DermaVision: Skin Lesion Classification Using Deep Learning

Hrithik Puri
Northeastern University
puri.hr@northeastern.edu

Shubham Pandey
Northeastern University
pandya.shu@northeastern.edu

Atharva Avinash Gosavi
Northeastern University
gosavi.at@northeastern.edu

Skin cancer, particularly melanoma, poses significant diagnostic challenges due to its severity and prevalence. This research focuses on enhancing skin lesion classification through the development of a convolutional neural network (CNN) model, leveraging the HAM10000 dataset, which includes 10,015 dermoscopic images across seven diagnostic categories. The project employs state-of-the-art CNN architectures such as CNN, Custom CNN with additional layers and ResNet50 incorporating transfer learning to improve accuracy and generalizability. Key stages include data preprocessing, model training, and evaluation using metrics like accuracy, precision, recall, and F1-score. The goal is to advance early detection and diagnostic precision in skin cancer, contributing to better patient outcomes through robust and accurate deep learning models.

Keywords—Skin Lesion, CNN, ResNet50, ReLu, Softmax, Cross Entropy, Melanoma, Dermoscopic Images

INTRODUCTION

Skin cancer is one of the most prevalent cancers globally, with millions of new cases diagnosed annually. It manifests in various forms, including basal cell carcinoma, squamous cell carcinoma, and melanoma. Among these, melanoma is particularly dangerous due to its rapid progression and high mortality rate if not detected early. Skin lesions, which can be benign or malignant, often require accurate classification to guide treatment. Dermoscopy, a non-invasive imaging technique, improves diagnostic accuracy by providing enhanced visualization of skin lesions. However, even with dermoscopy, diagnosis is subject to human error and variability, underscoring the need for more reliable methods. Traditional methods for diagnosing skin cancer include visual inspection by dermatologists and dermoscopic analysis, which, while effective, depend heavily on the expertise of the clinician and are prone to inconsistencies.

Motivated by the need to improve diagnostic accuracy and early detection of skin cancer, our project aims to leverage the capabilities of deep learning, particularly convolutional neural networks (CNNs), to classify skin lesions accurately. Recent advancements in CNN architectures, such as ResNet50 and custom CNN, have demonstrated remarkable success in image classification tasks across various domains, including medical imaging. These models utilize large annotated datasets and employ techniques like transfer learning and data augmentation to enhance performance.

However, achieving high diagnostic accuracy and generalizability remains challenging due to the variability in skin lesion appearances and patient demographics. By developing a robust deep learning model, we aim to address these challenges and contribute to better patient outcomes through improved diagnostic precision.

Our approach involves several key stages: data preprocessing, model selection and development, training, and evaluation. Initially, we will preprocess the images from the HAM10000 dataset, resizing and normalizing them for CNN input. We then selected a suitable CNN architecture, potentially employing pre-trained models for transfer learning to leverage existing knowledge. The model's performance will be evaluated using metrics like accuracy, precision, recall, and F1-score.

The HAM10000 dataset, a comprehensive collection of 10,015 dermoscopic images of pigmented lesions, will be utilized for this project. This dataset includes seven diagnostic categories, providing a diverse and well-annotated resource for training and evaluating deep learning models. The availability of such a rich dataset is crucial for developing a model capable of generalizing across different lesion types and patient populations.

The goal of this project is to develop a CNN-based model that significantly enhances the accuracy and reliability of skin lesion classification. By leveraging deep learning techniques and the extensive HAM10000 dataset, we aim to improve early detection and diagnostic precision for skin cancer, ultimately leading to better patient outcomes and advancements in medical image analysis.

DATASET

The dataset used for this research consists of dermoscopic images obtained from individuals with different skin diseases [1].

The dataset is divided into training, validation, and test sets, with the following dimensions:

Training set (X_train): 5187 images with corresponding disease labels.

Validation set (X_val): 1556 images with corresponding disease labels.

Test set (X_test): 667 images with corresponding disease labels.

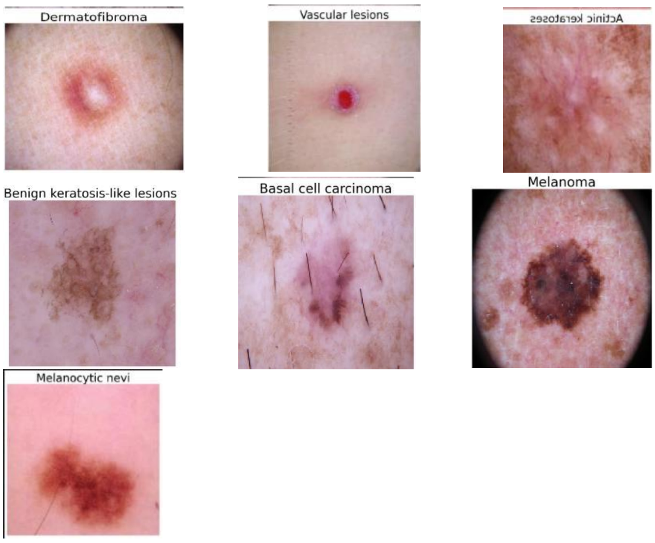


Fig. 1 Labels in the dataset

BACKGROUND

Skin diseases, ranging from benign conditions to malignant cancers, represent a significant portion of healthcare concerns worldwide. Among these, skin cancer remains one of the most prevalent and rapidly increasing forms of cancer, with melanoma being the deadliest. Early and accurate detection of skin lesions is critical for effective treatment and improved patient outcomes.

Recent developments in artificial intelligence (AI) and machine learning (ML) have shown substantial potential in diverse medical applications, particularly in image recognition and diagnostics. Among deep learning techniques, Convolutional Neural Networks (CNNs) have emerged as highly effective in analyzing and classifying intricate image data. By learning spatial hierarchies of features through backpropagation, CNNs are exceptionally well-suited for medical image analysis, which demands high accuracy and precision.

In dermatology, Convolutional Neural Networks (CNNs) can be trained to distinguish between various types of skin lesions, such as benign moles, malignant melanomas, and other skin conditions, through the analysis of dermoscopic images. These networks utilize extensive datasets of labeled images to identify complex patterns and features that may not be visible to the human eye. The application of CNNs for skin lesion classification holds significant potential to enhance the diagnostic capabilities of dermatologists, reduce diagnostic times, and improve the accuracy of early skin cancer detection.

This research aims to develop and evaluate a Convolutional Neural Network (CNN)-based model for skin lesion classification. Utilizing a comprehensive dataset of dermoscopic images, the study seeks to demonstrate the effectiveness of CNNs in distinguishing between various skin conditions. The objective is to contribute to the expanding field of AI-driven medical diagnostics and to advance more accessible, efficient, and reliable methods for skin disease screening.

APPROACHES

Our methodology focuses on designing and deploying a convolutional neural network (CNN) architecture specifically customized for the classification of skin diseases through the analysis of skin lesion images. This section offers an in-depth explanation of our techniques, encompassing data preprocessing, model architecture, training processes, and evaluation metrics.

DATA PRE-PROCESSING

Rescaling: The skin lesion images are adjusted to a standardized size of 176x176 pixels to maintain uniform input dimensions across all samples.

Normalization: Pixel intensity values are adjusted to fall within the range of [0, 1] by dividing each pixel value by 255. This standardization step ensures uniform input data and aids in the convergence process during model training.

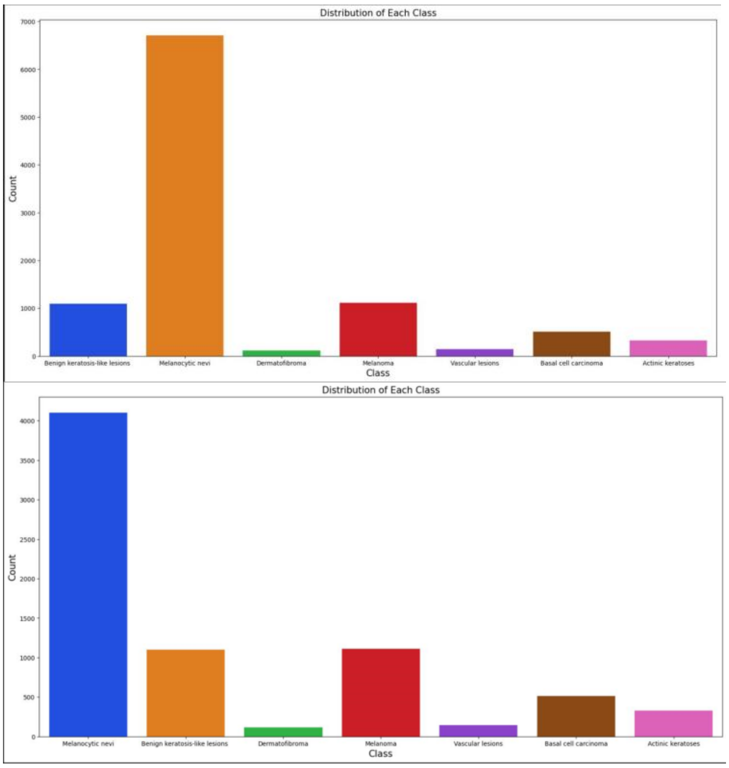


Fig. 2 Original and down-sized class sizes

MODEL ARCHITECTURE

Multiple convolutional and max-pooling layers make up our CNN architecture, which is followed by fully connected layers.

- Convolutional Layers: To extract spatial features from the input skin lesion images, two convolutional layers with different kernel sizes and strides are used. Non-linearity is introduced through the application of ReLU activation functions.
- Max-Pooling Layer: One max-pooling layer is positioned in between convolutional layers to down

sample the feature maps and lessen computational load while keeping significant features intact.

- Fully Connected Layer: One dense (fully connected) layer that serves as a classifier receives the flattened output from the convolutional layers. During training, dropout regularization randomly deactivates neurons to reduce overfitting.
- Output Layer: SoftMax activation makes up the last layer, which generates probabilities for every category of skin disease.

TRAINING PROCEDURE

The Adam optimizer with a categorical cross-entropy loss function is used to train the model.

Our approach involves down sampling of the dataset to guard against overfitting and preserve the top-performing model determined by validation loss.

Testing is done on a portion of the dataset, and model performance is tracked using validation data.

EVALUATION PROCEDURE

A variety of evaluation metrics, such as accuracy, precision, F1 score, recall, loss, confusion matrix, and classification report, are used to evaluate the performance of the model.

- Accuracy: Accuracy is calculated as the ratio of correctly predicted instances to the total instances.
- Loss: The difference between the expected and actual class labels, as indicated by the value of the categorical cross-entropy loss function.
- Confusion Matrix: A matrix that provides an overview of the quantity of accurate positive, accurate negative, false positive, and false negative forecasts.
- Classification Report: furnishes each class with precision, recall, F1-score, and support, providing valuable information about the model's efficacy in various skin conditions.

PREDICTION ON TEST DATASET

The model is assessed on a separate test dataset that was not used for training or validation after it has been trained and validated.
The skin lesion photos in the test dataset are paired with ground truth labels for the various types of skin diseases.
For every skin lesion image in the test dataset, the trained CNN model is utilized to forecast the type of skin disease.
By running the test images through the trained model and obtaining the output probabilities for each class (skin disease type), predictions are made.

The performance of the model on unobserved data is evaluated by comparing the predicted labels with the ground truth labels.
The model's predictions on the test dataset are used to calculate evaluation metrics like accuracy, loss, confusion matrix, and classification report.

Our goal with this method is to create a reliable and accurate CNN model that can identify the type of skin disease from pictures of skin lesions. We aim to improve patient care outcomes, advance automated skin disease classification, and detect skin cancer early by utilizing cutting-edge deep learning techniques and thorough evaluation.

MODEL 1

Data Splitting
The train_test_split function from the scikit-learn library was used to divide the dataset into training, testing, and validation sets. There were 5187 samples in the training set, 1556 samples in the testing set, and 667 samples in the validation set. Every sample featured a 176 x 176-pixel dimensions.

Model Architecture
Two convolutional layers made up the CNN model, which was then followed by 2 fully connected dense layers for classification, a flattening layer to turn the 2D output into a 1D vector, and max-pooling layers for down sampling.

Layer (type)	**Output Shape**	**Param #**
conv2d (Conv2D)	(None, 88, 88, 32)	896
conv2d_1 (Conv2D)	(None, 44, 44, 64)	18,496
max_pooling2d	(None, 22, 22, 64)	0
flatten	(None, 30976)	0
dense	(None, 1024)	31,720,448
dropout	(None, 1024)	0
dense_1	(None, 7)	7,175

Total params: 31,747,015 (121.11 MB)
Trainable params: 31,747,015 (121.11 MB)
Non-trainable params: 0 (0.00 B)

Fig. 3 Model 1 summary

Training
The Adam optimizer and the categorical cross-entropy loss function were used to compile the model. To avoid overfitting and preserve the best model, the model was trained over 11 epochs.

Training and Validation
On the training set, the model's peak accuracy was 85.23%, and on the validation set, it was 69.53%. The corresponding losses were, respectively, 40.53 and 99.59.

Plotting of accuracy and loss over epochs was done for training and validation.

Testing
After the trained model was assessed using the test set, the test accuracy was 66.26%.

MODEL 2

Model Architecture
4 convolutional layers with 32, 64, 128, and 256 filters each made up the customized CNN model. 2 maxpooling layers were then used for down sampling. After that, the output was flattened and fed into 2 dense layers that were fully connected for classification.

Layer (type)	**Output Shape**	**Param #**
conv2d_6 (Conv2D)	(None, 88, 88, 32)	896
max_pooling2d_3 (MaxPooling2D)	(None, 44, 44, 32)	0
conv2d_7 (Conv2D)	(None, 22, 22, 64)	18,496
max_pooling2d_4 (MaxPooling2D)	(None, 11, 11, 64)	0
conv2d_8 (Conv2D)	(None, 6, 6, 128)	73,856
conv2d_9 (Conv2D)	(None, 3, 3, 256)	295,168
flatten_2 (Flatten)	(None, 2304)	0
dense_4 (Dense)	(None, 1024)	2,360,320
dropout_2 (Dropout)	(None, 1024)	0
dense_5 (Dense)	(None, 7)	7,175

Total params: 2,755,911 (10.51 MB)
Trainable params: 2,755,911 (10.51 MB)
Non-trainable params: 0 (0.00 B)

Fig. 4 Model 2 summary

Training
The Adam optimizer and the categorical cross-entropy loss function were used to compile the model. To avoid overfitting and preserve the best model, the model was trained over 11 epochs.

Training and Validation

On the training set, the model's peak accuracy was 73.85%, and on the validation set, it was 68.05%. The corresponding losses were, respectively, 68.37 and 86.49. Plotting of accuracy and loss over epochs was done for training and validation.

Testing
After the trained model was assessed using the test set, the test accuracy was 67.76%.

MODEL 3

Model Architecture
The implemented model is a ResNet50 neural network, configured with 23,602,055 parameters, of which 23,548,935 are trainable and 53,120 are non-trainable. It is designed to classify images into 7 categories with an input shape of (176, 176, 3). The model employs the Adam optimizer and categorical cross entropy loss function, achieving validation through three epochs of training.

Layer	**Output Shape**	**Param #**	**Connected to**
conv5_block3_out	(None, 6, 6, 2048)	0	conv5_block3_add[0][0]
avg_pool	(None, 2048)	0	conv5_block3_out[0][0]
predictions	(None, 7)	14,343	avg_pool[0][0]

Total params: 23,602,055 (90.03 MB)
Trainable params: 23,548,935 (89.83 MB)
Non-trainable params: 53,120 (207.50 KB)

Fig. 5 Model 3 Summary

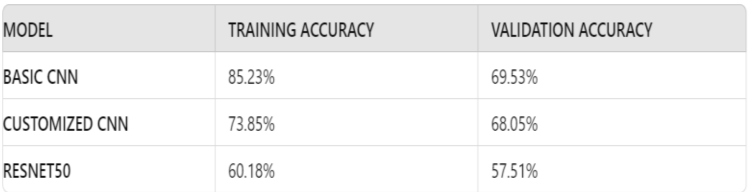
Training
The Adam optimizer and the categorical cross-entropy loss function were used to compile the model. To avoid overfitting and preserve the best model, the model was trained over 3 epochs.

Training and Validation
On the training set, the model's peak accuracy was 60.18%, and on the validation set, it was 57.51%. The corresponding losses were, respectively, 1.09 and 1.43. Plotting of accuracy and loss over epochs was done for training and validation.

Testing
After the trained model was assessed using the test set, the test accuracy was 57.87%.

RESULTS

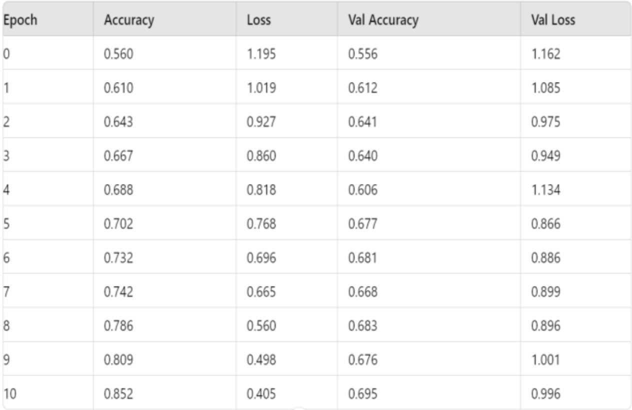
The training and validation accuracy of each model is tabulated below in Figure 6.



MODEL	TRAINING ACCURACY	VALIDATION ACCURACY
BASIC CNN	85.23%	69.53%
CUSTOMIZED CNN	73.85%	68.05%
RESNET50	60.18%	57.51%

Fig. 6 Training and Validation Accuracies

MODEL 1 RESULTS



Epoch	Accuracy	Loss	Val Accuracy	Val Loss
0	0.560	1.195	0.556	1.162
1	0.610	1.019	0.612	1.085
2	0.643	0.927	0.641	0.975
3	0.667	0.860	0.640	0.949
4	0.688	0.818	0.606	1.134
5	0.702	0.768	0.677	0.866
6	0.732	0.696	0.681	0.886
7	0.742	0.665	0.668	0.899
8	0.786	0.560	0.683	0.896
9	0.809	0.498	0.676	1.001
10	0.852	0.405	0.695	0.996

Fig. 7 Model 1 – Epochs vs Accuracy

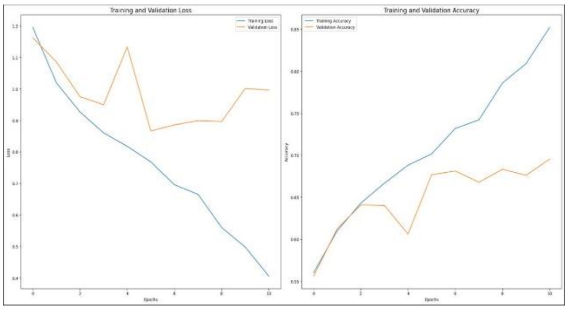
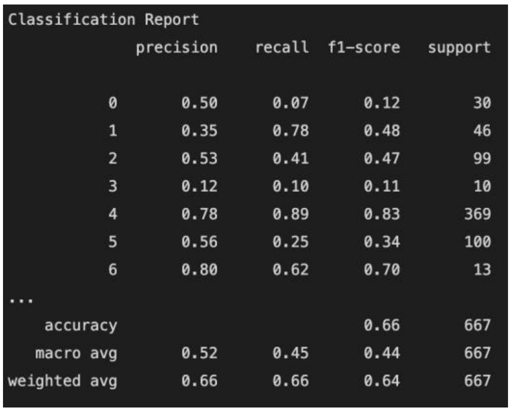


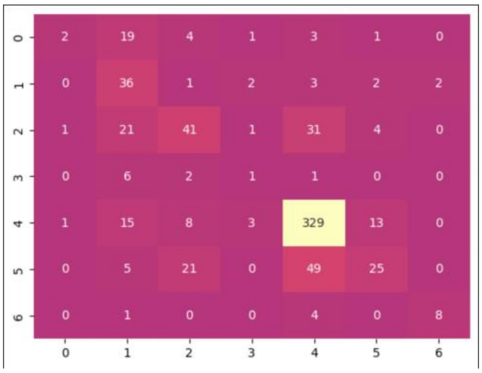
Fig. 8 Model 1 - Training vs Validation Loss/Accuracy



```
Classification Report
              precision    recall  f1-score   support

           0       0.50      0.07      0.12        30
           1       0.35      0.78      0.48        46
           2       0.53      0.41      0.47        99
           3       0.12      0.10      0.11        10
           4       0.78      0.89      0.83       369
           5       0.56      0.25      0.34       100
           6       0.80      0.62      0.70        13
...
    accuracy                           0.66       667
   macro avg       0.52      0.45      0.44       667
weighted avg       0.66      0.66      0.64       667
```

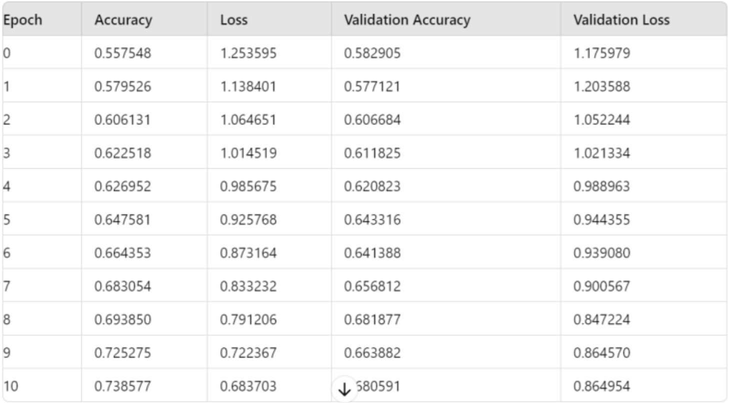
Fig. 9 Model 1 – Classification Report



	0	1	2	3	4	5	6
0	2	19	4	1	3	1	0
1	0	36	1	2	3	2	2
2	1	21	41	1	31	4	0
3	0	6	2	1	1	0	0
4	1	15	8	3	329	13	0
5	0	5	21	0	49	25	0
6	0	1	0	0	4	0	8

Fig. 10 Model 1 – Confusion Matrix

MODEL 2 RESULTS



Epoch	Accuracy	Loss	Validation Accuracy	Validation Loss
0	0.557548	1.253595	0.582905	1.175979
1	0.579526	1.138401	0.577121	1.203588
2	0.606131	1.064651	0.606684	1.052244
3	0.622518	1.014519	0.611825	1.021334
4	0.626952	0.985675	0.620823	0.988963
5	0.647581	0.925768	0.643316	0.944355
6	0.664353	0.873164	0.641388	0.939080
7	0.683054	0.833232	0.656812	0.900567
8	0.693850	0.791206	0.681877	0.847224
9	0.725275	0.722367	0.663882	0.864570
10	0.738577	0.683703	680591	0.864954

Fig 11. Model 2 – Epochs vs Accuracy

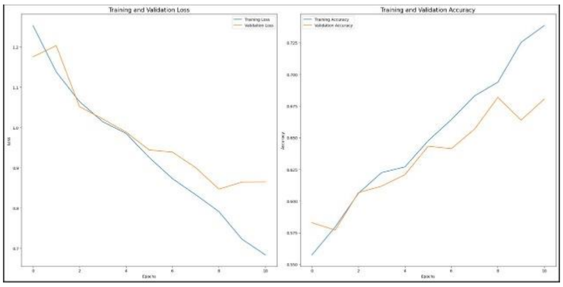
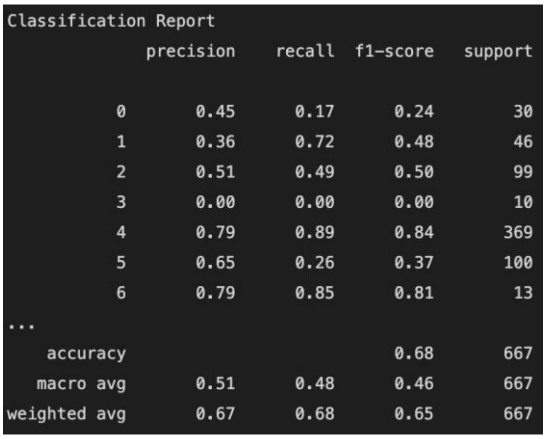


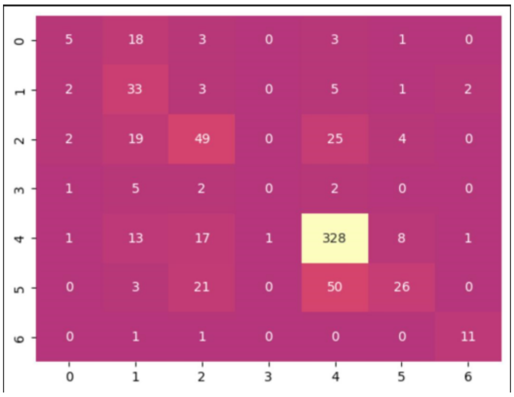
Fig 12. Model 2 – Training vs Validation Loss/Accuracy



```
Classification Report
              precision    recall  f1-score   support

           0       0.45      0.17      0.24        30
           1       0.36      0.72      0.48        46
           2       0.51      0.49      0.50        99
           3       0.00      0.00      0.00        10
           4       0.79      0.89      0.84       369
           5       0.65      0.26      0.37       100
           6       0.79      0.85      0.81        13
...
    accuracy                           0.68       667
   macro avg       0.51      0.48      0.46       667
weighted avg       0.67      0.68      0.65       667
```

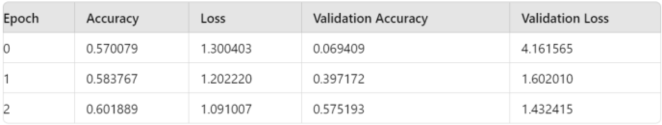
Fig.13 Model 2 – Classification Report



	0	1	2	3	4	5	6
0	5	18	3	0	3	1	0
1	2	33	3	0	5	1	2
2	2	19	49	0	25	4	0
3	1	5	2	0	2	0	0
4	1	13	17	1	328	8	1
5	0	3	21	0	50	26	0
6	0	1	1	0	0	0	11

Fig 14. Model 2 – Confusion Matrix

MODEL 3 RESULTS



Epoch	Accuracy	Loss	Validation Accuracy	Validation Loss
0	0.570079	1.300403	0.069409	4.161565
1	0.583767	1.202220	0.397172	1.602010
2	0.601889	1.091007	0.575193	1.432415

Fig. 15 Model 3 – Epochs vs Accuracy

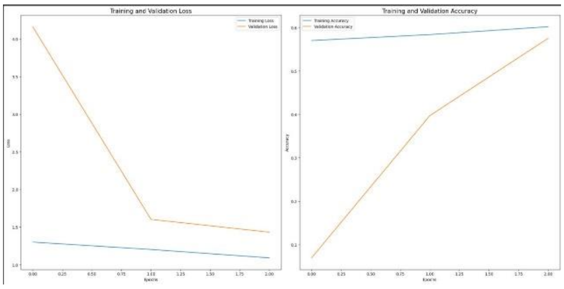
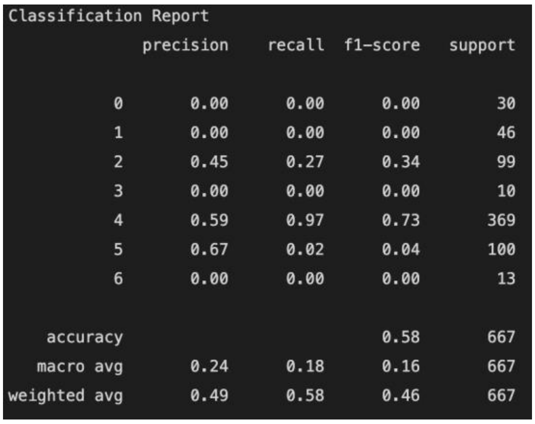


Fig. 16 Model 3 – Training vs Validation Loss/Accuracy

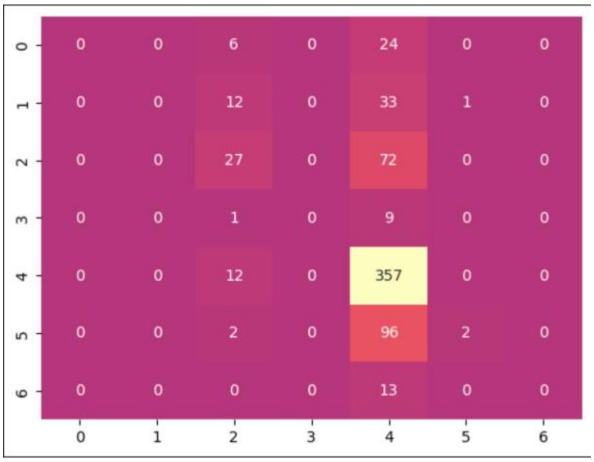


```
Classification Report
              precision    recall  f1-score   support

           0       0.00      0.00      0.00        30
           1       0.00      0.00      0.00        46
           2       0.45      0.27      0.34        99
           3       0.00      0.00      0.00        10
           4       0.59      0.97      0.73       369
           5       0.67      0.02      0.04       100
           6       0.00      0.00      0.00        13

    accuracy                           0.58       667
   macro avg       0.24      0.18      0.16       667
weighted avg       0.49      0.58      0.46       667
```

Fig. 17 Model 3 – Classification Report



	0	1	2	3	4	5	6
0	0	0	6	0	24	0	0
1	0	0	12	0	33	1	0
2	0	0	27	0	72	0	0
3	0	0	1	0	9	0	0
4	0	0	12	0	357	0	0
5	0	0	2	0	96	2	0
6	0	0	0	0	13	0	0

Fig. 18 Model 3 – Confusion Matrix

References

[1] Tschandl, P. et al.,” The HAM10000 dataset, a large collection of multi-sources dermato- scopic images of common pigmented skin lesions,” Harvard Dataverse, 2018. Available: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi: 10.7910/DVN/DBW86T

[2] Papers With Code, ”HAM10000 Dataset,” Available: https://paperswithcode.com/dataset/ham10000

[3]Activeloop,”HAM10000Dataset,”Available:https://datasets.activeloop. ai/dataset/ham10000