Image Classification for detection of Melanoma in teledermatology

Michael Sarubbe
Department of Electrical Engineering and Computer Science
Florida Atlantic University
Boca Raton, FL 33431
Email: msarubbe2017@fau.edu

Abstract— In an increasingly online era, the need for tools to help us adapt to this shift to remote meetings from in person ones are very high. Using deep learning tools, development of a classifier to identify a skin lesion as either benign or malignant can help doctors and patients in the teledermatology process by providing another layer of verification of a diagnosis. This study uses three different predefined convolutional neural network (CNN) models, resnet50, resnet 18, and GoogLeNet. These three CNN's of different architecture and complexity will be compared against each other to evaluate what model performs best as a tool for classification.

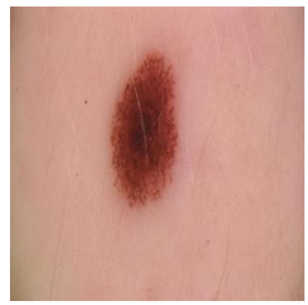
Keywords— Classification, benign, malignant, melanoma, CNN, resnet 50, resnet18, GoogLeNet

I. INTRODUCTION

In this report, two different CNN's will be tested in MATLAB to classify images of skin lesions as either benign or malignant. The skin lesions labeled as malignant are pictures of melanoma. Looking at different examples of the two, it is hard for a person to decide if the mark on their skin is cancerous or not. Figures 1 and 2 shown below.

Figure 1: Benign Skin Lesion Example

Figure 2: Malignant Skin Lesion Example (Melanoma)



Without having extensive knowledge of dermatology, it is hard to distinguish between the two. In most cases, people would go to see their dermatologist to get an opinion on the skin lesion. However, in the event of a pandemic like Covid-19, sometimes meeting these doctors in person is not an option. This is just one reason why advancements in the field of teledermatology are necessary. Teledermatology is a field that encompasses the study of dermatology, combined with telecommunications. One way of evaluating an issue with a patient's skin is by using an image. Due to many different factors, a dermatologist may not be as accurate in their diagnosis over a video call as they would be in person. The use of a classification tool could help validate the opinion of the doctor. For example, if the doctor says that the skin lesion is malignant, and then the classifier says the lesion is malignant as well, then the doctor will have more confidence in their answer, having it validated by another source.

This study attempts to implement a basic version of this classifier, using three different CNN models. The models used are resnet50, resnet18, and GoogLeNet. The performance of these three models will then be compared and evaluated. The goal of this study is to show proof of concept for implementing this classifier, and to show which neural network architecture is best suited for the task. This experiment was performed using MATLAB, on a laptop with Windows 10, with a NVIDIA GTX 1050Ti Graphics card to perform the network training.

II. ACQUISITION OF DATA

This data was sourced from Kaggle. The dataset used is called the “Melanoma Skin Cancer Dataset of 10000 Images.” It has testing and training data that adds up to 10000 total images of examples of skin lesions that are labeled as either benign or malignant. The images come standardized as RGB images with a size of 300x300. All of the network architectures we use have a standardized image input layer size of 224x224. So for the purposes of this experiment, all images from this dataset are rescaled to this size. Since this is the same ratio of pixels for both dimensions, the image quality is mostly retained.

Due to the length of training time required, only 40% of the training and testing images were used for training and validation of the model. However, since the dataset was 10000 images in total, we were still left with a large amount of images to work with. The unused images were left to later have the trained model test on as unseen data.

III. METHODOLOGY

The MATLAB code begins by loading the layer graph of the desired network. Then, the images are loaded into a datastore, and partitioned into training and testing datasets. A function is used that when images are read from the datastore, they are resized to 224x224. The training options for the neural network are then defined. These options are sourced from one of Dr.Marques' Assignments.

Figure 3: CNN Training Options

```
options = trainingOptions('sgdm', ...
    'MiniBatchSize',10, ...
    'MaxEpochs',4, ...
    'InitialLearnRate',1e-4, ...
    'Shuffle','every-epoch', ...
    'ValidationData',testingDatastore, ...
    'ValidationFrequency',3, ...
    'Verbose',false, ...
    'Plots','training-progress');
```

Then before training is started, the final few layers of the network architecture are redefined to incorporate the number of classes we are trying to identify. In our case, this is two, benign and malignant. A function from a MATLAB article called “Training Deep Learning Network to Classify New Images” [2] is utilized to do this. The last learnable layer,as well as the output layer are modified to fit our need for a binary classifier. Then, by using the function from the MATLAB Deep Learning Toolbox to train the network with the desired layers and options, the model began to iterate through its optimization process. This took a varying amount of time for each different model on the computer described earlier.

After the model was trained, the accuracy was presented from the validation phase. Then the portion of the dataset not used in testing is used as unseen data to further test the model on.

IV. RESULTS

Between the three networks, performance was pretty similar. After being trained, each network was tested on two different sets of validation data. The performance across these different sets were similar on the models as well.

This table below shows the classification accuracy of the different models on one set of validation data that was used during the training phase, and a totally unseen set of data, which was randomly sourced from the image dataset. These will be titled as data_unseen and data_unseen_random, along with accuracy_unseen and accuracy_unseen_dandom, respectively.

Figure 4: Accuracy among models

	accuracy_unseen	accuracy_unseen random
resnet50	0.8450	0.8600
resnet18	0.8375	0.8717
GoogLeNet	0.8650	0.8867

The results from these measures of accuracy show that the

GoogLeNet model outperforms both versions of the resnet models. The GoogLeNet is 22 layers deep [4] , vs the resnet 18 being 18 layers [5] , and resnet 50 being 50 layers [6]. This shows to me that the GoogLeNet model is the superior architecture. Being tested on such a large set of images confirms the validity of this accuracy. The GoogLeNet model being a more lightweight model than either of the resnet models, means that less resources will be needed to develop further research with this CNN architecture.

V. SUMMARY AND POTENTIAL FUTURE WORK

This entry-level experiment into making an image classifier for images of potential Melanoma instances showed that the GoogLeNet CNN architecture performs the best in correctly classifying images of a skin lesion as either benign or malignant (containing melanoma).

There are many ways this work could be capitalized upon and expanded into other research. Such as trying this classifier with video data, to see if accuracy is maintained when not always provided with a perfect image or set of images. As well as other image pre-processing techniques such as image segmentation and colorization can be implemented. These pre-processing techniques may skew the results of which model performs the best, but might allow for better performance overall.

Making this classifier work with video data instead of image data would make this more practical to work with in real time. In the field of teledermatology, it is likely that the patient will be on a live video call with their doctor. Being able to embed the classifier to read this live video feed makes less effort required on the patient's end. Since Melanoma can affect people of all backgrounds, not everybody affected may be proficient with technology. This means that some patients may have a hard time using a camera to upload an image file. Using the same video feed they are speaking to their doctor on would streamline the process.

ACKNOWLEDGMENT

I would like to thank my professor, Dr. Oge Marques, for allowing me the opportunity and providing the resources possible for me to do this research. I would also like to thank Florida Atlantic University, for providing me with the MATLAB license necessary to run this code.

REFERENCES

[1]https://www.kaggle.com/datasets/hasnainjaved/melanoma-skin-cancer-dataset-of-10000-images

[2] https://www.mathworks.com/help/deeplearning/ug/train-deep-learning-network-to-classify-new-images.html

[3] https://www.mathworks.com/help/deeplearning/ug/exp-mgr-weight-init-example.html

[4] https://www.mathworks.com/help/deeplearning/ref/googlenet.html

[5] https://www.mathworks.com/help/deeplearning/ref/resnet18.html

[6] https://www.mathworks.com/help/deeplearning/ref/resnet50.html