Vision-Based Classification of Skin Cancer using Deep Learning

Simon Kalouche *(kalouche@stanford.edu)*

***Abstract*— This study proposes the use of deep learning algorithms to detect the presence of skin cancer, specifically melanoma, from images of skin lesions taken by a standard camera. Skin cancer is the most prevalent form of cancer in the US where 3.3 million people get treated each year. The 5-year survival rate of melanoma is 98% when detected and treated early yet over 10,000 people are lost each year due mostly to late-stage diagnoses [2]. Thus, there is a need to make melanoma screening and diagnoses methods cheaper, quicker, simpler, and more accessible. This study aims to produce an inexpensive and fast computer-vision based machine learning tool that can be used by doctors and patients to track and classify suspicious skin lesions as benign or malignant with adequate accuracy using only a cell phone camera. The data set was trained on 3 separate learning models with increasingly improved classification accuracy. The 3 models included logistic regression, a deep neural network, and a fine-tuned, pre-trained, VGG-16 Convolutional Neural Network (CNN) [7]. Preliminary results show the developed algorithm's ability to segment moles from images with 70% accuracy and classify skin lesions as melanoma with 78% balanced accuracy using a fine-tuned VGG-16 CNN.**

I. Introduction

Skin cancer is the most common form of cancer in the United States. Of the several varieties of skin cancer (Melanoma, basal cell carcinoma, and squamous cell carcinoma), Melanoma is responsible for only 1% of diagnosed cases yet it accounts for nearly 75% of skin-cancer induced deaths [1]. Each year Melanoma claims 10,000 lives which amounts to one death every 52 minutes. Despite being such a malicious disease, Melanoma is highly treatable, with a 98% 5-year survival rate when detected and treated in its early stages. The survival rate drops to 63% when the cancer spreads from local sites to regional, and survival rates decrease further to just 17% in the later stages where cancer has spread distally to other organs from the initial site [1]. Because Melanoma propagates dangerously, proper monitoring and detection of skin lesions are vital to improve the survivability of this disease.

To aid with proper detection and suspicious mole tracking dermatologists have developed the ABCDEs which is a mnemonic that describes accepted visual features and cues of malignant Melanoma moles. The **ABCDE**'s mnemonic stands for **A**symmetrical shape, **B**order irregularities, **C**olor, **D**iameter, and **E**volution over time. Additionally, the morphology, location on the body, and arrangement of lesions may also provide information about the skin disease [2]. Yet due to the numerous factors at play simultaneously, visual diagnosis is complicated and often leads to subjective and un-reproducible results. Although features of malignant moles are well known, the variation and diversity of skin across patients makes visual classification still a very challenging problem. Preliminary studies have shown that a dermatologist and medical resident from Stanford University's hospital were able to successfully classify skin lesions 46 and 52% of the cases [4]. With classification rates as low as this by medically trained human eyes, proper diagnosis cannot rely on manual visual inspect therefore biopsies are required. Biopsies take time and money in addition to pain and scarring suffered by the patient which is not always necessary if the mole is found to be benign.

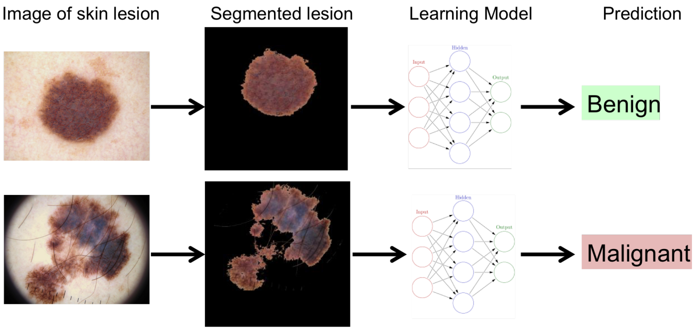

Fig. 1. Computer-vision and machine learning diagnostic tool for doctors and patients to screen suspicious skin lesions and moles.

Several previous machine learning based approaches have attempted the Melanoma classification problem and skin disease classification in general. Attempts include traditional models such as support-vector machines (SVM) and artificial neural networks (ANNs) while two more recent studies implemented fine-tuned VGG-16 ConvNets which inspired the latter approach of this study [4][5].

The two largest public data sets that exist for images of Melanoma include the ISIC database with 1,280 and Dermnet's 23,000 images. The ISIC database consists of 1031 benign images and only 249 images of malignant Melanoma which makes the unbalanced data set have roughly a 1:4 ratio for malignant examples [9]. Dermnet's database of 23,000 images is made up of 500 to 2,500 images of 23 different classes of skin disease. Thus roughly 1000 images exist for malignant Melanoma from this data set. The remaining 22,000 images from Dermnet could be used as the labeled images for benign Melanoma although this would create a heavily unbalanced data set with a 1:22 ratio of training examples for malignant (class 1) to benign (class 2). This study uses just the ISIC data set [9].

II. Image Preprocessing

The raw image data set of 1280 images downloaded from the ISIC Database was not captured or processed ideally and has several inherent sources of noise. Some issues with the raw data included:

1) Vignetting present in some images
2) Bright colored Band-Aids present in some images
3) Hair present in various densities

*This work was conducted in partial fulfillment of Stanford's Machine Learning course (CS 229) taught by Professor Andrew Ng and John Duchi.

[1] S. Kalouche (SUNetID: kalouche) is with Stanford University, School of Engineering, Stanford, CA 94305, USA kalouche@stanford.edu

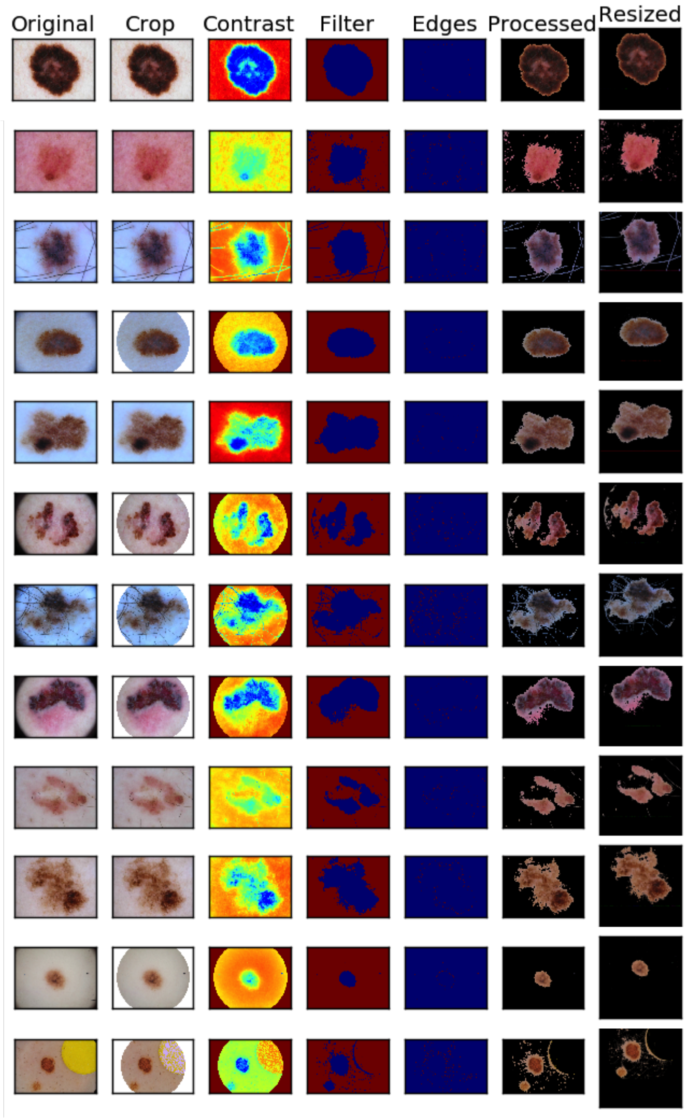

Fig. 2. Example of the pre-processing, lesion segmentation, and formatting from several raw images

4) Scale overlay on some images
5) Variation in ambient lighting
6) Variation in skin lighting
7) Variation in focal length (i.e. scale is unknown)
8) Variation in image aspect ratio and resolution (i.e. image set taken with different cameras)
9) Variation between standard camera and dermoscopic images

For these reasons the raw images were cleaned, pre-processed to segment the skin lesion, and formatted to all be the appropriate shape and size for input into the neural network.

The data set was highly imbalanced having a ratio of 4:1 for images containing benign lesions to images containing malignant lesions. This imbalance is likely to produce bias in the learning models so a balanced subset of the training and test data was created for training and observation purposes. The maximum balanced training data set contained a total of only 346 images which is significantly lower than the original set making learning much more difficult.

A. Lesion Segmentation

Several custom functions were written to automate the pre-processing and lesion segmentation step. This step required quite a bit of tuning and testing to determine which image-processing techniques and filters would yield the best segmentation. After tuning, the high-level methodology for the pre-processing step remained fairly simple and all 1280 images were processed in a matter of 2-3 minutes on a laptop CPU.

After each image is loaded a light Gaussian kernel blur is applied to smoothen the edges and reduce the outline of any thin and light colored hairs present in the image. Next, an algorithm runs to determine whether or not vignetting is present in the image. If vignetting is detected then a circle crop is applied to crop out the corners of the image so that a contrast filter can eventually be applied without increasing contrast in the corners which would lead the model into thinking skin lesions were present in the corners of the image. This crop was sized appropriately for each picture based on the image size and the strength of the vignette. For the most part, the images had the mole/lesion centrally located so cropping out the corners had a near negligible effect. Using built-in functions from Python's OpenCV library the image's contrast was then increased and afterwards the image was converted to gray scale. OpenCV's Canny edge detector was then applied to locate edges and find contours from those connected edges. The contours with the largest areas above a fine-tuned threshold were kept and the rest were discarded. This helped localize the actual mole while removing things like hair and freckles which are not considered as an area of interest. The algorithm then generates a normalized threshold based on the mean pixel value of each image and from that threshold a binary mask is created. The binary mask or filter is a matrix with the same size as the original image but only contains values of 0 and 1 such that when applied (multiplied) onto the original image yields an image that is the area of interest (i.e. the skin lesion) and the surrounding skin and environment is all black. The image then is formatted for input to the learning models.

The accuracy of the lesion segmentation is calculated using the manually labeled binary mask provided with the data set for each image. The pixel area's of the binary mask generated in this pre-processing step and the labeled masks provided are compared. If the difference is less than 20% of the image area then the image is considered segmented correctly. If not, the image is considered incorrectly segmented. The total error is then calculated by

$$error = \frac{\#\ images\ segmented\ incorrectly}{total\ \#\ of\ images} \tag{1}$$

where the resulting error was found to be 30%.

B. Image Formatting: Neural Network

Once the raw images were preprocessed and the mole or skin lesion was segmented out using the method described above the images then needed to be formatted for input into the learning models. The raw images consisted of a variations in resolution and aspect ratio whereas the learning algorithms required all the input training and test images be of the same size. Therefore each image was converted into a square image with a new side length equal to the larger of the two sides of the original raw image (most of which were rectangular). The added area in the new image was set to be black to match the convention of the processed images where a black area indicated a masked region that is not of interest to the learning algorithms.

The images are then scaled from a rough average of 1100 x 1100 x 3 to 256 x 256 x 3 using OpenCV's resizing function with an interpolation that resamples using pixel area relation for moire'-free results.

With the images now fully processed they must be parsed into proper matrix form for input into the various learning algorithms. For learning using logistic regression or a deep neural network the training and test examples are formatted into an mxn matrix X, where m is the number of training/test examples and n is the size of the feature vector (or number of features). Therefore for these two learning models the 3-dimensional image (256,256,3) needed to be converted into a 1-dimensional vector which represents a single row (one example) in the training/test matrix. Several methods of converting from the 3D image to a 1D training/test example are possible and several were tested to determine impact on learning performance. The first option is to convert the 3-channel RGB image into gray scale which then becomes a single channel image. The single channel (256,256,1) image can then be reshaped to a (256X256,1) = $(65536,1)^T$ vector where each row is concatenated to the row above it.

$$X_i = [p_1 \; p_2 \; p_3 \; ... \; p_{65536}]$$

The second option is to choose a single color channel which the algorithm designer may determine to be of more use than the other two color channels. The third option is to combine all three color channels into a single vector representing example image i which now takes the form

$$X_i = [p_{1_r} \; p_{1_g} \; p_{1_b} \; p_{2_r} \; p_{2_g} \; p_{2_b} \; ... \; p_{65536_r} \; p_{65536_g} \; p_{65536_b}]$$

where p_{j_r} denotes the red channel of pixel j where $j \in [1, 65536]$. The performance of these different structured training/test matrices are presented later in Section VI.

Using a 2-class softmax loss requires that the labels y take the form of an mx2 matrix where m is the number of train/test examples. In this problem we define $y_i = [1, 0]$ to be the label of image i if it is benign and $y_i = [0, 1]$ if it is malignant.

C. Image Formatting: Keras ConvNet (CNN)

The implemented convolution neural network was not implemented from scratch like the NN, but rather was built using Python's Keras deep learning library with a back-end using Google's TensorFlow architecture [8]. The images and labels did not need significant parsing for the Keras CNN models. Instead the training and test images (.jpg files) needed to be structured in a directory that had labeled folders for both training and test sets such that the training/testing labels of class 0 and class 1 were located in their respective directories.

$$../data/processed/train/label0/$$
$$../data/processed/train/label1/$$
$$../data/processed/test/label0/$$
$$../data/processed/test/label1/$$

III. Logistic Regression

For comparison purposes a simple logistic regression model was constructed using a 2-class softmax loss (i.e. logistic loss) and a stochastic gradient descent optimizer with update rule

$$\theta_j \leftarrow \theta_j - \alpha(y^{(i)} - h_\theta(x^{(i)}))x_j^{(i)} \qquad \forall j \in [1, k] \tag{2}$$

The learning rate was set to be constant over 10 epochs at $\alpha = .01$. Results are presented in Section VI but they were not promising and thus more complex models were investigated in an attempt to pick up more complex features given the size of the feature vectors and the non-linearity of the learning task.

IV. Neural Network (NN)

After verifying that a simple logistic regression technique was not capable of robustly classifying Melanoma based on images alone, a deep neural network (i.e. multi-layer perceptron) was built.

A. NN Architecture

A simple deep neural network is constructed to classify skin lesions as benign or malignant.

The NN implemented a 2-class softmax loss (l) and a loss gradient (∇l) defined as

$$l(y, h_\theta(x)) = l(y, \hat{y}) = \log\left(\sum_{j=1}^{k} e^{\hat{y_j}}\right) - \hat{y}^T y \tag{3}$$

$$\nabla l(y, \hat{y}) = \frac{e^{\hat{y}}}{\sum_{j=1}^{k} e^{\hat{y_j}}} - y \tag{4}$$

Before the optimization occurs a non-linear function fu_j is applied to the linear hypothesis function $h_\theta(x)$. Therefore, the hypothesis function now takes the form

$$h_\theta(x) = fu_{j+1}\Big(W_{j+1} fu_j(W_j x + b_j) + b_{j+1}\Big) \tag{5}$$

Here, fu is a non-linear activation function where common non-linear functions used are the sigmoid, hyperbolic tangent, and the rectified linear unit (ReLU). In this study we found that the ReLU activation function produced the best experimental results for all layers except the last layer which was set to be a linear activation function so that the confidence could take on both positive and negative values. The ReLU function was also chosen because it does not suffer from gradient vanishing (like sigmoid). ReLU is applied element-wise, and defined by $fu(x)_{ReLU} = max\{0, x\}$, where the function returns zero for values where x is negative and x for values of x that are positive.

The total number of layers varied from 5 (3 hidden) to 7 (5 hidden). Each layer size was exponentially spaced from layer 1 (L_1) having a size of the input data ($\mathbb{R}^k$), the 2nd layer having size of ($\mathbb{R}^k/2$), the third layer having size ($\mathbb{R}^k/4$) and so on until the last layer (L_s) having a size of ($\mathbb{R}^2$) because the last layers should yield the predicted confidence value of for each of the 2 classes (benign and malignant).

The weights matrix **W** and bias vector **b** are initialized randomly and then fed into a stochastic gradient descent optimization algorithm which was trained by searching for the parameters $\theta = [\mathbf{W}\;\mathbf{b}]$ which minimized the logistic/softmax loss function. An L-1 regularization term was added to the update on the weight matrix to ensure weights stayed reasonably small.

B. Tuning Model Parameters

The neural network was build robustly and with modularity in structure such that nearly every parameter of the model could be tuned easily and modulated from test to test to determine which parameters yielded the best results. The tunable parameters include: **1)** the number of hidden network layers (s), **2)** the learning rate, **3)** number of optimization epochs, and **4)** the lambda (λ) parameter for the L-1 regularization

C. Tuning Model Inputs

In addition to the model parameters several attempts at manipulating the input data were also tested for performance enhancement. These included **1)** the image input resolution/size, **2)** the number of training examples by artificial data augmentation, and **3)** using pre-processed and segmented images vs non-processed (raw), only formatted, images.

V. Fine-tuning the VGG-16 Deep CNN

For the final approach transfer learning is used on a Convolutional Neural Network (CNN). Transfer learning has been shown to be a time and computationally efficient method of training a deep CNN [5]. In transfer learning, rather than training the weights of the network from scratch at a random initializing, the weights of a pre-trained network can be imported to an instantiated convolutional base (for VGG-16 in this case) then retrained on a new data set using very small weight updates. The process of "fine-tuning" works well because the output layers and weights of a well-trained network contain generic, low-level, features like edge and blob detectors which are useful for classifying many types of images. Therefore, for the skin lesion data set these new learned features can be directly applied while fine-tuning the final layers of the network for proper classification for this problem.

The VGG-16 model is chosen for its relative ease of implementation and its success in the ILSVRC-2014 competition where it placed first in the in task *2a* challenge. The VGG-16 is a very deep, 16-convolutional-layer network that is originally trained on the ImageNet database consisting of millions of labeled images in 1000 classes.

The model was developed in Keras, a high-level neural networks library, written in Python and capable of running on top of either TensorFlow or Theano. The algorithm for fine-tuning the VGG-16 model was based off an example from [8] which was originally adapted from [7].

The fine-tuning is achieved by first instantiating the convolutional base of VGG16 and loading its pre-trained ImageNet weights. The VGG-16 model consists of 5 convolutional blocks with corresponding output filter sizes [54, 128, 256, 512, 512] and then a fully-connected classifier. With everything up to the fully-connected classifier instantiated in the model we run the model on the training and test data sets once. The last activation maps before the fully-connected layers are then saved in 2 Numpy arrays which will be used to train a small full-connected model on top off these stored features [8]. This small, fully-connected model becomes our top-layer model. Now, this trained top-layer model is added to the the VGG16 model and its weights (along with the VGG-16 ImageNet weights) are loaded. We then freeze the layers of the VGG-16 model up to the last convolutional block which prevents the weights of these layers from being changed. We allow the final convolutional block and our added fully-connected top-layer classifier to be fine-tuned using a stochastic gradient descent (SGD) optimizer with a slow learning rate. A small learning rate is used to prevent wrecking the previously learned feature weights [8].

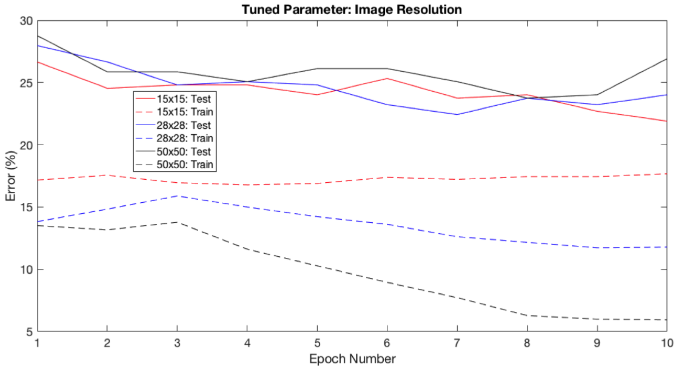

Fig. 3. Performance of NN on unbalanced test for various image resolutions.

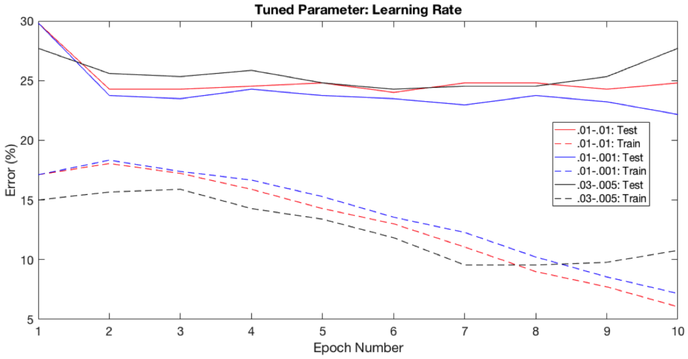

Fig. 4. Performance of NN on unbalanced test for various learning rates α.

VI. Results and Discussion

All three of the tested models were tuned to some degree in an attempt to optimize the performance of the learning model. The majority of this study was focused on the neural network and the VGG-16 CNN. The largest source of modeling frustration was derived from the fact that the data set was heavily imbalanced with 4 times more data for benign skin lesions than for malignant ones. Therefore, when training and testing on the full data set the average error hoovered around 20% which seems good at first glance but upon further inspection this 20% error is a direct result of the model learning to predict benign for all cases since the data set is skewed towards benign moles. To combat this a balanced subset of the data was created with an equal number of malignant and benign training and test examples. Training on this subset of the data, however, significantly reduces the number of examples which is already relatively small to begin with for deep neural network learners. This seemed to be the bottleneck for the test error reaching no lower than 44% (56% accuracy) for the neural network with the parameters listed in Table I.

The logistic regression model using an SGD optimizer hoovered around 48-50% error which is no more useful than flipping a coin. With these results being less than promising a third, more established model for image recognition, was built. The VGG-16 model was loaded with pre-trained weights from

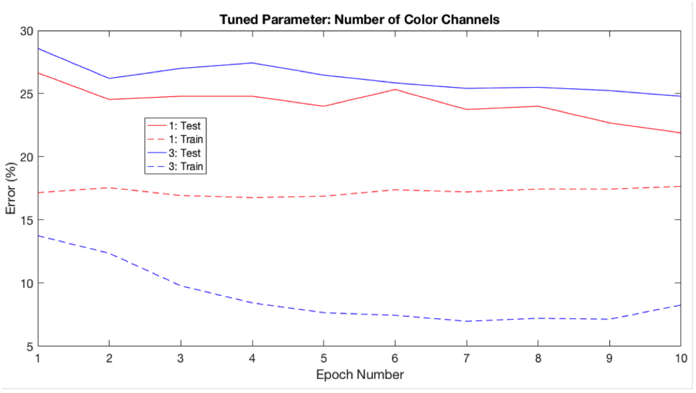

Fig. 5. Performance of NN on unbalanced test for numbers of color channels.

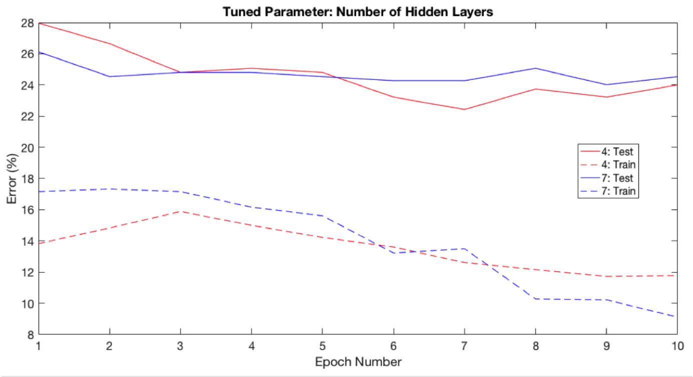

Fig. 6. Performance of NN on unbalanced test for varying number of layers.

ImageNet and then the last 3 layers were fine-tuned over the Melanoma training examples. This algorithm was significantly more effective with a minimum error of 22% in the balanced test set.

All the models were tested on the pre-processed and raw (only formatted) image sets to determine if the pre-processing helped filter out noise in the images for the learning model. The neural network is observed in Fig. 8 to have a lower maximum minimum error by nearly 4% indicating that the pre-processing does in fact help the model train and extract the more important features in the skin lesions.

The number of hidden layers was shown in Fig. 6 to have a very small effect on the performance of the neural network. This may be because there is not enough data from the ISIC data set to train and extract features from for very deep networks.

The learning rate was found in Fig. 4 to yield the best results in a dynamic configuration where the rate was slowly and linearly decreased from 0.01 to 0.001 over the epochs of the stochastic gradient descent.

Adding color channels and artificially augmenting the original inputs to simulate the effects of a larger training set was beneficial to a point and exceeding that point led to over fitting. Therefore using 2 image rotations was found in Fig. 7 to yield the best generalization error. Two image rotations are used to double the effective training set where the augmented set is now constructed by each of the original images and their corresponding 90-degree-rotated-clockwise image. All three colors channels were used in some instances but gray-scale also seemed to work better in other tests. Fig. 5 shows all parameters held constant except for the number of color channels used per training image. The model using only 1 channel in gray scale classifies with lower error by a few percent than that of the model using 3 color channels. It is important to note over-fitting for the 3-channel model where the training error is seen to drop significantly while the test error remains fairly high.

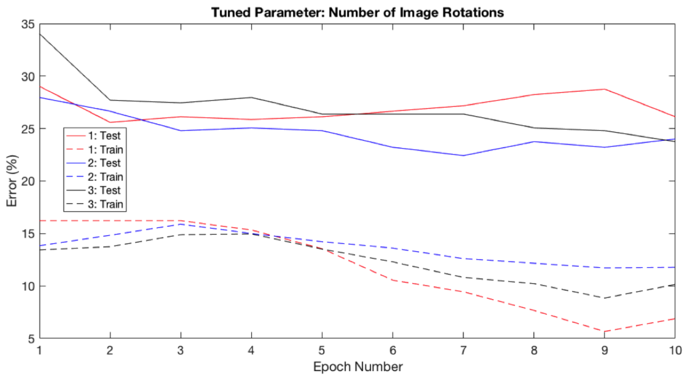

Fig. 7. Performance of NN on unbalanced test for various number of training examples from data augmentation by image rotations.

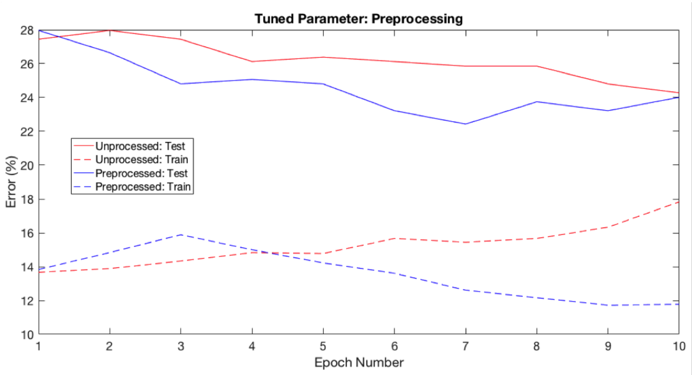

Fig. 8. Performance of NN on unbalanced test for a processed input data set and a raw input data set without lesion segmentation.

For both the neural network and the VGG-16 fine tuning models when the image resolution is increased the variance increases and the model starts to over-fit. This is evident from Fig. 3 by the training error decreasing while the test error remains stagnant or increases. Because the model would not generalize well with high resolution training images the resolution was kept to 50x50 or lower.

Another attempt at reducing over-fitting was to add a regularization term. By adding this term to keep the weights low the model increased in bias drastically and even for very small regularization coefficients the model would learn to always predict class 0 (i.e. benign) on the unbalanced data sets because it's the easiest model to learn that can maintain low weights and low error. The 20% error achieved on the imbalanced set is not a good indicator of true error however because all 20% of those classified incorrectly were malignant skin lesions where a false negative in cancer diagnoses is obviously a dangerous outcome. Therefore other performance metrics need to be investigated on these models.

Instead of accuracy — the ratio of correctly classified examples to total examples, precision — a measure of a classifier's exactness, recall — a measure of a classifier's completeness, or

TABLE I
NEURAL NETWORK PARAMETERS

Parameter Description	Value
Learning rate (α)	.05-.001
Number of epochs	30
Processed vs Unprocessed input data	processed
Number of color channels used	3
Color Channels merged or separate example	merged
Input image resolution	(20,20,3)
Data augmentation: number of image rotations	2
Number of hidden network layers	2
Balanced vs unbalanced data set	balanced
Regularization term constant (λ)	0.00
Activation functions	ReLU

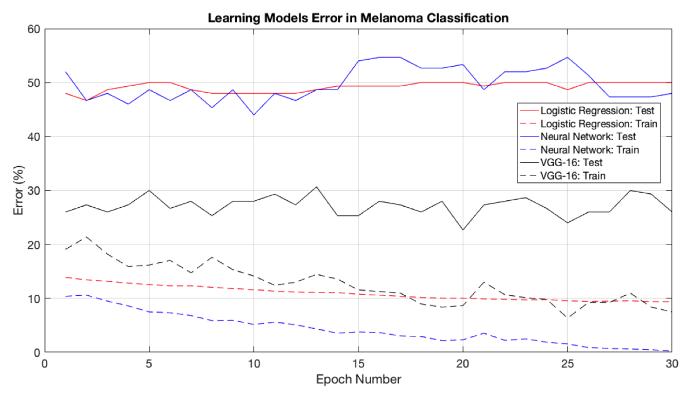

Fig. 9. Performance comparison of test and training error for the 3 major models tested all on a balanced training/test set.

F1 Score — a weighted average of precision and recall could be used to provide a more honest and useful evaluation of the learning model on a highly unbalanced data set. The use of a confusion matrix can also be insightful for analyzing the results of the model.

$$Precision = \frac{\#\ true\ pos.}{\#\ true\ pos.\ +\ \#\ false\ pos.} \tag{6}$$

$$Recall = \frac{\#\ true\ pos.}{\#\ true\ pos.\ +\ \#\ false\ neg.} \tag{7}$$

$$f_1 = 2 * \frac{precision * recall}{precision + recall} \tag{8}$$

The classification error of this study does not exceed the current state-of-art which has been achieved in [4] and [5]. Reasons for only reaching 78% accuracy in comparison to [4]'s 90.0% and [5]'s 91.0% may be due to the fine-tuning process and the data sets used. In both [4] and [5] the much larger DermNet database was used which had roughly 22,000 more images to train on. Additionally, these learning models use an average of VGG-16 and VGG-19 predictions as the final classifier which combines 2 powerful visual classifiers both pretrained on ImageNet. Although these images were not all malignant the larger training set can help the learning model learn features and weights more accurately for benign skin lesions which simultaneously will help classification of malignant lesions.

VII. FUTURE WORK

The VGG-16 network will be further tuned and the ISIC data set will be augmented with the DermNet data set such that model training can occur on a more comprehensive and diverse set of mole images. This will ideally expose the learning model to more variations in skin lesions and allow it to extract true patterns in the malignant lesions. Other pretrained models such as the VGG-19 or Inception v3 may be fine-tuned and tested.

It would also be interesting to implement a loss function which penalized false negatives more than false positive. This could be helpful in using the full imbalanced data set which currently has a bias toward learning the 'always choose benign' model when accuracy is the performance metric. The performance metric in the future will also be the F1-score — a weighted average of the precision and recall — to avoid the seemingly good results obtained when achieving 80% accuracy on a data set with a 4:1 ratio of benign to malignant images.

REFERENCES

[1] American Cancer Society, "Cancer Facts & Figures 2016," American Cancer Society, Atlanta, GA, USA, 2016.

[2] Skin Cancer Facts and Statistics [Online]. Available: www.skincancer.org

[3] Dermnet: Skin Disease Atlas [Online]. Available: http://www.lib.umich.edu/database/link/11961

[4] A. Esteva, B. Kuprel, and S. Thrun. "Deep Networks for Early Stage Skin Disease and Skin Cancer Classification."

[5] H. Liao. "A Deep Learning Approach to Universal Skin Disease Classification." University of Rochester Department of Computer Science, CSC 400 - Graduate Problem Seminar - Project Report, 2016.

[6] M. H. Jafari, E. Nasr-Esfahani, N. Karimi, S.M. Reza Soroushmehr, S. Samavi, K. Najarian. "Extraction of Skin Lesions from Non-Dermoscopic Images Using Deep Learning." arXiv:1609.02374. Sept. 8, 2016.

[7] K. Simonyan, A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." ICLR, 2015.

[8] F. Chollet. "Building Powerful Image Classification Models Using Very Little Data." [Online]. Available: https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

[9] K. Simonyan, A. Zisserman. "Skin Lesion Analysis toward Melanoma Detection: A Challenge at the International Symposium on Biomedical Imaging (ISBI) 2016, hosted by the International Skin Imaging Collaboration (ISIC)." arXiv:1605.01397v1, May 2016.

[10] Gutman, David; Codella, Noel C. F.; Celebi, Emre; Helba, Brian; Marchetti, Michael; Mishra, Nabin; Halpern, Allan. "Skin Lesion Analysis toward Melanoma Detection: A Challenge at the International Symposium on Biomedical Imaging (ISBI) 2016, hosted by the International Skin Imaging Collaboration (ISIC)". eprint arXiv:1605.01397. 2016.