Carter Brinton
Dr. Eric Henderson
CS334 - Algorithms of Machine Learning
December 5th 2023

Final Report:
Skin Cancer Detection Using Convolutional Neural Network

Motivation

One of the biggest reasons I chose this project is because my mom was diagnosed with a very rare form of skin cancer. Although her cancer didn't originate from a mole-like type of skin cancer, nor is her cancer a part of this list. It still got me wishing there was a form of identification for early signs of cancer without taking the time and money to go to the nearest healthcare provider. Hence, I thought it would be cool to train a model on a large data set of moles and expand it to let people take a picture of a mole or spot on their arm for self identification which could encourage people to go to their nearest health care provider.

Skin cancer is the most common cancer in the United States and worldwide. More than two people die of skin cancer in the U.S. every hour. This could potentially be reduced by catching the early symptoms of skin cancer before it's too late. By using a form of Skin Cancer Detection through Convolutional Neural Networks (CNNs) we can accurately identify skin cancer lesions from dermoscopic images. Our goal is to prove that CNNs, when trained on a diverse and well-annotated dermatology image dataset, can significantly improve the early detection of skin cancer, ultimately saving the lives of many around the globe.

Data

For this project, I considered several datasets to explore and evaluate their suitability for the skin cancer detection model. I ultimately decided on the Skin Cancer MNIST: HAM10000 dataset from Kaggle by K SCOTT MADER.

The HAM10000 ("Human Against Machine with 10000 training images") dataset consists of 10015 dermatoscopic images. which are released as a training set for academic machine learning purposes and are publicly available through the ISIC archive. I will try to detect 7 different classes of skin cancer using the Convolution Neural Network with Keras Tensorflow in the backend and then analyze the result to see how the model can be useful in a practical scenario.

The 7 different classes of skin cancer from the data set are listed below:

1. Actinic Keratoses: Actinic keratoses are precancerous, rough, scaly patches on the skin, typically caused by sun exposure and having the potential to become squamous cell carcinoma (Can develop into cancer if left untreated).
2. Basal Cell Carcinoma: Basal cell carcinoma is a common, slow-growing skin cancer that originates from basal cells in the skin's outer layer (Cancerous).
3. Benign Keratosis-Like Lesions: Benign keratosis-like lesions include various non-cancerous skin growths that resemble keratosis, such as seborrheic keratosis (Not cancerous).
4. Dermatofibroma: Dermatofibroma is a benign skin lesion characterized by small, raised bumps often with a brownish color and typically forming after minor skin trauma (Not cancerous).
5. Melanocytic Nevi: Melanocytic nevi, or moles, are benign skin growths formed from clusters of pigment-producing cells (Not cancerous).
6. Melanoma: Melanoma is a potentially life-threatening skin cancer that arises from the uncontrolled growth of melanocytes, often appearing as irregularly shaped, discolored moles (Cancerous).
7. Vascular Lesions: Vascular lesions are skin abnormalities related to blood vessels, encompassing conditions like hemangiomas, port-wine stains, and telangiectasias (Not cancerous).

Data Collectively

The dataset collectively is nearly 6 GB of data. It contains two folders with the 10,000 images, and it also contains a CSV file “HAM10000_metadata.csv " which is used to form a relationship through image_id as to what label should be associated with each image. The columns in this file are lesion_id, image_id, dx, dx_type, age, sex, and localization. Lesion_id is a generic id for each element in the data set. The ‘image_id’ is the reference ID used to match each of the images in the folders. ‘Dx’ is the skin lesion abbreviation type, and localization is where the skin lesion rests on a part of the body it was identified.

Source: https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000

Libraries and Frameworks

To develop and train the CNN model, I utilized the following libraries and frameworks: Python, TensorFlow, Keras, and scikit-learn for model evaluation and metrics calculation. I also took advantage of JupyterLab and Google Colab. I had to complete this project bouncing back and forth between JupyteLab and Google Colab because of the computational intensity of training a CNN with over 10,000 images. I would run into issues periodically using my GPU, so would also run code on Google Colab while also working locally. This was an ultimate time saver.

Insights

During data wrangling, I encountered some challenges in deciding whether to read the image data directly or if I should use the provided 'hmnist_28_28_RGB.csv' file. Since I wanted to have control of the resizing I chose to go with reading the images manually from the path. Unfortunately, this didn't provide the best results in terms of accuracy. I used the Sequential Model from Tensorflow Keras as it "is appropriate for a plain stack of layers where each layer has exactly one input tensor and one output tensor."

In other words, it is a simple model and common model to be used within tensorflow and keras. According to the documentation below, it listed out the best features to use, how to normalize a model, and other methods for fine-tuning to impute. The documentation from TensorFlow laid the groundwork for creating a successful model, where I could then start changing the values to experiment with the different results outputted.

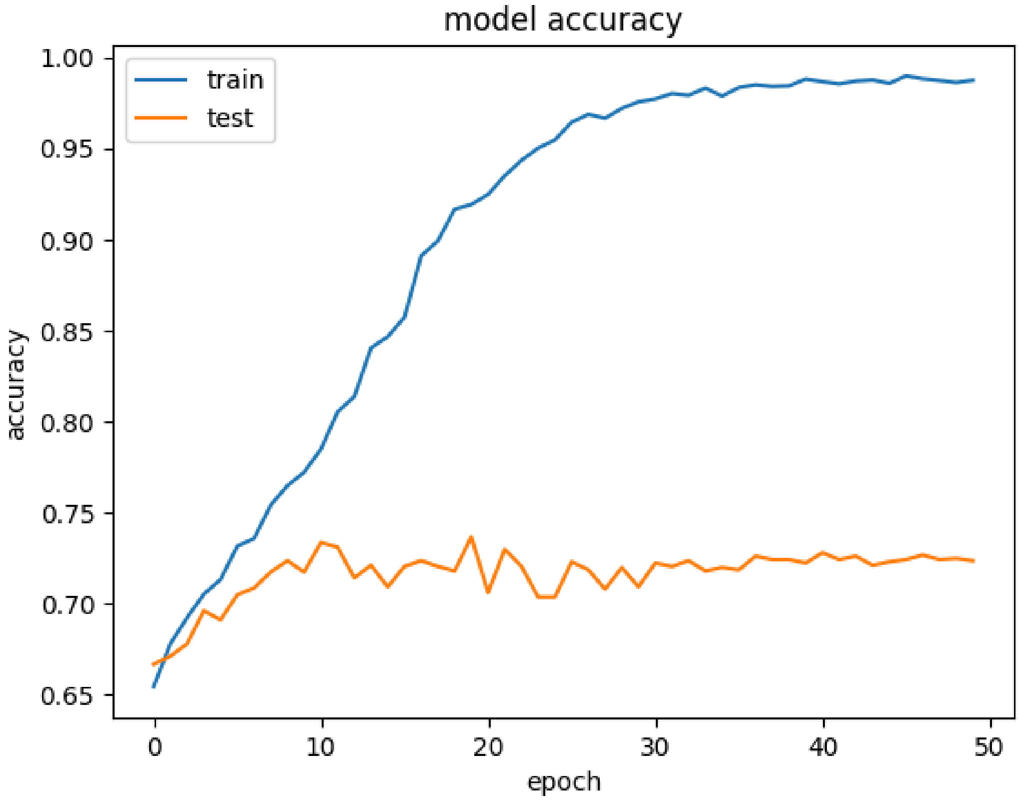
Sources:
-https://www.tensorflow.org/guide/keras/sequential_model
-https://www.analyticsvidhya.com/blog/2021/05/tuning-the-hyperparameters-and-layers-of-neural-network-deep-learning/

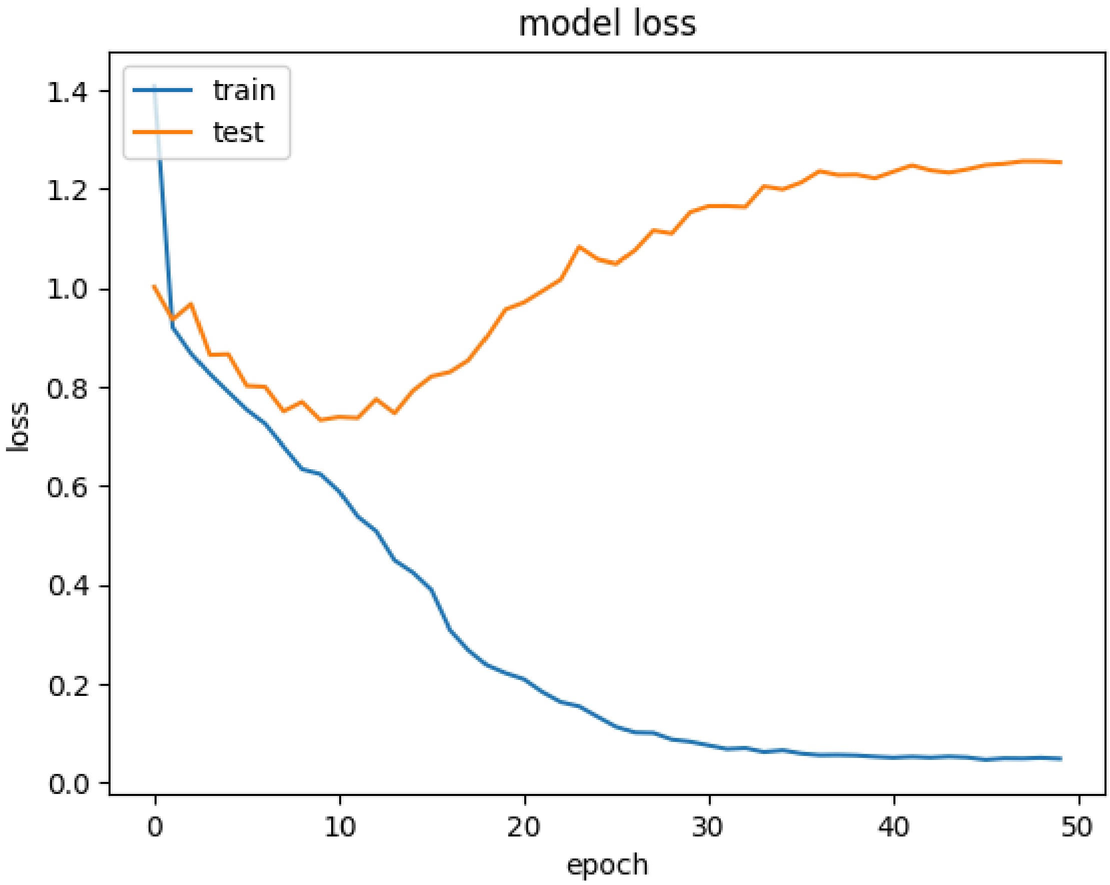
Results

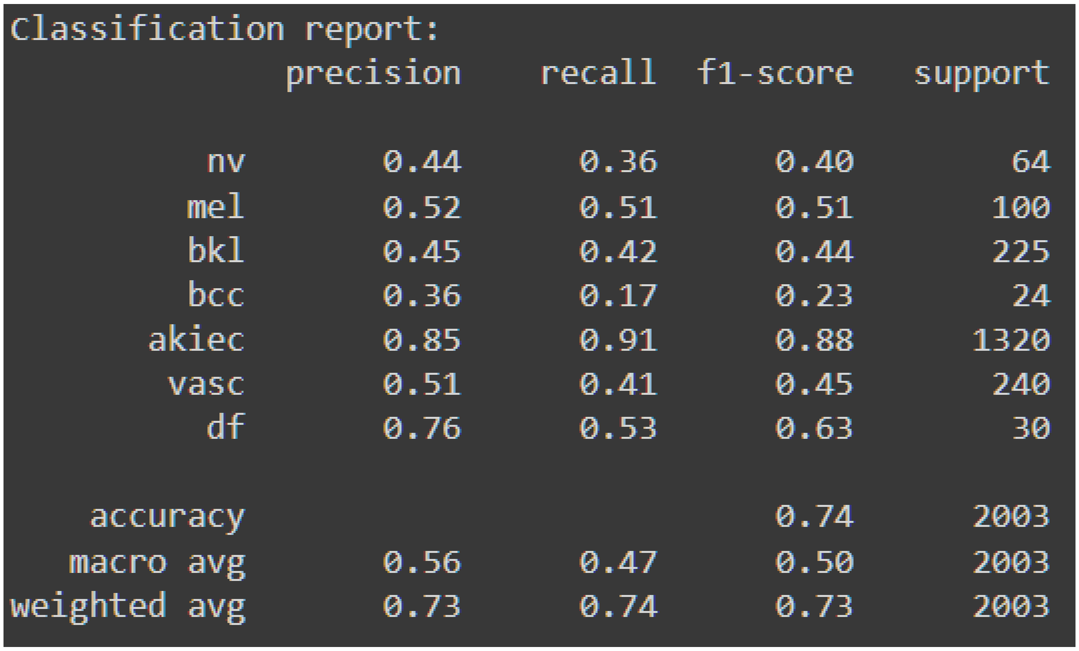
The trained CNN model demonstrated promising results in the classification of skin cancer lesions. However, more work and fine-tuning needs to be done in order to get a higher accuracy, and ultimately use the model within an app. I built out two different models each with their own unique fine tuning. Both models were built with a convolutional neural network, using the images directly when training the model. One model showed an accuracy of 98.9% which is a bit extreme, and definitely had some overfitting. I ended up changing the model structure and hyperparameters entirely, which resulted in an accuracy of 78.9%. Although this model wasn't great at predicting the skin lesion type from a random image off the internet, it gave me better insight as to what could've been done differently. With that being said, the model achieved accuracy, sensitivity, and specificity within acceptable ranges for starting grounds.

<u>Model 1 Results:</u>

- Highest Accuracy: 0.9897019863128662



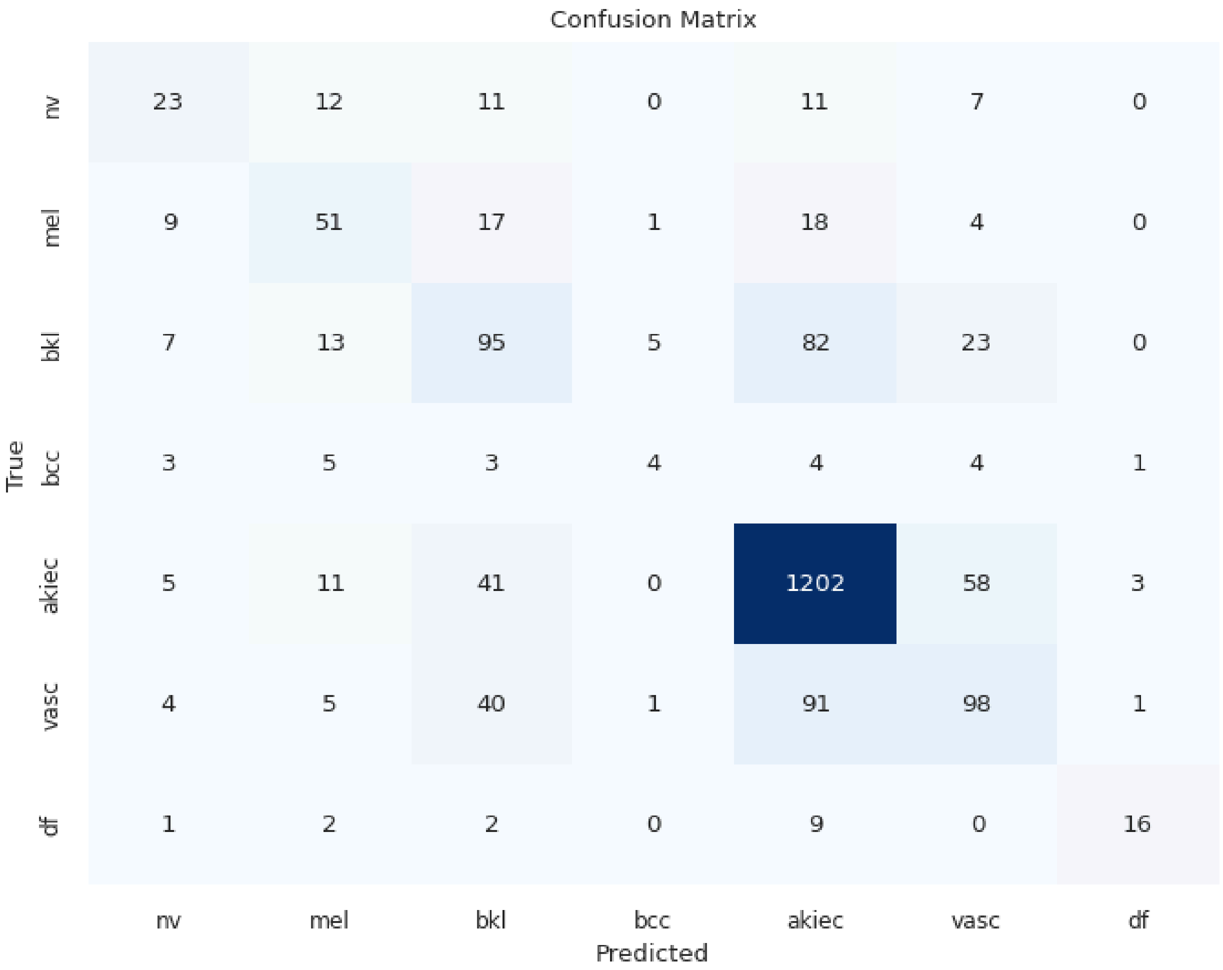




```
Classification report:
              precision    recall  f1-score   support

          nv       0.44      0.36      0.40        64
         mel       0.52      0.51      0.51       100
         bkl       0.45      0.42      0.44       225
         bcc       0.36      0.17      0.23        24
       akiec       0.85      0.91      0.88      1320
        vasc       0.51      0.41      0.45       240
          df       0.76      0.53      0.63        30

    accuracy                           0.74      2003
   macro avg       0.56      0.47      0.50      2003
weighted avg       0.73      0.74      0.73      2003
```

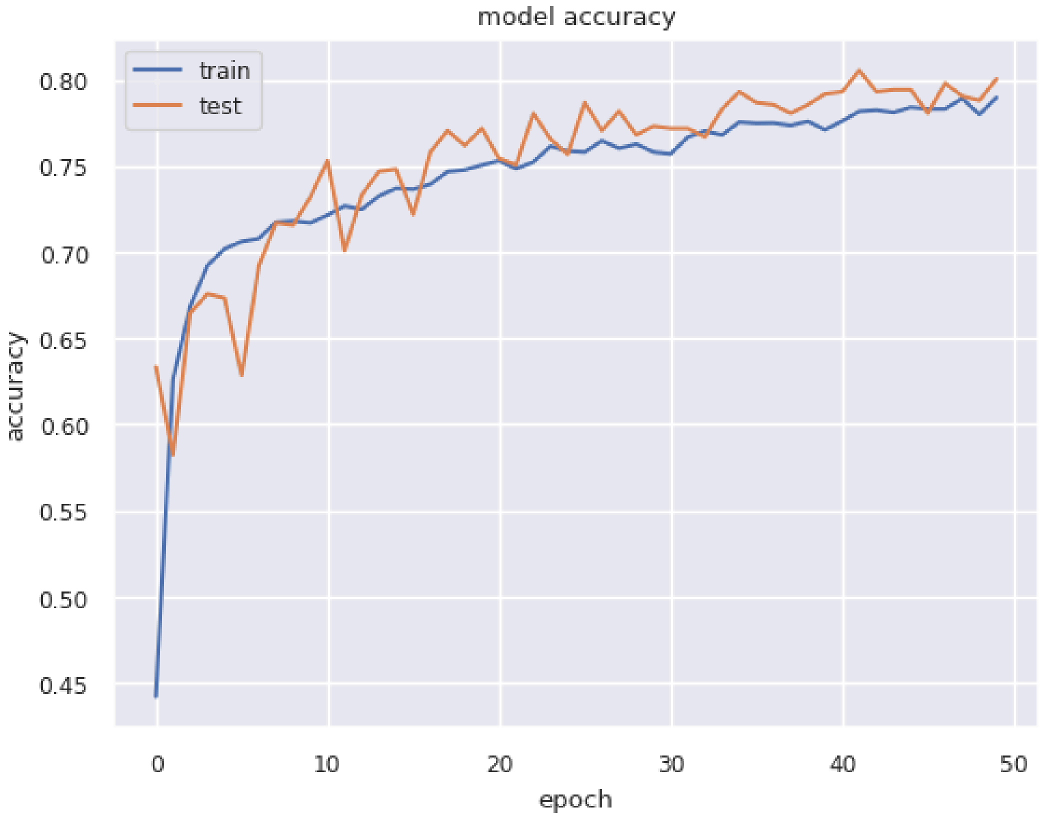


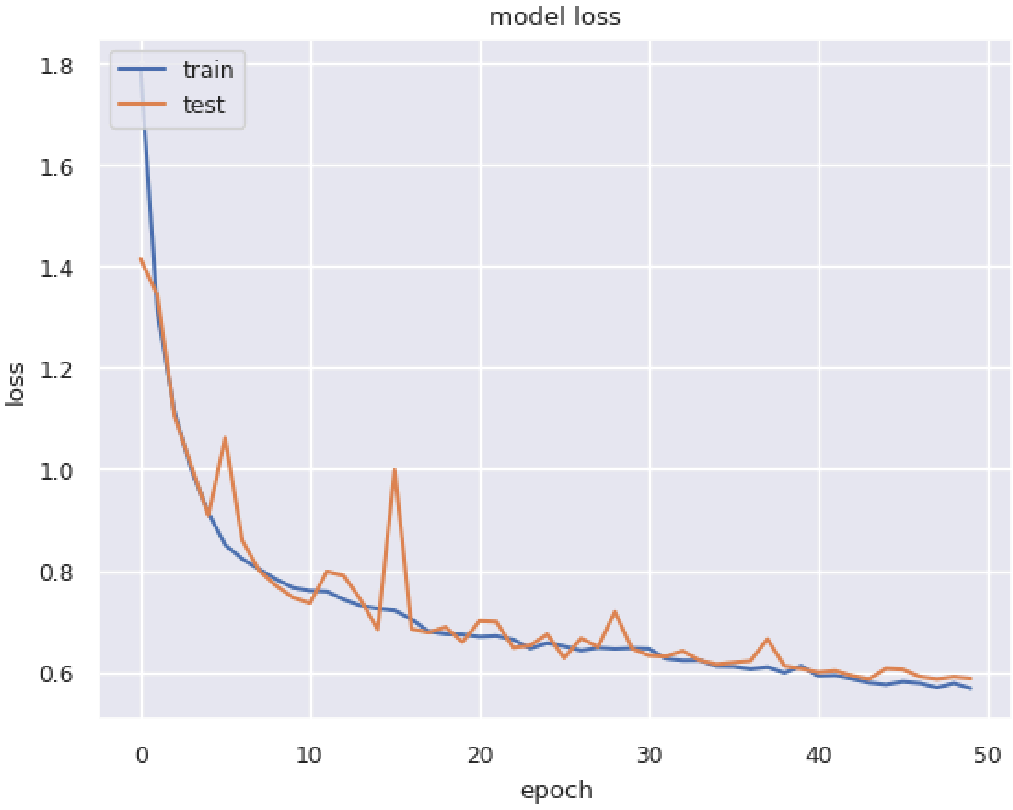
Confusion Matrix

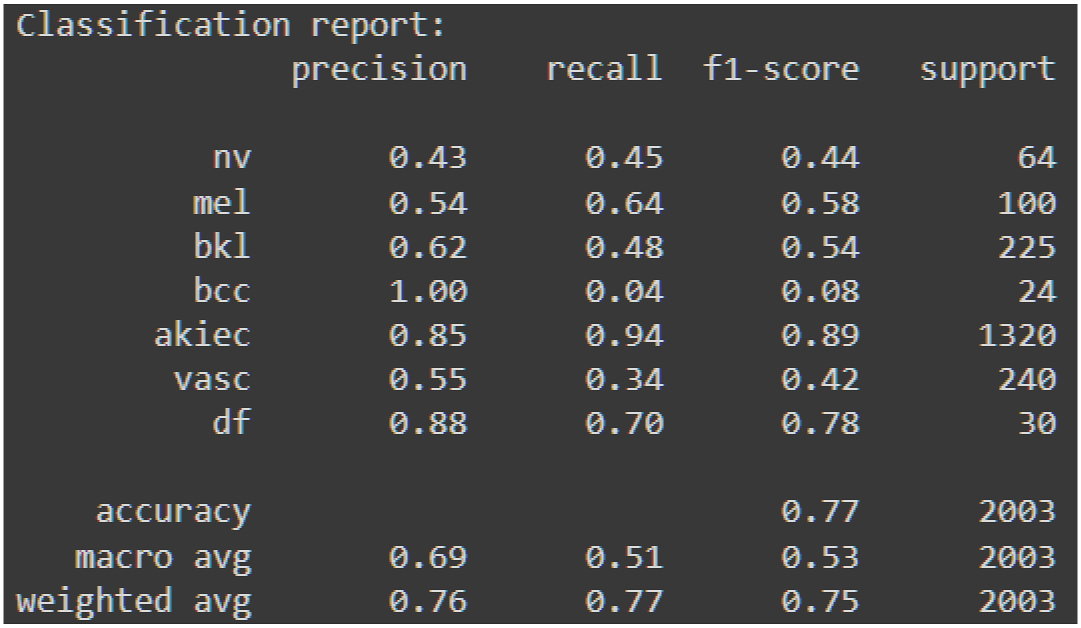
True \ Predicted	nv	mel	bkl	bcc	akiec	vasc	df
nv	23	12	11	0	11	7	0
mel	9	51	17	1	18	4	0
bkl	7	13	95	5	82	23	0
bcc	3	5	3	4	4	4	1
akiec	5	11	41	0	1202	58	3
vasc	4	5	40	1	91	98	1
df	1	2	2	0	9	0	16

Model 2 Results:

- Highest Accuracy: 0.7897364497184753


model accuracy
train
test
accuracy
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0.80
epoch
0
10
20
30
40
50

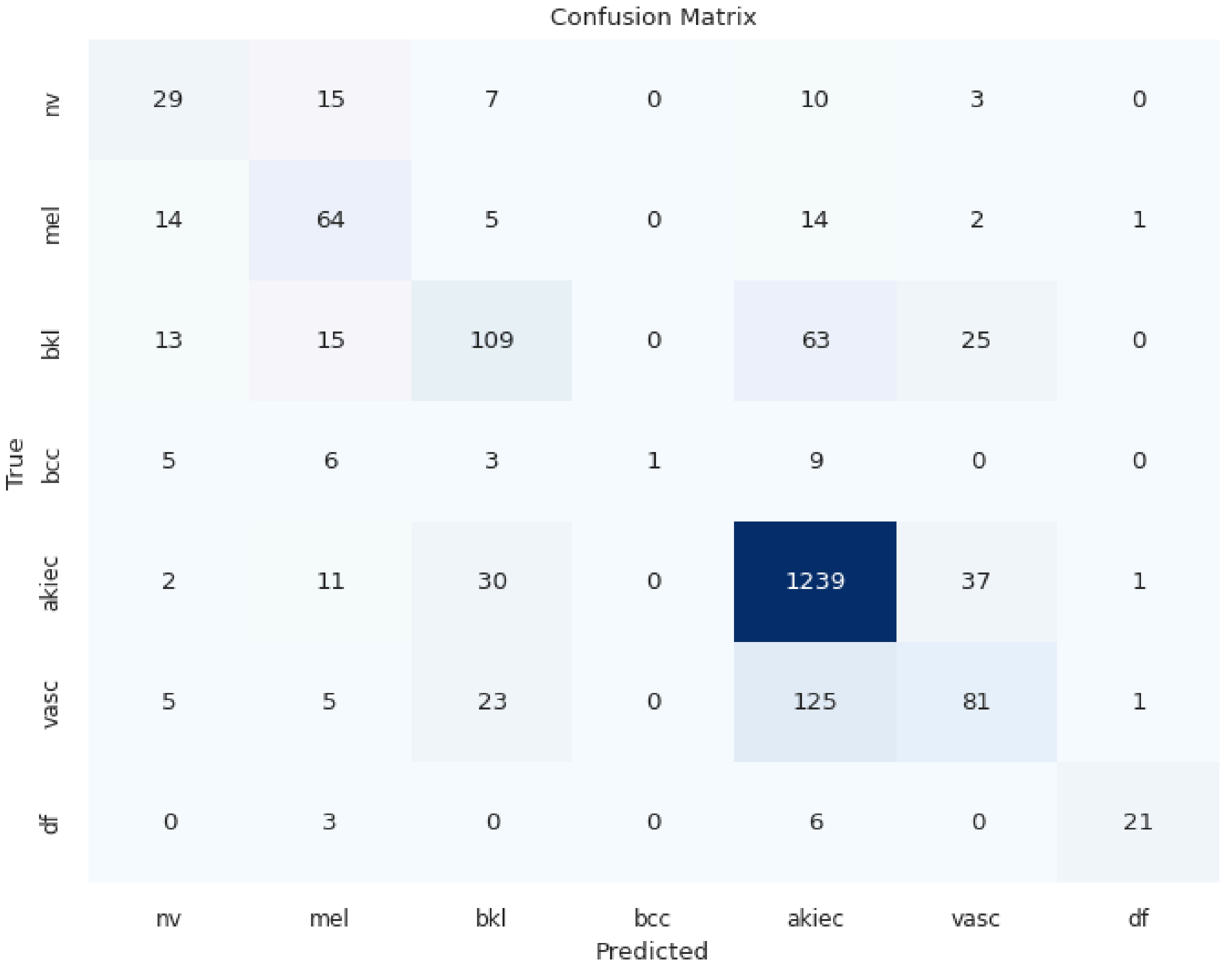

model loss
train
test
loss
0.6
0.8
1.0
1.2
1.4
1.6
1.8
epoch
0
10
20
30
40
50



```
Classification report:
              precision    recall  f1-score   support

          nv       0.43      0.45      0.44        64
         mel       0.54      0.64      0.58       100
         bkl       0.62      0.48      0.54       225
         bcc       1.00      0.04      0.08        24
       akiec       0.85      0.94      0.89      1320
        vasc       0.55      0.34      0.42       240
          df       0.88      0.70      0.78        30

    accuracy                           0.77      2003
   macro avg       0.69      0.51      0.53      2003
weighted avg       0.76      0.77      0.75      2003
```



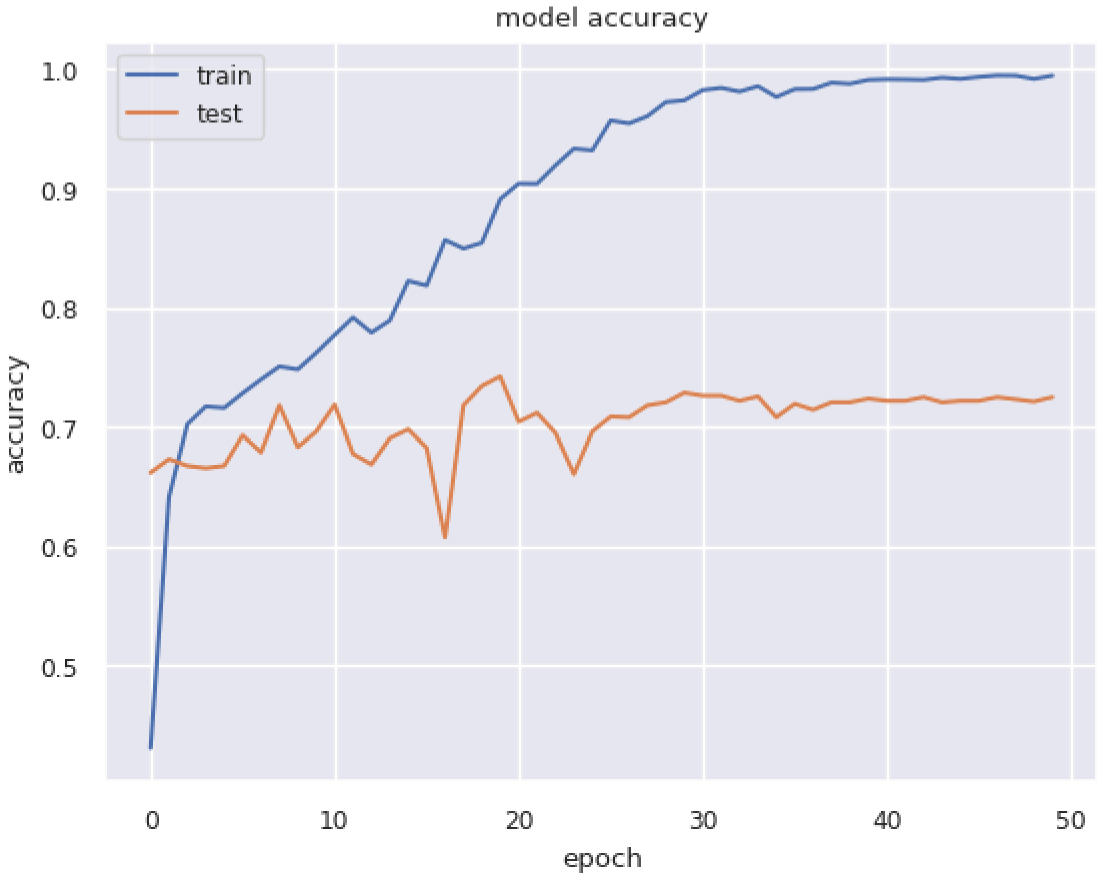
Confusion Matrix

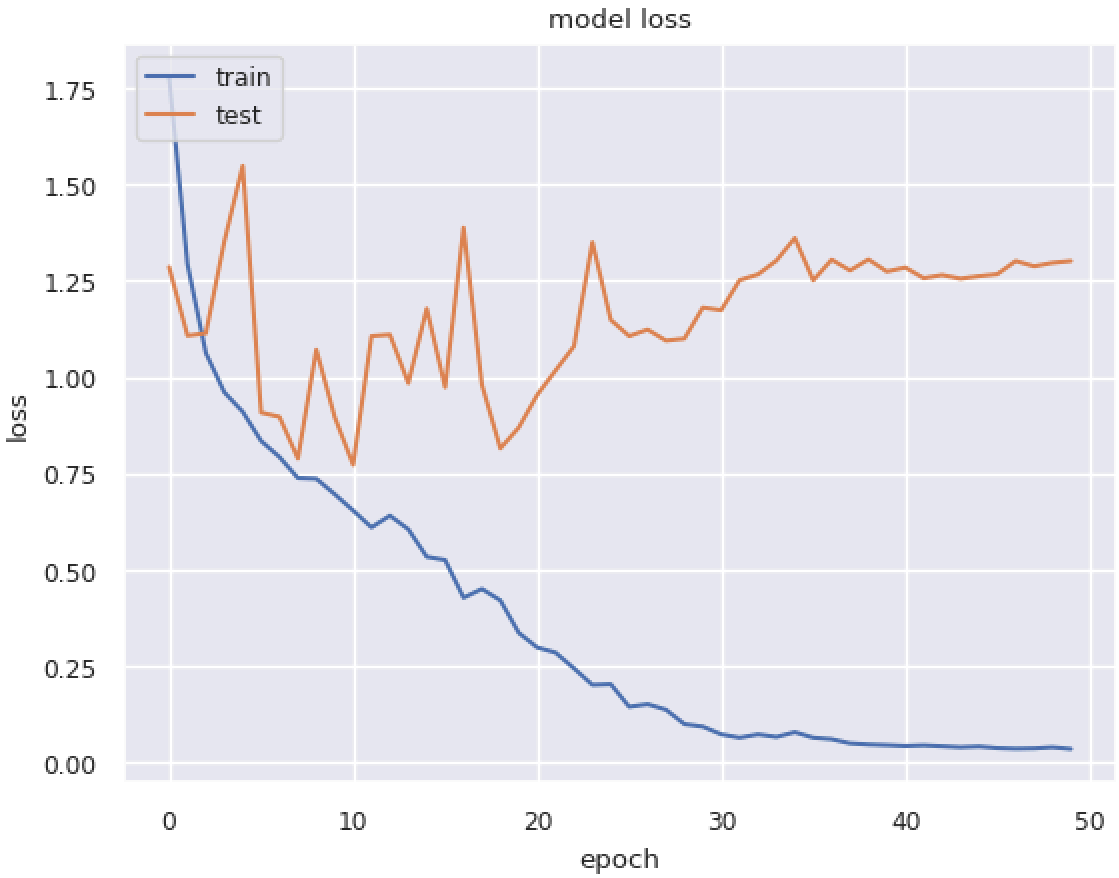
True \ Predicted	nv	mel	bkl	bcc	akiec	vasc	df
nv	29	15	7	0	10	3	0
mel	14	64	5	0	14	2	1
bkl	13	15	109	0	63	25	0
bcc	5	6	3	1	9	0	0
akiec	2	11	30	0	1239	37	1
vasc	5	5	23	0	125	81	1
df	0	3	0	0	6	0	21

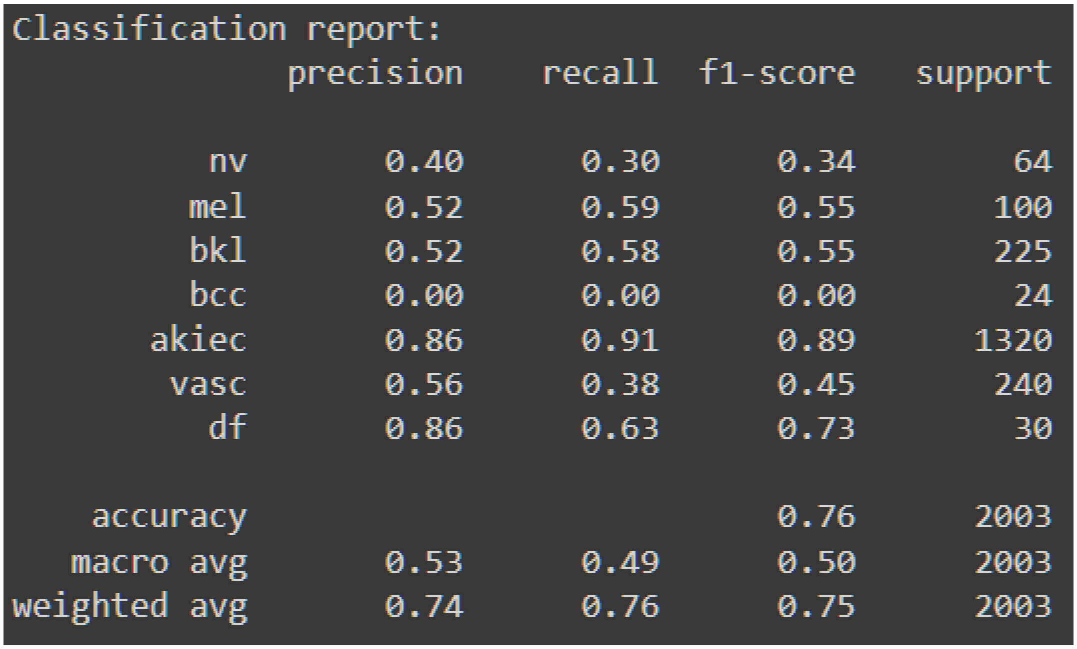
I was able to make time to train another model in a different way. The dataset provided us a couple different options for training. The first two models we trained are done so by looking at the images directly. Now we will see our results from taking advantage of the csv file provided to us that contains RGB image values for each image, which has already been resized for training use.

Model 3 Results:

- Highest Accuracy: 0.9803401231765747



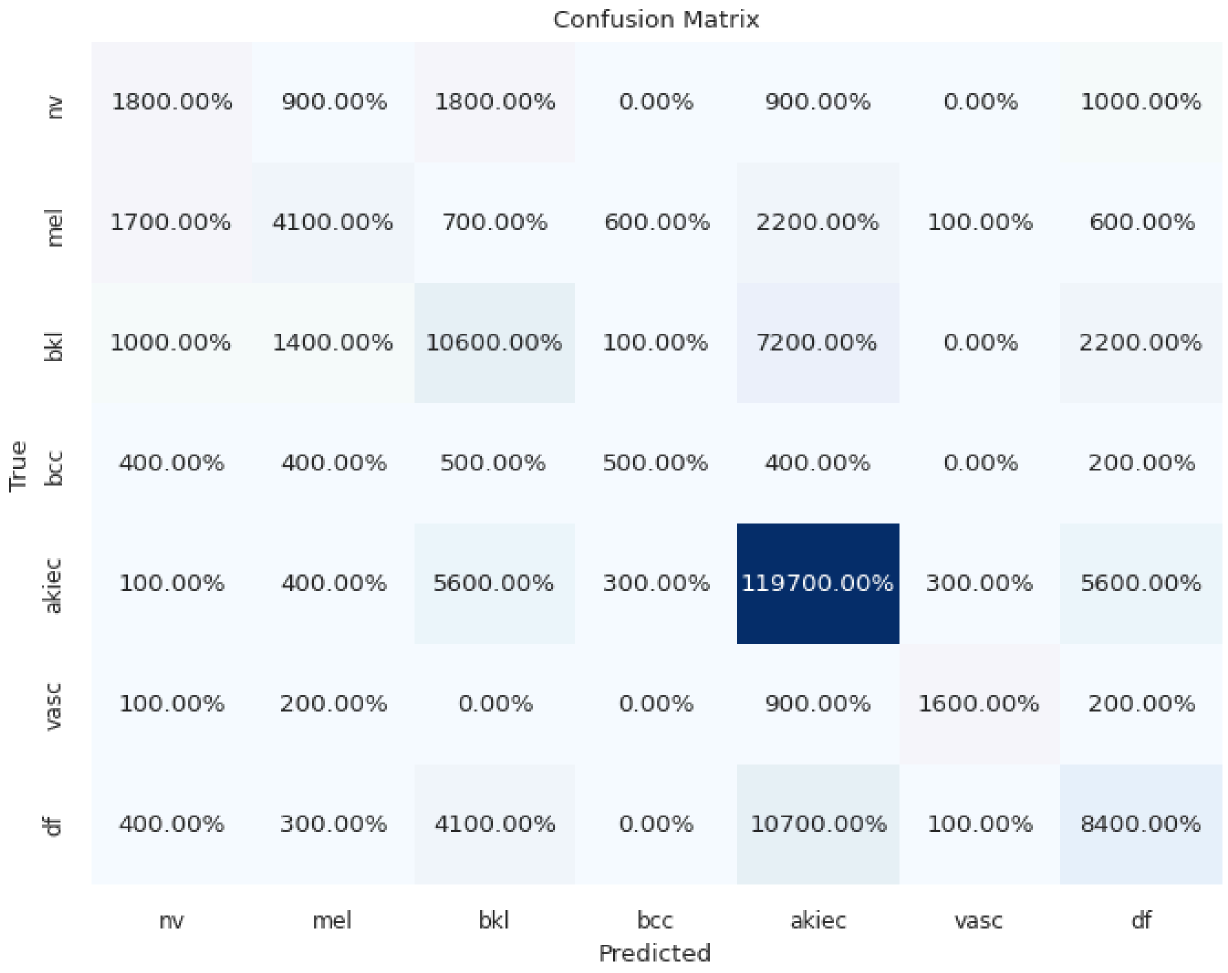




```
Classification report:
              precision    recall  f1-score   support

          nv       0.40      0.30      0.34        64
         mel       0.52      0.59      0.55       100
         bkl       0.52      0.58      0.55       225
         bcc       0.00      0.00      0.00        24
       akiec       0.86      0.91      0.89      1320
        vasc       0.56      0.38      0.45       240
          df       0.86      0.63      0.73        30

    accuracy                           0.76      2003
   macro avg       0.53      0.49      0.50      2003
weighted avg       0.74      0.76      0.75      2003
```



True \ Predicted	nv	mel	bkl	bcc	akiec	vasc	df
nv	1800.00%	900.00%	1800.00%	0.00%	900.00%	0.00%	1000.00%
mel	1700.00%	4100.00%	700.00%	600.00%	2200.00%	100.00%	600.00%
bkl	1000.00%	1400.00%	10600.00%	100.00%	7200.00%	0.00%	2200.00%
bcc	400.00%	400.00%	500.00%	500.00%	400.00%	0.00%	200.00%
akiec	100.00%	400.00%	5600.00%	300.00%	119700.00%	300.00%	5600.00%
vasc	100.00%	200.00%	0.00%	0.00%	900.00%	1600.00%	200.00%
df	400.00%	300.00%	4100.00%	0.00%	10700.00%	100.00%	8400.00%

Summary and Future Work

In conclusion, this project showcased the potential of CNNs in skin cancer detection. The developed model exhibited strong performance, but further work is needed to enhance the current models. I learned a lot about how much computational power it takes to train a model on thousands of images. I was jumping between two GPUs, one locally and one in the cloud and it still took a long time for me to complete training of a model without any errors. I learned how difficult it can be to fine-tune and find the right hyperparameters, and how image data can be stored and represented in variety of formats.

I also have to respect companies who have figured out how to build the software for others to build models and train. It took many days and weeks just to get a model built and working, not to mention the nitpicking required to start seeing accuracy improve. The fact we have the tools to take advantage of to further explore and learn is quite impressive to me.

As for some suggestions for further work. I would like to create a model that achieves reliable high accuracy that I could then save and export into an App. This would allow users to upload images for diagnosis and receive recommendations for further medical evaluation. At the time of writing, I haven’t been able to get it quite working. Since this report is due Tuesday, December 5th, and I don’t present until Thursday, December 7th, I may get the app working in time for my presentation. Who knows?

Another suggestion I have for further work would be a multi-class classification when distinguishing various types and stages of skin cancer. The dataset doesn’t specify stages of skin cancer, so it would be good practice for me to collect data from a source, label, and train on identifying stages of skin cancer.