Skin Cancer Classification

Ningwei Li
A12373581

Arshia Zafari
A11167578

Zhichen Zhang
A99112494

Jennifer Segura
A11668630

Abstract

More people are diagnosed with skin cancer each year in the U.S. than all other cancers combined. With the development of Neural Networks, it is possible to employ machine learning techniques to process and analyze images of skin lesions to effectively detect skin diseases. In this work we will train and implement a VGG-13 Convolutional Neural Network to extract features from a set of 9,758 pigmented skin lesions and correctly classify them as one of five diseases of benign or malignant cancer. The results need to be studied using various analytic metrics across iterations to determine the performance of the classifier. Creating a both robust and highly accurate architecture will give medical professionals and patients the tools to increase the survival rates for skin cancer.

1 Introduction

Every year there are about 5.4 million new cases of skin cancer in the United States, and while the five-year survival rate for melanoma detected in its earliest states is around 97%, that drops to approximately 14% if it's detected in its latest stages [5]. This makes early detection critical to successful treatment of skin cancer, and it requires both a vigilant, responsible patient as well as a qualified medical professional.

The diagnostic process for skin cancer is primarily visual, meaning a doctor performs a visual assessment using a dermatoscope before ordering any tests. What makes the process challenging is the similarities between lesion types, and this makes the diagnostic accuracy heavily reliant on the professional experience of the physician. Without additional technical support, dermatologists have a 65% to 80% accuracy rate in melanoma diagnosis [7]. Introducing other dermatoscopic tools like a skin image classifier will ultimately improve upon this accuracy.

A dermatologist uses a dermatoscope, a type of handheld microscope, to look at skin [5].

The idea is to supplement the examination process with a computer that is trained using the mechanisms of deep learning to assess images of skin lesions in a similar fashion as a physician and predict whether they are classified as benign (non-cancerous) or malignant (cancerous/potentially cancerous). For the context of this problem, we will look at five different classes of the most prevalent benign and malignant diseases: *Seborrheic keratosis* and *Melanocytic nevi* for benign, and *Actinic keratoses*, *Basal cell carcinoma*, and *Melanoma* for melignant.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

2 Method

2.1 Algorithm

Using Convolutional Neural Networks (CNNs) in the context of image classification involves a general pipeline outlined in **Figure 1**. There are many different architectures for a CNN and this report will be discussing the implementation of a variation on the VGG using only 13 layers called the VGG13, more on that later.

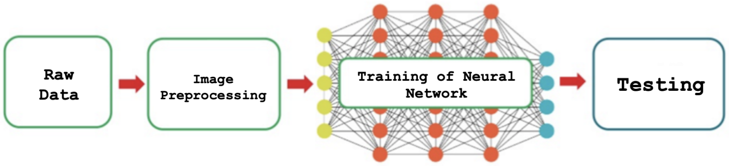


Figure 1: Algorithm for Image Classification

For the context of skin classification, we first start with a set of input data representative enough to be split into subsets for training and testing purposes. Before the network can be trained, the input data must undergo various forms of pre-processing. This is the most important part of the algorithm as we need to feed the network a consistent and high quality set of information in order to extract the most relevant features. Classification of a series of images of skin lesions requires various kinds of feature extraction, as lesions vary in color, shape, size, texture, and definition. This means our images must be in color and strategically reduced to a size that includes the boundaries of the skin lesions to factor in the shape and the original aspect ratio of the image. The images are first cropped, then resized to 50x50 for this purpose.

Once the images have been modified, they must be converted to a large NumPy array with the five corresponding labels to be fed into the network; this is done using Python loader scripts. With this, the network is then trained to be able to take the input data and extract the different features mentioned earlier without manual feature engineering. Learning these features will allow the network to classify the input data as one class or another, analyzing the accuracy along the way. Finally, with the trained network we can test it using our testing set to measure the network's performance.

2.2 Network Architecture

The network architecture used for this problem is a Very Deep Convolutional Neural Network (VGG) using a Keras API framework. Introduced by Simonyan and Zisserman in the ILSVRC 2014 competition, VGGNet is very appealing because of its uniform and simple architecture. The original architecture consisted of 13 convolution layers using a 3x3 convolution window, 5 of 2x2 max pooling layers, and 3 fully connected layers [3]. A modified version is implemented here for the skin cancer detection problem, the VGG13. This architecture retains the 5 of 2x2 max pooling layers of the original but reduces the number of convolution layers to 10, so 5 blocks of 2 convolutions and 1 max pool. This is followed up by a standard 3 fully-connected layers consisting of two ReLu activation functions and a softmax classifier. The full network with annotated layer sizes is shown in **Figure 2**.

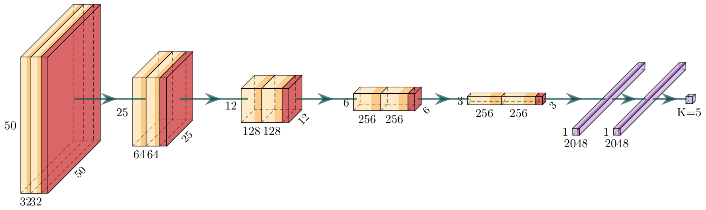


Figure 2: VGG13 Network [4]

Table 1: Network Parameters

Parameter	Value
Learning Rate	0.1
Epochs	30
Batch Size	32
Loss function	Categorical cross-entropy
Optimizer	AdaDelta

2.3 Experimental Setting

The dataset that will be used to train the network is called the HAM10000, which comes from a larger archive, the International Skin Imaging Collaboration (ISIC), containing thousands of labeled images of different skin diseases along with patient metadata [1]. As a subset of the archive, the "Human Against Machine 10000" in **Figure 3** is a collection of 10,0015 dermatoscopic images from different populations, acquired and stored by different modalities [6].

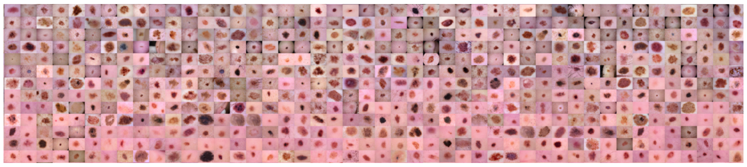


Figure 3: HAM10000 Dataset

The set includes a representative collection of all important diagnostic categories in the realm of pigmented lesions, What makes it a valuable training set for academic machine learning purposes is its consistency in the field of view of the images. Each image as a high quality close up of the lesion of interest, and as mentioned before the network needs a uniform set of input images to be properly trained.

From the 10,0015 images only 9,758 were chosen to maximize representation for each class.The training and testing set was split into 80% and 20% accordingly, and the validation set was 20%of the training set. The network parameters, outlined in **Table 1**, were optimized after various attempts of trial and error, paying close attention to both training and validation accuracy. The network is run on Intel I5-6500 processor, with 16GB RAM, GTX 1060 graphics card, using Python 3.6 (64-bit). The framework is built with Keras, which is a higher level interface providing both an abstract but simple data processing techniques utilizing TensorFlow back-end functions.

3 Results

The accuracy per epoch and cross-entropy loss per epoch from the training using the specified parameters and the 50x50 images are given in **Figure 4**.

The standard and top-2 accuracy percentages for the training and validation sets all seem to increase through iterations of the training. By the end of 30 epochs, the training accuracy (green) ended up at 96.38% and the validation accuracy (orange) at 69.33%. Running on the testing set we get an overall accuracy of 68.9% shown in **Table 2** which is not bad in the context of the problem, but it is not great. Notice how the validation accuracy does not increase with training beyond 20 epochs. The loss reflects this behavior, as it increases beyond 20 epochs as well due to over-fitting.

Table 2: Test Results

Image size	Accuracy	Top-2 Accuracy
50x50	68.9%	85.4%
100x100	67.1%	82.6%

We tried many different techniques to improve our network and avoid this over-fitting, including

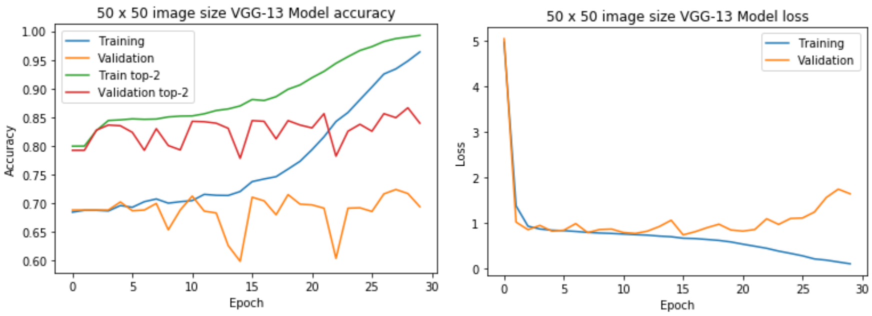


Figure 4: Accuracy and Cross-entropy loss (50x50)

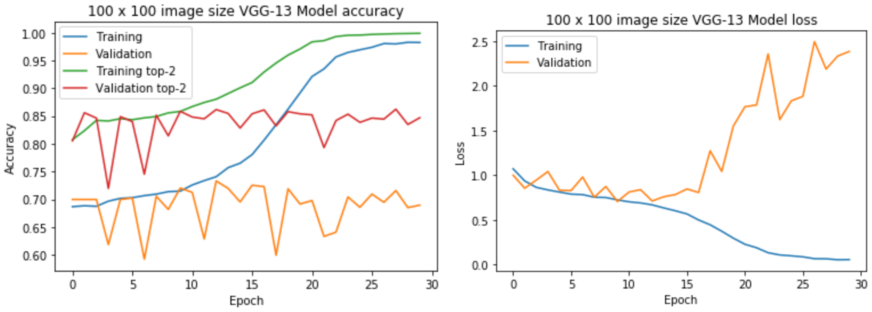


Figure 5: Accuracy and Cross-entropy loss (100x100)

dropout, factoring in class weights given the representation, and batch normalization. We also tried changing the number of epochs as well as the batch size but there was no real improvement in the network. The next approach was to try to improve the actual quality of the images being sent into the network in case we were losing important information. For this, the datasets were re-processed from the raw images for a larger resolution of 100x100. The results are shown in **Figure 5**.

The accuracy on the training set and the top-2 ranking again seems to increase through iterations of the training but actually ends up slightly worse than before. By the end of 30 epochs, the training accuracy ended up at 98.25% which is an improvement but the validation accuracy ends up at 68.95%. The loss also seems to increase starting at around 15 epochs. Running on the testing set we get an overall accuracy of 67.1% shown in **Table 2**. Why is this the case?

The improvement on the training accuracy is due to the fact that the network is doing a good job at changing the parameters to fit the training images, but at the same time it is not very representative of the validation set, hence the over-fitting. The increase in input image size may cause irrelevant information to fool the network, thereby giving us a worse performance. However, the performance from 50 and 100 can be described as statistically similar given the graphs due to the probabilistic nature of network initialization, data shuffling and dropout layers. Applying different over-fitting reduction techniques is not guaranteed to improve the network in the same way when the initialization are randomized, so when we saw the reduced image size gave a slightly better performance on the testing data, that could be just due to the randomness.

4 Discussion

Although the network does not perform perfectly, the results give us very important insight. It shows how intricate the diagnosis of skin diseases actually is. The visual features of different skin diseases can vary widely at different stages of the disease, and there can also be great similarities in appearance across these stages. So the inaccuracies mainly stem from the data set in which we are training our network, images of skin are too similar in general, see **Figure 6**. There seems to be an insufficient amount of visual data to identify multiple stages of skin cancers and differentiate them into different diagnoses.

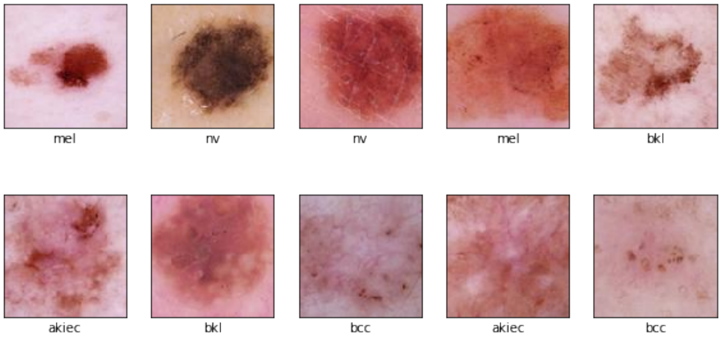


Figure 6: Actinic keratoses (akl), Seborrheic keratosis (bkl), Melanocytic nevi (nv), Basal cell carcinoma (bcc), Melanoma (mel)

Visual diagnosis itself requires a very high resolution microscopic image to be able to see all the valuable information. Physicians use high intensity magnification because the colors and textures of the lesion matter. Reducing our images to 50x50 from a dataset with images originally around 1000x1000 only retains around 0.25% of the information, that is a significant amount of information lost. But it is important to consider exactly how the images are reduced, because we do not want to feed the network irrelevant information which can be a reason why our accuracy slightly decreased as the images were larger.

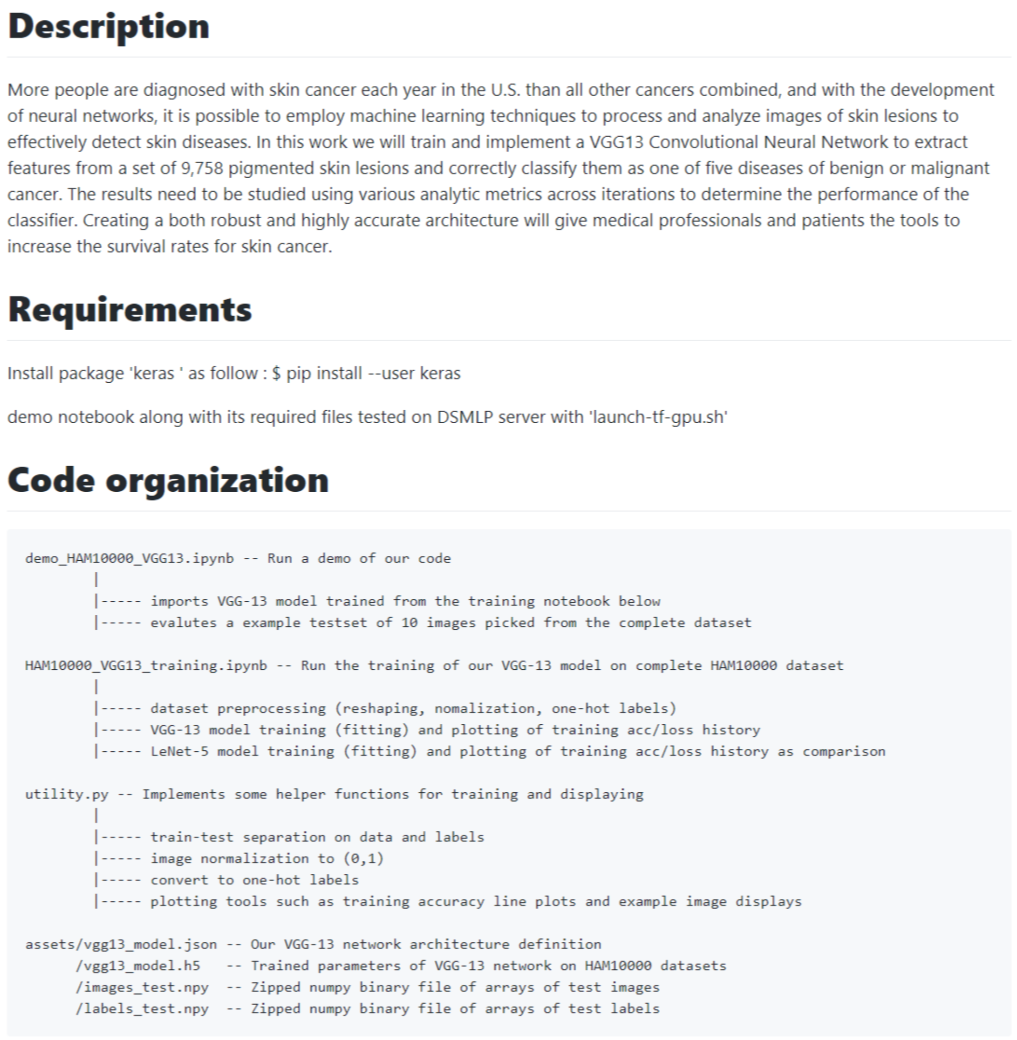
Another point to consider is the possibility that a significant number of additional images at any resolution is necessary to properly extract and separate features. When the network is fed high resolution samples, it might attempt to add for features, but since there is no additional data, the available data becomes sparse. Therefore, a much larger dataset would be necessary to train the network with higher resolutions.

Nevertheless, the application of this concept of skin disease classification does has many innovative potentials. Major sources of detection delay are insufficient self-evaluation, postponement of a visit to the doctor, incorrect evaluation by the doctor, and biopsies. Developing an algorithm that puts diagnostic capabilities with high confidence levels straight into the hands of both doctors and patients can avoid these delays, which is the overall goal of this network. With further development, this network shows that it is possible to eventually design a neural network to revolutionize the approach to dermatology.

References

[1] International Skin Imaging Collaboration. https://www.isic-archive.com/#!/topWithHeader/onlyHeaderTop/gallery, 2018.
[2] T. J. Brinker, A. Hekler, J. Utikal, N. Grabe, D. Schadendorf, J. Klode, C. Berking, T. Steeb, A. Enk, andC. von Kalle. Skin Cancer Classification Using Convolutional Neural Networks: Systematic Review. J MedInternet Res, 2018.
[3] S. Das. CNN Architectures: Lenet, AlexNet, VGG, GoogleNet, ResNet and more . . . . Medium, 2017.
[4] HarisIqbal88. Plotneuralnet. https://github.com/HarisIqbal88/PlotNeuralNet, 2016.
[5] T. Kubota. Deep Learning Algorithm Does as Well as Dermatologists in Identifying Skin Cancer. Stanford News, 2017.
[6] P. Tschandl, C. Rosendahl, and H. Kittler. The HAM10000 Dataset, A large collection of multi-source dermatoscopic images of common pigmented skin lesions.Sci. Data 5, 2018
[7] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun. Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks. Nature, 2017.

GitHub Repository:
https://github.com/liningwei/ECE285_Skin_Cancer_Classcification



Description

More people are diagnosed with skin cancer each year in the U.S. than all other cancers combined, and with the development of neural networks, it is possible to employ machine learning techniques to process and analyze images of skin lesions to effectively detect skin diseases. In this work we will train and implement a VGG13 Convolutional Neural Network to extract features from a set of 9,758 pigmented skin lesions and correctly classify them as one of five diseases of benign or malignant cancer. The results need to be studied using various analytic metrics across iterations to determine the performance of the classifier. Creating a both robust and highly accurate architecture will give medical professionals and patients the tools to increase the survival rates for skin cancer.

Requirements

Install package 'keras ' as follow : $ pip install --user keras

demo notebook along with its required files tested on DSMLP server with 'launch-tf-gpu.sh'

Code organization

```
demo_HAM10000_VGG13.ipynb -- Run a demo of our code
        |
        |----- imports VGG-13 model trained from the training notebook below
        |----- evalutes a example testset of 10 images picked from the complete dataset

HAM10000_VGG13_training.ipynb -- Run the training of our VGG-13 model on complete HAM10000 dataset
        |
        |----- dataset preprocessing (reshaping, nomalization, one-hot labels)
        |----- VGG-13 model training (fitting) and plotting of training acc/loss history
        |----- LeNet-5 model training (fitting) and plotting of training acc/loss history as comparison

utility.py -- Implements some helper functions for training and displaying
        |
        |----- train-test separation on data and labels
        |----- image normalization to (0,1)
        |----- convert to one-hot labels
        |----- plotting tools such as training accuracy line plots and example image displays

assets/vgg13_model.json -- Our VGG-13 network architecture definition
      /vgg13_model.h5   -- Trained parameters of VGG-13 network on HAM10000 datasets
      /images_test.npy  -- Zipped numpy binary file of arrays of test images
      /labels_test.npy  -- Zipped numpy binary file of arrays of test labels
```