

Module for the analysis of digital skin images aimed at early diagnosis of dermatological conditions based on deep learning methods*

Vasyl Teslyuk^{1,†}, Olha Narushynska^{1,†}, Maxym Arzubov^{1,†}, Tetiana Martyniuk^{2,†} and Tetiana Trachuk^{1,*,†}

¹ Lviv Polytechnic National University, 12 Stepan Bandera Street, Lviv, 79013, Ukraine

² Astra-vision (Inria + Valeo.ai) Paris Centre, 48, rue Barrault, CS 61534, 75647 PARIS CEDEX, France

Abstract

This paper presents an information technology for analyzing digital skin images for the early diagnosis of dermatological conditions, particularly dermatitis. The main problem addressed is the difficulty in accurate classification of skin conditions due to visual similarity of symptoms, limited reference datasets, and overfitting risks in neural networks. To address this, we developed a modular system consisting of a custom convolutional neural network for skin/non-skin validation—featuring stacked convolutional blocks, global pooling, and dual fully connected branches—and several classification models (ResNet, EfficientNet, MobileNet, and a custom convolutional neural network (CNN)) to identify specific pathologies.

The proposed system is implemented via a set of RESTful APIs: image validation, disease classification, and model retraining based on user-submitted new images with automatic replacement of the main model if improved metrics are observed. Standard evaluation metrics (accuracy, precision, recall, F1-score) were used to compare models. The best performance was demonstrated by EfficientNet with validation preprocessing, while the custom model showed high flexibility in adaptive retraining.

This system can be deployed in healthcare institutions, mobile diagnostic apps, and telemedicine platforms, offering rapid preliminary skin condition assessments. Thanks to its retraining capability, the system can continuously improve its accuracy and relevance in real-world environments.

Keywords

deep learning, dermatology, image classification, ResNet, EfficientNet, MobileNet, skin segmentation, API, retraining

1. Introduction

In today's world, the importance of early diagnosis of skin diseases is growing due to the increasing number of allergic, inflammatory, and chronic dermatological conditions. According to the World Health Organization (WHO), skin diseases affect over 1.8 billion people globally at any given time, with dermatitis being one of the most common conditions [1].

Professional diagnostics requires not only a doctor's experience, but also the use of auxiliary information technologies, including artificial intelligence methods. In particular, deep learning – a subfield of machine learning that uses multilayered neural networks for feature extraction and classification – has shown promising results in medical imaging [2].

In the context of the modern development of telemedicine – the use of digital communication technologies for remote medical care – automated solutions for primary diagnosis are of particular importance, especially in underserved regions [3].

ICyberPhyS'25: 2nd International Workshop on Intelligent & CyberPhysical Systems, July 04, 2025, Khmelnytskyi, Ukraine

* Corresponding author.

† These authors contributed equally.

✉ vasyi.m.teslyuk@lpnu.ua (V. Teslyuk); olha.o.narushynska@lpnu.ua (O. Narushynska); maksym.v.arzubov@lpnu.ua (M. Arzubov); tetyanka.martyniuk@gmail.com (T. Martyniuk); tetiana.trachuk.kn.2021@lpnu.ua (T. Trachuk)

ORCID: 0000-0002-5974-9310 (V. Teslyuk); 0009-0000-0628-8218 (O. Narushynska); 0009-0008-9978-7072 (M. Arzubov); 0000-0001-5254-0955 (T. Martyniuk); 0009-0002-3879-3897 (T. Trachuk)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Furthermore, to address the limitations of static models trained on fixed datasets, we apply adaptive retraining – a technique in which the model is dynamically updated using new data collected during its deployment [4]. This paper considers the problem of synthesizing information technology that will allow the analysis of patients' skin images with the subsequent classification of dermatological conditions.

The aim is to develop and design a digital image analysis module for automated diagnosis of dermatitis. The object of research is the processes of developing intelligent systems for medical image diagnostics. The subject of research is methods of augmentation and classification of digital skin images using deep learning models.

The value of the work lies in the fact that the developed system can be used in medical institutions for preliminary diagnosis, as well as in conditions of limited access to dermatologists – in rural areas, within mobile diagnostic platforms, etc.

2. Materials and methods of the study

To build an information technology for the early diagnosis of skin pathologies, a full-fledged software module consisting of several interdependent components was developed. The main function of this module is preliminary image validation (detection of the presence of skin in the photo), the use of augmentation to increase the variability of the training set, image classification based on diagnostic classes (types of dermatitis), and model updating based on the feedback received.

2.1. Input data

The images for training and testing the models were collected from the open dermatology dataset “Skin Diseases Image Datasets” [5], which contains 14,317 samples of skin pathologies, divided into the following classes:

- Eczema - 1677 images
- Atopic Dermatitis - approximately 1257 images
- Basal Cell Carcinoma (BCC) - 3313 images
- Benign Keratosis-like Lesions (BKL) - 2065 images
- Psoriasis, Lichen Planus and related diseases - approximately 2055 images
- Seborrheic Keratoses and other Benign Tumors - approximately 1847 images
- Warts, Molluscum and other Viral Infections - 2103 images

12,169 images were used for model training, while the remaining 2,148 samples were set aside for independent testing. Unlike standardized datasets such as HAM10000 [6] or ISBI 2017 Challenge [7], our dataset includes a broader variety of common skin diseases in natural environments.

In order to unify the input data, the images were pre-processed: all examples were resized to a standard size of 224×224 pixels for use with pre-trained models (ResNet, EfficientNet, MobileNet), and to 64×64 pixels for the custom CNN architecture. The choice of lower resolution (64×64) for the custom model is motivated by research findings indicating that reduced image size can be effective for lightweight models without significant accuracy loss, especially when using convolutional architectures focused on global features [8]. In addition, the images were normalized by channels according to PyTorch requirements and manually checked for the presence of objects of interest - skin lesions - necessary for correct model training.

Below on Figs. 1-3 are examples of input images used for validation and training of neural networks. These images show a variety of skin lesions corresponding to the classes that the model should be able to recognize. The list of images provided for training and testing includes various types of skin pathologies.

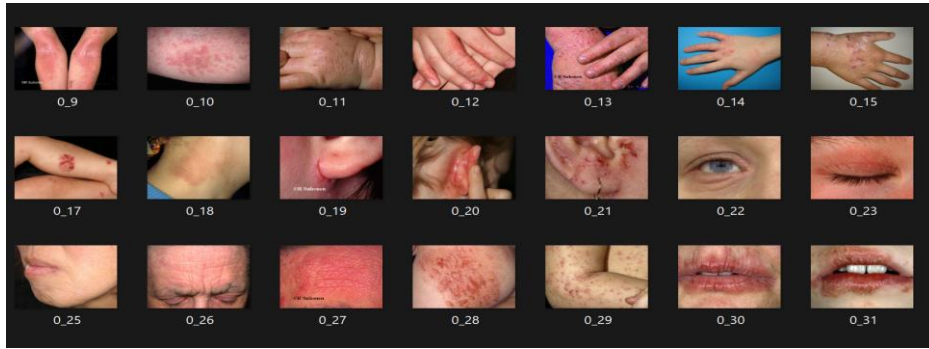


Figure 1: Example of input image for the class "Eczema - dermatitis".



Figure 2: Example of input image for the class "Atopic Dermatitis - atopic dermatitis".

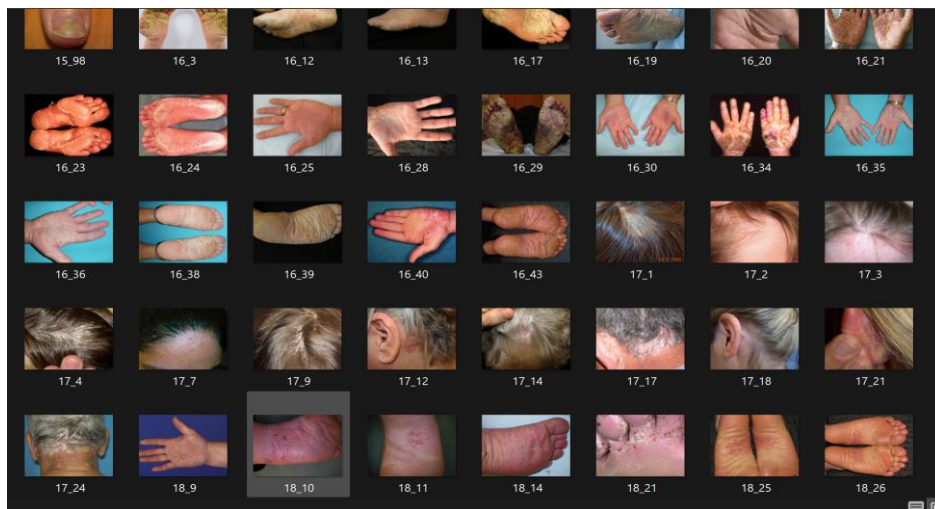


Figure 3: Example of input image for the class "Psoriasis, Lichen Planus and related diseases".

2.2. System architecture

The system consists of the following key modules:

The validation model performs a preliminary skin/non-skin image check and is used to filter the input data before it is fed to the classification module. Its main function is to detect irrelevant or unsuitable images for analysis, such as photos without skin, with background, artifacts, or poor quality. This check significantly reduces the load on the classification model, since the latter is applied only if the image actually contains skin.

In addition, this module prevents incorrect images from entering the system, in particular, the retraining process. This is especially important in the context of the retrain mechanism, where new images can be added to the training set. If a skinless image is accidentally introduced into the training process, it can degrade the model's accuracy or cause errors in its behavior. Thus, the validation

model acts as a filter that ensures data quality and increases the reliability of the entire system. This is comparable to the segmentation approach proposed in [9], although we used a lighter version optimized for speed.

Classification models - several architectures have been implemented, each of which is adapted to different conditions of use:

1. ResNet-50. A model with a deep structure and residual connections, which allows efficient training even on small samples. Implemented with `torchvision.models.resnet50(pretrained=True)`, it is adapted to our number of classes by modifying the output layer (`fc(fully connected) = nn.Linear(2048, num_classes)`). In testing, it showed high accuracy, but had a longer inference time.
2. EfficientNet-B0. Integrated through `efficientnet_pytorch.EfficientNet.from_pretrained('efficientnet-b0')`, the model demonstrates an optimal ratio of performance and accuracy. As part of our implementation, we modified `classifier._fc` to match the number of classes.
3. MobileNetV2. Thanks to `torchvision.models.mobilenet_v2(pretrained=True)`, the model was adapted to the task by replacing `classifier[10]` with `nn.Linear(1280, num_classes)`. It is used for mobile platforms, ensuring minimal resource consumption.
4. Custom CNN. Developed from scratch, consists of three convolutional layer blocks (Conv2d) using BatchNorm2d, ReLU(rectified linear unit), MaxPool2d, Dropout2d. To improve performance and reduce overfitting, two parallel layers are used for fully connected layers.

First Block includes:

1. Two convolutional layers (Conv2d) with 64 filters, using BatchNorm2d and ReLU activation.
2. Max pooling (MaxPool2d) with kernel size 2 and stride 2.
3. Dropout2d with a probability of 0.3 to reduce overfitting.

Second Block includes:

1. Two convolutional layers (Conv2d) with 128 filters, also using BatchNorm2d and ReLU activation.
2. Max pooling (MaxPool2d) with kernel size 2 and stride 2.
3. Dropout2d (0.3).

Third block includes:

1. Two convolutional layers (Conv2d) with 256 filters, using BatchNorm2d and ReLU activation.
2. Max pooling (MaxPool2d) with kernel size 2 and stride 2.
3. Dropout2d (0.3).

Global pooling includes adaptive average pooling (AdaptiveAvgPool2d) to size 4x4, allowing the model to preserve important features regardless of the original image size.

Parallel fully connected layers includes:

1. Two parallel fully connected pathways: first path - `Linear(256 * 4 * 4, 512)` with BatchNorm1d, ReLU, and Dropout, second path - `Linear(256 * 4 * 4, 512)` with BatchNorm1d, ReLU, and Dropout.
2. These layers are concatenated to form a single feature vector.

Output layer includes final layer: `Linear(1024, num_classes)` to produce the output classes.

Below on the Fig.4 is a generated diagram to visualize the structure of a custom neural network, mentioned in this description of classification modules.

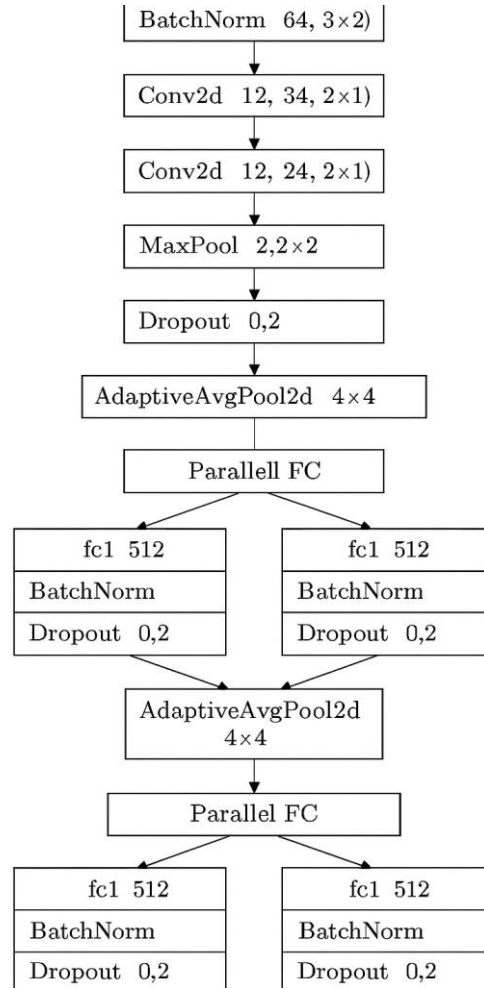


Figure 4: Custom CNN schema.

2.3. Implemented program interfaces (APIs)

To make it easy to integrate the system into any environment (for example, mobile or web applications), three RESTful APIs have been created:

1. POST /validate: accepts an image, returns a skin area mask or an invalidation message.
2. POST /classify: after validation, the image is classified by one of the models based on the request parameters (model type, speed or accuracy priority).
3. POST /retrain: accepts a list of classification result URLs, stored in an S3 object storage. The system iterates through each result and filters only those with a maximum predicted class probability below 0.8, identifying low-confidence predictions. Corresponding image files are then fetched from S3 and used to retrain the classification model. Evaluation is performed on a hold-out validation set to detect potential overfitting. Metrics before and after retraining are retrieved from and compared using a PostgreSQL database. If the new evaluation results demonstrate improvement (particularly in F1-score and accuracy) and overfitting is not detected, the updated model weights are saved back to S3, and new metrics are stored in PostgreSQL. Detailed version of this logic presented on Fig. 5 as block diagram.

2.4. Methodology for training models

The models were trained using PyTorch. An early stopping strategy was applied, as well as a mechanism for keeping the best weights on the validation set. The data was split 85:15 into training and validation samples.

To avoid overfitting, we implemented:

1. Regularization (Dropout = 0.3 for the custom CNN)
2. Augmentation (RandomHorizontalFlip, ColorJitter, Rotation)
3. Epoch control (patience = 5, max_epochs = 50)

Augmentation, including transformations such as RandomHorizontalFlip, ColorJitter, and RandomRotation, was applied using open-source libraries [11], [12]. These augmentations simulate natural variations in lighting, orientation, and contrast, thereby increasing the diversity and generalizability of the training set.

The developed CNN was trained using an input resolution of 64×64 pixels, which ensured reduced computational complexity suitable for lightweight experimentation. The training process employed a batch size of 32, with the Adam optimizer selected for its adaptive learning rate capabilities. The loss function used was CrossEntropyLoss, as it is well-suited for multi-class classification tasks and provides stable convergence during optimization.

2.5. Evaluation metrics

To comprehensively assess the performance of the implemented models, a set of standard evaluation metrics was used:

- Accuracy – measures the proportion of correctly classified instances over the total number of samples. This metric provides a general view of the model's overall correctness across all classes.
- Precision, Recall, and F1-score – these class-wise metrics offer deeper insights into the model's behavior on individual categories. Precision indicates the proportion of true positives among all predicted positives, while recall reflects the proportion of true positives among all actual positives. The F1-score, as the harmonic mean of precision and recall, balances the trade-off between these two measures, particularly in imbalanced class distributions.
- Confusion Matrix – a tabular visualization of prediction results that helps identify specific patterns of misclassification. This tool is crucial for understanding which classes tend to be confused with others and can guide further refinement of preprocessing or model architecture.

All models were evaluated using an independent validation sample, constructed to reflect real-world deployment conditions. This dataset included randomly selected images with diverse characteristics such as variable lighting, background artifacts, and varying skin tones, which allowed for a realistic estimation of model robustness and generalization capabilities. While ISIC 2019 [13] provides a reliable benchmark, our focus was on broader multi-class datasets with varied imaging conditions.

3. Analysis of the latest research and publications

The global scientific community continues to actively develop areas related to the use of deep learning methods for medical diagnostics, including dermatology. Studies have shown that the use of neural networks can achieve classification accuracy comparable to or higher than the average accuracy of dermatologists, especially in conditions of limited clinical experience or difficult cases of visual distinction of pathologies. Multitask approaches such as those explored in [14] also demonstrate potential in classification under complex criteria.

In publications [10,15], considerable attention is paid to the use of ResNet, EfficientNet, and MobileNet architectures[16]. ResNet (Residual Networks) is characterized by the depth and presence of residual connections, which facilitates the training of deep networks without losing the gradient. EfficientNet provides an optimal balance between accuracy and number of parameters by combining model depth, width, and resolution. MobileNet, on the other hand, is designed for mobile devices

with limited computing resources and uses deep convolution to reduce the number of parameters while maintaining acceptable accuracy.

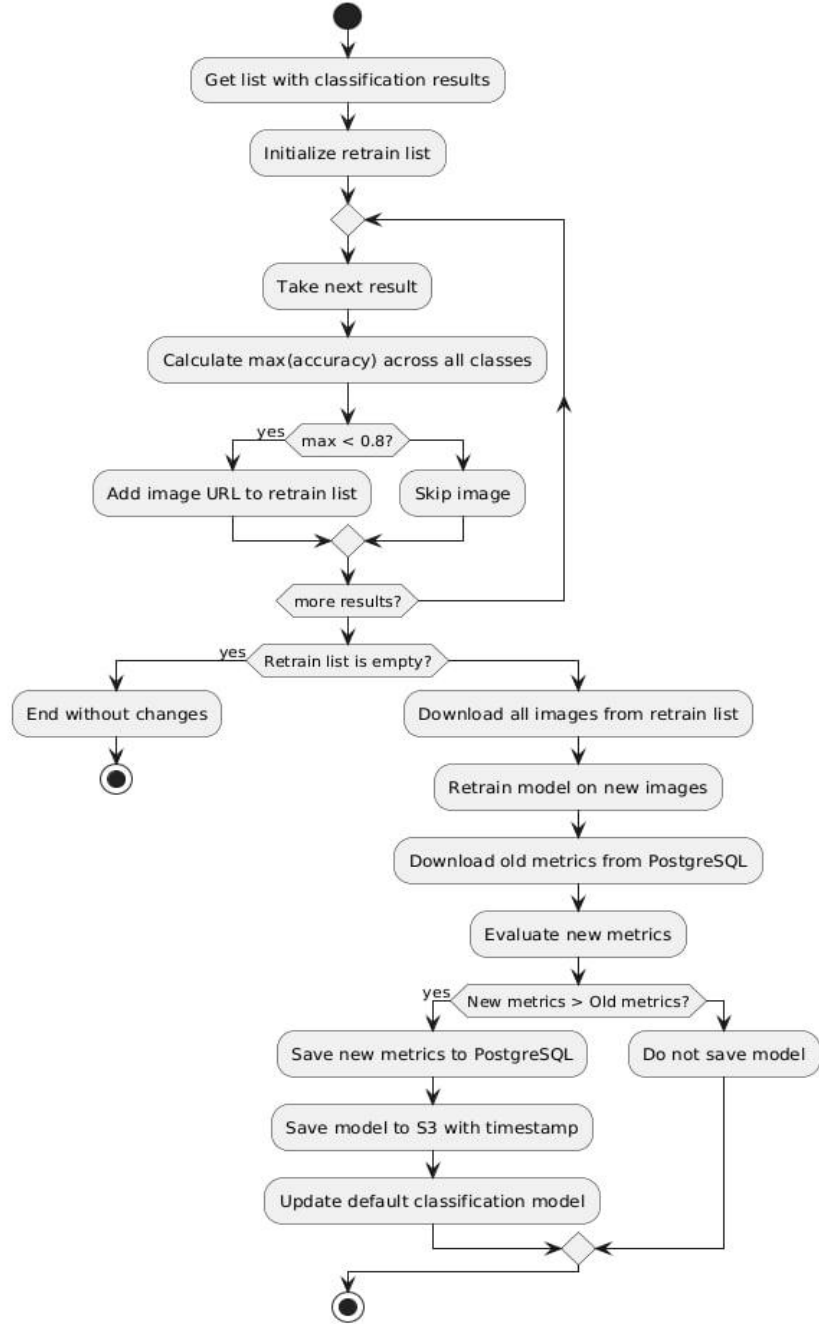


Figure 5: Block diagram of the retraining request logic.

A significant number of studies (e.g., [4,9,16]) focus on the task of skin segmentation in images, which is an important stage of data preprocessing before feeding it to classifiers. In particular, models such as U-Net (a neural network for biomedical image segmentation) [9] and its derivatives demonstrate a good ability to separate skin and background even in difficult lighting conditions or in the presence of artifacts. In our work, this stage was implemented as a separate validation network of a simplified structure, which allowed us to strike a balance between speed and filtering accuracy.

Another important aspect, according to publications [6-7], is the need to dynamically update classification models. Often, models trained on standard open datasets have limited generalizability when applied to new types of images, especially those captured by mobile devices with different camera quality. That is why modern approaches are actively researching methods of adaptive

retraining, including mechanisms for automatically tracking changes in metrics and replacing the main model in the production environment.

The implemented system uses four deep learning models that meet different requirements for accuracy, performance, and resource consumption. Among them, the most powerful is ResNet-50, a model with a deep architecture and residual connections that help to avoid the problem of gradient attenuation during training. It showed high classification accuracy, but its use requires significant computing resources, so this model is more suitable for server or desktop solutions.

A more balanced option was the EfficientNet-B0 model, which provided the best ratio between accuracy and efficiency with a much smaller number of parameters. It was this model that was chosen as the base model for implementation in the production environment, as it combines good quality with high performance.

For cases where speed and low hardware requirements are critical (for example, in mobile applications), MobileNetV2 was used. This model is optimized for devices with limited resources and allows for fast classification, although it demonstrates slightly lower accuracy compared to other models. A custom neural network developed manually plays a separate role in the study. It has a simple structure, a small number of parameters, and was created for the purpose of flexible experimentation. Although its metrics are inferior to previous models, the custom CNN is an effective tool for analyzing architectural solutions, rapid retraining, and testing new approaches. Due to its low complexity, it is well suited for research and prototyping purposes.

Thus, the work involved architectures that correspond to different scenarios: from high-performance to lightweight mobile solutions. This approach allowed us to flexibly evaluate the effectiveness of the models depending on the conditions of use. Table 1 shows a comparison of the main characteristics of the models used:

Table 1
Comparative characteristics of convolutional neural network architectures

Architecture	Amount of parameters	Advantages	Limitations
ResNet-50	~23.5 mln	High accuracy, stable training	Heavy weight, low performance
EfficientNet-B0	~5.3 mln	Balanced performance suitable for production	Long training time
MobileNetV2	~3.4 mln	Fast, lightweight, mobile-friendly	Slightly lower accuracy
Custom CNN	~2.1 mln	Simple, effective, easy to modify	Limited depth, lower accuracy

Thus, the analysis of scientific sources, combined with practical experience in deploying the system, confirms the feasibility of using an adaptive modular architecture with validation, classification, and a retraining mechanism. Each of the models is selected according to the conditions of use: performance - for the server environment, speed - for mobile, flexibility - for experimental expansion of the system. The developed custom CNN is worth mentioning, as it has increased resistance to retraining due to the use of such techniques as Batch Normalisation, Dropout, and parallel fully connected paths that allow deeper processing of input data. Despite having the smallest number of parameters among all architectures (~2.1 million), this structure demonstrated a stable quality of results. The model achieved an average F1-score of ≈ 0.80 , which indicates its effectiveness in tasks with limited resources and data volumes. Combined with its simple structure and modifiability, this model is promising for further experiments and adaptive learning.

4. Research results and their discussion

During the experimental part of the study, a series of tests were conducted on the implemented models for the task of classifying skin pathologies. The goal was to evaluate the accuracy, resistance to overfitting, efficiency of the segmentation module, and the potential for dynamic model updating based on user feedback. Particular attention was paid to the analysis of false classifications, the impact of the validation filter, and the behaviour of the custom network in complex cases.

4.1. The problem of overfitting

At the initial stage of training all models, we observed the effect of overtraining, especially pronounced in the custom CNN and ResNet-50. This was manifested in a significant gap between the training and validation samples: the accuracy on the training data reached 95-98%, while on the validation data it dropped to 75-80%. The reasons were:

1. limited diversity in the examples of the training set;
2. visual redundancy of the background in many images;
3. the presence of artefacts that affected model attention.

The implementation of a lightweight validation module (skin/non-skin stage) allowed us to **eliminate $\approx 12\%$ of unsuitable images and increase the average F1-score** of the models by 6-8%. This was especially important for stabilising the results of the custom network, where the F1-score increased from 0.74 to 0.80.

4.2. Comparative evaluation of models

Table 2 shows the metrics of the main models on the independent test set.

Table 2
Comparison of Models by Accuracy and F1-score

Model	Accuracy	F1-score (avg)	Recall	Precision	Parameters	Training time (min) - 12169 images	Inference time (min) - 1000 images
ResNet-50	88.7%	0.86	0.85	0.88	$\sim 23.5\text{M}$	1500	300
EfficientNet-B0	91.2%	0.89	0.90	0.88	$\sim 5.3\text{M}$	1250	250
MobileNetV2	87.4%	0.85	0.84	0.86	$\sim 3.4\text{M}$	1000	200
Custom CNN	83.1%	0.80	0.79	0.81	$\sim 2.1\text{M}$	800	160

The EfficientNet-B0 model delivered the highest balanced performance and, along with MobileNetV2, is considered the most suitable candidate for production deployment. EfficientNet demonstrated high classification accuracy while maintaining reasonable inference speed. MobileNetV2, in turn, showed comparable performance but with lower computational demand, making it especially valuable for mobile or embedded environments.

Confusion matrices on the Figs. 6-7 were used to visualize and compare prediction quality across different architectures. EfficientNet-B0 correctly identified 481 cases of Basal Cell Carcinoma (Class 4), while MobileNetV2 slightly outperformed it with 485 correct classifications. Custom CNN achieved 455 correct predictions for the same class, whereas ResNet-50 handled approximately 466 correctly. This trend was consistent across other classes as well — MobileNetV2 and EfficientNet generally produced fewer false positives and better overall balance, particularly in distinguishing overlapping cases such as Psoriasis vs Seborrheic Keratoses or Warts vs Atopic Dermatitis.

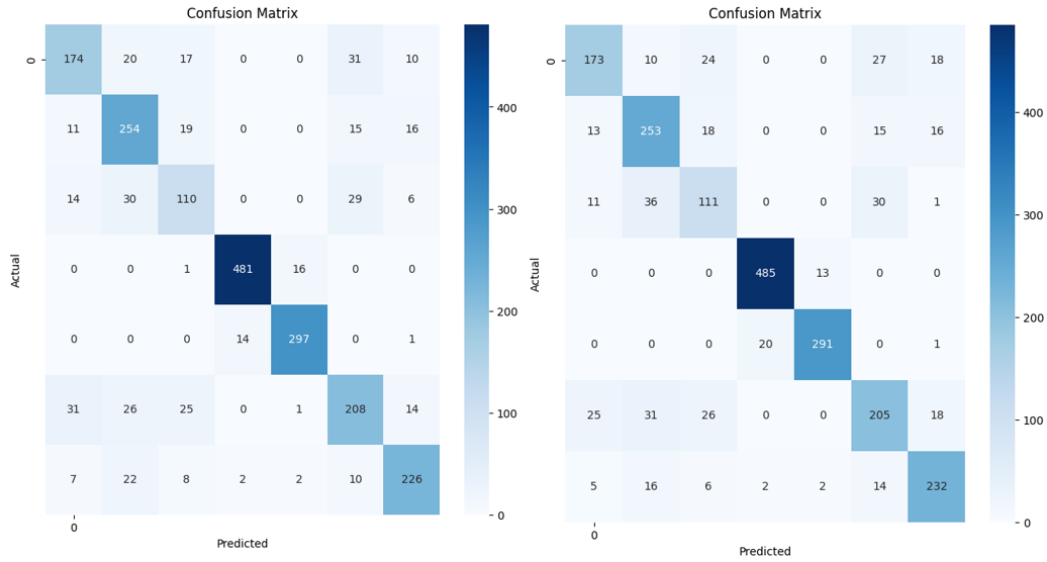


Figure 6: Confusion matrices for EfficientNet and MobileNetV2 models.

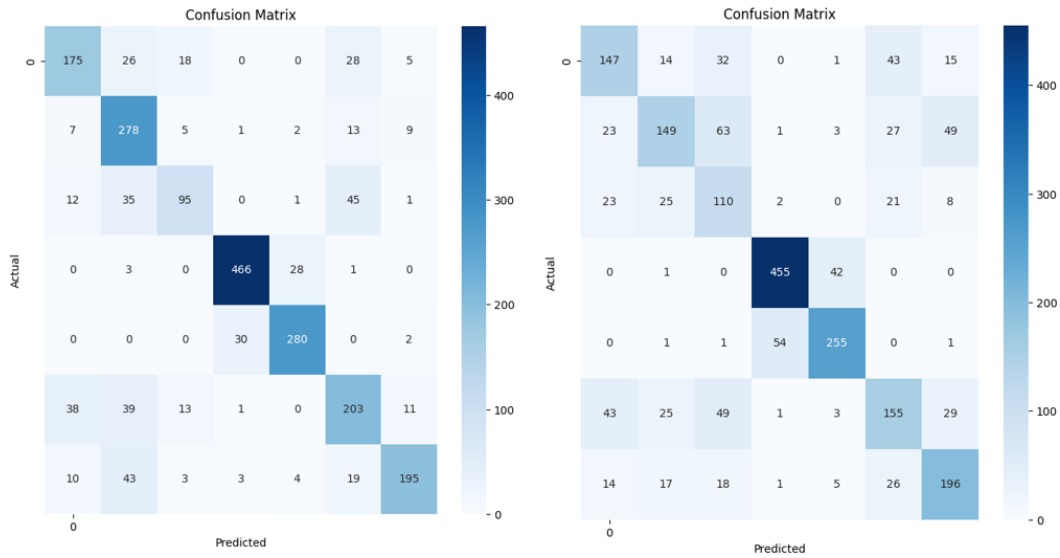


Figure 7: Confusion matrices for ResNet-50 and Custom CNN models.

Both models demonstrate performance that is on par with, or surpasses, the most prominent benchmarks in the field of automated dermatological diagnostics. For comparison, this article [17] demonstrated that top-performing CNNs reached diagnostic accuracy levels comparable to expert dermatologists, with accuracies up to 86% on pigmented lesion classification tasks. For instance, the system developed by Liu et al. in Nature Medicine (2020) [18] reported a top-1 accuracy of 0.85 in classifying over 400 types of skin conditions using a large-scale multiclass dataset. Similarly, the comparative diagnostic study published in Lancet Oncology (2019) showed that top-performing convolutional neural networks reached accuracies of approximately 0.86, comparable to expert dermatologists in controlled experimental settings. In contrast, our system — specifically the EfficientNet-B0 and MobileNetV2 architectures — achieved an overall classification accuracy of up to 0.91 and a macro-averaged F1-score of 0.89, based on a diverse, real-world dataset. Comparable high accuracy levels were also reported by Han et al. [19] for binary classification of benign vs. malignant tumors. A combination of MobileNetV2 with LSTM has also been explored by Ahsan et al. [20], showing enhanced sequential image processing. These results highlight the clinical applicability and competitive advantage of the proposed approach in practical diagnostic scenarios [18,19]. The Figs. 8-9 below show typical training dynamics.

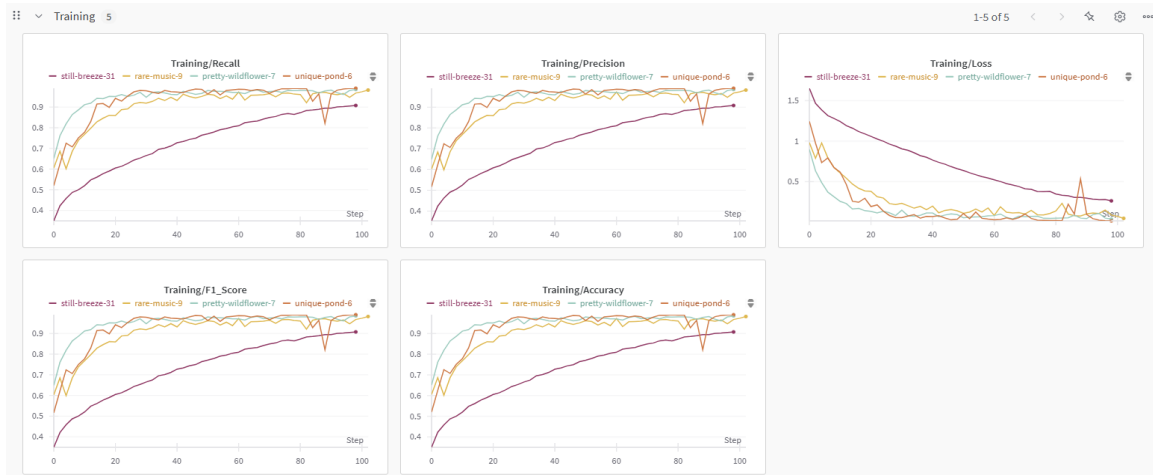


Figure 8: Diagrams of metrics changes during model training.

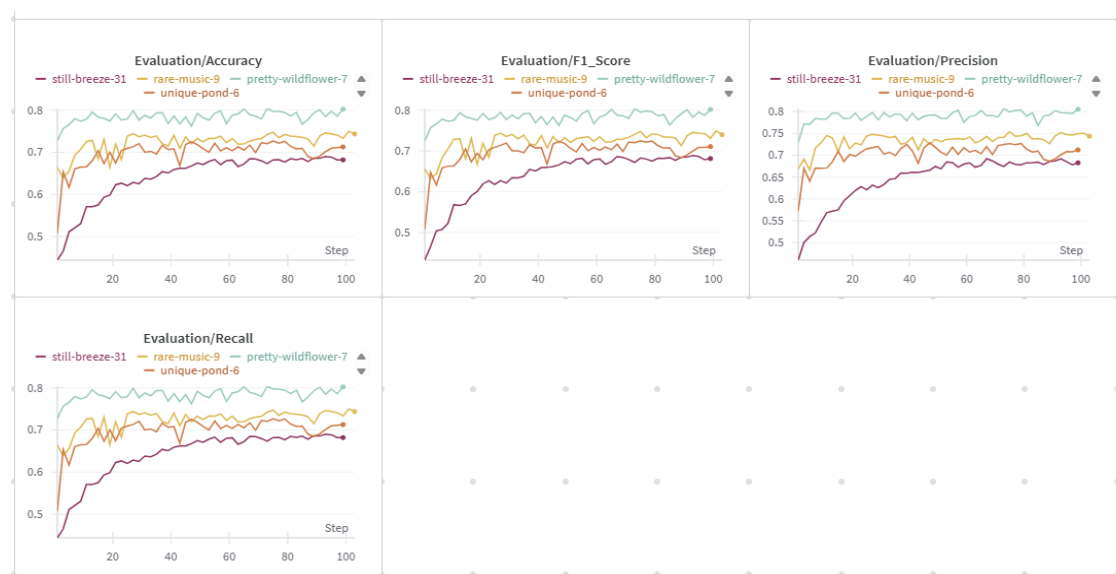


Figure 9: Charts of metrics change on the test (evaluation) sample (allows to assess the degree of generalisation and avoidance of overlearning).

4.3. Impact of validation segmentation

The segmentation module has two key functions: filtering of unsuitable images - up to 12% of input images that do not contain skin, are backlit, blurred, etc. are filtered out; reduced errors - using only relevant fragments reduced the number of false positives by $\approx 9\%$. The implemented segmentation has significantly improved the quality of classification in weak or noisy images (especially on mobile cameras with automatic white balance). On the Fig. 10 displayed some of the elements for two classes in the validation dataset.

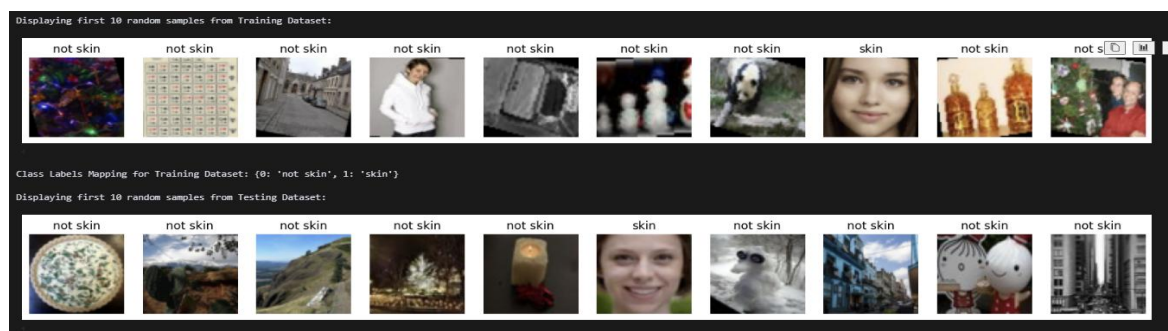


Figure 10: Examples of input images for a validation neural network.

4.4. Behaviour and effectiveness of the retrain mechanism

Adaptive retraining was tested in an emulated user feedback environment: 200 images were manually labelled as misclassified. Results:

1. EfficientNet-B0: F1-score increased from 0.89 to 0.91
2. The system automatically updated the model after exceeding the metrics (torchmetrics integration)
3. Retraining time: ~4 min 20 s on RTX 3060 (10 epochs, batch=32)

The /retrain mechanism allows the system to improve in real time, reducing the need for full retraining. This meets modern requirements for the life cycle of AI systems.

4.5. Analysis of misclassification

1. Overlapping pathologies: For example, the visual similarity between seborrhoeic dermatitis and psoriasis can lead to misclassifications.
2. Atypical skin areas: Dermatitis localised to the scalp, ears or fingers. However, this problem will be less of an issue when using an existing dataset, as the sample contains a large number of different variations, including both typical and atypical cases.
3. Poor lighting and blurred images: Low contrast of skin lesions can make it difficult to recognise them correctly [21].

To minimise these problems impact, the model will be trained using image augmentation, which allows for additional image variations and thus reduces the likelihood of overfitting and improves the overall generalisation capability of the model.

Figs. 11-12 show examples of image augmentation that demonstrate changes that can be applied to skin lesions, such as random rotation, changes in brightness, contrast, or the addition of noise, which allows for more variation in training and improves the model's robustness to different real-world image conditions.



Figure 11: Examples of image augmentation using the Resnet model.



Figure 12: Examples of image augmentation using the MobileNet model.

The analysis of these cases allowed us to adjust the augmentation strategy and strengthen the role of validation filtering.

4.6. Generalised conclusions of the experiments

1. The EfficientNet-B0 model is optimal for use in mobile and production environments.
2. Validation segmentation is critical - it not only improves accuracy but also stabilises the model's behaviour.

3. Implementation of a retrain mechanism allowed us to adapt the model to new data without re-training.
4. The custom CNN showed high flexibility and suitability for rapid prototyping and experimentation, but rather low accuracy of the results compared to other networks.

Conclusions

This paper presents an information technology for the preliminary diagnosis of dermatological conditions based on deep learning methods, which made it possible to improve the accuracy of automated classification of skin images through the implementation of a modular architecture that includes preliminary validation, classification, and adaptive retraining.

The developed system solves an urgent problem in the field of digital medicine - ensuring reliable preliminary diagnosis when using heterogeneous input data (mobile photos, different lighting quality, artefacts). The developed validation module filters out irrelevant images, which makes it possible to reduce classification errors associated with background noise.

The classification system is implemented as a set of interchangeable models (ResNet-50, EfficientNet-B0, MobileNetV2, custom CNN), each of which is focused on a specific application scenario: high accuracy, speed, or resource minimisation. Testing has shown that the EfficientNet-B0 model provides the best balance between performance and computational efficiency.

The key advantage of the developed technology is the support for adaptive retraining based on user feedback. Experimental results have shown that this mechanism allows the model to be improved without the need for complete retraining, which significantly reduces the time required to deploy updates in a real environment.

However, several limitations of the current system should be noted. First, the dataset used for training contains a limited number of atypical cases (e.g., lesions on the scalp, folds, or under varying lighting conditions), which may affect generalization performance. Second, although EfficientNet-B0 and MobileNetV2 demonstrate acceptable inference speed, latency could still pose a challenge for real-time mobile or embedded systems, especially in low-resource environments.

The obtained results confirm the practical significance of the developed information technology, which can be used in both research and commercial medical products for primary diagnosis. The developed system demonstrates stable quality in real-world image classification, scalability, and self-learning capability.

Future work includes expanding the dataset to cover atypical and rare dermatological conditions, optimizing inference speed for edge deployment, and improving the robustness of validation segmentation under diverse image conditions. As highlighted in [22], the integration of explainable AI techniques is essential in the medical domain to enhance transparency and trust. Incorporating such post-hoc interpretability tools could further improve the clinical applicability of our system.

Declaration on Generative AI

The author(s) have **not employed any Generative AI tools**.

References

- [1] World Health Organization, WHO First Global Meeting on Skin NTDs Calls for Greater Efforts to Address Their Burden, 2023. URL: <https://www.who.int/news/item/31-03-2023-who-first-global-meeting-on-skin-ntds-calls-for-greater-efforts-to-address-their-burden>
- [2] A. Esteva et al., Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks, *Nature*, 542(7639) (2017) 115–118. doi: 10.1038/nature21056.
- [3] A. Holzinger, C. Biemann, C. S. Pattichis, and D. B. Kell, What Do We Need to Build Explainable AI Systems for the Medical Domain? *arXiv preprint*, arXiv:1712.09923,
- [4] M. Pietron and M. Wielgosz, Retrain or Not Retrain? – Efficient Pruning Methods of Deep CNN Networks, *arXiv preprint*, arXiv:2002.07051, 2020.

- [5] Kaggle, Skin diseases image dataset. URL: Available: <https://www.kaggle.com/datasets/ismailpromus/skin-diseases-image-dataset>
- [6] P. Tschandl, C. Rosendahl, and H. Kittler, The HAM10000 Dataset: A Large Collection of Multi-Source Dermatoscopic Images of Common Pigmented Skin Lesions, *Sci. Data*, 5 (2018). doi: 10.1038/sdata.2018.161.
- [7] N. C. Codella et al., Skin Lesion Analysis Toward Melanoma Detection: A Challenge at the 2017 International Symposium on Biomedical Imaging, in *Proc. IEEE Int. Symp. Biomed. Imaging (ISBI)*, 2018, pp. 168–172, doi: 10.1109/ISBI.2018.8363547.
- [8] M. Combalia et al., BCN20000: Dermoscopic Lesions in the Wild, *arXiv preprint*, arXiv:1908.02288, 2019.
- [9] O. Ronneberger, P. Fischer, and T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, in *Proc. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*, 2015, pp. 234–241.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, Deep Residual Learning for Image Recognition, in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [11] Albumentations, Fast and Flexible Image Augmentations. URL: <https://albumentations.ai/docs/>
- [12] PyTorch, Torchvision Transforms Documentation. URL: <https://pytorch.org/vision/stable/transforms.html>
- [13] ISIC 2019, International Skin Imaging Collaboration Challenge – Official Dataset. URL: <https://challenge2019.isic-archive.com>
- [14] J. Kawahara, S. Daneshvar, G. Argenziano, and G. Hamarneh, Seven-Point Checklist and Skin Lesion Classification Using Multitask Multimodal Neural Nets, *IEEE J. Biomed. Health Inform.*, 23,(2) (2019) 538–546.
- [15] M. Tan and Q. V. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 6105–6114.
- [16] A. G. Howard et al., MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, *arXiv preprint*, arXiv:1704.04861, 2017.
- [17] P. Tschandl et al., Comparison of the Accuracy of Human Readers Versus Machine-Learning Algorithms for Pigmented Skin Lesion Classification, *Lancet Oncol.*, 20 (7) (2019) 938–947. doi: 10.1016/S1470-2045(19)30333-X.
- [18] Y. Liu et al., A Deep Learning System for Differential Diagnosis of Skin Diseases, *Nat. Med.*, 26, 6, (2020) 900–908. doi: 10.1038/s41591-020-0842-3.
- [19] S. S. Han et al., Classification of the Clinical Images for Benign and Malignant Cutaneous Tumors Using a Deep Learning Algorithm, *J. Invest. Dermatol.*, 138 7 (2018) 1529–1538. doi: 10.1016/j.jid.2018.01.028.
- [20] M. M. Ahsan, M. Mahmud, P. K. Saha, I. H. Sarker, and A. Khandakar, Classification of Skin Disease Using Deep Learning Neural Networks with MobileNet V2 and LSTM, *ResearchGate*, 2021.
- [21] A. Bissoto, M. Fornaciali, S. Avila, E. Valle, and L. Oliveira, (De)Constructing Bias on Skin Lesion Datasets, *arXiv preprint*, arXiv:1904.08818, 2019.
- [22] Y. Yao, X. Liu, and X. Li, A Review of Explainable Deep Learning for Medical Imaging, *IEEE Access*, 10 (2022) 20705–20722.