Fall 2017

Convolutional Neural Networks for Predicting Skin Lesions of Melanoma

Anuruddha Jayasekara Pathiranage

Convolutional Neural Networks for Predicting Skin Lesions of Melanoma

Anuruddha Jayasekara Pathiranage

Regis University

Abstract

Diagnosis of an unknown skin lesion is crucial to enable proper treatments. While curable with early diagnosis, only highly trained dermatologists are capable of accurately recognize melanoma skin lesions. Expert dermatologist classification for melanoma dermoscopic images is 65-66%. As expertise is in limited supply, systems that can automatically classify skin lesions as either benign or malignant melanoma are very useful as initial screening tools. Towards this goal, this study presents a convolutional neural network model, trained on features extracted from a highway convolutional neural network pretrained on dermoscopic images of skin lesions. This requires no lesion segmentation nor complex preprocessing. Further, it doesn't cost much computational power to train the model. This proposed approach achieves a favorable training accuracy of 98%, validation accuracy of 64.57% and validation loss 0.07 in the model with 46% sensitivity and 64% classification accuracy in testing data.

Keywords: convolutional neural network, classification, melanoma

Acknowledgment

I would like to thank the individuals who contributed their input to the fulfillment of this Thesis. I would like to recognize Prof. Nathan George, Prof. Kevin Pyatt and Prof. Robert Mason for their motivation, enormous support and feedback given me on this thesis project. In addition, I would like to thank the IRB members in Regis University for approving the thesis proposal. I am also grateful to all the faculty and staff at Regis University who provided an absolutely wonderful learning environment. I would like to thank my fellow Masters students for their feedback and cooperation. Lastly, I would like to thank my family; my wife, my parents and to my brother and sister for supporting me spiritually throughout writing this thesis and my life in general.

Table of Content

List of Tables

List of Figures

Convolutional Neural Networks for Predicting Skin Lesions of Melanoma

Chapter 1- Introduction

Direct digital imaging for medical diagnosis has become a popular method with modern computation power and machine learning methods. Today, various deep learning models have been created and applied in the field of medical diagnosis due to their ability to recognize patterns in digital images (Cıcero, Oliveira, Botelho, & da Computaçao, 2016). Convolutional neural networks (CNNs) are currently the best-performing technique for image classification. As such, CNNs have led to breakthroughs in many medical image analysis tasks such as classification and detection of illnesses. For example, CNN models have been used to classify malignant and benign lesions from dermoscopy images of human skin (Nasr-Esfahani et al., 2016).

Skin cancer is by far the most common form of cancer in the United States. Melanoma is the deadliest form of skin cancer caused by abnormal multiplication of pigment producing cells that give color to the skin (Nasr-Esfahani et al., 2016). According to the American Cancer Society's estimations for 2017, there are about 87,110 new melanoma cases that will be diagnosed, and about 9,730 people are expected to die of melanoma (American Cancer Society, "Key Statistics for Melanoma Skin Cancer, 2017"). As the incidence of melanoma has doubled and increased at a faster rate than any other types of cancer, it has become a major public health threat not only in the United States, but also globally. Early diagnosis is a critical issue when combatting this disease, and it allows for more treatment options. Nowadays more sophisticated equipment and highly trained professionals are needed for accurate and early detection of melanoma. Dermoscopy is a specialized technique for obtaining high-resolution magnified images of the skin, by controlling light and removing surface skin reflectance (Codella et al.,

2016, p.1). However if the clinicians are not well trained in recognizing melanoma from dermoscopy images, many patients will be misdiagnosed. Also, the expert dermatologist classification for melanoma classification is 65-66% (Esteva et al., 2017).Therefore, there is a high demand for automated image analysis systems to identify melanoma from dermoscopy images. But still it is challenging because of the huge variation of melanoma in terms of color, texture, shape, size, location of the dermoscopy images, and the visual similarity of melanoma lesions and non-melanoma lesions (Yu, Chen, Dou, Qin, & Heng, 2017). In order to overcome this challenge, some researchers propose to perform segmentation first. This attempts to automatically extract the area of the image that contains the melanoma. Then based on the segmentation results, the melanomas are identified (Celebi et al., 2007; Ganster et al., 2001). Celebi et al presented a methodological approach to the classification of melanoma, in which automatic border detection was performed first to separate the lesion from background skin. Then shape, color, texture features were extracted, fed into an optimization framework and determined the optical feature subset to perform the classification (Celebi et al., 2007). However, some methods do not perform well due to the limited discrimination capability.

Due to the limited supply of dermatology experts, novel research efforts have been taken to develop automated dermoscopy image analysis to identify skin diseases. Such technology could be used as a diagnostic tool to interpret dermoscopy images (Codella et al., 2016). Among the variety of automated image analysis techniques used, many applications are restricted to low-level visual feature representations, segmentation algorithms and classical machine learning techniques such as k-nearest neighbor (kNN), support vector machines (SVM) and convolutional neural networks (CNNs) (Codella et al., 2016). CNNs are comprised of a stack of convolutional modules, and each module typically consists of three components; convolutional layers, pooling

layers and dense (fully connected) layers. CNNs apply a series of convolutional filters to the raw data to extract low- and high-level features (such as lines at different angles and patterns matching human faces), which can be used for classification in many applications.

Statement of the Problem

The fatal effects of melanoma trigger the need of early diagnosis system, so that countermeasures can be taken. At the moment, dermoscopy is used as a diagnosis tool to accurately detect the skin lesions of melanoma, because if melanoma is diagnosed in its early stages, there is a good chance of recovery. However, melanoma diagnosis through dermoscopy images is difficult as it requires extensively trained specialists. As a result, deep learning models such as CNNs are currently deployed for automatic detection of melanoma from dermoscopy images. These existing CNN architectures have used different types of convolutional architectures and various techniques for classification with varying prediction accuracy. Therefore, the problem being studied in this work is to create a convolutional architecture in such a way that can extract useful features from biomedical images for high-accuracy melanoma classification. This study proposes a novel model based on deep CNNs to classify skin lesions of dermoscopy images into malignant and benign melanoma categories.

Purpose of the study

As CNNs currently have been shown an impressive performance on recognizing patterns in digital images, it is well-suited to identify melanoma from dermoscopy images (Cıcero et al., 2016). Motivated by recent works performed by research groups, which have used GPUs to train a deep convolutional neural network (Krizhevsky, Sutskever, & Hinton, 2012), the study aims at optimizing the array of layers in CNN architecture to better learn a classifier to extract high level features of melanoma from dermoscopy images. The research will be conducted to

determine how high of accuracy and recall that the model can achieve on melanoma classification with CNNs. Recall is the number of true positives divided by all positives. A high recall means fewer false negatives (cases where someone has melanoma, but it is not detected). Classification accuracy is the percent of correctly predicted cases in image dataset.

In the first part of this study, several CNN architectures are created with varying a) number of layers in the convolutional module, b) number of training steps and learning rate, c) activation functions and d) different optimizers. Then the performance of melanoma classification is compared within those CNN architectures by total loss and accuracy of the trained model. In the second part of the study, a novel CNN model is designed with chosen hyperparameters from previously tested architectures to classify skin lesions of dermoscopy images with high accuracy and recall (sensitivity). In addition to creating a novel CNN model, other publicly available CNN models such as AlexNet, VGGNet, Highway CNN and Google Inception v3 will be applied to the same dataset to evaluate the performance of the proposed model.

Theoretical Basis

Even though CNNs saw use in the 90's, the best models of today beats the champion model from 2012 in the ImageNet classification challenge (Krizhevsky et al., 2012). A CNN consists of one or more convolutional layers followed by one or more fully-connected layers. As CNNs assume that the inputs are images, we can encode certain properties into the architecture. This makes a more efficient function with a fewer number of properties in the network. Another advantage of convolutional neural networks over full-connected networks is easiness to train the model. There are four main operations can be seen in convolutional networks (Harley, 2015) as shown in Figure 1;

1. Convolution

2. Non-Linearity
3. Pooling (Subsampling)
4. Classification (Fully-connected)

An image from a standard digital camera has three channels; red, green and blue. These are like three stacked 2D matrices, and have pixel values from 0 to 255. A convolutional neural network is comprised of layers and each layer transforms an input layer to an output layer with differentiable functions. The three main types of layers that are involved in building convolutional neural networks are convolutional layers, pooling layers and fully-connected layers (Zeiler & Fergus, 2014).

Convolutional layer. The convolutional layers constitute the core building block of the convolutional network. Most of the computational heavy lifting is performed in this layer (Zeiler & Fergus, 2014). Parameters of convolutional layers are made of learnable filters, which are sometimes referred to as kernels. Every filter consists of a small area of an image, and this area is called a receptive field. The filter extends to a depth that is similar to the full depth of the input volume (Dumoulin & Visin, 2016). The filter is an array of numbers on a layer of the convolutional network has a size, ***w x h x d*** (**w** pixels width, ***h*** height, and depth ***d***, the color channels). As the filter is sliding or convolving across the width and height of the input image, it multiplies the values in the filter with pixel values of the input. In each convolutional layer, there is a set of filters and when sliding the filter over the input volume, it produces a 2-dimensional activation map (Dumoulin & Visin, 2016). When stacking them along the depth, it builds the output volume. The spatial extent of this connectivity is a hyperparameter called the receptive field of the neuron (Luo, Li, Urtasun, & Zemel, 2016).

The input volume used in this study has size ***w*** x ***h*** x ***d*** and the receptive field size is ***x*** x **y**, then each neuron in the convolutional layer will have weights to ***x*** x ***y*** x ***d*** area in the input volume that has a total of **x*y*d** weights. The size of the output volume is controlled by three parameters; the depth, stride and zero-padding (Dumoulin & Visin, 2016).

- The depth corresponds to the number of filters that are used in the learning to look for something different in the input. If the raw image input is taken as the input to the first convolutional layer, different neurons along the depth dimension will be activated when there are various oriented edges, colors etc.
- When sliding the filter across the input volume, strides are specified. The amount by which the filter shifts is the stride. When moving the filter one pixel at a time, the stride is 1. When filter jumps two pixels at a time while moving around the input, the stride is 2.
- The size of the zero padding is a hyperparameter. When applying filters to an input volume, the resulted spatial size of output volume will decrease. As one keeps applying convolutional layers, the size of the output volume will decrease. However, it is sometimes preferential to preserve as much information from original input volume to extract low level features from the edges of images. Thus, spatial size of the output volume is often kept the same as the size of input volume. Zero padding is a feature which controls the spatial size of the output volume. It pads the input volume with zeros around the border to accomplish this (Dumoulin & Visin, 2016, Luo et al., 2016).

The spatial size of the output volume can be computed as a function of the input size (W), the receptive field size of the convolutional layer neurons (F), the strides applied (S) and the used zero padding (P) on the border (Dumoulin & Visin, 2016).

The spatial size of the output volume is calculated as

$$(W-F+2P)/S+1$$

Pooling layer. Generally, a pooling layer is inserted in between successive convolutional layers in convolutional network. It reduces the spatial size of the representation in order to reduce the parameters used in the network. A pooling layer operates independently on every depth slice of the input and resizes it partially using the MAX operation, which is a more popular operation than averaging or other operations (Boureau, Ponce, & LeCun, 2010) . Usually it takes a filter of size 2x2 and a stride of same length and every MAX operation is taking a maximum over 4 numbers. The general features of pooling layers are

- The pooling layer accepts the volume size of $\mathbf{W_1 \times H_1 \times D_1}$
- It requires two hyperparameters; their partial extent **F** and the stride **S**
- No zero padding is applied before pooling layers (Zeiler & Fergus, 2013)

In practice, there are only two variations of max pooling layer; a pooling layer with F=3, S=2 and F=2. S=3.This is due to the destructive nature of pooling layers with larger receptive fields. However, most of the time pooling operations are not used in convolutional architectures, and it only consists of a stack of convolutional layers (Dumoulin & Visin, 2016). A common method is using a larger stride in convolutional layer once in a while to reduce the size of the output of the convolutional layer. Moreover, by discarding pooling layers, good generative models can be trained. Therefore, modern convolutional architectures seldom use pooling layers.

Fully connected layer. In a fully-connected layer, neurons are fully connected to all activations in the previous layer, whose structure is same as regular neural networks. The only difference between fully-connected layers and convolutional layers is, convolutional layers are connected to only a local region of the input. Fully-connected layer takes the input volume and outputs an N-dimensional vector. In an output layer, N is the number of classes in which one is

trying to classify an image (Egmont-Petersen, de Ridder, & Handels, 2002). A fully-connected layer that is also an output layer takes the output of the previous layer (a convolutional, layer of activation function or pooling layer), which represents the activation map of high-level features, and determines which features are associated to a particular class.

Discrete convolutions. Affine transformations are applied in neural networks. In Affine transformations, a vector is received as input and is multiplied with a matrix to produce an output. For any type of input such as an image, a sound clip or any other collection of features, their representation should be flattened before go for the transformation. These types of inputs are stored as multi-dimensional arrays, and they feature one or more axes for which ordering matters (Dumoulin & Visin, 2016, p.6). However when an affine transformation is applied, all axes are considered equally; their topological information is not considered. A discrete convolution is a linear transformation in which only few input units affect the given output. The product between each kernel element and the input element that overlaps it, is computed at each location and summed up to get the output of the current location. The same method will be repeated using different kernels to obtain many output features (Dumoulin & Visin, 2016, p.6). The kernels in the discrete convolution has a shape with some permutation (n, m, k_1 ……., k_N), where

n = number of output feature maps

m =number of input feature maps

k_j = kernel size along axis j

Choosing Hyperparameters. The structure of the convolutional architecture is based on the type of data such as data size, complexity of image, and the type of image processing task. When looking at the selected database, programmers can choose hyperparameters to find the

right combination that creates abstractions of the image at a proper scale. In addition to the three main types of layers in convolutional architectures, some other layers such as ReLU and dropout layers are applied in practice.

Dropout Layer. The function of dropout layers is to reduce the problem of overfitting. Sometimes training examples are classified well by the model, but the model does not perform well on unseen data. The idea behind the addition of dropout layer is to drop (ignore)a random set of activations in that layer by setting that to zero (Nair & Hinton, 2010). After dropping out some of the activations, the network learns to correctly classify images with only part of its neurons. A dropout layer is only utilized in training, and it avoids the network overfitting the training data.

Activation functions. Every activation function (or non-linearity) takes a single number and performs a certain fixed mathematical operation on it. There are a number of common activation functions in use with neural networks.

***Sigmoid activation function*.** Figure 2a shows the image of sigmoid activation function. It generally takes a real-valued number and squashes it into the range between 0 and 1. Here, large negative numbers become 0 and large positive numbers become 1. Historically, it has been used frequent, as it shows nice interpretation on firing rate of a neuron; from not firing at all (0) to fully-saturated firing at an assumed maximum frequency (1).

Tanh activation function. Figure 2b illustrates the image of tanh activation function. By the use of tanh activation function, it squashes a real-valued number to the range [-1, 1]. The output of that is zero-centered. Thus, in practice the tanh non-linearity is preferred to the sigmoid nonlinearity.

ReLU (Rectified Linear Units) activation function. Sometimes, a layer is applied immediately after each convolution layer in order to introduce nonlinearity to the system (Nair & Hinton, 2010). After the convolutional layers, ReLU layers introduce nonlinearity to the system. Researchers have recently found that ReLU performs well as other nonlinear functions (such as the sigmoid function) due to its faster learning ability without making a significant difference in accuracy (Le, Jaitly, & Hinton, 2015). Further, it helps to alleviate the vanishing gradient issue, where the higher layers of the network train slowly due to a decrease of gradient exponentially through the layers. Figure 3 shows the image of ReLU activation function. The ReLu layer applies the function $f(x) = \max(0,x)$ to all the values in the input volume in order to add nonlinearity to the model (Nair & Hinton, 2010).

Softmax activation function. As in sigmoid function, the softmax function squashes the output of each unit to be between 0 and 1. Not only that, it divides each output such that the total sum of the outputs is equal to 1.

Significance of study

Convolutional neural networks have outperformed for feature extraction from images and recently achieved great successes in image processing applications. The major power of CNN relies on its deep architecture that allows the extraction of specific features from images at multiple levels of abstraction. As CNN uses relatively little preprocessing in comparison to other image classification algorithms (KNN, SVM etc.), they are inspired by many biomedical applications. Increasingly, CNN are applied for the diagnosis of diseases. In this study, dermoscopy images are classified as malignant and benign by creating different CNN architectures with varying parameters. Since this work studies the influence of the structure of CNN architecture (composition and sequence of layers) for a better CNN model, this study will

be helpful for other researchers in designing effective CNN models. This is hoped to lead to earlier detection times, which would improve chances of successful treatment of harmful melanoma instances.

Chapter 2- Literature Review

With the growing number of new cases in the world each year, melanoma has attracted considerable research efforts, specifically in developing software solutions to identify and analyze dermoscopy photographs of patients' skin lesions. The diversity of approaches is broad and each research covering varying combination of techniques such as low-level feature extractions and different machine learning approaches (Codella et al., 2015, p.120). Various methods are used in dermatology to characterize skin melanoma images. One method is the ABCD rule, which is used by dermatologists to classify melanomas. ABCD stands for asymmetry, border irregularity, color patterns and diameter. Gola Isasi et al. has presented an automated dermatological tool for the parameterization of melanomas, in which a system based on the ABCD rule and dermatological pattern recognition protocols were used. In their study they have used three automatic algorithms for skin image processing to recognize appropriate patterns for melanoma. The pattern recognition system developed by Gola Isasi et al. achieved 85% of accuracy, and was shown to be a reliable system for melanoma diagnosis (Gola Isasi, García Zapirain, & Méndez Zorrilla, 2011). A study done by Chang et al. developed an effective CADx (Computer-aided diagnosis) software to classify melanocytic and non-melanocytic skin lesions using conventional digital micrographs. Their method included conventional and new color feature extraction using SVM (support vector machine) and the system achieved sensitivity and specificity of 85.63% and 87.65%, respectively, and accuracy of 90.64 (Chang et al., 2013). To evaluate the dermoscopy lesions, Martin et al. report an approach based on ABCDE classification (E stands for Evolution), image processing and artificial neural networks, reaching a sensibility of 76.56% and a specificity of 87.58% (Marín, Alférez, Córdova, & González, 2015).

Input images contain both healthy and melanoma lesion parts. As this may mislead the training of convolutional neural networks, melanoma lesions should be separated from healthy areas (normal) of the skin. However, cropping healthy skin may cause the loss of important information such as color difference between healthy skin and lesion that is used to discriminate melanoma from benign condition (Nasr-Esfahani et al., 2016). Segmentation of the skin lesion from dermoscopy images is an important aspect in melanoma identification, because dermatologists use the shape of skin lesion in recognition (Yang et al., 2017). Therefore, current skin lesion classification follow three steps; i) lesion segmentation ii) feature extraction from segmented region and iii) lesion classification. More often, preprocessing step is performed before segmentation to reduce the noise of the image (Fornaciali, Carvalho, Bittencourt, Avila, & Valle, 2016). Hence most of researchers impose segmentation to their image classification systems. Research work by Cavalcanti et al have presented an automatic classification system comprised of preprocessing, segmentation and feature extraction steps. They have used a two stage classifier aiming to reduce the error rate of melanoma cases (specifically, false negative cases) (Cavalcanti, Scharcanski, & Baranoski, 2013). Their proposed framework which combines ABCD rule and melanin-variation feature extraction technique has shown 100% accuracy in melanoma classification. In the literature, there are several research works done for melanoma classification based on segmentation followed by classification using SVM and kNN. One such melanoma recognition system proposed by Abbas et al. consists of six consecutive stages; color space transformation, preprocessing, , black-frame and hair artifacts removal, lesion-area segmentation, feature quantification and normalization, and feature selection and classification (Abbas, Emre Celebi, Garcia, & Ahmad, 2013). SVM learning algorithm was used for classification in this study and they achieved a sensitivity of 88.2% and specificity of 91.3%.

Barata et al. also described two strategies to obtain a lesion classification based on extracting local and global features of skin lesions. This method involves automatic segmentation, feature extraction and training a classifier to perform binary classification as melanoma or benign and they achieved sensitivity of 98% and specificity of 79% (Barata, Ruela, Francisco, Mendonça, & Marques, 2014). Recently, local methods have been proposed to classify melanoma skin lesions. Situ et al. described an algorithm for lesion classification using bag-of-features approach (BoF). In their method, they represent each image in 16x16 grid on the lesion and using Gabor-like filters they extract 23 features of the lesion with the best performance of 82% on a dataset of 100 dermoscopy images (Situ, Yuan, Chen, & Zouridakis, 2008).

When Krizhevsky et al. won the classification challenge of the ImageNet large scale visual recognition challenge 2012, Deep Neural Networks (DNN) reached to the top in computer vision (Krizhevsky et al., 2012). As Deep Learning methods have shown promising solutions for many applications such as natural language processing, speech and facial recognition, object detection and image classification at this time, there is a trend to use Deep Learning methods for high accuracy medical image classifications. Some researchers started to use Convolutional Neural Networks (CNNs) for melanoma classification due to its higher discriminating capability in recognition. CNN is a type of Deep Learning method, where trainable filters are applied on raw data to extract selectable features automatically from complex systems (Nasr-Esfahani et al., 2016). In one of the research studies, a CNN classifier was trained by large number of trained clinical skin images to distinguish melanoma from benign cases reaching 81% of sensitivity and 80% of specificity (Nasr-Esfahani et al., 2016). Cıcero et al. have presented a technique for the classification of non-malignant melanotic lesions from the melanomas. In their study, they classified the image dataset and transformed the image dataset with rotations to make the deep

network invariant to rotations in order to improve generalization. For the evaluation they used the deep learning framework Caffe and the deep neural network ResNet as a feature extractor (Cıcero et al., 2016). Recently, a novel method for melanoma recognition has been proposed using fully convolutional residual network (FCRN) for accurate skin lesion segmentation with a multi-scale contextual information integration scheme (Yu et al., 2017).

Codella et al. presented an approach for identification of melanoma in dermoscopy images by combining deep learning, sparse coding and support vector machine (SVM) learning algorithms. They claimed that their approach is beneficial because it used unsupervised learning within the domain and feature transfer from domain of natural photographs, eliminating the need of annotated data (Codella et al., 2015). Focusing on avoiding lesion segmentation and complex preprocessing, Kawahara et al. used a fully-convolutional neural network to extract multi-scale features from skin lesions in order to classify melanomas with a higher accuracy than other state-of-the-art techniques (Kawahara et al., 2016). Very recently, Codella et al. proposed a system that combines deep leaning with well-established machine learning techniques to create ensemble of methods that can segment skin images for analysis (Codella et al., 2016). Towards the goal of melanoma recognition with a higher accuracy rate, this study trains a model to identify melanoma-positive dermoscopy images using a novel CNN model.

Chapter 3- Methodology

This section presents techniques that were used in developing a novel convolutional neural network (CNN) model to classify melanoma as malignant and benign. This CNN model was trained using skin lesion images from a publicly available dataset. Here, few CNN architectures were created by varying parameters and then a novel CNN model with high accuracy was proposed by choosing the most appropriate features/parameters from different architectures. To evaluate the proposed model, its performance was compared with few published network architectures.

Dataset

The International Skin Imaging Collaboration (ISIC) has released a collection of annotated dermoscopy images of melanoma for the 2016/2017 International Symposium on Biomedical Imaging challenge (https://isic-archive.com/#). This study uses dermoscopy images that are publicly available in International Skin Imaging Collaboration archive. These are stored as RGB images with varied sizes in JPG format. There were total 2236 dermoscopy images in the dataset (1118 each for malignant and benign). Sample images from the dataset are shown in Figure 4.

Preprocessing

Raw input images (dermoscopy digital images) were preprocessed in different ways to apply them into convolutional neural networks. In order to analyze dermoscopy images of melanoma skin lesions, this study used labeled melanoma dermoscopy images. Before fed them into convolutional architectures, these images were resized to a constant value in order to reduce the problem complexity (Gola Isasi, García Zapirain, & Méndez Zorrilla, 2011). Therefore all images were resized to 128x128 for few experiments and then resized them to 50x50 dimensions using OpenCV. Since the color information of input image is not necessary to detect skin lesions,

the RGB images were converted to grayscale image. The grayscale conversion was performed by summating the values of each pixel of the red, green and blue images (Gola Isasi et al., 2011, p.745).

Data Augmentation

Generally, there should be a large number of samples for proper training for any CNN model (Nasr-Esfahani et al., 2016). However, because of the difficulties in image collection and labeling, there are a limited number of images for melanoma detection. Therefore, for the purpose of artificially expand the training data set, data augmentation techniques were followed. In this study, dermoscopy images were subjected to data augmentation by elastic distortion, perspective transformation (there are a total of 12 different types), preserving rotation, size preserving shearing and cropping. An augmentation tool, called Augmentor was used for data augmentation in this study. Augmentor is a Python package designed to aid the augmentation and artificial generation of image data for machine learning. The dataset was increased 6 times total using the above mentioned augmentation techniques. These augmented feature vectors were used as additional samples to train the classifier and also for testing and validation.

System Implementation

The proposed model is implemented with Python 3.5.2 based on Tensorflow 1.1.0 and TFlearn 0.3 high level API on a server with Intel Core i7 processor with 16 GB of RAM.

Convolutional Neural Network Architecture

In order to classify melanoma as malignant or benign, several convolutional neural network (CNN) architectures were created by varying

1) No. of layers and their arrangement
2) No. of training steps and learning rate

3) Activation functions
4) Optimizer

This section describes all the CNN architectures in detail. The structure of the CNN architectures include a series of convolutional layers, pooling layers, several activation functions, single and deep fully-connected layer. Some tunable parameters were included to this network. The size of the convolutional kernels were managed as a constant value across all stages. The number of filters in the convolutional layers were also tuned. Then, preprocessed dermoscopy images were fed into many different CNN architectures. Figure 5 presents a diagram of a general CNN architecture used for skin lesion analysis. Following are the four CNN architectures designed (default) prior to construct the model to classify melanoma as malignant and benign.

CNN architecture 1

1. Convolutional Layer #1: Applies 32, 2x2 filters with ReLU activation function.
2. Pooling Layer #1: Performs max pooling with a 2x2 filter and stride of 1.
3. Convolutional Layer #2: Applies 64 2x2 filters with ReLU activation function.
4. Pooling Layer #2: Again performs max pooling with a 2x2 filter and stride of 1.
5. Fully-connected Layer: 2 neurons, one for each class (malignant or benign)

CNN architecture 2

1. Convolutional Layer #1: Applies 32, 2x2 filters with ReLU activation function.
2. Pooling Layer #1: Performs max pooling with a 2x2 filter and stride of 1.
3. Convolutional Layer #2: Applies 64 2x2 filters with ReLU activation function.
4. Pooling Layer #2: Performs max pooling with a 2x2 filter and stride of 1.
5. Convolutional Layer #3: Applies 32, 2x2 filters with ReLU activation function.
6. Pooling Layer #3: Performs max pooling with a 2x2 filter and stride of 1.

7. Convolutional Layer #4: Applies 64 2x2 filters with ReLU activation function.
8. Pooling Layer #4: Again performs max pooling with a 2x2 filter and stride of 1.
9. Fully-connected Layer #1: 1024 neurons with dropout regularization rate of 0.8.
10. Fully-connected Layer #2: 2 neurons, one for each class (malignant or benign)

CNN architecture 3

The structure of this architecture is similar to the CNN architecture 2. But this architecture consists of 6 convolutional and 6 max-pooling layers.

CNN architecture 4

This architecture contains 10 convolutional layers and 10 max-pooling layers. All other parameters are same to architecture 2 and 3.

Classification

There were total 2236 dermoscopy images in the dataset. From that 20% were selected as testing data and the rest was divided in to 80:20 ratio for training and validation respectively. The last fully connected layer in each CNN architecture provided probability of class based on votes.

Evaluation Metrics

In the first part of the study, better CNN architectures were selected based on total loss, validation loss, accuracy and validation accuracy. CNN architectures with lower loss and greater accuracy were considered in creating a novel CNN model.

For quantitative evaluation of the performance of the proposed system, three commonly used metrics; Recall or Sensitivity (SENS), Classification accuracy (ACC), and Specificity (SPEC) were used. The metrics are defined as follow:

$$Recall\ (Sensitivity) = \frac{True\ positive\ cases}{True\ positive\ cases + False\ nagative\ cases}$$

$$Classification\ Accuracy = \frac{True\ detected\ cases}{All\ cases}$$

$$Specificity = \frac{True\ detected\ non-melanoma\ cases}{All\ non-melanoma\ cases}$$

Experiments on other published CNN models

The same dermoscopy image dataset was fed into four published CNN models namely AlexNet, VGGNet, Highway CNN and Google Inception v3. Then their performance on classifying melanoma was compared.

Design of a novel CNN model

According to the results obtained by the tested CNN architectures, a general CNN model was created by doing variations across layers. By evaluating the performance of published CNN models, fine tuning was done to the created model to propose a novel CNN model with high accuracy and recall.

Chapter 4 - Results and Discussion

In this study, a novel CNN model was proposed by optimizing the conditions of the CNN architecture based on the performance of the previously created architectures. All created CNN architectures were constructed using TensorFlow. TensorFlow provides high-level API to construct a neural network easily. It facilitates the creation of convolutional layers, dense layers (fully-connected), adding activation functions and applying dropout regularization. All CNN architectures created in this study composed of number of convolutional layers, pooling layers fully connected layers, dropout layers. Convolutional layers apply convolutional filters to the image and the layer performs mathematical operations for each sub region, to produce a single value in the output of the feature map. Pooling layers reduce the size of the feature maps by using some function to summarize subregions. Max pooling was utilized in constructing the CNN architectures that extract subregions of the feature map, keeps their maximum value and disregard all other values. Fully-connected layers perform the classification based on the features extracted by convolutional layers. The purpose of applying dropout layer is to reduce overfitting. When developing a CNN model for classifications, we usually come up with several questions; how many layers to use, how many convolutional layers, what are the filter sizes or the values for stride and padding. But there is not a standard way to do that, because the network largely depends on the type of data, which can be vary by size, complexity of the image etc. Therefore, one way to think about to choose the hyperparameters is to find the right combination.

A typical CNN is composed of a stack of layers to perform feature extraction. In the first part of this study, four different CNN architectures were created by varying the number of layers and its arrangement. The objective of using different CNN architectures is to identify good architectures which extract high-level features of melanoma from dermoscopic lesions as much

as possible. Four CNN architectures used in this study contains 2, 4, 6 and 10 convolving layers respectively with the kernel size of 2x2. There is one pooling layer after each convolutional layer. In the first CNN architecture, there is only one fully-connected layer, whereas other three architectures contain two fully-connected layers. In these three architectures, a dropout layer is applied to the pooling layers before the second fully-connected layer. In addition to the number of layers in the CNN architecture, there are two parameters that we can change to modify the behavior of each layer. After choosing the kernel size, we have to choose the stride and padding. A convolutional layer's output shape is affected by the shape of its input and the choice of kernel size, zero padding and strides. Stride controls how the kernel convolves around the input. In other words it measures how much the kernel is translated and it shows how much of the output is retained. Stride is normally set in a way, so that the output volume is an integer and not a fraction. For all CNN architectures of this study, the stride was set to 1. Because by setting stride to 1, the receptive fields overlap more and it increases the feature extraction capability. As keep applying the convolutional layers, the size of the output volume decreases fast. If we can retain the output volume same to the original input volume, we can preserve as much details about the original input volume. By applying zero padding, the output volume remains same in size to the original input volume. Thus, to all convolutional layers in the created CNN architectures, zero padding was added.

For the first set of experiments, all images of 128x128 size were fed in to an architecture of 2 convolutional layers with 32 and 64 convolutional filters, 2 pooling layers and a fully-connected layer. Table 1 presents a summary of the experiments in the first set. When training steps were increased from 230 to 4600, there was not a significant change in training and validation accuracies. Then the input image size was reduced from 128x128 to 50x50 to see

whether it affects for the training efficiency of the model. The results showed that reduction of the size of images was not greatly affected to the loss or the accuracy of the training dataset. Therefore for the next models, the image size was maintained in 50x50, because it speeds up running the model even in low computational power. When introduced Tanh activation function for convolutional layers in place of ReLU activation, the model produced lower loss and greater training and validation accuracies. Also, the total loss was decreased as goes along with training steps, which is an indicator of good performance. However, when Tanh was used in the fully-connected layer, it didn't even calculate the loss, because the back propagation was not occurred. As shown in Table 1, the models performed better with Tanh activation function in convolutional layers than ReLU and sigmoid activation. When compare the performance of models, Adam optimizer worked better than Stochastic gradient descent (SGD) optimizer. To examine the effect of data augmentation for the performance of models, data augmentation was carried out to some experiments. The models with data augmentation showed higher accuracy (both training and validation) and lower loss than the models without data augmentation. In the data augmented models also, Tanh and Adam were the best activation function and best optimizer respectively.

For the second set of experiments, a different architecture with 4 convolution layers (alternating 32 and 64 convolutional filters) and 4 pooling layers was used. There was a fully connected layer followed by a dropout layer of 0.8 and again a fully-connected layer. For all the networks that used this architecture, followed Softmax activation function in second fully-connected layer. The second set of experiments were summarized in Table 2. When increasing the training steps from 690 to 5024, and reducing learning rate from 1E-3 to 1E-6, the total loss was significantly dropped and training and validation accuracies were increased to ~56%. Therefore for this set of experiments best number of training steps was 5024 and best learning

rate was 1E-6. Figure 8 shows that Tanh activation moreover reduces the total loss while enhancing the accuracy to 58.20%. The performance of models were not affected by reducing the dropout layer to 0.4, from 0.8.

For the third set of experiments, the number of alternating convolutional and pooling layers were increased to 6 each. For these networks 5024 training steps, 1E-6 learning rate and Adam optimizer for regression were used. As shown in Table 2 and 3, all the five tested CNN architectures showed better results for loss and accuracy than the previous networks with 4 convolutional and pooling layer pairs. When Tanh activation was introduced to all convolutional layers replacing ReLU activation, it showed similar training and validation accuracies and total loss and validation loss (Figure 10 and 11). These results demonstrate that Adam optimizer performs better than SGD in regression in all network structures.

For the next set of experiments, number of alternating convolutional and pooling layers were further increased to 10 each. According to the results obtained from these networks (Table 4), the performance still places the networks in the same place as with 6 convolutional and pooling layers. This demonstrates that, further increment of layers doesn't make a contribution to greater accuracies and lower loss. Based on these results, it can be concluded that deeper structures do not always provide better performance. Among the tested networks in this structure, the one with Tanh activation on convolutional layers and Adam optimizer regression performed better than other networks with ReLU and Sigmoid activation (Figure 12).

In the final set of experiments, few well-known CNN models were used to train the same dataset with 5024 training steps and 1E-6 learning rate which used in this study . The AlexNet that regarded as one of the most influential publication in the field of computer vision and

convolutional neural network, has the structure of 5 convolutional layers, max-pooling layers, dropout layers and three fully-connected layers (Krizhevsky et al., 2012) with ReLU activation. As shown in the Table 5, this network showed total loss 0.53, validation loss 0.73, training accuracy 72.76% and validation accuracy 60.46%. Then the same dataset was fed into the network of VGG Net published in 2014, which is a 19 layer CNN with 2x2 max-pooling layers of stride 2 (Simonyan & Zisserman, 2014). Figure 14 shows that it performs better than AlexNet for the given dataset with total loss 0.17, validation loss 2.31, training accuracy 97.96% and validation accuracy 65.69%. When used the Highway convolutional network architecture that was published in 2015 (Srivastava, Greff, & Schmidhuber, 2015), it showed total loss 0.22, validation loss 1.54, training accuracy 93.18% and validation accuracy 60.88% (Figure 15). This architecture contains highway convolutional layers with normalized function. By comparing these three network architectures, it showed that simple architectures performs well in image classification. When it comes to the GoogleNet (inception v3), the idea of simplicity in the network architecture is out as it contains huge number of layers. GoogleNet was one of the first CNN architectures that strayed from the general approach of simply stacking convolutional and pooling layers on top of each other in a sequential structure (Szegedy et al., 2015). In the architecture of GoogleNet, the layers were not arranged in a sequential order. In the general approach we have to make a choice of whether to have a pooling operation or convolutional operation. But in Inception v3 module, all these operations perform parallel. When the training dataset was applied to Inception v3, it showed total loss 0.37 validation loss 0.82, training accuracy 81.83% and validation accuracy 61.68% (Figure 16). The performance of Inception v3 places it at the third rank in the classification of melanoma in this study, while VGG and Highway CNN models place first and second ranks respectively.

By comparing the outcome of each architecture it was easier to get a general idea, how to optimize the array of layers in the architecture to train a better model with lowest loss and best accuracy. The lowest total loss and validation loss obtained for created CNN architecture were 0.67. The maximum training accuracy and the validation accuracy were 58.2% and 57.92% respectively. Many of the research on CNNs showed that depth of neural networks is a crucial ingredient for their success. The published paper of VGGNet reinforced the notion that convolutional neural networks have to have a deep network of layers in order for hierarchical representation of visual data to work (Simonyan & Zisserman, 2014). In their study, they increased the depth of network by using an architecture with very small convolutional filters (3x3) which shows a significant improvement on the prior-art configurations by pushing the depth to 16-19 weight layers (Simonyan & Zisserman, 2014). Therefore, when the structure goes deeper and simple, the success rate of the network become high. Based on that the structure of the array of layers were changed to two successive convolutional layers followed by a max-pooling layer. Following is the structure of the CNN architecture.

Convolutional Layer #1: Applies 16, 5x5 filters with Tanh activation function.

Convolutional Layer #2: Applies 16, 5x5 filters with Tanh activation function.

Pooling Layer #1: Performs max pooling with a 2x2 filter and stride of 1.

This structure was repeated six times by doubling the filter size of convolutional layer from one to next. Finally there is a fully connected later with 0.8 dropout regularization. This network showed total loss 0.71, validation loss 0.70, training accuracy 48.56% and validation accuracy 69.82%. Training of this model was difficult, as this structure is deep, and it took lots of computational cost. Network training becomes more difficult with increasing depth. To overcome this problem Srivastava et al. proposed a highway network with hundreds of layers,

which can be trained directly using SGD with variety of activation functions (Srivastava et al., 2015). For the proposed model, convolutional highway layers were used because convolutional highway layers utilize weight sharing and local receptive fields for non-linear transformation. For the proposed model, batch normalization was applied after each max-pooling operation because this technique accelerate deep network training by reducing internal covariate shift (Ioffe & Szegedy, 2015). Following is the proposed architecture.

1. Highway Convolutional Layer #1: Applies 32, 3x3 filters with elu activation function.
2. Highway Convolutional Layer #2: Applies 32, 2x2 filters with elu activation function.
3. Highway Convolutional Layer #3: Applies 32, 1x1 filters with elu activation function.
4. Pooling Layer #1: Performs max pooling with a 2x2 filter and stride of 1.
5. Batch normalization
6. Highway Convolutional Layer #4: Applies 64, 3x3 filters with elu activation function.
7. Highway Convolutional Layer #5: Applies 64, 2x2 filters with elu activation function.
8. Highway Convolutional Layer #6: Applies 64, 1x1 filters with elu activation function.
9. Pooling Layer #2: Performs max pooling with a 2x2 filter and stride of 1.
10. Batch normalization
11. Highway Convolutional Layer #7: Applies 32, 3x3 filters with elu activation function.
12. Highway Convolutional Layer #8: Applies 32, 2x2 filters with elu activation function.
13. Highway Convolutional Layer #9: Applies 32, 1x1 filters with elu activation function.
14. Pooling Layer #3: Performs max pooling with a 2x2 filter and stride of 1.
15. Batch normalization
16. Highway Convolutional Layer #10: Applies 64, 3x3 filters with elu activation function.
17. Highway Convolutional Layer #11: Applies 64, 2x2 filters with elu activation function.

18. Highway Convolutional Layer #12: Applies 64, 1x1 filters with elu activation function.
19. Pooling Layer #4: Performs max pooling with a 2x2 filter and stride of 1.
20. Batch normalization
21. Highway Convolutional Layer #13: Applies 32, 3x3 filters with elu activation function.
22. Highway Convolutional Layer #14: Applies 32, 2x2 filters with elu activation function.
23. Highway Convolutional Layer #15: Applies 32, 1x1 filters with elu activation function.
24. Pooling Layer #5: Performs max pooling with a 2x2 filter and stride of 1.
25. Batch normalization
26. Highway Convolutional Layer #16: Applies 64, 3x3 filters with elu activation function.
27. Highway Convolutional Layer #17: Applies 64, 2x2 filters with elu activation function.
28. Highway Convolutional Layer #18: Applies 64, 1x1 filters with elu activation function.
29. Pooling Layer #6: Performs max pooling with a 2x2 filter and stride of 1.
30. Batch normalization
31. Fully-connected Layer #1: 128 neurons with elu activation function.
32. Fully-connected Layer #2: 256 neurons with elu activation function.
33. Fully-connected Layer #3: 1024 neurons with elu activation function.
34. Dropout regularization (0.8)
35. Fully-connected Layer #4: 2 neurons, one for each class; Benign and Malignant, with Softmax activation function.

This structure showed total loss 0.07, validation loss 1.05, training accuracy 98% and validation accuracy 64.57%. This structure showed lowest loss and greater accuracy in comparison to tested other published networks. The sensitivity, specificity and classification accuracy of this model were 46%, 60% and 64% respectively.

Chapter 5- Conclusion

This paper proposes a novel CNN for the classification of melanoma from dermoscopic images of skin. The method was evaluated on the largest public benchmark for melanoma recognition available. This paper presented an extended study on how to optimize the conditions of CNN architecture to train a model to classify melanoma as malignant and benign with high validation accuracy and low loss. This study demonstrated the performance of a trained model by varying the array of different layers, activation functions, number of filters and filter size, optimizers and dropout regularization. Then the same dataset was used to train the models of state-of-the-art networks such as AlexNet, Highway CNN, VGGNet and Google Inception v3. Based on the performance of each tested CNN architecture and state-of-the-art networks, a novel neural network model was proposed. The proposed model is a deep highway CNN with batch normalization. The proposed system produced training accuracy of 98% and validation accuracy of 64.57% and it performs similar to the state-of the-art networks.

References

Abbas, Q., Emre Celebi, M., Garcia, I. F., & Ahmad, W. (2013). Melanoma recognition framework based on expert definition of ABCD for dermoscopic images. *Skin Research and Technology*, *19*(1), e93–e102. https://doi.org/10.1111/j.1600-0846.2012.00614.x

Barata, C., Ruela, M., Francisco, M., Mendonça, T., & Marques, J. S. (2014). Two systems for the detection of melanomas in dermoscopy images using texture and color features. *IEEE Systems Journal*, *8*(3), 965–979.

Boureau, Y.-L., Ponce, J., & LeCun, Y. (2010). A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 111–118). Retrieved from http://machinelearning.wustl.edu/mlpapers/paper_files/icml2010_BoureauPL10.pdf

Cavalcanti, P. G., Scharcanski, J., & Baranoski, G. V. G. (2013). A two-stage approach for discriminating melanocytic skin lesions using standard cameras. *Expert Systems with Applications*, *40*(10), 4054–4064. https://doi.org/10.1016/j.eswa.2013.01.002

Celebi, M. E., Kingravi, H. A., Uddin, B., Iyatomi, H., Aslandogan, Y. A., Stoecker, W. V., & Moss, R. H. (2007). A methodological approach to the classification of dermoscopy images. *Computerized Medical Imaging and Graphics*, *31*(6), 362–373. https://doi.org/10.1016/j.compmedimag.2007.01.003

Chang, W.-Y., Huang, A., Yang, C.-Y., Lee, C.-H., Chen, Y.-C., Wu, T.-Y., & Chen, G.-S. (2013). Computer-aided diagnosis of skin lesions using conventional digital photography: a reliability and feasibility study. *PloS One*, *8*(11), e76212.

Cıcero, F., Oliveira, A., Botelho, G., & da Computaçao, C. de C. (2016). Deep learning and convolutional neural networks in the aid of the classification of melanoma. SIBGRAPI.

Retrieved from http://sibgrapi.sid.inpe.br/col/sid.inpe.br/sibgrapi/2016/09.01.15.04/doc/16.pdf

Codella, N., Cai, J., Abedini, M., Garnavi, R., Halpern, A., & Smith, J. R. (2015). Deep Learning, Sparse Coding, and SVM for Melanoma Recognition in Dermoscopy Images. In *Machine Learning in Medical Imaging* (pp. 118–126). Springer, Cham. https://doi.org/10.1007/978-3-319-24888-2_15

Codella, N., Nguyen, Q.-B., Pankanti, S., Gutman, D., Helba, B., Halpern, A., & Smith, J. R. (2016). Deep Learning Ensembles for Melanoma Recognition in Dermoscopy Images. *arXiv:1610.04662 [Cs]*. Retrieved from http://arxiv.org/abs/1610.04662

Dumoulin, V., & Visin, F. (2016). A guide to convolution arithmetic for deep learning. *arXiv Preprint arXiv:1603.07285*. Retrieved from https://arxiv.org/abs/1603.07285

Egmont-Petersen, M., de Ridder, D., & Handels, H. (2002). Image processing with neural networks—a review. *Pattern Recognition*, *35*(10), 2279–2301. https://doi.org/10.1016/S0031-3203(01)00178-9

Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, *542*(7639), 115–118.

Fornaciali, M., Carvalho, M., Bittencourt, F. V., Avila, S., & Valle, E. (2016). Towards automated melanoma screening: Proper computer vision & reliable results. *arXiv Preprint arXiv:1604.04024*. Retrieved from https://arxiv.org/abs/1604.04024

Ganster, H., Pinz, P., Rohrer, R., Wildling, E., Binder, M., & Kittler, H. (2001). Automated melanoma recognition. *IEEE Transactions on Medical Imaging*, *20*(3), 233–239. https://doi.org/10.1109/42.918473

Gola Isasi, A., García Zapirain, B., & Méndez Zorrilla, A. (2011). Melanomas non-invasive diagnosis application based on the ABCD rule and pattern recognition image processing algorithms. *Computers in Biology and Medicine*, *41*(9), 742–755. https://doi.org/10.1016/j.compbiomed.2011.06.010

Harley, A. W. (2015). An interactive node-link visualization of convolutional neural networks. In *International Symposium on Visual Computing* (pp. 867–877). Springer. Retrieved from http://link.springer.com/chapter/10.1007/978-3-319-27857-5_77

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning* (pp. 448–456).

Kawahara, J., BenTaieb, A., & Hamarneh, G. (2016). Deep features to classify skin lesions. In *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)* (pp. 1397–1400). https://doi.org/10.1109/ISBI.2016.7493528

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1097–1105). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Le, Q. V., Jaitly, N., & Hinton, G. E. (2015). A Simple Way to Initialize Recurrent Networks of Rectified Linear Units. *arXiv:1504.00941 [Cs]*. Retrieved from http://arxiv.org/abs/1504.00941

Luo, W., Li, Y., Urtasun, R., & Zemel, R. (2016). Understanding the effective receptive field in deep convolutional neural networks. In *Advances in Neural Information Processing*

Systems (pp. 4898–4906). Retrieved from http://papers.nips.cc/paper/6202-understanding-the-effective-receptive-field-in-deep-convolutional-neural-networks

Marín, C., Alférez, G. H., Córdova, J., & González, V. (2015). Detection of melanoma through image recognition and artificial neural networks. In *World Congress on Medical Physics and Biomedical Engineering, June 7-12, 2015, Toronto, Canada* (pp. 832–835). Springer. Retrieved from http://link.springer.com/chapter/10.1007/978-3-319-19387-8_204

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 807–814). Retrieved from http://machinelearning.wustl.edu/mlpapers/paper_files/icml2010_NairH10.pdf

Nasr-Esfahani, E., Samavi, S., Karimi, N., Soroushmehr, S. M. R., Jafari, M. H., Ward, K., & Najarian, K. (2016). Melanoma detection by analysis of clinical images using convolutional neural network. In *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 1373–1376). https://doi.org/10.1109/EMBC.2016.7590963

Simonyan, K., & Zisserman, A. (2014a). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [Cs]*. Retrieved from http://arxiv.org/abs/1409.1556

Simonyan, K., & Zisserman, A. (2014b). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [Cs]*. Retrieved from http://arxiv.org/abs/1409.1556

Situ, N., Yuan, X., Chen, J., & Zouridakis, G. (2008). Malignant melanoma detection by bag-of-features classification. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE* (pp. 3110–3113). IEEE. Retrieved from http://ieeexplore.ieee.org/abstract/document/4649862/

Srivastava, R. K., Greff, K., & Schmidhuber, J. (2015). Highway networks. *arXiv Preprint arXiv:1505.00387*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9). Retrieved from http://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Szegedy_Going_Deeper_With_2015_CVPR_paper.html

Yu, L., Chen, H., Dou, Q., Qin, J., & Heng, P. A. (2017). Automated Melanoma Recognition in Dermoscopy Images via Very Deep Residual Networks. *IEEE Transactions on Medical Imaging*, *36*(4), 994–1004. https://doi.org/10.1109/TMI.2016.2642839

Zeiler, M. D., & Fergus, R. (2013). Stochastic Pooling for Regularization of Deep Convolutional Neural Networks. *arXiv:1301.3557 [Cs, Stat]*. Retrieved from http://arxiv.org/abs/1301.3557

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818–833). Springer. Retrieved from http://link.springer.com/chapter/10.1007/978-3-319-10590-1_53

Tables

Table 1. Accuracy and loss of all predictions in the *CNN architecture 1*.

	Aug	Training steps	Learning rate	Act. Function in Conv	Act. Function in FC	Optimizer	Total loss	Validation loss	Training Accuracy %	Validation accuracy %
a	no	230	1.00E-05	ReLU	Softmax	Adam	11.46	11.37	50.19	50.56
b	no	4600	1.00E-05	ReLU	Softmax	Adam	11.37	11.44	50.59	50.28
c	no	1150	1.00E-02	ReLU	Softmax	Adam	11.56	11.57	50.2	50.6
d	**no**	**1150**	**1.00E-06**	**Tanh**	Softmax	**Adam**	**0.69**	**0.69**	**52.83**	**51.4**
e	no	1150	1.00E-06	Softmax	Softmax	Adam	0.69	0.69	50.57	49.72
f	no	1150	1.00E-06	Sigmoid	Softmax	Adam	0.69	0.69	49.33	47.77
g	no	1150	1.00E-06	ReLU	Tanh	Adam	_	_	50.61	50.28
h	no	1150	1.00E-06	ReLU	Softmax	SGD	11.45	11.72	49.96	48.88
i	yes	15700	1.00E-02	ReLU	Softmax	Adam	11.47	11.51	50.13	50
j	yes	4710	1.00E-06	ReLU	Softmax	Adam	11.47	11.51	50.17	50
k	**yes**	**5024**	**1.00E-06**	**Tanh**	Softmax	**Adam**	**0.67**	**0.68**	**57.41**	**54.91**
l	yes	4710	1.00E-06	ReLU	Softmax	SGD	11.12	11.51	51.66	50
m	yes	5024	1.00E-06	Sigmoid	Softmax	Adam	0.72	0.69	53.9	54.08
n	yes	5024	1.00E-06	Tanh	Softmax	SGD	0.85	0.7	53.66	52.66

Table 2. Accuracy and loss of all predictions in the *CNN architecture 2*.

	Training steps	Leraning rate	Act. Function in Conv	Act. Function in FC 1	Dropout Layer	Optimizer	Total loss	Validation loss	Training Accuracy %	Validation accuracy %
a	690	1.00E-03	ReLU	ReLU	0.8	Adam	11.31	11.57	50.85	49.72
b	5024	1.00E-06	ReLU	ReLU	0.8	Adam	0.73	0.7	56.85	56.4
c	**5024**	**1.00E-06**	**Tanh**	**ReLU**	**0.8**	**Adam**	**0.67**	**0.67**	**58.2**	**57.92**
d	5024	1.00E-06	Sigmoid	ReLU	0.8	Adam	0.69	0.69	50.77	50
e	5024	1.00E-06	ReLU	ReLU	0.8	SGD	2.15	0.79	52.04	52.74
f	5024	1.00E-06	Tanh	ReLU	0.8	SGD	0.69	0.69	50.84	50.45
g	5024	1.00E-06	ReLU	ReLU	0.4	Adam	2	0.75	57.25	58.59
h	**5024**	**1.00E-06**	**ReLU**	**Tanh**	**0.8**	**Adam**	**0.68**	**0.66**	**56.11**	**61.6**

Table 3. Accuracy and loss of all predictions in the *CNN architecture 3*.

	Act. Function in Conv & FC1	Optimizer	Total loss	Validation loss	Training Accuracy %	Validation accuracy %
a	**ReLU**	**Adam**	**0.68**	**0.68**	**54.47**	**56.4**
b	**Tanh**	**Adam**	**0.69**	**0.69**	**54.98**	**57.07**
c	Sigmoid	Adam	0.69	0.69	52.49	50
d	ReLU	SGD	0.73	0.69	49.79	48.61
e	Tanh	SGD	0.69	0.69	49.9	50

Table 4. Accuracy and loss of all predictions in the *CNN architecture 4*.

	Act. Function in Conv & FC1	Optimizer	Total loss	Validation loss	Training Accuracy %	Validation accuracy %
a	ReLU	Adam	0.69	0.69	53.65	49.4
b	**Tanh**	**Adam**	**0.69**	**0.69**	**55.8**	**55.1**
c	Sigmoid	Adam	0.69	0.69	50.61	50
d	ReLU	SGD	0.69	0.69	48.74	49.95
e	Tanh	SGD	0.69	0.69	50.07	50

Table 5. Accuracy and loss of all predictions in published networks

	CNN Model	Total loss	Validation loss	Training Accuracy %	Validation accuracy %
a	AlexNet	0.53	0.73	72.76	60.46
b	VGGNet	0.17	2.31	97.96	65.69
c	Highway CNN	0.22	1.54	93.18	60.88
d	Google Inception v3	0.37	0.82	81.83	61.68

Figures

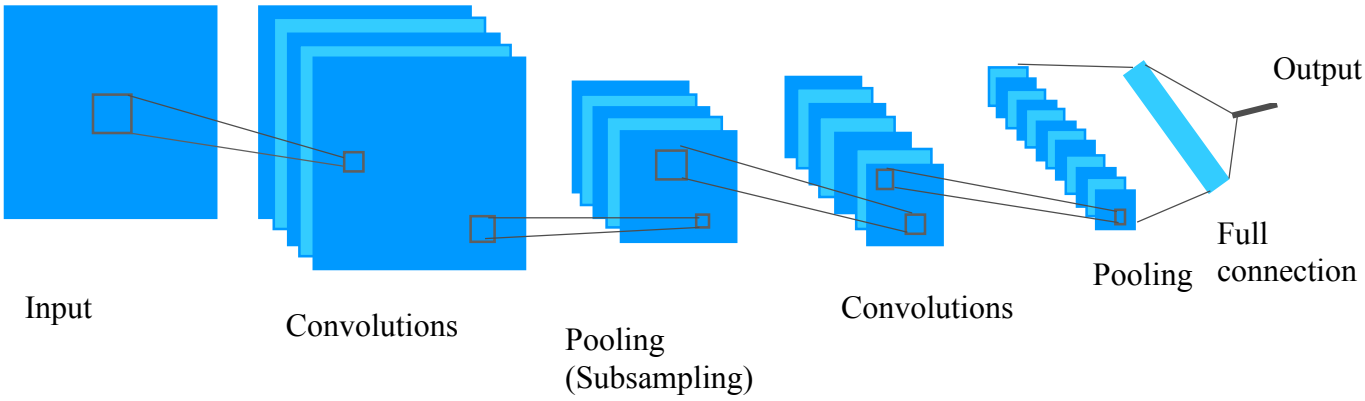


Figure 1: Typical illustration of convolutional neural network (Haley, 2015)

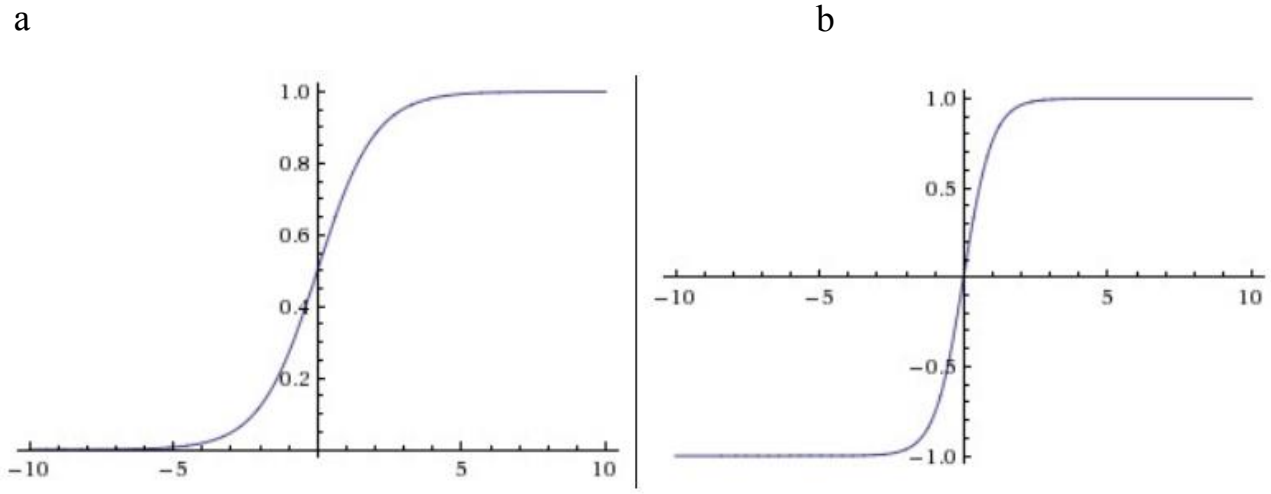


Figure 2: **a)** The sigmoid non-linearity squashes real numbers to range between [0, 1] **b)** The Tanh non-linearity squashes real numbers to range between [-1,1]

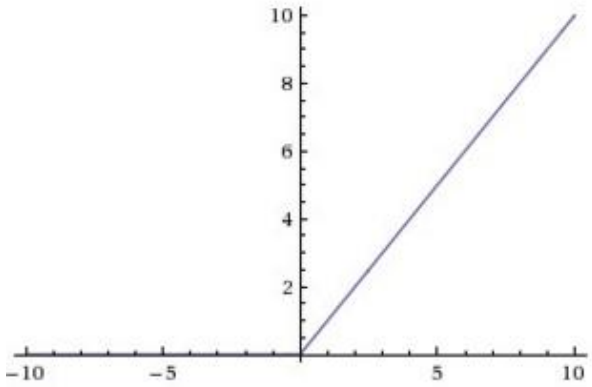


Figure 3: Rectified Linear Unit (ReLU) activation function, which is zero when x < 0 and then linear with slope 1 when x > 0.

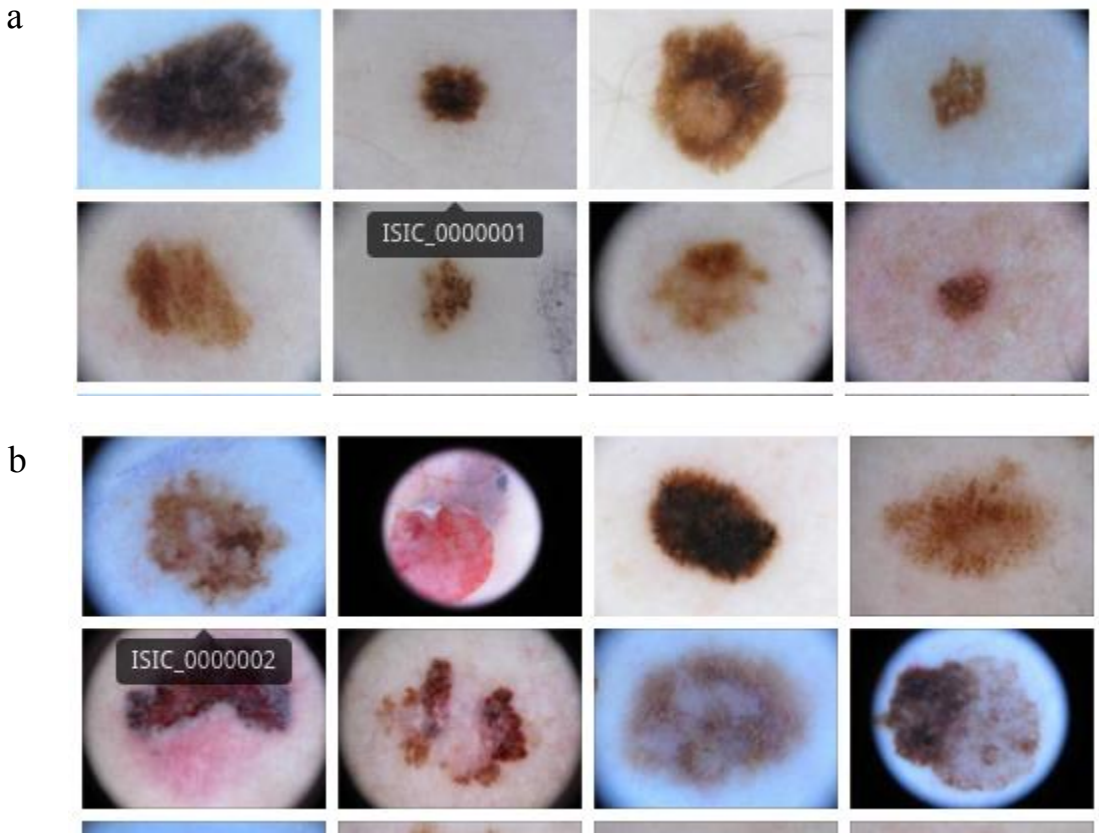


Figure 4: Example dermoscopic images from ISBI 2017 Challenge

a) Benign b) Malignant

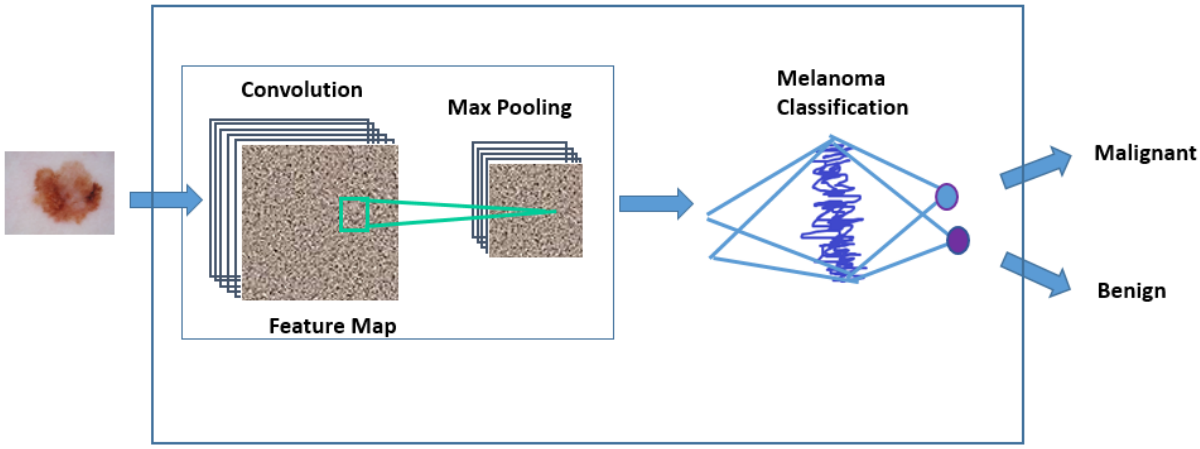
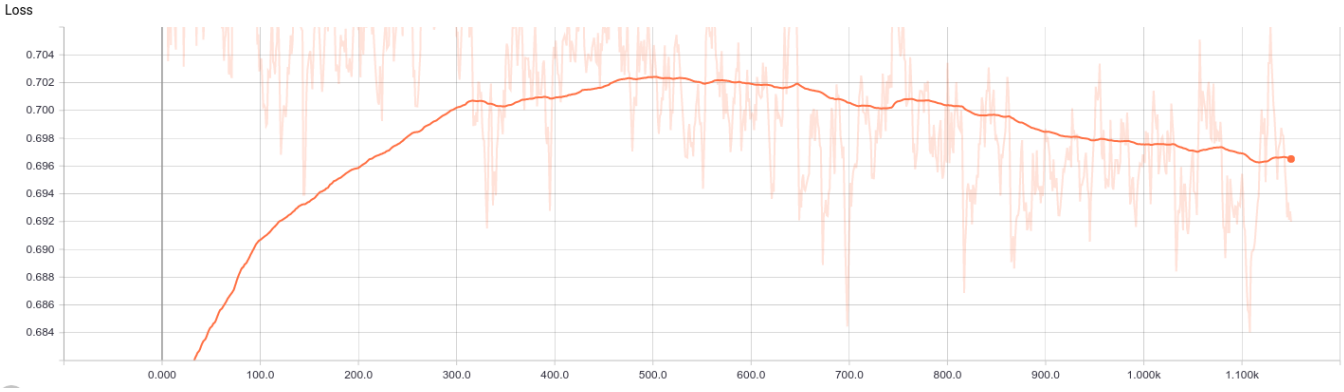
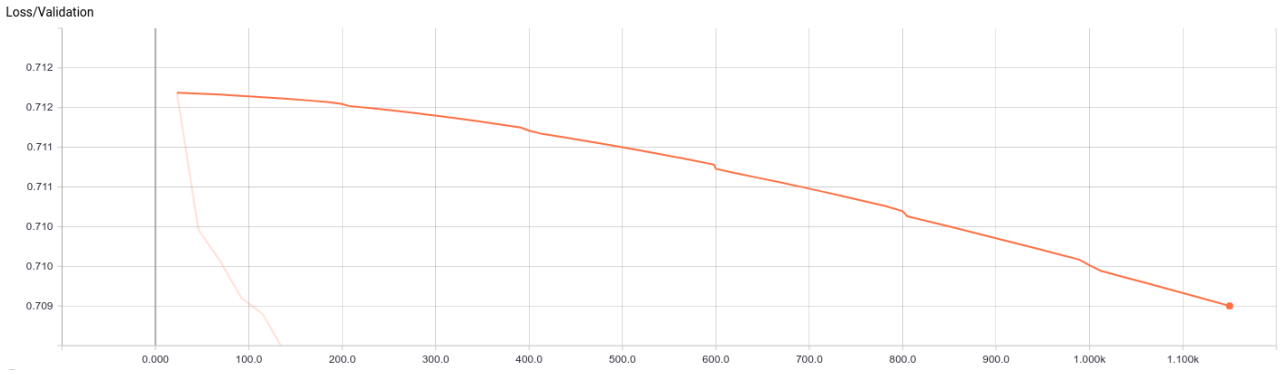


Figure 5: Diagram of a general CNN architecture used for skin lesion analysis

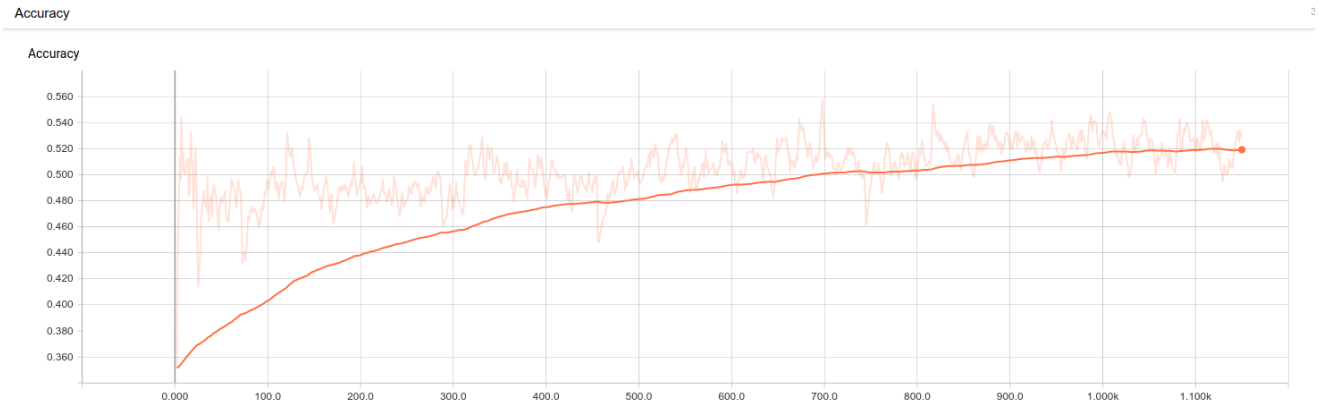
a



b



c



d

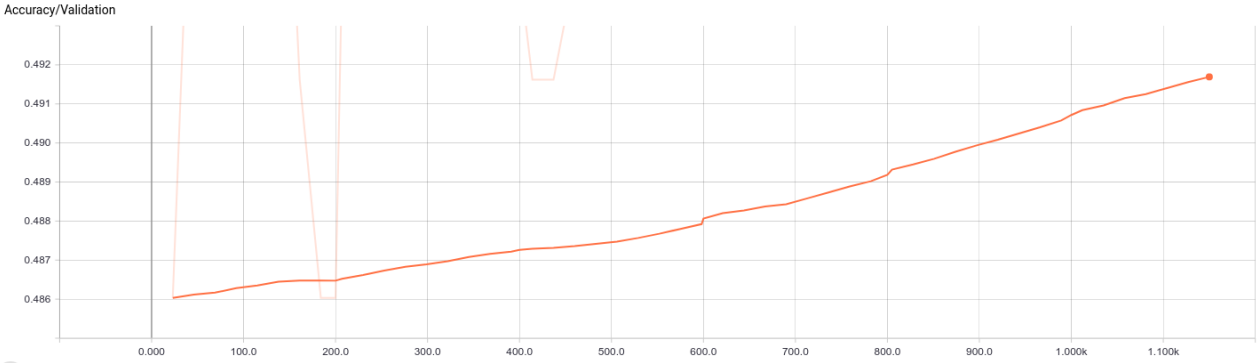
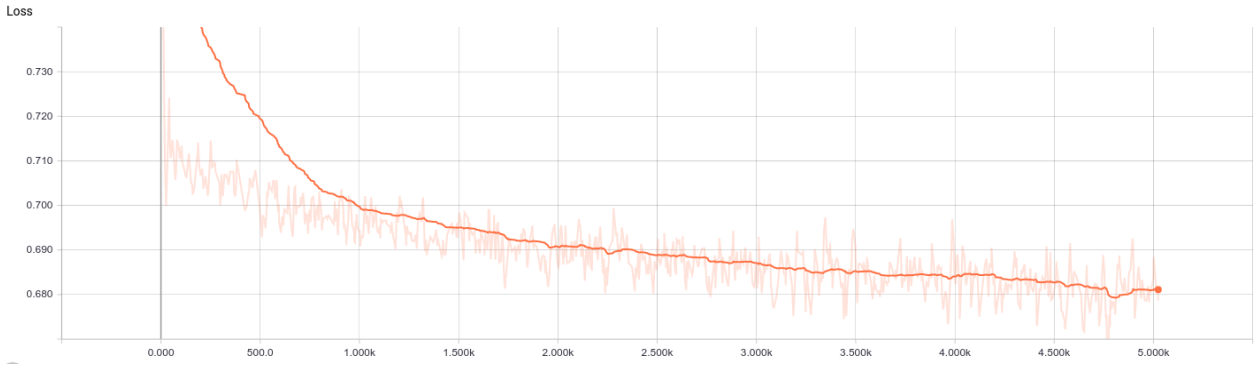


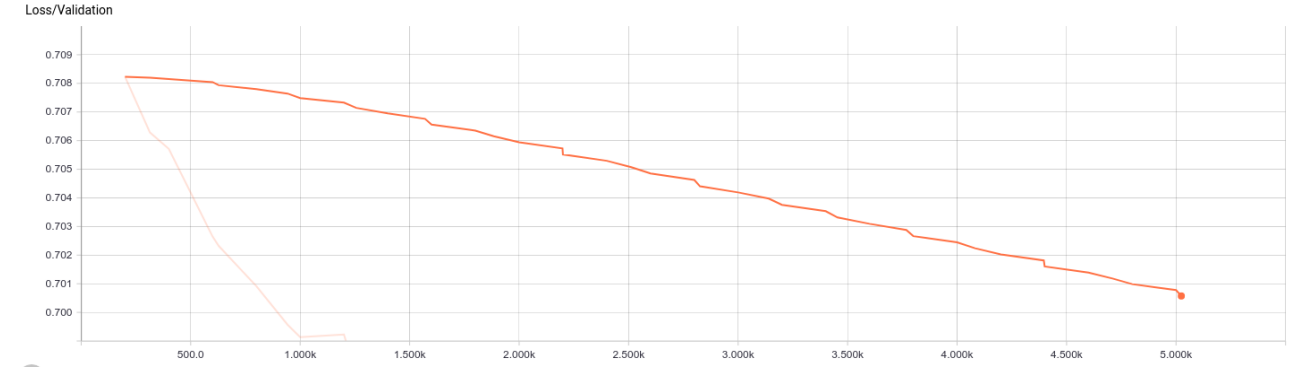
Figure 6: The Performance analysis of CNN architecture 1-d

(a): Total loss (b):Validation loss (c): Training accuracy (d): Validation accuracy

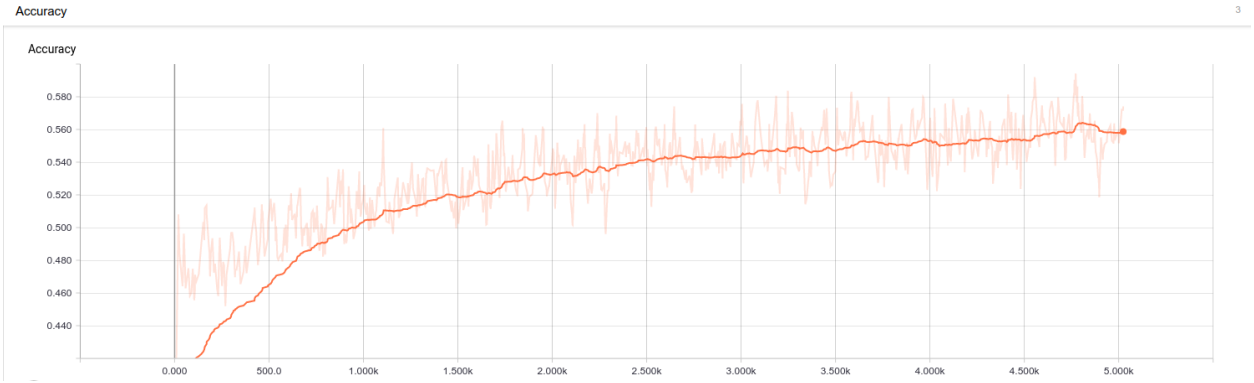
a



b



c



d

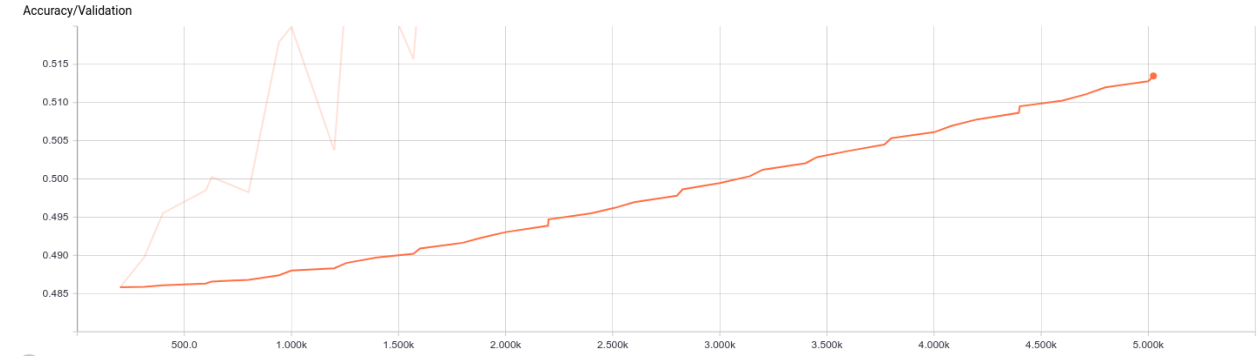
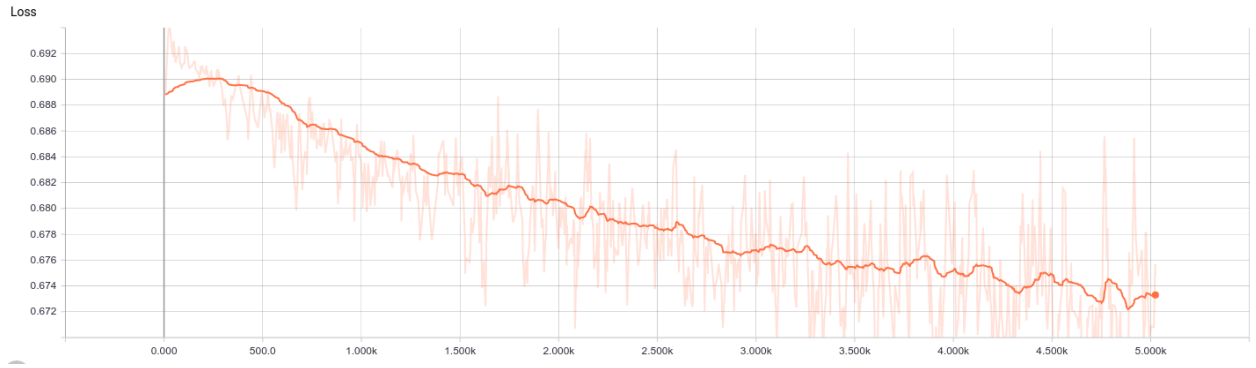


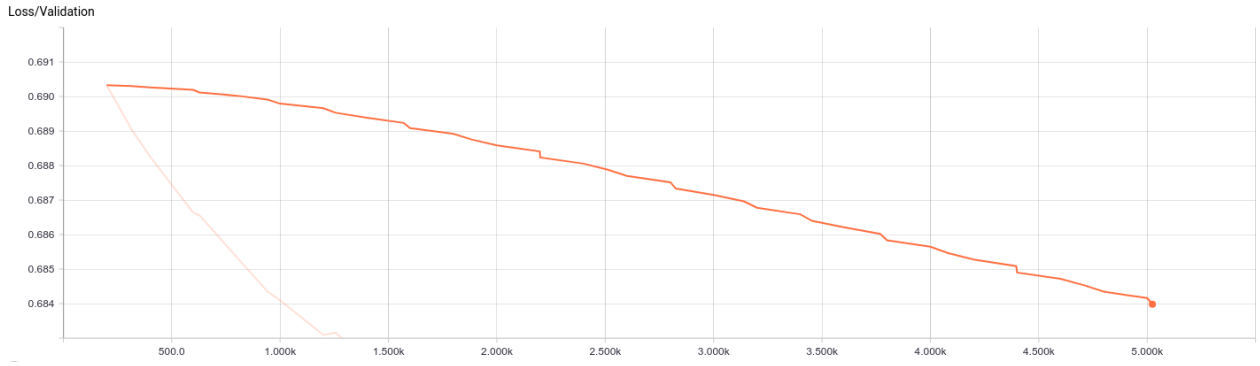
Figure 7: The Performance analysis of CNN architecture 1-k

(a): Total loss (b):Validation loss (c): Training accuracy (d): Validation accuracy

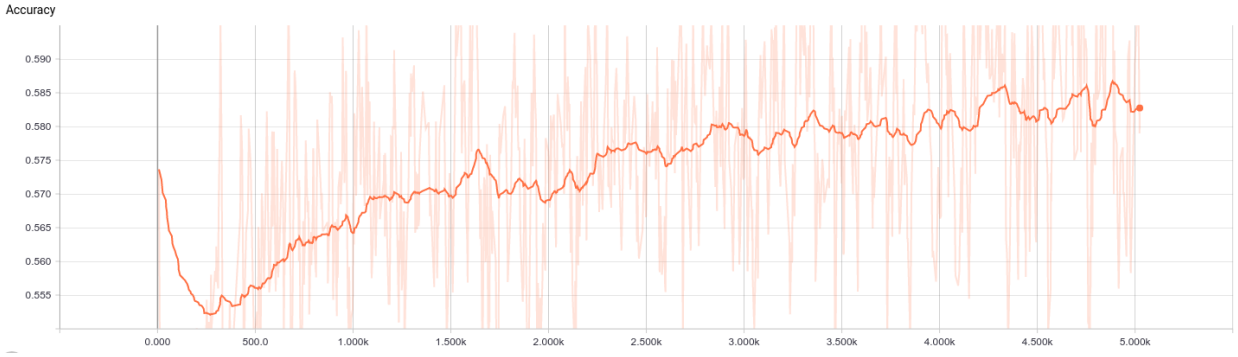
a


Loss
0.692
0.690
0.688
0.686
0.684
0.682
0.680
0.678
0.676
0.674
0.672
0.000
500.0
1.000k
1.500k
2.000k
2.500k
3.000k
3.500k
4.000k
4.500k
5.000k

b


Loss/Validation
0.691
0.690
0.689
0.688
0.687
0.686
0.685
0.684
500.0
1.000k
1.500k
2.000k
2.500k
3.000k
3.500k
4.000k
4.500k
5.000k

c


Accuracy
0.590
0.585
0.580
0.575
0.570
0.565
0.560
0.555
0.000
500.0
1.000k
1.500k
2.000k
2.500k
3.000k
3.500k
4.000k
4.500k
5.000k

d

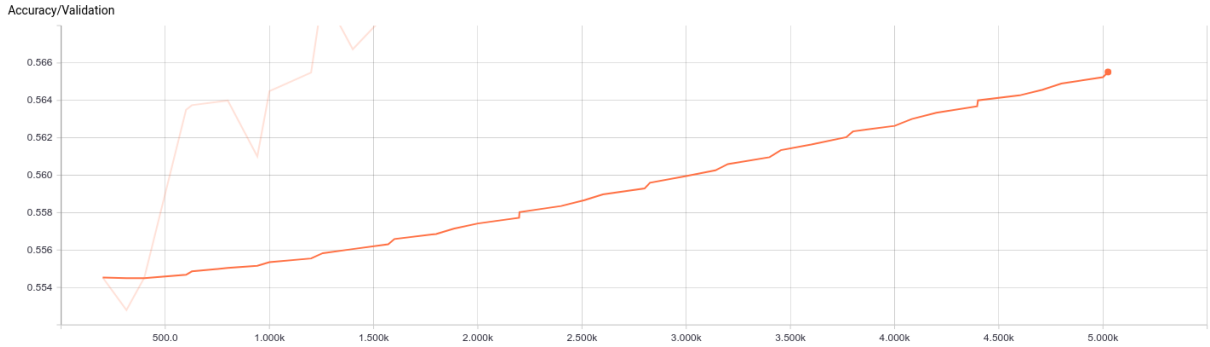
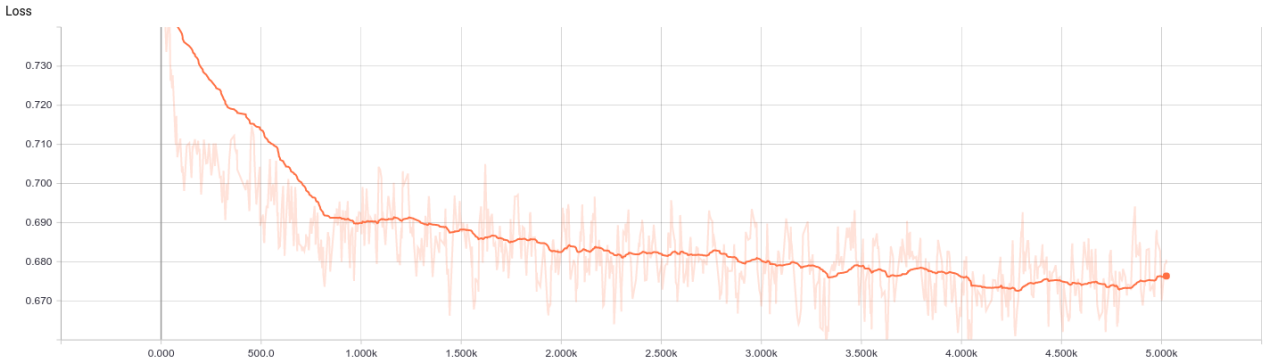


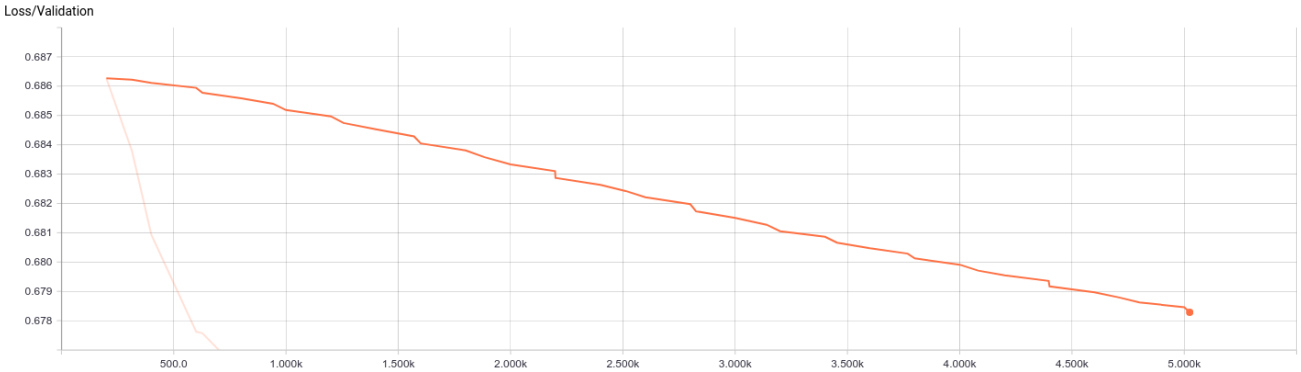
Figure 8: The Performance analysis of CNN architecture 2-c

(a): Total loss (b):Validation loss (c): Training accuracy (d): Validation accuracy

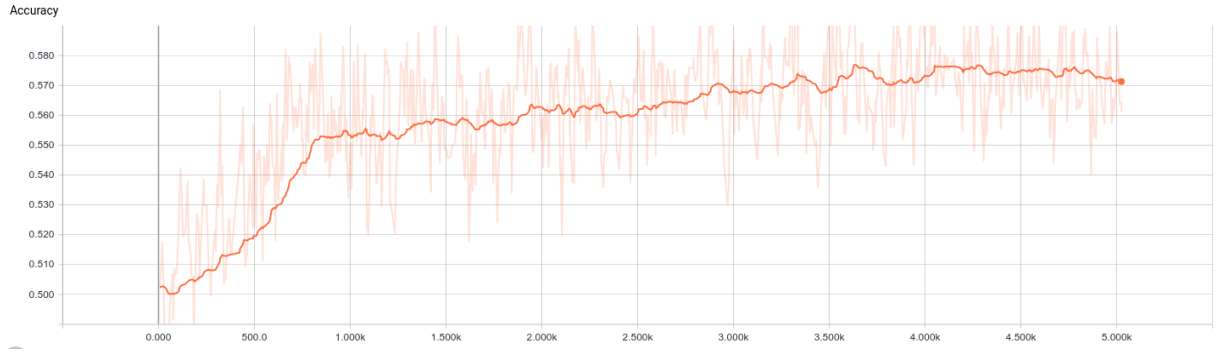
a



b



c



d

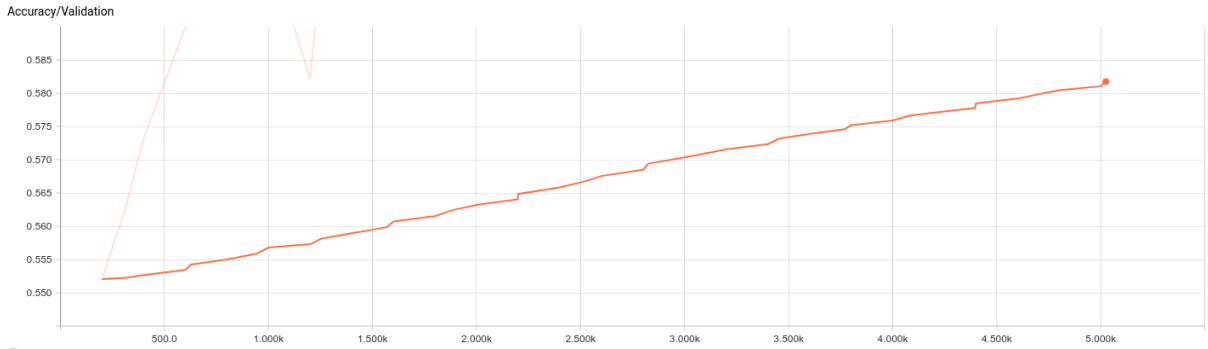
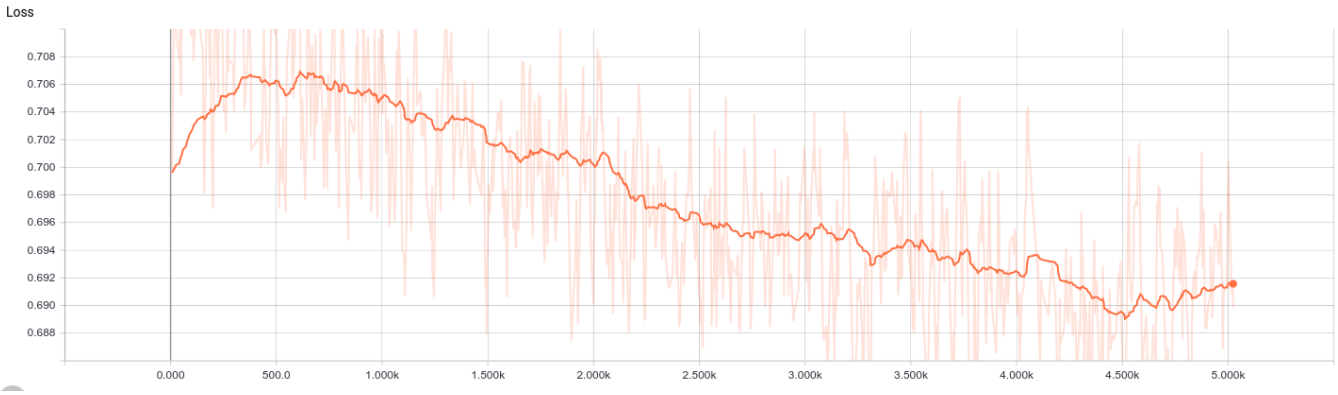


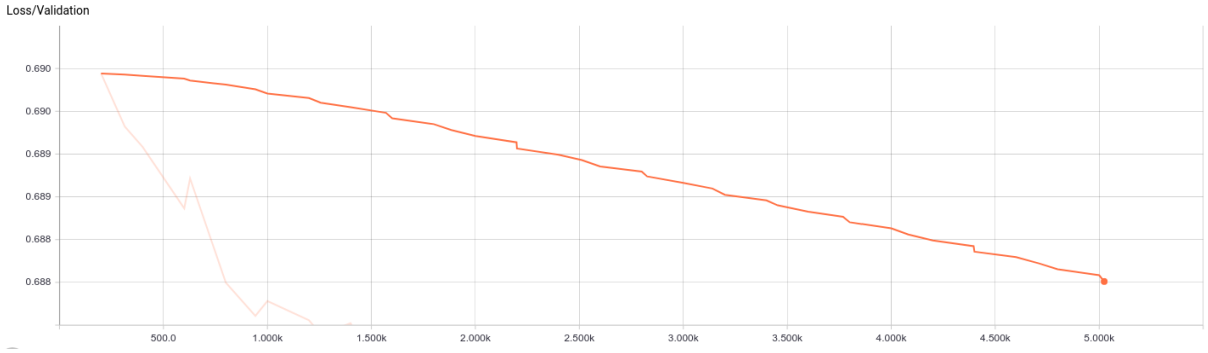
Figure 9: The Performance analysis of CNN architecture 2-h

(a): Total loss (b):Validation loss (c): Training accuracy (d): Validation accuracy

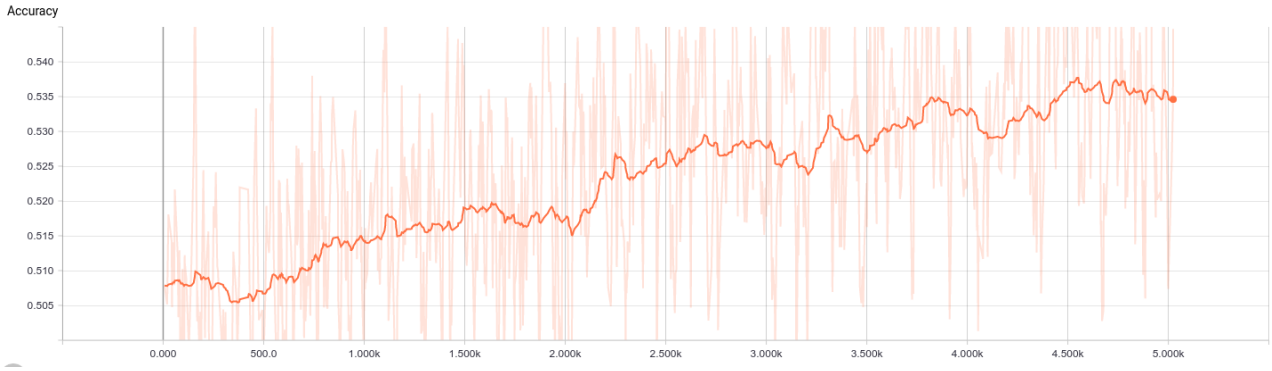
a



b



c



d

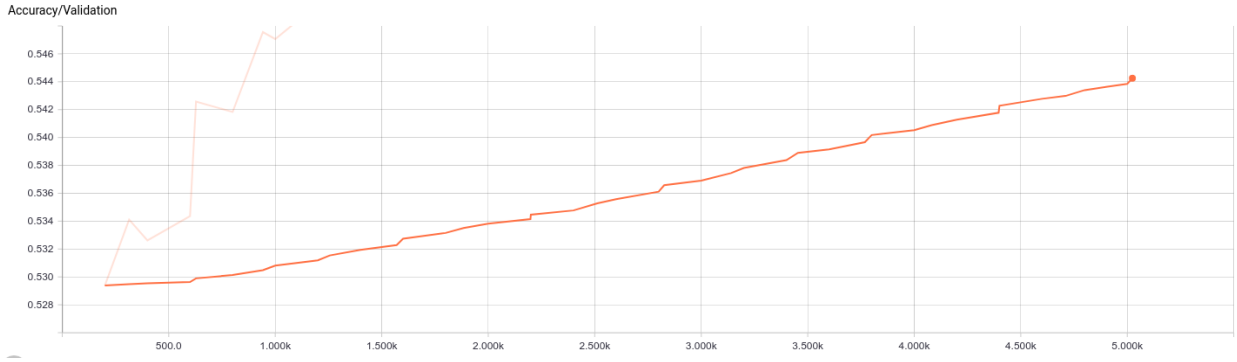
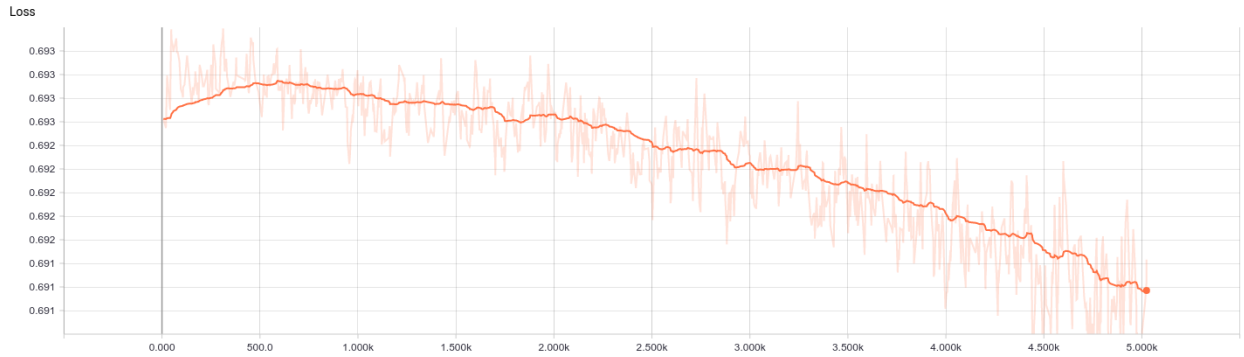


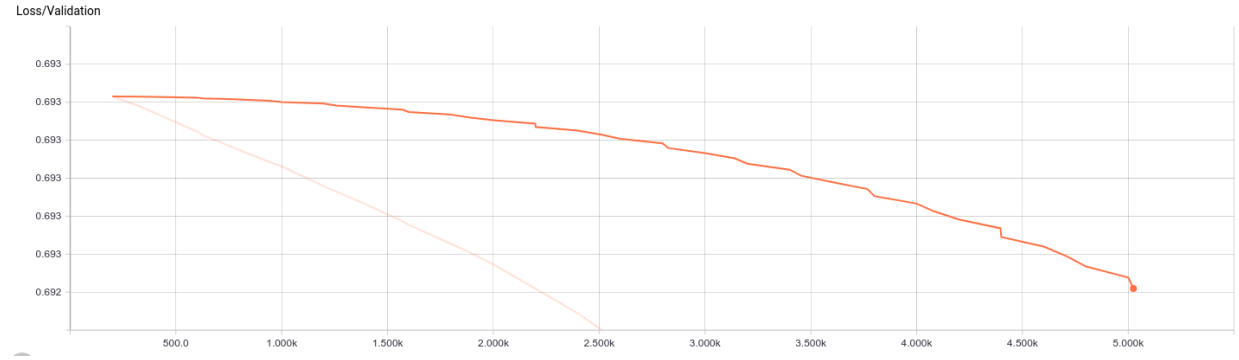
Figure 10: The Performance analysis of CNN architecture 3-a

(a): Total loss (b):Validation loss (c): Training accuracy (d): Validation accuracy

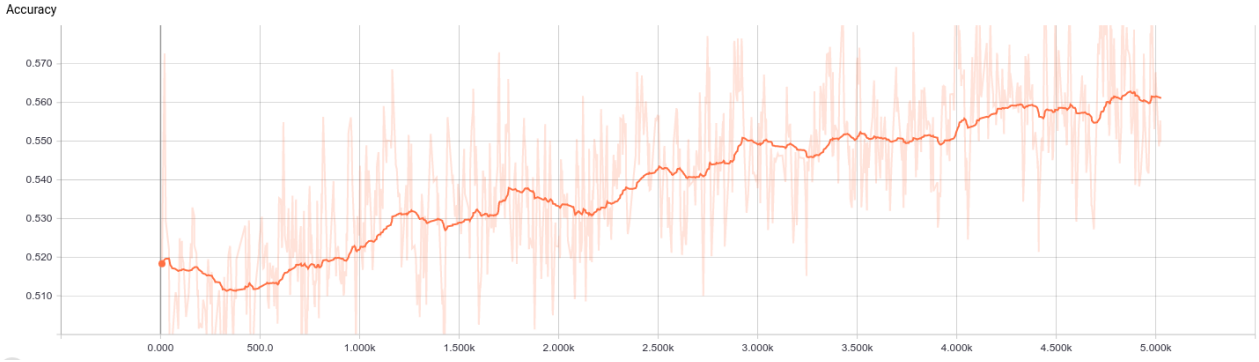
a


Loss
0.693
0.693
0.693
0.693
0.692
0.692
0.692
0.692
0.692
0.691
0.691
0.691
0.000
500.0
1.000k
1.500k
2.000k
2.500k
3.000k
3.500k
4.000k
4.500k
5.000k

b


Loss/Validation
0.693
0.693
0.693
0.693
0.693
0.693
0.692
500.0
1.000k
1.500k
2.000k
2.500k
3.000k
3.500k
4.000k
4.500k
5.000k

c


Accuracy
0.570
0.560
0.550
0.540
0.530
0.520
0.510
0.000
500.0
1.000k
1.500k
2.000k
2.500k
3.000k
3.500k
4.000k
4.500k
5.000k

d

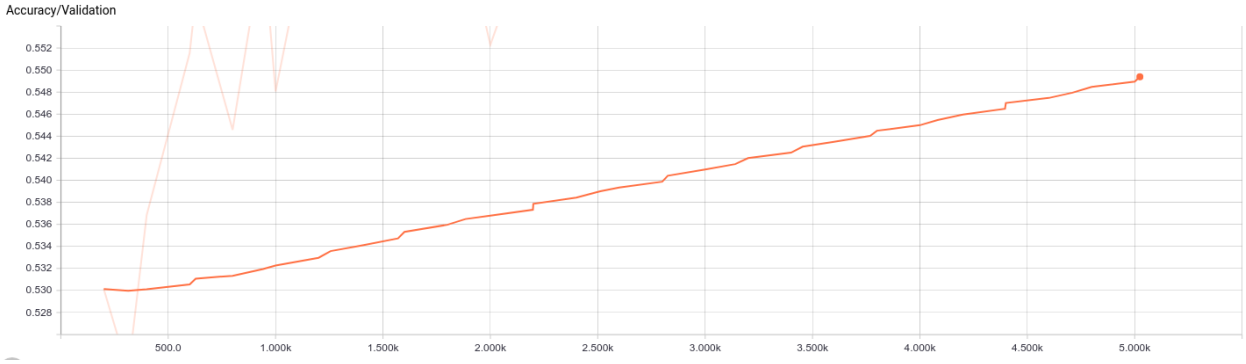
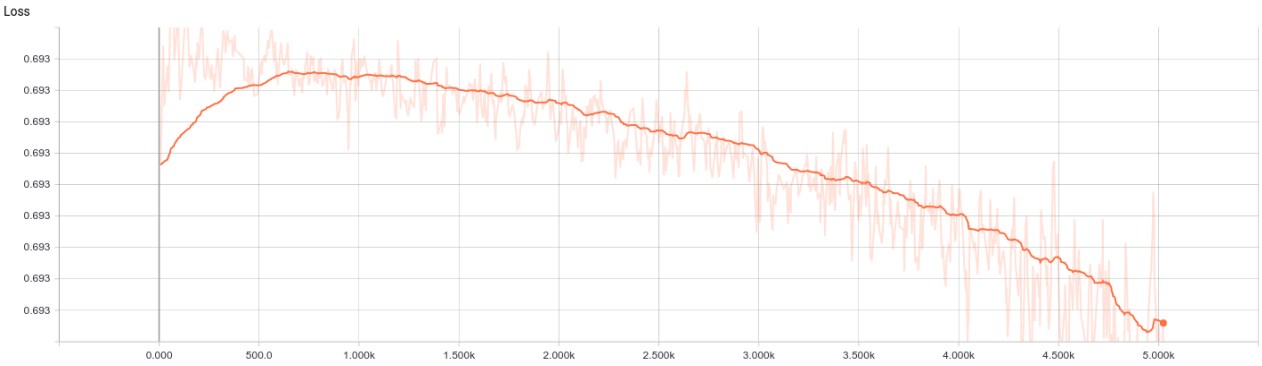


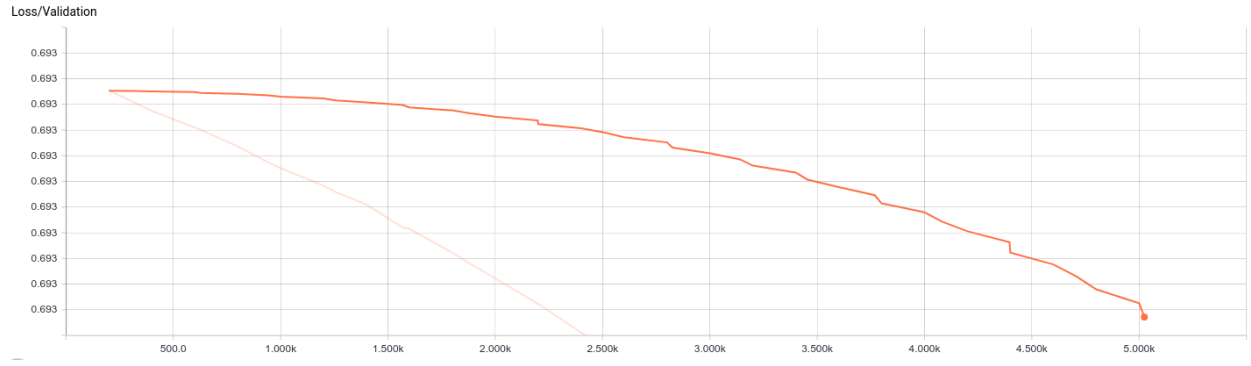
Figure 11: The Performance analysis of CNN architecture 3-b

(a): Total loss (b):Validation loss (c): Training accuracy (d): Validation accuracy

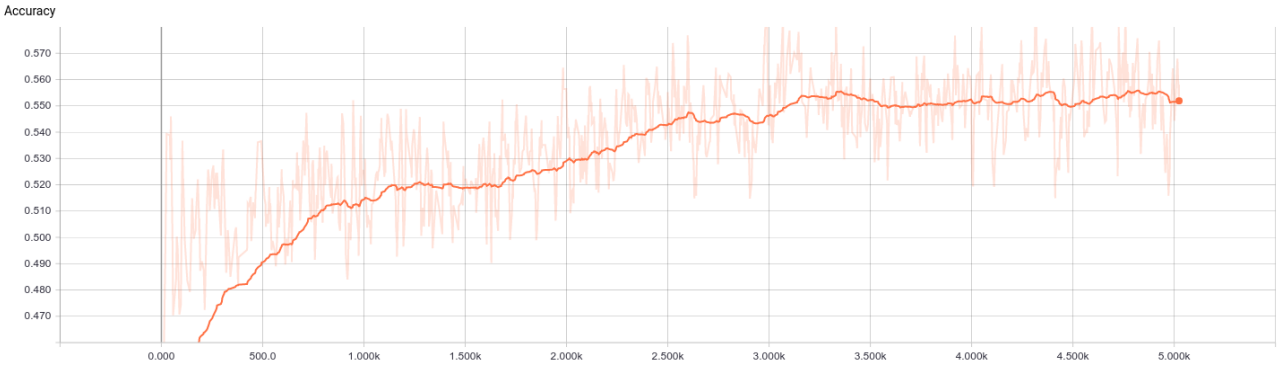
a



b



c



d

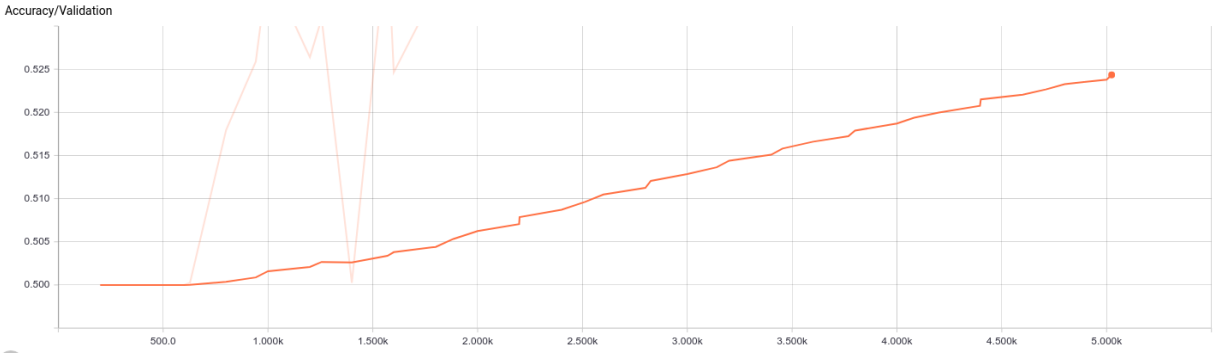
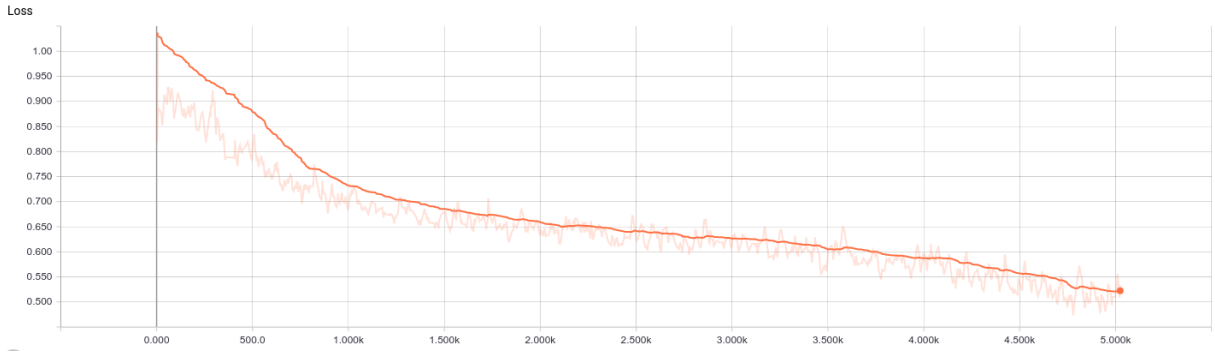


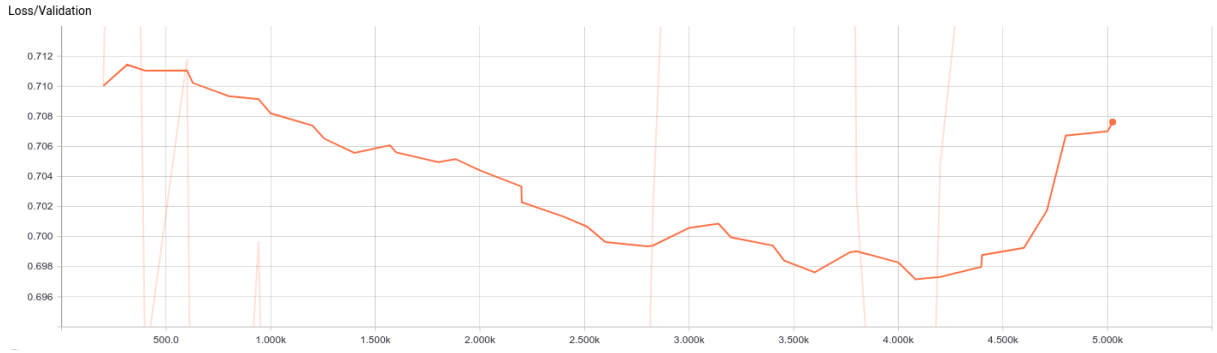
Figure 12: The Performance analysis of CNN architecture 4-b

(a): Total loss (b):Validation loss (c): Training accuracy (d): Validation accuracy

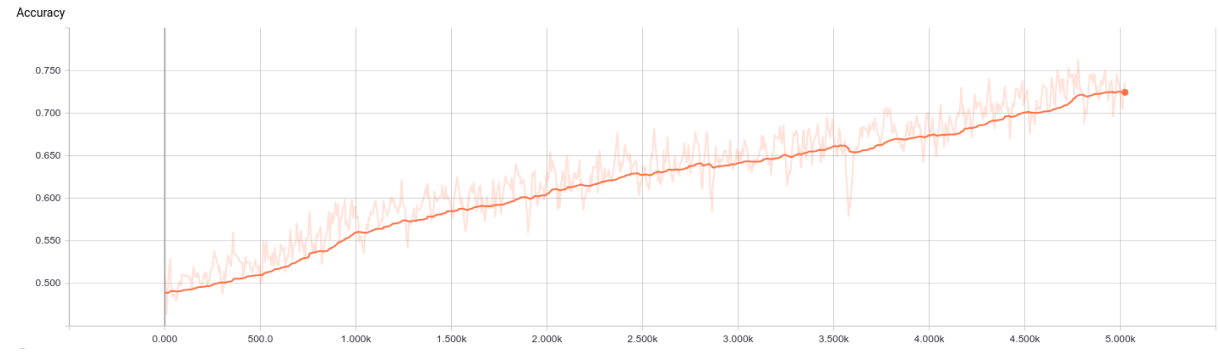
a



b



c



d

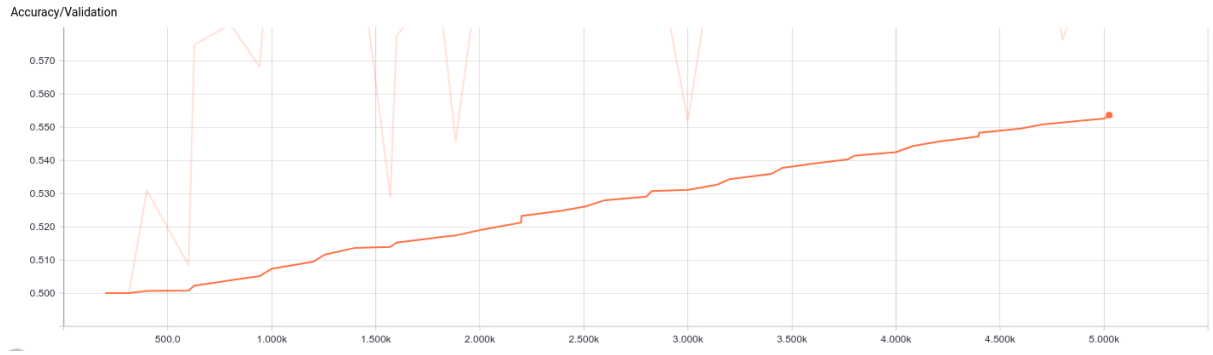
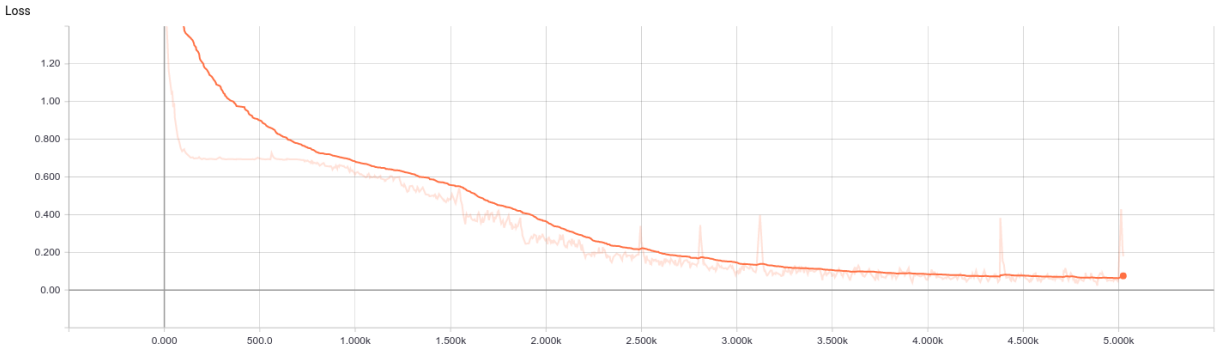


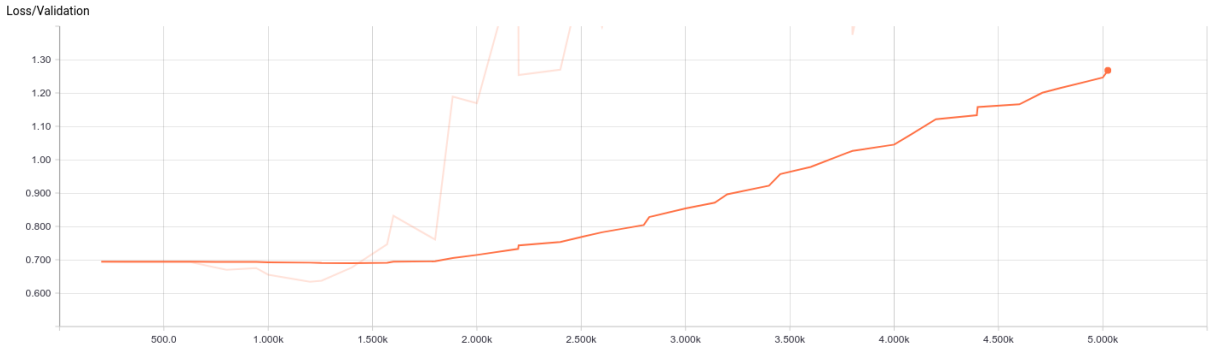
Figure 13: The Performance analysis of AlexNet

(a): Total loss (b):Validation loss (c): Training accuracy (d): Validation accuracy

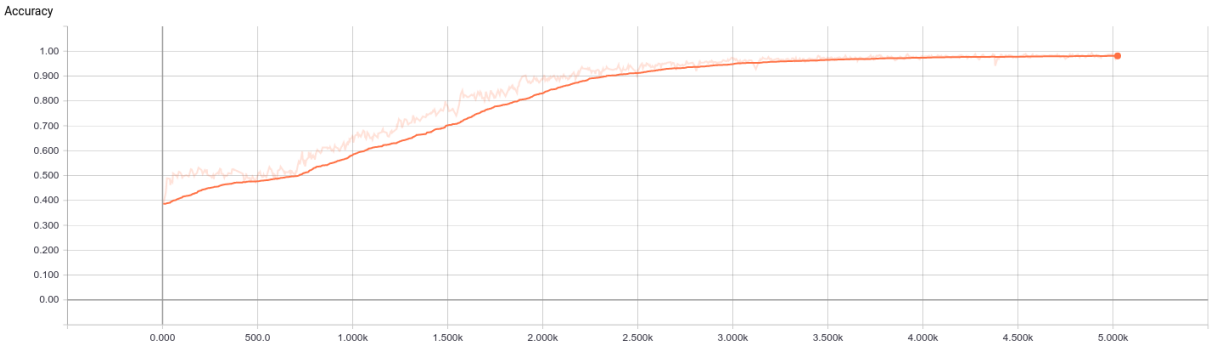
a


Loss
1.20
1.00
0.800
0.600
0.400
0.200
0.00
0.000
500.0
1.000k
1.500k
2.000k
2.500k
3.000k
3.500k
4.000k
4.500k
5.000k

b


Loss/Validation
1.30
1.20
1.10
1.00
0.900
0.800
0.700
0.600
500.0
1.000k
1.500k
2.000k
2.500k
3.000k
3.500k
4.000k
4.500k
5.000k

c


Accuracy
1.00
0.900
0.800
0.700
0.600
0.500
0.400
0.300
0.200
0.100
0.00
0.000
500.0
1.000k
1.500k
2.000k
2.500k
3.000k
3.500k
4.000k
4.500k
5.000k

d

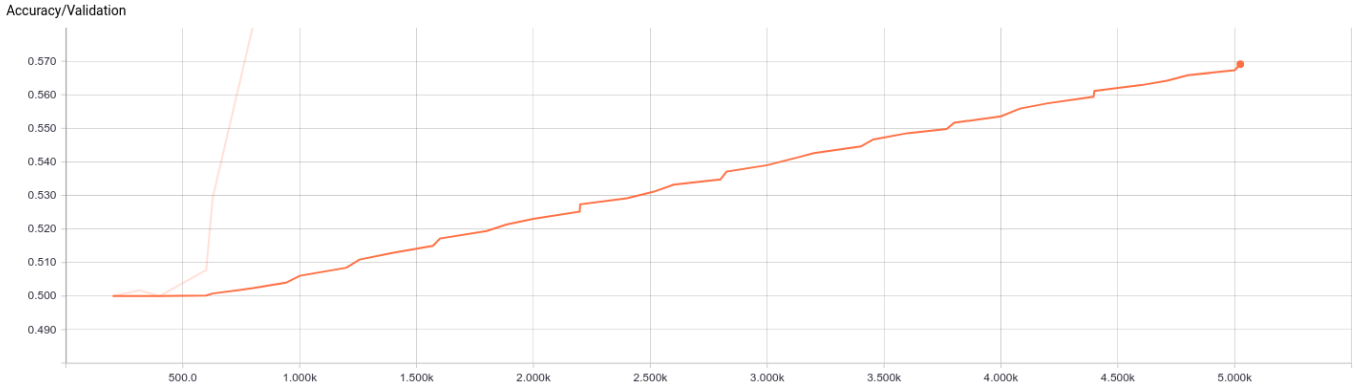
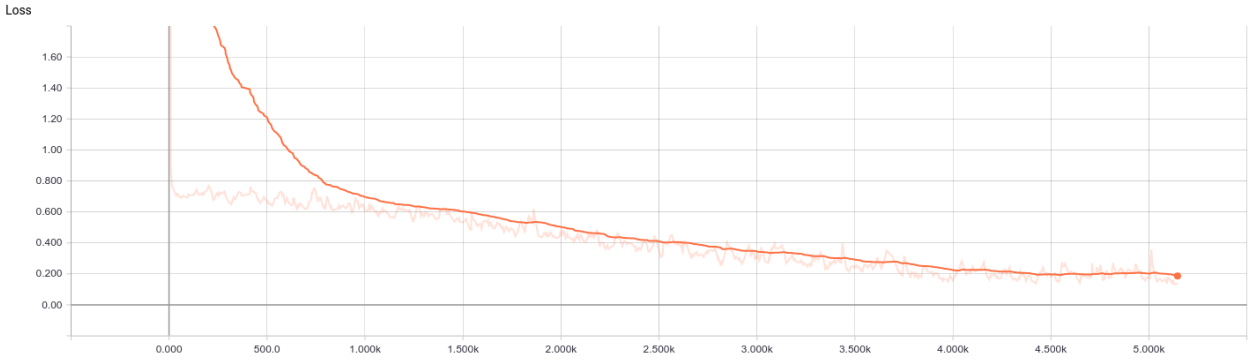


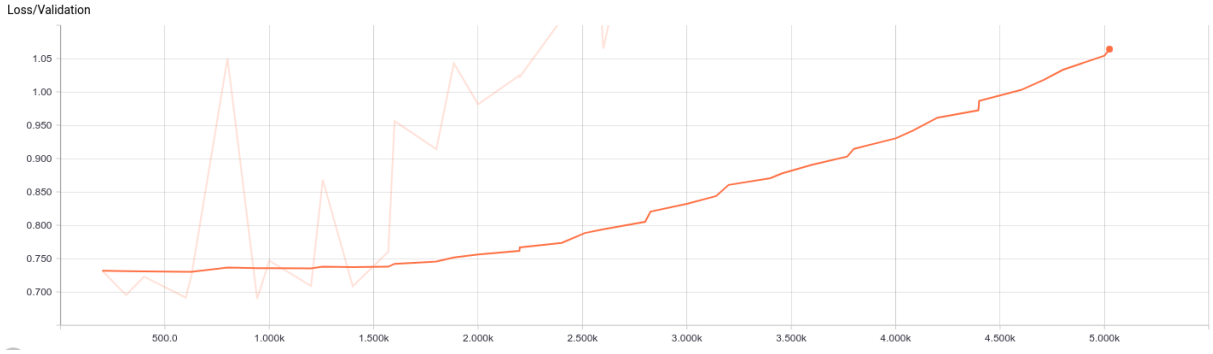
Figure 14: The Performance analysis of VGGNet

(a): Total loss (b):Validation loss (c): Training accuracy (d): Validation accuracy

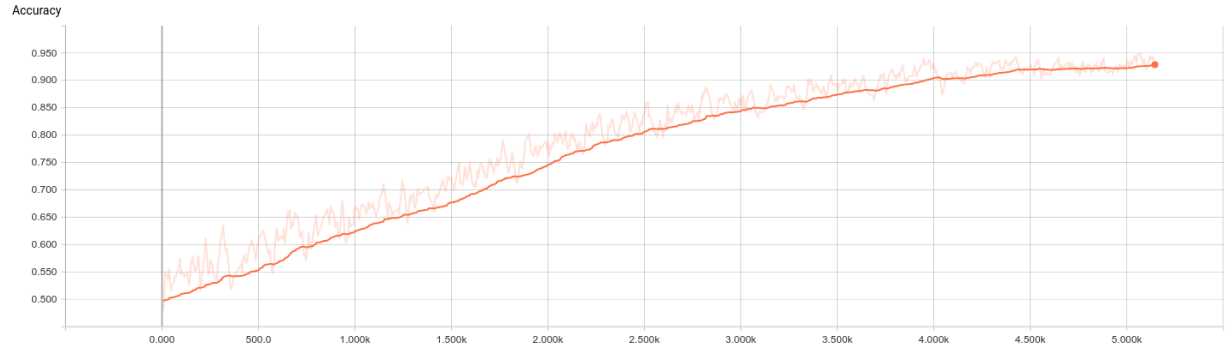
a



b



c



d

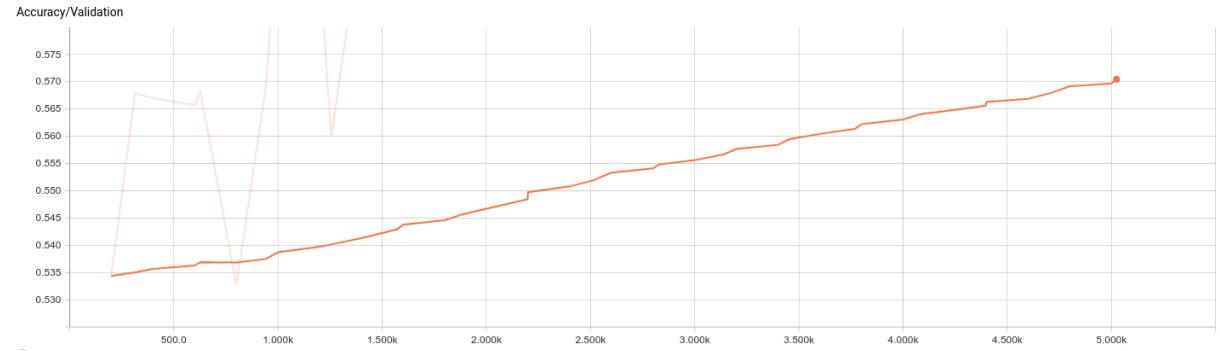
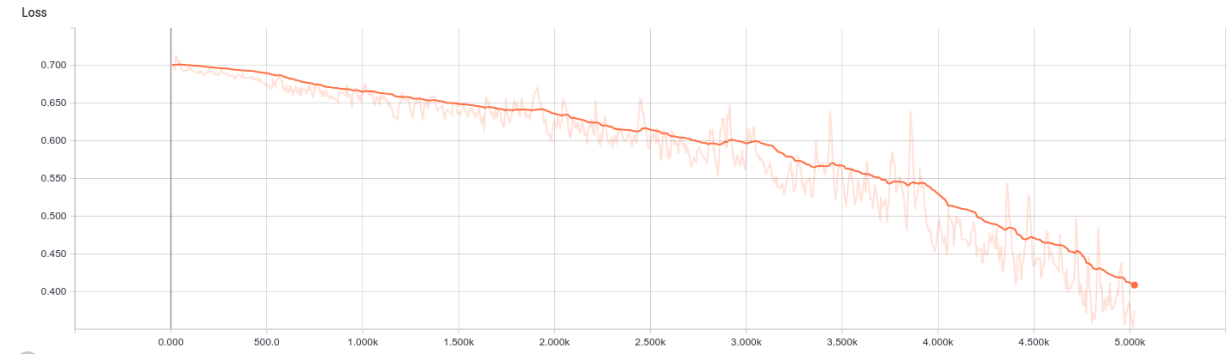


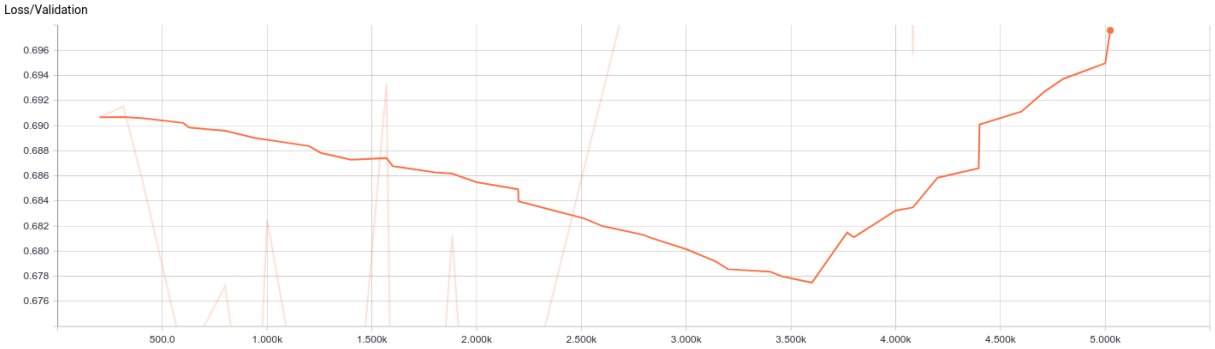
Figure 15: The Performance analysis of Highway CNN

(a): Total loss (b):Validation loss (c): Training accuracy (d): Validation accuracy

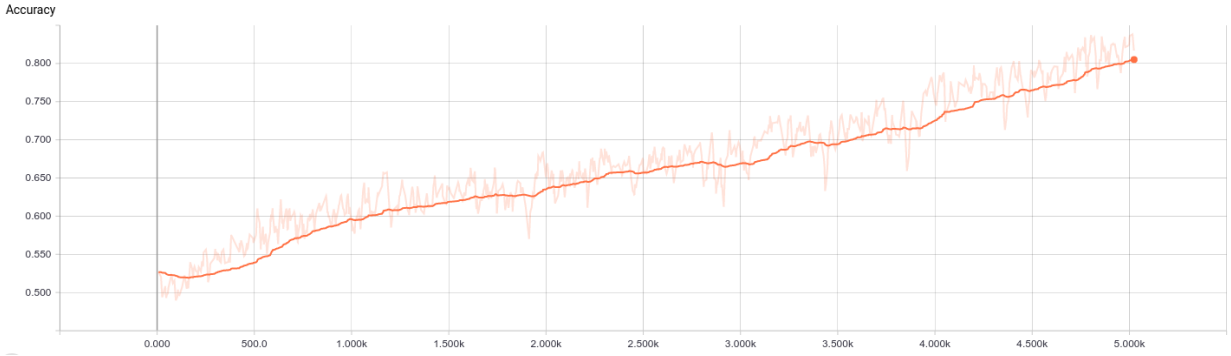
a



b



c



d

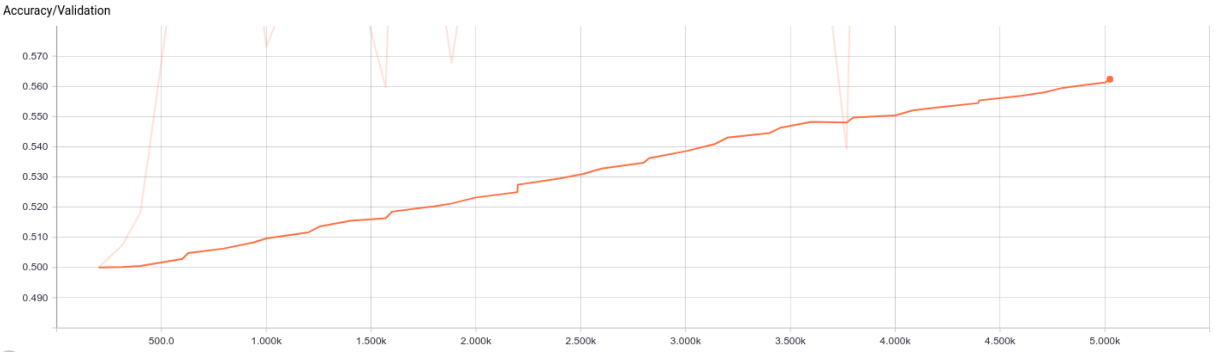
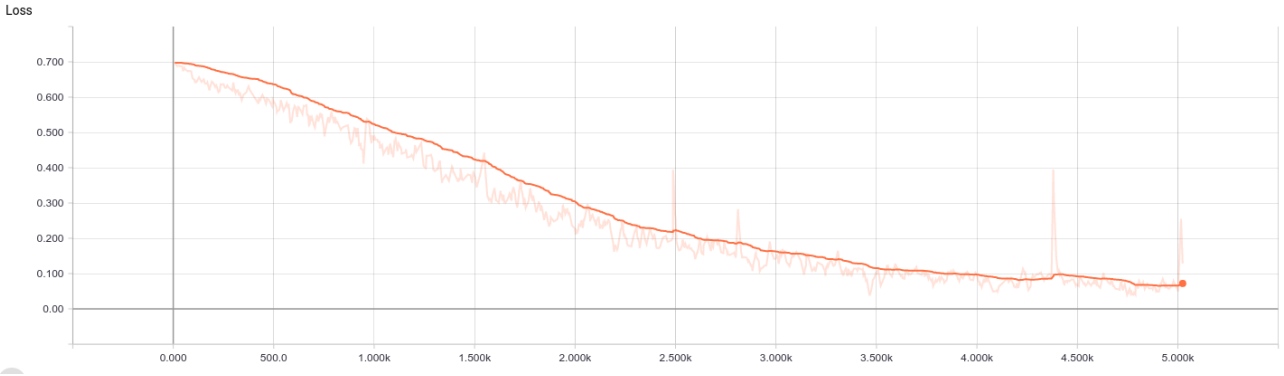


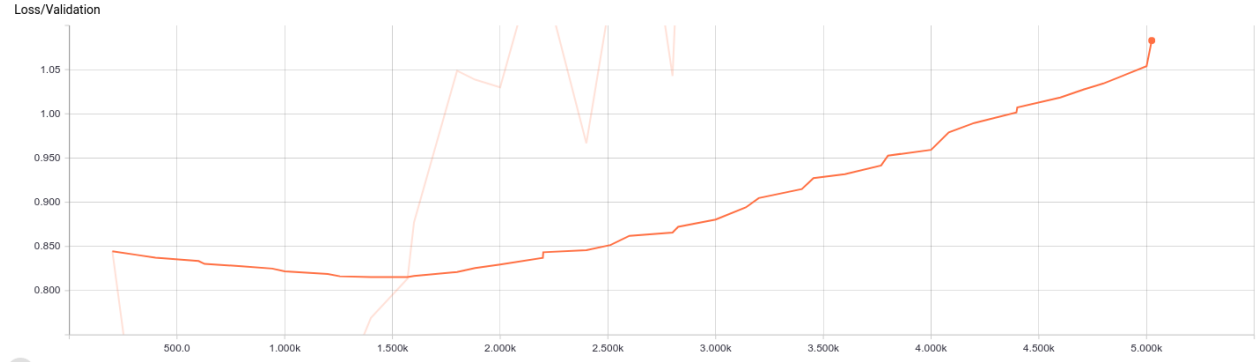
Figure 16: The Performance analysis of Google Inception V3 model

(a): Total loss (b):Validation loss (c): Training accuracy (d): Validation accuracy

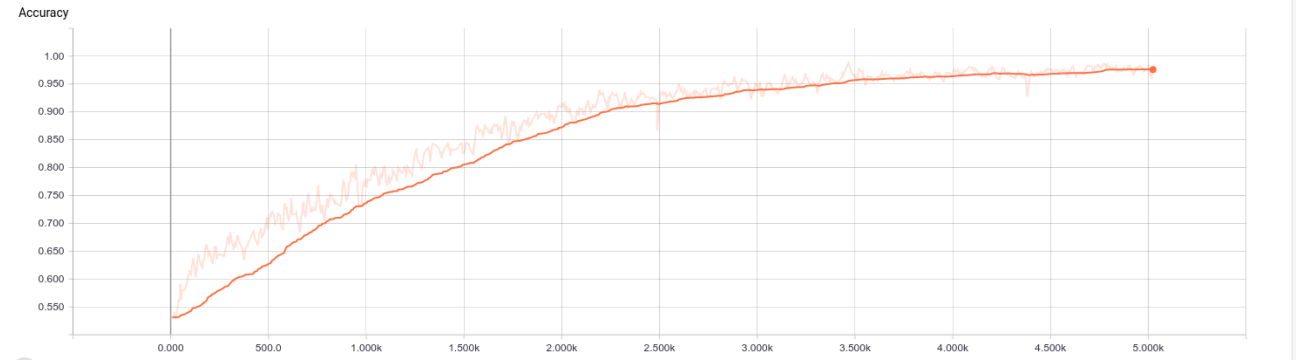
a


Loss
0.700
0.600
0.500
0.400
0.300
0.200
0.100
0.00
0.000
500.0
1.000k
1.500k
2.000k
2.500k
3.000k
3.500k
4.000k
4.500k
5.000k

b


Loss/Validation
1.05
1.00
0.950
0.900
0.850
0.800
500.0
1.000k
1.500k
2.000k
2.500k
3.000k
3.500k
4.000k
4.500k
5.000k

c


Accuracy
1.00
0.950
0.900
0.850
0.800
0.750
0.700
0.650
0.600
0.550
0.000
500.0
1.000k
1.500k
2.000k
2.500k
3.000k
3.500k
4.000k
4.500k
5.000k

d

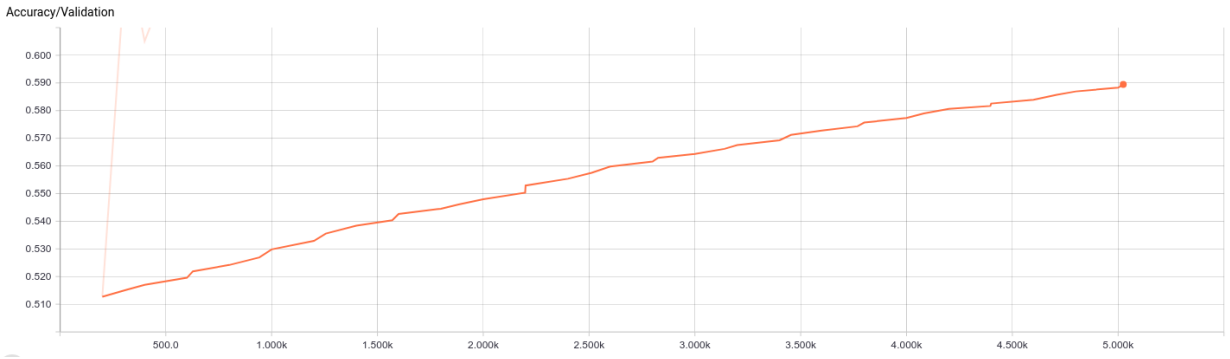


Figure 17: The Performance analysis of proposed Highway convolutional neural network (a): Total loss (b):Validation loss (c): Training accuracy (d): Validation accuracy