7. DukeMTMC-VideoReID dataset

The DukeMTMC dataset [13] is another large public benchmark dataset created for person video re-identification. The dataset contains 702 identities for training, 702 identities for testing, and 408 identities as distractors totaling 2196 videos for training and 2636 videos for testing. The sampling rate is every 12 frames from the video of each person's identity. Query images consist of different identities in different cameras. Gallery images which are used for testing consist of 702 identities and 408 distractors. For testing purposes, query image sequences are used to match the identities contained in the gallery.

Fig. 15. DukeMTMC dataset example [13]

Due to further research, it has been noted that the mentioned dataset has been removed from public access given the ethical and privacy legal concerns. The dataset data was collected illegally without prior consent from the captured individuals. For this specific reason the dataset got banned for not

complying with GDPR and will not be further used in this project but will still be noted as many of the most popular research papers have cited the use of this dataset for benchmarking.

- Identities: 1812
- Data count: 4832 video tracklets
- Data format: sequences
- Camera count: 6 cameras
- Used for: Sequence to sequence re-id
- Data example: train/0005/0007/0001_C1_F0001_X00111.jpg

In this example "train" denoted the training subset, "0005" is persons identity, "0007" video tracklet id, "C1" means camera 1, "F0001" says it's the 1st frame of the tracklet, "X" denotes the normal image, whereas "D" would mean distractor and "00111" says it is the 111th frame from the whole video on camera 1.

1.8. MARS (Motion Analysis and Re-identification Set) dataset

The large video-based individual re-id MARS dataset [14] is an extension of the Market-1501 [12] dataset. Image sequences are captured using 6 nearly synchronized cameras on the Tsinghua University campus. Contrary to Market-1501 [12] consisting of images, MARS dataset is made of image sequences/tracklets for each identity. This dataset is perfect for sequence to sequence-based re-identification research. The dataset used deformable part-based model object detection framework and Gaussian mixture model to form tracklets. Collected data wasn't manually filtered. [14]

- Identities: 1261
- Data count: 1191003 images
- Data format: sequences
- Camera count: 6 cameras
- Used for: Image to image / Sequence to sequence re-id
- Data example: bbox_train/0001/0001C1T0001F001.jpg

In this example "bbox_train" is the training subset, "0001" shows the person identity, "c1" means it's the 1st camera, "T0001" is the 1st frame of the tracklet, "F001" means it's the 1st frame of the video.

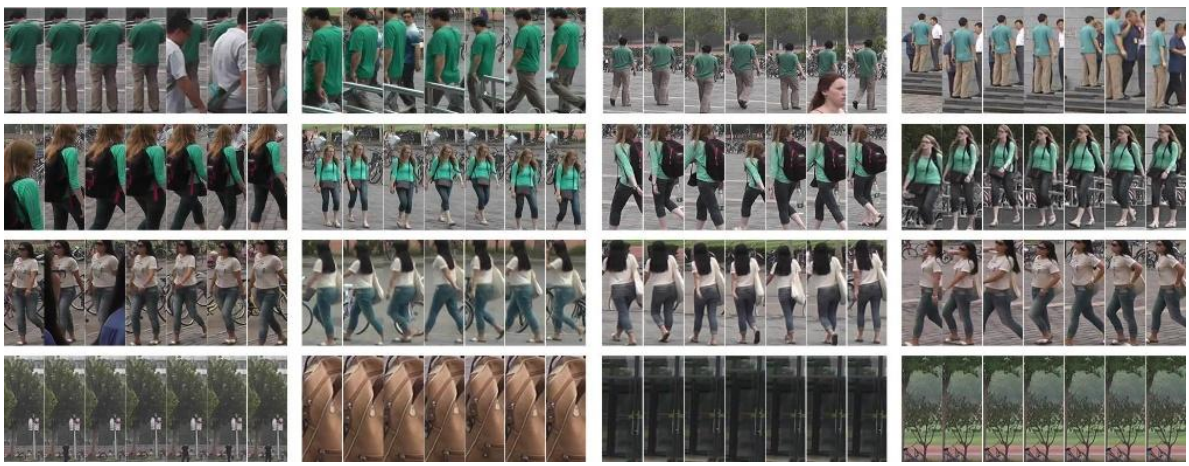

Fig. 16. MARS dataset example [14]

1.9. IUST_PersonReId dataset

A new person re-id developed in cultural environments of Islamic countries like Iran and Iraq. The dataset aims to introduce more difficult examples contrary to market-1501 dataset [12] providing identities wearing cultural clothes such as hijabs, and other coverings in many indoor and outdoor locations as well as various weather conditions. By providing variety the dataset helps reduce demographic bias for most popular person re-identification models and helps to generalize them and improve accuracy. It was noted that models, which were trained on market-1501 dataset, accuracy dropped significantly when used on IUST_PersonReID as an evaluation dataset.[15]

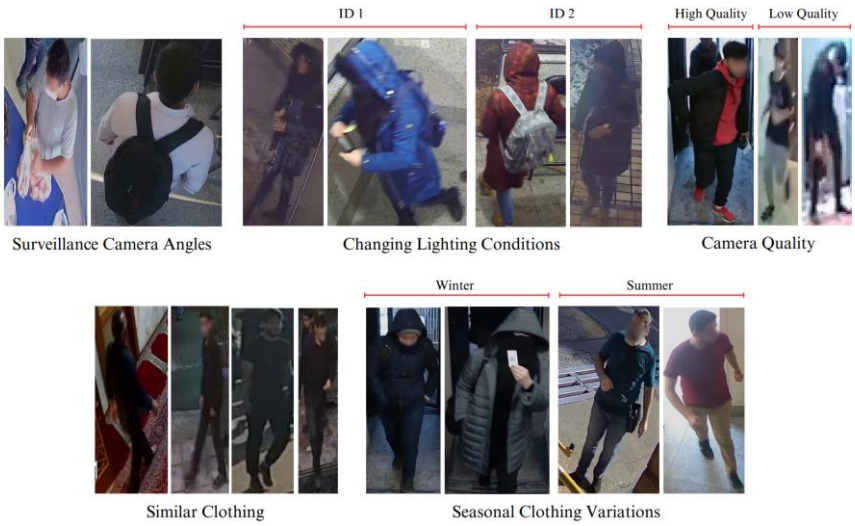

Fig. 17. IUST_PersonReId dataset example [15]

Dataset contains 1847 identities captured across 19 different cameras placed in outdoor as well as indoor locations of the IUST University campus, Market, Mosque and Streetview. In total the dataset captures 117455 images of varying lighting conditions, camera angles, seasonal attire, occlusions, sizes and quality.

- Identities: 1847
- Data count: 117455 images
- Data format: sequences
- Camera count: 19 cameras
- Used for: Image to image / Sequence to sequence re-id
- Data example: bounding_box_train/0001_c2s1_024962_01.jpg

For this example “bounding_box_train” is the training subset, “0001” is the identity of the person, “c2” means camera 2, “s1” says it is the first sequence of the identity of the person within the same camera context, “024962” means it is the 24962nd frame of the whole video from the camera, “01” stands for detection id if there are multiple identities detected within a single frame.

2. Methodology and experimental design for Video-Based Person Re-Identification

This section describes the building and training process of a person re-identification system, with a focus on its application to the IUST_PersonReId [15], MARS [14] as well as the Market-1501 [12] datasets. For the IUST_PersonReId dataset [15] a deep neural network backbone with adapted loss functions and feature extraction techniques such as soft positive mining, soft negative mining and class aware attention designed to improve performance in realistic multi-camera environments. For Market-1501 [12] and MARS [14] datasets a DINOv2 pre-trained vision transformer backbone [16] with loss functions such as AMSoftmax, batch-hard triplet, circle, novel temporal stripe consistency and feature extraction techniques such as multi-granularity horizontal stripe pooling, padding aware temporal transformer encoder, bi-directional GRU and learned attention pooling used to map tracklets to embeddings for cross-camera matching.

For this AI task a python environment was selected thanks to the plethora of implementations and libraries that exist in helping to tackle tasks of person re-identification. The chosen library for model implementations depends on the researchers' solutions which are being analyzed. So far both tensorflow and torch libraries were noted to being used in implementing these solutions, however the two examples which will be talked about so far are both implemented using torch and other miscellaneous libraries like sklearn for data augmentation, matplotlib for visualizations, shutil, os, re libraries for file management, PIL for image management, tqdm for progress bars, torchvision for base models and a few other minor libraries for minor tasks.

2.1. IUST_PersonReID pipeline

The backbone for training on IUST_PersonReId [15] is based on GoogLeNet (Inception v1) with a couple changes to adapt it to the re-identification task. Firstly, batch normalization was added after each inception module, which helped with training stability and generalization. The final feature vector is made up of 1024 dimensions and is projected into a configurable embedding, depending on the experiment. L2 normalization is applied to the embeddings to ensure the same scale, which makes comparisons of distance more stable and reliable.

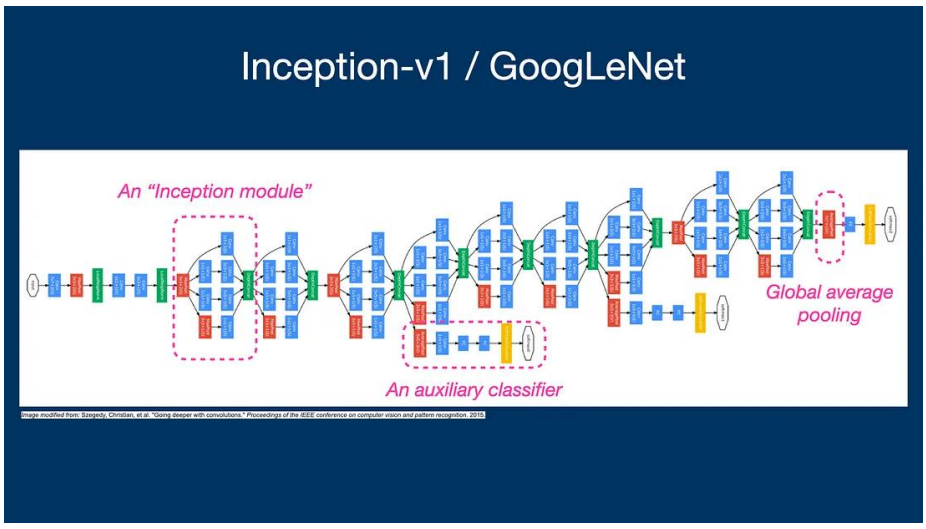

Fig. 18. GoogLeNet (Inception v1) model architecture [17]

2.1.1. IUST_PersonReID Data Preparation and Sequence Splitting

The dataset follows a specific format common for many person re-id datasets. The filename format contains information like the person id, the camera number, and frame id. An example of a filename looks like this: 0001_c2s1_024962_01.jpg. Images are resized to 224x224, normalized using ImageNet stats, and kept as 3 channel input (RGB). Just enough augmentation was applied to prevent model overfitting. The sequence splitting is based on time gaps between frames – if the gap is larger than 50 frames, it is considered as new sequence to prevent the creation of a single sequence when the same identity was captured on the same camera, but over long periods of time after walking out of view and coming back in. Two-frame minimum for a sequence to count.

During training a CUDA-enabled python environment was used with GPU acceleration to reduce the training time. During training many CUDA out of memory (OOM) issues popped up and necessary actions to reduce the impact of training on the GPU were taken. For data loading a multi-worker loader was implemented to reduce data loading times, batch sizes were adjusted to avoid OOM issues, used pin memory for faster data transfer to GPU and added memory monitoring between training and evaluation processes.

2.1.2. Weighted Contrastive Loss and Attention Design

The main loss function that was used was Weighted Contrastive Loss (WCL), which includes three parts: positive pair loss, negative pair loss, and a metric learning component. Positive pairs are scored using a Gaussian formula $\exp\left(-\frac{d^2}{\sigma^2}\right)$, and attention weights are identity-agnostic, so they don't rely on labels directly. For negatives, a margin-based function a $max(0, \alpha - d)$ was used with a dynamic attention mechanism that gives more focus to hard negatives. The third part tries to pull apart embeddings of different identities while keeping same-identity pairs close, especially across cameras.

A few attention mechanisms were added to the model itself, all directly inside the training script. There's a general identity-agnostic attention block, a more dynamic one that changes focus based on training stage, and a hierarchical module that combines features from different depths. Fine tuning these parameters will still be in question, they tend to interact in unpredictable ways with sampling and batch composition.

For batches we used a class-balanced sampling approach described in the paper of OSM with CAA features [18]: three identities per batch, each with 18 samples, for a total of 54.

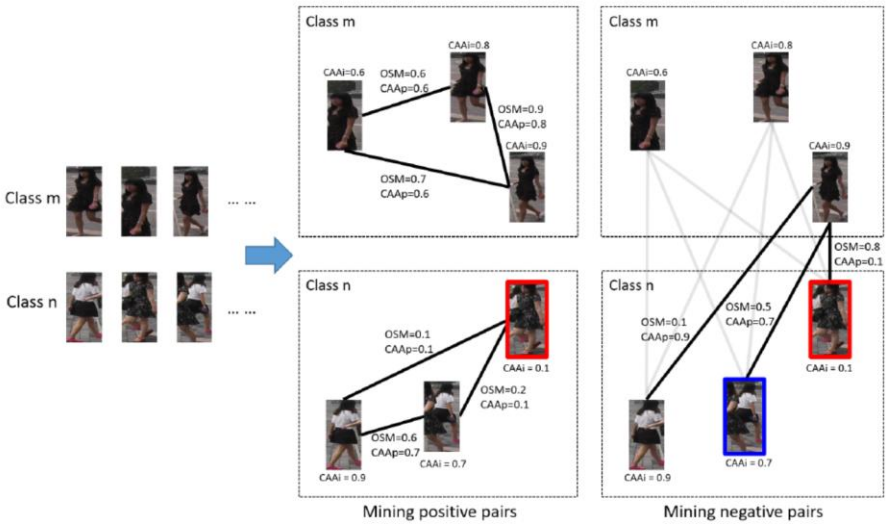

Fig. 19. The illustration of Online Soft Mining (OSM) and Class-Aware Attention (CAA) for pair mining. [18]

That worked well in terms of GPU memory and loss convergence. The SGD optimizer was used with a 1e-3 learning rate, momentum at 0.9, and some small weight decay of (5e-4). ReduceLROnPlateau was used to adjust the learning rate – it halves the LR when the validation loss doesn't improve for 5 epochs. Lowest LR it goes to is 1e-6.

2.1.3. IUST_PersonReID Evaluation Setup and Results

For evaluation, the usual setup was followed. Metrics include mAP (mean Average Precision) which calculates overall accuracy of the queries images and their matches respectively and CMC (Cumulative Matching Curve) like CMC@1, CMC@5, CMC@10 which show the probabilities of finding correct matches within Rank-1, Rank-5 and Rank-10 respectively. The query and gallery are separated by camera view – same ID images from the same camera aren't allowed in the query-gallery match. Visualization tools were used to display the query and its top retrievals. Not automated, had to write a simple script to display the top-N results with their scores. Sometimes visually it's obvious where the system struggles (like with occlusions or bad lighting). The final results of each sequence length experimentation are then aggregated into a .html file to scroll and view the results for each sequence length going from n to n_{max}.

Inference was done in batches on GPU, with memory tracking enabled to avoid crashes. Everything was L2-normalized, and the loader used pin_memory and multiple workers. Tried to avoid holding intermediate tensors longer than needed, as it slowed down larger sequences.

Augmentations were basic – resizing, normalization. Flips, rotations, cropping or any other more various image data augmentations weren't applied since the sequences are short, and temporal data is extremely sensitive to such transformations and would defeat the whole purpose of having sequential data.

The codebase is neatly structured and modular where each process group e.g.

Mining.py contains all the functions necessary for feature mining such as dynamic attention, hierarchical features, metric learning loss, online soft positives mining, online soft negatives mining, weighed contractive loss.

evaluate.py contains functions necessary for model evaluation which are feature extraction, similarity computing, cmc mAP computing, and retrieval results plotting.

analyse_frame_gaps.py used for analyzing frame gap size, sequence gap means, standard deviations and distribution of sequence lengths.

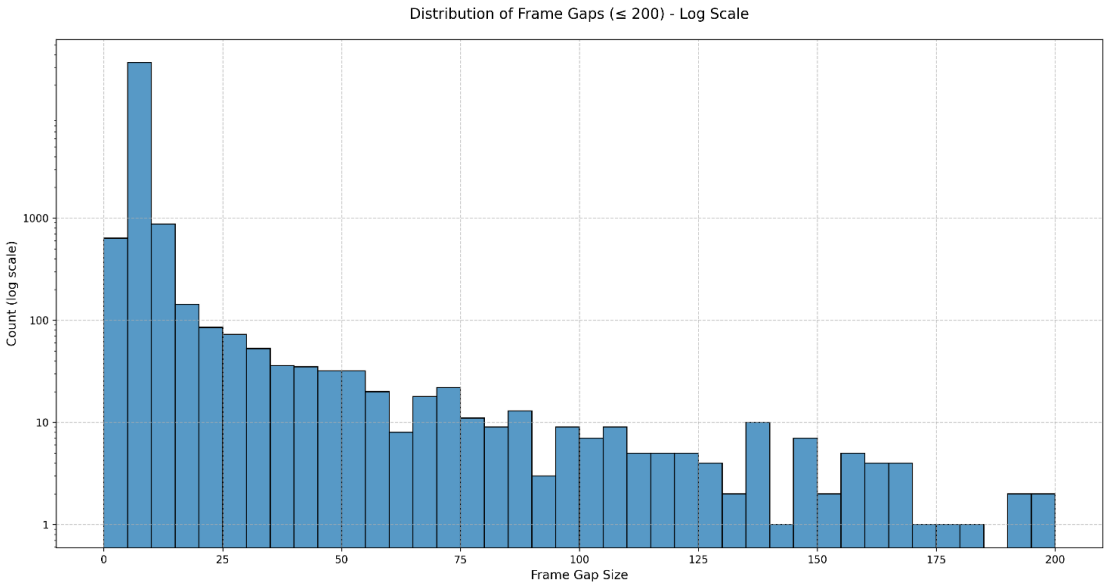

Fig. 20. frame gaps distribution taking frames only with gaps ≤ 200

Overall Statistics for frame gap analysis on training set with max gap limit set to 200 frames:

- Total number of identities: 601
- Total number of sequences: 1205
- Mean gap size across all sequences: 6.92
- Median gap size across all sequences: 6.00
- Standard deviation of gap sizes: 7.85
- Maximum gap size: 199
- Minimum gap size: 4

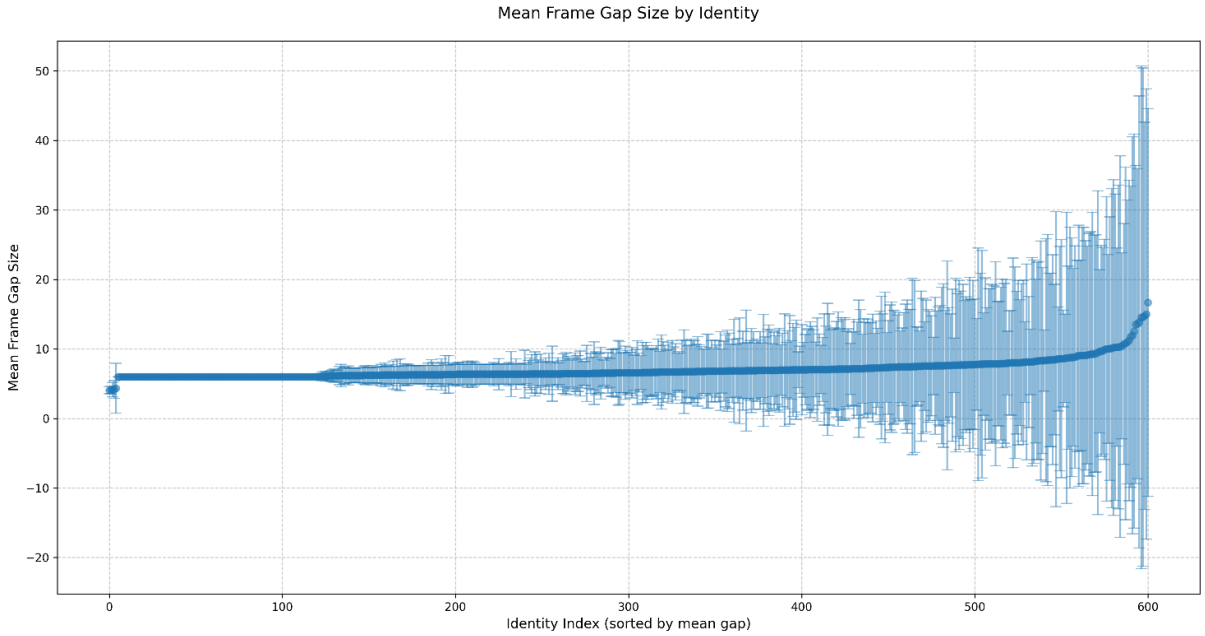

Fig. 21. Mean frame gap size diagram by identity

Some identities have massive gaps between sequence frames and stop at around 50 frames maximum, for this specific reason it was chosen to split sequences into two separate ones if they exceed the noted limit.

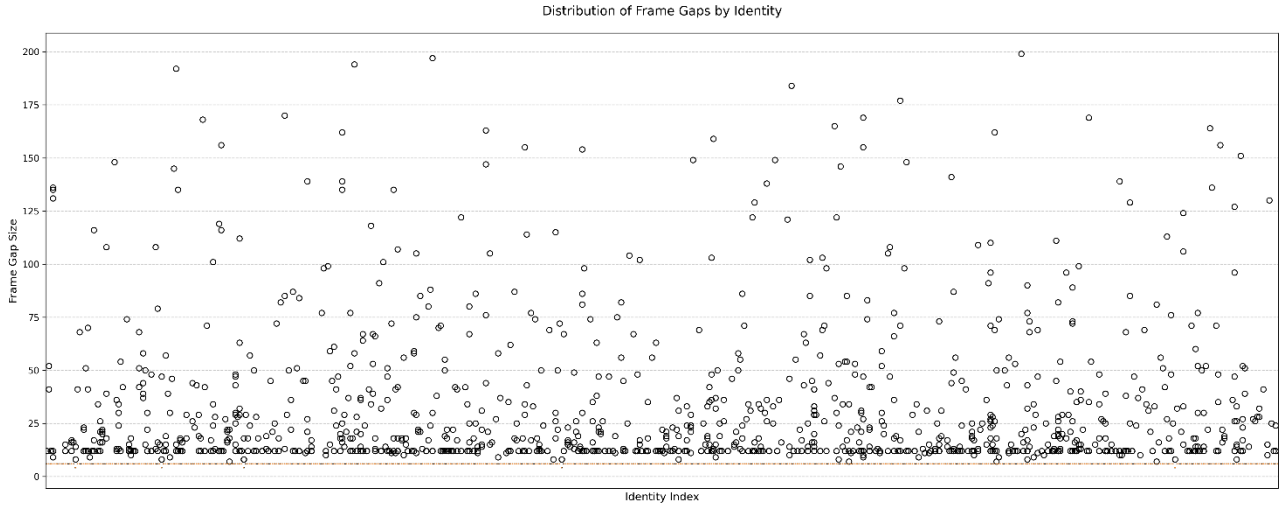

Fig. 22. Gap boxplot for diagram for each identity

As seen in the gap boxplot there are quite a few identities containing large frame gaps which show that sequences should be divided into two or more unique sequences.

Preprocess_data.py takes care of splitting data into appropriate sequences for identities, resizing, normalizing, moving to tensors, extracting features and sampling for training, testing and query subsets to be used in further calculations.

batch_sampler.py is used to sample batches according to the parameters of the paper [18].

video_reid.py evaluates the custom reformatted “bounding_box_test” subset and displays 10 visualizations for top 5 matches for each sequence length which was chosen, in this case 2-18 sequence lengths and aggregated into an .html type file for easier viewing.

From the figures Fig. 23, Fig. 24 of queried person results it is clearly seen that one of many issues in person re-id is occlusion in which a part of a doorframe looking object is obstructing a part of the view and the model gets caught on that quirk and tries finding similar images to the queried image. This problem should solve itself when querying using longer sequences as the occlusion no longer obstructs the view of the camera as the person moves around, which is seen in the gallery images shown below the queried images.

Fig. 23. query_sequence=2 person results showing CMC@5 matched sequences by similarity and displaying whether the result is a match or no match (1)

Fig. 24. query_sequence=2 person results showing CMC@5 matched sequences by similarity and displaying whether the result is a match or no match (2)

Another main issue to note is the change in lighting which sometimes tricks the model the same way people get tricked without having enough context to go around. A good example was a debate that had sparked on the internet over the Blue-Black or White-Gold dress[19].

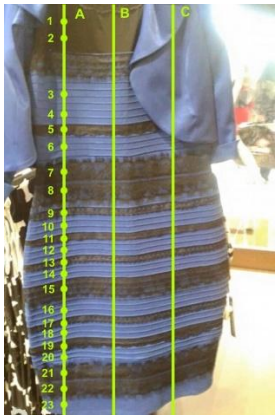

Fig. 25. Blue-Black or White-Gold dress [19]

Due to color theory same colors in different lighting look completely different and without enough temporal data context of sequential frames of a person moving over different lighting the model gets tricked the same way human eye does. This can be clearly seen in Fig. 23, Fig. 24 where the persons whose ID:0053 backpack first appeared to be light blue, then with less lighting appeared to be black or dark blue and then once again at the end of the sequence appeared light blue.

When querying using 13 image overall accuracy reaches its peak, however exceeding the query length of 13 images reduces the overall accuracy of the sequence-to-sequence matching results. This could be due to the image sequence containing information too diverse for the model to efficiently convert it into a single vector and generalize too much focusing on color rather than subtle features of the person as seen in Fig. 26, Fig. 27. As seen in the queried sequences the model starts generalizing on color trying to find all identities from the same camera which were wearing a similar colored jacket. To mitigate this generalization, it could be worth considering a part-based vector approach where each body part gets a vector representation over the sequence of frames. This way more than main body information would be saved, and temporally important data will be maintained while still generalizing the sequence of the given identity.

Sequence Length: 2 Fig. 23, Fig. 24
- Number of valid query sequences: 1705
- Correct Matches: 420/4190 (10.02%)

Possible matches statistics:
- Average possible matches per query: 2.46
- Min possible matches: 0
- Max possible matches: 10

Fig. 26. query_sequence=12 person results showing CMC@5 matched sequences by similarity and displaying whether the result is a match o r no match (1)

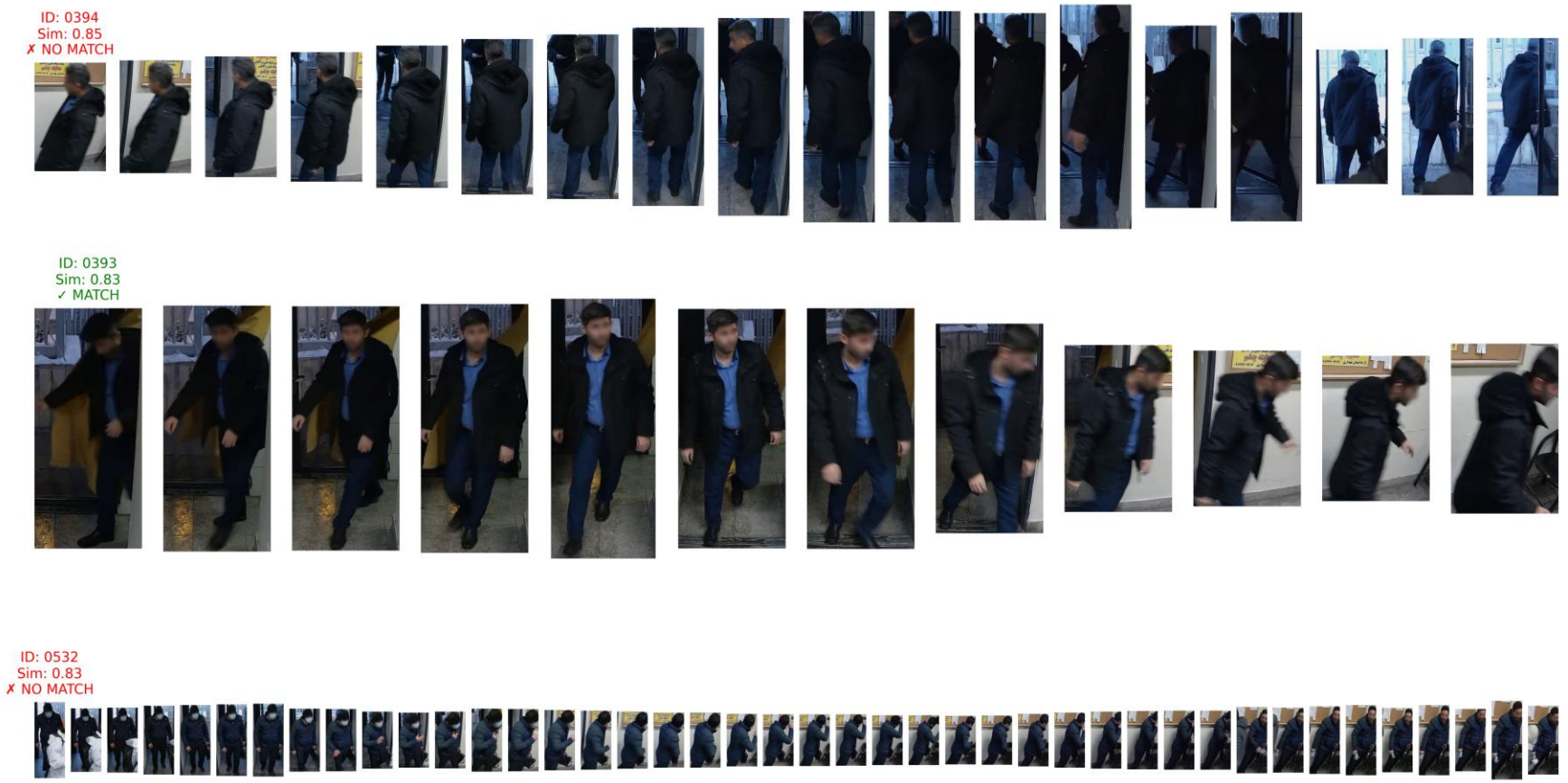

Fig. 27. query_sequence=12 person results showing CMC@5 matched sequences by similarity and displaying whether the result is a match o r no match (2)

Sequence Length: 12 Fig. 26, Fig. 27

- Number of valid query sequences: 1267
- Correct Matches: 345/2866 (12.04%)

Possible matches statistics:

- Average possible matches per query: 2.26
- Min possible matches: 0
- Max possible matches: 10

Table 1. Evaluation results

Sequence Length	**Number of valid query sequences**	**Correct Matches**	**Average possible matches per query**	**Min possible matches**	**Max possible matches**
2	1705	420/4190 (10.02%)	2.46	0	10
3	1673	433/4070 (10.64%)	2.43	0	10
4	1624	427/3907 (10.93%)	2.41	0	10
5	1586	431/3771 (11.43%)	2.38	0	10
6	1554	423/3670 (11.53%)	2.36	0	10
7	1514	411/3542 (11.60%)	2.34	0	10
8	1482	397/3434 (11.56%)	2.32	0	10
9	1434	377/3294 (11.45%)	2.30	0	10
10	1387	376/3167 (11.87%)	2.28	0	10
11	1321	355/3007 (11.81%)	2.28	0	10
12	1267	345/2866 (12.04%)	2.26	0	10
13	1204	328/2713 (12.09%)	2.25	0	10
14	1115	291/2478 (11.74%)	2.22	0	10
15	1051	276/2309 (11.95%)	2.20	0	10
16	989	256/2157 (11.87%)	2.18	0	10
17	932	230/2037 (11.29%)	2.19	0	10
18	869	202/1876 (10.77%)	2.16	0	10

After evaluating the trained model, we can see the results displayed in Table 1 when using sequence of 2 images the percentage of correctly matched images were at their lowest being 10.02% and accuracy reaches peak when sequence length reaches 13 images at 12.09% and then drops to 10.77% at sequence length of 18 images.

As for future improvements, I'd probably explore transformer-based backbones (ViT or some hybrid) and a curriculum learning schedule. Another area worth exploring is better loss weighting – right now the three loss components are just equally scaled, but dynamic reweighting could help. Evaluation-wise, some real-world scenarios like long-term id tracking or cross-dataset tests (especially from public surveillance sets) might give more insight into generalization

2.2. DINOv2 video ReID pipeline

On MARS [14] and Market-1501 [12] datasets a pre-trained visual transformer DINOv2 [16] backbone using loss functions such as AMSoftmax, batch-hard triplet, circle, novel temporal stripe consistency and feature extraction techniques such as multi-granularity horizontal stripe pooling, padding aware temporal transformer encoder, bi-directional GRU and learned attention pooling were used to train a model for a person re-identification pipeline. Model using the MARS dataset was trained on an A100 80GB VRAM GPU due to the dataset being so large, meanwhile model trained using the Marker-1501 did not require as many computing resources so a single RTX 3080Ti 12GB VRAM GPU was sufficient for the training.

2.2.1. Market-1501 and MARS Data Preparation and Evaluation Splits

After training each model on different datasets they were evaluated using an unofficial custom filtered data evaluation split and an official data evaluation split provided together with dataset as a separate file with IDs or a folder with query and test images. Custom split showcases best model performance in real-world scenarios where person on camera is detected, and a sequence is formed without any distractors or background images. Official split allows model results to be comparable with current person re-identification publications.

Table 2. Dataset sequence analysis

Dataset	**Market-1501**	**MARS**
Mean	1.8	61.5
Median	1	36
P95	4	186
P99	6	277
Max	17	900

After a quick analysis seen from Table 2 it is clear that the Market-1501 dataset lacks temporal information diversity, it is evident from the sequence length mean being 1.8 and median being 1. Market-1501 dataset is chosen to display the difference in results between a dataset with large temporal diversity like MARS which has the sequence length mean of 61.5 and median of 36. Thanks to this short analysis for training each model we choose MAX_SEQ_LEN parameters as 8 and 64 respectively to fully utilize the dataset and limit padding sequences to the largest sequence while batching to conserve computing resources and speed up model training.

Table 3. Model evaluation splits

Dataset	Split	Query size	Gallery size	Junk in gallery
MARS	Official (MATLAB protocol)	1980	12180	PID=-1, same cam
	Custom cross-camera	626	5853	no
Market-1501	Official (query/test folders)	3368	11783	PID=-1, backgrounds, same cam
	Custom cross-camera	750	6060	no

For both datasets displayed in Table 3 we use official splits found within either original dataset folders, or public official evaluation splits and custom cross-camera splits. Official split for MARS dataset contains distractor images marked with PID=-1 and same camera images. Official Market-1501 dataset contains distractor images marked with PID=-1, background images and same camera images as well. For custom splits both were filtered to not contain any junk images to simulate an ideal dataset and without same camera sequences to avoid artificially inflating Rank-1 and mAP scores.

In preparation for data augmentation, we utilize multiple different data augmentation methods to increase data variety and introduce more difficult examples during model training the augmentations presented in Table 4

Table 4. Augmentation parameters for train and test splits

Applied to	Data augmentation type	Value
Train/Test	Resize	(256, 128)
Train/Test	Normalization	mean=[0.485, 0.456, 0.406] std=[0.229, 0.224, 0.225]
Train	Random Horizontal Flip	-
Train	Pad	10
Train	Random Crop	(256, 128)
Train	Color Jitter	brightness=0.3, contrast=0.3, saturation=0.3, hue=0.1
Train	Random Erasing	p=0.5, scale=(0.02, 0.33), ratio=(0.3, 3.3)

Standard data pre-processing steps like resizing images to 256x128 and normalization were performed to both training and test data splits. For training alone, we utilized random horizontal flipping, image padding, random crop from images, color jittering and random erasing.

2.2.2. DINOv2 Tracklet Embedding Architecture and Combined Loss

This section showcases basic pipeline working principle and how data is processed and moved to the next stage. A rough outline is presented in the (5) formula and presented in detail.

Let $x \in \mathrm{R}^{B\times T\times 3\times H\times W}$, where B is a mini batch of zero-padded tracklets of max length T with 3 color channels, height of image H and width of image W.

$$x \to (ViT) \to tokens \to (stripe\ pool) \to f \in R^{B\times T\times 2048} \to (T-TF) \to f' \to (BiGRU) \to h \in R^{B\times T\times 1024} \to (attention\ pool) \to z \in R^{B\times 512} \quad (5)$$

Model returns one L2 normalized embedding $z \in \mathrm{R}^{B\times d}(d = 512)$ for each sequence and ID logits based on MAX_CLASSES provided at model creation time along with stripe features for loss calculations.

After frames are resized to 256x128 using data augmentation from Table 4, they are passed through a DINOv2 visual transformer which generates an 18x9 patch grid and a global summary token.

These tokens are then passed through multi-granular stripe pooler with inputs reshaped to $(H_p = 18, W_p = 9{,}768)$ and cropped to 16 rows which guarantee that chunking them produces 2/4/8 equally sized horizontal stripes. Each stripe has mean-pooling applied to it over multiple frames and projected linearly.

- $g \in \mathrm{R}^{512}$ – projection of the CLS token (global descriptor).
- $L_2 \in \mathrm{R}^{2\times 256}$ – two stripes (upper/lower body), 256-d each.
- $L_4 \in \mathrm{R}^{4\times 128}$ – four stripes (head, torso, hips, legs), 128-d each.
- $L_8 \in \mathrm{R}^{8\times 64}$ – eight stripes, 64-d each.

Per frame feature is combined into $f = concat(g, L_2, L_4, L_8) \in \mathrm{R}^{2048}$, but four stripe features are kept separately $L_4 \in \mathrm{R}^{T\times 4\times 128}$ to be forwarder to the temporal stripe consistency loss calculation which is a novel solution compared to existing stripe feature calculations in published papers which only calculate stripes on a single image.

Resulting combined per frame features are then passed forward to a two-layer transformer encoder after which the resulting masked features are directly passed to a bi-directional gated recurrent unit with hidden size 512 and 0.3 dropout for each layer which produces an output $h \in R^{B\times T\times 1024}$.

A dot-product attention pooler is used to calculate scores for frames to determine which frames from the sequence have the most importance and then gives them a higher weight when creating the final sequence representation. Padded frames are ignored by assigning scores of negative infinity to receive zero weight after SoftMax. Final embeddings are calculated as a weighted average of all features from frames. Lastly, we apply Linear(1024->512) with BatchNorm1D along with L2 normalization to get final tracklet embedding $z \in R^{B\times 512}$

Finally tracklet embedding is then passed to a linear classifier with C classes denoted by MAX_CLASSES passed during model creation.

$$L_{combined} = L_{AMS} + L_{triplet} + L_{circle} + L_{TSCL} \tag{6}$$

Besides a classifier we utilize a combined loss consisting of AMSoftmax CosFace ID loss, batch-hard triplet loss, circle loss and novel suggested temporal stripe consistency loss Equation (6).

$$L_{AMS} = -\frac{1}{B}\sum_{b=1}^{B} \log \frac{\exp\left(s(\hat{p}_{b,y_b} - m)\right)}{\exp\left(s(\hat{p}_{b,y_b} - m)\right) + \sum_{c\neq y_b} exp\left(s\,\hat{p}_{b,c}\right)} \tag{7}$$

, where B - batch size, $\hat{p}_{b,c}$ - cosine similarity between sample b and class c, y_b- correct class label, m - additive margin, s - scaling factor, $\hat{p}_{b,y_b}$ – correct class score [20].

$$L_{triplet} = \frac{1}{B}\sum_{b=1}^{B}\left[max_{\ y_p=y_b,\,p\neq b}\, d(z_b, z_p) - min_{\ y_n\neq y_b}\, d(z_b, z_n) + \alpha\right]_+ \tag{8}$$

, where $max_{\ y_p=y_b,\, p\neq b}\, d(z_b,z_p)$ same identity, but furthest away, $min_{\ y_n\neq y_b}\, d(z_b,z_n)$ same identity, but closest, α – safety margin [21].

$$\alpha_p = [1 + m - s_p]_+ \ \alpha_n = [s_n + m]_+ \ \ \Delta_p = 1 - m, \Delta_n = m \tag{9}$$

, where s_p - positive pair similarity, m – margin, s_n – negative pair similarity, Δ_p – similarity threshold, Δ_n – negative similarity threshold, α_p – positive pair weighting, α_n -negative pair weighting [22].

$$L_{circle} = \frac{1}{|\ B_{valid}\ |}\sum_{b\in B_{valid}} log\left(\begin{array}{c} 1 + \exp\left(\sum_{n\in N_b} \gamma\, \alpha_n (s_{bn} - \Delta_n)\right) \\ exp\left(-\sum_{p\in P_b} \gamma\, \alpha_p (s_{bp} - \Delta_p)\right) \end{array}\right) \tag{10}$$

, where s_{bp} – positive pairs, s_{bn} – negative pairs, b – anchor, p – positive sample, n – negative sample, α_p & α_n – positive and negative weights, γ – scale factor, Δ_p & Δ_n- positive and negative similarity circthresholds [22].

$$L_{intra}^{(b)} = \frac{1}{S}\sum_{s=0}^{S-1} \frac{1}{\ell_b(\ell_b - 1)} \sum_{t=0}^{\ell_b - 1} \sum_{t'\neq t}^{\ell_b - 1} \left[m_{intra} - \hat{L}_{t,s}^{(b)} \cdot \hat{L}_{t',s}^{(b)}\right]_+ \tag{11}$$

, where S – sequence count, ℓ_b – frames within the sequence b, $\hat{L}_{t,s}^{(b)}$ – feature of frame t at position s.

$$L_{inter}^{(b)} = \left[\overline{\hat{L}_{4,t,s}^{(b)} \cdot \hat{L}_{4,t,s'}^{(b)}}\ |_{s\neq s'} - m_{inter}\right]_+ \tag{12}$$

, where $\hat{L}_{4,t,s}^{(b)} \cdot \hat{L}_{4,t,s'}^{(b)}$ – is a similarity check between two feature dimensions

Similarly named function overlaps with existing paper mentioning temporal stripe consistency loss. However, their L_intra solution does not utilize stripes or position dependent structures and L_inter is basically a batch-triplet on unlabeled tracklets which differs from our proposed solution [23].

2.2.3. Market-1501 and MARS Evaluation Results

In the following section model evaluation results are displayed as Rank-1 to Rank-20, mAP, CMC curves and sequence length influence on the model performance. Official evaluation split results are directly comparable to existing published research, while unofficial evaluation split results showcase a best-case scenario. Finally, image sequence query examples will be displayed and commented to determine where the model struggles and excels.

After evaluating model on official Market-1501 and MARS evaluation splits we apply re-ranking as well. Results of evaluation before and after re-ranking are showcased in Table 5.

Table 5. Evaluation results after re-ranking on official splits

Dataset	Method	Rank-1	Rank-5	Rank-10	Rank-20	mAP
MARS (official)	Cosine similarity	0.8960	0.9626	0.9742	0.9808	0.8390
	AQE (k=7, alpha=3)	**0.9131**	-	-	-	**0.8573**
	k-reciprocal reranking (k1=12, lambda = 0.35)	0.8813	-	-	-	0.8491
	Cosine similarity	0.9175	0.9697	0.9831	0.9887	0.8291

Market-1501 (official)	AQE (k=7, alpha=3)	0.9148	-	-	-	0.8526
	k-reciprocal reranking (k1=12, lambda = 0.25)	**0.9344**	-	-	-	**0.8866**

Results displayed in Table 5 are competitive with recently published person re-identification works on both benchmarks. On MARS dataset recently published results show Rank-1 ~0.85-0.93 | mAP ~0.82-0.88 [24] and Market-1501 dataset recently published results show Rank-1 ~0.94-0.97 | mAP ~0.88-0.93 [25].

Official split model evaluation results on MARS before reranking we achieve Rank-1 = ~0.90 and mAP = ~0.84 and after reranking we reach Rank-1 = ~0.91 and mAP = ~0.86. The reranking increases Rank-1 by 0.01 and mAP by 0.02, it may not seem like a lot at first, but it adds up to be an increase in performance by 1% and 2% respectively which displays results to be competitive to top performing solutions.

Official split model evaluation results on Market-1501 before reranking we achieve Rank-1 = ~0.92 and mAP = ~0.83 and after reranking results we reach Rank-1 = ~0.93 and mAP = ~0.89. The reranking increases Rank-1 by 0.01 and mAP by 0.06, which adds up to be an increase in performance by 1% and 6% respectively. The results fall a bit short of existing solutions, however, keep in mind that the query sequences are only 1 image long and gallery sequences are on average 2 images long as mentioned in Table 2.

From this we can conclude that although performance falls short of single image query models, it is still competitive with existing solutions utilizing datasets with longer sequences.

Table 6. Evaluation results after re-ranking on custom splits

Dataset	**Method**	**Rank-1**	**Rank-5**	**Rank-10**	**Rank-20**	**mAP**
MARS (custom)	Cosine similarity	0.9441	0.9712	0.9760	0.9808	0.8914
	AQE (k=7, alpha=3)	0.9233	-	-	-	0.8855
	k-reciprocal reranking (k1=16, lambda = 0.30)	**0.9425**	-	-	-	**0.9081**
Market-1501 (custom)	Cosine similarity	0.9253	0.9827	0.9893	0.9907	0.8455
	AQE (k=7, alpha=3)	0.8880	-	-	-	0.8459
	k-reciprocal reranking (k1=16, lambda = 0.25)	**0.9253**	-	-	-	**0.8857**

Results displayed in Table 6 showcase model performance on a custom split for both Market-1501 and MARS datasets. MARS and Market-1501 dataset splits were built using cross-camera image sequences from test images folder taking one tracklet per person and putting the rest into gallery.

Comparing Rank-1 and mAP results without re-ranking, between Table 5 and Table 6 on MARS dataset Rank-1 and mAP shoots up from 0.90 to 0.94 and 0.84 to 0.89 respectively which is a big 4% and 5% increase in performance.

Comparing Rank-1 and mAP metrics between results on Market-1501 dataset, the improvement isn't as large. Rank-1 and mAP rose from 0.918 to 0.925 and 0.829 to 0.846 respectively, which is a near

negligible increase in Rank-1 and increase in performance of 1.7% for mAP. Although increased in mAP is noticeable it does not increase Rank-1 meaningfully.

This tells us that either the data split was not effective for model performance in Rank-1 or sequence length was lacking to showcase what model can achieve with additional temporal data per tracklet. Going by elimination we see that this split was indeed effective on the MARS evaluation split, so we're left to the conclusion that we lack enough temporal data per tracklet.

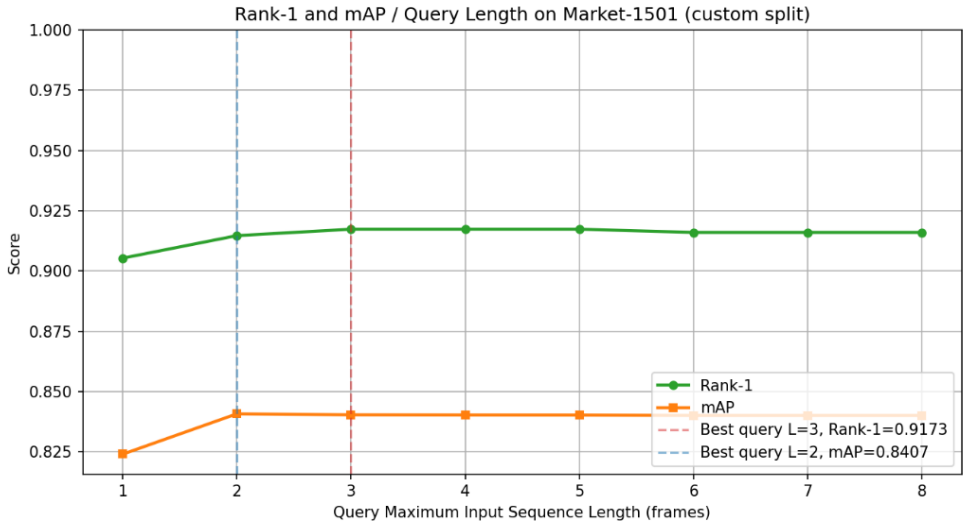

Fig. 28. Rank-1 and mAP dependence on sequence length (Market-1501 query + gallery split)

Fig. 28 shows Rank-1 and mAP score dependence on sequence length of input images. The images contain no junk like background, distractor images and exclude same camera images sequences of a person. Remembering that mean image sequence length of Market-1501 is 1.8 as displayed in Table 2, model performance does not clearly improve with increasing the input sequence length. Best score of Rank-1 = 0.9173 was achieved using MAX_SEQ_LEN=3 and best mAP = 0.8407 was achieved using MAX_SEQ_LEN=2

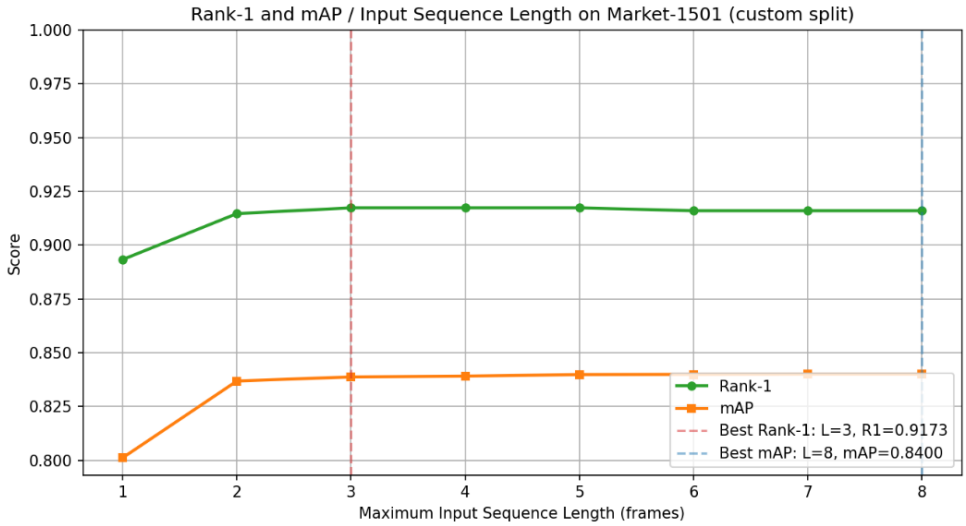

Fig. 29. Rank-1 and mAP dependence on sequence length (gallery split)

Fig. 29 displays the same metrics as in Fig. 28, however query sequences are selected from the gallery as a sequence of different identities while eliminating remaining same-camera images to avoid junk matches of query sequence and gallery sequence being from same PID and camera. Unfortunately, there wasn't any noticeable difference between query image length being 1 image or up to 8 images within a dataset containing minimal amount of 2+ length sequences. Best score of Rank-1 = 0.9173 was achieved using MAX_SEQ_LEN=3 and best mAP = 0.8400 was achieved using MAX_SEQ_LEN=8.

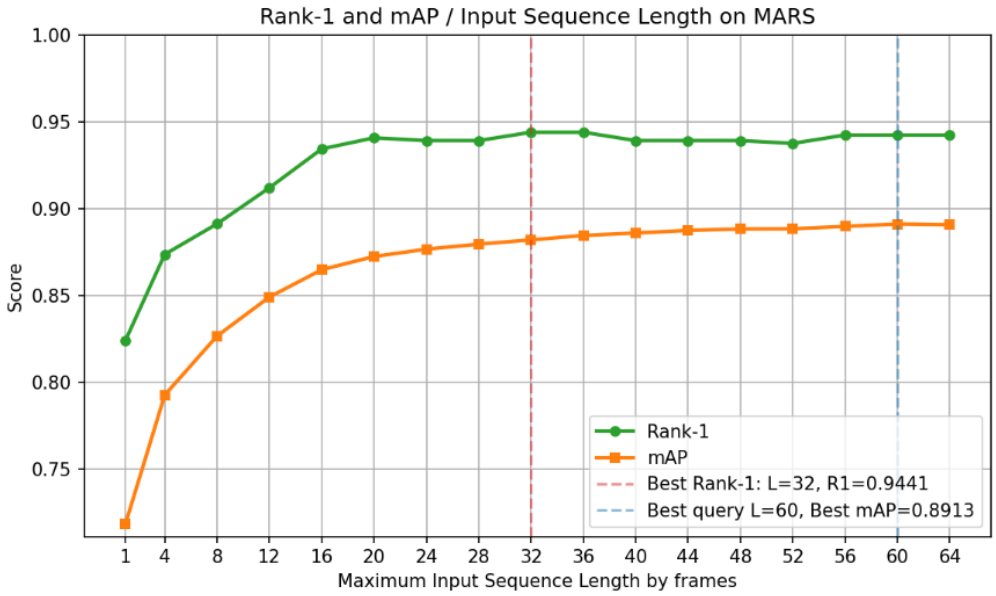

Fig. 30. Rank-1 and mAP dependence on sequence length capping query and gallery at the same rate (MARS)

In the scenario results displayed in Fig. 30 both query and gallery MAX_SEQ_LEN were capped at the same rate. Using this approach time is saved while embedding gallery sequences which would help solve scenarios where gallery size is massive such as gathering all detected person sequences from security cameras across an entire city. Results show that embedding both query and gallery at MAX_SEQ_LEN=1 it starts out with Rank-1=~0.83 and mAP=~0.71. Best score of Rank-1 = 0.9441 was achieved using MAX_SEQ_LEN=32 and best mAP=0.8913 was achieved using MAX_SEQ_LEN=60.

In the scenario results displayed in Fig. 31 only query MAX_SEQ_LEN was capped while gallery MAX_SEQ_LEN was kept at maximum of 64. While always keeping gallery length maximum we notice an increase in mAP=~0.780, but Rank-1=~0.825 stays around the same. We can conclude that capping gallery sequence length at max does help initially, however computational costs do not provide sufficient improvement. Best score of Rank-1 = 0.9425 was achieved using MAX_SEQ_LEN=28 and best mAP = 0.8909 was achieved using MAX_SEQ_LEN=64.

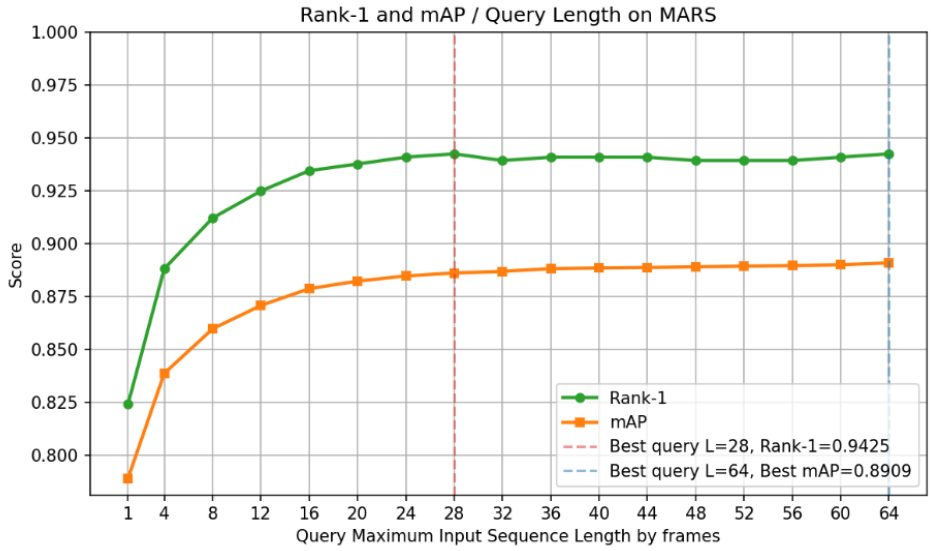

Fig. 31. Rank-1 and mAP dependence on sequence length capping query length only (MARS)

Comparing results from capping only queries and capping both query/gallery we can note that keeping gallery max length at maximum the increase in mAP is noticeable, however not helpful enough to increase Rank-1 metric as well. Computational costs are not justified, and we can determine that it is not worth wasting resources just to increase mAP alone.

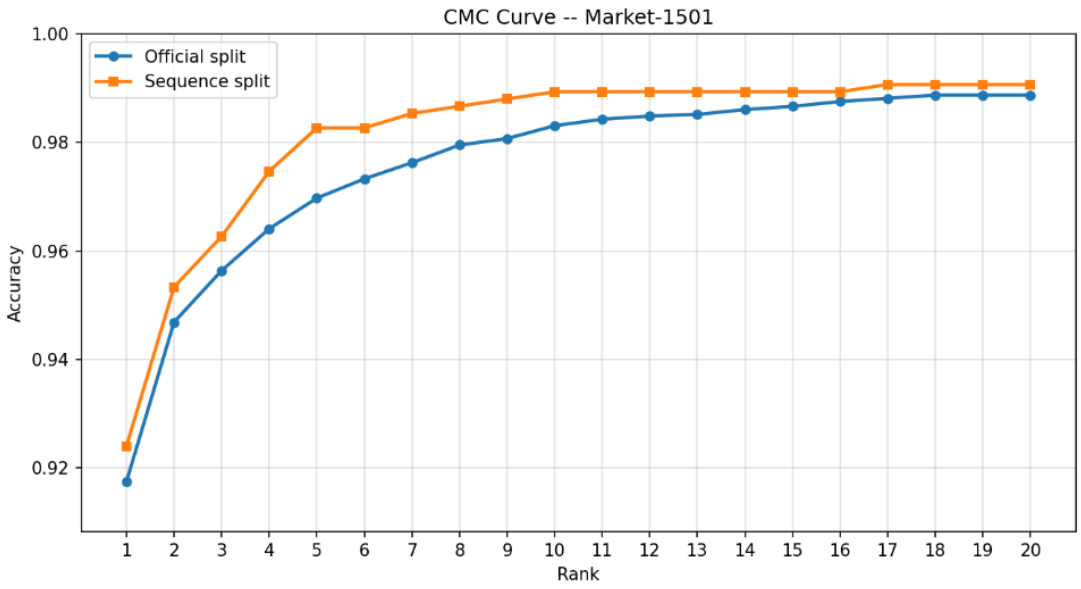

Fig. 32. CMC curve of official and sequence Market-1501 splits

To test out how accuracy is affected by query sequence length an official split containing only 1 image as query input and another sequence split generated using cross-camera sequences so the identity of queried person and camera will never be the same in gallery. For official evaluation purposes all the junk images such as distractor and background images were kept to accurately display model performance and to be comparable to existing published works. From Fig. 32 the trend of sequence split outperforming official 1 image query length sequence split match the expected results.

Model trained on temporal image data naturally achieves better results than passing a single image as query as the generated embeddings do not provide extra information compared to a sequence of images that do.

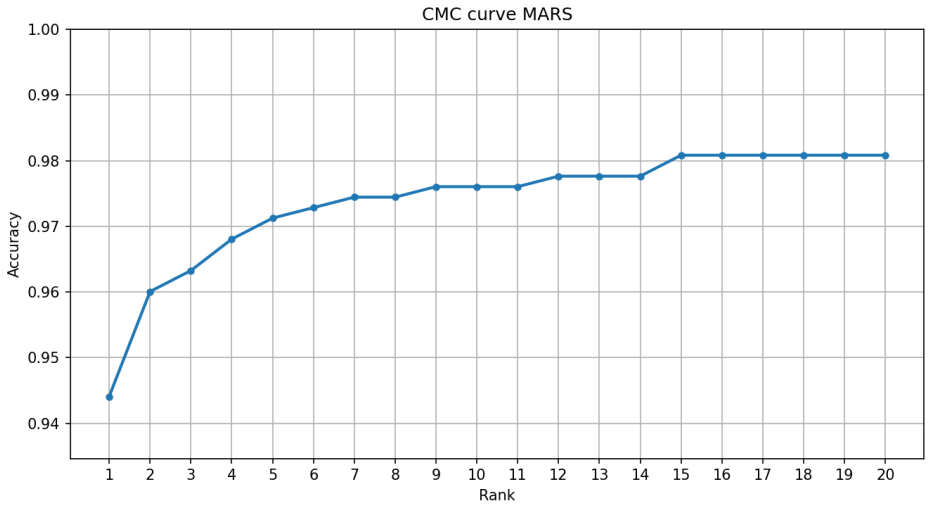

Fig. 33. CMC curve of custom MARS split

The CMC curve presented in Fig. 33 showcases accuracy and rank at which model correctly predicts person identity in the gallery using cosine similarity score. This split is built by taking one query tracklet per person from test folder and putting the rest in the gallery. Cross-camera evaluation split ensures no sequences from the same camera and same person are kept in both gallery and query to avoid score inflation. The split still contains junk images like distractors or backgrounds contained in the original test split. Rank-1 score starts out at ~0.9435, keeps on increasing up to Rank-15 and plateaus at ~0.98 from Rank-15 to Rank-20. Even with all junk images model achieves competitive results with SOTA methods which usually achieve accuracy with all junk files at around Rank-1=94-97%.

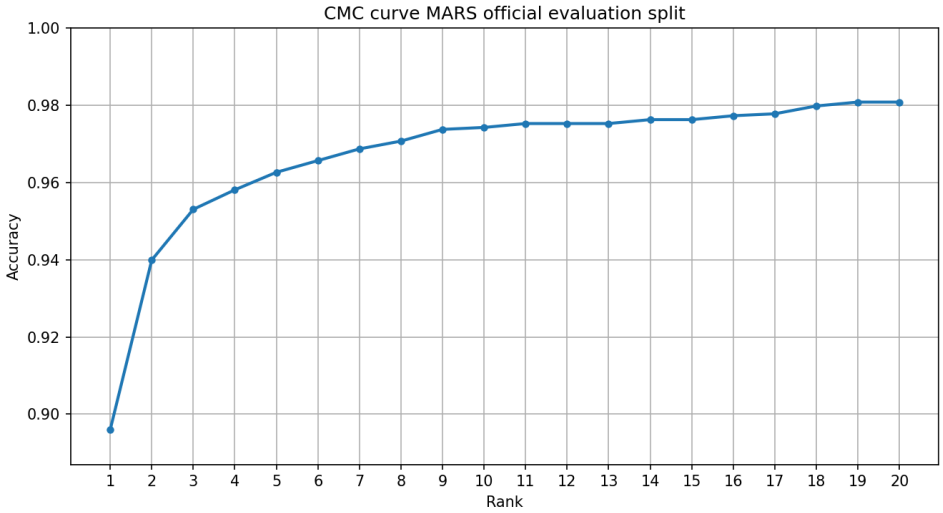

Fig. 34. CMC curve of official MARS split

The CMC curve presented in Fig. 34 displays accuracy and rank at which model correctly predicts person identity in the gallery using cosine similarity score. This split is built using official MARS-evaluation query_IDX.mat file containing person tracklet IDs. Just like before, the gallery contains junk images such as backgrounds and distractors already present in test folder, however in this scenario CMC curve results are directly comparable to currently published works. Compared to Fig. 33 displaying results with unofficial split There's a noticeable drop in Rank-1 From ~0.944 to ~0.898, which taken into account is a difference of almost 5%. Only from Rank-8 to Rank-20 do both splits converge to the same accuracy. From this we can determine that the official split query and gallery images are more difficult examples for the model to correctly predict the embeddings using cosine similarity score.

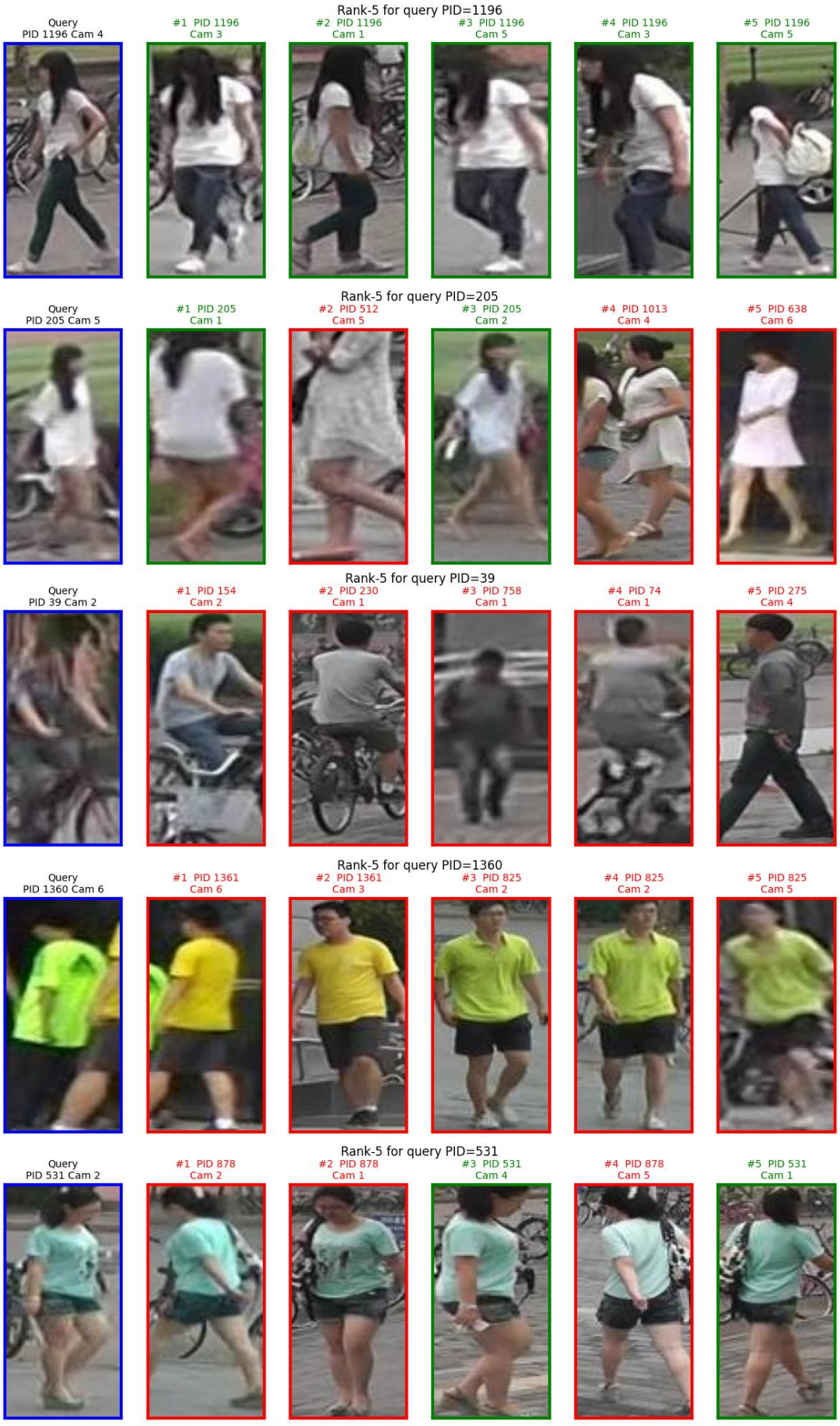

Fig. 35. Model evaluation on Market-1501 image sequences

Model trained on Market-1501 dataset and evaluated on the official query/gallery split. Firstly, the query and gallery sequences were sorted by sequence length and evaluated on using random seed 42 for reproducibility. Results displayed in Fig. 35 contain the image used as query and then the first image of each tracklet in the top 5 (Rank-5) matches. Each query sequence is marked using a blue border, correct matches are marked with a green border, and incorrect matches are marked with a red border respectively.

As seen from the results showcased in Fig. 35, PID=1196 got all correct matches across Rank-5, query contained well detailed information about the person like black hair, white blouse, dark jeans and white shoes which allowed the model to easily find matching sequences with the same features.

On query PID=205 it's unclear what sort of lower wardrobe the person has, so in the resulting images we can see the model found closest matching images all having a white dress/blouse and bare legs of which only 2 out of 5 were correct matches at Rank-1 and Rank-3.

3rd query with PID=39 contains details with least clarity, only a part of the bike, bare arms and grey shirt. From a query with such low detail fidelity model failed to find the right person from gallery at all, however all top5 image seem to match the description and we can determine that model embeds sequences and finds most similar matches as intended.

On the 4th query PID=1360 the model clearly confuses different people with similar colored shirts; however, the hair, shorts and shoes/socks are all almost exactly the same. This right here shows that temporal stripe consistency loss works correctly and the image split up into horizontal stripes catches sequences where 4 out of 5 parts matches to near perfection, but the body (shirt) part only the color is slightly off. Although none of the top 5 (Rank-5) matches queried person, the results are satisfactory.

Finally the 5th of the Market-1501 dataset evaluation query image results PID=531 we can clearly see that all of the sequences match the same person – same shirt, shorts, sandals and even the purse, but Market-1501 dataset is not without flaws and we can deduce the creators of this dataset made a mistake while marking this person either by hand or by machine.

Overall, model trained on the Market-1501 dataset and evaluated using least optimally with query length of only 1 image per query, the results are satisfactory. They display a near-exact match on all example queries, and the model can be used in real world person re-identification tasks. To fully use the capabilities of the trained model an evaluation on MARS dataset will be performed where input sequences are longer than 1 image. This will allow the model to display its ability to utilize temporal information such as tracklets.

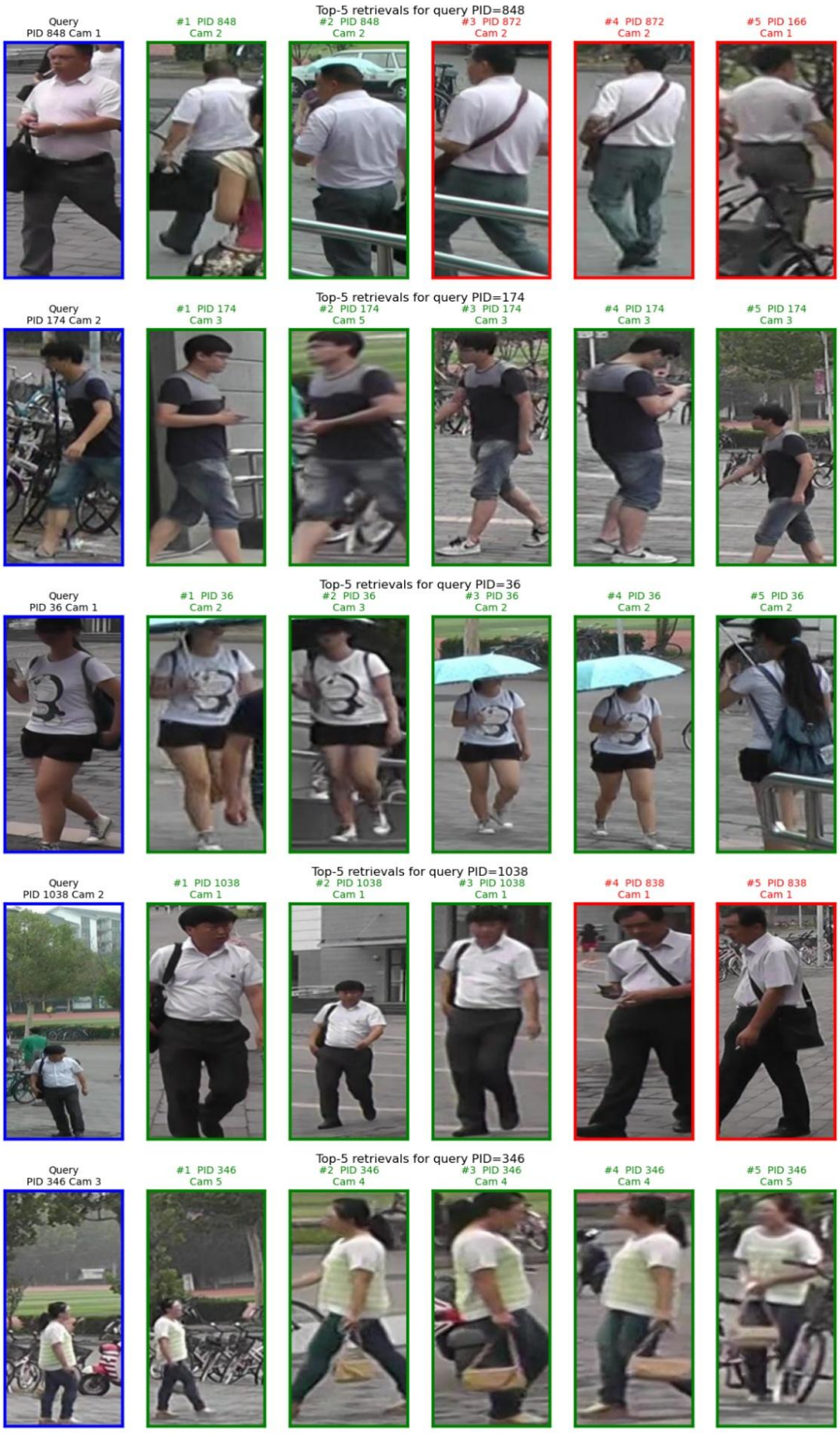

Fig. 36. Model evaluation on MARS image sequences

Model trained on Market-1501 dataset and evaluated on the official query/gallery split. The official evaluation split was created from a query_IDX.mat file which contained query person IDs and the tracklets were loaded from tracks_test_info.mat. Both files were found within the official evaluation GitHub page https://github.com/liangzheng06/MARS-evaluation [14].

Model evaluation results shown in Fig. 36 display 5 query examples of top 5 (Rank-5) matches evaluated on the official query/gallery split using random seed 42 for reproducibility. Evaluation was performed on data splits containing junk such as distractor and background images, but during results display the junk queries were filtered out to see where the model falls short. Each query sequence is marked using a blue border, correct matches are marked with a green border, and incorrect matches are marked with a red border respectively.

On the 1st query PID=848 we can see a short haired man with a relatively solid build, white t-shirt, grey/blue pants and a bag in hand. Model correctly matches the description with predicted sequence embeddings having correct matches Rank-1 and Rank-2, however Rank-3 through Rank-5 were all incorrect, but still close in matching the resemblance to the queried sequence.

Great results on the 2nd query PID=174 we can see a slim man with short hair, distinctly patterned shirt of light and dark grey colors, grey/blue shorts and white sneakers. Model predictions show all correct matches across all results most likely thanks to the distinct features of the shirt which show models ability to distinguish full grey shirts from patterned ones.

Another great model performance example can be seen on the 3rd query PID=36 as well. In it we see a woman with a distinct logo in the middle of the shirt, backpack, black shorts, mostly bare legs, white sneakers and an umbrella. Image sequences from different cameras were all correctly captured as noted by the green borders in the resulting first images from each sequence with a green border. Model once again displayed the ability to pay attention to details rather than generalize everything and find any other person with a white shirt and an umbrella.

On the 4th query PID=1038 we see a man with a white shirt, dark grey pants, short hair and a black back at his side. Although all the returned sequences match the description only the first 3 predictions are correct. This shows a real-world scenario where model will struggle to distinguish different identities when people dress the same around office areas or schools with a specific dress code. One of the ways the model could be improved in this scenario would be having close-up shots of each person's face and running a facial matching model, which in the current scenario is out of scope, but could be considered in further research to improve future person re-identification models.

Finally on the 5th query PID=346 we see a woman with a striped light green/white shirt, a light brown purse, blue jeans, ponytails and brown shoes. The rest of the queries match the correct description and identity. From this we can determine that the increase in unique features a person has from their clothing or accessories, the better the accuracy of matching the correct person from different cameras and sequences is.

From results displayed in evaluation on MARS dataset few quirks of the model pop up. Firstly, the model always finds other sequences closely matching the query sequence which was the whole point of the created sequence-embedding model and we can determine the model is working as expected. Although model correctly finds most similar matches, it will struggle distinguishing people who have similar descriptions like an average office worker or a student in a university with a strict dress-code.

Finally, model displayed a high attention to details in finding most similar sequences of people who had had distinct features like clothing or accessories. For example: umbrellas, bags or uniquely patterned/colored clothing.

Conclusions

1. During the research there were main issues with two of the three used datasets that were found. Out of IUST_PersonReID, Market-1501 and MARS datasets, the IUST_PersonReID didn't contain cropped images where only the person was seen and too much background noise didn't allow the model to train on person identity features, so the first model started training on background noise from specific cameras instead. Market-1501 although a clean dataset with cropped person images, it did not contain long enough tracklets to allow the model to train to its fullest potential and the results were satisfactory, but somewhat lacking. Finally, model trained on the MARS dataset achieved competitive results at the higher end with currently published SOTA methods. From this we can determine that current solution is trained successfully and for further research we would need to test model performance on inference time between our and competitive solutions to determine the superior model.
2. Two different pipelines were created, one on the IUST_PersonReID dataset and another of the Market-1501 and MARS datasets. In total three final models were trained. Each pipeline consisted of data loading, sampling and augmentation which was then passed into the model for training. IUST_PersonReID pipeline calculated embeddings for the entire dataset and loaded them in by id when required to avoid any additional processing time required by re-embedding training data when model training is ran anew. Market-1501 and MARS datasets used the same pipeline where data analysis, splits and augmentation methods used were presented in Table 2, Table 3 and Table 4.
3. Second and third models trained on Market-1501 and MARS datasets were evaluated using official and custom data splits to display research paper comparative and best-case scenario results respectively. From results displayed in Table 5 and Table 6 we concluded that the suggested pipeline is competitive across both datasets. Although Market-1501 official split results of Rank-1=~0.93 and mAP=~0.89 came close to SOTA Rank-1=~0.97 and mAP=~0.93 were satisfactory, more was left to be desired. After evaluating model performance on the MARS dataset official split, the results of Rank-1=~0.91 and mAP=~0.86 were along top performing SOTA methods with Rank-1=~0.93 and mAP=~0.88, the difference being by 0.02 points or 2%/100% shows that the suggested pipeline works nearly as well as best pipelines around.
4. After reviewing second pipeline query evaluation results we can see our model performing quite well when using Market-1501 dataset where data input and gallery sequences are short and official evaluation containing query images of sequences being 1 image long. Visually, all of the query and retrieved images seemed to match the same description so already we can deduce that model works as intended and could be used for real-world person re-identification tasks. When evaluating the second pipeline on MARS dataset, the evaluation results showed a substantially better performance, nearly all the query and retrieved images matched the same correct PID. We also notice that the model struggles distinguishing different people where they dress near exactly the same like office workers or people with a dress code, one of the ways to solve this problem could be by adding a face detection and identification model, but it would be unrealistic and will cause legal and moral concerns if data is collected without consent. Model performed really well with people who had distinct features in clothing or accessories like having an umbrella, bag or uniquely patterned/colored clothing.

Declaration of AI tool usage

During the writing of this research Anthropic Claude was used to generate parts of code responsible for visual results display such as graphs or query images and to translate programming/AI terms from English to Lithuanian.

List of references

1. Jianxin Wu. Introduction to Convolutional Neural Networks, May 1, 2017 [viewed Jan 16, 2025]. Available from: https://cs.nju.edu.cn/wujx/paper/CNN.pdf.

2. *Types of Convolution Kernels.* -07-22, 2024 [viewed Jan 18, 2025]. Available from:https://www.geeksforgeeks.org/types-of-convolution-kernels/.

3. *Image Kernels Explained Visually.* [viewed Jan 18, 2025]. Available from:https://setosa.io/ev/image-kernels/.

4. SUN, Y., et al. Learning Part-Based Convolutional Features for Person Re-Identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, March 1, 2021, vol. 43, no. 3 [viewed Jan 16, 2025]. pp. 902–917. Available from: https://ieeexplore.ieee.org/abstract/document/8826008 ISSN 1939-3539. DOI 10.1109/TPAMI.2019.2938523.

5. LI, J., et al. Pose-Guided Representation Learning for Person Re-Identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, February 1, 2022, vol. 44, no. 2 [viewed Jan 16, 2025]. pp. 622–635. Available from: https://ieeexplore.ieee.org/abstract/document/8764426 ISSN 1939-3539. DOI 10.1109/TPAMI.2019.2929036.

6. GUO, M., et al. Attention Mechanisms in Computer Vision: A Survey. *Computational Visual Media*, September 1, 2022, vol. 8, no. 3 [viewed Jan 19, 2025]. pp. 331–368. Available from: https://doi.org/10.1007/s41095-022-0271-y ISSN 2096-0662. DOI 10.1007/s41095-022-0271-y.

7. QI, Y., et al. Attention-Guided Spatial–temporal Graph Relation Network for Video-Based Person Re-Identification. *Neural Computing and Applications*, July 1, 2023, vol. 35, no. 19 [viewed Jan 19, 2025]. pp. 14227–14241. Available from: https://doi.org/10.1007/s00521-023-08477-1 ISSN 1433-3058. DOI 10.1007/s00521-023-08477-1.

8. CHEN, G., et al. Person Re-Identification Via Attention Pyramid. *IEEE Transactions on Image Processing*, 2021, vol. 30 [viewed Jan 19, 2025]. pp. 7663–7676. Available from: https://ieeexplore.ieee.org/abstract/document/9528019 ISSN 1941-0042. DOI 10.1109/TIP.2021.3107211.

9. BHUIYAN, A. and HUANG, J.X. STCA: Utilizing a Spatio-Temporal Cross-Attention Network for Enhancing Video Person Re-Identification. *Image and Vision Computing*, -07-01, 2022, vol. 123 [viewed Jan 19, 2025]. pp. 104474. Available from: https://www.sciencedirect.com/science/article/pii/S0262885622001032 ISSN 0262-8856. DOI 10.1016/j.imavis.2022.104474.

10. A Comprehensive Overview of Transformer-Based Models: Encoders, Decoders, and More. -07-03, 2024 [viewed Jan 19, 2025]. Available from: https://medium.com/@minh.hoque/a-comprehensive-overview-of-transformer-based-models-encoders-decoders-and-more-e9bc0644a4e5.

11. HE, S., et al. *TransReID: Transformer-Based Object Re-Identification.* , 2021 [viewed Jan 19, 2025]. Available from: https://openaccess.thecvf.com/content/ICCV2021/html/He_TransReID_Transformer-Based_Object_Re-Identification_ICCV_2021_paper.html.

12. ZHENG, L., et al. *Scalable Person Re-Identification: A Benchmark.* IEEE, Dec 1, 2015 Available from: https://ieeexplore.ieee.org/document/7410490 Technology Research Database. DOI 10.1109/ICCV.2015.133.

13. ZHANG, Z., WU, J., ZHANG, X. and ZHANG, C. Multi-Target, Multi-Camera Tracking by Hierarchical Clustering: Recent Progress on DukeMTMC Project, Dec 27, 2017. Available from: https://arxiv.org/abs/1712.09531 DOI 10.48550/arxiv.1712.09531.

14. ZHENG, L., et al. Computer Vision - ECCV 2016Switzerland: Springer International Publishing AG, 2016 *MARS: A Video Benchmark for Large-Scale Person Re-Identification*, pp. 868–884. Available from: http://ebookcentral.proquest.com/lib/SITE_ID/reader.action?docID=5588219&ppg=891&c=UERG ISBN 3319464655. DOI 10.1007/978-3-319-46466-4_52.

15. MOGHADDAM, A.S., et al. IUST_PersonReId: A New Domain in Person Re-Identification Datasets. *arXiv Preprint arXiv:2412.18874*, 2024.

16. M. OQUAB, et al. *DINOv2: Learning Robust Visual Features without Supervision.* February 2, 2024 [viewed May 15, 2026]. Available from:http://arxiv.org/abs/2304.07193 DOI https://doi.org/10.48550/arXiv.2304.07193.

17. Inception-v1 / GoogLeNet (2014) | one minute summary. -09-13, 2021 [viewed Jun 18, 2025]. Available from: https://medium.com/one-minute-machine-learning/going-deeper-with-convolutions-2014-one-minute-summary-dd0e11c4152.

18. WANG, X., et al. Deep Metric Learning by Online Soft Mining and Class-Aware Attention. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33. pp. 5361–5368 DOI 10.1609/aaai.v33i01.33015361.

19. MELGOSA, M., GÓMEZ-ROBLEDO, L., ISABEL SUERO, M. and FAIRCHILD, M.D. What can we Learn from a Dress with Ambiguous Colors?. *Color Research & Application*, 2015, vol. 40, no. 5. pp. 525–529. Available from: https://doi.org/10.1002/col.21966 ISSN 0361-2317. DOI 10.1002/col.21966.

20. WANG, H., et al. *CosFace: Large Margin Cosine Loss for Deep Face Recognition.* IEEE, Jun 2018 Available from: https://ieeexplore.ieee.org/document/8578650 CiNII Research of NDL. DOI 10.1109/CVPR.2018.00552.

21. HERMANS, A., BEYER, L. and LEIBE, B. In Defense of the Triplet Loss for Person Re-Identification, Mar 22, 2017. Available from: https://arxiv.org/abs/1703.07737 DOI 10.48550/arxiv.1703.07737.

22. SUN, Y., et al. *Circle Loss: A Unified Perspective of Pair Similarity Optimization.* Piscataway: IEEE, Jan 1, 2020 Available from: https://ieeexplore.ieee.org/document/9156774 CiNII Research of NDL. DOI 10.1109/CVPR42600.2020.00643.

23. RAYCHAUDHURI, D.S. and ROY-CHOWDHURY, A.K. *Exploiting Temporal Coherence for Self-Supervised One-Shot Video Re-Identification.* Ithaca: Cornell University Library, arXiv.org. Jul 21, 2020 Available from: https://www.proquest.com/docview/2426381948 Publicly Available Content Database.

24. HOU, R., et al. *BiCnet-TKS: Learning Efficient Spatial-Temporal Representation for Video Person Re-Identification.* Ithaca: Cornell University Library, arXiv.org. Apr 30, 2021 Available from: https://www.proquest.com/docview/2521280820 Publicly Available Content Database.

25. CHEN, W., et al. *Beyond Appearance: A Semantic Controllable Self-Supervised Learning Framework for Human-Centric Visual Tasks.* Piscataway: IEEE, Jun 2023 Available from: https://ieeexplore.ieee.org/document/10203783 DOI 10.1109/CVPR52729.2023.01445.

Appendices

Appendix 1. Model training hyperparameters

Table 7. Model training hyperparameters

Parameter group	Parameter	Value
Backbone	Model	vit_base_patch14_dinov2.lvd142m (timm)
	Input size	256x128
	Patch grid	18 x 9 cropped to 16 x 9
	Hidden dim	768
Stripe pooling	Granularities	Global + 2 + 4 + 8 (stripes)
	Granularity dim	512 / 256 / 128 / 64
	Combined features/frame	2048
Temporal Transformer	Layers	2
	Heads	8
	Dropout	0.1
Bi-GRU	Hidden dim	512
	Layers	2
	Dropout	0.3
Pooling head	Pooler	Scalar dot-product attention
	Head	Linear (1024 -> 512) + BatchNorm1D -> L2 norm
	Classifier	Linear (512 -> C) -> L2 norm
	NUM_CLASSES	741 (Market-1501) / 625 (MARS)
CombinedReIDLoss	AMSoftmax scale	30
	AMSoftmax margin	0.35
	Label smoothing	0.1
	Triplet margin	0.3
	Circle margin	0.25
	Circle gamma	80
	Circle lambda	0.2 (Market-1501) / 0.3 (MARS)
	Temporal Stripe Consistency Loss margin intra/inter	0.5 / 0.3
	Temporal Stripe Consistency Loss lambda inter	0.3
	Lambda stripe	0.3
Optimiser (Adam)	LR decay	1 x 10 ^-4
	ViT base LR	1 x 10 ^-6
	Head LR	5 x 10 ^-5
Scheduler	Warm-up	LinearLR start_factor=1^10-3, end_factor=1.0, total_iters=5

	Base	ReduceLROnPlateau mode=max, factor=0.5, patience=5, min_lr = 1*10^-8
	Monitoring	Rank-1 (Market-1501) / Rank-1 + mAP (MARS)
Sampler (PKSampler)	P (identities/batch)	12
	K (sequences/identity)	4
	Batch size	P x K = 48 sequences
	MAX_SEQ_LENGTH	8 (Market-1501) / 64 (MARS)