SKIN DISEASE DETECTION PROGRES REPORT

Jacky Li
Student# 1011271678
jakkii.li@mail.utoronto.ca

Jordan Cui
Student# 1011026916
jordan.cui@mail.utoronto.ca

Lawrence Ding
Student# 1011439025
larryzm.ding@mail.utoronto.ca

Soham Shorey
Student# 1010845169
soham.shorey@mail.utoronto.ca

ABSTRACT

This report outlines our progress on developing a deep-learning based system to classify dermatological conditions from skin images. We aim to create a diagnostic tool that allows patients to detect potentially severe skin conditions early and aids healthcare professionals in diagnosis. It includes our individual contributions to the project and our delegated next tasks to complete our project on time. We have combined the HAM10000, BCN20000, and PAD-UFES-20 datasets with 9 standardized class labels. For reference, we trained a baseline KNN model with a ResNet-18 feature embedder achieving a 56.3% accuracy. Our best current CNN classifier uses a ResNet-34 feature embedder backbone with a custom linear classifier to achieve 81.9% accuracy. We outline the training strategies, hyperparameters, and architecture we used to train the model, and potential future improvements in data preprocessing and model architecture. —-Total Pages: 10

1 BRIEF PROJECT DESCRIPTION

Our team is developing a deep learning model to detect and classify common and rare skin conditions from dermoscopic and clinical images. The project focuses on distinguishing between various harmful and harmless skin conditions including nevus, melanoma, keratosis, basal cell carcinoma, squamous cell carcinoma, and other dermatological conditions to assist both patients and healthcare professionals in early detection and diagnosis.

The motivation behind our project is the fact that skin conditions affect over a third of the global population (Li et al., 2024). In Canada, 20% of people experience acne, 1 million have psoriasis, and melanoma rates have risen by over 2% annually since the 1980s (Canadian Dermatology Association). In 2013, 84 million Americans, or a quarter of their population, saw a doctor for skin issues (American Academy of Dermatology). A tool that classifies skin diseases from images could ease the burden on dermatologic healthcare by supporting early detection and aiding diagnosis.

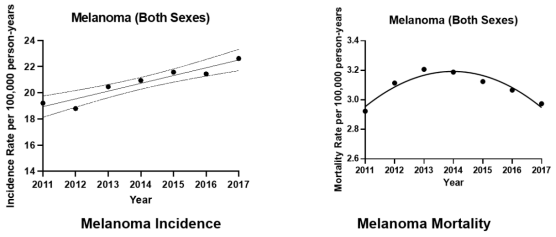

Figure 1: Melanoma Incidence and Mortality Rate per 100,000 persons-year vs. Year (McGill Newsroom, 2022).

Figure 1 demonstrates that the incidence of melanoma has been increasing steadily each year; however, it also shows that the mortality rate is decreasing relative to the incidence rate, likely due to advances in treatment techniques. This suggests that early diagnosis of melanoma is critical, as the disease is becoming more prevalent while also more treatable when caught early. These trends motivate us to build a disease detection tool to help individuals identify potential signs of melanoma sooner, encouraging timely medical attention and reducing the risk of severe outcomes.

The goal of our project is to create an automated skin disease detection system that can provide preliminary screening for various skin conditions, assist healthcare professionals in diagnostic decision-making, encourage timely medical consultation for potentially serious conditions, and ultimately reduce the burden on healthcare systems. The system takes dermoscopic images as input and outputs predicted skin disease classifications along with recommended next steps for patients.

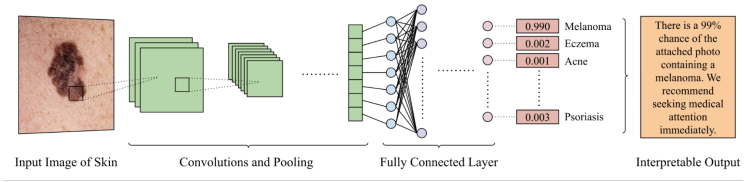

Figure 2: Visualization of the Proposed Model Architecture with Input and Output. The use of CNNs can be inferred from the figure.

Deep learning models, particularly convolutional neural networks (CNNs), are the most appropriate approach for this image classification task. CNNs use trainable filters to extract high-level spatial patterns in images. The use of shared weights also increases the computational efficiency of CNNs compared to fully connected alternatives. Since skin disease detection requires processing large amounts of image data, learning complex visual patterns where subtle differences can affect diagnosis, and using datasets with expert-labelled diseases, using a CNN model trained via supervised learning would yield the most effective classifier.

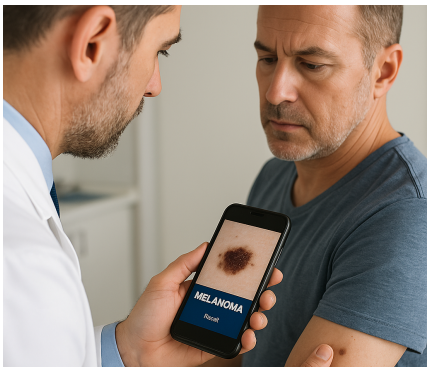

Figure 3: AI-generated image of how we envision the system to be used.

Lastly, we find this project interesting because it lies at the intersection of biomedical and machine learning engineering, two fields we are both passionate about and hope to explore further. Once completed, we plan to deploy the system as a publicly accessible web application. This will allow medical professionals to test the prototype in real-world settings and provide feedback, while also offering individuals without a diagnosis a chance to perform an initial screening of their skin condition. By making the tool accessible, we hope to raise awareness about early detection, improve access to preliminary assessment, and ultimately contribute to better health outcomes through the responsible use of AI in healthcare.

2 Individual Contributions and Responsibilities

Our team collaborates through GitHub for code management, Discord for daily communication, and a shared Google Sheet to track tasks and deadlines. We hold regular meetings to review progress and assign new responsibilities, while maintaining continuous communication via Discord. Although tasks are assigned to specific individuals, team members frequently assist one another. This approach helps ensure timely completion and provides redundancy in case someone faces unforeseen challenges. The individual contributions and task assignments listed below reflect those on our Google Sheet; however, in practice, our work is highly collaborative and often overlaps beyond these formal assignments.

Title	Assignee	Status	Description	Comments	Deadline
Repository Setup and Workflow	Jacky Li	Closed	Initialize the GitHub repository, set up the develop	Repository created, branch protection rules set, workflow docum	30-06
Model Research and Literature Review	Jacky Li	Closed	Conduct literature review and select appropriate b	Reviewed recent papers and selected SVM as baseline and CNN	30-06
Baseline Model Implementation and Evaluation	Lawrence Ding	Closed	Implement the baseline model and perform initial	Baseline models implemented and evaluated; results document	01-07
Data Collection and Organization	Soham Shorey	Closed	Download, merge, and organize all required datas	All datasets collected, merged, and organized for preprocessing	01-07
Data Preprocessing Pipeline	Soham Shorey	Closed	Develop and execute the data preprocessing pipel	Preprocessing pipeline complete and reproducible.	02-07
Dataset Statistics and Visualization	Soham Shorey	Closed	Generate statistics and visualizations to assess cl	Class distribution and sample images visualized; figures ready f	02-07
Primary Model Implementation and Initial Evaluati	Soham Shorey	Closed	Implement the primary model and perform initial e	Primary model implemented and initial results obtained.	03-07
Documentation and Results Organization	Jordan Cui	Closed	Organize results, maintain project documentation,	Results organized and documentation up to date.	03-07
Evaluation Metrics and Visualization	Jordan Cui	Closed	Add evaluation metrics and visualizations for moc	Evaluation metrics and visualizations implemented.	04-07
Dataset Integration and Preprocessing Optimizati	Lawrence Ding	Closed	Assist with integrating datasets and optimizing pr	Preprocessing optimized and datasets fully integrated.	04-07
Data Splitting and Augmentation Scripts	Jacky Li	Closed	Develop and maintain scripts for splitting data int	Data splitting and augmentation scripts complete.	05-07
Model Training, Validation, and Hyperparameter Ti	Lawrence Ding	Closed	Support model training, validation, and perform hy	Model training and tuning completed for initial experiments.	05-07
Update and Finalize Data Visualizations	Soham Shorey	Open	Update and finalize all data visualizations and pre	Figures and visualizations will be updated and prepared for subr	12-07
Support Integration and Comparison of Advanced	Jacky Li	Open	Assist with integrating and comparing advanced r	Integration and comparative analysis of advanced models will b	13-07
Finalize Model Training and Evaluation	Jordan Cui	Open	Finalize model training scripts, update evaluation	Training will be completed and all evaluation outputs updated.	14-07
Prepare Scripts for Final Testing on Unseen Data	Lawrence Ding	Open	Develop scripts for evaluating the model on new, u	Scripts will be prepared and results analyzed for unseen data	15-07

Figure 4: An image of the Google Sheet we used to track, assign, and efficiently take on tasks.

Jacky Li

- Established the GitHub repository and development workflow.
- Led model research, including literature review on potential baseline models and transfer learning models like ResNet and EfficientNet for classifiers.
- Developed and maintained the data splitting and augmentation scripts.
- Next steps: Support integration and comparison of advanced model architectures; assist with final model selection and analysis.

Soham Shorey

- Led data collection and processing including downloading and organizing all datasets.
- Developed and executed the data preprocessing pipeline, including label standardization and image resizing.
- Generated dataset statistics and visualizations to assess class balance.
- Implemented the primary model and performed initial evaluation.
- Next steps: Update and finalize data visualizations; prepare cleaned data examples for the report; assist with model retraining as needed.

Jordan Cui

- Led the documentation and organization of results into this progress report.
- Assisted in the implementation of the baseline model.
- Added evaluation metrics and visualizations for model performance.
- Next steps: Finalize model training scripts; update evaluation metrics; help compile results and figures for the final report.

Lawrence Ding

- Researched typical baseline models used for medical imaging.
- Implemented the baseline model and performed initial evaluation.
- Assisted with dataset integration and preprocessing optimization.
- Supported model training, validation, and hyperparameter tuning.
- Next steps: Prepare scripts for final testing on unseen data; analyze and summarize model results; contribute to the final report and presentation.

3 Notable Contributions

3.1 Data Processing

We combined three publicly available skin lesion datasets to create a comprehensive and diverse training set that offers robust coverage of various skin conditions and imaging modalities.

3.1.1 Data Sources

Our data was sourced from three datasets: BCN20000, HAM10000, and PAD-UFES-20.

- The BCN20000 dataset contains 18,946 dermoscopic images sourced from the Hospital Clínic de Barcelona corresponding to 8 skin conditions (International Skin Imaging Collaboration (ISIC), 2018). The dataset was sourced using the ISIC archive using a metadata CSV to extract only the BCN20000 images.
- The HAM10000 dataset contains 10,015 dermatoscopic images from 7 different classes sourced from the Department of Dermatology at the Medical University of Vienna, Austria, and the skin cancer practice of Cliff Rosendahl in Queensland, Australia over 20 years (Tschandl et al., 2018). The dataset was sourced from Kaggle and stored in 2 separate folders which were combined.
- The PAD-UFES-20 dataset contains 2,298 labelled images of dermatological conditions sourced by the Federal University of Espírito Santo (UFES) in Brazil. The dataset was sourced from Kaggle, which required merging images from three separate folders into a unified directory structure (Mahdavi, 2022).

3.1.2 Data Cleaning and Processing Steps

The unification of these three datasets required multiple preprocessing steps.

Since the BCN20000 dataset is not publicly separated from the broader ISIC archive, we used a metadata CSV file to isolate and include only the BCN20000-labeled images. For the HAM10000 and PAD-UFES-20 datasets, we consolidated images from all respective folders into unified directory structures.

Label information was extracted from the diagnosis_3 field in the BCN20000 metadata, the dx field in the HAM10000 metadata, and the diagnostic field in the PAD-UFES-20 CSV. We then standardized all labels into nine unified classes: nevus, melanoma, actinic (pre-cancerous) keratosis, basal cell carcinoma (BCC), squamous cell carcinoma (SCC), lentigo, vascular lesion, dermatofibroma, and keratosis.

We first used the ISIC API along with the BCN20000_metadata.csv file to extract only the BCN20000 images. We downloaded the HAM10000 and PAD-UFES-20 datasets from Kaggle.

Before consolidation, we resized all images to 512×512 using the PIL.Image library for consistent input sizes. Then, we organized all files into unified directories based on standardized label categories. For example, images labelled nevi were mapped to the *nevus* folder.

Approximately 1,000 images labeled as scars or left unlabeled were grouped under *other* and excluded from training. We considered applying a hair removal filter to reduce skin image noise but decided to gauge model performance before its application (Kalpana et al., 2025).

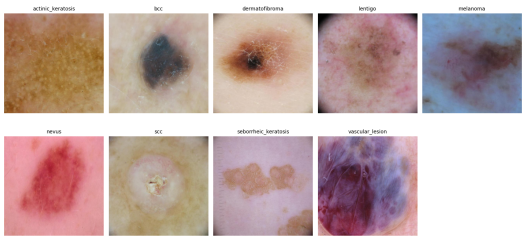

Figure 5: Sample resized and labelled images from each of the 9 classes in the combined dataset.

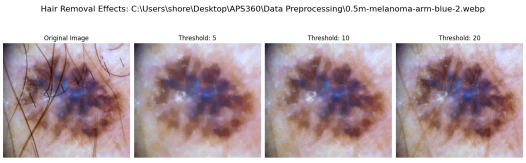

Figure 6: Effect of hair removal filter at thresholds 5, 10, and 15 on sample melanoma image. Filter removes hair and measurement lines from the image.

3.1.3 Dataset Statistics

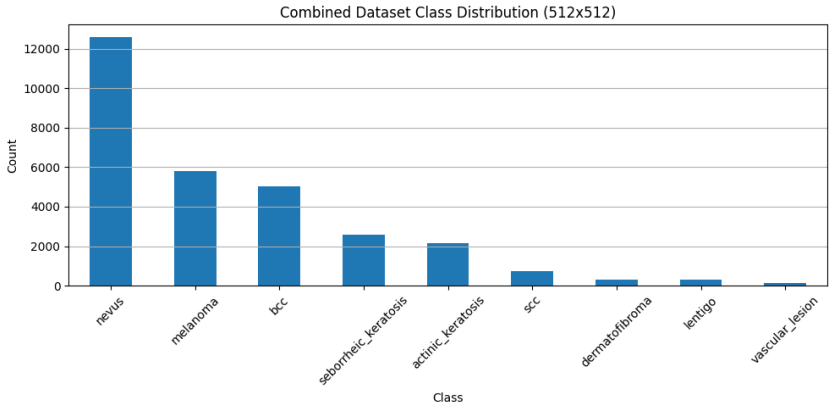

Figure 7: Class distribution of combined dataset.

Dataset is imbalanced with nevus as the majority class.

We split the processed dataset into 70% training, 15% validation, and 15% test sets to ensure robust model training and evaluation. Each split maintains the same class distribution as visualized in the dataset summary above. However, the dataset remains significantly imbalanced, with the *nevus* class accounting for approximately 42.5% of all samples.

In the future, if the class imbalance remains an issue, we may add more samples of underrepresented classes using datasets like DERM12345 and reputable dermatological image sources like DermNet.

We created class distribution visualizations using matplotlib to identify label imbalances. These bar charts allow us to verify data quality and plan appropriate sampling and augmentation strategies for training.

Table 1: Class counts after preprocessing

Class	**Nevus**	**Melanoma**	**BCC**	**Benign keratosis**	**Actinic keratosis**
Count	12,596	5,801	5,035	2,602	2,145
Class	**SCC**	**Dermatofibroma**	**Lentigo**	**Vascular lesions**	
Count	751	283	283	142	
Total: 29,638					

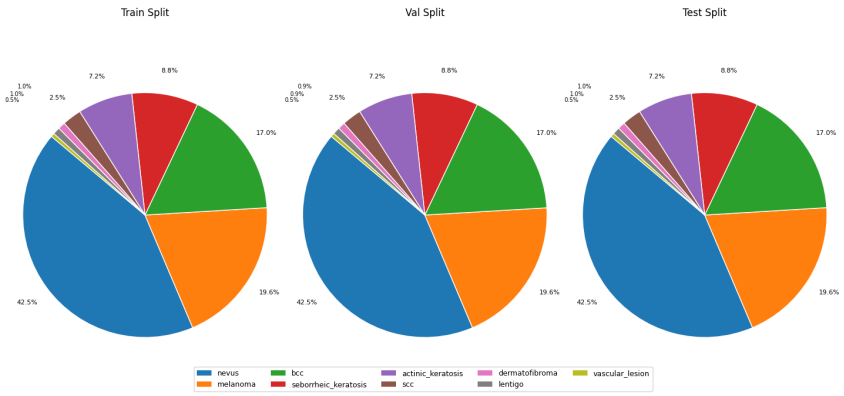

Figure 8: Even class distribution between training, validation, and testing datasets.

3.1.4 Challenges

We encountered several significant challenges during data processing.

- Since HAM10000 and BCN20000 images are present in the ISIC challenge datasets, we identified duplicate images as a potential issue. To ensure neither dataset overlapped, we compared ISIC IDs of HAM10000 and BCN20000 images to ensure no duplicate images were present in our dataset.
- BCN20000 images are not publicly available as a separate dataset. As such, BCN20000 images were extracted from the ISIC20000 dataset using a metadata CSV file.
- Label inconsistencies between datasets required building a comprehensive mapping system to consolidate similar labels into unified classes.

3.1.5 Plan for Final Testing on Unseen Data

Our final model will be evaluated on the dedicated test data containing 4,446 images unseen by the model during training. We will also test our model on completely separate datasets like DERM12345 and Fitzpatrick17k if not previously used in our training process. Ideally, we aim to manually collect dermoscopic images from open sources or clinical repositories to ensure completely unseen test data to best evaluate the generalization capabilities of our model.

3.2 Baseline Model

For our baseline model, we initially considered using simpler SVM or KNN algorithms to compare our model to. However, due to the complexity of skin detection, a standalone SVM or KNN is unlikely to outperform random chance. As such, we chose to first extract the features of the images using a pre-trained model (in this case the ResNet-18) and train a KNN model using the extracted features. In short, we imported the ResNet-18 and removed the fully-connected layers, only preserving the convolutional and average pool layers, and used it to extract the features of all the images in

our dataset. The features were then used as data to fit a KNN model. The final test accuracy of this baseline model was only 56.32% with the F1-scores not being acceptable for real-world applications. However, this model performs better than chance (11.11%), and serves as a suitable baseline model for this project.

Validation Accuracy: 55.38%
Test Accuracy: 56.32%

	precision	recall	f1-score	support
actinic_keratosis	0.31	0.41	0.35	322
bcc	0.43	0.60	0.50	756
dermatofibroma	0.00	0.00	0.00	42
lentigo	0.00	0.00	0.00	42
melanoma	0.56	0.37	0.45	870
nevus	0.70	0.83	0.76	1889
scc	0.07	0.01	0.02	113
seborrheic_keratosis	0.22	0.05	0.08	390
vascular_lesion	1.00	0.09	0.17	22
accuracy			0.56	4446
macro avg	0.36	0.26	0.26	4446
weighted avg	0.53	0.56	0.53	4446

Figure 9: Accuracy, precision, recall, and F1-score of the baseline model. This model struggles with the imbalanced dataset, with a high F1-score for the most common class, but completely fails at identifying the less common classes.

3.3 Primary Model

To develop an effective skin condition classifier, we tested several models with various CNN backbones, image transformations, classifier architectures, and model complexities. Each model leveraged transfer learning using ImageNet-pretrained feature extractors such as ResNet, EfficientNet, and MobileNet, combined with different custom classifier heads.

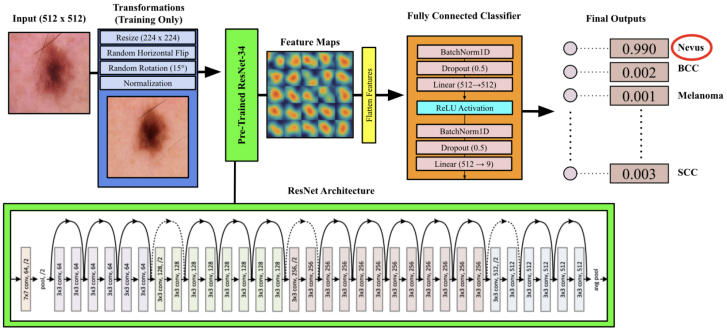

Figure 10: Architecture of the best-performing model. The model uses a ResNet-34 convolutional backbone as a feature extractor, followed by a custom two-layer fully connected classifier. ResNet-34 includes multiple convolutional blocks with residual connections and average pooling. Data transformations are applied to input images during training to improve generalization.

Currently, our best performing classifier exhibits a validation accuracy of 81.95%, precision of 79.82%, recall of 71.34%, and F1 of 74.91%. This model outperforms the baseline model and may still improve with further training. This model leverages a pre-trained ResNet-34 backbone for feature extraction, followed by a two-layer fully connected classifier. The classifier includes ReLU activation between the layers and applies batch normalization and dropout (with a rate of 0.5) before each linear transformation. The first linear layer receives the flattened output features from the ResNet-34 and projects them into a 512-dimensional hidden space. The final linear layer maps these features to 9 output units, corresponding to the number of skin lesion classes.

Table 2: Best model hyperparameter values

Hyperparameter	Value	Description
NUM_CLASSES	9	Number of output classes
BATCH_SIZE	64	Samples per training batch
EPOCHS	60	Maximum training epochs
LEARNING_RATE	0.0003	Initial learning rate
PATIENCE	10	Epochs to wait before early stopping
WEIGHT_DECAY	1e-4	L2 regularization strength
LABEL_SMOOTHING	0.1	Softens target labels to reduce overconfidence
BACKBONE	ResNet34 (pretrained)	CNN used for feature extraction
DROPOUT_RATES	0.5 (applied twice)	Dropout to reduce overfitting in classifier head
HIDDEN_LAYER_UNITS	512	Units in intermediate fully connected layer
IMAGE_SIZE	224 × 224	Input image resolution
AUGMENTATIONS	Flip, Rotation	Data augmentation for robustness
NORMALIZATION_MEAN	[0.485, 0.456, 0.406]	Standard normalization mean
NORMALIZATION_STD	[0.229, 0.224, 0.225]	Standard normalization std dev
SAMPLER	WeightedRandomSampler	Balances class frequencies during training
SCHEDULER	ReduceLROnPlateau (factor=0.5)	Reduces LR when validation accuracy plateaus
OPTIMIZER	Adam	Optimizer used
CRITERION	CrossEntropyLoss (smoothing=0.1)	Loss function used

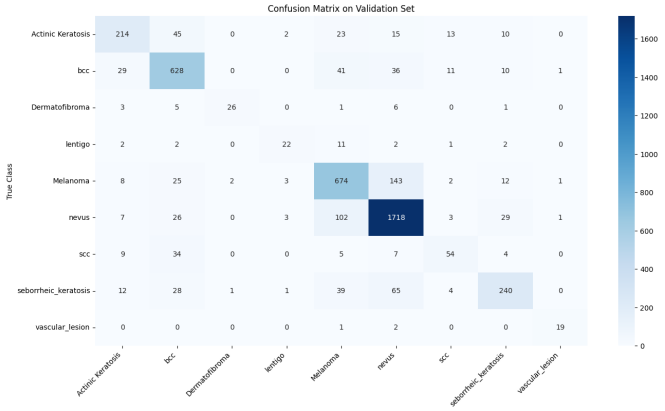

Figure 11: Confusion matrix of best model. The model predicts most classes effectively. However, it continues to confuse melanoma with nevus, actinic keratosis with BCC, and seborrheic keratosis with nevus, and SCC with BCC. In general, the model must reduce the chance of dangerous conditions like melanoma from being misclassified as benign conditions like nevi.

The 512 × 512 images were resized and augmented to 224 × 224 images with random rotations and reflections to reduce model complexity. The model is trained using the Adam optimizer with an initial learning rate of 0.0003 and a weight decay of 0.0001. Cross-entropy loss with label smoothing (ε = 0.1) is used to prevent overconfident predictions. A learning rate scheduler (ReduceLROnPlateau) monitors validation accuracy and reduces the learning rate by a factor of 0.5 if no improvement is seen for 2 consecutive epochs. We trained our model for 45 epochs using early stopping with patience of 10 epochs. Due to the data imbalance, precision, recall, and F1 score were measured in addition to loss and accuracy.

3.3.1 Challenges

Each major training challenge and some proposed or implemented solution(s) are listed below.

1. Dataset imbalance

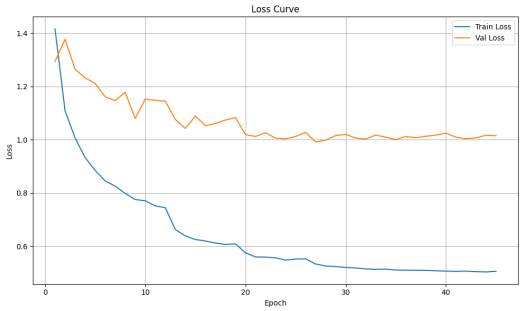

Figure 12: Best model validation and training loss per epoch. Train loss is steadily lower than validation loss due to the effects of WeightedRandomSampler, which makes the class distribution seen by the model in training different than in validation.

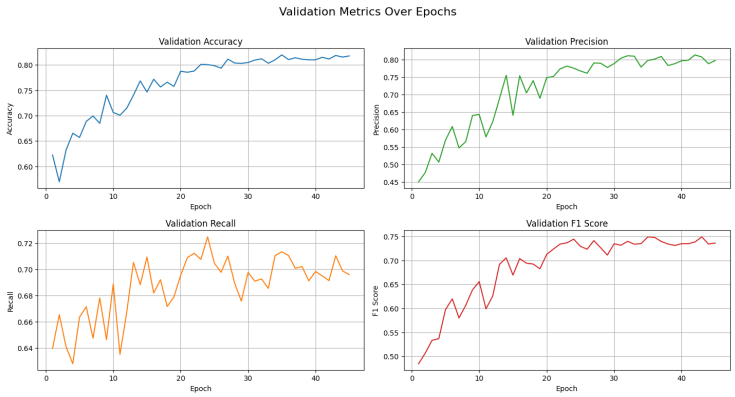

Figure 13: Best model validation accuracy, precision, recall, and F1 per epoch. Validation metrics plateau after epoch 35.

The imbalance of our combined dataset led our initial models to overpredict majority classes like nevi and melanoma and ignore minority classes like vascular lesion. The models exhibited high accuracy but low recall and precision, indicating poor generalization. To address the effects of dataset imbalance, we applied WeightedRandomSampler in training to sample classes inversely proportional to their frequency in the training dataset. This step balances each batch seen by the model during training so the model does not learn to predict only the majority class. Additionally, we randomly augmented images using random rotation and horizontal flipping, allowing the model to train on more unique minority class images.

2. GPU Usage and Training Time

To train our model we used a NVIDIA GeForce RTX 4050 GPU with a VRAM of 6 GB. Using 512 × 512 images and training large feature extractors like ResNet-50 and MobileNet required small batch sizes or was infeasible given our time frame. To increase training speed and feasibility, we resized the images to 224 × 224 before input into the model and unfroze only the last several layers in the feature extractors or opted for less complicated models. For example, our best model used ResNet-34 instead of the more complicated ResNet-50. This step also allowed us to use a batch size of 64 rather than 32 or 16 to smoothen training.

3. Overfitting to Training Data

Training on larger models like ResNet-50 led to sharp declines in training loss with little plateauing validation loss, indicating overfitting. To prevent overfitting, we applied two dropout layers of 0.5 to prevent the model from relying too heavily on specific neural pathways, used the smaller ResNet-34 and froze layers to reduce model size, and weight decay of 0.0001 to prevent overreliance on large

weights. Data augmentation of the training set also reduced overfitting by preventing the model from memorizing images in their default orientations.

REFERENCES

American Academy of Dermatology. Burden of skin disease. URL https://www.aad.org/member/clinical-quality/clinical-care/bsd. Accessed: Jul. 9th, 2025.

Canadian Dermatology Association. Skin conditions. URL https://dermatology.ca/public-patients/diseases-conditions/skin-conditions/. Accessed: Jul. 9th, 2025.

International Skin Imaging Collaboration (ISIC). Isic archive - collection 249, 2018. URL https://api.isic-archive.com/collections/249/.

B. Kalpana, A. Senthilselvi, S. Muruganandam, and S. V. Kumar. Enhancing skin disease diagnosis: Light gbm-dms algorithm for accurate image classification. *Cognitive Computation*, 17(3):1–18, 2025.

Q. Li, M. T. Patrick, S. Sreeskandarajan, J. Kang, J. M. Kahlenberg, J. E. Gudjonsson, Z. He, and L. C. Tsoi. Large-scale epidemiological analysis of common skin diseases to identify shared and unique comorbidities and demographic factors. *Frontiers in Immunology*, 14, January 2024. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC10800546/.

M. Mahdavi. Skin cancer dataset, 2022. URL https://www.kaggle.com/datasets/mahdavi1202/skin-cancer.

McGill Newsroom. Melanoma map shows skin cancer is on the rise in canada. *Health e-News (McGill University)*, June 2022. URL https://healthenews.mcgill.ca/melanoma-map-shows-skin-cancer-is-on-the-rise-in-canada/. Accessed: Jul. 9th, 2025.

P. Tschandl, C. Rosendahl, and H. Kittler. Ham10000 dataset: A large collection of multi-source dermatoscopic images of common pigmented skin lesions, 2018. URL https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000.