# Classifying Skin Cancer Lesions Using Convolutional Neural Networks

Cliff Pham (cliffpm), Idika Kontu (idikak), Giannah Donahoe (giannahd), Hanan Kedir (hanank)

## Background

Our final project revolved around a Kaggle dataset titled "Skin Cancer MNIST: HAM10000." Our model seeks to accelerate automatic diagnosis of pigmented skin lesions by collecting 10015 dermascopic images of skin lesions from diverse populations stored by different modalities. We process a representative collection of different kinds of lesions including multiple variations of carcinomas, melanoma, vascular lesions and keratoses. Most of the images have a confirmed diagnosis via histiopathology. When that is not available, diagnoses are verified through follow-up appointments, confocal microscopy or general expert consensus.

In order to develop a clean, high accuracy model that would account for this diverse dataset and provide meaningful insights accurately, we chose to develop a convolutional neural network that supports our well labeled, multi-class dataset. A convolutional neural network (CNN) is a specialized type of neural network designed for processing matrix-like grid data. It uses parameter sharing and multiple layers (input layer, convolutional layer, pooling layer, and fully connected layers) that apply filters to create feature maps. This allows the networks to learn complex features by applying non-linearity, downsamples to allow for small changes throughout the input. It then performs classification for our image recognition. The parameter sharing, automatic feature learning and translation invariance (via pooling) helps recognize features of a skin cancer regardless of location in the image. In addition, parameter sharing reduces the total number of parameters by reusing the same filter weights across the image.

## Methodology

Our input images were (180 x 180 x 3) in dimension. We utilized a CNN model with a 80/20 train validation split with 40 epochs. We then introduced data augmentation to our model while keeping the same network architecture and training schedule. The data augmentation we adopted included random horizontal and vertical flips as well as random resized crops. To continue to address overfitting, we included batch normalization into our CNN architecture after each convolutional and dropout layer. Lastly, we incorporated class-weighted cross-entrophy loss to account for class imbalance in the training set. We computed balanced class weights from the training labels and trained the model for 40 epochs using these weights. We have learned and utilized CNN models, epochs, dropout layers, batch normalization, entropy loss, and data augmentation in class.

## Results

We evaluated our CNN for the classification of nine classes of skin lesions using an 80/20 train validation split. Tracking the performance over 40 epochs, we measured training and validation accuracy and loss throughout the epochs.

In the baseline model, it was observed that the training accuracy steadily increased from 27.7% (epoch 1) to 86.9% (epoch 20). However, the validation accuracy plateaued around 50 - 55%. Validation loss began to flatten and later increase as the number of epochs grew. The rising gap between the training and validation performance allowed us to conclude that the model was overfitting. Overfitting occurs when patterns are memorized in the training set without improving generalization to unseen input.


*Figure 1. Training vs Validation Loss over 20 epochs in Baseline Model*

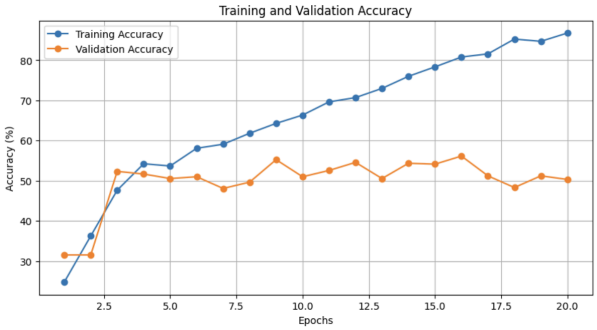



*Figure 2. Training vs Validation Accuracy over 20 epochs in Baseline Model*

Data augmentation resulted in a training accuracy curve that was lower and smoother. Training accuracy curve ended at 65.6% while validation accuracy improved to about 55.7% maximum. The training and validation curves were significantly closer together and the loss curves were no longer diverging as strongly in the baseline model. In all, this suggests that the data augmentation successfully reduced overfitting and led to a more robust model.


*Figure 3. Training vs Validation Loss with data augmentation*

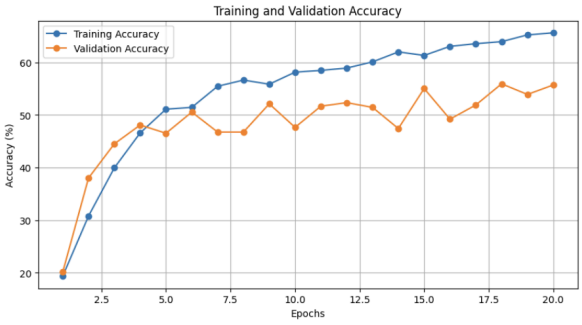
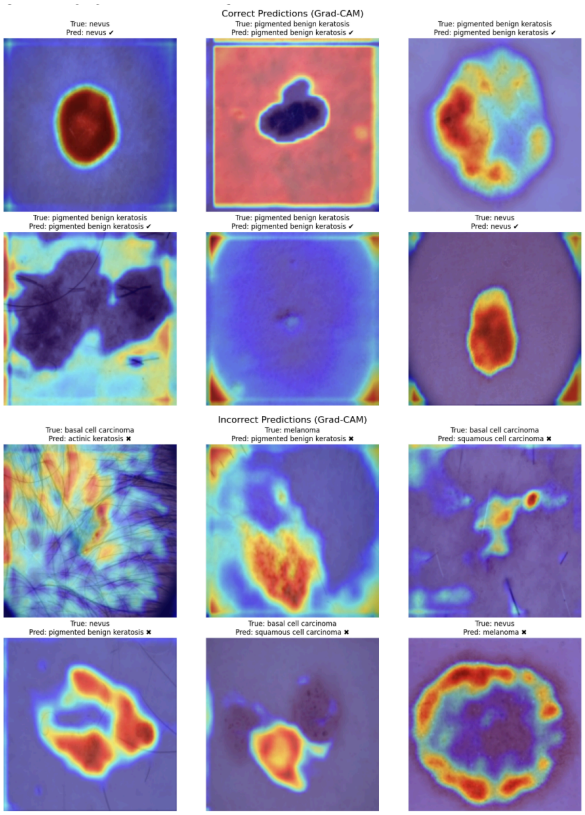



*Figure 4. Training vs Validation Accuracy with data augmentation*

With the batch normalization and the dropout layer, the training accuracy remained around 45-51% and noisier validation loss and accuracy curves. Dropout results in randomly dropping nodes during the training which introduces additional variance between epochs. Batch normalization did not perform better than the data augmentation. The peak validation accuracy remained around the same. Due to the shallow depth and limited capacity of our CNN, these

techniques seemed to reduce underfitting less than they reduced the model's ability to fit the data. This led to a slightly worse overall performance.

With class-weighted cross entropy loss , training accuracy remained lower and validation stayed in the same range. There was significant variability across epochs. We noticed that the weighted loss made more errors on minority classes and overall validation performance decreased compared to the unweighted, augmented model.




To better understand what the CNN was learning, we generated Grad-CAM visualization on correctly and incorrectly classified validation images. The final convolutional layer was used as the target. These heatmaps allowed us to inspect which regions of the input images contributed most to the model's predictions. With this visualization, we can compare model attention between successful and failed classifications. This analysis gave us insight into how the model makes decisions and provides a starting point for future work in auditing the model and exploring its clinical relevance.

*Figure 5. Grad-CAM visualization for correctly and incorrectly classified lesions*

**Conclusion**

Across our experiments, the model trained with data augmentation alone performed the best. It achieved a peak validation accuracy of about 55-56% and showed a smaller gap between the training and validation curves. This indicated better generalization. Our key finding is that simple augmentation is more effective than adding dropout, batch normalization, or class-weighted loss in our shallow CNN. Grad-CAM visualizations indicated where the model focused during predictions. Challenges include moderate overall accuracy, class imbalance, limited dataset diversity, and the constraints of a small model architecture. This highlights the need for deeper models, richer datasets, and further validation.

References

IBM. (2021, October 6). Convolutional Neural Networks. Ibm.com. https://www.ibm.com/think/topics/convolutional-neural-networks

ML Practicum: Image Classification | Machine Learning Practica. (n.d.). Google Developers. https://developers.google.com/machine-learning/practica/image-classification/convolutional-neural-networks

Skin Cancer MNIST: HAM10000. (n.d.). Www.kaggle.com. https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000