Melanoma Detection Using Neural Networks

Indiana University Bloomington
ENG533 - Deep Learning Systems
December 26, 2019

Naimesh Chaudhari[1], Tyler Peterson[2], and Akshat Lakhiwal[3]

***Abstract*— Melanoma is a malignant form of skin cancer that is identified by medical providers. The diagnosis of melanoma typically begins with a visual assessment of the skin, which means this stage of the heuristic process may lend itself well to using computer vision technologies to assist providers in forming a diagnosis. The goal of this project is to train a neural network capable of performing image classification which can be used as a form of augmented intelligence that seeks to assist providers, as opposed to replace providers. To accomplish this goal of providing an assistive technology, we do not require the model to render a prediction if it displays an elevated level of uncertainty. To create this model, we use an image data set provided by the International Skin Imaging Collaboration. The data set contains 25,341 images of skin lesions. Using these images, we develop a model that predicts whether an image is of malignant melanoma skin lesion or a non-malignant lesion, and in the case of high uncertainty, no prediction is returned at all. Source code for this project can can found within the GitHub repository. https://github.com/petersontylerd/deep-learning-systems-fall-2019/.**

I. INTRODUCTION

Melanoma is a malignant and dangerous form of skin cancer. The disease develops from pigment-containing cells known as melanocytes. In its initial stages, the disease primarily manifests as a skin lesion. If identified and treated early, survival is likely. If the condition is not identified early, the disease can spread to other areas of the human body, which complicates treatment and increases the likelihood of the patient dying from the disease. Diagnosis of melanoma typically begins with a dermatologist, who identifies a suspicious lesion through a visual assessment of the skin. A portion of the lesion is biopsied and a pathologist assesses the biopsied tissue to confirm that the lesion is melanoma.

Since the initial identification of melanoma is visual in nature, it is conceivable that computer vision may play a beneficial role in assisting health care providers with making accurate decisions. The importance of accurate predictions is particularly high in this application due to the danger of this form of cancer. Minimizing false negatives is the primary goal because cancerous lesions must be identified so that treatment can be undertaken quickly. That being said, incorrectly identifying non-cancerous lesions as melanoma should also be minimized, as false positives lead to unnecessary anxiety for the patient and unnecessary medical expenses for both the patient and health system.

Therefore, we do not limit this computer vision application to a straightforward binary classification task of melanoma versus non-melanoma. We allow the model to indicate that it has a low level of confidence because we do not want to mislead users of this application. Our goal is develop a model that simultaneously accomplishes two aims. First, we want to maximize true positives and true negatives. Second, we want to minimize the percent of samples that receive an uncertain prediction.

We pursue this goal by utilizing convolutional neural networks (CNN). To find the optimal solution, our experiments include the utilization of home-grown CNNs, as well as pre-trained networks, which we customize through transfer learning. To optimize the algorithm, we use a multitude of different optimizers along side a cross entropy loss function. To evaluate the performance of our models, we focus on maximizing classification accuracy while also monitoring the level of uncertainty that the models experience when evaluating each sample. In addition to model accuracy, we also evaluate the model's F-score, precision, and recall to better understand the presence of false negatives and false positives. The formulas for accuracy[1], precision[2], recall[3], and f1 score[4] are displayed below:

$$Accuracy = \frac{\text{TP + TN}}{\text{TP + FP + TN + FN}} \tag{1}$$

$$Precision = \frac{\text{TP}}{\text{TP + FP}} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{4}$$

This project utilizes the Python implementations of the TensorFlow and Keras libraries. All models were trained on a GPU within the Google Cloud platform.

II. DATA DESCRIPTION

The International Skin Imaging Collaboration (ISIC) is an academia and industry partnership designed to facilitate application of digital skin imaging to help reduce melanoma mortality. ISIC makes available an image data set containing 25,431 images of skin lesions. Specifically, the data set contains 4,532 images of malignant melanoma skin lesions and 20,809 of other types of skin lesions that are non-malignant. The data may be obtained through the website https://www.isic-archive.com/. Below are examples of two images with their respective classes:

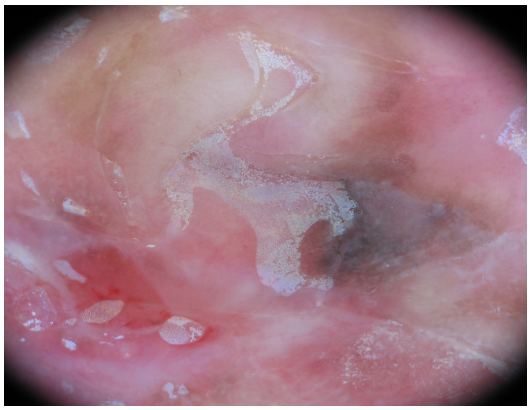

Fig. 1: Melanoma

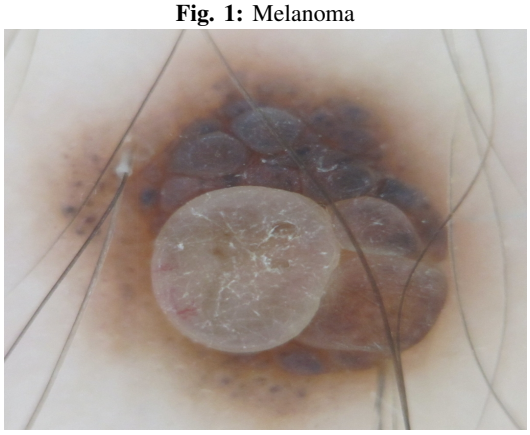

Fig. 2: Non-melanoma

III. Strategy

Our strategy can be separated into a data processing phase, a model development phase, a training/evaluation phase and an uncertainty detection phase. In the data processing phase, images are scaled such that each value in each color channel is represented by a number between 0 and 1, randomly rotated with a max rotation of 270 degrees, and randomly flipped horizontally. These transformations have the effect of multiplying the total number images evaluated by the models because each image is likely to appear differently in each epoch due to the random data augmentations. This increases a model's ability to learn, lowers a model's tendency to overfit the training data, and increased a model's tendency to accurately predict the class of unseen data. In addition to data augmentation, we also seek to address the class imbalance present in the data through upsampling. Melanoma images constitute only 17.9 percent of the data set, but upsampling generates synthetic copies of the available melanoma images and randomly samples with replacement to create a 50 percent balance between the two classes. This balanced data set is split into a training data set and testing data set, where the test data set is composed of 30 percent of the full upsampled data set.

In the model development phase, two distinct approaches are undertaken. First, we attempt to construct a custom, homegrown CNN from scratch. Second, we utilize pre-trained models based on state-of-the-art CNN architectures, which are modified to suit our specific task. This is also known as transfer learning. The pre-defined CNN architectures utilized during our transfer learning experiments are Resnet18, Resnet34, DenseNet201, MobileNet, and NASnet.

In the training/evaluation phase, we monitor the train data set accuracy and loss, as well as the test data set accuracy and loss. We train each model for a minimum of 20 epochs.

After identifying the best model through our experiments, we leverage the model's dropout layers to perform Monte Carlo simulations on a sub-sample of the test data, allowing us to detect uncertainty. In standard practice, dropout is used during the training phase to help the model avoid overfitting the data. It does so by randomly suppressing a certain proportion of the model's weights. Dropout can be conceptualized as forcing the model to work through the disadvantage of fewer nodes, and due to the random nature of the node suppression, dropout effectively produces many different models that are training to accomplish the same task. This is also known as ensembling.

To capture uncertainty, we perform N forward passes on each individual training sample while keeping dropout enabled. This contrasts with the typical use of dropout during the test phase, when dropout is disabled and the full model is used to generate predictions. If the model consistently returns high probabilities for one particular class for a given sample, then we regard this as a prediction of which the model is fairly certain. On the other hand, if the model consistently returns low probabilities and potentially even different class predictions during the N trials, we regard this as a prediction of which the model is uncertain. To evaluate this, we calculate summary statistics describing the model's probabilities for each samples, and compliment this with visualizations of the probabilities distributions.

IV. Experiments: Custom Model

The architecture of the custom model is below:

```
Model: "model_10"
_________________________________________________________________
Layer (type)
=================================================================
input_11 (InputLayer)
_________________________________________________________________
conv2d_16 (Conv2D)
_________________________________________________________________
conv2d_17 (Conv2D)
_________________________________________________________________
max_pooling2d_8 (MaxPooling2
_________________________________________________________________
conv2d_18 (Conv2D)
_________________________________________________________________
conv2d_19 (Conv2D)
_________________________________________________________________
max_pooling2d_9 (MaxPooling2
_________________________________________________________________
flatten_4 (Flatten)
_________________________________________________________________
dense_18 (Dense)
_________________________________________________________________
dropout_7 (Dropout)
_________________________________________________________________
dense_19 (Dense)
_________________________________________________________________
dense_20 (Dense)
=================================================================
Total params: 8,957,098
Trainable params: 8,957,098
Non-trainable params: 0
```

Fig. 3: Architecture of the best custom model

The model contains 8,857,098 trainable parameters. We use cross-entropy as our loss function and RMSprop as the optimizer during the training phase. The model is trained for 20 epochs. The accuracy and loss metrics of the train and validation data are displayed on figure 4.

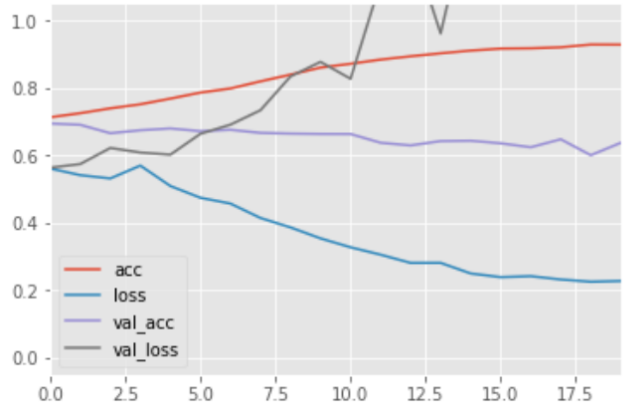

Fig. 4: Best Custom Model Results

We did not get promising results when testing this model. As we can see from the results, although the train accuracy increases and the loss decreased, the performance on the test data is poor. We observe that the test loss increases dramatically and the test accuracy decreases with each epoch.

V. Experiments:Transfer Learning

Our experimentation with transfer learning yields much more promising results. We evaluated several pre-trained models, but found that the model trained with DenseNet-201 architecture provides the best results for this task. The architecture of DenseNet-201 is shown in figure 5, and the results are illustrated in figure 6.

Layers	Output Size	DenseNet-169	DenseNet-201
Convolution	112 × 112	7 × 7 conv, stride 2	
Pooling	56 × 56	3 × 3 max pool, stride 2	
Dense Block (1)	56 × 56	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56 × 56	1 × 1 conv	
	28 × 28	2 × 2 average pool, stride 2	
Dense Block (2)	28 × 28	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28 × 28	1 × 1 conv	
	14 × 14	2 × 2 average pool, stride 2	
Dense Block (3)	14 × 14	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 48$
Transition Layer (3)	14 × 14	1 × 1 conv	
	7 × 7	2 × 2 average pool, stride 2	
Dense Block (4)	7 × 7	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1\times1 \text{ conv} \\ 3\times3 \text{ conv} \end{bmatrix} \times 32$
Classification Layer	1 × 1	7 × 7 global average pool	
		1000D fully-connected, softmax	

Fig. 5: Architecture of DenseNet-201

With this model we achieve 80 percent accuracy on the test data set. Figure 6 visualizes the loss and accuracy for both the train and test data. The loss for both data sets increases gradually with each epoch, and the accuracy increases for both data sets as well. Given this model's performance, we use this model for further evaluation on the test data set by reviewing the precision, recall, and f1 score metrics, as well as detection of uncertainty.

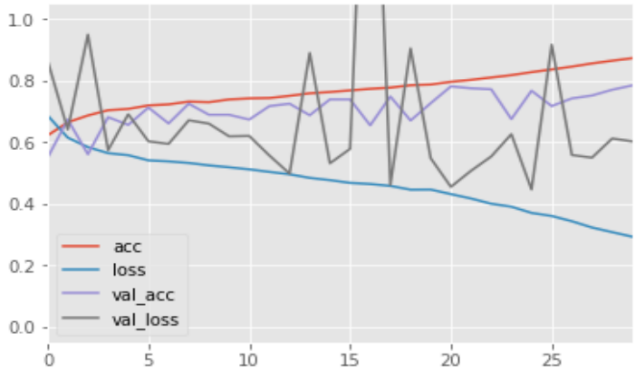

Fig. 6: DenseNet-201 Results

VI. Uncertainty, Precision, Recall, F1 Score

We evaluate the performance of our DensetNet-201 model on 100 randomly sampled test set images. These images are stratified such that the distribution of melanoma vs non-melanoma images is identical to the training data set. With this data, we perform two evaluations. First, we calculate precision, recall and the f1 score prior to removing any images of which the model is uncertain. Second, we determine which images indicate uncertainty in the model, remove those images, and recalculate precision, recall and the f1 score.

When testing against all 100 test set images, we achieve an accuracy score of 80 percent. The model also achieves a recall rate of 74.1 percent, a precision rate of 66.7 percent, and a overall f1 score of 70.2 percent. The ROC curve for these image predictions is illustrated in figure 7.

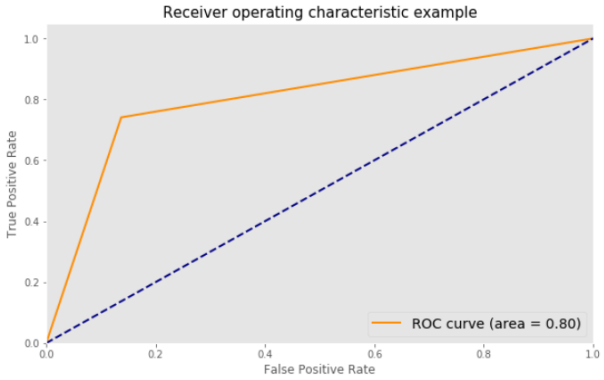

Fig. 7: ROC Curve of the Sub-Sampled Test Data

To ascertain the model's uncertainty on individual images, we used the Monte Carlo simulation method described in the strategy section. Below, we visualize samples of images of which the model is highly certain, as well as samples of images of which the model is highly uncertain.

Figures 8 and 9 illustrate that when the model is quite certain about its predictions, the mean value of the prediction

percentages tends to be closer to the edges of the probability range and the standard deviation to be small. When the model is uncertain about its prediction the probabilities tend to have a mean closer to the middle of the probability range and a higher standard deviation.

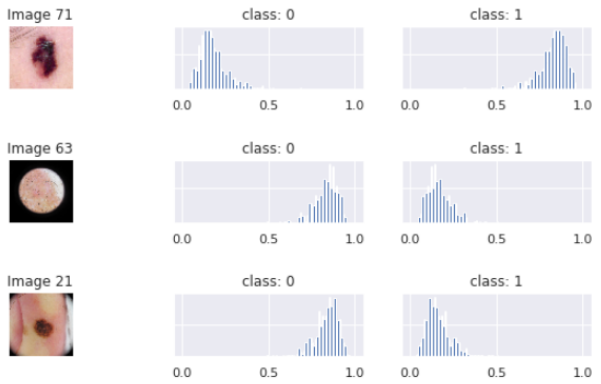

Fig. 8: Model Confident In Its Predictions

Fig. 9: Model Not Confident In Its Predictions

We repeat out metric calculations after removing images which we deem to be samples that cause the model to exhibit uncertainty. We regard an image as a source of uncertainty if the mean probability of a given class is less than 95 percent for all N predictions for one image. In total, we removed 30 of the 100 images in the test set sample. With the uncertain samples removed, the model now achieves a recall rate of 81.2 percent, a precision rate of 86.7 percent, and a overall f1 score of 83.9 percent. The updated ROC curve illustrated in figure 10. Additionally, a confusion matrix is illustrated in figure 11.

VII. Conclusions

The model clearly improves its performance of detecting malignant melanoma when given an image of a skin lesion once images of which it is uncertain of how to classify have been set aside. We know this by observing the increase in the key metrics that we calculate before and after removing images with cause the model to display uncertainty. While removing 30 of the 100 images in our samples represents a

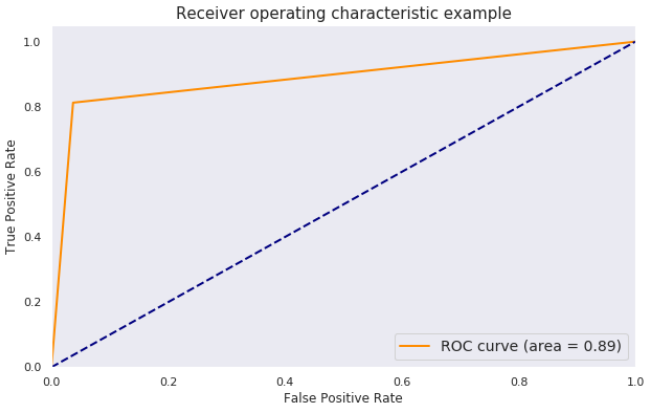

Fig. 10: ROC Curve, uncertain samples removed,

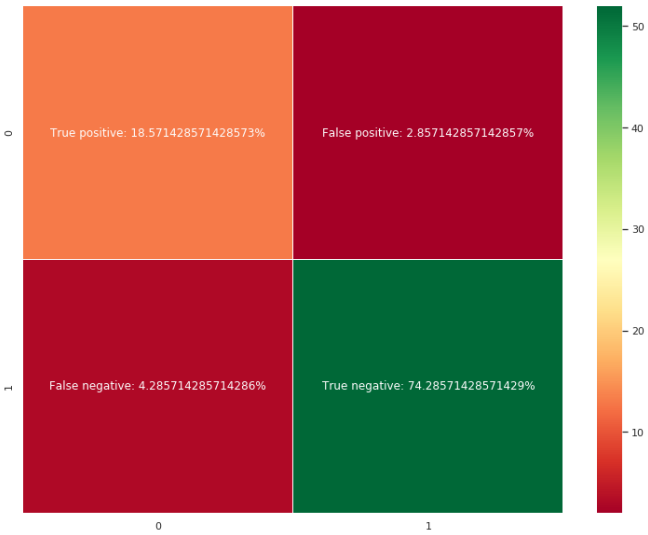

Fig. 11: Confusion Matrix, uncertain samples removed

30 percent uncertainty rate, and will therefore result in no prediction being provided, we find this to be acceptable given that the intent is for this application to be used alongside a trained medical professional as an augmented intelligence tool. This is consistent with the intent of our project from the outset - to provide a tool that can assist a trained medical professional and, most importantly, not mislead by forcing the model to make a decision. By accepting that we should not seek to replace medical professional, but instead improve the capacity to make an accurate decision, we have trained a model that can admit uncertainty and may prove to be a valuable tool when used in clinical practice.

References

[1] Y. Gal, "Uncertainty in deep learning," tech. rep., University of Cambridge, 2016.

[2] A. L. Mattia Segu and D. Scaramuzza, "A general framework for uncertainty estimation in deep learning," tech. rep., University of Zurich and ETH Zurich, Switzerland, 2019.

[3] S. E. Volodymyr Kuleshov, Nathan Fenner, "Accurate uncertainties for deep learning using calibrated regression," tech. rep., 2018.

[4] M. K. Murat Sensoy, Lance Kaplan, "Evidential deep learning to quantify classificationuncertainty," tech. rep., Ozyegin University, Turkey, US Army Research Lab, Melih Kandemir, 2018.

[5] Z. G. Yarin Gal, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” tech. rep., University of Cambridge, 2016.