Third Capstone Report
Skin Disease Image Classification Using CNN

Problem

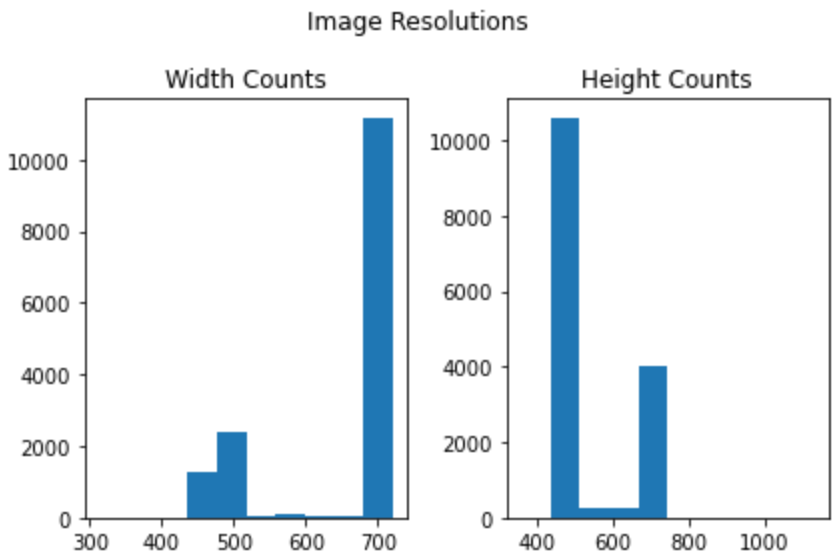
At the beginning of the pandemic, personal healthcare suffered dramatically. In person access to a medical professional was either rare, limited or non-existent. Video conferencing with a physician was possible but believe it or not, there are still individuals who either don't have internet, a cell phone, or even a webcam. My relative was a member of the last two groups. Being very self reliant, and a little stubborn they simply managed an unexpected skin rash the best they could with the hopes it would resolve itself.

Unfortunately, that didn't happen. A few weeks went by with no improvement. Eventually they visited a clinic but were effectively turned away and told them the Doctor could only examine them over video conference. Ultimately they simply dealt with the issue and it did eventually subside. However, what if this had been a more serious issue? As well if this weren't a serious issue, why waste resources and time on a condition a familiar Physician should be able to diagnose very quickly? It seemed pretty obvious to me that if trained properly, a ML algorithm could save both the patient and Doctor money and time by either diagnosing the condition or giving a diagnostic starting point for further research. At the very least it could provide peace of mind to the patient by confirming whether the condition was serious enough to have a Physician involved.

With this in mind I set out to create and train a Convolutional Deep Neural Network that could classify a skin condition with at least 50% accuracy.

Data

Dermnet is an online skin disease atlas and contains over thousands of images across 23 broad categories. The dataset I used was taken from Kaggle and scraped by Shubham Ggoel, Skin Data Set, and contained 15,557 images for training and 3,946 for testing. The first issue I found with the dataset was that all images had a Dermnet watermark, however it was very light. The median image resolution was 720x480.



Upon inspecting some of the images, I won't show any for more sensitive audiences, I realized that within each category there were actually numerous sub-categories all with varying numbers of pictures and of varying body parts. My first thought was to attempt dividing each category into additional sub-categories, however there was no thorough naming convention that could be used to automate the process. I performed some light NLP on the category and file names to see if such an operation was possible. Even after removing the pictures with the smallest number of examples by file name, the varying nature of the filenames was too great and I concluded that it wasn't feasible to recategorize them. I restored the dataset to its original size and proceeded with modeling.

Image Augmentation

I used the Keras Image Data Generator to generate my training and validation streams, applying standard rescaling on each. For the training dataset I had the image generator randomly apply up to 30% shear, 30% zoom, 30% rotation and horizontal flipping. I felt it best to keep the RGB colorspace for each image as with skin conditions color is an important feature.

Training Skin Dataset

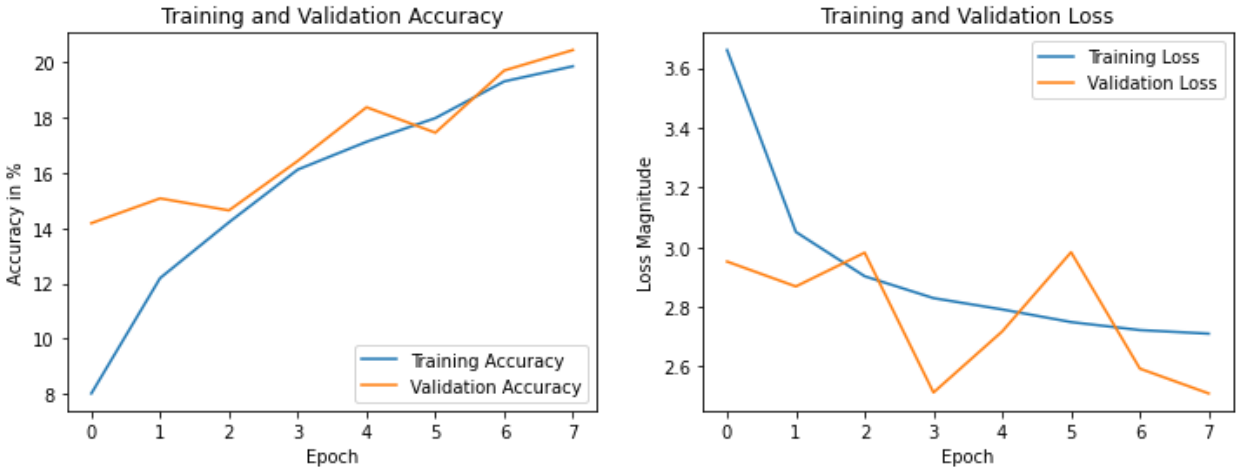
I chose to use a Convolutional Deep Neural Network to learn and classify the images across the 23 separate categories. The sequential network consisted of 7 layers, 4 convolutional layers and 2 fully connected layers and an output layer. Concerned about the quality of my data, I built a function to build and compile the

network implementing early stopping and checkpoints based on the validation loss metric. I used this function to train on a different imageset later on to validate that the model itself was good.

Skin Dataset Results

After running for about 8 epochs, unfortunately, my validation accuracy only reached a little over **20%**, far from my goal of 50%. This was after trying various dropout rates for the convolutional and fully connected layers, various reshape resolutions, activation layers and kernel initializers. Out of 23 layers random chance is between 4-5% so 20% makes the model worthwhile however not by much. A 1 out of 5 chance is still a rather large output range.

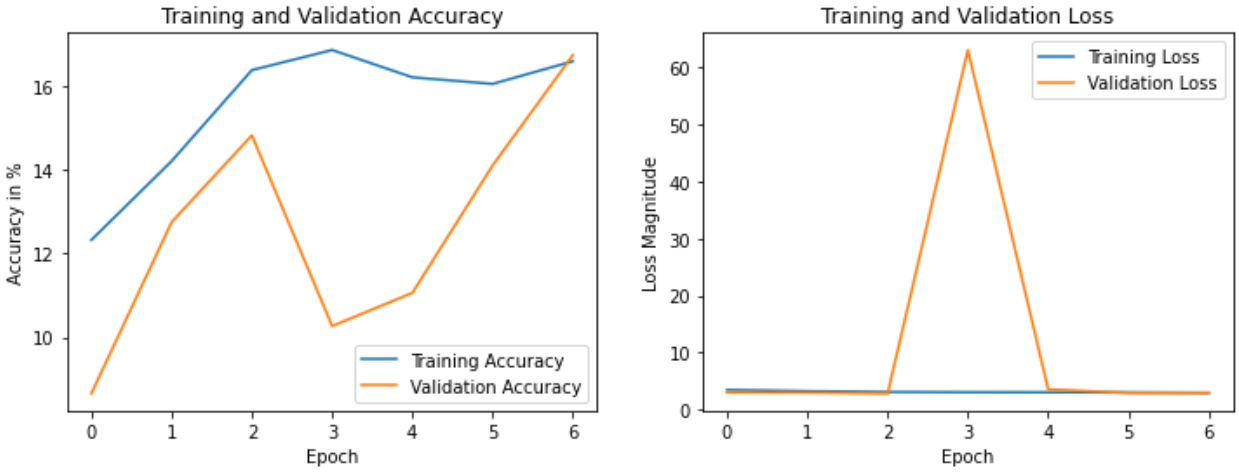
My Model Skin Disease Image Set



Transfer Learning - Resnet 50

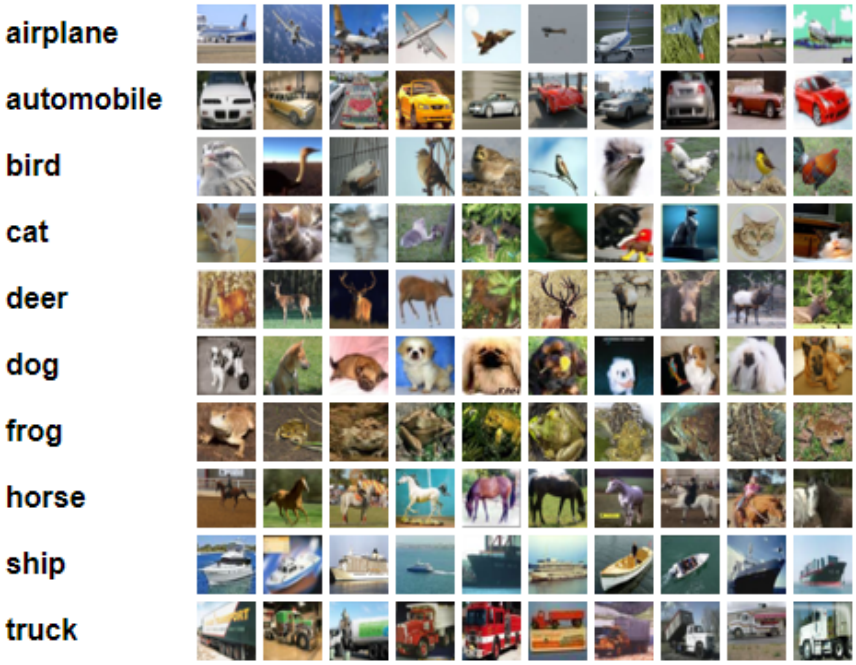
Not absolutely sure if my model or the data was responsible for the low accuracy, I imported and performed transfer learning on the Resnet50 model. Using the default imagenet weights, I froze its layers and added just one FC and one output layer. I experimented with 2 FC layers but this didn't improve the model's validation accuracy. Initially, overfitting was severe, the training accuracy was in the 30% range while the validation accuracy wasn't much better than random chance. Ultimately, I un-froze the layers and simply trained the entire Resnet50 model on the skin dataset. This improved things from an overfitting standpoint, however the validation accuracy still only reached **16%**.

Resnet50 Skin Disease Image Set



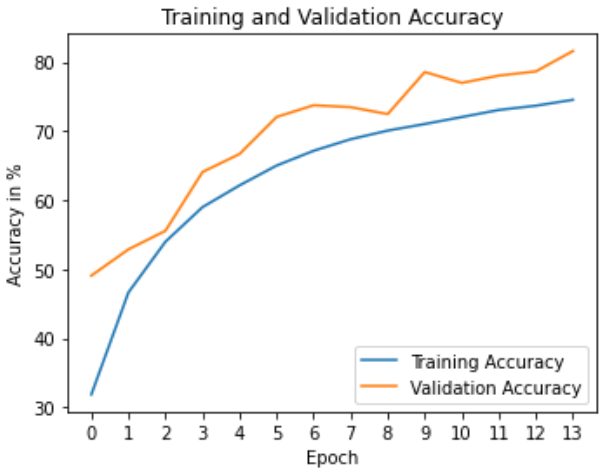
Testing on CiFar-10 Dataset

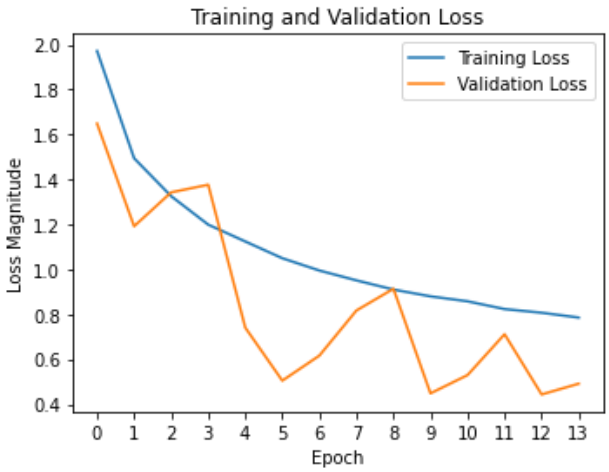
Since both my model and the Resnet50 model performed so poorly on the skin dataset, I wanted to validate the model's quality on a trusted dataset where I knew the images were of a certain quality. I used the CiFar-10 dataset which consists of 60000 color images across 10 classes with 50000 training images and 10000 test images.



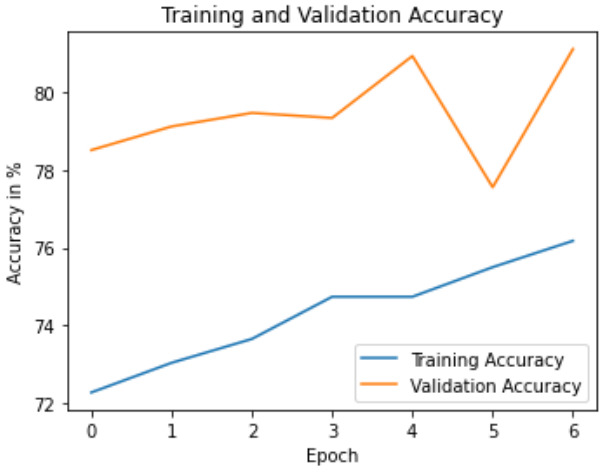
Using the same Image Data Generator as before in order to keep the same augmentation as with the skin dataset, I trained both models and both showed very acceptable results of greater **80%** validation accuracy.

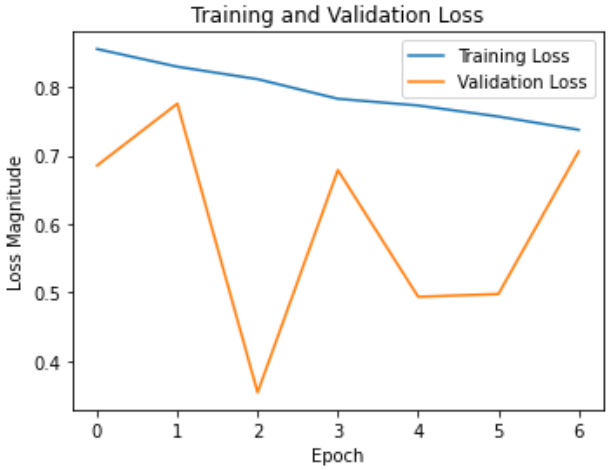
My Model CiFAR10 Image Set





Resnet50 Model CiFar10 Image Set





After receiving much better results on the CiFar-10 image set I feel confident that the issue is the skin dataset and not the quality of my Neural Network.

Further Research

Feeling confident that my Network's poor performance was due to the quality of the data I have listed some modifications that I believe would improve accuracy and usefulness.

- Perform watermark removal on the image set or train on images without watermarks.

- Sub divide the image set into even more categories, ideally one per specific condition instead of broadly grouped conditions.

- Run the images through a body part classifier and create even more categories, perhaps a body part per specific condition.

- Tune the model as best as possible to avoid false negatives for more serious conditions.

- Have the model return the top 5 most likely Skin Conditions, not just the most likely.

Recommendations

- Implement the model as an API and or a mobile device app.

- Allow users to submit their own pictures to improve accuracy even more and have them classified by body part. Design app so when inputting images there are only fixed options to select.

- Work with Physicians and Medical Students to check accuracy of submitted pictures and provide free/low cost human classification, similar to CAPTCHAs.

- Encourage Nurses and Physician Assistants to use for initial diagnosis.