Int. J. Math. And Appl., 8(3)(2020), 207–210
ISSN: 2347-1557
Available Online: http://ijmaa.in/

International Journal of *Mathematics And its Applications*

Melanoma Detection From Images of Moles Using Neural Networks

Matthew Song[1,*]

1 Westford Academy, Westford, MA, United States of America.

Abstract: Moles can be analyzed for certain characteristics to determine whether the mole is malignant or benign and if a patient is at risk for melanoma, a type of skin cancer. A standard multilayer perceptron and convolutional neural network are used to investigate whether machine learning can serve as a substitution for a human examination of such moles. Both neural networks were developed using the Tensor Flow python module, and images of moles were fed as training points and test points to be classified. Both types of networks were able to outperform random guessing, with accuracies of 0.82 and 0.84 for the multilayer perceptron and convolutional neural network respectively.

Keywords: Melanoma Detection, Neural Networks, Skin cancer.

1. Introduction

Melanoma is a serious type of skin cancer forming in melanocytes: skin cells that responsible for the pigmentation of the skin. The exact cause of melanoma is unknown, but exposure to ultraviolet (UV) radiation attributes to a higher risk for the disease [4]. An early sign of melanoma disease development is the formation of moles on the skin. The formation of a mole may be the result of a tumor developing on the epidermis of the skin, which the developing stage is known as melanoma *in situ* [3]. If the disease is detected in the early stages, melanoma can be easily treated by excision of the tumor, with a survival rate of 98 percent after treatment [2]. If not treated, the tumor will expand and spread to nearby lymph nodes and other parts of the body, and later treatment will be less effective in the survival of the patient. Therefore, early detection of melanoma is imperative for the survival of a patient.

When differentiating between benign and malignant moles, the characteristics of shape, border, color change, diameter, and evolution are examined. Unusual moles that may indicate melanoma usually have an asymmetrical shape, irregular border, irregular or various colors, large diameter, and evolving changes in the mole, such as new symptoms or changes in the aforementioned characteristics. Healthy moles are uniform in color, have a definite border, symmetrical halves, and a diameter less than 6 millimeters in length [4].

Machine learning can serve to be a cheap and efficient way for diagnosis of early forms of melanoma. With an effective machine learning algorithm, both the number of uncaught cases of melanoma and the number of false melanoma diagnoses can be greatly reduced. This article investigates the effectiveness of neural networks in image classification of moles for skin cancer. It was predicted that a neural network would perform better than a random guess, because the network would easily pick out unusual characteristics of malignant moles, such as the asymmetrical halves and irregular border.

* *E-mail: matzsong@gmail.com*

2. Methods

A. Data Collection

Roughly 3300 images of moles categorized as benign and malignant were taken from The International Skin Imaging Collaboration (ISIC) Archive via Kaggle [1]. Each image was opened using the Pillow module and processed into a numpy array from the opened image, with each RGB image having a (224, 224, 3) shape. Every element in the array was an integer between 0-255 representing the R, G, and B values for a pixel. Each image was also labeled with a '0' for benign and '1' in a numpy array for labels. The data had already been pre-sorted into test and train sets when downloaded from Kaggle, and the numpy array for the image was added to either the training set numpy array or test set numpy array. Finally, the values in both image sets were normalized by dividing the numpy array by 255.0, which divides each element in the array by 255.0, putting the RGB values between 0 and 1. The normalization of values helps decrease training time for certain activation functions. Example images are provided, which a numpy array from the training set was converted back to an image using the matplotlib module.

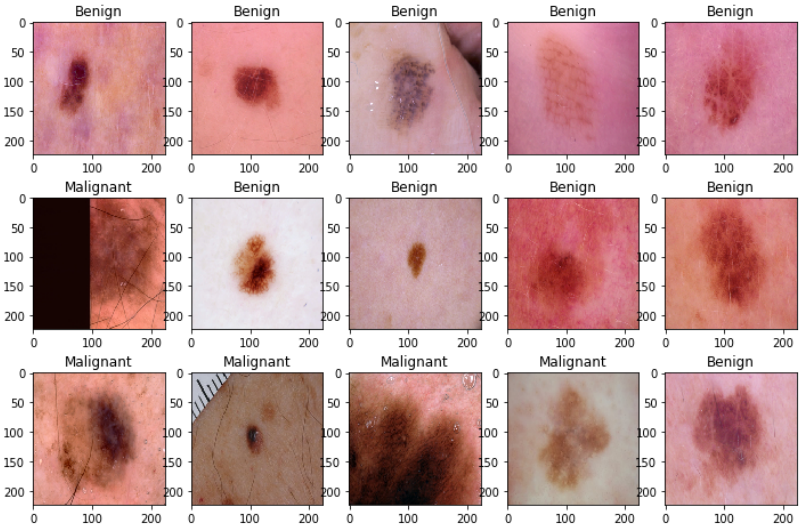


Figure 1: Example Labeled Images from Training Data Set

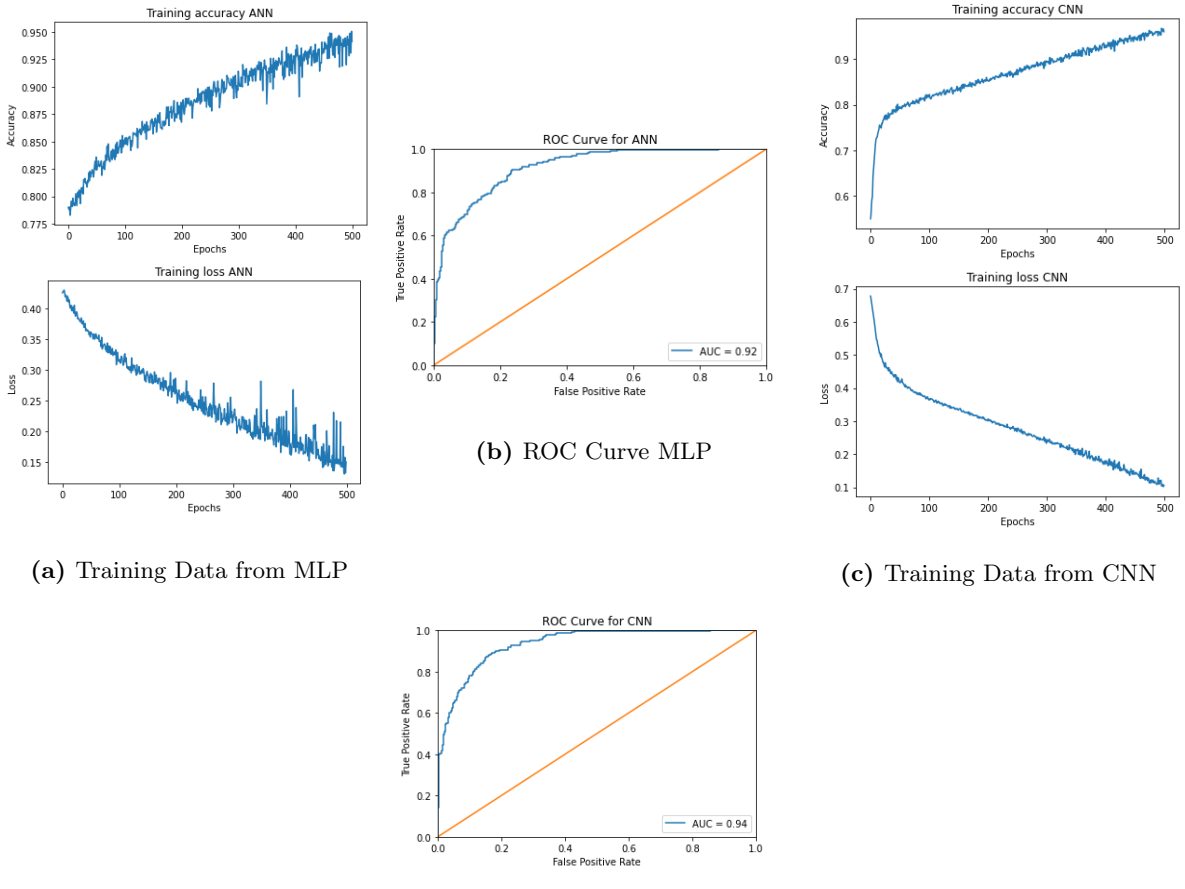
B. Network Creation

Both a simple multilayer perceptron (MLP) and a convolutional neural network (CNN) were developed to be trained with the image set. The TensorFlow package and keras subclass were utilized for the creation and training of the MLP and CNN. In the MLP, three `tf.keras.layers.Dense` layers, standard hidden layers, were added to a `tf.keras.Sequential` model, with the first layer containing 128 nodes, followed by 12 nodes in the second layer and a final node for the output. The first two layers used the ReLU activation function, $max(0, t)$, where t is a value of a node. The final layer used the sigmoid activation function, $\frac{1}{1+e^{-t}}$, which is used for binary classification, because the range of the function is $(0, 1)$. The model is then trained with the training set using the `tf.keras.Model.compile` function. The loss function used in the model fitting was a binary cross-entropy/log loss, which is $L(y) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(p(y_i) + (1 - y_i)(\log(p(1 - y_i)$, y is the output vector. The method of optimization is gradient descent, $w = w - a\frac{\delta loss}{\delta w}$, where w is the weight and a is a learning rate. The learning rate chosen was 0.001, as prior experimentation with higher learning rates led to greater variation in accuracy between epochs. This model would be trained for 500 epochs with the training set. Finally, the results were recorded during compilation of the model, the model was fitted to the test set, and metrics such as accuracy, loss, and the receiver operating

characteristic (ROC) curve were recorded from keras and the sklearn module and plotted using matplotlib.

In the CNN, the optimization function, loss function, epochs, and remained the same as the MLP, but the model layers were adjusted. Three convolution layers, `tf.keras.layers.Conv2D`, and three pooling layers, `tf. keras.layers.MaxPooling2D` were added before the Dense layers, and one Dense layer was taken out. Using the convolution layers, the CNN would be able to pick up features of the image more easily, such as the border of the mole. This is because the convolution layers contain filters of a certain shape, which are applied to a piece of the image with the same shape, with a stride length controlling which pieces are filtered. Likewise, a max-pooling layer acts the same as a filter, having a certain shape and traversing the image by a certain stride length, but the max pooling layer performs dimensionality reduction by returning only the maximum value of a piece of the convoluted image [5]. In the first and second convolution layers, a filter shape of 3 x 3 was used, but the first layer used 16 filters while the second and third used 64 to capture more obscure details. The max pooling layers had shapes of 2x2, and a max pooling layer was performed after a convolution layer. The same procedure for training and analysis from the MLP was used for the CNN.

3. Results



(a) Training Data from MLP

(b) ROC Curve MLP

(c) Training Data from CNN

(d) ROC Curve CNN

Figure 2: Result Graphs from Training and Testing

Both the MLP and CNN performed at the same level, with the CNN performing slightly better in terms of accuracy, loss, and area-under-ROC curve (AUC-ROC) in the test set. According to Figure II.a and II.c, both the MLP and CNN had around a 0.95 accuracy rate after 500 epochs, and the log loss was lower in the CNN, with 0.10 compared to 0.14 in the

MLP. However, when the models were fitted against the test set, the CNN was slightly better-performing in accuracy, loss, and ROC-AUC, with values of 0.84, 0.41, and 0.94 respectively for the three metrics. The MLP had values of 0.82, 0.45, and 0.92 respectively. Both the MLP and CNN were able to greatly outperform random guessing as shown in Figure II.b and II.d, which the blue ROC curves were well above the orange guessing curve.

4. Discussion

Based on the metrics of accuracy and AUC-ROC, the MLP and CNN outperformed random guessing by a wide margin, with the CNN outperforming the MLP by a small margin in training and testing. As stated before, it seems that the two types of neural networks were effectively able to learn to identify peculiarities of malignant and benign moles.

Further examination of the learning and loss graphs for the MLP show that the MLP learned very erratically, which indicated that the learning rate of gradient descent may have been too high. The learning rate could have been lowered to achieve better learning. In addition, both the MLP and CNN have seemed to overfit the training set when training, as the training accuracy was roughly 0.95 for both, while the test accuracy was only 0.82 and 0.84 respectively for the test set. Overfitting can be corrected by adding more testing and training images, reducing the number of epochs in training, or implementing a cross-validation image set.

Further study of the image set could be to increase the accuracy and effectiveness through careful tweaking of optimizer type, network depth, node count, etc. There have also been improved versions of the standard CNN, such as the EfficientNet, that are able to achieve greater accuracy at the cost of higher resource usage.

References

[1] Claudio Fanconi, Skin Cancer: Malignant vs Benign, www. kaggle.com/ fanconic / skin - cancer - malignant - vs - benign, Accessed 8 Sept. 2020. July 2019.

[2] ISIC About, www. isic-archive.com, Accessed 8 Sept. 2020.

[3] Melanoma, www.mayoclinic.org/diseases-conditions/melanoma/ symptoms-causes/syc-20374884, Accessed 8 Sept. 2020. Mar. 2020.

[4] Melanoma Skin Cancer Stages, www.cancer.org/cancer / melanomaskin - cancer / detection - diagnosis - staging / melanoma - skin cancer-stages.html, Accessed 8 Sept. 2020. Aug. 2019.

[5] Saha Sumit, A Comprehensive Guide to Convolutional Neural Networks-the ELI5 way, Accessed 8 Sept. 2020. Dec. 2018.