Skin Lesion Classification:

Jake Pistotnik, Jinsoo Chung, Brian Tung
MIDS w207

REPORT:

Background

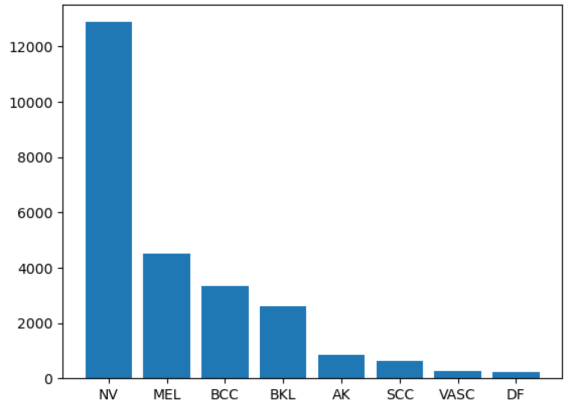
A skin lesion is any skin that has different characteristics compared to the skin around it. Most skin lesions are harmless although some may cause discomfort or pain. However, Melanoma, a type of skin cancer, may be fatal if not detected early. Melanoma can occur in any part of the body and may appear to be harmless at first, but can have fatal consequences. Traditionally, Melanoma is diagnosed by a dermatopathologist, or sent to a lab for testing. However, in many low income areas of the world, these resources are as widely available, making diagnosing melanoma more difficult. That being said, the shape, size, color, and growth profile of the skin lesion may give clues to whether or not the lesion is melanoma or not. Tools that aid in the identification of Melanoma can greatly improve patient outcomes and survival rate, as early detection and treatment of Melanoma has a 5 year survival rate of 99%.

A model like the one we have proposed can be used in order to help better diagnose skin lesions, especially in areas of the world where experts in skin lesion diagnoses are less prevalent. This will lead to better diagnostics, and thus better and earlier treatment of these skin lesions.

Explanation of the Data Set

The data used in this study is from the International Skin Imaging Collaboration (ISIC) Challenge in 2019, which is built upon data from previous years as well. This dataset contains 25,331 dermoscopic images for classification across 8 diagnostic categories of skin lesions, including Melanoma, Melanocytic nevus, Basal cell carcinoma, Actinic keratosis, Benign keratosis, Dermatofibroma, Vascular lesion, Squamous cell carcinoma, and other images with no values (ignored for our purposes).

As seen from the figure below, the dataset had an imbalanced number of images in each skin lesion class, with the largest number of images belonging to Melanocytic nevus (n=12,875). When we look at the breakdown of Melanoma versus non-Melanoma images, Melanoma images (n = 4,502) are much less represented compared to the totality of other classes (n = 20,809).



<u>Sampling Strategies:</u>

In order to deal with our imbalanced data in building our model, we explored multiple sampling strategies in order to form a more balanced data set. The reason that this is critical is because if we ran our model based off of a random sample of our dataset, without balancing our data between our two classes, our model will not have enough data on melanoma images to learn what they are, and will most likely overfit for our non-melanoma images, leading to an inaccurate model.

The first sampling technique we explored was to take an equal number of samples from each of our non-melanoma classes, with the sample size set to the number of images for our smallest class, Dermatofibroma, which had 239 image examples. We then sampled 1673 images from our melanoma class, which is equal to the number of samples we took from our non-melanoma classes (239*7). This is a form of undersampling from our non-melanoma images, and although our sample is no longer representative of the true distribution of diseases from our original data, it balances our dataset and creates equal representation for our CNN to learn how to classify between the two classes.

Another undersampling technique that we explored was a more simplified version of the above sampling method, where we split images up into their respective Melanoma vs. Non-melanoma classes, and then sampled 2261 images (half of the total number of melanoma images) from each class. The reason we chose this sample size was due to memory and RAM constraints while augmenting our data set and running our CNN model. Another note about this sampling technique is that it ignores the subtypes of the non-melanoma class, giving us a more accurate representation of our non-melanoma class and allowing our model to become more accurate on subtypes like Melanocytic nevus, Basal cell carcinoma, and Benign keratosis.

The last sampling technique we explored was using SMOTE sampling in order to overfit our melanoma class by using a KNN classifier to generate more melanoma image examples. Although SMOTE is traditionally a good method for overfitting underrepresented data, in our case, it was not the best technique as our biggest constraint was our available RAM (due to the number of images we had and the size of those images, and with overfitting our melanoma class, we ran out available RAM and were unable to train our model.

In our final analysis, we chose to utilize our second undersampling technique of splitting our data into two classes and taking even samples with size 2261 in order to conduct our analysis, as it gave us the highest model accuracy and leveraged the largest number of images to train our model with.

Data Preprocessing

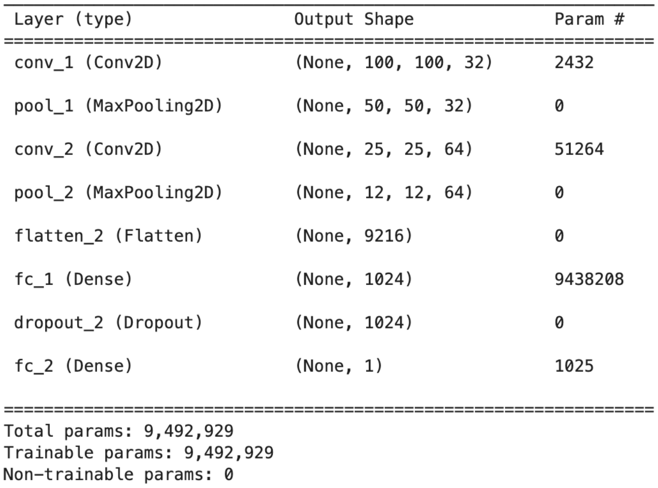
Prior to applying machine learning algorithms, it is crucial to assess the quality of the dataset through a variety of image augmentation and transformation techniques. Our analysis attempts to capture the significance of data quality on the ability for models to learn by comparing the model results of augmented data and non-augmented data.

The preprocessing begins with splitting the sampled dataset into a 60:20:20 split for train, validation, and test sets. For all data splits, we applied image transformation by resizing all images to 200x200 pixels and converted images to grayscale by rescaling all pixel values to (0,1). The original image is a larger 300x225 pixels, so resizing the image is critical for optimizing the model's training efficiency. Additionally, implementing grayscale images further helps to simplify the algorithm and reduces computational requirements.

All image augmentation techniques were applied on the training dataset split only and are tuned to default values as follows: **1)** Delta (brightness) = 0.5; **2)** Contrast Factor = 4; **3)** Left right random flip. We anticipate that applying image augmentation techniques will improve the performance of our model by increasing the amount of training data to learn our model. A larger amount of training data could also force the model to generalize all samples rather than overfit.

CNN

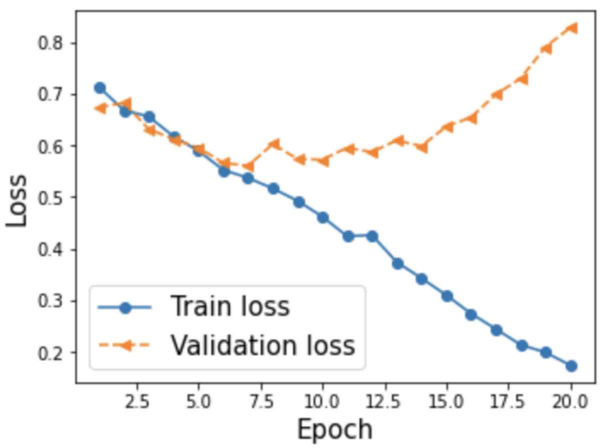
We decided to utilize a convolutional neural network (CNN) as our first machine learning model due to the high-dimensionality of image classification problems, where each pixel is considered a feature. CNN's convolutional layers are highly effective in reducing the number of parameters and complexity of images without losing on the quality of the model, which will improve computational efficiency.

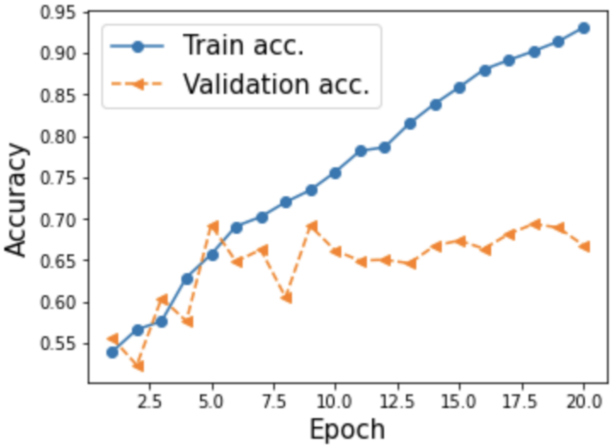


```
Layer (type)                 Output Shape              Param #
=================================================================
conv_1 (Conv2D)              (None, 100, 100, 32)      2432

pool_1 (MaxPooling2D)        (None, 50, 50, 32)        0

conv_2 (Conv2D)              (None, 25, 25, 64)        51264

pool_2 (MaxPooling2D)        (None, 12, 12, 64)        0

flatten_2 (Flatten)          (None, 9216)              0

fc_1 (Dense)                 (None, 1024)              9438208

dropout_2 (Dropout)          (None, 1024)              0

fc_2 (Dense)                 (None, 1)                 1025

=================================================================
Total params: 9,492,929
Trainable params: 9,492,929
Non-trainable params: 0
```

In deciding the model specifications, we utilized common hyperparameter values used in image classification: 1) Optimizer: Adam is one of the most widely adopted optimizers due to its characteristics as a computationally efficient and effective way to deal with large numbers of parameters. 2) Learning rate: Initially set to 0.001 and determined via hyperparameter tuning. 3) Loss Function: Binary classification calculates the loss value using "binary cross-entropy". 4) Epochs: The epoch count is 20. We found that our model began to overfit at around ~10 epochs, which is ultimately why we decided to reinforce our study with additional ML algorithms.

- Optimizer: Adam
- Learning rate: 0.001
- Loss function: Binary Crossentropy
- Epoch: 20

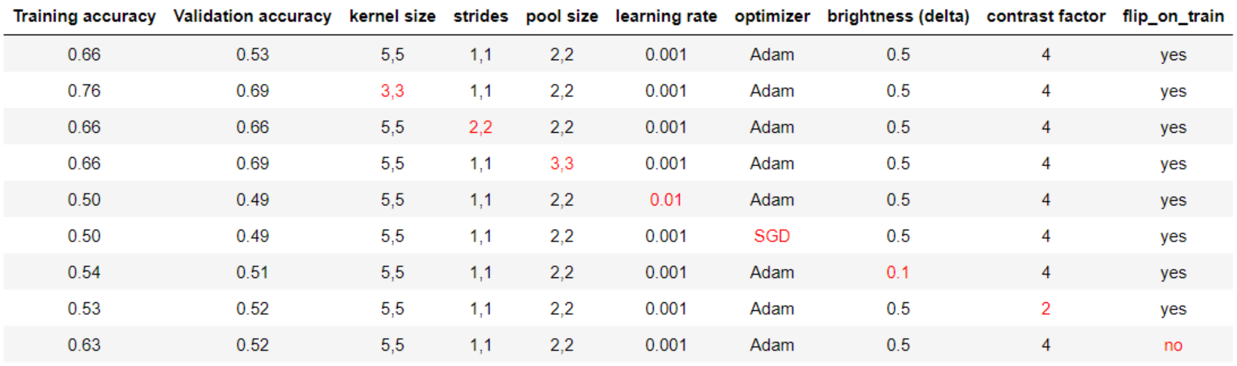




As expected, we achieved better model performance by training the model on augmented and transformed images compared to the original unprocessed dataset. Our CNN model was able to achieve a test accuracy of 68.51% for the augmented data. On the contrary, the model only achieved a test accuracy of 64.00% for the unprocessed data, which is a fair improvement over the 48.83% baseline prediction for “not melanoma”.

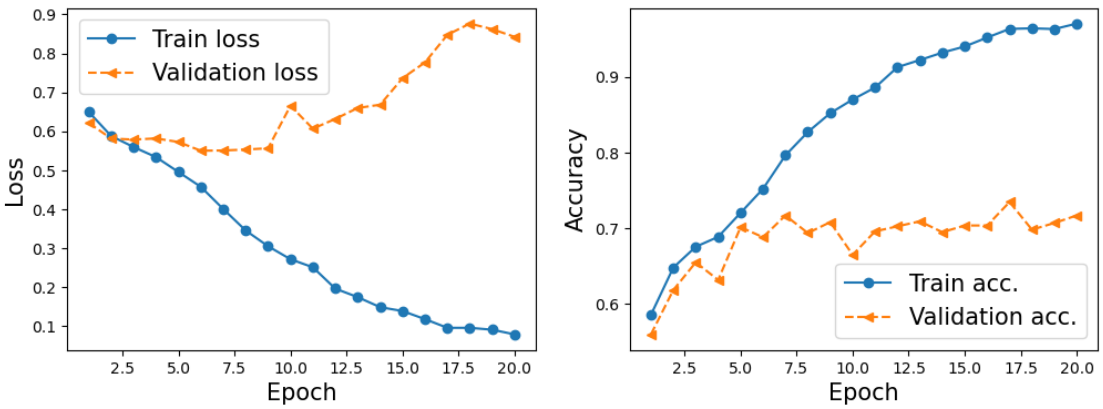
In order to improve the accuracy of the results, we ran through different hyperparameters to see how they would improve the training and validation accuracy of the results. We decided to tune one parameter at a time to check how they would improve or change the results from that of the baseline parameters mentioned above. For the analysis, we reduced the number of epochs to 5 as the baseline model was overfitting.

    - *Tuning Parameters:*
        - *Kernel Size = (3,3)*
        - *Strides = (2,2)*
        - *Pool Size = (3,3)*
        - *Learning Rate = 0.01*
        - *Optimizer = ‘SGD’*
        - *Brightness/Contrast Factor/Flip*

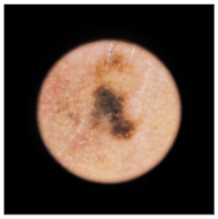


Training accuracy	Validation accuracy	kernel size	strides	pool size	learning rate	optimizer	brightness (delta)	contrast factor	flip_on_train
0.66	0.53	5,5	1,1	2,2	0.001	Adam	0.5	4	yes
0.76	0.69	3,3	1,1	2,2	0.001	Adam	0.5	4	yes
0.66	0.66	5,5	2,2	2,2	0.001	Adam	0.5	4	yes
0.66	0.69	5,5	1,1	3,3	0.001	Adam	0.5	4	yes
0.50	0.49	5,5	1,1	2,2	0.01	Adam	0.5	4	yes
0.50	0.49	5,5	1,1	2,2	0.001	SGD	0.5	4	yes
0.54	0.51	5,5	1,1	2,2	0.001	Adam	0.1	4	yes
0.53	0.52	5,5	1,1	2,2	0.001	Adam	0.5	2	yes
0.63	0.52	5,5	1,1	2,2	0.001	Adam	0.5	4	no

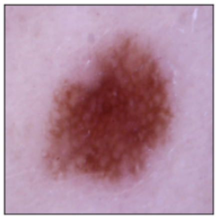
The final model used for CNN was updated with a kernel size of (3,3) and pool size of (3,3) after finding that the two hyperparameters helped with the results (See table above). The final model had a training accuracy of 97%, validation accuracy of 72%, and a test accuracy of 71%, which was an improvement over the previous test accuracy of 64% (No Hyperparameter Tuninig).



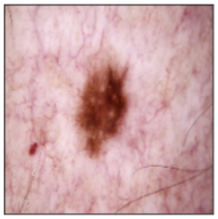
Some of the limitations we experienced while training the CNN models came from the images themselves. As seen from the images below, the images don't have a set orientation, some are surrounded by dark space, and some have hair in the images.



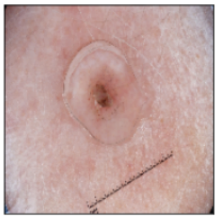
GT: no mel
Pr(mel)=87%



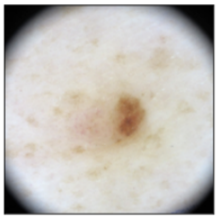
GT: no mel
Pr(mel)=38%



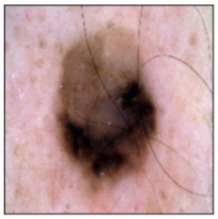
GT: mel
Pr(mel)=27%



GT: no mel
Pr(mel)=10%



GT: mel
Pr(mel)=45%



GT: mel
Pr(mel)=73%

It appears that our model does a good job of classifying non-melanoma skin lesions that are lighter in color, but tends to classify all images that have a lot of darkness (whether due to the color of the skin lesion or the existence of hairs in the photos). To improve the model moving forward, it would be best to shave the area with the skin lesion as much as possible to reduce hair in the images, and implement a standardized system for how the photos of the skin lesions are taken.

Random Forest (Jake):

Following the building of our CNN, we wanted to see if we could build a separate model that leveraged our metadata in order to predict the type of skin lesion. As we saw in our EDA, there appeared to be some interesting relationships between patient sex and the type of skin lesion, their age and the type of skin lesion, and most of all the anatomic location and the type of skin lesion. In order to more deeply explore this data and use it to classify our images, we decided to utilize a Random Forest Decision Tree. We decided on using a Random forest due to their ability to drastically reduce overfitting as compared to our CNN model while limiting any increases in error that are due to bias.

In order to actually build out our Random Forest Decision Tree, we had to extract image data from 11,021 images in our dataset and import them into a Collab Notebook for further analysis, as Tensorflow's Random Forest Decision Tree Library is only compatible with Mac OS x 12.03 operating software.

Our Random Forest Model was based on both our metadata (sex, age approximation, anatomic location) as well as out image data, and was able to utilize over 11,000 datapoints in order to produce a decision tree of depth 226 that was able to accurately classify between melanoma and nonmelanoma images 79.14% of the time in our test dataset. This model ended up being more accurate in terms of classifying melanoma as compared to our CNN. This is due to the random forest decision trees ability to consider our metadata in conjunction with our image data. We believe that without the RAM constraints that limited the size of the dataset we used to train our CNN, we could achieve similarly accurate, if not more accurate results.

Consequences (Round out):

Overall, although we were able to more accurately classify between melanoma and non melanoma skin lesions much better than our baseline model (only predicting non-Melanoma), we found that our model was not nearly accurate enough for practical purposes in terms of helping diagnose melanoma in the medical field. In order to increase accuracy, we would need to take the measures listed above such as having a more uniform dataset of images, making sure all lesion images have no obstructions and are taken in a similar fashion, and having access to more computing power. A model like this, however, may help doctors make a more informed baseline decision when looking at the skin lesion image. When looking at the image in conjunction with the prediction probability that the model assigns to the image, doctors can get a sense as to what the model thinks, keeping in mind where the model struggles (dark spots, obstructions) and use that to help inform their diagnosis.

Reference

1. https://www.healthline.com/health/skin-lesions
2. https://www.mayoclinic.org/diseases-conditions/melanoma/symptoms-causes/syc-20374884
3. https://www.cancer.org/cancer/melanoma-skin-cancer/detection-diagnosis-staging/survival-rates-for-melanoma-skin-cancer-by-stage.html
4. https://towardsdatascience.com/decision-trees-and-random-forests-df0c3123f991#:~:text=A%20random%20forest%20is%20simply,different%20samples%20of%20the%20data.