Melanoma Skin Cancer Detection via Dermoscopic Images

Steven Wilson
Department of Mechanical Engineering
University of Ottawa
Ottawa, Canada
swils129@uottawa.ca

***Abstract*—Melanoma, is one of the most aggressive and life-threatening types of skin cancer due to its high propensity to metastasize. This project proposes the development of an automated diagnostic tool utilizing a Convolutional Neural Network (CNN) to enhance the early detection of melanoma through image analysis. By leveraging the openly accessible "Skin Cancer ISIC" dataset on Kaggle, which comprises of 2,357 annotated images, the system is designed to discriminate between malignant and benign lesions. Overall, this work seeks to demonstrate the potential of machine learning in medical diagnostics, ultimately by providing a solution that enhances early detection and intervention.**

Keywords—melanoma, Convolutional Neural Networks, machine learning, medical diagnostic

I. Introduction

Melanoma is a highly aggressive form of skin cancer that, despite it is the least common type of skin cancer, presents a significant health threat due to its rapid rate of metastasis if not identified or treated quickly. Early detection is critical, as survival rates drastically improve when intervention occurs at the initial stages of the disease. However, traditional diagnostic methods rely on visual examination and clinical judgement, which can slow down the diagnostic process of malignant lesions.

In recent years, advances in machine learning and medical imaging have opened the door to more reliable and efficient diagnostic tools. This project leverages the power of Convolutional Neural Networks (CNNs) to analyze dermoscopic images, aiming to distinguish between benign and malignant skin lesions. The core of this approach is based on the dataset sourced from the International Skin Imaging Collaboration (ISIC), which provided a foundation for training and validating the model.

The proposed system integrates modern programming frameworks and libraries—such as Python, TensorFlow, and other data analysis tools—to develop an end-to-end solution for early melanoma detection using a React app. By automating image assessment and interpretation, the project not only aspires to reduce diagnostic delays but also support healthcare professionals in making more informed clinical decisions. Furthermore, the eventual integration of this technology into a mobile application seeks to increase access to early diagnostic services, facilitating preliminary assessments.

Ultimately, this project is set to enhance current screening protocols and improve patient outcomes by providing technological innovation with medical expertise, representing a promising advancement in the field of medical diagnostic engineering.

II. Background

Historically, clinicians have relied on visual examination and manual dermoscopic analysis, using criteria such as asymmetry, border irregularity, color variation, and diameter (known as the ABCDE rule). While these methods provide a fundamental framework, they are inherently subjective and can be limited by the variability in lesion appearance and expertise of the clinician.

Over the past decades, the field of medical diagnostics has witnessed significant change with the integration of computational techniques that can process and learn from bigger quantities of data. Early efforts in computer-aided diagnosis employed traditional image processing and feature extraction methods. However, these approaches have difficulties with the complexity and subtlety of melanoma characteristics, leading to inconsistencies. Presently, with the development of CNNs there has been a transformative shift. CNNs are capable of learning hierarchical representations directly from raw image data, significantly outperforming conventional techniques in various image classification tasks.

III. Problem Definition

Melanoma is a critical health issue due to its high potential for rapid metastasis and significant reduced survival rates when diagnosis and treatment are delayed. The primary challenge lies in the limitations of conventional diagnostic methods.

The project addresses the following key problems:

A. Inconsistent Diagnosis

Traditional methods based on the ABCDE criteria can result in variable interpretations between clinicians. This increases the risk of false negatives and false positives.

B. Limited Early Detection

Melanoma's subtle characteristics in early stages often escapes detection by standard clinical assessments.

C. Scarcity and Accessible Diagnostic Tools

In many areas, the availability of expert dermatological evaluations is limited. This increases the need for reliable, automated diagnostic system capable of rapidly analyzing skin lesions.

D. Technological Constraints

Earlier computational methods used traditional image processing and feature extraction techniques that struggled with the variability and complexity of melanomas.

By developing a machine learning model based on Convolutional Neural Networks (CNNs), the project aims to create an objective, accurate, and scalable solution for early melanoma detection. This approach seeks to overcome the variability of human diagnosis and enhance access to fast, reliable skin cancer screening.

IV. Solution Design & Implementation

Our implementation begins with data loading and preprocessing. Raw dermoscopic images are downloaded from the ISIC dataset on Kaggle and loaded into our Google Colab notebook. Each image is resized (180 x 180 pixels), **pixel value normalization**, and data augmentation techniques, such as random rotations, flips and color adjustments, are performed to increase dataset diversity and reduce overfitting during training.

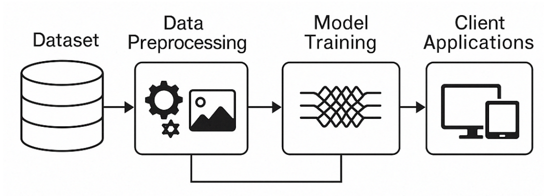

Figure I. High-level Architecture

At the core of the system is a Convolutional Neural Network (CNN), implemented with TensorFlow and Keras. The model comprises multiple convolutinoal blocks, consisting of convolution layers with ReLU activations, batch normalization, and max-pooling. Dropout layers are used to prevent overfitting, and a final dense layer with softmax activation outputs the probabilities for malignant versus benign classes.

Layer (type)	Output Shape	Param #
rescaling (Rescaling)	(None, 180, 180, 3)	0
conv2d (Conv2D)	(None, 178, 178, 32)	896
max_pooling2d (MaxPooling2D)	(None, 89, 89, 32)	0
conv2d_1 (Conv2D)	(None, 87, 87, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 43, 43, 64)	0
conv2d_2 (Conv2D)	(None, 41, 41, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 20, 20, 128)	0
dropout (Dropout)	(None, 20, 20, 128)	0
flatten (Flatten)	(None, 51200)	0
dense (Dense)	(None, 128)	6,553,728
dropout_1 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 9)	1,161

Table I. CNN Model Summary

The train loop incorporates callbacks for early stopping, learning rate reduction on plateau, and model checkpointing. Performance metrics are computed on a held-out validation set, and k-fold cross-validation is used to ensure robustness.

For deployment, the trained model is exported and wrapped in a Flask-based REST API. The API exposes an */analyze-image* endpoint that accepts image uploads, runs the machine learning model, and returns JSON-formatted results. This service is consumed by a React web frontend, enabling real-time lesion assessment.

V. Data

The “Skin Cancer ISIC” dataset is paramount for this study, it is a publicly available repository on Kaggle containing 2,357 dermoscopic images of both malignant and benign skin lesions, curated by the International Skin Imaging Collaboration. Each image comes with verified labels reflecting the ISIC’s diagnostic classifications, providing a reliable ground truth for supervised learning.

The images are classified among 9 categories:

- **Actinic Keratosis:** A rough, scaly patch on the skin caused by years of sun exposure.
- **Basal Cell Carcinoma:** A type of skin cancer that is most common but least aggressive, originating in the basal cells.
- **Dermatofibroma:** A common benign skin nodule that usually develops on the lower legs and is firm to touch.
- **Melanoma:** The most serious type of skin cancer, which forms from pigment-producing skin cells.
- **Nevus:** Commonly known as a mole, it is a benign proliferation of skin cells that produce pigment.
- **Pigmented Benign Keratosis:** A benign skin growth that appears as a small, dark spot, which can be mistaken for skin cancer but is harmless.
- **Seborrheic Keratosis:** A benign skin growth that looks like a waxy or wart-like bump, often appearing in middle age or later.
- **Squamous Cell Carcinoma:** A form of skin cancer that develops in the squamous cells, which compose most of the skin's upper layers (epidermis).
- **Vascular Lesion:** A generic term for lesions formed by blood vessels that may appear as red or purple patches on the skin, often benign.

The samples in Figure II. illustrate the diverse visual characteristics of our samples used to train and validate our CNN model.

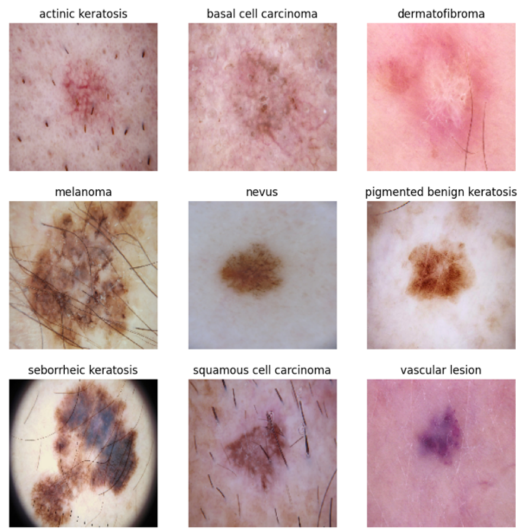

Figure II. Samples of 9 Classes of Malignant and Bening Skin Lesions

Data Partitioning

The image subsets are divided into varying numbers of images. Pigmented benign keratosis and melanoma have the highest number of images. Seborrheic keratosis and dermatofibroma have significantly fewer images compared to the largest classes.

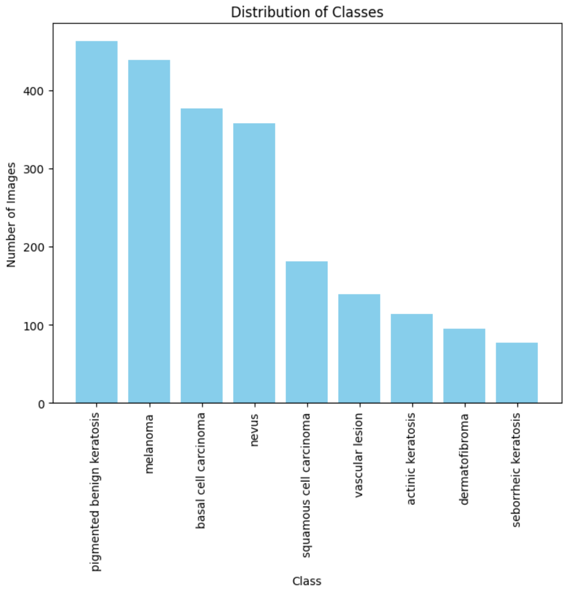

Figure III. Distribution of Classes

Data Balancing

From Figure III. it is apparent that the classes are imbalanced. The images are unevenly distributed, with some classes having significantly more samples than others.

We used the Augmentor library to apply class-specific rotations, flips and brightness variations, generating new images for underrepresented categories. By sampling a fixed number of augmented images per class, we equalized the training set size across all lesion types, mitigating class imbalance.

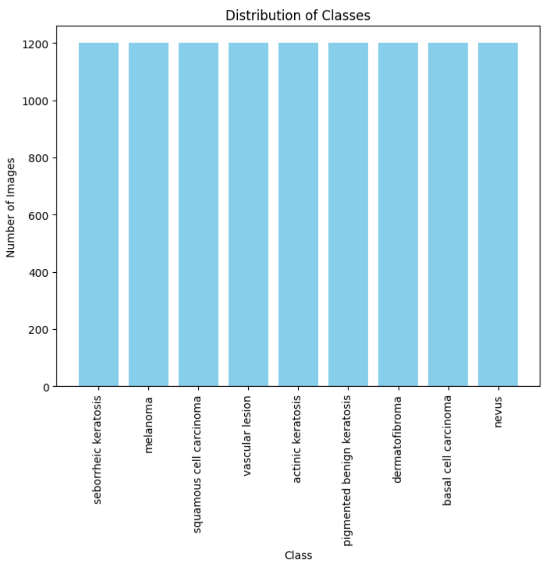

Figure IV. Balanced Distribution of Classes

VI. Training and Testing

A 20% split of the data is reserved for validation, one dataset is used for training and another for validation. Both datasets are created with a batch size of 32 and resized to 180 x 180 pixels, and labels are encoded in a categorical (one-hot) format. To maximize throughput and reduce bottlenecks during model training, both datasets are cached in memory, ensuring that data preparation and augmentation overlap efficiently with model execution.

The CNN is constructed as a sequential model using three main convolutional blocks with 32, 64, and 128 filters respectively, each followed by ReLU activation and max-pooling layers to progressively learn spatial hierarchies. There are two dropout layers (each with a rate of 0.5) are used to mitigate overfitting before flattening and passing through a fully connected layer of 128 units. The final output layer uses SoftMax activation across nine classes for multiclass classification.

The model is compiled with the Adam optimizer and categorical cross-entropy loss, and training is controlled via callbacks: model checkpoint saves the best-performing model based on validation accuracy, while early stopping halts training if validation loss fails to improve for five consecutive epochs. Over a maximum of 20 epochs, the fit method iteratively trains and evaluates the model.

VII. Results

Applying data augmentation using the Augmentor library significantly improved the model's generalization and robustness. Ideally, we would have sourced a better dataset that included more images and had balanced classes. However, by using synthetic data we were able to increase the number of images per class to 1200, the dataset became more balanced and diverse, helping the model learn more meaningful patterns rather than memorizing training data. This is reflected in the validation accuracy, which rose from around 0.60 to 0.91, and the reduce hap between training and validation curves, showing less overfitting and more stable learning behaviour.

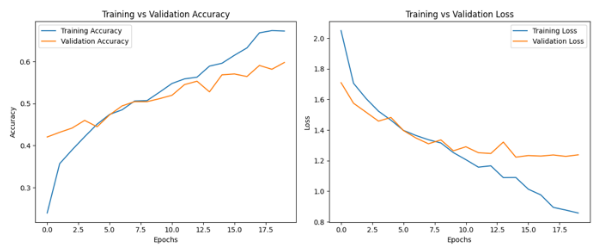

Figure V. Initial Accuracy and Loss

Additionally, increasing the number of training epochs from 20 to max 40 allowed the model to converge more effectively, while setting the final dropout layer to 0.4 added stronger regularization. This combination improved training stability and further minimized overfitting.

Figure VI. Improved Accuracy and Loss

Together, these changes contributed to a smoother training process, more accurate predictions, and a more reliable model across all classes. Nonetheless, underrepresented classes still pose challenges, the data imbalance was not completely solved with synthetic data and is suboptimal solution.

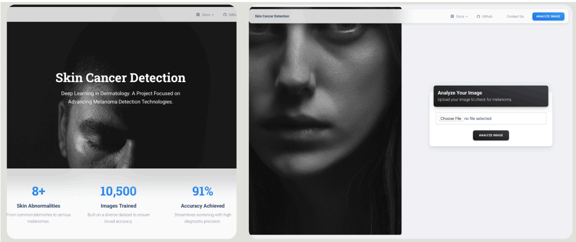

Figure VII. React App with Flask Backend

The React application was built to provide a seamless user experience across desktop and mobile devices. The main interface includes an image upload form that captures dermoscopic photos, displays a preview, and triggers an asynchronous POST request to the Flask backend via the */analyze-image* endpoint. State management with React hooks ensure real-time feedback, showing a loading spinner while the model processes the image and rendering the predicted class along with confidence scores. Error handling routines notify the users of unsupported file types or network failures.

On the backend, a Flask server load the trained CNN model at startup, ensuring that subsequent requests reuse the in-memory model for low-latency inference. The */analyze-image route* accepts form data, preprocesses incoming images to the required shape and scale, runs model prediction, and returns a JSON object containing predicted labels and probabilities.

For production deployment, we deployed both services separately on Heroku: The Flask API uses the Python buildpack with a Procfile and serves the trained model at the */analyze-image* endpoint, while the React frontend is built and served via the Node.js buildpack. The React app is configured with an environment variable pointing to the Flask app's URL, and whenever a user uploads an image it sends a POST request to */analyze-image* for inference. This setup lets the two Heroku apps scale and update independently yet work together seamlessly to process and display the model's predictions.

VIII. Discussion

The CNN-based diagnostic tool demonstrated promising performance in distinguishing between malignant and benign lesion, achieving high classification accuracy and robust sensitivity among the classes that were most represented, with poor results on those underrepresented. These results underscore the effectiveness of deep learning approaches in capturing subtle dermoscopic features that are often challenging to discern through manual examination. By leveraging extensive data augmentation, the model generalized well to unseen samples, suggesting its potential utility as adjunct to clinical assessment.

Despite these encouraging outcomes, several limitations warrant consideration. First, the reliance on the ISIC dataset, while comprehensive, may not fully represent the diversity of skin types, lighting conditions, and image acquisition protocols encountered in real-world settings. Moreover, certain classes with fewer images remained prone to misclassification, even after augmentation, highlighting the need for larger, more balanced datasets. Finally, while early stopping and dropout helped mitigate overfitting, further regularization techniques or ensemble methods might improve stability for underrepresented lesion categories.

Looking ahead, integrating explainability mechanisms, such as Grad-CAM heatmaps, to visualize model decision processes could enhance clinician trust and facilitate model refinement. Expanding the dataset through multi-institutional datasets would address generalizability concerns, while clinical validation studies are essential to asses real-world impact on the proposed diagnostic tool. Ultimately, embedding this system into an e-hospital tool would allow early access to melanoma screening tools, contributing to improved patient outcomes through faster and more reliable detection.

IX. Conclusion

The work presented here establishes a comprehensive, end-to-end pipeline for melanoma detection, from dataset curation and augmentation through model training and real-world deployment. By systematically integrating data preprocessing, a custom CNN architecture, and automated performance tracking, we delivered a reproducible framework capable of distinguishing malignant from benign lesions. The successful export of the trained model into a lightweight Flask API and deployment in Heroku demonstrate not only technical feasibility but also the potential for seamless integration into existing healthcare infrastructures.

Beyond technical achievements, this project underscores the value of combining open datasets with modern machine learning practices to address pressing clinical challenges. The modular design allows for straightforward updates, such as changing machine learning models, ensuring the system remains adaptable as imaging standards evolve. Looking forwards, collaboration with clinicians for prospective validation and user-experience testing will be key steps towards translating this prototype into a diagnostic aid, ultimately contributing to earlier interventions and improved patient outcomes.

References

[1] Public Health Agency of Canada. (n.d.). Melanoma Skin Cancer. Retrieved 2025, from https://www.canada.ca/en/public-health/services/chronic-diseases/cancer/melanoma-skin-cancer.html

[2] American Cancer Society. (n.d.). What Is Melanoma Skin Cancer? Retrieved 2025, from https://www.cancer.org/cancer/types/melanoma-skin-cancer/about/what-is-melanoma.html

[3] Skin Cancer Foundation. (n.d.). Melanoma Warning Signs. Retrieved 2025, from https://www.skincancer.org/skin-cancer-information/melanoma/melanoma-warning-signs-and-images/

[4] IEEE. (2023). Design of a Skin Cancer Detection Classification with Python GUI and Tensorflow. 2023 10th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Palembang, Indonesia, pp. 38-42, doi: 10.1109/EECSI59885.2023.10295620. Retrieved from https://ieeexplore.ieee.org/document/10295620

[5] Muñoz, L. (n.d.). Python in Medical Imaging: A ROI Extraction Example. Retrieved 2025, from https://medium.com/@lamunozs/python-in-medical-imaging-a-roi-extraction-example-0f315336dd6d

[6] OpenAI. (2022, November 30). Introducing ChatGPT. Retrieved April, 2025, from https://openai.com/index/chatgpt/ — used by the author to assist in code development and report drafting

[7] Analytics Vidhya. (2020). Image Classification Using CNN. Retrieved 2025, from https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/

[8] Kriplani, A. (n.d.). Melanoma Skin Cancer Detection. Retrieved 2025, from https://github.com/akashkriplani/melanoma-skin-cancer-detection/tree/main?tab=readme-ov-file#model-summary