



A Hybrid Lightweight Deep Learning Framework for Colorectal Cancer Detection Using an Adaptive Gated Feature Fusion and Grid Search–Based Hyperparameter Optimization

Emmanuel Tunbosun Aderemi¹ · Temitayo Matthew Fagbola² · Fagbuagun Ojo Abayomi² ·
Roseline Oluwaseun Ogundokun^{3,4,5} · Pius Adewale Owolawi⁴

Received: 20 April 2026 / Revised: 27 August 2026 / Accepted: 28 August 2026
© The Author(s) 2026

Abstract

The timely and precise diagnosis of colorectal cancer is a critical area for enhancing the quality of care for patients, which could lead to better outcomes. Despite the fact that deep learning has demonstrated great potential in the field of histopathological image analysis, the accuracy of models can be influenced by various factors, including network architecture, optimization method, and data imbalance. The aim of this study is to explore the impact of various optimization algorithms on the classification of colorectal cancer in CRC-VAL-He-7 K database. A hybrid lightweight (HYL) model was designed and tested using the CNN, DeeperCNN, ResNetLike, DenseNet121, EfficientNetB0, and L2RMLP architectures. For the purpose of improving model reliability, it was decided to use grid search for the automatic tuning of the hyperparameters, and class imbalance issues were tackled by the following data augmentation and class-weighted learning techniques: five optimizers: Adam, AdamW, RMSprop, SGD, and Adadelta were evaluated in identical experiments. Both model architecture and the choice of optimizer had a profound influence on the results of classification. The proposed hybrid lightweight model provided competitive accuracy while requiring fewer computational resources, whereas EfficientNetB0 and DenseNet121 exhibited the best overall performance. The results of statistical analysis also showed significant differences between the model–optimizer combinations evaluated. The results offer real-world applications for designing effective and accurate deep learning systems for colorectal cancer diagnosis and highlight the need for optimization and data-balancing approaches in medical image classification.

Keywords Colorectal cancer · Deep learning · Grid search · Optimization algorithms · Hybrid lightweight model · Medical image analysis

Introduction

Colorectal cancer (CRC) ranks as one of the most common types of cancer around the world and has become one of the leading causes of cancer-related deaths [1, 2]. This cancer affects both men and women, and the frequency of its occurrence has been rising over the years [3–5]. Colorectal cancer arises in the colon or rectum and often begins in the form of benign polyps that may convert into cancerous tumors due to late diagnosis [6–8]. More recently, weakly supervised colorectal polyp segmentation has been strengthened through text–image co-alignment, in which structured clinical descriptions provide semantic supervision for localizing abnormal regions [9]. The key to success in treatment is early diagnosis. However, in practice, professionals typically perform the diagnosis manually by examining medical images and slides. This process is lengthy and yields varying results,

✉ Roseline Oluwaseun Ogundokun
ogundokunroseline1@gmail.com;
roseline.ogundokun@ktu.lt

¹ Department of Software Engineering, Nile University of Abuja, Abuja, Nigeria

² Department of Computer Science, Federal University, Oye Ekiti, Ekiti State, Nigeria

³ Department of Multimedia Engineering, Kaunas University of Technology, Kaunas, Lithuania

⁴ Department of Computer Systems Engineering, Tshwane University of Technology, Pretoria, South Africa

⁵ Department of Computer Science, Redeemer's University, Osun State, Ede, Nigeria

depending on the professional's skills in conducting the diagnosis and analyzing the images. This leads to inconsistent outcomes and adds a burden to specialists with experience in this domain. Due to these factors, artificial intelligence, particularly deep learning, is increasingly being applied to medical image analysis, including weakly supervised transformer-based computational pathology systems that support diagnosis and molecular marker discovery [10–13]. Transfer learning and ensemble-based approaches have also been integrated with explainability techniques to improve the transparency and diagnostic utility of medical image classification systems [14–17]. Beyond diagnostic classification, deep convolutional networks have been applied to histopathological images to predict clinically important outcomes such as cancer metastasis and recurrence risk [18]. Diagnostic assessment may also vary depending on the imaging modality used, as MRI and endorectal ultrasonography have distinct strengths and limitations in the preoperative local staging of rectal cancer [19].

Even though advancements have taken place, there are still problems that need to be addressed. Existing studies primarily focus on improving model accuracy by applying deep architectures such as CNNs or transfer learning, but devote less effort to optimizers, hyperparameter sensitivity, training stability, and related aspects. It is also important to note that current work mostly examines binary classification, thus not considering the complexity of the variations that exist in colorectal tissues.

Current medical AI research is also moving toward multimodal frameworks that combine imaging and clinical information for screening, staging, and risk stratification of disease progression [20]. The broader development of multimodal agent-based AI further suggests that future clinical systems may combine visual, textual, and other patient-information sources within interactive decision-support environments [21]. The major limitation of the current studies is that performance is typically reported in a specific training setting, making it difficult to assess the model's stability under divergent optimization conditions. To solve these issues, we propose a hybrid lightweight model for colorectal cancer classification. The present work emphasizes the influence of different optimizers on the model's performance and stability. It aims to develop a more efficient and stable approach to colorectal cancer classification.

Research Contributions

The main contributions of this study are as follows:

- 1) Dual-path lightweight architecture network is developed using complementary MobileNet-inspired and EfficientNet-inspired pathways to capture both local textural and higher-level semantic features from histopathological images.

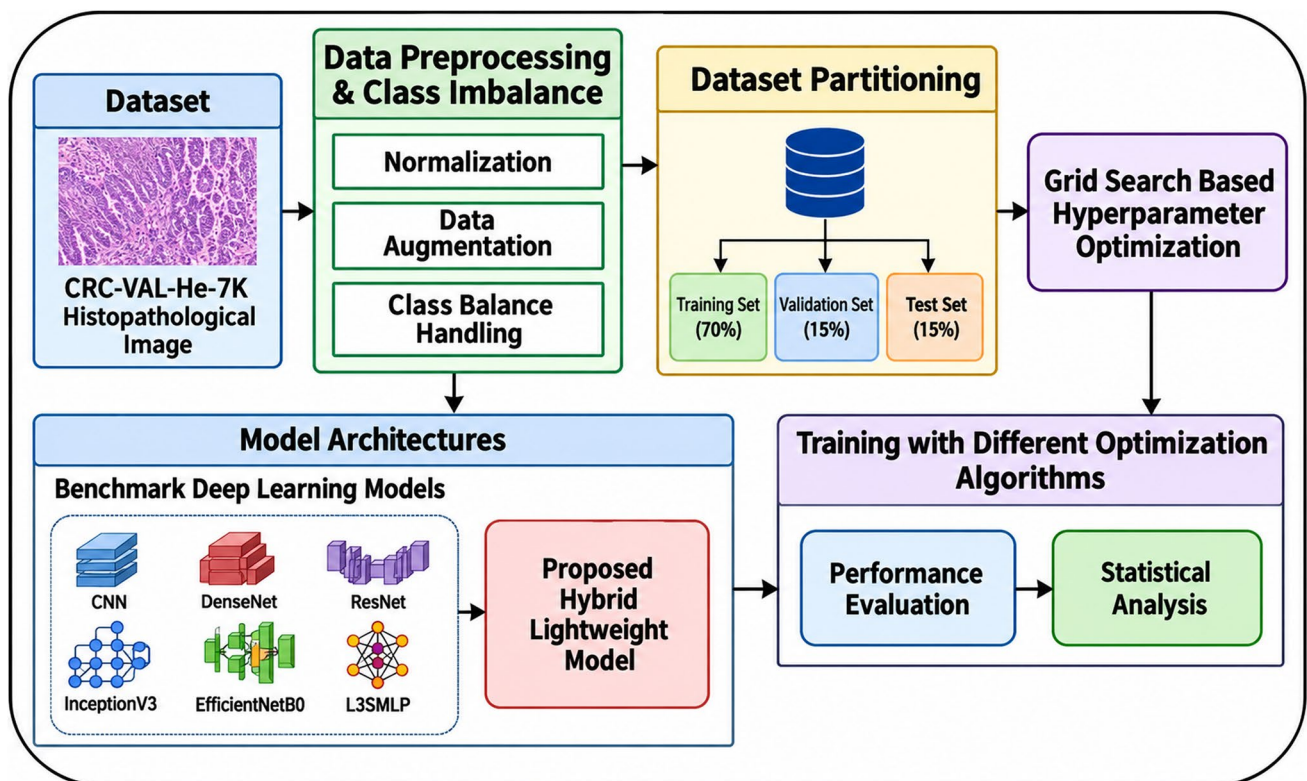


Fig. 1 Conceptual framework of the hybrid lightweight classification model

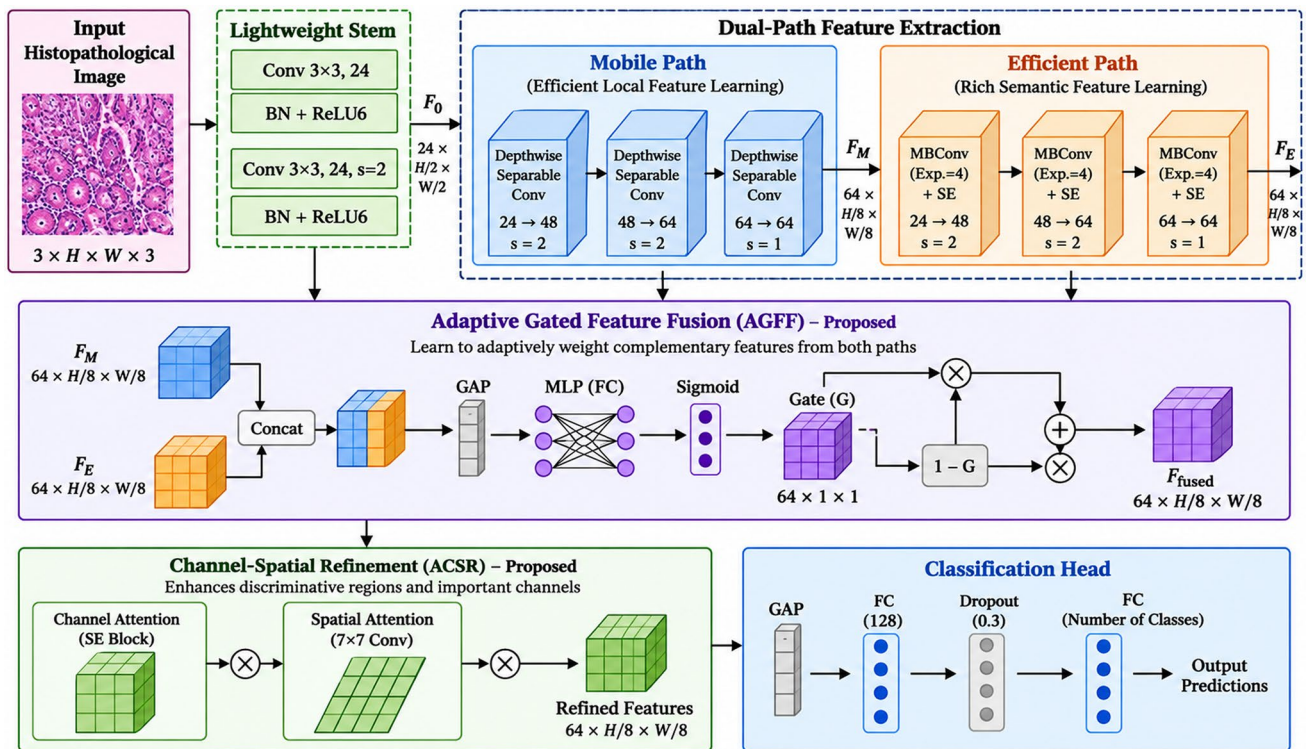


Fig. 2 The proposed adaptive dual-path hybrid lightweight model

- Adaptive Gated Feature Fusion (AGFF) is introduced to learn the relative contribution of the two pathways and adaptively combine their complementary representations instead of simply concatenating them.
- Channel-Spatial Refinement (ACSR) is incorporated to emphasize informative feature channels and spatial regions that are relevant to histopathological classification.
- Computational efficiency is designed to improve feature representation while maintaining a relatively low number of parameters and computational cost, making it suitable for resource-constrained deployment.
- Component-level validation is conducted to quantify the individual contributions of the Mobile path, Efficient path, adaptive gated fusion, and channel-spatial refinement to the overall model performance.

Related Work

Recently, deep learning has emerged as a significant player in colorectal cancer classification, thereby enhancing performance in computer-aided diagnosis systems. Research has been done on different models and their efficiencies and shortcomings. Recent advancements in healthcare AI reveal that the field is moving away from traditional CNN designs toward techniques such as weak supervision, attention mechanisms,

multimodal learning, and predictive models aimed at solving clinical problems. In Jiang et al. [13], a transformer-based method for glioma diagnosis and molecular marker discovery using weak supervision in computational pathology is elaborated. The results show that transformer-based architectures can generate diagnostic insights using histopathology images. The use of explainable ensemble transfer learning for medical image recognition is another example of combining different pretrained models with interpretability methods [16]. In the study by Ye et al. [18], deep learning is successfully applied to predict metastasis and recurrence in patients with cervical cancer. In colorectal imaging, Huang et al. [9] describe a method for weakly supervised polyp segmentation that uses a co-alignment of text and images.

For instance, Sun et al. [22] implemented CNNs in polyp detection in automated medicine and performed transfer learning using datasets outside of medicine to enhance

Table 1 Ablation study results

Variant	Accuracy	Params (M)
Baseline CNN	0.849721707	4.522921
Mobile-style	0.841372913	0.070835
Efficient-style	0.707792208	0.070835
Simple Hybrid	0.933209647	0.070835
Hybrid + Gate	0.938886827	0.070835

Table 2 Computational efficiency analysis (with state-of-the-art models on the CRC-VAL-He-7 K dataset)

Model	Optimizer	Final_ Train_Acc	Final_Val_ Acc	Best_Val_ Acc	Total_ Params	Trainable_ Params	Inference_ Time_ Batch_ms	Inference _Time_Per_ _Image_ms
CustomCNN	Adam	0.82408	0.818013	0.859796	4,522,345	4,522,345	27.64399	0.431937
CustomCNN	SGD	0.573532	0.45311	0.45311	4,522,345	4,522,345	28.6065	0.446977
CustomCNN	RMSprop	0.76995	0.679666	0.752089	4,522,345	4,522,345	27.99003	0.437344
CustomCNN	AdamW	0.823284	0.82637	0.831012	4,522,345	4,522,345	28.86974	0.45109
CustomCNN	Adadelta	0.088358	0.088208	0.088208	4,522,345	4,522,345	28.16629	0.440098
DeeperCNN	Adam	0.845771	0.839369	0.860724	354,409	354,409	28.8177	0.450276
DeeperCNN	SGD	0.548259	0.522748	0.522748	354,409	354,409	28.7783	0.449661
DeeperCNN	RMSprop	0.822886	0.828227	0.883937	354,409	354,409	28.99971	0.45312
DeeperCNN	AdamW	0.84995	0.802228	0.842154	354,409	354,409	28.68588	0.448217
DeeperCNN	Adadelta	0.191244	0.28598	0.289694	354,409	354,409	30.3363	0.474005
ResNetLike	Adam	0.885771	0.904364	0.904364	342,793	342,793	42.85975	0.669684
ResNetLike	SGD	0.780697	0.827298	0.827298	342,793	342,793	42.68384	0.666935
ResNetLike	RMSprop	0.874627	0.877437	0.877437	342,793	342,793	43.17101	0.674547
ResNetLike	AdamW	0.885373	0.923863	0.923863	342,793	342,793	43.03828	0.672473
ResNetLike	Adadelta	0.49592	0.520891	0.520891	342,793	342,793	43.15128	0.674239
HybridLightweight	Adam	0.863284	0.886722	0.886722	86,969	86,969	17.37576	0.271496
HybridLightweight	SGD	0.722786	0.760446	0.765088	86,969	86,969	16.98132	0.265333
HybridLightweight	RMSprop	0.857512	0.438254	0.843083	86,969	86,969	19.01075	0.297043
HybridLightweight	AdamW	0.933682	0.955794	0.955794	86,969	86,969	19.76793	0.308874
HybridLightweight	Adadelta	0.423483	0.442897	0.442897	86,969	86,969	16.92254	0.264415
DenseNet121	Adam	0.95801	0.966574	0.98143	6,956,809	6,956,809	61.91203	0.967375
DenseNet121	SGD	0.973532	0.983287	0.984215	6,956,809	6,956,809	61.33432	0.958349
DenseNet121	RMSprop	0.933333	0.771588	0.941504	6,956,809	6,956,809	61.46641	0.960413
DenseNet121	AdamW	0.959801	0.975859	0.975859	6,956,809	6,956,809	61.79812	0.965596
DenseNet121	Adadelta	0.814726	0.849582	0.849582	6,956,809	6,956,809	61.10893	0.954827
EfficientNetB0	Adam	0.973532	0.987001	0.9935	4,018,501	4,018,501	25.83278	0.403637
EfficientNetB0	SGD	0.943682	0.954503	0.954503	4,018,501	4,018,501	24.56808	0.383876
EfficientNetB0	RMSprop	0.960597	0.987001	0.987001	4,018,501	4,018,501	25.06162	0.391588
EfficientNetB0	AdamW	0.973532	0.985144	0.985144	4,018,501	4,018,501	25.11737	0.392459
EfficientNetB0	Adadelta	0.543284	0.62117	0.62117	4,018,501	4,018,501	25.76068	0.402511
L2RMLP	Adam	0.499303	0.538533	0.538533	17,472,265	17,472,265	2.170408	0.033913
L2RMLP	SGD	0.49194	0.537604	0.539461	17,472,265	17,472,265	2.08612	0.032596
L2RMLP	RMSprop	0.50408	0.55896	0.55896	17,472,265	17,472,265	2.204685	0.034448
L2RMLP	AdamW	0.4999	0.535747	0.536676	17,472,265	17,472,265	2.449751	0.038277
L2RMLP	Adadelta	0.340498	0.438254	0.438254	17,472,265	17,472,265	2.118321	0.033099

Table 3 Model complexity (with state-of-the-art models on the CRC-VAL-He-7 K dataset)

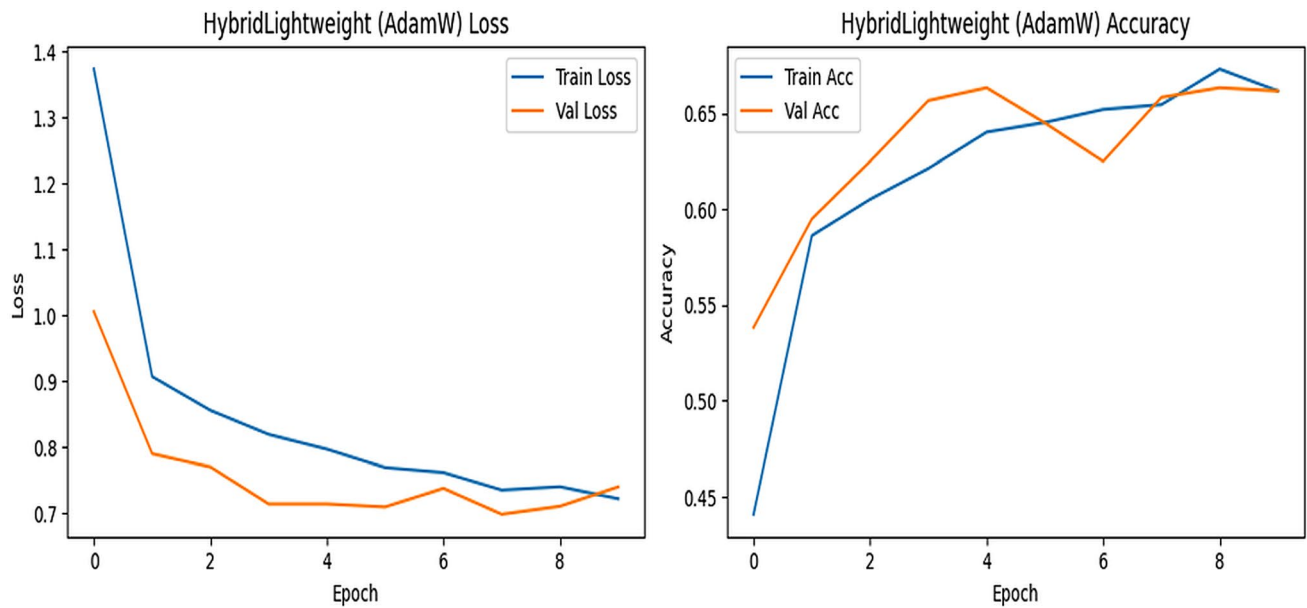
Model	Optimizer (best run)	Test accuracy	Total Params (M)	Trainable Params (M)	Inference (ms/img)	Batch time (ms)
CustomCNN	Adam	0.799010513	4.522345	4.522345	0.462409547	29.59421102
DeeperCNN	AdamW	0.849103278	0.354409	0.354409	0.442231841	28.30283782
ResNetLike	AdamW	0.893320965	0.342793	0.342793	0.680747497	43.56783979
HybridLightweight	AdamW	0.938886827	0.086969	0.086969	0.266471987	17.05420718
DenseNet121	SGD	0.980210266	6.956809	6.956809	0.970962293	62.14158675
EfficientNetB0	Adam	0.987012987	4.018501	4.018501	0.408911968	26.17036595
L2RMLP	RMSprop	0.534632035	17.472265	17.472265	0.02483914	1.589704962

Table 4 The comparison with state-of-the-art models on the CRC-VAL-He-7 K dataset

Model	Optimizer	Accuracy	Precision	Recall	F1_score
CustomCNN	Adam	0.816327	0.808809	0.816327	0.806297
CustomCNN	SGD	0.449907	0.316482	0.449907	0.360213
CustomCNN	RMSprop	0.69295	0.686207	0.69295	0.648151
CustomCNN	AdamW	0.825603	0.817731	0.825603	0.811167
CustomCNN	Adadelata	0.088126	0.007766	0.088126	0.014274
DeeperCNN	Adam	0.846939	0.860126	0.846939	0.832666
DeeperCNN	SGD	0.525974	0.374899	0.525974	0.423773
DeeperCNN	RMSprop	0.846939	0.849334	0.846939	0.837872
DeeperCNN	AdamW	0.816327	0.86331	0.816327	0.818702
DeeperCNN	Adadelata	0.280148	0.129658	0.280148	0.160652
ResNetLike	Adam	0.912801	0.915976	0.912801	0.909149
ResNetLike	SGD	0.834879	0.821141	0.834879	0.816998
ResNetLike	RMSprop	0.899814	0.894793	0.899814	0.892871
ResNetLike	AdamW	0.923933	0.924162	0.923933	0.921097
ResNetLike	Adadelata	0.533395	0.436241	0.533395	0.449043
HybridLightweight	Adam	0.909091	0.915869	0.909091	0.90924
HybridLightweight	SGD	0.766234	0.704404	0.766234	0.730768
HybridLightweight	RMSprop	0.455473	0.522735	0.455473	0.444332
HybridLightweight	AdamW	0.931874	0.938485	0.931874	0.931513
HybridLightweight	Adadelata	0.44898	0.363082	0.44898	0.347638
DenseNet121	Adam	0.961967	0.962688	0.961967	0.961481
DenseNet121	SGD	0.981447	0.982181	0.981447	0.981184
DenseNet121	RMSprop	0.772727	0.881298	0.772727	0.71851
DenseNet121	AdamW	0.973098	0.973313	0.973098	0.97307
DenseNet121	Adadelata	0.843228	0.827041	0.843228	0.825624
EfficientNetB0	Adam	0.987013	0.987096	0.987013	0.986818
EfficientNetB0	SGD	0.951763	0.95183	0.951763	0.950753
EfficientNetB0	RMSprop	0.988868	0.988867	0.988868	0.988832
EfficientNetB0	AdamW	0.986085	0.986179	0.986085	0.986055
EfficientNetB0	Adadelata	0.62616	0.621651	0.62616	0.602059
L2RMLP	Adam	0.540816	0.542943	0.540816	0.471108
L2RMLP	SGD	0.539889	0.423454	0.539889	0.465968
L2RMLP	RMSprop	0.546382	0.440807	0.546382	0.47038
L2RMLP	AdamW	0.541744	0.440411	0.541744	0.472386
L2RMLP	Adadelata	0.437848	0.472983	0.437848	0.356429

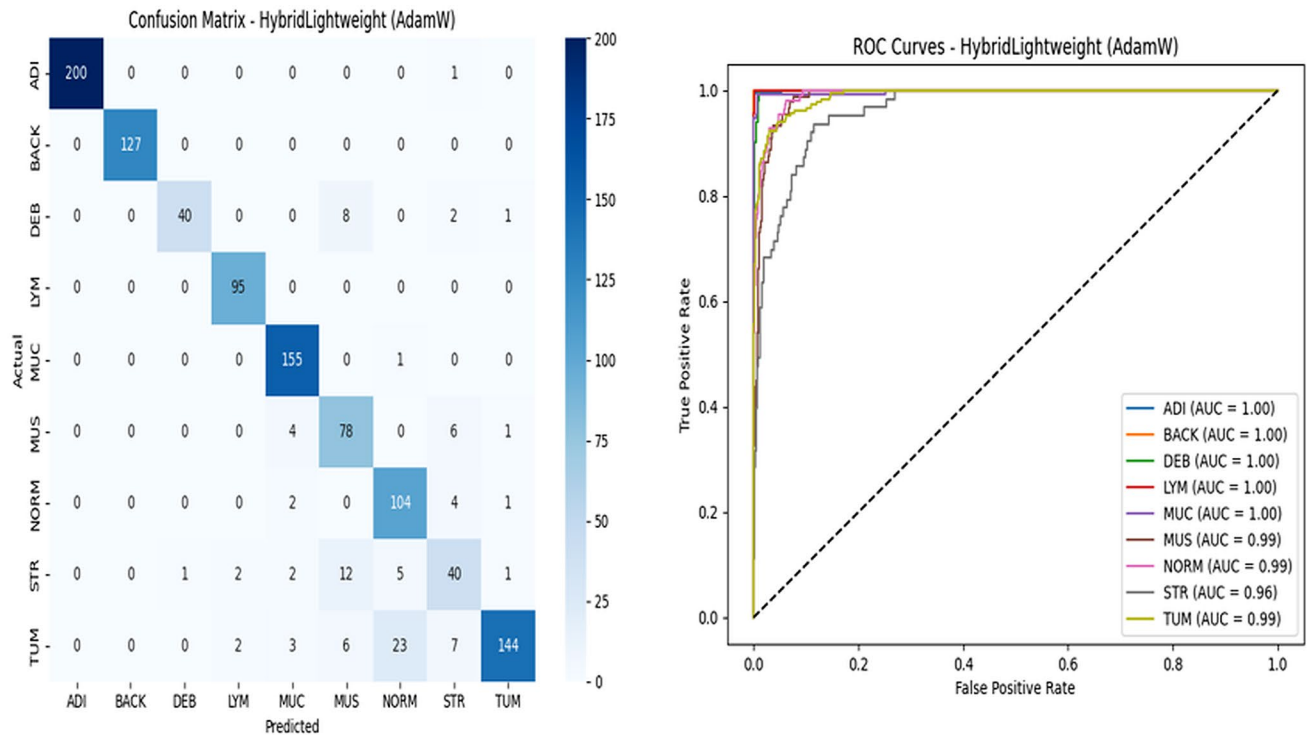
efficiency. The study by Zhang et al. [23] introduced RCC-Net, a lightweight CNN for binary classification of colon cancer nuclei. Notably, the performance of this CNN in terms of correct classification is about 80.61%, with an F1-score of about 0.7887; however, it shows a decline in performance when challenged by complex data, suggesting poor generalizability. Likewise, Shawesh and Chen [5] implemented a CNN based on ResNet-50 to classify colorectal cancer, achieving an accuracy of about 97.7% on the NCT database. Yet again, it only raises concerns about how well the model performs in real-world scenarios, since transfer learning is applied to datasets with variable distributions. Another study was conducted on CT-based detection using a neural network and the firefly optimization algorithm [24].

In the study by Sujatha et al. [25], a framework utilizing a two-stage CNN process was presented for classifying colon images, with the best performance achieved by the implementation based on the VGG19 model. Despite this success, the implementation in question required substantial preprocessing and manual labelling, making it less practical for large-scale deployment. Also, in Sharma et al.'s [24] study, an encoder-decoder model was introduced for polyp segmentation, achieving good average performance but showing low accuracy on clinical images collected during various procedures. In the article referenced as [26], the combination of deep transfer learning with color normalization brought about the problem of histological classification with high accuracy—96.2%; however, it is noted that the dependence on color normalization can have a negative impact on the robustness of the technique in



(a) Training and Validation Loss over epochs. (b) Training and Validation Accuracy over epochs CRC-VAL-He-7K dataset

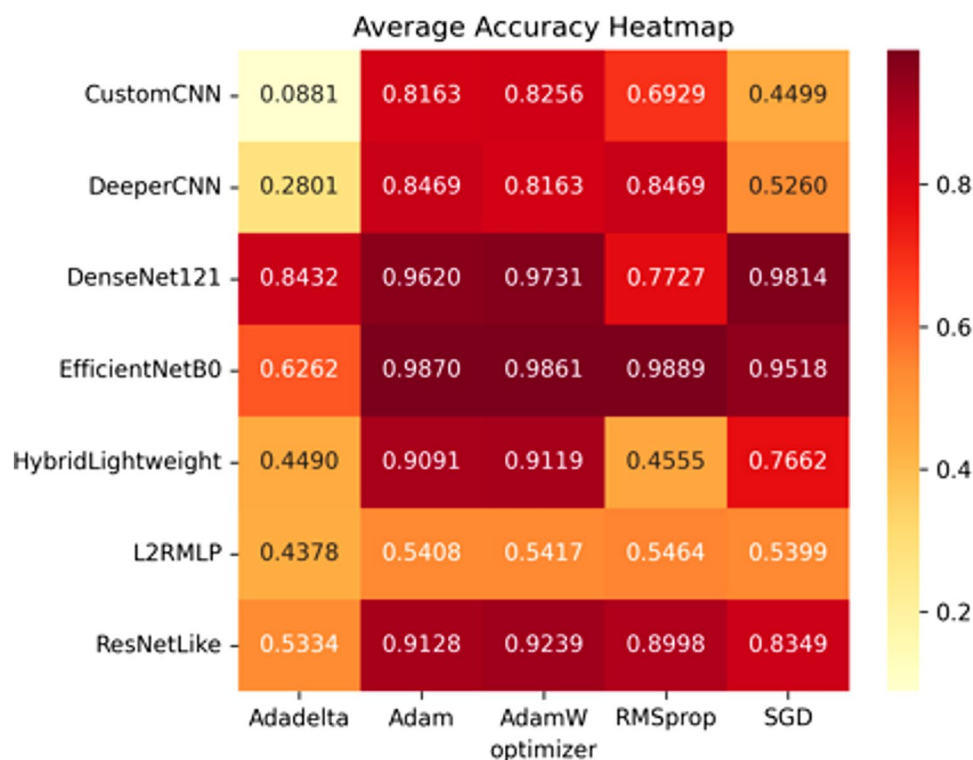
Fig. 3 Training dynamics of the proposed hybrid lightweight model



(c) Confusion Matrix over training epochs. (d) ROC over training epochs over CRC-VAL-He-7K dataset

Fig. 4 Training dynamics of the confusion matrix and ROC of the proposed hybrid lightweight model

Fig. 5 Present the average accuracy heatmap on CRC-VAL-He-7 K dataset



case the grayscale pictures are processed or different staining methods are applied. In the work presented in [14], an impressive performance of 98.4% was achieved, yet the complex architecture required more computational resources during training and implementation. The DenseNet architecture has been applied in [27], achieving 97.2% performance.

However, the absence of preprocessing in this case might make the algorithm less effective at handling noisy, low-quality images. The article [7] shows that good results were obtained through hyperparameter education, but at high resource costs and with poor reproducibility of the outcomes. Some progress was made in [28]; however, the organization in question proposed a semi-supervised approach, reducing the number of necessary annotations, yet performance may vary significantly across institutions due to image quality. Architectural adaptation has also been investigated for small-sample biomedical datasets, with Cheng et al. [29] proposing a specialized deep learning framework to improve feature learning from limited Raman spectroscopy observations. In summary, [28] introduced color CADx, a hybrid system that combines CNNs with the characteristics of the discrete cosine transformation, thereby enabling improved medical diagnostic assistance. The research outlines important advancements in deep learning applications for colorectal cancer classification. However, some shortcomings of the mentioned studies can be noted. First, there are not enough papers focusing on the impact of different types of optimizers on the models, while others do not even describe the process of training stability. Secondly, the studies rely mostly on simple binary

classification and overlook the nature of connective tissue in the colorectal organs. Models such as ResNet, DenseNet, EfficientNet, and various hybrid CNNs demonstrate excellent accuracy but come with disadvantages, including high computational cost and enormous data requirements. Multi-branch attention architectures have also been explored in other medical data modalities; for example, Chen et al. [30] combined a dual-branch attention network with serum Raman spectroscopy to diagnose diabetic kidney disease. Nevertheless, the findings were derived from spectroscopic signals and may not be directly transferable to colorectal histopathology images. More recently, Liu et al. [31] developed a dual-branch Faster R-CNN for breast ultrasound image detection by combining ResNet50 and Pyramid Vision Transformer pathways to exploit complementary local and global feature representations. Although the modality and detection task differ from colorectal histopathology classification, the study provides a relevant methodological precedent for parallel feature-extraction branches and feature fusion in medical imaging. Qian et al. [32] also proposed a multi-view, multi-feature evolutionary deep fusion method for protein secondary structure prediction, demonstrating the broader value of combining complementary representations in biomedical deep learning tasks.

The rest of the article is organized as follows: the “[Related Work](#)” section presents the literature review; the “[Materials and Methods](#)” section describes the methodology; the “[Results and Discussion](#)” section presents the experiments and results; finally, the “[Conclusion and Future Work](#)” section provides the conclusion.

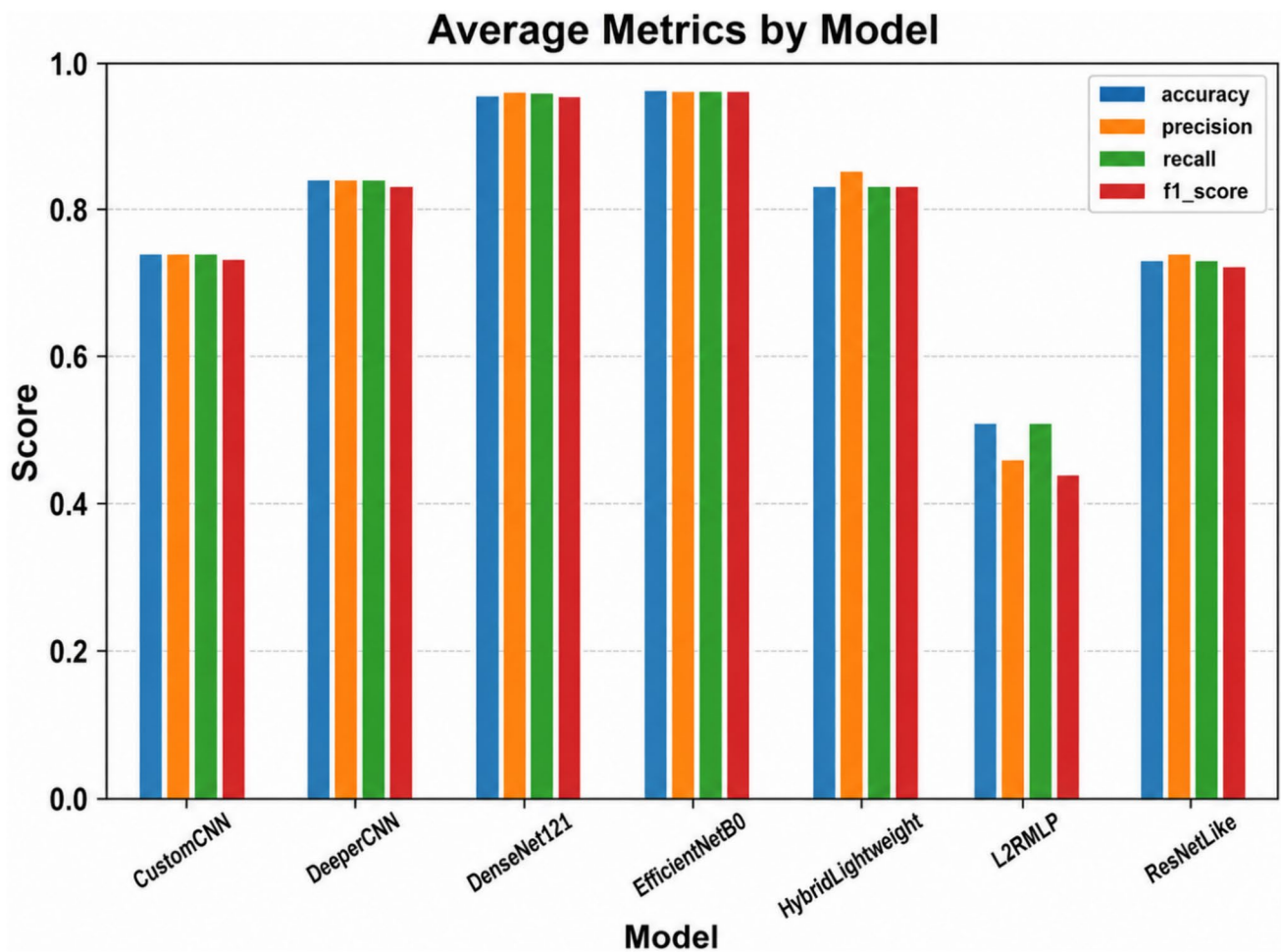


Fig. 6 Present average evaluation metrics by model on CRC-VAL-He-7 K dataset

Materials and Methods

Datasets

Tasks involved in the analysis of medical images, such as pixel-level classification and segmentation, are highly challenging due to the complexity of histopathological data. In our research, we tested deep learning models on a single dataset for detecting colorectal cancer. The dataset represents different tissue types in various ways, enabling easier interpretation and proper training of the model. The CRC-Val-He-7 K dataset comprises 7000 histological images that belong to nine categories, namely adipose tissue (ADI), background (BACK), debris (DEB), lymphocytes (LYM), mucus (MUC), smooth muscle (MUS), normal mucosa (NORM), stroma (STR), and tumor (TUM). This dataset contains 7000 grayscale images of 128 by 128 pixels. The dataset can be found on Kaggle. Figure 1 provides a view of the hybrid lightweight classification model proposal.

Data Preprocessing

All datasets were standardized to ensure uniformity. Images in the datasets were resized in terms of pixel resolution to be 128×128 pixels in size. Preprocessing included scaling the pixel values to a range of $[0, 1]$ while converting images to grayscale wherever necessary. Categorical labels were also converted into numerical values by means of label encoding to ensure compatibility with machine learning models. Moreover, data augmentation was employed to prevent the models from overfitting and to increase their ability to generalize to the data. Hence, these measures made it possible to develop a clean and uniform dataset:

Data Normalization

The dataset was normalized using z-score standardization, which adjusts pixel values based on the dataset's mean and standard deviation. This ensures that all features are on a common scale, improving training stability and making

model learning more effective. The normalization process is defined as follows:

$$x' = \frac{x - \text{mean}(x)}{\text{std}(x)} \quad (1)$$

Algorithm 1: Z-Score Normalisation Algorithm

where x' is the normalized value of x , $\text{mean}(x)$ is the mean of the feature x , and $\text{std}(x)$ is the standard deviation of the feature x . The normalization formula is applied to each value in the dataset to transform the values into the desired range or distribution. This process is typically performed separately for each feature, ensuring that each feature is normalized independently of the others.

Input: dataset (List of images with pixel values)

Output: normalized_dataset

1. Initialise an empty list `normalized_dataset`
2. Flatten all images in the dataset into a single list `all_pixels`
3. `pixel_mean = mean(all_pixels)`
4. `pixel_std = standard_deviation(all_pixels)`
5. **for each** image in the dataset:
6. Initialise an empty array `normalized_image`
7. **for each** pixel in image:
8. `normalized_pixel = (pixel - pixel_mean) / (pixel_std + epsilon)`
9. Append `normalized_pixel` to `normalized_image`
10. Append `normalized_image` to `normalized_dataset`
11. **return** `normalized_dataset`

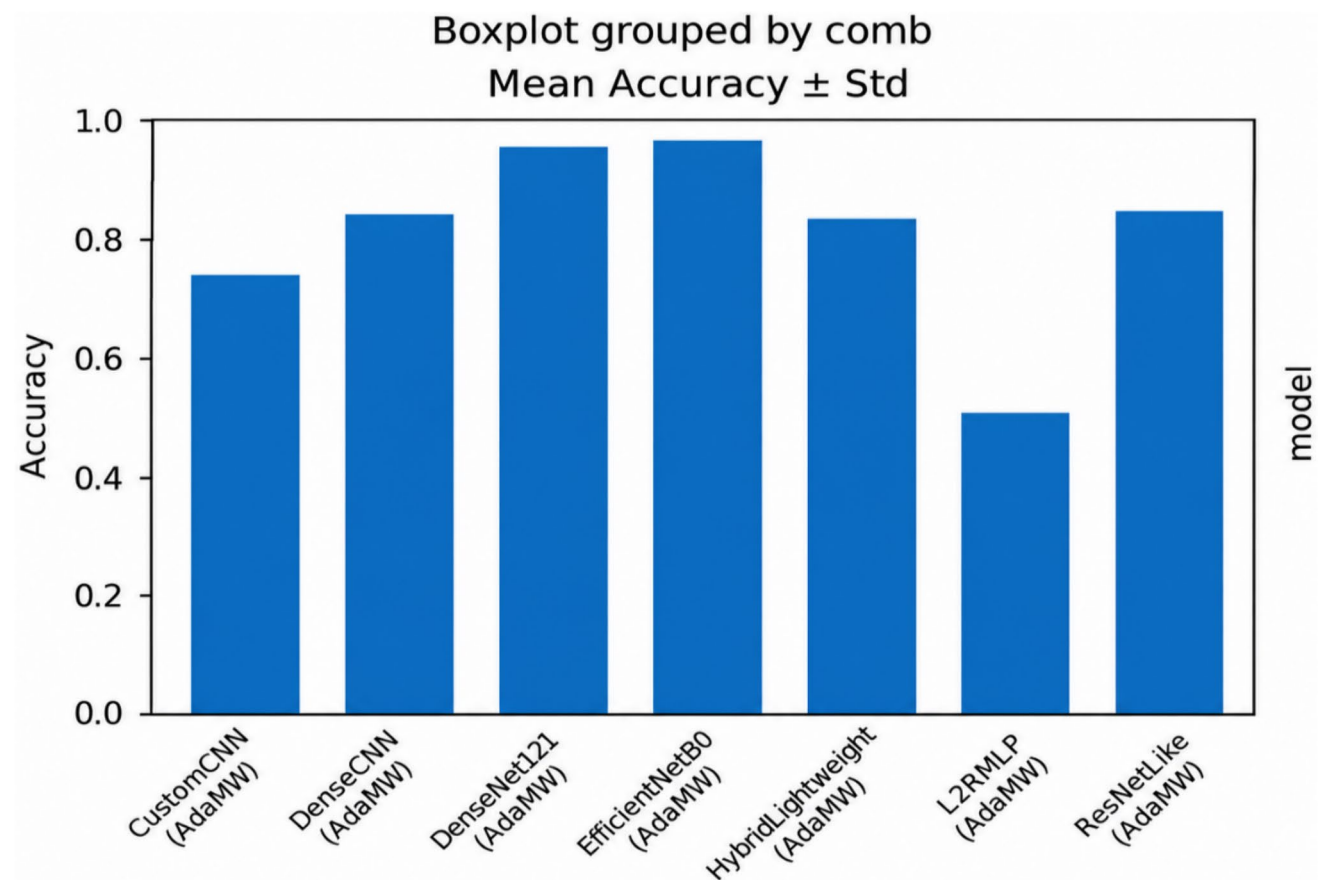


Fig. 7 Mean accuracy and model stability on CRC-VAL-He-7 K dataset

Data Augmentation

To increase the variation in the training images, conventional data augmentation techniques were applied. The augmentation included random rotation, horizontal and vertical flipping, and random cropping. These transformations change the orientation and spatial arrangement

of the images while retaining the original image features. This approach was preferred over generative augmentation because it does not create new tissue structures or modify the underlying morphology of the images. The augmented images were used only during training, while the validation and test images were kept unchanged for an unbiased evaluation of the model [6].

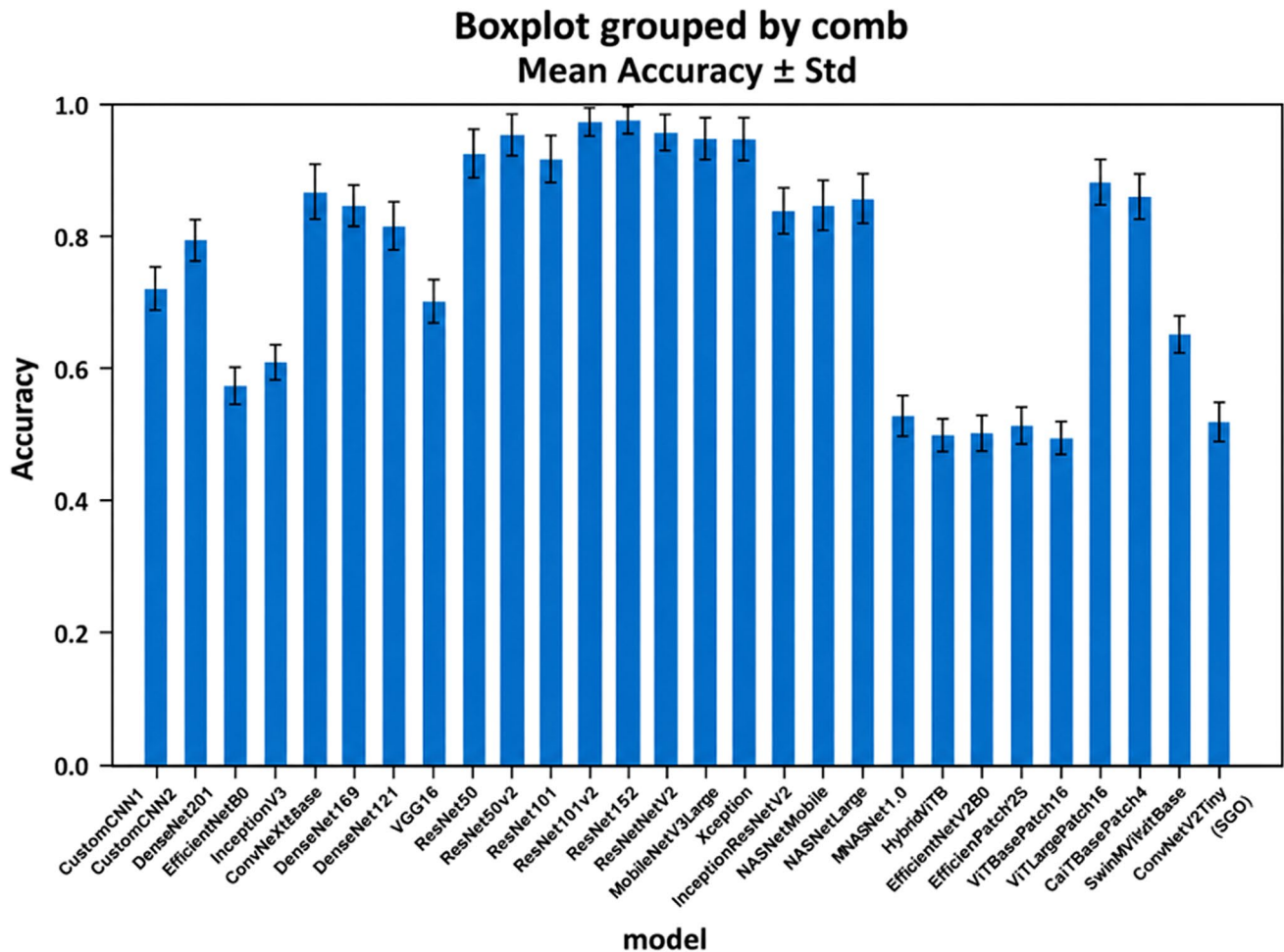


Fig. 8 Mean classification accuracy of compared models with standard deviation

Algorithm 2: Data Augmentation Algorithm

Input: normalized_dataset (normalized training images from Algorithm 1), rotation_range, flip_probability, crop_parameters
Output: augmented_dataset

1. Initialize an empty list, augmented_dataset
2. **for each** image in normalized_dataset:
3. Randomly sample a rotation angle within rotation_range
4. rotated_image = Rotate(image, rotation angle)
5. Randomly apply horizontal and/or vertical flipping according to flip_probability
6. flipped_image = Flip(rotated_image)
7. Randomly crop flipped_image according to crop_parameters
8. augmented_image = Resize(cropped_image, target image size)
9. Append augmented_image to augmented_dataset
10. **end for**
11. Combine normalized_dataset and augmented_dataset to form final_dataset
12. **return** final_dataset

Label encoding was a process in which every category intrinsic to a categorical variable was assigned a specific integer label. This operation assigned an arbitrary but unique numerical value to each discrete category, thereby enabling subsequent computational operations on the transformed data. This process of label encoding is formalized as a concise mathematical formulation. Supposed C represents the categorical variable with n distinct categories, denoted as

$C = \{C_1, C_2, \dots, C_n\}$. Label encoding maps each category C_i to a unique integer label L_i , where $1 \leq i \leq n$. The label encoding function f is defined as:

$$f(C_i) = L_i \quad (2)$$

where f is the encoding function, C_i represents the categorical variable, and L_i denotes the unique integer label.

Algorithm 3: Label Encoding Algorithm

Input: data
Output: encoded_data, label_encoder (A dictionary mapping original labels to integers)

1. Initialise an empty dictionary label_encoder
2. unique_labels = set(data)
3. code = 0
4. **for each** label in unique_labels:
5. label_encoder[label] = code
6. code += 1
7. Initialise an empty list encoded_data
8. **for each** item in data:
9. Append label_encoder[item] to encoded_data
10. **return** encoded_data, label_encoder

Table 5 Computational efficiency analysis (with state-of-the-art models on the Kvasir dataset)

Model	Optimizer	Final_ Train_Acc	Final_ Val_Acc	Best_ Val_Acc	Total_ Params	Trainable_ Params	Inference_ Time_ Batch_ms	Inference_ Time_per_ Image_ms
CustomCNN	Adam	0.615934	0.583333	0.651667	13,173,608	13,173,608	84.22716	1.316049
CustomCNN	SGD	0.462308	0.51	0.51	13,173,608	13,173,608	84.673	1.323016
CustomCNN	RMSprop	0.594141	0.638333	0.638333	13,173,608	13,173,608	84.43973	1.319371
CustomCNN	AdamW	0.588782	0.553333	0.658333	13,173,608	13,173,608	82.69279	1.292075
CustomCNN	Adadelata	0.134334	0.125	0.125	13,173,608	13,173,608	85.38616	1.334159
DeeperCNN	Adam	0.602001	0.665	0.665	354,856	354,856	90.33515	1.411487
DeeperCNN	SGD	0.386567	0.403333	0.403333	354,856	354,856	90.41755	1.412774
DeeperCNN	RMSprop	0.623794	0.621667	0.633333	354,856	354,856	90.59319	1.415519
DeeperCNN	AdamW	0.613076	0.645	0.676667	354,856	354,856	91.57991	1.430936
DeeperCNN	Adadelata	0.130761	0.125	0.125	354,856	354,856	91.40491	1.428202
ResNetLike	Adam	0.66881	0.683333	0.701667	343,112	343,112	131.5849	2.056015
ResNetLike	SGD	0.617006	0.625	0.625	343,112	343,112	131.1368	2.049013
ResNetLike	RMSprop	0.659521	0.68	0.68	343,112	343,112	131.5333	2.055208
ResNetLike	AdamW	0.663451	0.711667	0.711667	343,112	343,112	131.4553	2.053989
ResNetLike	Adadelata	0.360843	0.401667	0.403333	343,112	343,112	131.8507	2.060167
HybridLightweight	Adam	0.662379	0.66	0.671667	87,128	87,128	54.95822	0.858722
HybridLightweight	SGD	0.585924	0.63	0.63	87,128	87,128	55.09706	0.860891
HybridLightweight	RMSprop	0.665595	0.681667	0.698333	87,128	87,128	54.63221	0.853628
HybridLightweight	AdamW	0.661665	0.661667	0.663333	87,128	87,128	54.09622	0.845253
HybridLightweight	Adadelata	0.314398	0.41	0.41	87,128	87,128	53.80423	0.840691
DenseNet121	Adam	0.737406	0.625	0.733333	6,962,056	6,962,056	194.0133	3.031458
DenseNet121	SGD	0.680957	0.72	0.72	6,962,056	6,962,056	193.358	3.021219
DenseNet121	RMSprop	0.673812	0.613333	0.678333	6,962,056	6,962,056	190.9909	2.984232
DenseNet121	AdamW	0.72776	0.641667	0.685	6,962,056	6,962,056	191.1061	2.986032
DenseNet121	Adadelata	0.461951	0.486667	0.486667	6,962,056	6,962,056	195.2183	3.050286
EfficientNetB0	Adam	0.687388	0.746667	0.746667	4,017,796	4,017,796	74.87578	1.169934
EfficientNetB0	SGD	0.63487	0.603333	0.65	4,017,796	4,017,796	76.35109	1.192986
EfficientNetB0	RMSprop	0.667024	0.686667	0.686667	4,017,796	4,017,796	75.05243	1.172694
EfficientNetB0	AdamW	0.695606	0.691667	0.728333	4,017,796	4,017,796	75.60961	1.1814
EfficientNetB0	Adadelata	0.180779	0.235	0.246667	4,017,796	4,017,796	75.40624	1.178223
L2RMLP	Adam	0.547338	0.648333	0.658333	1.55E+08	1.55E+08	9.118009	0.142469
L2RMLP	SGD	0.535191	0.64	0.64	1.55E+08	1.55E+08	9.366727	0.146355
L2RMLP	RMSprop	0.569132	0.651667	0.66	1.55E+08	1.55E+08	9.04963	0.1414
L2RMLP	AdamW	0.549839	0.64	0.665	1.55E+08	1.55E+08	9.183908	0.143499
L2RMLP	Adadelata	0.402644	0.596667	0.598333	1.55E+08	1.55E+08	9.142423	0.14285

Table 6 Model complexity (with state-of-the-art models on the Kvasir dataset)

Model	Optimizer (best run)	Test accuracy	Total Params (M)	Trainable Params (M)	Inference (ms/img)	Batch time (ms)
CustomCNN	RMSprop	0.592346	13.17361	13.17361	1.260765	80.68895
DeeperCNN	Adam	0.668885	0.354856	0.354856	1.328802	85.04336
ResNetLike	AdamW	0.687188	0.343112	0.343112	1.915077	122.565
HybridLightweight	Adam	0.687188	0.087128	0.087128	0.827172	52.93903
DenseNet121	SGD	0.682196	6.962056	6.962056	2.759596	176.6142
EfficientNetB0	RMSprop	0.725458	4.017796	4.017796	1.101309	70.48376
L2RMLP	Adam	0.657238	154.8356	154.8356	0.14227	9.105301

Table 7 The comparison with state-of-the-art models on the C Kvasir dataset

Model	Optimizer	Accuracy _mean	Precision _mean	Recall _mean	F1_score mean
CustomCNN	Adadelta	0.1248	0.0156	0.1248	0.0277
CustomCNN	Adam	0.5424	0.6167	0.5424	0.5305
CustomCNN	AdamW	0.5358	0.6098	0.5358	0.5055
CustomCNN	RMSprop	0.5923	0.6141	0.5923	0.5843
CustomCNN	SGD	0.4842	0.4473	0.4842	0.4256
DeeperCNN	Adadelta	0.1248	0.0156	0.1248	0.0277
DeeperCNN	Adam	0.6689	0.6972	0.6689	0.6476
DeeperCNN	AdamW	0.6656	0.6914	0.6656	0.6481
DeeperCNN	RMSprop	0.6323	0.5836	0.6323	0.5807
DeeperCNN	SGD	0.391	0.3451	0.391	0.2922
DenseNet121	Adadelta	0.4692	0.5824	0.4692	0.4192
DenseNet121	Adam	0.5957	0.6711	0.5957	0.5479
DenseNet121	AdamW	0.6256	0.6854	0.6256	0.5689
DenseNet121	RMSprop	0.6323	0.646	0.6323	0.6033
DenseNet121	SGD	0.6822	0.6934	0.6822	0.6826
EfficientNetB0	Adadelta	0.198	0.109	0.198	0.123
EfficientNetB0	Adam	0.7105	0.721	0.7105	0.704
EfficientNetB0	AdamW	0.6739	0.7251	0.6739	0.6488
EfficientNetB0	RMSprop	0.7255	0.7451	0.7255	0.7063
EfficientNetB0	SGD	0.6206	0.6369	0.6206	0.6211
HybridLightweight	Adadelta	0.3894	0.2856	0.3894	0.3006
HybridLightweight	Adam	0.6872	0.6906	0.6872	0.6824
HybridLightweight	AdamW	0.6805	0.6845	0.6805	0.6744
HybridLightweight	RMSprop	0.6772	0.6879	0.6772	0.6787
HybridLightweight	SGD	0.6439	0.6528	0.6439	0.6367
L2RMLP	Adadelta	0.6123	0.6416	0.6123	0.5505
L2RMLP	Adam	0.6572	0.6578	0.6572	0.6423
L2RMLP	AdamW	0.6489	0.6497	0.6489	0.6351
L2RMLP	RMSprop	0.6406	0.6416	0.6406	0.6287
L2RMLP	SGD	0.6406	0.6419	0.6406	0.6093
ResNetLike	Adadelta	0.411	0.2684	0.411	0.2831
ResNetLike	Adam	0.6722	0.6777	0.6722	0.6703
ResNetLike	AdamW	0.6872	0.685	0.6872	0.6818
ResNetLike	RMSprop	0.6689	0.6748	0.6689	0.6664
ResNetLike	SGD	0.6456	0.6589	0.6456	0.6365

Data Partitioning and Imbalance Mitigation Strategy

To ensure a proper and reliable assessment of the findings, it was important to split the data into three sets: training,

validation, and test. In the given data split, images from the same patient were included in a single subset to avoid overlapping between the training and evaluation data. The dataset is split 70% for training, 15% for validation, and 15% for testing. Importantly, data augmentation was performed

Table 8 The results of a two-way ANOVA model optimizer interaction

Two-way ANOVA (model × optimizer interaction)				
	Sum_sq	df	F	PR(> F)
C(model)	1.715650	6.0	72.029876	5.563736e-28
C(optimizer)	2.385102	4.0	150.204250	1.435930e-33
C(model):C(optimizer)	0.863796	24.0	9.066409	2.281525e-13
Residual	0.277884	70.0	NaN	NaN

Table 9 The statistical analysis of model accuracy using the Kruskal–Wallis test and normality checks. $H=98.7687$, $P=0.0000$

Normality test for accuracy			
Model	W	p	Normality test
CNN_Adam	0.7531	0.0068	Not normal
CNN_SGD	0.9527	0.5811	Normal
CNN_RMSprop	0.9289	0.4844	Normal
CNN_AdamW	0.7603	0.0228	Not normal
CNN_Adadelta	0.8602	0.2683	Normal
DeeperCNN_Adam	0.9202	0.4529	Normal
DeeperCNN_SGD	0.9691	0.6627	Normal
DeeperCNN_RMSprop	0.8131	0.1461	Normal
DeeperCNN_AdamW	0.8710	0.2983	Normal
DeeperCNN_Adadelta	0.7727	0.0509	Normal
ResNetLike_Adam	0.9029	0.3948	Normal
ResNetLike_SGD	0.8888	0.3508	Normal
ResNetLike_RMSprop	0.8830	0.3332	Normal
ResNetLike_AdamW	0.9691	0.6627	Normal
ResNetLike_Adadelta	0.9967	0.8899	Normal
DenseNet121_Adam	0.9608	0.6192	Normal
DenseNet121_SGD	0.9643	0.6369	Normal
DenseNet121_RMSprop	0.9117	0.4237	Normal
DenseNet121_AdamW	0.9908	0.8168	Normal
DenseNet121_Adadelta	0.8745	0.3083	Normal
EfficientNetB0_Adam	0.9932	0.8428	Normal
EfficientNetB0_SGD	0.8322	0.1939	Normal
EfficientNetB0_RMSprop	0.9998	0.9731	Normal
EfficientNetB0_AdamW	0.7840	0.0769	Normal
EfficientNetB0_Adadelta	0.9715	0.6761	Normal
L2RMLP_Adam	0.9231	0.4633	Normal
L2RMLP_SGD	0.9868	0.7804	Normal
L2RMLP_RMSprop	0.9796	0.7262	Normal
L2RMLP_AdamW	0.9465	0.5542	Normal
L2RMLP_Adadelta	0.8959	0.3726	Normal
Hybrid Lightweight Model_Adam	0.9656	0.6437	Normal
Hybrid Lightweight Model_SGD	0.7998	0.1139	Normal
Hybrid Lightweight Model_RMSprop	0.8459	0.2297	Normal
Hybrid Lightweight Model_AdamW	0.9583	0.6072	Normal
Hybrid Lightweight Model_Adadelta	0.9643	0.6369	Normal

only after this split and involved only the training data. As for the class imbalance that can occur in this case, there is a solution that is applied at three levels: data, algorithm, and training.

Algorithm-Level: Class-Weighted Loss Function

To further address the imbalance during training, a class-weighted loss function was used:

$$L_w = -\frac{1}{N} \sum_{i=1}^N W_{y1} \log(p(y_i|x_i)) \quad (3)$$

where N is the total number of training samples, (x_i) is the input sample, (y_i) is the true label, $(p(y_i|x_i))$ is the predicted probability, and W_{y1} is the weight assigned to each class. This ensures that errors from minority classes are given more importance during training.

Proposed Hybrid Lightweight Model

The proposed hybrid lightweight model is designed to provide an effective balance between classification performance and computational efficiency for histopathological image analysis (Fig. 2). The architecture combines two

complementary lightweight feature-extraction pathways: a MobileNet-inspired pathway based on depthwise separable convolutions and an EfficientNet-inspired pathway based on MBConv blocks with squeeze-and-excitation (SE). The two pathways extract complementary representations, with the former focusing on efficient local texture features and the latter capturing richer semantic information.

Rather than directly concatenating the features from the two pathways, the model employs an Adaptive Gated Feature Fusion (AGFF) mechanism to learn the relative contribution of each feature representation. The resulting fused features are subsequently processed by a Channel-Spatial Refinement (ACSR) module, which emphasizes informative feature channels and spatial regions while reducing the

influence of less relevant information. The refined representation is then passed through global average pooling and a fully connected classification head to generate the final class prediction.

The overall design is intended to retain the computational advantages of lightweight convolutional networks while providing a more discriminative representation of histopathological patterns. The use of adaptive feature fusion and channel-spatial refinement distinguishes the proposed architecture from a straightforward combination of existing lightweight convolutional blocks and provides the basis for evaluating whether the additional mechanisms improve classification performance without substantially increasing computational cost.

Algorithm 4: Proposed MS-AMHNet Classification Algorithm

Input: image, trained_model

Output: predicted_class, class_probabilities

1. Load the trained MS-AMHNet model.
2. $F0 = \text{LightweightStem}(\text{image})$
3. $FM = \text{MobilePath}(F0)$
4. $FE = \text{EfficientPath}(F0)$
5. $G = \text{Sigmoid}(\text{MLP}(\text{GAP}(\text{Concat}(FM, FE))))$
6. $F_{\text{fused}} = G \times FM + (1 - G) \times FE$
7. $FR = \text{ACSR}(F_{\text{fused}})$
8. $h = \text{GAP}(FR)$
9. $\text{class_probabilities} = \text{Softmax}(\text{FC}(\text{Dropout}(\text{ReLU}(\text{FC}(h))))))$
10. $\text{predicted_class} = \text{argmax}(\text{class_probabilities})$
11. **return** predicted_class, class_probabilities

Performance Evaluation

To evaluate the effectiveness of the hybrid lightweight model we designed for colorectal cancer detection, we focused on several classification metrics recognized by the research community that characterize a model's performance. So we investigated the case of accuracy, precision, recall, F1-score, and ROC-AUC, as they all add something to our understanding of how the model performed:

- **Accuracy:**

$$\text{Accuracy} = \frac{\text{TruePositives} + \text{TruesNegatives}}{\text{Total Predictions}} \quad (4)$$

- **Precision (positive predictive value):**

$$\text{Precision} = \frac{\text{TruePositives}}{\text{True Positives} + \text{False Positives}} \quad (5)$$

- **Recall (sensitivity or true positive rate):**

$$\text{Recall} = \frac{\text{TruePositives}}{\text{Total positives} + \text{False Negatives}} \quad (6)$$

- **F1-score:**

$$\text{F1 Score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (7)$$

- **Area under the ROC curve (AUC):**

$$FPR = \frac{FP}{FP + TN}, TPR = \frac{TP}{TP + FN} \quad (8)$$

Computational Efficiency Evaluation

The computational efficiency of the proposed MS-AMHNet is evaluated alongside its classification performance. Since the objective is to develop a lightweight architecture, the evaluation considers not only accuracy and F1-score but also the number of trainable parameters, floating-point operations, model size, inference time, and memory consumption.

Number of trainable parameters:

$$P_{Cov} = K_h K_w C_{in} C_{out} + C_{out} \quad (9)$$

Model size:

$$S_{model} = P_{total} \times 4 \quad (10)$$

Inference time:

$$T_{avg} = \frac{1}{N} \sum_{i=1}^N T_i \quad (11)$$

Results and Discussion

This research has proposed a hybrid lightweight model trained on the preprocessed CRC-VAL-He-7 K dataset. The analysis consisted of a performance assessment of the hybrid lightweight model against several deep learning architectures, under different optimization settings. In terms of performance classification, several factors have been considered, including accuracy, precision, recall, F-Measure, and ROC-AUC, and statistical analysis has been conducted to assess the importance of the models and optimizers evaluated.

Automated Hyperparameter Configuration

The various hyperparameter settings utilized to optimize learning models were established via a grid search to identify the best combinations for effective colorectal cancer detection. The hybrid lightweight model has been trained using five of the most popular optimization algorithms: Adam, SGD, Adadelata, AdamW, and RMSprop, indicating the hybrid nature of the process. These optimizers have unique characteristics regarding gradient updates and learning rate adaptation, thereby enabling performance evaluation of the learning process. The following hyperparameters were specified: learning rate (0.001, 0.01, 0.1), batch size (16, 32, 64), dropout rate (0.3, 0.5, 0.7), and

the number of hidden units (32, 64, 128), with ReLU for hidden network layers and Softmax for the output network layer. The models were trained with all hyperparameter combinations for 50 epochs.

The comparison of different models on the CRC-VAL-He-7 K dataset is shown in Table 1. The results show that the choice of optimizer affects model performance. The Adadelata algorithm yields poor results across all models, while Adam, AdamW, and, in some cases, SGD give satisfactory and stable results. Although CNNs perform poorly, their performance improves with more complex architectures. DenseNet121 and EfficientNet-B0 receive the best results overall. It should be noted that EfficientNet-B0, when trained with RMSprop, achieves 0.98% accuracy, indicating low variability. The suggested lightweight hybrid model achieves an accuracy of 0.91% using the optimizer AdamW (Tables 2, 3, and 4).

Figures 3 and 4 clearly show that the training and validation losses decrease as the number of epochs increases. The training losses show a steep decrease in the first few epochs, followed by a gradual decrease till the convergence is reached. The same trend is observed in the validation losses, which closely track the training curve throughout the training process. In this case, if the loss curves are so similar, the model is successful in classifying the data; thus, there are no signs of overfitting either. The receiver operating characteristic (ROC) curve in Fig. 3 illustrates the trade-off between sensitivity and false-positive rate, providing an assessment of the model's classification performance. It should be noted that all classes' AUCs equal 1.00, indicating good class distinguishability. This makes it evident that the neural model can detect both positive and negative cases quite successfully, as there were no classification errors.

The average classification accuracy of the various models tested with different optimizers is shown in Fig. 1. The highest classification accuracy was achieved by EfficientNetB0, at 98.70%. The next best performer was DenseNet121, achieving 97.22% accuracy. Other models that delivered their average classification accuracy include ResNetLike (85.99%), DeeperCNN (83.67%), and Hybrid-Lightweight (84.60%). On the contrary, CustomCNN achieved only 73.65%. Lastly, L2RMLP achieved the lowest result, 51.02%. The results indicate that EfficientNetB0 and DenseNet121 performed exceptionally well at identifying relevant characteristics in histopathological images.

As demonstrated in Fig. 5, accuracy, precision, recall, and F1-score are evaluated for all models. It is noted that EfficientNetB0 and DenseNet121 emerge as victors across the four metrics, hence proving to be accurate and effective in terms of false positives and false negatives. The remaining models, including ResNetLike, DeeperCNN, and HybridLightweight, perform well on the metrics.

ResNetLike has recorded the worst results among the tested models. The four metrics for the high-performing models align closely.

The average accuracy across models is shown in Fig. 6. The average accuracies of EfficientNetB0 and DenseNet121 were the highest among all models, indicating their superior ability to generalize (Figs. 7 and 8). In between are ResNetLike, DeeperCNN, and Hybrid-Lightweight, while CustomCNN and L2RMLP had very low average accuracies. Overall, this indicates that more advanced convolutional architectures achieve better accuracy and more consistent performance (Tables 5, 6, and 7).

Statistical Analysis

To assess the significance of differences among combinations of model optimizers and to measure their performance, a detailed statistical analysis was conducted. The very first step is to establish whether the performance metrics (accuracy, precision, recall, and F1-score) are normally distributed. The Shapiro–Wilk test was used for each model optimizer combination. If the results did not indicate any deviation from normality ($p > 0.05$), parametric statistics were used. Otherwise, nonparametric statistics were applied. To check whether model performance differs across combinations of optimizers, use either a one-way ANOVA or a Kruskal–Wallis test, depending on the outcome of the normality test. These tests assess whether at least one group performed differently. If differences in model performance are found ($p < 0.05$), pairwise comparisons are conducted. If parametric statistics are used, Tukey’s HSD test is utilized. For nonparametric statistics, the Mann–Whitney U test with Bonferroni correction is used. The effect sizes were calculated using Cohen’s d to analyze the strength of the differences. The interpretation of the values is as follows: negligible (less than 0.2), small (0.2 to 0.5), medium (0.5 to 0.8), and large (greater than or equal to 0.8). Also, bootstrap resampling with 1000 iterations was performed to obtain 95% confidence intervals for the mean configuration performance, thereby providing a more accurate picture of performance variation. A two-way ANOVA was also conducted to assess how different model types and optimizers influenced performance and to examine any interaction effects. All analyses were performed in Python using the packages SciPy, Statsmodels, and scikit-posthocs, with a significance threshold of 0.05.

The findings of a two-way ANOVA are provided in Table 3, which assesses the impact of model type, optimizer, and their interaction on performance. The results demonstrate that both the model and optimizer factors are statistically significant. The model factor is significant ($F = 72.03$, $p < 0.001$), implying that the model architecture contributes to the performance variability. The optimizer factor is

also significant ($F = 150.20$, $p < 0.001$), indicating that the optimizer has an important effect on model performance. Moreover, there is a significant interaction effect between model and optimizer ($F = 9.07$, $p < 0.001$), indicating that the optimizer effect depends on the model used; no single optimizer works for all models.

The data analysis in Table 8 utilizes the Kruskal–Wallis test along with normality tests. From the Kruskal–Wallis test, there is a significant difference in model performance across models and optimizers ($H = 98.77$, $p < 0.001$), indicating that not all combinations of model and optimizer perform equally well. The normality tests suggest that most distributions are normal ($p > 0.05$), indicating that the data behave as expected, with a few CNNs with Adam and AdamW deviating from normality (Table 9). The analysis shows that the results are statistically significant across setups, with most distributions meeting the requirements for comparison.

Conclusion and Future Work

The study evaluated how different optimization strategies affect deep learning systems used to classify histopathological images for colorectal cancer. The results have shown that the interaction between the network architecture and the choice of an appropriate optimizer greatly influences model performance. It has been shown that the proposed hybrid lightweight model achieved satisfactory results in classification accuracy and efficiency, with EfficientNetB0 and DenseNet121 outperforming the other models tested. One of the merits of the research is that it enabled a systematic comparison of deep learning architectures under varying optimization conditions. The research added hyperparameter optimization via grid search, as well as other methods aimed at eliminating class imbalance, such as appropriate data augmentation and appropriate class-weighted learning methods. However, the study still has numerous shortcomings. Only a single open-access dataset was used to conduct this experiment, and the outcomes may not be representative of the method’s performance under in vivo conditions. Additionally, the research was constrained by only a few typical structures and optimization techniques. The proposed method is advisable for further examination with larger, more diverse datasets from additional sites. Future work could consider using more advanced optimization processes, automated machine learning, transformer-based approaches, and similar complex procedures to solve imbalanced class problems. Self-supervised pretraining and parameter-efficient adaptation could also be evaluated when annotated histopathology data are limited. Recent disease image research has shown that such strategies can improve fine-grained recognition under severe label scarcity [33], although their transferability to colorectal histopathology remains to be established.

Author Contribution Emmanuel Tunbosun Aderemi: conceptualization, methodology, resources, software, writing—original draft, visualization.

Temitayo Matthew Fagbola: conceptualization, supervision, data curation, visualization, formal analysis.

Fagbuagun OjoAbayomi: conceptualization, supervision, data curation, visualization, formal analysis.

Roseline Oluwaseun Ogundokun: methodology, resources, software, data curation, writing—review and editing, visualization.

Pius Adewale Owolawi: validation, investigation, resources, writing—review and editing.

All authors reviewed the results and approved the final version of the manuscript.

Funding Open access funding provided by Tshwane University of Technology.

Data Availability The data supporting this study's findings are openly available in the Kaggle Repository at <https://www.kaggle.com/datasets/debeshjhal/kvasirseg>.

Declarations

Ethics Approval Not applicable.

Conflict of Interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Crespo, J., Victorino, A. P., Araujo, K., Araujo, L. H., & Vieira, F. A. C. (2021). Colorectal cancer biomarkers and their impact on clinical practice. *Brazilian Journal of Oncology*, 17, e20210025.
2. Silva, V. R., Santos, L. D. S., Dias, R. B., Quadros, C. A., & Bezerra, D. P. (2021). Emerging agents that target signaling pathways to eradicate colorectal cancer stem cells. *Cancer Communications*, 41(12), 1275–1313.
3. Kolligs, F. (2016). Diagnosis and epidemiology of colorectal cancer. *Visceral Medicine*, 32(3), 158–164. <https://doi.org/10.1159/000446488>
4. Rawla, P., Sunkara, T., & Barsouk, A. (2019). Epidemiology of colorectal cancer: Incidence, mortality, survival, and risk factors. *World Journal of Gastroenterology*, 14(2), 84–103.
5. Shawesh, R. A., & Chen, Y. X. (2021). Enhancing histopathological colorectal cancer image classification by using a convolutional neural network. *medRxiv*. <https://doi.org/10.1101/2021.03.17.21253390>
6. Debele, T. G., Kebede, S. R., Schwenker, F., & Shewarega, Z. M. (2020). Deep learning in selected cancers' image analysis—a survey. *Journal of Imaging*, 6(11), 121. <https://doi.org/10.3390/jimaging6110121>
7. Tsai, M.-J., & Tao, Y.-H. (2021). Deep learning techniques for the classification of colorectal cancer tissue. *Electronics*, 10(1662). <https://doi.org/10.3390/electronics10141662>
8. Hossain, M. S., Karuniawati, H., Jairoun, A. A., Urbi, Z., Ooi, D. J., John, A., Lim, Y. C., Kibria, K. M. K., Mohiuddin, A. M., & Ming, L. C. (2022). Colorectal cancer: A review of carcinogenesis, global epidemiology, current challenges, risk factors, preventive and treatment strategies. *Cancers*, 14(7), 1732. <https://doi.org/10.3390/cancers14071732>
9. Huang, W., Pan, Z., Wang, X., Zhang, Y., He, J., Gee, J. C.,... Zheng, Y. (2026). Text-Image Co-Alignment for Weakly Supervised Polyp Segmentation. *IEEE Transactions on Medical Imaging*, 1. <https://doi.org/10.1109/TMI.2026.3674592>
10. Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., & Asari, V. K. (2018). The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches. *arXiv preprint arXiv:1803.01164*.
11. Adegun, A. A., Viriri, S., & Ogundokun, R. O. (2021). Deep learning approach for medical image analysis. *Computational Intelligence and Neuroscience*, 2021(1), 6215281.
12. Liu, X., Song, L., Liu, S., & Zhang, Y. (2021). A review of deep learning-based medical image segmentation methods. *Sustainability*, 13(1224).
13. Jiang, R., Yin, X., Yang, P., Cheng, L., Hu, J., Yang, J.,... Lv, H. (2024). A transformer-based weakly supervised computational pathology method for clinical-grade diagnosis and molecular marker discovery of gliomas. *Nature Machine Intelligence*, 6(8), 876–891. <https://doi.org/10.1038/s42256-024-00868-w>
14. Panic, J., Defeudis, A., Mazzetti, S., Rosati, S., Giannetto, G., Vassallo, L., ... & Giannini, V. (2020, July). A Convolutional Neural Network based system for Colorectal cancer segmentation on MRI images. In *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* (pp. 1675–1678). IEEE.
15. Taha, D., Alzu'bi, A., Abuarqoub, A., Hammoudeh, M., & Elhoseny, M. (2022). Automated colorectal polypclassification using deep neural networks with colonoscopy images. *International Journal of Fuzzy Systems*, 24(5), 2525–2537.
16. Yang, J., Wang, G., Xiao, X., Bao, M., & Tian, G. (2024). Explainable ensemble learning method for OCT detection with transfer learning. *PloS one*, 19(3), e0296175. <https://doi.org/10.1371/journal.pone.0296175>
17. Yu, G., Sun, K., Xu, C., Shi, X. H., Wu, C., Xie, T., Meng, R. Q., Meng, X. H., Wang, K. S., Xiao, H. M., & Deng, H. W. (2021). Accurate recognition of colorectal cancer with semi-supervised deep learning on pathological images. *Nature Communications*, 12, 6311. <https://doi.org/10.1038/s41467-021-26435-4>
18. Ye, Z., Zhang, Y., Liang, Y., Lang, J., Zhang, X., Zang, G.,... Yang, J. (2022). Cervical Cancer Metastasis and Recurrence Risk Prediction Based on Deep Convolutional Neural Network. *Current Bioinformatics*, 17, 164–173. <https://doi.org/10.2174/1574893616666210708143556>
19. Yang Y, Yan JW, Luo Y, Wu S, Jing JG, Wu YT, Wang ZQ, Zhuang H. An optimal imaging modality based on different predictive risk factors for evaluating local staging of rectal cancer preoperatively: a comparative study of MRI and endorectal ultrasound. *Quant Imaging Med Surg* 2025;15(11):10516–10525. <https://doi.org/10.21037/qims-24-2120>
20. Gao, Y., Li, C., Chang, W., Du, B., Ye, X., Yeo, Y. H.,... Shi, Y. (2026). Multi-modal AI for opportunistic screening, staging and progression risk stratification of steatotic liver disease. *Nature Communications*, 17(1), 1562. <https://doi.org/10.1038/s41467-026-68414-3>

21. Sun, Y., Sun, H., Ma, J., Zhang, P., & Huang, X. (2025). Multi-modal Agent AI: A Survey of Recent Advances and Future Directions. *Journal of Computer Science and Technology*, 40(4), 1046–1063. <https://doi.org/10.1007/s11390-025-4802-8>
22. Sun, X., Zhang, P., Wang, D., Cao, Y., & Liu, B. (2019). Colorectal polyp segmentation by U-Net with dilation convolution. *arXiv preprint*, [arXiv:1912.11947v1](https://arxiv.org/abs/1912.11947v1) [eess.IV].
23. Zhang, R., Zheng, Y., Mak, T. W. C., Yu, R., Wong, S. H., Lau, J. Y. W., & Poon, C. C. Y. (2017). Colorectal polyps transferring low-level CNN features from the nonmedical domain. *IEEE Journal of Biomedical and Health Informatics*, 21(1), 2017.
24. Sharma, P., Bora, K., & Balabantaray, B. K. (2020). Two stage classification with CNN for colorectal cancer detection. *Oncologie (Tech Science Press)*, 22(3)129.
25. Sujatha, K., Reddy, C., Ponmayal, B., & Cao, R. S. (2019). Analysis of CT images for the detection of colorectal cancer using hybrid artificial neural network and firefly algorithm. In *Third International Conference on Computing and Network Communication (COCONet 19)*. <https://doi.org/10.1016/j.procs.2020.04.162>
26. Sun, X., Wang, Y., Li, H., & Zhang, Y. (2020). An end-to-end deep-learning framework for colorectal polyp segmentation. *Journal of Medical Imaging*, 7(2), 123–132.
27. Bosch, M., & Bangalore, M. (2021). Deep transfer learning with structure-preserved color normalization for texture-based classification of colorectal cancer histology. *Computers in Biology and Medicine*, 132, 104309.
28. Huang, K., Lin, B., Liu, J., Liu, Y., Li, J., Tian, G., & Yang, J. (2022). Predicting colorectal cancer tumor mutational burden from histopathological images and clinical information using multi-modal deep learning. *Bioinformatics*, 38(22), 5108–5115. <https://doi.org/10.1093/bioinformatics/btac641>
29. Cheng, Z., Chen, F., Zuo, E., Gong, Z., Xiao, M., Li, C.,... Chen, C. (2025). The MLSE-SCAM architecture combines with the improved DRSN-TIC model for Raman spectroscopy small-sample data learning. *Expert Systems with Applications*, 279, 127462. <https://doi.org/10.1016/j.eswa.2025.127462>
30. Chen, X., Chen, C., Tian, X., He, L., Zuo, E., Liu, P., Xue, Y., Yang, J., Chen, C., & Lv, X. (2024). DBAN: An improved dual branch attention network combined with serum Raman spectroscopy for diagnosis of diabetic kidney disease. *Talanta*, 266(Pt 2), 125052. <https://doi.org/10.1016/j.talanta.2023.125052>
31. Liu, C., Yang, B., Zhang, L., Liu, C., & Zheng, W. (2025). Breast Ultrasound Image Detection Based on Dual-Branch Faster R-CNN. *Electronics*, 14(21), 4247. <https://doi.org/10.3390/electronics14214247>
32. Qian, Y., Su, L., Xu, M., & Tang, L. (2025). Enhanced Protein Secondary Structure Prediction Through Multi-View Multi-Feature Evolutionary Deep Fusion Method. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 9(5), 3352–3363. <https://doi.org/10.1109/TETCI.2024.3523772>
33. Liu, Z., Ye, S., Li, X., Luo, X., Pan, J., He, M.,... Feng, H. (2026). A self-supervised learning based method for fish disease diagnosis utilizing extremely limited samples. *Computers and Electronics in Agriculture*, 250, 111952. <https://doi.org/10.1016/j.compag.2026.111952>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.