

SKIN CANCER CLASSIFICATION USING CONVOLUTIONAL NEURAL NETWORKS

Yazeed Al Moaiad¹

Faculty of Computer and Information
 Al-Madinah International University
 Kuala Lumpur, Malaysia
 yazeed.alsayed@mediu.edu.my

Marwan Ehssi²

Faculty of Computer and Information
 Al-Madinah International University
 Kuala Lumpur, Malaysia
 CM315@lms.mediu.edu.my

Abstract—Early assessment of suspicious skin lesions can improve access to care, but dermoscopic interpretation depends on specialist expertise and automated classifiers must be evaluated carefully under severe class imbalance. This paper presents a seven-class convolutional neural network pipeline and a cloud-connected desktop prototype developed from a thesis-scale study. The classifier uses an ImageNet-initialized MobileNetV2 backbone followed by global average pooling, dropout, a 128-unit ReLU layer, and a seven-way softmax output. The HAM10000 dataset contains 10,015 dermoscopic images and was stratified into training, validation, and test subsets of 7,010, 1,502, and 1,503 images. Images were resized to 224×224 pixels, normalized, augmented, and trained with class weighting in TensorFlow/Keras on Google Colab. On the held-out test set, the reported accuracy was 93.20%, with weighted precision, recall, and F1-score of 94.06%, 93.21%, and 93.51%, respectively. Melanoma recall was 88.62%, whereas dermatofibroma precision was 45.16% with only 17 test examples, underscoring the effect of class imbalance. A Flask REST API and PySide6 client produced an average end-to-end response time of 3.3 s. The results support the feasibility of the prototype as a decision-support tool, while the absence of clinical and external validation limits claims of clinical readiness.

Keywords— skin cancer; skin lesion classification; convolutional neural networks; MobileNetV2; transfer learning; HAM10000; cloud deployment

I. INTRODUCTION

Skin cancer is a prevalent health concern, and timely assessment of suspicious lesions is important because delayed or incorrect diagnosis may adversely affect treatment. Conventional assessment relies on visual examination and dermoscopy by trained clinicians. Although these procedures are established, interpretation can vary with experience, and specialist access is not uniform. Automated analysis is therefore of interest as an adjunct to—not a replacement for—clinical judgment.

Deep learning has changed medical-image analysis by learning hierarchical image representations directly from data. Convolutional neural networks (CNNs) learn edges, textures, and progressively higher-level structures through convolution, nonlinear activation, pooling, and classification layers [1], [3]. In skin-lesion research, large benchmark datasets and challenge protocols have enabled reproducible comparison of classification methods [2], [4]. Transfer learning further reduces the cost of training by adapting representations learned from large image corpora to smaller medical datasets [5], [6].

Many studies emphasize model accuracy, while fewer describe an end-to-end system that connects preprocessing, inference, and a usable interface. The source study addressed both aspects by training a CNN-based classifier on HAM10000 and integrating the resulting model with a Flask service and a desktop graphical interface. The present paper condenses that work into a conference-style report while retaining the reported data, configuration, and performance evidence.

A. Contributions

The work makes three practical contributions. First, it documents a seven-class dermoscopic-image pipeline built around MobileNetV2 transfer learning, data augmentation, class weighting, early stopping, and selective fine-tuning. Second, it reports both aggregate and per-class results, allowing examination of the impact of HAM10000 class imbalance. Third, it demonstrates an integrated client-server prototype in which a PySide6 desktop client communicates with a Flask prediction endpoint hosted in a cloud runtime.

II. RELATED WORK

Earlier computer-aided skin-lesion systems commonly separated preprocessing, handcrafted feature extraction, and conventional classification. Color, texture, and shape descriptors were supplied to algorithms such as support-vector machines, k-nearest neighbors, random forests, or naïve Bayes. These pipelines were interpretable but depended heavily on manually designed features and were less able to represent complex lesion morphology.

CNNs replaced much of this manual engineering with end-to-end feature learning. Esteva et al. demonstrated the potential of deep neural networks for dermatologist-level image classification [1], while the ISIC challenge work of Codella et al. established benchmark tasks and evaluation protocols [2]. Architectures such as ResNet and EfficientNet subsequently improved representation learning through residual connections and compound scaling [9], [10]. Transfer learning has become especially important where labeled medical data are limited [5], and the source literature reports applications of deep CNNs, transfer learning, and multi-resolution ensembles to skin lesions [11], [12].

Dataset composition remains a central issue. HAM10000 provides 10,015 multi-source dermoscopic images across seven diagnostic categories [4], but the class distribution is highly skewed toward melanocytic nevi. Consequently,

overall accuracy can conceal poor minority-class behavior; precision, recall, F1-score, and confusion-matrix analysis are needed alongside accuracy. The source study additionally identified a deployment gap: many experimental classifiers do not provide an accessible interface or a modular inference service. Its prototype therefore combined classification with a cloud-hosted REST API and desktop client.

III. DATASET

A. Composition and Split

The experiments used the HAM10000 dataset, a public collection of 10,015 dermoscopic images labeled by medical experts [4]. The seven categories are actinic keratosis/intraepithelial carcinoma (akiec), basal cell carcinoma (bcc), benign keratosis-like lesions (bkl), dermatofibroma (df), melanoma (mel), melanocytic nevi (nv), and vascular lesions (vasc). The source reports stratified, mutually exclusive training, validation, and test subsets containing 7,010 (70%), 1,502 (15%), and 1,503 (15%) images, respectively.

Table 1. Reported dataset split

Subset	Images	Percentage
Training	7,010	70%
Validation	1,502	15%
Test	1,503	15%
Total	10,015	100%

The test set preserves the underlying imbalance: nv accounts for 1,006 of 1,503 images, whereas df and vasc contain only 17 and 22 images. The remaining supports are 49 akiec, 77 bcc, 165 bkl, and 167 mel images. This distribution provides broad category coverage but makes weighted aggregate metrics strongly influenced by the majority class.

B. Preprocessing and Augmentation

All images were converted to RGB, resized to 224×224 pixels, and scaled from 8-bit intensity values to $[0, 1]$ by division by 255. Training-time augmentation included rotations, horizontal flips, and width and height shifts. Class weights were computed from the training labels to reduce majority-class dominance. The test generator applied only resizing and normalization and did not shuffle samples, allowing predictions to remain aligned with reference labels.

IV. METHODOLOGY

A. Classifier Architecture

The implementation uses MobileNetV2 with ImageNet weights as the convolutional backbone. The original classification layer is removed (include_top=False), and the backbone output is passed through global average pooling, dropout with rate 0.3, a 128-unit fully connected layer with ReLU activation, and a seven-unit softmax layer. This architecture is more precisely described as a transfer-learning CNN than as a newly designed custom network. Its lightweight backbone supports both cloud training and responsive inference.

B. Training Protocol

Training was performed with TensorFlow/Keras [7] in Google Colab using GPU acceleration. A random seed of 42 was reported for Python, NumPy, and TensorFlow operations. In the first phase, the pretrained backbone was

frozen and the classification head was optimized with Adam [8], an initial learning rate of 0.001, categorical cross-entropy, and a batch size of 32. Training was configured for at most 25 epochs, with early stopping on validation loss (patience 5), learning-rate reduction on plateau, and retention of the model with the highest validation accuracy. Training stopped at epoch 18, and the best reported validation loss was 0.1823.

In the second phase, upper MobileNetV2 layers were made trainable while earlier layers remained frozen. Fine-tuning used Adam at 1×10^{-5} for up to 10 additional epochs. The reduced learning rate was intended to adapt high-level ImageNet features to dermoscopic appearance without destabilizing the pretrained representation. Table 2 summarizes the configuration explicitly reported in the thesis and its implementation appendix.

Table 2. Model and training configuration

Item	Reported setting
Input / output	$224 \times 224 \times 3 / 7$ classes
Backbone	MobileNetV2, ImageNet initialized
Classification head	GAP $\rightarrow$ Dropout(0.3) $\rightarrow$ Dense(128, ReLU) $\rightarrow$ Softmax
Batch size	32
Optimizer / loss	Adam / categorical cross-entropy
Initial learning rate	0.001
Initial training	Maximum 25 epochs; stopped at epoch 18
Early stopping	Validation loss; patience = 5
Fine-tuning rate	1×10^{-5} ; upper backbone layers trainable
Imbalance handling	Balanced class weights

C. Evaluation Measures

Performance was evaluated using accuracy, precision, recall, F1-score, and a seven-class confusion matrix. For each class, precision measures the fraction of predicted positives that are correct, recall measures the fraction of reference positives recovered, and F1 is the harmonic mean of precision and recall. Weighted averages account for class support. In this multiclass medical setting, class-wise recall is especially important because aggregate accuracy may be dominated by benign majority classes.

D. Prototype Integration

The trained model was embedded in a modular client-server prototype. A PySide6 desktop application selects and uploads a dermoscopic image. The Flask endpoint POST /predict receives the multipart image, converts it to RGB, resizes and normalizes it, invokes the saved TensorFlow model, and returns a JSON object containing the predicted class and probability scores for all seven categories. The source states that images are processed transiently rather than retained by the server.

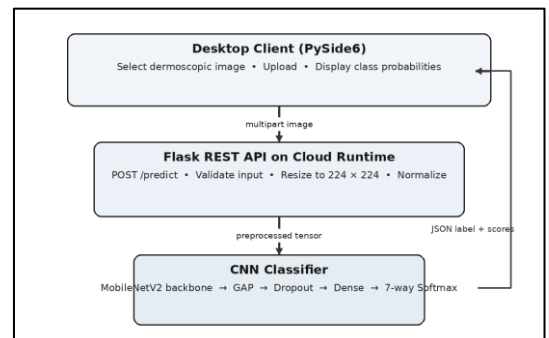


Figure 1. Source-supported client-server inference workflow.

V. RESULTS AND DISCUSSION

A. Aggregate Performance

On the 1,503-image test set, the reported overall accuracy was 93.20%. Weighted precision, recall, and F1-score were 94.06%, 93.21%, and 93.51%, respectively. The close weighted scores indicate stable aggregate classification, while the class-wise results in Table 3 reveal substantially greater variation for minority categories.

Table 3. Per-class classification report (test set, $n = 1,503$)

Class	Precision	Recall	F1	Support
akiec	82.69%	87.76%	85.15%	49
bcc	86.59%	92.21%	89.31%	77
bkl	90.24%	89.70%	89.97%	165
df	45.16%	82.35%	58.33%	17
mel	82.68%	88.62%	85.55%	167
nv	99.07%	95.13%	97.06%	1,006
vasc	68.97%	90.91%	78.43%	22
Weighted	94.06%	93.21%	93.51%	1,503

B. Class-wise Error Analysis

Melanocytic nevi achieved the highest F1-score (97.06%), consistent with the large number of nv examples. Melanoma recall was 88.62%, corresponding to 148 correctly classified melanoma images out of 167. The confusion matrix shows seven melanoma images assigned to nv and five to bkl. Conversely, 23 nv images were predicted as melanoma. These errors are consistent with the visual similarity among pigmented lesions noted in the source discussion.

Dermatofibroma illustrates the limitations of accuracy alone. Its recall was 82.35% (14 of 17), but precision was only 45.16%, indicating that other lesion types were frequently assigned to df. Vascular lesions also had limited support: recall reached 90.91%, but precision was 68.97%. These results suggest that class weighting and augmentation helped minority-class sensitivity, yet the small supports leave their estimates less stable than those for nv, bkl, or mel.

Actual class	akiec	bcc	bkl	df	mel	nv	vasc
akiec	43	1	3	1	0	0	1
bcc	0	71	0	3	1	1	1
bkl	0	5	148	3	7	0	2
df	0	0	1	14	0	1	1
mel	2	1	5	2	148	7	2
nv	7	4	7	6	23	957	2
vasc	0	0	0	2	0	0	20
Predicted class	akiec	bcc	bkl	df	mel	nv	vasc

Figure 2. Confusion matrix reported for the 1,503-image test set (1,401 correct predictions).

C. Training Behavior and Literature Context

Training and validation curves in the source improved through epoch 18, and the reported best validation loss was 0.1823. The similarity of the curves was interpreted as limited overfitting, supported by augmentation, dropout, class weighting, early stopping, and fine-tuning. Nevertheless, a single internal validation trajectory cannot establish generalization to other acquisition devices, institutions, or patient populations.

For context, the thesis compared its 93.20% accuracy with literature-reported values of 89.50% for a ResNet50 study [13] and 85.34% for a MobileNetV2 study [14]. These figures were not reproduced under a common split or experimental protocol and therefore should be interpreted as contextual benchmarks rather than controlled head-to-head evidence. The present paper does not claim statistical superiority over those studies.

D. System Response and Limitations

Prototype tests covered image upload, preprocessing, prediction, API communication, and interface interaction. The reported mean end-to-end response time was 3.3 s, composed of 0.8 s for upload, 1.1 s for preprocessing, and 1.4 s for prediction. This latency is compatible with interactive use in the demonstrated environment, although throughput under concurrent load was not evaluated.

Table 4. Reported prototype response time

Operation	Average time
Image upload	0.8 s
Image preprocessing	1.1 s
Prediction generation	1.4 s
Total response	3.3 s

Several limitations constrain interpretation. The evaluation used one public dataset with marked class imbalance, no independent external dataset, and no clinical validation by medical professionals. The source did not report patient-level partitioning, confidence intervals, or statistical uncertainty for class-wise metrics. Weighted averages are also strongly influenced by the 1,006-image nv test class. Finally, the prototype was evaluated as a technical system rather than in a clinical workflow. Future work should use larger and more diverse datasets, patient-level splits, external and prospective validation, explainable outputs, and clinician-centered usability testing.

VI. CONCLUSION

This paper presented a concise account of a seven-class dermoscopic-image classification system based on MobileNetV2 transfer learning. Using the reported 70/15/15 split of HAM10000, the classifier achieved 93.20% test accuracy and a weighted F1-score of 93.51%. Class-wise analysis showed strong performance for melanocytic nevi and clinically relevant melanoma recall of 88.62%, while dermatofibroma precision remained low because of severe minority-class scarcity. The model was integrated with a Flask REST API and PySide6 desktop client, with a reported mean response time of 3.3 s.

The results demonstrate the technical feasibility of combining transfer learning, cloud-based inference, and an accessible interface. They do not establish clinical effectiveness. The system should therefore be regarded as a research decision-support prototype, not an autonomous diagnostic tool. External clinical validation, stronger data-

partition controls, uncertainty reporting, and evaluation on more diverse populations are required before practical medical use.

VII. REFERENCES

- [1] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.
- [2] N. C. F. Codella, D. Gutman, M. Celebi, et al., "Skin lesion analysis toward melanoma detection: A challenge at the 2017 International Symposium on Biomedical Imaging," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 2, pp. 501–512, 2019.
- [3] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," *International Conference on Engineering and Technology (ICET)*, pp. 1–6, 2017.
- [4] P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset: A large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Scientific Data*, vol. 5, no. 180161, 2018.
- [5] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [6] O. Russakovsky, J. Deng, H. Su, et al., "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
- [7] M. Abadi, P. Barham, J. Chen, et al., "TensorFlow: A system for large-scale machine learning," *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pp. 265–283, 2016.
- [8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations (ICLR)*, 2015.
- [9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.
- [10] Al Moaiad, Y., D., Alkhateeb, M., & Alokla, M. (2024). Enhancing Cybersecurity Practices in Nigerian Government Institutions an Analysis and Framework. *J. Electrical Systems*, 20(3), 5964-5967.
- [11] Al Moaiad, Y., D., Alkhateeb, M., & Alokla, M. (2024). Deep Analysis of Iris Print and Fingerprint for Detecting Drug Addicts. *J. Electrical Systems*, 20(3), 5639-5644.
- [12] Al Moaiad, Y., Alobed, M., & Alsakhnini, M. (2024). Python Solutions to Address Natural Language Challenges. *International Journal*, 10(3), 3594-3603.
- [13] Al Moaiad, Y., Alobed, M., Alsakhnini, M., & Momani, A. M. (2024). Challenges in natural Arabic language processing. *Edelweiss Applied Science and Technology*, 8(6), 4700-4705.
- [14] Al Moaiad, Y., Alobed, M., Alsakhnini, M., & Momani, A. M. (2024). Challenges in natural Arabic language processing. *Edelweiss Applied Science and Technology*, 8(6), 4700-4705.