Efficacy of a Convolutional Neural Network at Skin Cancer Detection

Ronan Akira Konishi

***Abstract*— With the increasing trend of big data, global smartphone subscriptions, and high processing power, the development of a smartphone application that uses convolutional neural network shows potential at providing effective and accessible skin cancer diagnosis. Despite recent successes at machine learning-based skin cancer detection, there is a lack of success when testing with smartphones. The following study tested the efficacy of a deep learning, feedforward, supervised learning, convolutional neural network that followed a LeNet model. After training on skin lesion images from the ISIC archive, the results were unfavorable. Nevertheless, this study provides a groundwork for future studies to follow and issues to avoid.**

I. Introduction

Skin cancer is prevalent in the United States, afflicting approximately one in five Americans by the age of 70 [1]. The annual number of patients diagnosed with skin cancer each year is greater than that of all other forms of cancers combined [2]. With such an alarmingly high rate, it is imperative that precautions are taken before the cancer can grow excessively dangerous. One significant factor in decreasing its lethality, especially when dealing with melanoma, is early diagnosis and treatment [3]. Although basal and squamous cell carcinoma are the most common and least deadly types skin cancers, early diagnosis is still important to alleviate excessive malignant tumor and lymph node growth, which can eventually lead to death [4].

Although getting regularly checked for skin cancer is extremely important in diagnosing skin cancer during its earlier stages, there are numerous factors that dissuade people from seeking aid. As stated in a Stanford news article, some patients live too far from a doctor, struggle financially, or lack the time to get the necessary checkups [5]. However, a novel solution to such problems may be possible in today's technologically growing society because of the rapidly increasing global usage and availability of smartphones. As predicted in the 2013 Ericsson Mobility Report [6], there will be a total of 9.3 billion mobile subscriptions in 2019, of which 5.6 billion will be smartphone subscriptions. Therefore, a smartphone application that can reliably perform skin cancer diagnosis at a performance level on par with (or even superior to) professional doctors will make early detection for skin cancer conveniently and globally accessible; artificial intelligence demonstrates great potential at attaining this high-performance level for diagnosis.

II. Motivation

In this age of data proliferation, there is a substantial reliance on computers to discern patterns within large sets of data with a level of efficiency and effectiveness unattainable by humans [7]. More so, in the medical field, artificial intelligence shows great potential in its ability to analyze big data and perform accurate medical diagnoses for patients' diseases [8]. In 2017, Kavya Kopparapu invented a smartphone application for diagnosing diabetic retinopathy (eye cancer) by using a convolutional neural network algorithm called ResNet-50 [9], [10]. The network was trained to detect anomalies in the eye using the data of approximately 34,000 retinal scans from the National Institutes of Health [9], [10]. She stated that "[of the] 415 million diabetics worldwide, one-third will develop retinopathy. Fifty percent will be undiagnosed. Of patients with severe forms, half will go blind in five years" [9], [10]. From a more general perspective, the idea of creating a smartphone application for diagnosing diabetic retinopathy can be applied to all types of diseases, which will lead to saving more money, time, and lives [11].

III. Related Works

Over the past two decades, there have been a multitude of methods used to classify skin cancer lesions. In 1995, a study on applying neural classifiers for classifying skin lesions using texture statistics found promising results in their studies [12]. Since then, there have been numerous other methods tested for classifying lesions such as statistics, K-nearest neighborhood, classification and regression trees, ADWAT methods, and artificial neural networks [13].

In a study performed by Stanford students, the use of a deep convolutional neural networks system to classify skin cancer was found to be very effective [11]. The system was able to classify skin cancer lesions by using only the pixels from the images and the labels for the diseases as input parameters into their program during training [11]. Using a dataset of 129,450 clinical images with 2,032 different diseases, the system tested for the most common and deadliest skin cancers: keratinocyte carcinomas vs benign seborrheic keratosis and malignant melanomas vs benign nevi, respectively [11]. It was found that the system performed as well as experts on both tasks, clearly demonstrating the capabilities of artificial intelligence in skin cancer classification [11].

Although studies indicate promising results with neural networks, there is a lack of success when applying these methods to practical situation, especially with smartphones. An analysis of 4 smartphone applications (written for the 2 most popular smartphone platforms) showed poor accuracy when tested to discern whether a given skin cancer lesion was melanoma or not [14]. The sensitivity of the applications ranged from 6.8% to 98.1%, while the specificity ranged from 30.4% to 93.7% [14]. The lowest accuracy was found for the application that used an automated algorithm to analyze the images [14].

Currently, there are multiple major research institutes hoping to apply their artificial intelligence techniques to smartphones. For example, the group of Stanford students who created the effective deep convolutional neural networks system for skin cancer detection claimed that "[it would] be relatively easy to transition the algorithm to mobile devices but there still needs to be further testing in a real-world clinical setting" [5]. In addition, NVIDIA, a company that built a deep learning algorithm that classifies skin lesions as being benign or malignant as accurately as a dermatologist, stated that "the team intends to continue testing its solution in real-world clinical settings before attempting to establish the algorithm as the basis for a mobile application" [15].

IV. PROPOSED METHOD

The method for classifying skin cancer moles was a deep learning, feedforward, convolutional neural network that used supervised learning [16] - [18]. The process was as follows: the information in the database was preprocessed by labeling, transforming, and sorting the data into a training and testing dataset; the newly formatted data from the training dataset was then fed into a deep learning classification model through which the neural network optimized its weights and biases [16], [18]; and finally the testing data passed through the trained model to evaluate the effectiveness of the model.

A. International Skin Cancer Imaging Collaboration (ISIC) archive

The ISIC skin cancer archive is a collection multiple sub-databases each containing both skin cancer lesion JPG files and a corresponding description of their malignancy, type of skin cancer, and other pertinent information in JSON files. The sub-databases used were MSK1, MSK2, MSK3, MSK4, MSK5, SONIC, UDA1, UDA2, and Dermoscopedia; they were later all combined into a single directory, which was the database used for this project [19].

B. Image Preprocessing

Image preprocessing refers to the transformation, normalization, labeling, and categorization of the ISIC dataset [20]. These images were transformed by scaling them to a uniform aspect ratio of 1 to 1 (length to width) and an equal dimension of 360 pixels by 360 pixels by 3 pixels (length by width by depth). This was necessary because the convolutional layers will be unable to properly convolve features over unequally dimensioned images. Then normalization was applied to the pixels of all images from a range of 0 through 255 to 0 through 1 in order to reduce computational complexity for the model cost, weights, and leaky RELU.

The purpose for using all three color channels (red, green, and blue) rather than a single grayscale channel was to accommodate if color was an indicator of skin cancer. Then, each image was labeled as either malignant or benign by extracting the description from each image's corresponding JSON file. Lastly, the labeled images were randomly categorized into one of two subsets labeled as either training or testing in a distribution ratio of approximately 75% and 25%, respectively. This was achieved by assigning each image and its respective label a randomized value ranging from 0.0 to 1.0 and placing the image in the testing subset if the value was greater than 0.25, otherwise placing it inside the training subset. Although this method is not the most accurate for equal distribution, in this case, such factor was not important. The distribution of datasets in a ratio of approximately 75% to 25% was based on Andrew Ng's recommendations [21].

C. Deep Learning-based Classification LeNet Model

As shown in figure 1, the deep learning convolutional neural network model chosen for this study was the LeNet model [22], which consists of two major parts: a feature extraction model with convolutional, Leaky RELU, and Max Pooling layers and a classification model containing a fully connected layer and an output layer. The model used a learning algorithm called stochastic gradient descent and backpropagation to optimize the model's weights and biases to attain a minimal model error.

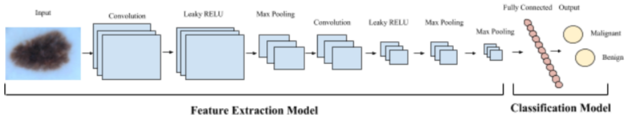

Figure 1: LeNet Model
Source: Adapted from [22]

1. *Feature Extraction Model*

I. *Convolutional Layer*

The purpose of the convolutional layer is to extract features from the input images that are indicative of being either malignant or benign. This layer consists of a equal number of filters and filtered output images. The filter convolves over each input images in a given m1ini-batch, in which the filter is layered over the images (the size determined by a set kernel size), and the dot product (product of each pair of overlapping pixels) is calculated. The sum of the summation of each dot product and the bias is then calculated to represent the entire cluster of pixel values from the input images as a single value on the newly filtered image. This calculation is performed over the images of the entire mini-batch starting from the top left cluster of pixels until it reaches the top right and shifting down a given number of pixels defined by the value of the vertical stride. Then the process is repeated for each filter, resulting in a number of filtered output images equal to the number of existing filters.

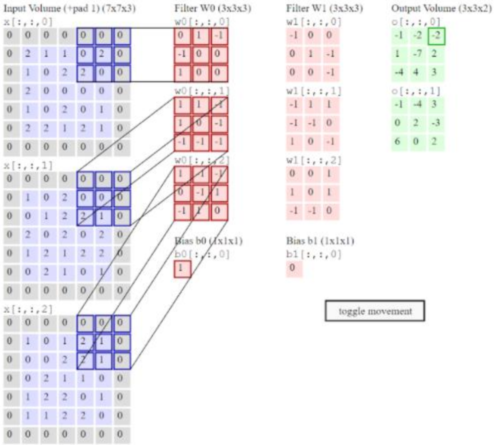

Figure 2: Convolutional Layer
Source: Adapted from [23]

II. Leaky RELU

The Leaky RELU layer attempts to minimize the effects of negative pixel values resulting from the calculations taken in the preceding convolutional layer. This layer starts with the top left most pixel to check if it is positive, in which case it does nothing, or negative, in which case the pixel value gets scaled to a value very close to zero. The reason for maintaining a slightly negative value for the pixel (which cannot practically exist), is because a study by Cornell University shows that by doing so, the neural network performs more accurately [24].

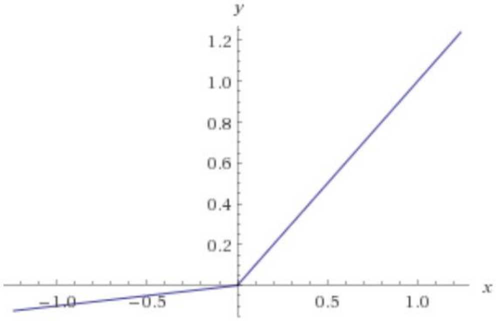

Figure 3: Leaky RELU
Source: Adapted from [25]

III. Maximum Pooling

Maximum pooling attempts to alleviate the extensive processing power required by the neural network by scaling the image down. Starting from the top left cluster of pixels (determined by the kernel size), until the bottom right cluster, the largest values from each bunch is calculated to represent the given cluster in the newly filtered image.

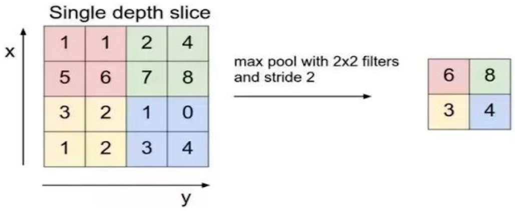

Figure 4: Maximum Pooling Layer
Source: Adapted from [26]

2. Classification Model

I. Fully Connected Layer

The fully connected layers attach each pixel from the previous pooling layer to each node in the fully connected layer. Each pixel has a weight, or level of influence, on the node it is connected to. The value of each node is determined by the summation of the averages of the weights, biases, and pixel values.

II. Output Layer and Softmax

Connecting the neurons from the fully connected layer to the output layer, is a function called Softmax, which forces the values in the output neurons to total a sum of one.

III. Stochastic Gradient Descent and Backpropagation

Backpropagation is an algorithm used to determine how the weights and biases of the neural network are tweaked for best performance. After a given input image passes through the entire neural network and ends up at the output layer, the model score for that single mini-batch can be found by taking the summation of the squared differences between the values from the output layer and the values from the labeled vectors (if benign (1,0) and if malignant (0,1)). Then by using backpropagation, the values for the weights and biases are tuned in accordance to the given mini-batch.

Stochastic Gradient Descent is the process by which the neural network tweaks the values of the weights and biases to attain the lowest model score. With stochastic gradient descent, the neural network trains in sets of images called mini-batches, so that when backpropagation is used to tune the weights and biases, the tweaks are representative of the changes determined only by that given mini-batch. By performing backpropagation in this process, the neural network can be trained at a much faster rate.

IV. Training and Testing Datasets

When using a neural network, it is ideal to split the data into a training and testing dataset. The training set is used to determine the values for the weights and biases of a function by passing through each image and performing backpropagation. The testing dataset is saved until a well-performing set of hyperparameters is found; the images from the testing dataset are then passed through the neural network to determine how well it performs.

V. EVALUATION RESULTS

Dataset	Mini test dataset
Learning rate	0.001
Updater	Nesterov's
momentum	0.9
Weight decay	0.005
Mini-batch size	20
epochs	3
Number of Channels	3
Image dimensions	360x360
Optimization algorithm	Stochastic Gradient Descent
Weight init	RELU
1st Conv kernel size	5x5
1st Conv stride	1x1
1st Conv # of Filters	20
1st Max Pooling kernel size	2x2
1st Max Pooling stride	2x2
2st Conv kernel size	5x5
2st Conv stride	1x1
2st Conv # of Filters	50
2nd Max Pooling kernel size	2x2
2nd Max Pooling stride	2x2
Activation function	Softmax

Table 1: Hyperparameters from Test 1, LeNet CNN, where hyperparameters are based on banana classifier [22], mini dataset

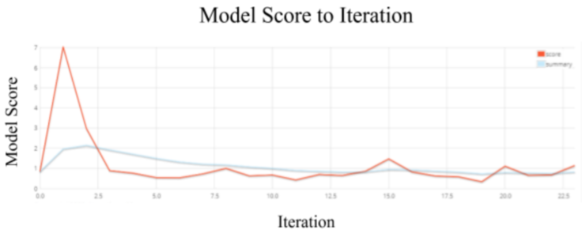

Figure 5: Model Score to Iteration Graph from Test 1

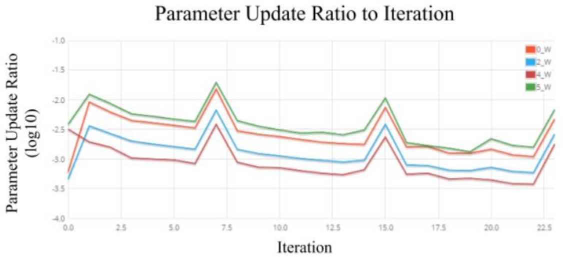

Figure 6: Parameter Update Ratio to Iteration Graph from Test 1

Dataset	Mini test dataset, then full dataset
Learning rate	0.001
Updater	Nesterov's
momentum	0.9
Weight decay	0.005
Mini-batch size	20
epochs	1
Number of Channels	3
Image dimensions	360x360
Optimization algorithm	Stochastic Gradient Descent
Weight init	RELU
1st Conv kernel size	5x5
1st Conv stride	1x1
1st Conv # of Filters	20
1st Max Pooling kernel size	2x2

1st Max Pooling stride	2x2
2st Conv kernel size	5x5
2st Conv stride	1x1
2st Conv # of Filters	50
2nd Max Pooling kernel size	2x2
2nd Max Pooling stride	2x2
Activation function	Softmax

Table 2: Hyperparameters from Test 2, single epoch (number of times run through entire dataset), full dataset

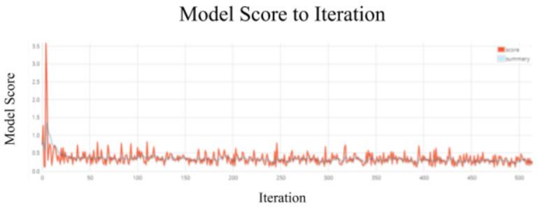

Figure 7: Model Score to Iteration Graph from Test 2

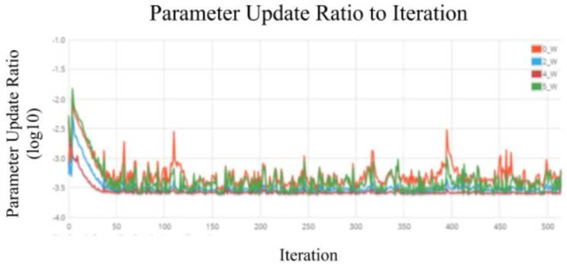

Figure 8: Parameter Update Ratio to Iteration Graph from Test 2

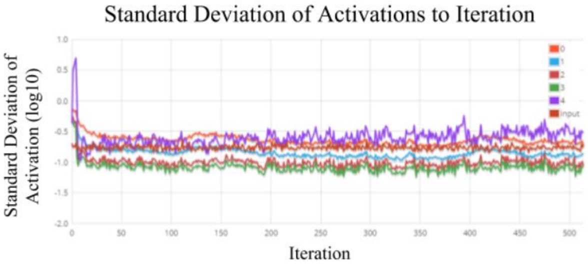

Figure 9: Standard Deviation of Activations to Iteration Graph from Test 2

Dataset	Full Dataset
Learning rate	0.001
Updater	Adam
momentum	N/A
Weight decay	0.005
Mini-batch size	20
epochs	1
Number of Channels	3
Image dimensions	360x360
Optimization algorithm	Stochastic Gradient Descent
Weight init	RELU
1st Conv kernel size	5x5
1st Conv stride	1x1
1st Conv # of Filters	20
1st Max Pooling kernel size	2x2
1st Max Pooling stride	2x2
2st Conv kernel size	5x5
2st Conv stride	1x1
2st Conv # of Filters	50
2nd Max Pooling kernel size	2x2
2nd Max Pooling stride	2x2
Activation function	Softmax

Table 3: Hyperparameters from Test 3, Adam Updater, with evaluation output, full dataset

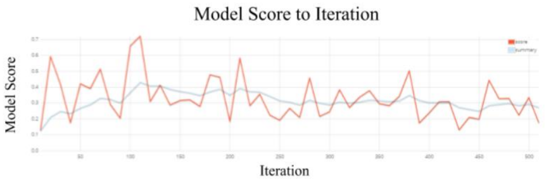

Figure 10: Model Score to Iteration Graph from Test 3

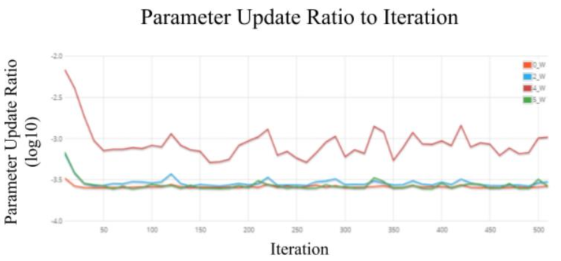

Figure 11: Parameter Update Ratio to Iteration Graph from Test 3

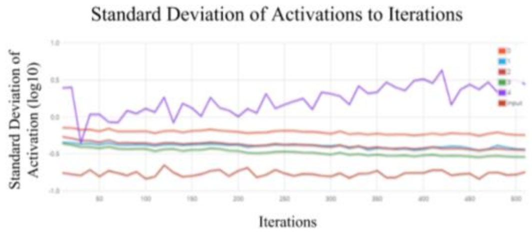

Figure 12: Standard Deviation of Activations to Iteration Graph from Test 3

Examples labeled as 0 classified by model as 0:	2611 times
Examples labeled as 0 classified by model as 0:	566 times
Examples labeled as 0 classified by model as 0:	211 times
Examples labeled as 0 classified by model as 0:	76 times

# of Classes	Accuracy	Precision	Recall	F1 Score
2	0.7735	0.5202	0.5389	0.1619

Table 4: Evaluation Table from Test 3

VI. DISCUSSION

Test 1: LeNet CNN, where hyperparameters are based on banana classifier

The purpose of the first test was to analyze the efficacy of the skin cancer classification neural network when using a method called transfer learning. Transfer learning [27] is the application of a set of hyperparameters already tuned for another similar machine learning problem. In this case, a banana disease classifier which had the hyperparameters below, acted as a template for the skin cancer detection model.

Parameter	Optimization algorithm	Learning Rate	Momentum	Weight Decay	Batch Size	Activation Function	Iterations
Value	Stochastic Gradient Descent	0.001	0.9	0.005	10	Sigmoid	30

Table 5: Banana Classifier Hyperparameters
Adapted from [22]

To save time, the amount of data trained on the neural network was a little under 2% that of the entire dataset. From analyzing the model score to iteration graph, the results appear to indicate accurate readings [28], and the learning rate seems to be at a satisfactory good level since the graph closely resembles that of figure 13 below [23]. In addition, because the model score steadily decreased as the number of iterations increased, the network should have been properly learning to correctly classify images [28]. From analyzing the parameter ratio to iterations graph, another evaluation could be made about the learning rate, which is that because the ratio is approximately 10 ^ -3, the learning rate is at a good value [28]. One concern from this graph were the sudden spikes at around iteration 7, 15 and 23, but they may have simply been caused by the repeating of data (given that the epoch was set to 3). Also, this graph did not diverge past 10 ^ -2 and 10 ^ -4 (when not considering the spikes), thus the weights could be assumed to have been set to a good enough value to learn useful features [28]. Although the spikes were not concerning enough to apply a different weight initialization to try and help alleviate the spikes, one may attempt to do so in a future test.

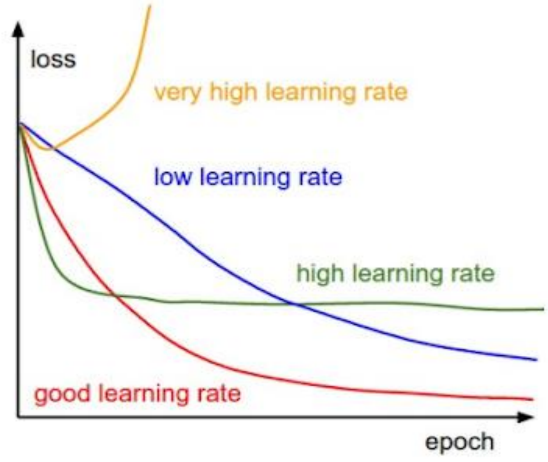

Figure 13: Model Score to Iteration Graph with Varying Learning Rates
Source: Adapted from [23]

Test 2: single epoch (number of times run through the entire dataset)

After comparing the results from the training of the skin cancer detecting neural network with that of an ideally trained neural network [28], the next step was to train with the entire dataset. After training, the dataset showed a model score to iteration function with a good learning rate, averaging out to about 0.3, which is something commonly seen from an optimally set training rate, normalization, optimization method, and mini batch quantity [28]. Also, the parameter ratio to iteration graph evened out to a value of about 10^-3, and never peaked past 10^-2 or 10^-4, indicating a good learning rate and weight initialization [28]. However, there were a few unusual spikes in the graph at iteration 50, 100, and 400 that prompted some questioning. After consulting Alex Black, a computer science graduate from Monash University, with over 5 years of experience in the artificial intelligence industry, who currently works at SkyMind, he stated that the spikes were not too concerning, but gradient chipping was an optional method that could reduce such spikes if concerned (it was not implement in this study). He also advised that the updater be changed from Nesterov's to Adam and that the iterations only display in intervals of 10 to prevent stark oscillations in the graph, which makes analysis of the graphs difficult. Because the standard deviation of activations to iterations graph stabilizes after a few hundred iterations to a value between 0.5 and 2.0, the weight initializations, regulation, data normalization, and learning rates could be said to be at ideal values [28].

Test 3: Adam Updater, with evaluation output

When training the neural network with the updated parameters, the model score to iteration graph remained relatively low throughout the training process (approximately 0.3), although not following the typical path of a well-trained neural network [22, Fig. 13]. Looking at the light blue summary path, it could be described as slightly increasing (which may indicate an excessively large learning rate or an incorrect data normalization) or flat/slowly decreasing score (which may indicate an excessively small learning rate or an unideal optimization method) [28]. Also, from looking at the parameter ratio to iterations chart, the values seem to diverge to 10^-4, with only a single weight floating around 10^-3. In addition, the standard deviation of activation to iterations chart indicate values between 10^-1.5 to 10^0.5. After consulting Alex Black, he recommended running an evaluation test with the neural network to analyze how well the neural network is really training. After running that test, the results indicated a high rate of incorrectly classifying the image. The evaluation table showed that the neural network correctly identified the image as either malignant or benign 77.35 percent of the time [16]. The percent of images labeled malignant out of all the lesions predicted as either correctly or incorrectly malignant was 52.02 [16]. The percent of correctly detected malignant images out of all of all the lesions labeled malignant was 53.89 [16]. A combination of both precision and recall can be described by the F1 score which was 16.15 [16].

Future Tests:

Despite finishing with undesirable test results, the tests still give rise to a few concerns and considerations for future testing. After consulting Adam Black about the results, he stated that the class imbalance (unequal distribution of malignant and benign images) had been a major issue. When dealing with class imbalance, depending on the ratio (in this case 1:13, malignant: benign), there are several steps that should be taken. As with this case, the imbalance is not too drastic, and can likely be compensated for by changing the threshold value for the evaluation class. By plotting a Receiver Operating Characteristics (ROC) curve and an area under the ROC curve [29], [30], the desired thresholds for the evaluation tests can be obtained.

Another problem with the neural network could be the images in the datasets themselves. The dataset used to train the neural network was a compilation of smaller datasets obtained from the ISIC-archive. Although most of the images were clean, there were some with additional objects on the side of the image and others that were already filtered. Recently, the addition of a new dataset from a study called "Human-Against-Machine" contained 10,000 images, all of which contained much cleaner and desirable images [19], [31]. A future test with only these images should be considered.

Smartphone application:

Although no tests were directly run on a smartphone, the program used from the preceding tests are compatible with smartphones. The deeplearning4j library that was used to develop the software in this project has a well-tested built-in library for Android capabilities, which allows for the easy integration with smartphones. In addition, the software was developed in a way so that the training could be performed on the computer, and after, the trained network with the tuned hyperparameters, weights, and biases can be transferred to a smartphone via a .zip file. The evaluation and testing on the testing dataset can also be performed on the computer before exporting the .zip file on the neural network, all of which significantly reduces the memory and processing load required by the smartphone. In addition, the .zip file only requires a total of about 0.686 gigabytes of memory, making it feasible to create an application under 1 gigabytes.

Due to the lack of time, many components of this project were incompletable and left untested. Future works for this project could include developing the smartphone application to ensure that the processor can properly handle the program and the amount of memory on a typical smartphone is sufficient. In addition, the smartphone application should have picture taking capabilities to enable testing on live subjects, rather than on only pre-gathered images.

VII. CONCLUSION

Although, the results in this study do not indicate an effective neural network, the potential in applying convolutional neural networks for skin cancer classification is hardly diminished. On the contrary, this study provides a steady groundwork for future studies regarding machine learning-based classification problems. By avoiding major issues encountered during this study, including the use of an undesirable set of database images and the neglect of optimizing the threshold value when evaluating the convolutional neural network, one can use this study as a guide to start his or her own machine-learning classification project. One may also choose to continue from where this study concluded, by downloading this open source software, finish training the neural network, making it compatible on smartphones, and testing it in the real world.

VIII. REFERENCES

[1] H. Rogers, M. Weinstock, S. Feldman and B. Coldiron, "Incidence Estimate of Nonmelanoma Skin Cancer

(Keratinocyte Carcinomas) in the US Population, 2012", *JAMA Dermatology*, vol. 151, no. 10, p. 1081, 2015.

[2]*Cancer Facts & Figures*. Atlanta: American Cancer Society, 2018 [Online]. Available: https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/annual-cancer-facts-and-figures/2018/cancer-facts-and-figures-2018.pdf. [Accessed: 07- Jan- 2018]

[3]"Survival Rates for Melanoma Skin Cancer, by Stage", *Cancer.org*, 2016. [Online]. Available: https://www.cancer.org/cancer/melanoma-skin-cancer/detection-diagnosis-staging/survival-rates-for-melanoma-skin-cancer-by-stage.html. [Accessed: 15- Jan- 2018]

[4]"Basal and Squamous Cell Skin Cancer Stages", *Cancer.org*, 2017. [Online]. Available: https://www.cancer.org/cancer/basal-and-squamous-cell-skin-cancer/detection-diagnosis-staging/staging.html. [Accessed: 15- Jan- 2018]

[5]T. Kubota, "Deep learning algorithm does as well as dermatologists in identifying skin cancer", *Stanford News*, 2017 [Online]. Available: https://news.stanford.edu/2017/01/25/artificial-intelligence-used-identify-skin-cancer/. [Accessed: 16- Jan- 2018]

[6]Ericsson, "Ericsson Mobility Report: Global smartphone subscriptions to reach 5.6 billion by 2019", Ericsson, Stockholm, 2013 [Online]. Available: http://mb.cision.com/Main/15448/2245678/661730.pdf. [Accessed: 23- Jan- 2018]

[7]U. Fayyad, G. Piatetsky-Shapiro and P. Smyth, "From Data Mining to Knowledge Discovery in Databases", *AI Magazine*, vol. 17, no. 3, 2018 [Online]. Available: https://www.aaai.org/ojs/index.php/aimagazine/article/viewFile/1230/1131. [Accessed: 09- Jan- 2018]

[8]S. Dilsizian and E. Siegel, "Artificial Intelligence in Medicine and Cardiac Imaging: Harnessing Big Data and Advanced Computing to Provide Personalized Medical Diagnosis and Treatment", *Current Cardiology Reports*, vol. 16, no. 1, 2014.

[9]A. Bleicher, "Teenage Whiz Kid Invents an AI System to Diagnose Her Grandfather's Eye Disease", *IEEE Spectrum*, 2017. [Online]. Available: https://spectrum.ieee.org/the-human-os/biomedical/diagnostics/teenage-whiz-kid-invents-an-ai-system-to-diagnose-her-grandfathers-eye-disease. [Accessed: 15- Dec- 2018]

[10]S. Dean, "Science Alert", *This Teenage Girl Invented an AI-Based App That Can Quickly Diagnose Eye Disease*, 2017. [Online]. Available: https://www.sciencealert.com/this-teenage-girl-invented-a-brilliant-ai-based-app-that-can-quickly-diagnose-eye-disease. [Accessed: 15- Apr- 2018]

[11]A. Esteva, B. Kuprel, R. Novoa, J. Ko, S. Swetter, H. Blau and S. Thrun, "Dermatologist-level classification of skin cancer with deep neural networks", *Nature International Journal of science*, vol. 542, no. 7639, pp. 115-118, 2017.

[12]M. Hintz-Madsen, L. Hansen, J. Larsen, E. Olesen and K. Drzewiecki, "Design and evaluation of neural classifiers application to skin lesion classification", *Neural Networks for Signal Processing [1995] V. Proceedings of the 1995 IEEE Workshop*, 2002.

[13]A. Hoshyar, A. Al-Jumaily and R. Sulaiman, "Review on automatic early skin cancer detection", *2011 International Conference on Computer Science and Service System (CSSS)*, 2011.

[14]J. Wolf, J. Moreau, O. Akilov, T. Patton, J. English, J. Ho and L. Ferris, "Diagnostic Inaccuracy of Smartphone Applications for Melanoma Detection", *JAMA Dermatology*, vol. 149, no. 4, p. 422, 2013.

[15]T. Kontzer, "How an AI App Can Translate a Photo into a Skin Cancer Diagnosis", *NVIDIA*. 2017 [Online]. Available: https://blogs.nvidia.com/blog/2017/05/23/ai-app-skin-cancer-diagnosis/. [Accessed: 08- Jan- 2018]

[16]A. Ng, *Data Science, Machine Learning by Andrew Ng*. Stanford: Stanford, 2014.

[17]Y. Abu-Mostafa, *Learning From Data, Machine Learning course*. Pasadena: California Institute of Technology, 2012.

[18]Eclipse Deeplearning4j Development Team, "Deeplearning4j: Open-source distributed deep learning for the JVM", 2018. [Online]. Available: https://deeplearning4j.org/. [Accessed: 15- Apr- 2018]

[19] International Skin Imaging Collaboration archive. [Online] Available: https://isic-archive.com/. Accessed Apr. 10, 2018.

[20]Eclipse Deeplearning4j Development Team, "Customized Data Pipelines for Images etc.", *Deeplearning4j*, 2018. [Online]. Available: https://deeplearning4j.org/simple-image-load-transform. [Accessed: 27- Jan- 2018]

[21]A. Ng, *Train / Dev / Test sets*. 2018 [Online]. Available: https://www.coursera.org/learn/deep-neural-network/lecture/cxG1s/train-dev-test-sets. [Accessed: 06- Apr- 2018]

[22]J. Amara, B. Bouaziz and A. Algergawy, "A Deep Learning-based Approach for Banana Leaf Diseases Classification", *PubMed*, 2017 [Online]. Available: https://www.btw2017.informatik.uni-stuttgart.de/slidesandpapers/E1-10/paper_web.pdf. [Accessed: 14- Mar- 2018]

[23]J. Johnson and A. Karpathy, "CS231n Convolutional Neural Networks for Visual Recognition", 2018. [Online]. Available: http://cs231n.github.io/neural-networks-3/. [Accessed: 19- Mar- 2018]

[24]B. Xu, N. Wang, T. Chen and M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network", *Cornell University Library*, 2015 [Online]. Available: https://arxiv.org/abs/1505.00853. [Accessed: 23- Jan- 2018]

[25]"What is the "dying ReLU" problem in neural networks?", *Stack Exchange*, 2016. [Online]. Available: https://datascience.stackexchange.com/questions/5706/what-is-the-dying-relu-problem-in-neural-networks. [Accessed: 17- Apr- 2018]

[26]A. Karpathy and J. Johnson, "CS231n Convolutional Neural Networks for Visual Recognition", 2018. [Online]. Available: http://cs231n.github.io/convolutional-networks/.

[Accessed: 17- Apr- 2018]

[27]A. Karpathy and J. Johnson, "CS231n Convolutional Neural Networks for Visual Recognition", 2018. [Online]. Available: http://cs231n.github.io/transfer-learning/. [Accessed: 17- Apr- 2018]

[28]Eclipse Deeplearning4j Development Team, "Visualize, Monitor and Debug Network Learning", *Deeplearning4j*, 2018. [Online]. Available: https://deeplearning4j.org/visualization. [Accessed: 27- Jan- 2018]

[29]R. Patwari, *ROC Curves*. 2013 [Online]. Available: https://www.youtube.com/watch?v=21Igj5Pr6u4. [Accessed: 17- Apr- 2018]

[30]K. Markham, *ROC Curves and Area Under the Curve (AUC) Explained*. 2014 [Online]. Available: https://www.youtube.com/watch?v=OAl6eAyP-yo. [Accessed: 17- Apr- 2018]

[31]P. Tschandl, C. Rosendahl and H. Kittler, "The HAM10000 Dataset: A Large Collection of Multi-Source Dermatoscopic Images of Common Pigmented Skin Lesions", *Cornell University Library*, 2018 [Online].