Integrating convolutional neural networks and autoencoders for skin lesion diagnosis

Kateryna Bilyk*,† Olga Narushynska†, Petro Liashchynskyi† and Vasyl Teslyuk†

Lviv Polytechnic National University, 12 Bandera Street, Lviv, 79013, Ukraine

Abstract

This study explores the effectiveness of deep learning models for the automated detection and classification of skin lesions in dermoscopic images. A combination of models was tested, starting with an autoencoder for preliminary detection of mole presence. This step helps filter out irrelevant images before further processing. The subsequent stage involves comparing several models for lesion classification, with a custom ResNet-50-based classifier achieving the highest performance, with a validation F1-score of 0.886, confirming its suitability for diagnostic tasks. For segmentation, a Mask R-CNN model was employed, achieving an Intersection over Union (IoU) of 87%. This model accurately detects and segments all visible moles, regardless of their size or location, enabling the classification of each individual lesion – addressing a key limitation of traditional methods. The models were trained and evaluated using a combination of publicly available datasets and synthesized images with artificially added lesions, enhancing dataset variability and realism. The findings indicate that combining the ResNet-50 classifier with the Mask R-CNN segmentation model constitutes a robust pipeline for integration into clinical decision-support systems, providing valuable assistance for healthcare professionals in skin lesion analysis.

Keywords

deep learning, skin cancer, melanoma, classification, segmentation, autoencoders

1. Introduction

One of the pressing challenges in modern medicine is the early detection of malignant skin tumors, particularly melanoma and other types of skin cancer. According to the World Health Organization (WHO), skin cancer is among the most common types of cancer worldwide. However, there is good news: early diagnosis significantly increases the chances of successful treatment [1], making the automation of skin image analysis [9] a highly relevant task.

The application of deep learning methods such as segmentation and classification opens new opportunities for improving diagnostic accuracy. These approaches help reduce the time required for image analysis, support clinical decision-making, and ease the workload on healthcare professionals.

The relevance of this research lies in the need to develop efficient tools for analyzing and segmenting skin lesions. This is especially important under conditions of increasing pressure on medical personnel and the need for rapid decision-making. The use of image analysis methods will contribute to more accurate diagnostics, which in turn will improve the quality of healthcare services and reduce the risk of missing potentially dangerous tumors.

The goal of this work is to enhance the efficiency of skin cancer detection, which includes primary image validation, lesion classification, and segmentation. The object of this study is the process of

MoMLeT-2025: 7th International Workshop on Modern Machine Learning Technologies, June, 14, 2025, Lviv-Shatsk, Ukraine

* Corresponding author.

† These authors contributed equally.

kkaterynabilyk@gmail.com (K. Bilyk); olha.o.narushynska@lpnu.ua (O. Narushynska); petro.b.liashchynskyi@lpnu.ua (P. Liashchynskyi); vasyl.m.teslyuk@lpnu.ua (V. Teslyuk)

0009-0007-8865-6774 (K. Bilyk); 0009-0000-0628-8218 (O. Narushynska); 0000-0002-3920-6239 (P. Liashchynskyi); 0000-0002-5974-9310 (V. Teslyuk)

analyzing skin lesion photographs using deep learning methods, while the subject is the algorithms, models, and tools for their effective monitoring and classification.

To achieve this goal, the following key objectives have been defined:

- To conduct an analysis of current research and scientific publications in the field of automated skin lesion diagnostics, particularly in the context of segmentation and classification methods.
- To investigate the effectiveness of various image processing algorithms for detecting and analyzing skin lesions.
- To justify, select, and develop neural network models for building an automated skin image analysis system, including data preprocessing steps and result evaluation methods.

This study is aimed at creating an effective tool for medical professionals in the process of diagnosing skin cancer. The proposed methods and models have the potential to improve the accuracy of diagnostic decisions and support early detection of dangerous pathologies, which may significantly impact patient survival rates.

2. Literature review

In the article *"Skin Cancer Detection Using Deep Learning – A Review"* by Naqvi et al. (2023) [9], five key networks are identified as the foundation for modern skin cancer detection systems: AlexNet (Krizhevsky et al., 2012) [13], VGG (Simonyan & Zisserman, 2015) [17], ResNet (He et al., 2016) [12], DenseNet (Huang et al., 2017) [18], and MobileNet (Howard et al., 2017) [19]. However, it is worth noting that these architectures are relatively outdated, as they were proposed between 2012 and 2017. Nowadays, an increasing number of modern models are emerging, particularly transformer-based and hybrid approaches, designed for enhanced performance in medical computer vision tasks. Some of them, especially commercial solutions, are not accessible for academic research, complicating their implementation. Therefore, adapting state-of-the-art architectures to the task of skin lesion detection remains an important area of scientific investigation.

One innovative direction involves the use of Spiking Neural Networks (SNNs) for mole image classification. These networks mimic the way information is transmitted in the human brain, where data is passed through short pulses rather than continuous signals, as in traditional networks [20]. A neuron in an SNN "decides" to send a spike when the accumulated information reaches a certain threshold. SNNs aim to merge neurobiology and machine learning by employing biologically realistic neuron models for computation [20].

Gilani et al. [21] demonstrated that SNNs, trained with fewer parameters, can outperform traditional CNNs in terms of F1-score and overall accuracy, while consuming significantly less energy. However, specificity and precision remain lower compared to VGG-13, and the hardware implementation of SNNs requires additional modules to process spiking events [9].

Abdar et al. [22] proposed a hybrid neural network model with uncertainty quantification, which is crucial for understanding the reliability of deep neural network predictions. Traditional neural networks do not provide information about the confidence of their decisions, which can be critical for medical applications. To address this issue, various uncertainty estimation methods have been developed, such as Monte Carlo Dropout (randomly deactivating neurons to generate different predictions for the same input), Ensemble MC Dropout (creating several models with different parameters, each generating predictions for the same input combined with Monte Carlo Dropout), and Deep Ensemble (training multiple independently initialized models on different data subsets) [22]. The proposed method achieved 88.95% accuracy and an F1-score of 0.909 on the ISIC 2019 dataset, indicating high potential [9].

Lu and Firoozeh Abolhasani Zadeh [23] proposed a modified version of XceptionNet using the Swish activation function (which combines properties of the linear function and ReLU). Xception, first introduced by François Chollet (2017) [24], is based on depthwise separable convolutions – a

two-step convolutional operation that first performs spatial filtering on each channel individually and then combines the extracted features using a 1×1 convolution. This design significantly reduces the number of parameters and computational cost while maintaining a high capacity for capturing complex spatial features.

Thanks to the integration of the Swish function, the improved XceptionNet trained on HAM10000 data demonstrated excellent performance: classification accuracy reached 100%, and the F1-score was 0.953. Additionally, there was a notable improvement in metrics compared to other convolutional networks, confirming the effectiveness of the proposed approach for mole detection tasks.

Khan et al. [25] presented a fully automated CNN-based approach that combines preprocessing, segmentation, and classification stages for skin lesion analysis. In the preprocessing phase, the Local Color-Controlled Histogram Intensity Values (LCcHIV) method was used to enhance contrast and normalize local skin region lighting. The enhanced images were then used as input for the segmentation network. Segmentation was performed using a novel Deep Saliency Segmentation method, which generates a heatmap and then converts it into a binary mask via a thresholding function. After that, deep features were extracted using pre-trained ResNet101 and DenseNet201 models, and classification was performed using a Kernel Extreme Learning Machine. While the model demonstrated high classification accuracy on the HAM10000 dataset, its segmentation effectiveness was evaluated on a small set of 200 images only, indicating the need for further validation on larger datasets [9].

All of the above-mentioned methods show high performance and significant potential for application in skin lesion detection, particularly due to innovative approaches such as spiking neural networks, hybrid models with uncertainty estimation, or enhanced architectures like XceptionNet with Swish activation. However, implementing such solutions requires a deep understanding of the corresponding methods as well as access to open-source code or specific hardware. Given this, the present study focuses on verified and more accessible architectures, emphasizing their adaptation and practical application to the task of mole detection. This approach helps maintain a balance between implementation complexity and the quality of the results obtained.

2.1. Validation model

In the task of mole detection in images, it is essential to synthesize a model capable of correctly and reliably identifying their presence despite high variability in skin appearance and the presence of noise. One-Class Classification (OCC) proves to be appropriate in this context, as it enables the model to focus on learning only the positive class and detecting deviations from it [3].

One of the most effective tools for implementing OCC is the autoencoder – a type of unsupervised neural network capable of generating a compressed representation of input data [7]. Literature [3,7] highlights that autoencoders are highly effective for anomaly detection tasks due to their ability to reconstruct only those data that share common features with the training set. A significant discrepancy between the original and reconstructed image indicates atypical input data, allowing the detection of anomalies – such as the absence of a mole or its unusual appearance.

In particular, in work [7], an autoencoder with fully connected layers was implemented and tested on the MNIST dataset. Despite the small image size, the model showed high effectiveness (AUC = 0.960 ± 0.002), indicating the potential of autoencoders in one-class classification tasks. However, for medical images characterized by more complex structures, it is advisable to apply convolutional architectures and test them on real medical data with higher resolution.

Thus, autoencoders are a justified choice for implementing OCC in the task of mole detection, as they allow for effective modeling of the "normal" skin structure and detecting deviations from it.

2.2. Segmentation model

One of the key challenges when using neural networks for skin cancer diagnosis is the presence of artifacts in dermatoscopic images, such as hair, shadows, marker lines, bubbles, or rulers, which can reduce classification quality [9]. To improve results, segmentation is often used as a

preprocessing step, enabling the separation of relevant objects (e.g., moles) from the background and extraneous elements. Given that several moles may be present in a single image, it is appropriate to apply instance segmentation, which allows identifying each individual object within the same category [10].

One of the most common architectures for segmentation is Mask R-CNN – a model that combines localization, classification, and pixel-level mask generation for each detected object [11]. The model consists of several key components: a feature extractor (e.g., ResNet), a RPN (Region Proposal Network), a RoI Align (Region of Interest Align) module for precise region alignment, and separate branches for classification, bounding box regression, and binary mask generation (see Figure 1). This combination ensures high segmentation accuracy even in cases of complex mole morphology.

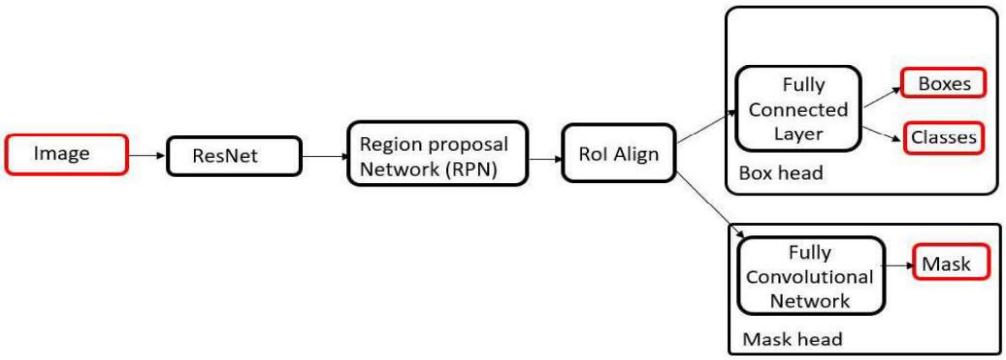

Figure 1: Mask R-CNN Schema [11]

Modern versions of the YOLO model, particularly YOLOv8, combine high processing speed with accurate object segmentation [30]. YOLOv8 is based on an improved CNN architecture that effectively extracts image features while maintaining a balance between speed and accuracy. It supports instance segmentation by generating precise masks for each detected object, making it suitable for real-time applications. To train the model on a custom dataset, the COCO format must be used, which includes object coordinates, polygon contours, and class information, providing flexibility in representing complex annotations.

2.3. Classification model

In the context of automated image classification tasks for detecting malignant skin lesions, the use of deep convolutional neural networks (CNNs) is highly relevant. The complexity of this task is due to the high visual similarity between different types of pathologies, the variability in mole appearance, the presence of artifacts, and inconsistent lighting. Therefore, numerous studies focus on comparing different architectures to determine the most effective approach.

One of the first breakthrough architectures in the field of computer vision was AlexNet [13]. It introduced the use of deeper networks, ReLU activation functions, Dropout, and efficient max pooling techniques. This approach not only achieved high classification accuracy on ILSVRC-2012 but also initiated the deep learning era in medical image analysis.

This was further developed with the appearance of ResNet [12], which introduced residual connections (skip connections) – an effective solution to the vanishing gradient problem when training deep models. Using bottleneck blocks and Batch Normalization, the model enables stable training even with considerable network depth, which is especially important for analyzing complex dermatological images.

The issue of limited computational resources prompted the development of more compact models, such as SqueezeNet [14], which achieves results comparable to AlexNet while having 50 times fewer parameters. Thanks to its unique Fire modules combining 1×1 and 3×3 convolutions, the model

efficiently extracts features while maintaining low complexity, making it suitable for mobile applications.

Another direction of optimization involves multi-scale feature processing. In this context, the Inception network model [15] stands out due to the use of modules that simultaneously analyze information using convolutions of different sizes (1×1, 3×3, 5×5) and pooling. The architecture is optimized by factorizing large filters and using auxiliary classifiers, improving the training quality of deeper layers.

Another modern approach is implemented in EfficientNet [16], which proposes scaling the model in three dimensions simultaneously-depth, width, and image resolution (compound scaling). This strategy allows achieving a better balance between performance and accuracy. Depending on available computational resources, one can choose from multiple variants (from B0 to B7), which provides additional flexibility.

Among the classic models, it is also worth mentioning the VGG architecture, which, thanks to its simple but deep structure (sequential 3×3 convolutions with ReLU and max pooling), demonstrates stable accuracy across many tasks. Its main drawbacks remain the large number of parameters (~138 million) and high computational load, which limit its applicability in real-world medical settings [17].

Thus, the literature presents a wide range of architectures, each with its advantages depending on the requirements for accuracy, speed, and available resources. Comparing these models in the context of skin cancer diagnosis enables a well-grounded selection of the most relevant solution.

3. Materials and methods

3.1. Input data and sources

For this study, we used the publicly available HAM10000 dataset (Human Against Machine with 10,000 training images) [2, 5]. This dataset contains 10,000 images of moles, most likely captured using a dermatoscope. All images have a resolution of 600×450 pixels and are stored in three-channel RGB format.

The dataset includes images from the following seven classes:

- Actinic keratoses and Bowen's disease are non-invasive skin lesions that can progress into squamous cell carcinoma and are often caused by UV exposure (AKIEC – Actinic Keratoses / Bowen's Disease, see Figure 2) [5].

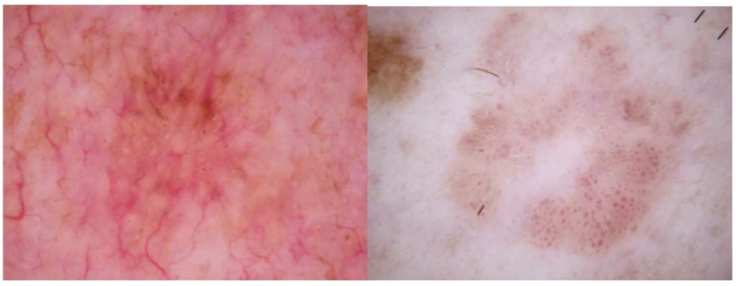

Figure 2: Images of Actinic Keratoses

- Basal cell carcinoma is a common form of skin cancer that rarely metastasizes but can grow destructively if left untreated (BCC – Basal Cell Carcinoma, see Figure 3) [5].

Figure 3: Images of Basal Cell Carcinoma

- Benign keratosis includes seborrheic keratoses, lentigines, and lichen planus-like keratoses – pigmented lesions that may mimic melanoma (BKL – Benign Keratosis, see Figure 4) [5].

Figure 4: Images of Benign Keratosis

- Dermatofibroma is a benign skin lesion that often features a central white area and may develop due to minor injuries (DF – Dermatofibroma, see Figure 5) [5].

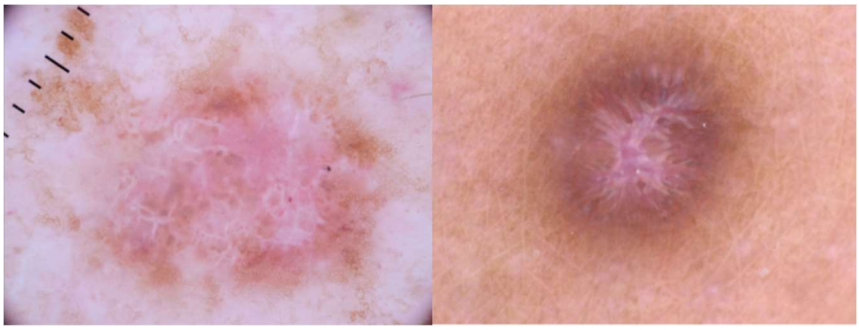

Figure 5: Images of Dermatofibroma

- Melanocytic nevi are benign tumors of melanocytes that typically exhibit symmetric structures and homogeneous coloring (NV – Melanocytic Nevi, see Figure 6) [5].

Figure 6: Images of Melanocytic Nevi

- Melanoma is a malignant tumor that can be effectively treated through surgical excision if diagnosed early (MEL – Melanoma, see Figure 7) [5].

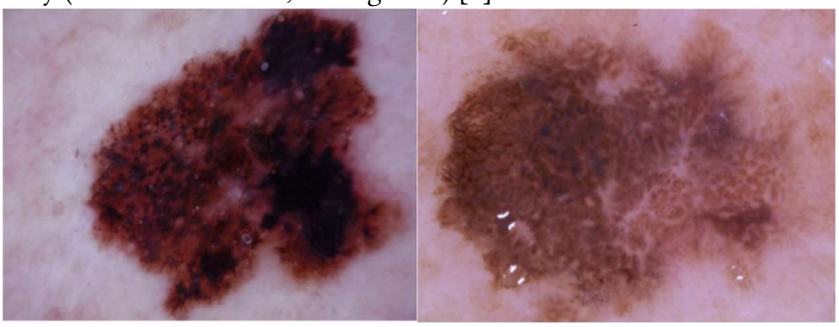

Figure 7: Images of Melanoma

Vascular lesions, such as angiomas or hematomas, usually appear as red or purple spots with well-defined borders (VASC – Vascular Lesions, see Figure 8) [5].

Figure 8: Images of Vascular Lesions

One of the key advantages of using the HAM10000 dataset is the availability of segmentation masks. This enables not only image classification but also the improvement of segmentation algorithms.

3.2. Balancing and preprocessing of input data

Due to a significant class imbalance in the input dataset for classification (e.g., the NV class contains 6705 images, while the DF class includes only 115), a balancing strategy was applied [26]. The oversampling method SMOTE [8] was used to increase the number of samples for minority classes. Additionally, to ensure an even distribution of images across data subsets, stratified splitting was employed.

The balancing was performed using the following strategy:

- A subset was selected for each class, not exceeding the maximum number of images per class.
- Stratified splitting ensured that all subsets contained proportionally the same number of images from each class.
- The final data split consisted of a training set (~83%), validation set (~10%), and test set (~7%).

To increase the model's robustness to variations in lighting, scale, and image positioning, moderate data augmentation was applied. All images were subjected to light blurring with a probability of 30%, minor shifts, rotations up to 5°, and scaling. Additionally, brightness and contrast

adjustments were made, and pixel values were normalized to the [0,1] range. This approach helped improve the model's generalization ability while preserving key features (see Figure 9).

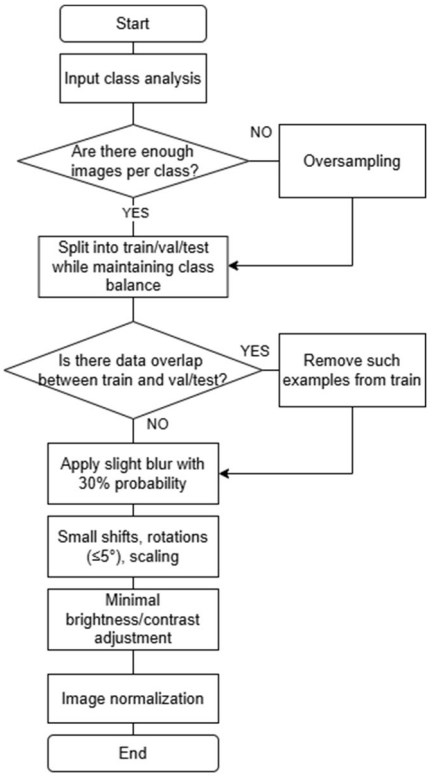

Figure 9: Data preprocessing pipeline

The segmentation model was trained on data containing images with multiple moles and their corresponding segmentation masks (see Figure 10). This approach allows for proper processing of scenes with multiple lesions. To expand the dataset, 1000 synthetic images were generated by randomly placing 2–4 mole fragments from the original HAM10000 dataset onto an artificial background. Object scaling was applied, and overlaps were avoided. A corresponding segmentation mask was automatically created for each image.

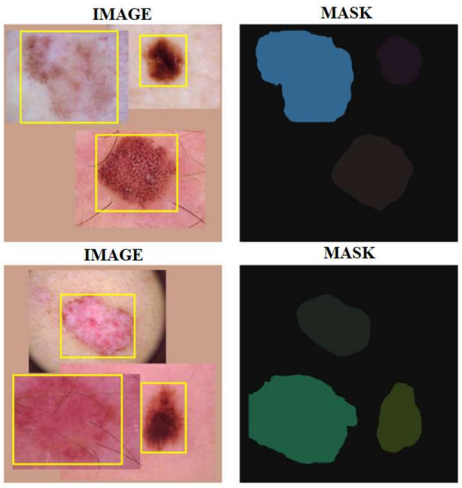

Figure 10: Synthesized images for model training

Additionally, a validation neural network model based on an autoencoder was developed to detect the presence of a mole in the image. The same augmentation strategy used for the classification

model was applied here, ensuring data consistency and improving the image processing effectiveness.

3.3. Developed models and their characteristics

As part of this study, neural network models were developed based on existing architectures, each with specific structural features and advantages for image classification tasks. For example, ResNet50, due to its residual connections, effectively minimizes the vanishing gradient problem and achieves high accuracy. EfficientNet provides a balance between accuracy and the number of parameters through optimal scaling of depth, width, and resolution. VGG, despite its simple architecture, performs well in image processing tasks. For comparison, AlexNet was used – one of the first successful CNN architectures that laid the foundation for deep learning development. SqueezeNet, due to its compactness, processes images quickly without loss of accuracy. Inception, by using filters of various sizes, efficiently extracts features, which is particularly useful for analyzing complex structures of skin lesions.

Image validation was performed using an autoencoder, specially designed for this task. As a one-class classifier, it was trained on mole images, enabling it to effectively reconstruct known patterns and detect anomalous deviations related to the presence of skin lesions. The schema of the developed autoencoder is shown in Figure 11.

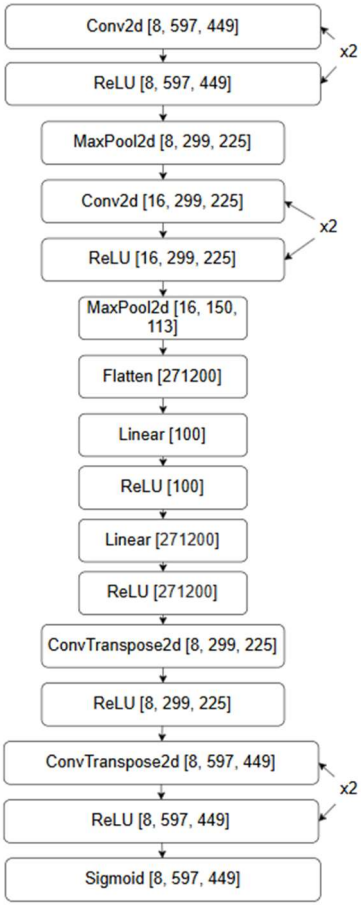

Figure 11: Schema of the developed autoencoder

For segmentation, a neural network model based on Mask R-CNN was developed, which enabled not only the identification of moles in images but also the precise delineation of their boundaries. This is critically important for further analysis and diagnosis.

3.4. Neural network model optimization

To ensure stable training of the neural network and reduce the risk of overfitting, a number of well-established techniques were applied. In particular, the use of the Adam (Adaptive Moment Estimation) optimizer enabled effective adaptation to the specifics of the data and contributed to fast and stable convergence during training. Adam is one of the most popular optimizers due to its ability to automatically adjust the learning rate for each parameter individually, thereby improving training efficiency.

To overcome overfitting, the early stopping technique was used. This involves halting the training process when the model's performance on the validation set stops improving over a certain period. This helps avoid overfitting and ensures better generalization to new, unseen data.

In addition, a dynamic learning rate reduction approach was implemented when a "plateau" was reached – i.e., when model performance metrics stopped improving. This helped further optimize training and avoid stagnation.

The use of these techniques collectively significantly improved the stability of the training process, optimized the model parameters, and enhanced its generalization capability on new data.

3.5. Software tools used

The models were implemented using the Python programming language and the PyTorch [27], scikit-learn [28] and NumPy [29] libraries. Python was used for general data processing and manipulation, PyTorch for efficient neural network construction and training, scikit-learn for performance evaluation. NumPy was utilized for efficient mathematical operations and handling large data arrays.

The models were trained on a GPU (Graphics Processing Unit) in the Kaggle environment, which significantly accelerated the training process and enabled the handling of large data volumes. The use of such an environment ensured stable and fast data processing due to access to powerful computational resources.

4. Research results and their discussion

4.1. Developed validation model

An autoencoder was selected as a one-class classifier based on the study by Isuru Jayarathne and Michael Cohen [7]. The model described in their work was adapted: instead of fully connected layers, convolutional layers were used, which allowed for more efficient processing of large-sized images. Additionally, moderate data augmentation was applied, which is described in detail in the section "Balancing and Preprocessing of Input Data."

The loss function was calculated using Mean Squared Error (MSE), which is better suited for multi-class classification of color images. This differs from the approach in the original paper [7], where Binary Cross Entropy Loss (BCELoss) was used, which is more appropriate for binary classification, particularly in the case of grayscale MNIST images.

- The Binary Cross Entropy Loss formula (see Formula 1):

$$BCE = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)], \quad (1)$$

where N is the number of samples in the dataset, y_i is the true label for the i-th sample ($y_i = 1$ for the positive class and $y_i = 0$ for the negative class), and $\hat{y}_i$ is the predicted probability that the i-th sample belongs to the positive class.

- The Mean Squared Error formula (see Formula 2):

$$MSE = \frac{1}{N}\sum_{i=1}^{N} log(x_i - \hat{x}_i)^2, \quad (2)$$

where N is the number of samples in the dataset, x_i is the pixel value of the original image, and $\hat{x}_i$ is the pixel value of the reconstructed image.

To convert the autoencoder into a fully functional one-class classification tool, the PSNR (Peak Signal-to-Noise Ratio, see Formula 3) method was selected, as it provides a more accurate assessment of similarity between the input and reconstructed images.

$$PSNR = 20 * log_{10}\left(\frac{MAX_{pixel}}{\sqrt{MSE}}\right), \quad (3)$$

where PSNR is measured in decibels (dB), MAX_{pixel} denotes the maximum possible pixel value of the image (e.g., 1.0 for normalized images or 255 for 8-bit images) and MSE is defined in Formula (2).

Preliminary analysis of other metrics-cosine similarity and MSE (Mean Squared Error) – showed that they are not sensitive enough to small but important changes in the images, which is critical in medical analysis tasks.

The classification method involves transforming both the original and reconstructed images into vectors and computing their similarity.

- If the PSNR value exceeds 25, the image is classified as correct (i.e., contains a mole).
- If the value is below the threshold, the image is classified as potentially anomalous or in need of retaking.

Additionally, synthetic data with artificially added moles was used for training. Several model configurations were tested during training of the autoencoder by changing parameters such as latent space size (latent_dim – the size of the compressed representation of the input data), learning rate, and batch size. The platform Weights & Biases [4] was used to monitor metrics and save model weights.

The table below (see Table 1) presents the training results for different model configurations; the best results are marked in green, based on a gradient from worst (red) to best (green):

Table 1
Validation model training results

Models	latent_dim	batch_size	learning_rate	epoch	loss	psnr
Autoencoder	256	32	0.0001	20	0.004	24.20
Autoencoder	256	32	0.0001	17	0.003	24.86
Autoencoder	100	32	0.001	20	0.003	25.26
Autoencoder	100	8	0.001	3	0.028	15.60
Autoencoder	100	16	0.001	6	0.014	18.80
Autoencoder	100	16	0.001	10	0.021	16.82
Autoencoder	200	16	0.0001	10	0.004	23.95
Autoencoder	200	16	0.01	10	0.011	19.67
Autoencoder	200	16	0.001	10	0.003	25.07
Autoencoder	200	15	0.0001	10	0.004	23.71
Autoencoder	200	15	0.0001	1	0.494	1.65
Autoencoder	100	8	0.002	10	0.357	3.77
Autoencoder	200	8	0.002	1	0.507	1.17
Autoencoder	64	8	0.002	10	0.519	0.96
Autoencoder	32	8	0.002	10	0.562	-0.02
Autoencoder	32	16	0.001	10	0.563	-0.06

Models	latent_dim	batch_size	learning_rate	epoch	loss	psnr
Autoencoder	32	8	0.001	10	0.645	6.03
Autoencoder	128	8	0.001	10	0.621	10.79
Autoencoder	200	8	0.001	10	0.590	26.38

- First configuration: batch_size = 32, learning rate = 0.0001, latent_dim = 256. Training lasted for 17 epochs and was automatically stopped due to the absence of further improvement in metrics. Obtained metrics: PSNR = 24.8586, loss = 0.0032. The analysis of the graphs indicated no overfitting and stable model convergence (see Figure 12).

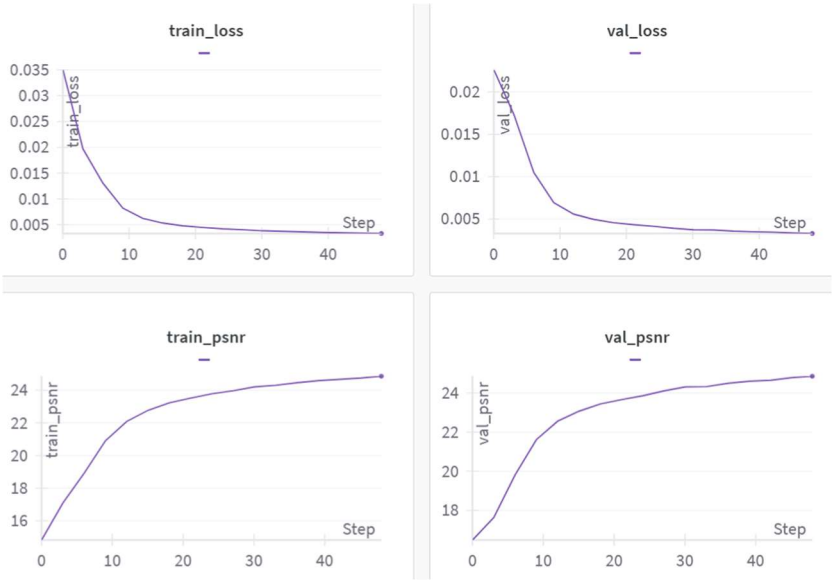

Figure 12: Metrics dynamics for the model with the first configuration: loss curves (top) and PSNR metric (bottom) on the training and validation datasets.

- Second configuration: batch_size = 32, learning rate = 0.001, latent_dim = 100. Training lasted for 20 epochs. Obtained metrics: PSNR = 25.228, loss = 0.003. The graph analysis shows no overfitting and stable model convergence (see Figure 13).

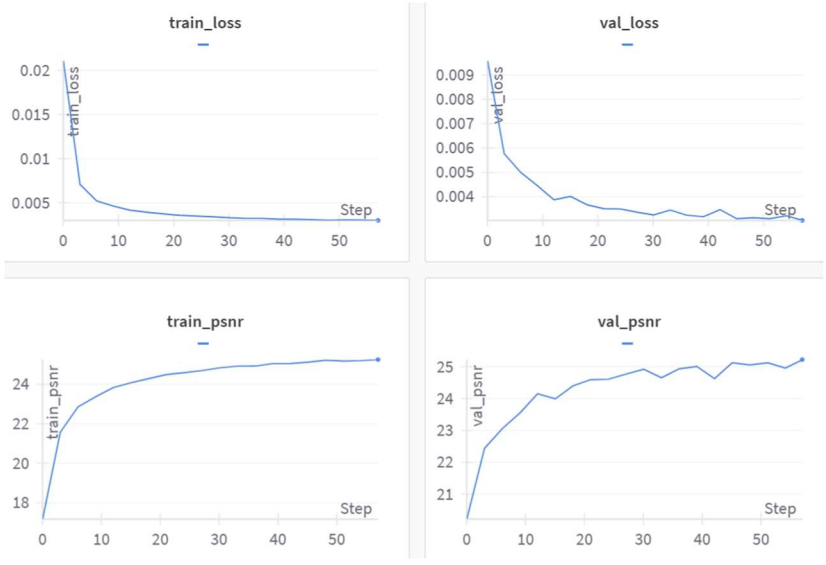

Figure 13: Metrics dynamics for the model with the second configuration: loss curves (top) and PSNR metric (bottom) on the training and validation datasets.

- Third configuration: batch_size = 32, learning rate = 0.0001, latent_dim = 200. Training ended after 10 epochs. Metrics: PSNR = 25.07, loss = 0.0031 (see Figure 14).

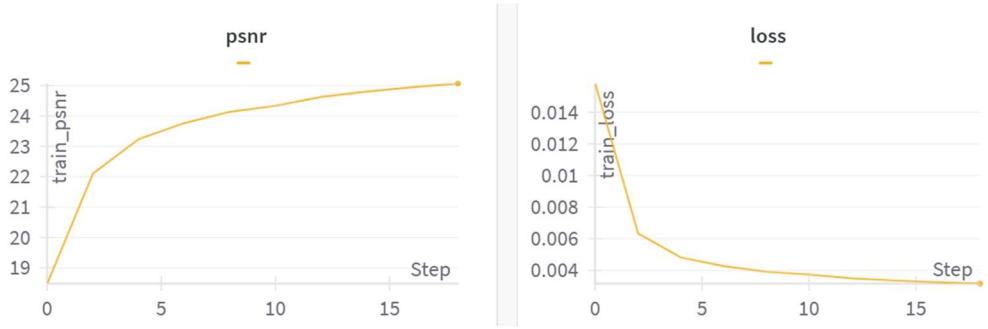

Figure 14: Metrics dynamics for the model with the third configuration: PSNR (left) and loss function (right).

Analysis of the results shows that the reconstruction quality strongly depends on the latent space size. Reducing the latent_dim to 32 significantly degrades accuracy, regardless of the learning rate. Reducing the batch size slows down model convergence. As for the learning rate, 0.001 turned out to be optimal: at 0.01, convergence was unstable, further reduction to 0.0001 did not yield significant improvement. The obtained results confirm the importance of proper hyperparameter tuning for stable and efficient autoencoder training.

To evaluate the algorithm's performance, visual examples are provided for both successfully reconstructed images and those with noticeable reconstruction errors (see Figures. 15–16).

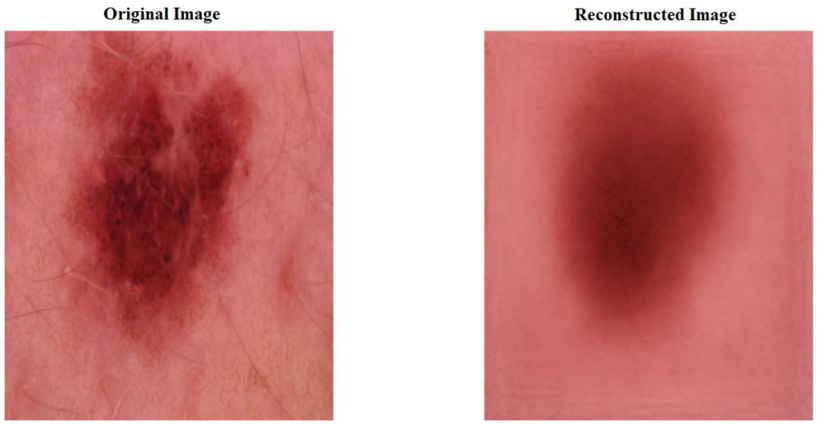

Figure 15: Example of a successfully reconstructed image

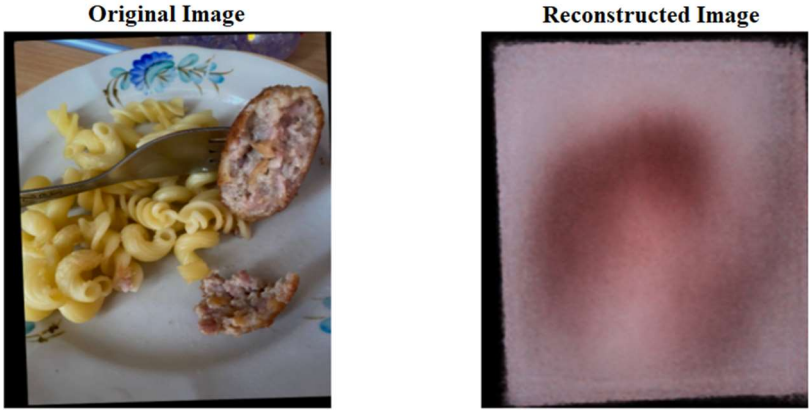

Figure 16: Example of a reconstructed image with noticeable errors

The developed model is used for the initial validation step of the service: the user uploads an image, the autoencoder analyzes it, and returns a decision on whether a new image is needed or whether the current image is of sufficient quality for further analysis.

4.2. Developed segmentation model

After passing the validation check, the image is sent to the preliminary segmentation stage, for which a neural network model based on Mask R-CNN was developed. The chosen model does not require significant preprocessing of the data. Training was conducted in fine-tuning mode: most of the pre-trained network weights were retained, except for the roi_heads module, which was adapted to the specifics of the task.

Experiments were conducted with various hyperparameter configurations, including changes in the learning rate, choice of optimizer, and number of epochs. Satisfactory results were achieved after just five epochs of training. The quality of segmentation was evaluated on both the training and validation datasets, confirming the effectiveness of the chosen approach.

An example of the resulting mole segmentation is presented (see Figure 17):

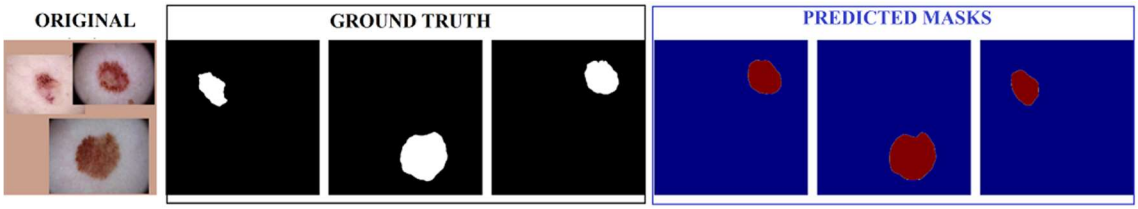

Figure 17: Example of mole segmentation: reference images on a black background, predictions on a blue background.

The model demonstrated high IoU (Intersection over Union, see Formula 4) values even after minimal modifications and only five epochs of training (see Figure 18).

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union}, \tag{4}$$

where area of overlap is the region where the predicted bounding box and the ground truth bounding box intersect, and area of union is the total area covered by both the predicted and ground truth bounding boxes combined.

However, the main issue remains the duration of training: one epoch of Mask R-CNN takes at least 40 minutes due to the large amount of training data, even on a high-performance GPU.

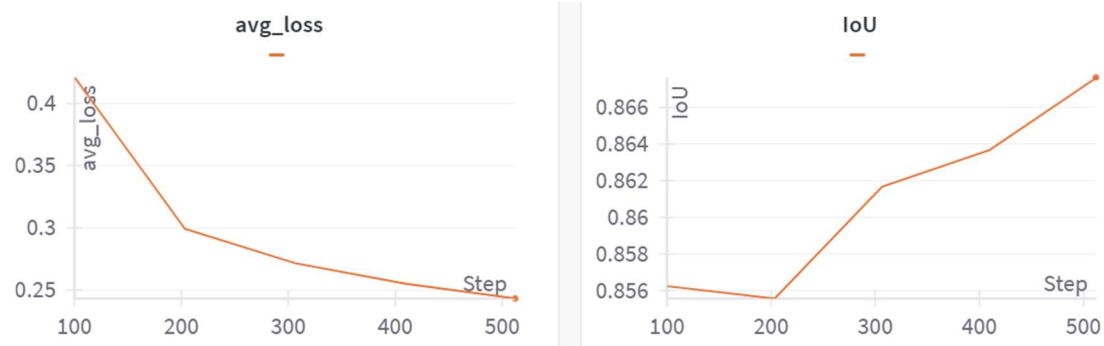

Figure 18: Metrics evolution during the training of the segmentation model: average loss (left) and IoU (right).

4.3. Developed classification models

To compare the performance of different neural network architectures, various models were trained on the same dataset. This process included testing different model variations, particularly by tuning hyperparameters such as batch size and learning rate. However, not all models showed significant improvements even after hyperparameter optimization. Some models continued to yield poor results, indicating their insufficient effectiveness for this study regardless of configuration adjustments.

Model performance was evaluated using the Accuracy (see Formula 5), Precision (Formula 6), Recall (Formula 7), and F1-score (Formula 8) metrics on both the training and validation datasets. The following formulas were used to compute these metrics:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (5)$$

$$Precision = \frac{TP}{TP + FP}, \quad (6)$$

$$Recall = \frac{TP}{TP + FN}, \quad (7)$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}, \quad (8)$$

where TP is the number of true positive predictions, TN is true negatives, FP is false positives, and FN is false negatives.

The table below (see Table 2) presents the training results of models with various configurations; the best results are highlighted in green, following a gradient from worst (red) to best (green):

Table 2
Comparison of classification model results across various metrics

Models	epoch	batch_size	learning_rate	accuracy	f1_score	loss	precision	recall	val_accuracy	val_f1_score	val_loss	val_precision	val_recall
VGG	10	16	0.0001	0.864	0.861	0.288	0.861	0.864	0.764	0.760	0.730	0.769	0.764
EfficientNet b0	29	16	0.001	0.900	0.900	0.235	0.903	0.900	0.882	0.882	0.509	0.883	0.882
VGG	12	16	0.001	0.131	0.111	1.968	0.161	0.131	0.126	0.062	1.960	0.089	0.126
EfficientNet b0	23	16	0.0001	0.724	0.724	0.632	0.744	0.724	0.742	0.743	0.654	0.759	0.742
AlexNet	33	16	0.001	0.738	0.728	0.524	0.731	0.738	0.710	0.701	0.808	0.707	0.710
SqueezeNet	23	16	0.001	0.635	0.629	0.814	0.678	0.635	0.596	0.595	0.948	0.634	0.596
InceptionV3	38	16	0.001	0.574	0.571	0.918	0.628	0.574	0.688	0.690	0.710	0.741	0.688
EfficientNet b5	6	16	0.001	0.106	0.094	1.943	0.208	0.106	0.098	0.087	1.942	0.213	0.098
Resnet50	18	16	0.001	0.956	0.956	0.098	0.956	0.956	0.886	0.886	0.565	0.888	0.886
EfficientNet b5	8	16	0.001	0.146	0.090	1.941	0.113	0.142	0.170	0.108	1.934	0.149	0.143
Resnet50	10	16	0.001	0.921	0.913	0.238	0.917	0.911	0.714	0.729	1.013	0.746	0.723
EfficientNet b5	4	16	0.0001	0.147	0.105	1.943	0.182	0.136	0.143	0.060	1.939	0.197	0.122
EfficientNet b5	3	16	0.001	0.144	0.106	1.946	0.276	0.147	0.164	0.100	1.943	0.274	0.142
EfficientNet b5	16	8	0.001	0.126	0.091	1.942	0.304	0.154	0.155	0.105	1.941	0.128	0.169
AlexNet	22	8	0.001	0.719	0.715	0.743	0.724	0.719	0.536	0.530	1.173	0.547	0.536
AlexNet	17	8	0.001	0.636	0.633	0.916	0.671	0.636	0.506	0.494	1.132	0.546	0.506
SqueezeNet	16	8	0.001	0.738	0.734	0.722	0.751	0.738	0.574	0.564	1.054	0.596	0.574
InceptionV3	27	8	0.001		0.500	1.188	0.543	0.499		0.524	1.119	0.585	0.546

EfficientNet b5	1	8	0.001	0.114	1.948	0.289	0.165	0.133	1.900	0.690	0.213
Resnet50	16	8	0.001	0.876	0.311	0.877	0.875	0.720	0.907	0.732	0.718
Resnet50	10	8	0.001	0.763	0.551	0.773	0.759	0.656	1.020	0.714	0.645

According to the research findings, the best results were demonstrated by the following models (see Table 3):

Table 3
Best model results based on the study findings

Models	val_accuracy	val_f1_score	val_loss	val_precision	val_recall
ResNet-50 (lr=0.001)	0.886	0.886	0.565	0.888	0.886
EfficientNet-B0 (lr=0.001)	0.882	0.882	0.509	0.883	0.882
VGG (lr=0.0001)	0.764	0.76	0.73	0.769	0.764
ResNet-50 (lr=0.0001)	0.714	0.729	1.013	0.746	0.723

To assess the training effectiveness of the best-performing models, graphs showing changes in the training loss function were constructed (see Figure 19).

Figure 19: Comparative visualization of training loss functions for the best-performing models ResNet, VGG, EfficientNet

Among the tested architectures, ResNet-50 achieved the best classification performance, significantly outperforming others in key metrics. The VGG and EfficientNet-B0 models also showed strong results, although they slightly lagged behind ResNet-50.

On the other hand, architectures such as Inception v3, SqueezeNet, AlexNet, and EfficientNet-B5 were less effective for the given task. In particular:

- Inception v3 achieved a validation F1-score of 0.690 and accuracy of 0.688;
- SqueezeNet – F1-score of 0.596 and accuracy of 0.595;
- AlexNet showed an F1-score of 0.701 and accuracy of 0.710;
- EfficientNet-B5 significantly underperformed compared to other models, with an F1-score of just 0.105 and accuracy of 0.155.

This may be due to their architectural characteristics, insufficient adaptation to the specific task, or suboptimal hyperparameter settings. Figure 20 shows the loss dynamics for all models and training runs for comparison:

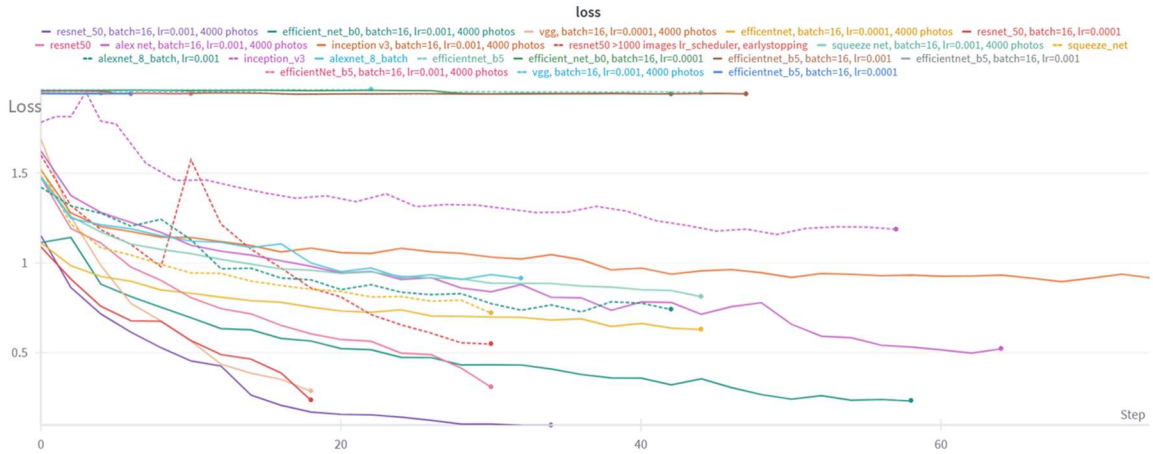

Figure 20: Loss curves for all runs of the classification models

5. Conclusions

In the course of this study, various scientific publications were reviewed regarding the application of neural networks in computer vision, particularly in dermatology. Several deep learning models were tested for the classification and segmentation of skin images, and the most effective approaches were identified based on empirical evaluation.

For the skin lesion classification task, the custom model based on ResNet-50 achieved the best results, with a validation accuracy of 88.6%, F1-score of 0.886, outperforming the EfficientNet-B0, EfficientNet-B5 SqueezeNet, AlexNet, Inception v3 and VGG-based models tested under similar conditions. This performance indicates a strong generalization capability even with a relatively limited number of training epochs.

For segmentation, the Mask R-CNN-based model demonstrated high reliability, achieving an Intersection over Union (IoU) of 87%. It effectively detects all visible moles in an image, regardless of size or location. This model is particularly valuable for distinguishing individual lesions from the background, enabling the precise extraction of each mole for subsequent classification. Although computationally intensive, its high precision in localized analysis supports its use for detailed lesion identification and individual mole classification.

An autoencoder model was also employed for preliminary mole presence detection, effectively filtering out irrelevant images and contributing to overall pipeline efficiency. All the tested models are planned to be integrated into a unified AI-based module for automated skin lesion analysis. The system will include stages of initial image validation, mole segmentation, and lesion classification.

Future work will focus on developing a clinical prototype capable of delivering actionable recommendations. This integrated approach has the potential to significantly enhance early detection of skin conditions, streamline dermatological workflows, and improve patient outcomes.

Declaration on Generative AI

During the preparation of this work, the authors used GPT-4 to assist with grammar and spelling check, paraphrasing and rewording, and improving the writing style. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] World Health Organization (WHO). Skin cancer factsheet. https://www.iarc.who.int/cancer-type/skin-cancer/

[2] Kaggle. *Skin Cancer : HAM10000* [dataset]. https://www.kaggle.com/datasets/surajghuwalewala/ham1000-segmentation-and-classification

[3] Perera, P., Oza, P., & Patel, V.M. (2021). *One-Class Classification: A Survey*. arXiv preprint arXiv:2101.03064. https://arxiv.org/abs/2101.03064

[4] Weights & Biases. Documentation. https://docs.wandb.ai/

[5] Tschandl, P., Rosendahl, C., & Kittler, H. (2018). The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific Data*, 5, 180161. https://doi.org/10.1038/sdata.2018.161

[6] Siddique, N., et al. (2023). Comparison of VGG-16, VGG-19, and ResNet-101 CNN Models for Suspicious Activity Detection. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 8(1). https://doi.org/10.32628/CSEIT2390124

[7] Jayarathne, I., & Cohen, M. (2019). *Autoencoder-based One-class Classification*. 326th SICE Tohoku Branch Workshop. https://www.researchgate.net/publication/337929447_Autoencoder-based_One-class_Classification.

[8] Chawla, N.V., Bowyer, K.W., Hall, L.O., & Kegelmeyer, W.P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16, 321–357. https://doi.org/10.1613/jair.953

[9] Naqvi, M., Gilani, S.Q., Syed, T., Marques, O., & Kim, H.-C. (2023). Skin Cancer Detection Using Deep Learning – A Review. *Diagnostics*, 13(11), 1911. https://doi.org/10.3390/diagnostics13111911

[10] Hafiz, A.M., & Bhat, G.M. (2020). A Survey on Instance Segmentation: State of the Art. arXiv preprint arXiv:2007.00047. https://arxiv.org/abs/2007.00047

[11] Ramesh, C., Sreya, & Vinod Kumar, V. (2020). A Review on Instance Segmentation Using Mask R-CNN. *Proceedings of the International Conference on Systems, Energy & Environment (ICSEE) 2021*. SSRN. https://ssrn.com/abstract=3794272

[12] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. https://doi.org/10.1109/CVPR.2016.90

[13] Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25. https://doi.org/10.1145/3065386

[14] Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50× fewer parameters and <0.5 MB model size. arXiv preprint arXiv:1602.07360. https://arxiv.org/abs/1602.07360

[15] Szegedy, C., et al. (2015). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9. https://doi.org/10.1109/CVPR.2015.7298594

[16] Tan, M., & Le, Q. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv:1905.11946. https://arxiv.org/abs/1905.11946

[17] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556. https://arxiv.org/abs/1409.1556

[18] Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K.Q. (2017). Densely Connected Convolutional Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2261–2269. https://doi.org/10.1109/CVPR.2017.243

[19] Howard, A.G., et al. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861. https://arxiv.org/abs/1704.04861

[20] Yamazaki, K., Vo, K., Bulsara, D., & Le, N. (2022). Spiking Neural Networks and Their Applications: A Review. *Brain Sciences*, 12(7), 863. https://doi.org/10.3390/brainsci12070863
[21] Gilani, S. Q., Syed, T., Umair, M., & Marques, O. (2023). Skin Cancer Classification Using Deep Spiking Neural Network. *Journal of Digital Imaging*, 36(3), 1137–1147. https://doi.org/10.1007/s10278-023-00776-2
[22] Abdar, M., et al. (2021). Uncertainty quantification in skin cancer classification using three-way decision-based Bayesian deep learning. *Computers in Biology and Medicine*, 135, 104418. https://doi.org/10.1016/j.compbiomed.2021.104418
[23] Lu, X., & Abolhasani Zadeh, Y. F. (2022). Deep learning-based classification for melanoma detection using XceptionNet. *Journal of Healthcare Engineering*, 2022, 2196096. https://doi.org/10.1155/2022/2196096
[24] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. arXiv preprint arXiv:1610.02357. https://doi.org/10.48550/arXiv.1610.02357
[25] Khan, M. A., Sharif, M., Akram, T., Damaševičius, R., & Maskeliūnas, R. (2021). Skin lesion segmentation and multiclass classification using deep learning features and improved moth flame optimization. *Diagnostics*, 11(5), 811. https://doi.org/10.3390/diagnostics11050811
[26] Buda, M., Maki, A., & Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks, 106*, 249–259. https://doi.org/10.1016/j.neunet.2018.07.011
[27] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, *32*, 8024–8035. https://arxiv.org/abs/1912.01703
[28] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf
[29] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2
[30] Hermens, F. (2024). Automatic object detection for behavioural research using YOLOv8. *Behavior Research Methods*, 56, 7307–7330. https://doi.org/10.3758/s13428-024-02420-5