ASHESI

ASHESI UNIVERSITY COLLEGE

DEVELOPING A MACHINE LEARNING MODEL FOR MALARIA DIAGNOSIS IN RURAL AREAS

APPLIED PROJECT

B.sc Computer Science

Vladimir Fomene

2018

DEVELOPING A MACHINE LEARNING MODEL FOR MALARIA DIAGNOSIS IN RURAL AREAS

Applied Project submitted to the Department of Computer Science, Ashesi University College in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer Science

Applied Capstone Project

Vladimir Fomene

2018

DECLARATION

I hereby declare that this applied project is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

………………………………………………………………………………………………………

Candidate's Name:

………………………………………………………………………………………………………

Date:

………………………………………………………………………………………………………

I hereby declare that preparation and presentation of this applied project were supervised in accordance with the guidelines on supervision of applied projects laid down by Ashesi University College.

Supervisor's Signature:

………………………………………………………………………………………………………

Supervisor's Name:

………………………………………………………………………………………………………

Date:

………………………………………………………………………………………………………

Acknowledgements

To my supervisor, **Prof. Lorenzo Torresani** without whose advice and guidance, this project implementation will not have been possible. To my friend, **Hassan Sillah**, whose work with Tuberculosis diagnosis gave rise to this project.

Abstract

Medical diagnosis of diseases like Malaria and tuberculosis still use microscopy as a standard, but this procedure is usually very tiring for pathologists and health workers as it imposes much stress on their vision. Due to the fatigue that health workers get from this process, they might end up misdiagnosing a case. In most Rural areas of Cameroon and Ghana, there are no qualified personnel to do these diagnoses. Moreover, according to the World Bank, malaria still kills millions of people every year in Sub-Saharan Africa. To solve this problem, we used a machine learning approach; transfer learning to retrain an already existing model to perform binary classification on malaria blood smear images. The pretrained model was already optimized for devices with low memory, therefore this project's model can work on low memory devices with no network connectivity. This project also explored Generative Adversarial networks as an alternative way of training a classifier for scenarios with data scarcity. This project shows how a model trained on a different task can be retrained to solve a similar task and shows a technique for developing a classifier in scenarios of data scarcity.

Table of Contents

Table of Figures

Chapter 1: Introduction

1.1 Introduction:

This research will focus on using machine learning algorithms to improve medical diagnosis of tropical diseases using microscopy. To explore this field of medical diagnosis, this project will focus on malaria diagnosis. This project will use the recent advancements in the field of computer vision to improve medical diagnosis via microscopy.

1.2 Problem Statement:

Microscopy is still the gold standard for diagnosing diseases such as malaria and tuberculosis which are very common in West and Central Africa. Previous research done by Boray Tek, Andrew Dempster and Izzet Kale points out the following shortcomings about manual microscope diagnosis (Boray Tek, 2009):

- Error-prone

Although microscopy has served us for long, it usually poses a problem of accuracy, efficiency and always requires trained personnel to do a diagnosis. After interviewing the lab workers at Noguchi Medical Institute at the University of Ghana, I learned that some malaria diagnosis at hospitals are not accurate. That is, some patients can have malaria but when diagnosed by the medical personnel their test was negative. According to this workers, these errors were coming because the medical professional who oversaw the diagnosis had misread the slides.

- Time-consuming and Repetitive

Moreover, the malaria diagnosis process is often time consuming and repetitive, patients must wait for some few minutes before they get their lab results and since the process is repetitive, it could easily be automated to save time for both patients and health workers.

Lastly, West and Central Africa states do not have many of these trained personnel to operate this microscopes in rural health centres.

1.3 Motivation:

According to the World Health Organization, in 2015 90% of 212 million malaria cases where from Sub-Saharan Africa and of the 429 000 malaria deaths, 92% of them were from Sub-Saharan Africa. There have been great developments and advancements in the field of machine learning (sub-field of artificial intelligence) especially in the field of computer vision with the development of convolutional neural network which can be used to address this problem. Due to this development, we have seen the emergence of tools and libraries that are helping developers who are not machine learning experts develop intelligent software. We will use these libraries and tools to develop a machine learning algorithm that will learn how to diagnose malaria. This software cannot run on our current microscopes given that they do not have an operating system. We will develop a mobile application using machine learning models which will read malaria blood smear images and will detect the malaria parasite on these images.

To validate our research, we will compare the efficiency of this machine learning software to skilled personnel in malaria and tuberculosis disease diagnosis.

1.4 Project Objectives:

- Collect positive and negative samples of malaria blood smear images from a hospital or research institute.

- Train a machine learning model to diagnose malaria in areas with no connectivity and on low memory devices.
- Port this classifier to an android application for testing with malaria blood smear images.

1.5 Related Work:

F Boray TekEmail author, A. G. (2009). Computer vision for microscopy diagnosis of malaria. *Malaria Journal*.

This article explores a lot of techniques that has been used in automating the diagnosis of Malaria. The paper uses images from a thin blood smear for the detection of plasmodium falciparum. This paper follows the following methods sequentially for doing an automated diagnosis: image acquisition, image variations, illumination and thresholding, scale and granulometry, stained pixels and objects and finally classification. The paper highlights the challenges of variations in imaging due to microscope lighting and blood smear preparation and proposes many techniques to deal with it. The paper ends up by talking about classification as a differentiation of the different structures found in the blood and a differentiation between healthy red blood cells and unhealthy red blood cells. The researchers also talk of classification as a differentiation between different life stages of the parasite. For future work, the paper proposes the use of these methods on thick blood smear because it is more sensitive in detection of the parasite.

This paper is gives a very good background on the methodologies used in past research for automated malaria diagnosis using computer vision. Although the paper is old (2009), it makes a good analysis on previous work by highlighting its shortcoming and proposing better methods to use. The article falls short of expectations because it only focuses on thin blood smears and does not talk much about a practical computer vision software that was developed. Its emphasis on theoretical analysis makes it a good choice for our background research.

Mehrjou, A. (2017). Towards life cycle identification of malaria parasites using machine learning. *Cornell University Library*.

This article focuses on using an unsupervised algorithm and manifolds in automated diagnosis process of malaria. It focuses on solving the limitations of supervised algorithms by providing an unsupervised algorithm which reduces variations in the whole diagnosis process and increases speed of diagnosis without undermining the sensitivity level. Unlike other papers, this one is very practical, as it discusses the machine learning software used in automated diagnosis. The paper highlights the following as drawbacks of supervised approaches: time, considering single pixels, ignoring colors in the marking process and susceptible to uneven illumination. The paper also focuses on classifying the different life stages of the malaria parasite.

This paper was released this year and shows the MATLAB running version of the machine learning program. It proposes a completely different approach to most of the previous research, by using unsupervised learning. It focuses on reducing diagnosis time and also explain how this unsupervised approach reduces variations in the diagnosis process. It is well suited for my research because it proposes a new method for developing an automated diagnosis tool and aside from detecting the parasite, it also focuses on classifying different life stages of the parasite.

Luıs Rosadoa, J. M. (2016). Automated detection of malaria parasites on thick blood smears via. *Procedia Computer Science*.

This paper uses supervised classification to assess the present of Plasmodium Falciparum trophozoites and white blood cells in a Giemsa stained thick blood film. The researchers in this article used a smartphone to get the images for the automatic diagnosis process. They used OpenCV (an open source computer vision library) for the supervised classification. This research used 194 images. They used support vector machine classifier and 314 features to classify the images. The researchers ran their software on a computer and an android phone to test the speed and memory usage of their software in each instance. The paper shows a 98.2% of sensitivity and 72.1% of specificity in automatic detection of white blood, while the Plasmodium Falciparum trophozoites detection achieved a sensitivity of 80.5% and a specificity of 93.8%.

This paper is one of the few papers to use a supervised approach of automatic diagnosis on a thick blood smear. The researchers used Valgrind to analyze the memory usage of their program on a computer and an android phone. This is very relevant for my research because my research is focused on developing a computer vision software that will work on android smartphones. The researchers used a smartphone to capture the images they used for the diagnosis and this has the disadvantage that it will not pick all the parasites and other blood particles. The paper is also narrow because it only focuses on detecting one life stage of the parasite.

Arnon Houri-Yafin, Y. E.-S. (2016). An Enhanced Computer Vision Platform for Clinical Diagnosis of Malaria. *Malaria Control and Elimination*.

In this research, a device for malaria diagnosis, speciation and parasite quantification. These researchers did an evaluative test of this device clinically in Lancet laboratories Johannesburg and City Hospital India. The researchers conducted their trials on plasmodium vivax and plasmodium falciparum

This research is important to me because of their use of two species of the parasite and methods they used for quantification of the parasites. It is also valuable because it shows a clinical implementing of a computer vision tool for automated malaria diagnosis.

(Andrew G. Howard, 2018). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications

Researchers working on this pretrained model decided to use a different type of convolution layers called depth wise separable convolutions. These layers are made up of a depth wise convolution and a pointwise convolution. This kind of layers are more efficient compared to normal convolution layers. MobileNets require low memory than other pretrained models like Inception and VGG. According to the paper, MobileNets are faster during predictions than Inception and VGG.

(Apple, 2017). Improving the Realism of Synthetic Images

This paper outlines how Generative Adversarial networks(GANS) can be used to generate eye gaze images from noise. It is great for our context because it gives a good explanation of GANS and talks about the different losses to optimize during the training process. It will help us implement a GAN that generates negative malaria blood smear images that are very similar to positive images so that hopefully the malaria classifier ends up learning the real difference between negative and positive malaria blood smear images.

Chapter 2: Requirement Specification

2.1 Requirement Gathering Procedures:

Interview:

I carried out an interview with the lab workers at Korle-Bu hospital lab for tuberculosis to understand the whole tuberculosis diagnosis procedure.

Observation:

While interning at Noguchi Institute, I observed the whole malaria diagnosis process and was able to figure out how computers could be used in the process.

Previous work:

Some of the requirements were extracted from the related literature work.

Users:

This application is going to be used by health workers to identify how well the classifier is doing compared to the lab workers diagnosing malaria.

Use Case:

This application will be used by health workers to evaluate how our classifier is doing compared to humans on the task of diagnosing malaria.

2.2 Requirements:

We will focus on two types of requirements in this chapter, user requirements and system requirements. The project is still at the level of data collection from stakeholders (doctors in clinics) so will solely focus on system requirements.

System requirements are usually divided into two groups: functional and non-functional requirements. For the malaria diagnosis, we will focus on developing a computer vision software that can work in regions without internet connectivity.

2.3 Functional Requirements:

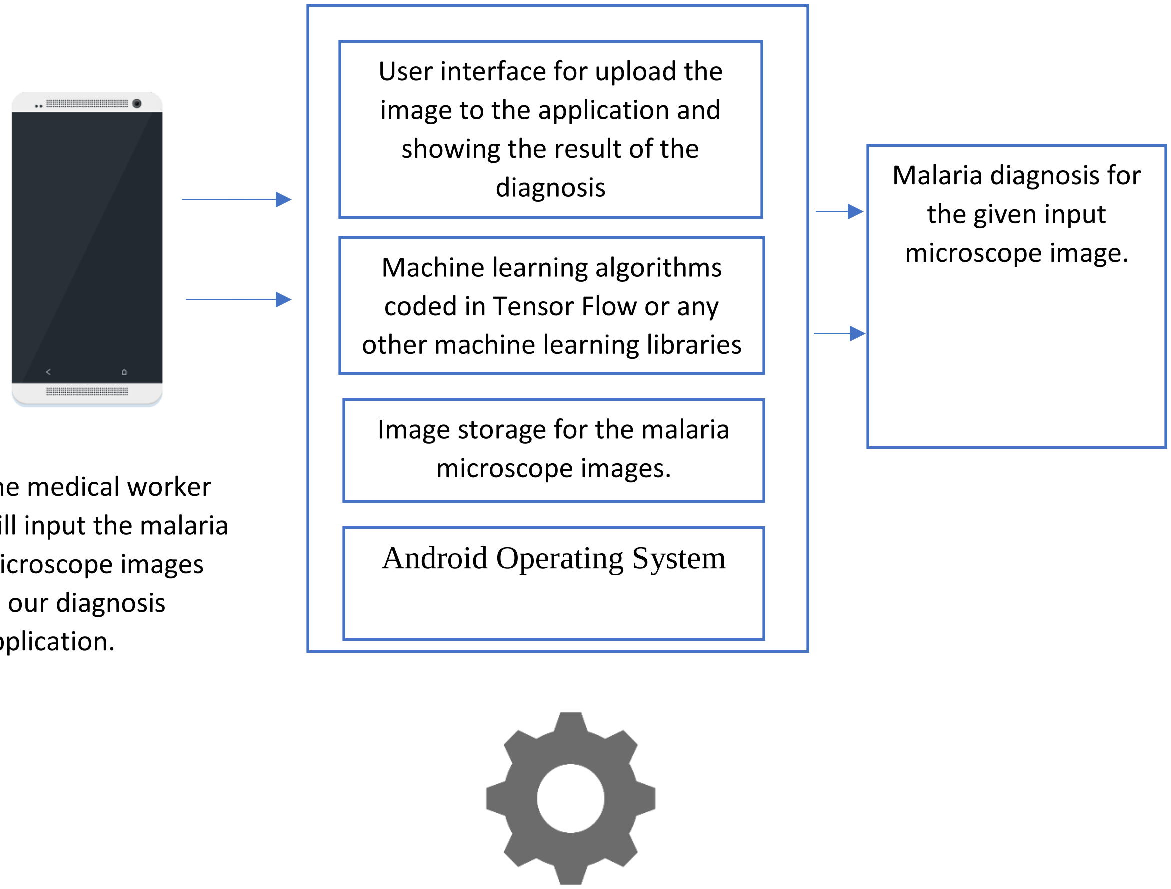
- The diagnosis software is supposed to run on mobile devices with limited internal memory.
- The software should work offline since many health centers in the rural areas do not have access to the internet.
- The application will have an upload image functionality to allow health worker to upload a patient’s image from a microscopic blood film.
- The application will increase clarity of blurred images to increase accuracy during diagnosis.
- The application will process the image and tell if the blood film has malaria or not.

2.4 Non-Functional Requirements:

- The application should work on mobile devices with very low memory and a slow graphic processing unit.
- The application will provide the same user experience online and offline.

Chapter 3: High level Architecture

This schema depicts the high-level architecture of the malaria diagnosis clinical application.


User interface for upload the image to the application and showing the result of the diagnosis
Machine learning algorithms coded in Tensor Flow or any other machine learning libraries
Image storage for the malaria microscope images.
Android Operating System
Malaria diagnosis for the given input microscope image.

The medical worker will input the malaria microscope images to our diagnosis application.

Figure 1

Architectural Components

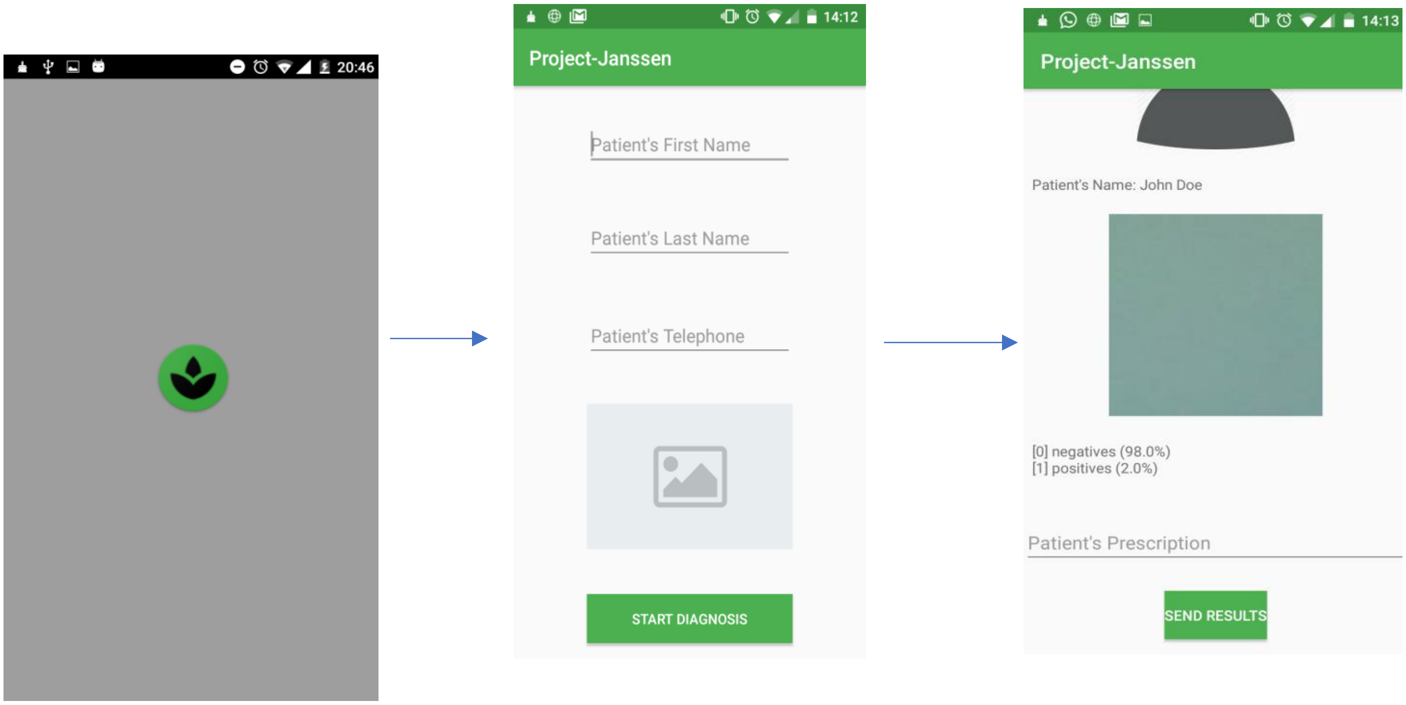
3.1 User Interface:

The user interface of the malaria diagnosis app is very simple. It is a set of android application activities that will take user information and displays results. It will take in the patient’s personal details like their phone number, full name and address.

3.2 User Activity Diagram:

See Figure 3 below.

3.3 Mobile app activities workflow:



Patient's Name: John Doe

[0] negatives (98.0%)
[1] positives (2.0%)

Figure 2

3.4 Image Storage:

The application will have a local storage for all the images it analyzes. Although this will overload the android device, we cannot put the image storage in the cloud because the application will be tested in areas where there is either no internet connectivity or low internet bandwidth.

Machine learning Algorithm:

The machine learning algorithm will be developed as part of the android application using already existing machine learning libraries like Tensorflow.

3.5 Model Architecture:

To predict if a patient is malaria positive or negative a classifier was trained to take in an image of a malaria blood smear from a health worker and classifies as either negative or positive. Our classifier was built by fine-tuning MobileNet, one of Google's multiple pretrained models on ImageNet (1.2 million image databases with 1000 classes) that has been optimized for devices with low memory and slow computational power. Below is the architecture of the model:

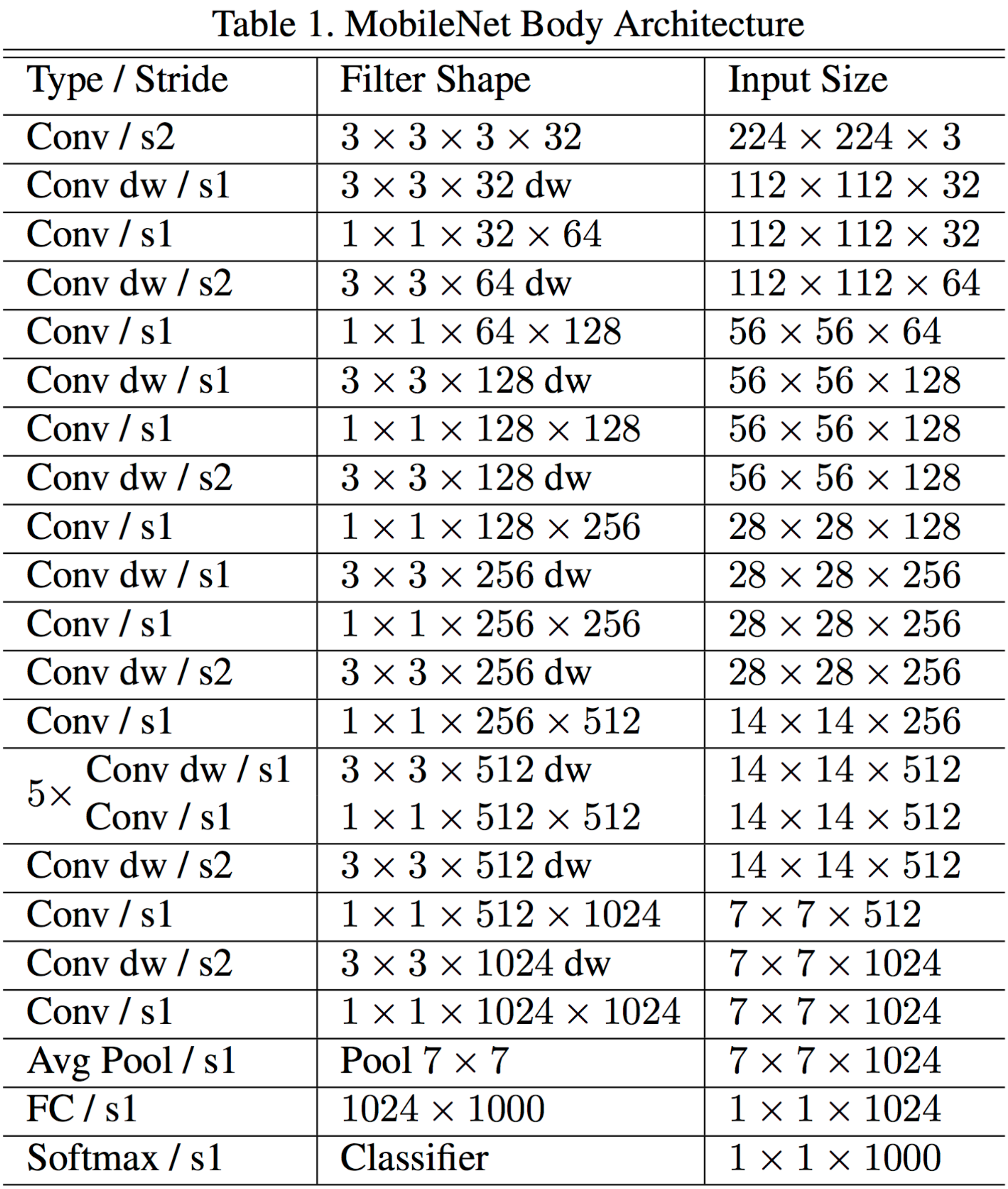


Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
5× Conv dw / s1 Conv / s1	3 × 3 × 512 dw 1 × 1 × 512 × 512	14 × 14 × 512 14 × 14 × 512
Conv dw / s2	3 × 3 × 512 dw	14 × 14 × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

Table 1.0

Figure 3

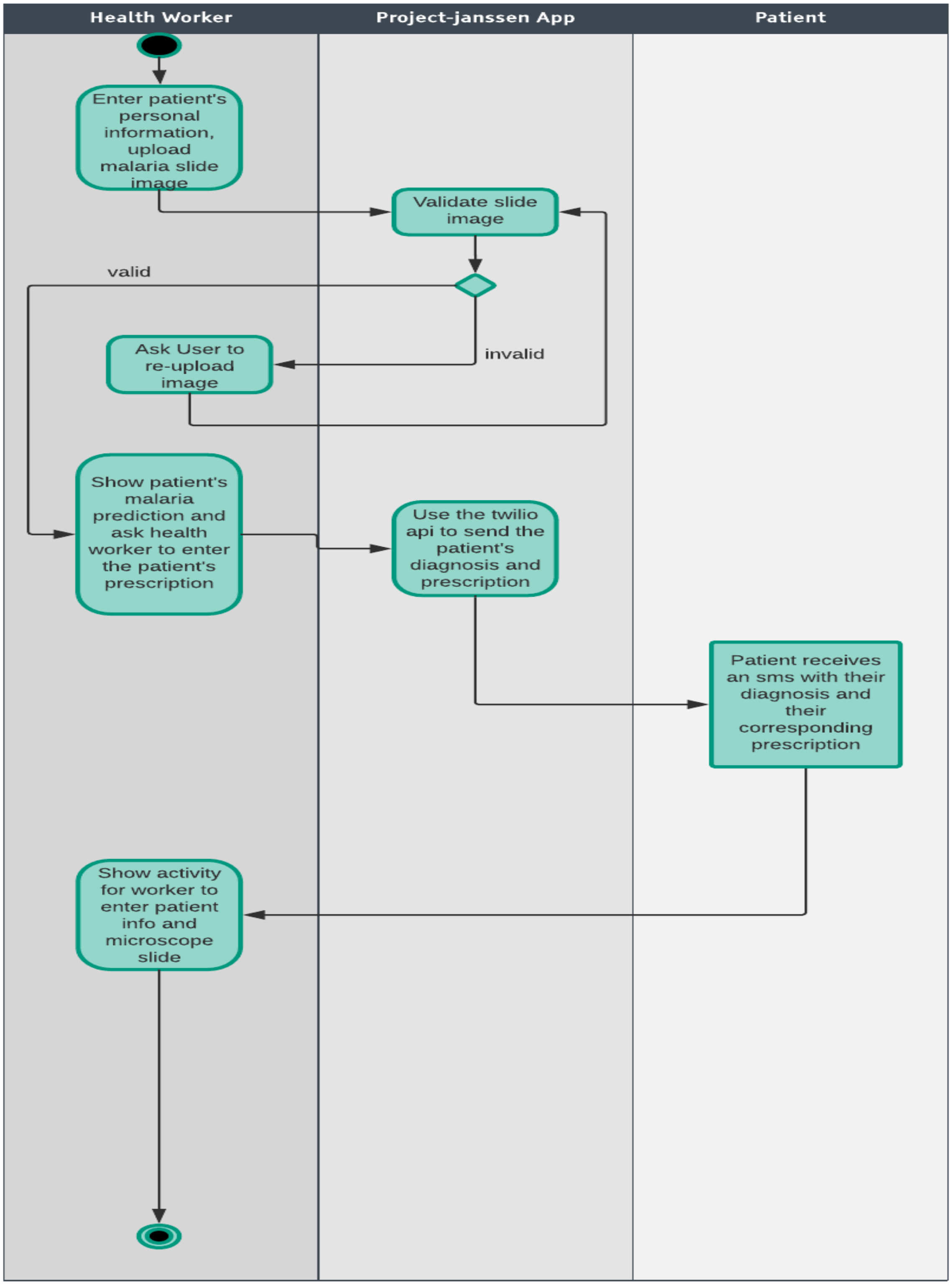

Health Worker
Project-janssen App
Patient
Enter patient's personal information, upload malaria slide image
Validate slide image
valid
invalid
Ask User to re-upload image
Show patient's malaria prediction and ask health worker to enter the patient's prescription
Use the twilio api to send the patient's diagnosis and prescription
Patient receives an sms with their diagnosis and their corresponding prescription
Show activity for worker to enter patient info and microscope slide

Figure 3

Chapter 4: Implementation

4.1 Implementation Tools and libraries:

To implement this project, we used the following tools and libraries:

Tensorflow (Machine learning library from Google), Atom editor, Android Studio, Pycharm IDE for python, python Augmentor library, Jupyter Notebook and ImageMagick.

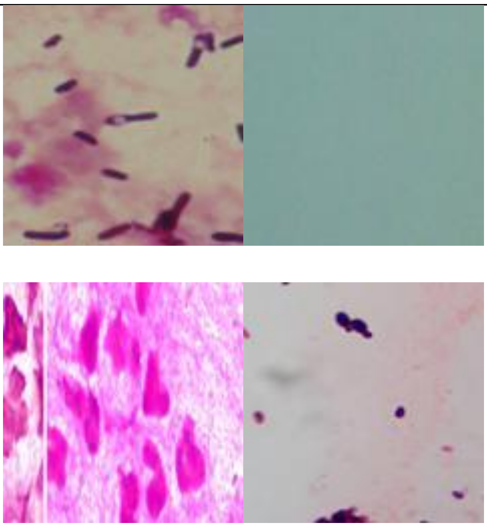
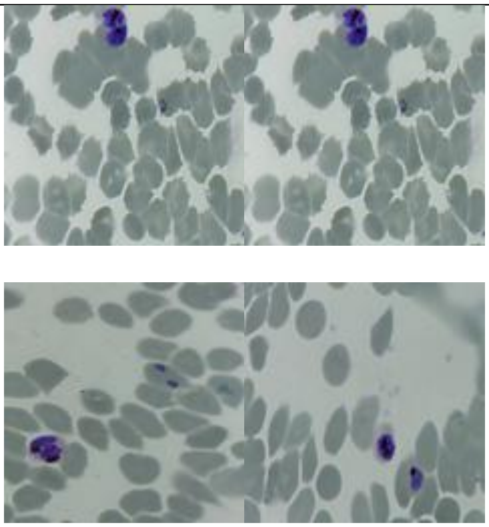
4. 2 Data Collection:

After communication with the team of researchers at Noguchi Institute, we were able to get 107 negative bitmap images of blood smears of patients and 354 positive bitmap images of malaria blood smears. This data was not balanced, diverse and the quantity was not substantial to train a convolutional neural network on it. This means the data has to be augmented.

4.3 Preprocessing:

MobileNet can only train on Jpeg image format and the Noguchi dataset are all bitmap images. To train on the MobileNet model, we resize our images to 224px X 224px and converted all of them to jpeg format. To make the two classes balanced and increase the diversity in our training set, we augmented the data in each class of our training set using reflection, vertical flipping, horizontal flipping, 90 degree rotation, 270 degree rotation and random cropping. After augmentation we have 2000 images, in both classes. Data augmentation is important because it will reduce the possibility of our model overfitting over particular orientations of the dataset images because the model will learn from several orientations of the images. Therefore, our two sets of data are finally balanced and ready for training. We use a python package called Augmentor for this step.

4.4 Sample Data:

Negative Samples	Positive Samples
	

4.5 Training:

For training, the images were arranged into two folders, one for positive malaria images named “Positive” and another for negative images named “Negative”. During training 20% of the images were selected for validation and 10% for testing. The classifier will learn 1001 * N parameters where N represent the number of labels (Google, n.d.). The classifier was trained using mini-batches of 10 images in 4000 steps. A learning rate of 0.01 was used for training. A new layer was added to the Mobilenet_1.0_224 architecture and then a SoftMax layer to produce prediction probability of our new classes. Gradient descent was used for optimization and cross-entropy as the loss. Let S be a SoftMax function, (x, y) a training pair from a training set and L the one hot encoding vector representing our label y. The cross-entropy loss, D is the following:

$D(S,L) = -\sum_i Li * \log(Si)$ where Li and Si both represents the i-component in L and S respectively.

4.6 TensorFlow Model Representation:

This graph shows TensorFlow's representation of the model. Looking at it you can see the input layer, bottleneck layer (layer before the output layer) and the output layer.

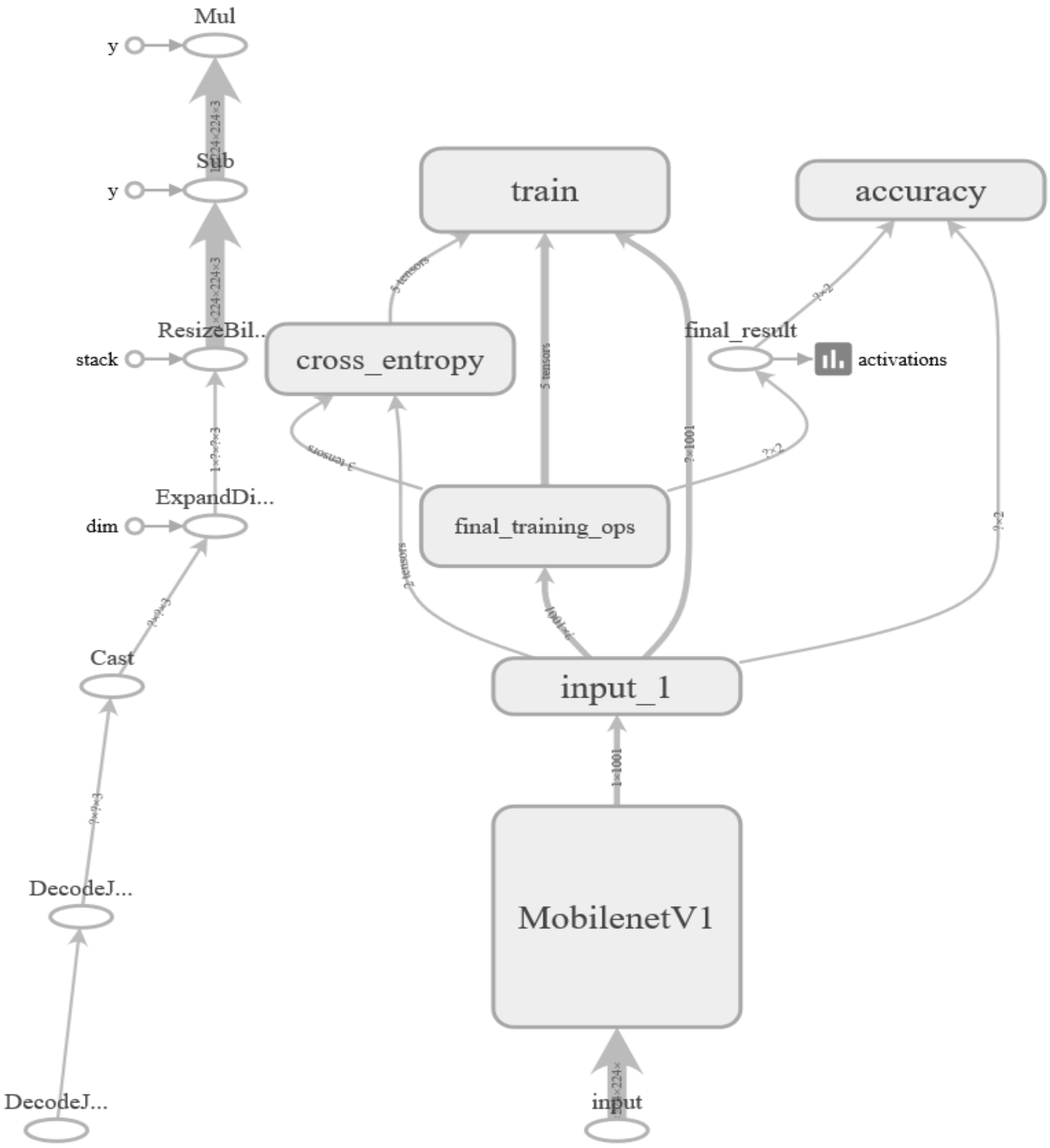


Figure 4

4.7 Porting to Android:

After training the classifier on a laptop, we added it to the asset folder of the android project as a graph file (retrained_graph.pb). In addition, a text file was added with the labels used for classification. The TensorFlow dependency for Android was added to Gradle build. Three activities were built into the android application. The splash screen activity which serves as a starter activity for the app. The MainActivity.java which serves as a screen for the health worker to enter the patient's information and then select the image corresponding to the patient's malaria blood smear. Once, all this information entered the health worker will tap the diagnose button which will prompt Android SDK to pass the image to the Android NDK which has a TensorFlow C++ wrapper. This wrapper will convert the image into a tensor and pass it as input to the model which will an array of length two with probabilities for the negative and positive predictions. The TensorFlow inference interface allows the SDK to interact with the NDK. These predictions are sent to the MedicalResultActivity.java, the same activity also prompts the health worker to enter a diagnosis for the patient. This diagnosis is sent to the patient via SMS using Twilio's API.

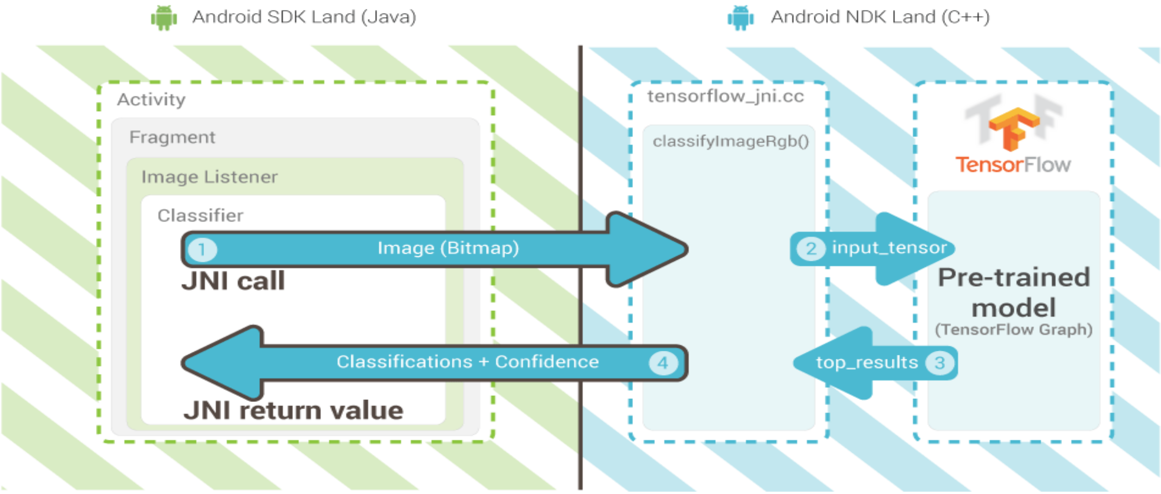


Figure 5

4.8 Generative Adversarial Network(GAN) Training:

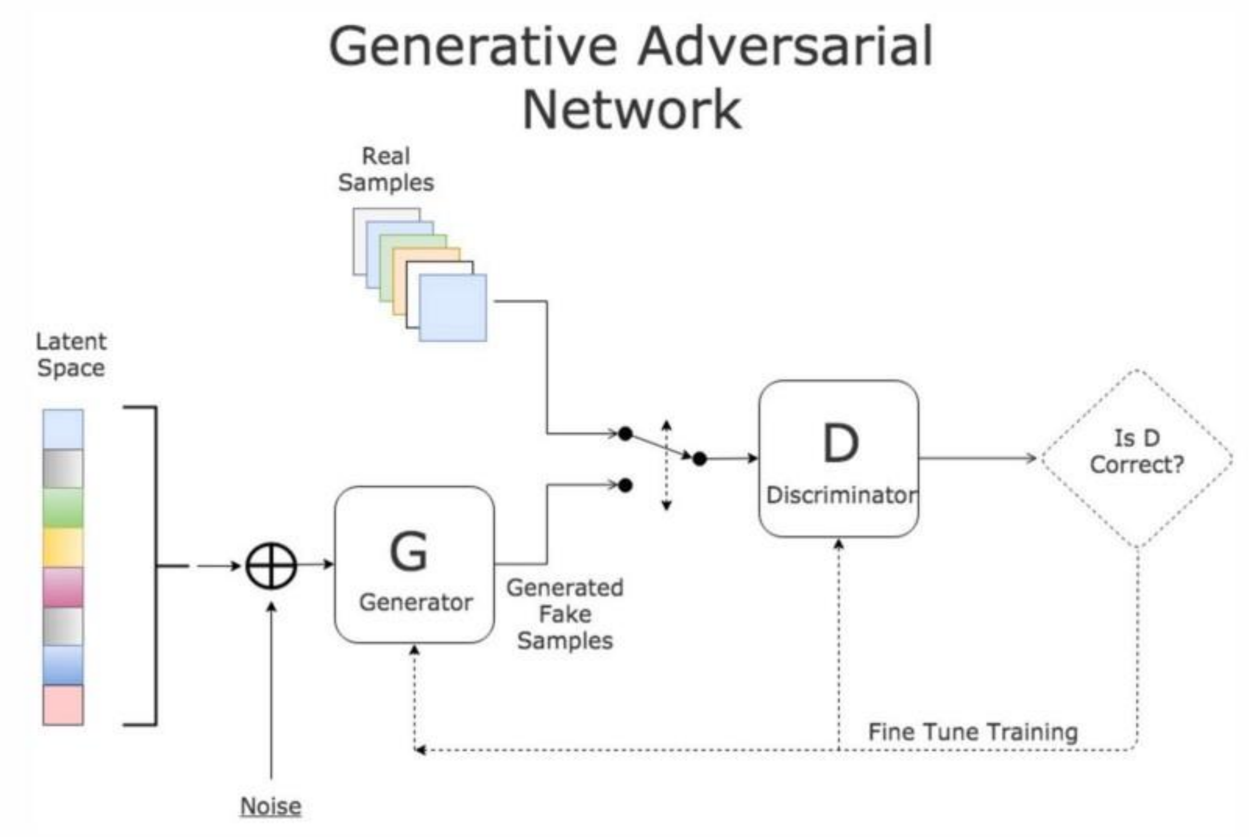


Figure 6

A generative adversarial network is a model made of up of two neural networks, a generator and a discriminator. A generator is a network which focuses on forging images which look exactly like the dataset images from noise input and the discriminator is train to determine if an image is real or fake given an image by taking the real images and the noise as input (Mosquera, 2018). In our case instead of our generator learning taking random noise as input, it is going to take our current negative dataset and try to generate positive images from it. Our discriminator will take our current positive and negative data and will learn how to discriminate the positive from the negatives. The benefit of this approach is that after training our discriminator will have really learned the features of positive images and will be able to discriminate if an image is positive or not(negative).

To develop this model, we trained both the discriminator and the generator on 1000 epochs. We trained both the discriminator(classifier) and generator on batches of 10 images. The generator was trained by taking negative images from the training set as input. The discriminator was trained with positive images and output images from the generator.

Architecture of the Generator:

The generator is made up of three linear layers which are compose with RELUs to add non-linearity to the model. The final layer is a linear layer compose with a tanh function as activation.

Architecture of the Discriminator:

The discriminator is made up of three linear layers which are composed with RELUs and then dropouts. For the output layer the discriminator uses a linear model.

For optimizing the model, we used the ADAM optimizer for both the generator and the discriminator with a learning rate of 0.0002. The following equations represent the cross-entropy loss for both network:

Discriminator Loss:

$$Gen_{loss} = \frac{1}{m}\sum_{i=0}^{m}[\log(D(x^{i}) + \log(1 - D\left(G(Z^{i})\right))]$$

Generator Loss:

$$\mathrm{D}iscrimin_{loss} = \frac{1}{\mathrm{m}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{m}} \log(1 - \mathrm{D}(\mathrm{G}\left(Z^{i}\right))$$

Where D : discriminator network; G : generator network; m: number of training examples; x^i: i-real image(in our case i-positive image) and z^i: i-fake images(in our case i-negative image).

Chapter 5: Testing and Result

5.1 Training Results:

After retraining the MobileNet neural network on the positive and negative datasets, they produced the following results:

Training Operation	**Images Size**	**Accuracy**
Training	70% of the dataset	100%
Validation	20% of the dataset	100%
Testing	10% of the dataset	100%

From the above table, the model is doing extremely well on this classification task. Although these are great results, it is also hinting that this classification task is trivial for the model. This task is trivial because the negative samples in our dataset are very different from the positive samples which is not the case in practice. In particular, the negative dataset is not representative of a healthy person. Therefore, the model is not learning the right parameters to differentiate between negative samples and positive samples.

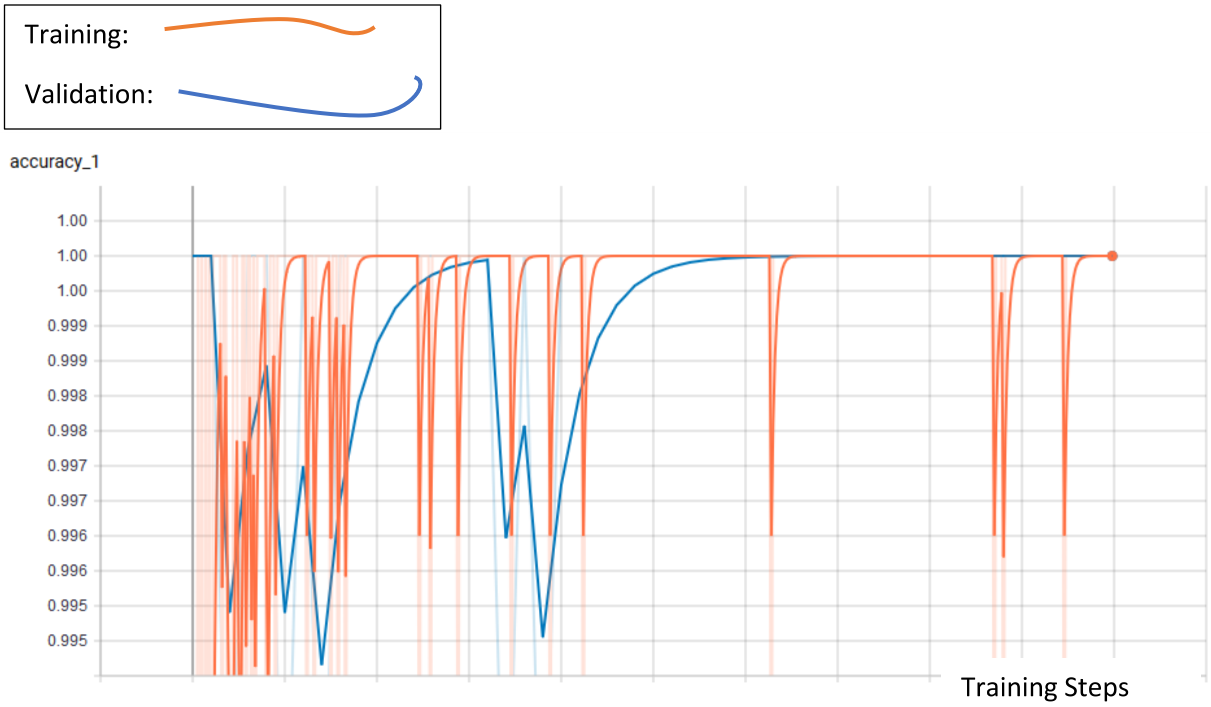
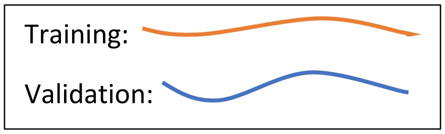


Figure 7

The above figure shows how the training accuracy starts off very small and gradually starts increasing until it finally plateaus very close to 100%. The validation accuracy shows the same trends as the training accuracy. This means that the model was able to perform well on the training set without overfitting because it still performed well on the validation set.



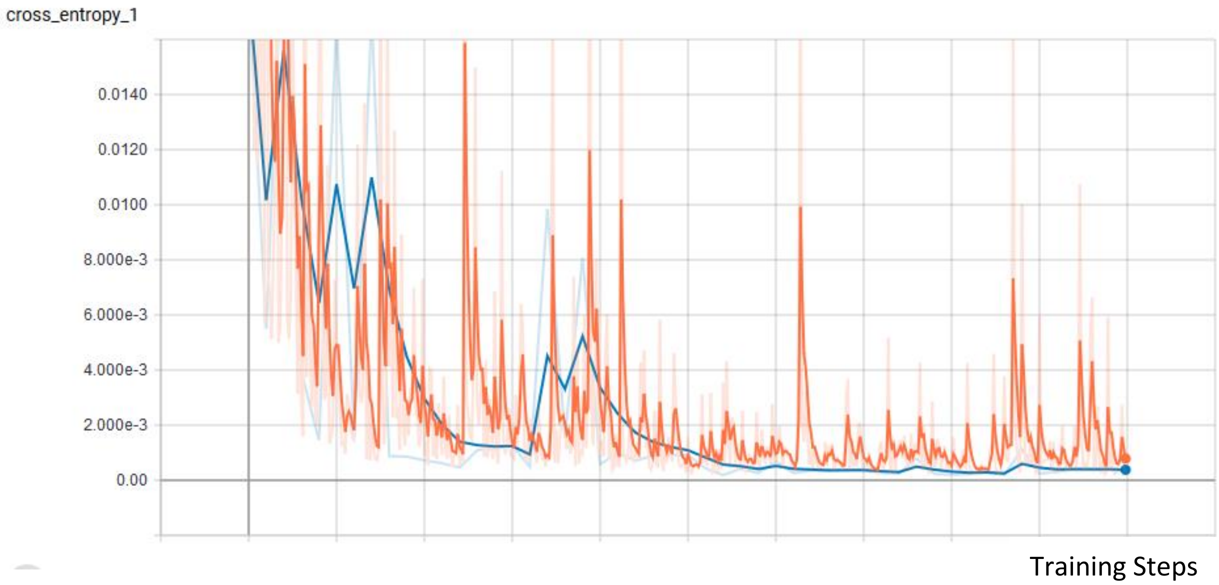


Figure 8

The above graph shows that the cross-entropy loss decreasing both during the training and validation. This is indication that the model made good progress on the classification both during training and validation.

Despite having such amazing results while training the classifier, it cannot be proven that the classifier is learning the correct features because the negatives images are not representative of a healthy individual who is not suffering from malaria. To solve this data problem, a generative adversarial network will be trained on the data.

5.2 Generative Adversarial Network Training Results:

After training the GAN for 1000 epochs, we got the following progression of the generator and discriminator loss:

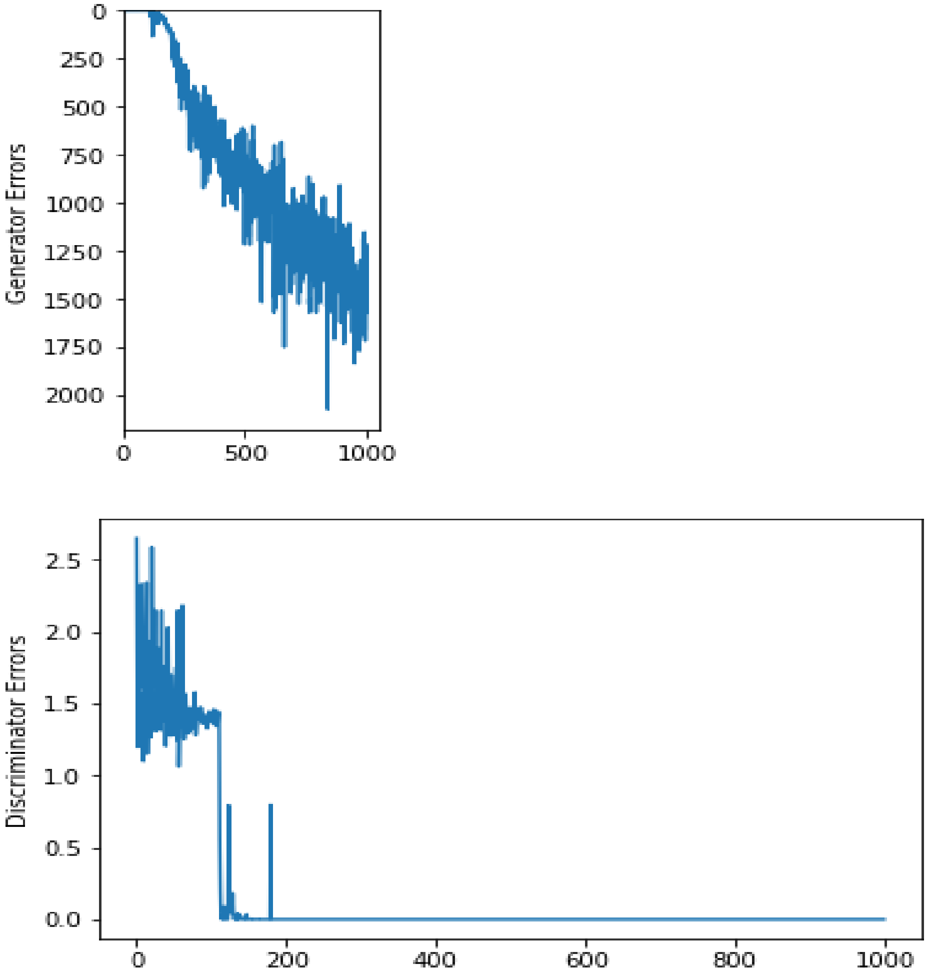


Figure 9

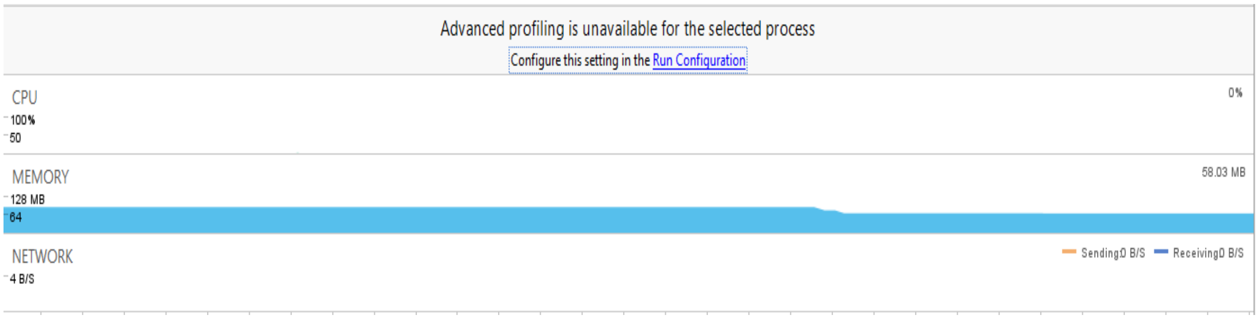
From the figure above, we can say that at the beginning of the training process the generator loss is low and the discriminator loss is high because the generator start off with images that are not too far from the positive ones compared to noise and the discriminator has not yet learnt how to distinguish between positives and negatives. As the discriminator becomes confident in discriminating between positive and negative images the generator loss starts increasing.

5.3 Application Memory Usage:

The application was tested on one android phone to access how much memory and CPU is used by the model. After building the model into the app, the APK file had a size of 52.28MB which is smaller than social apps like WhatsApp and Facebook which are about 100 MB, but stills run effectively on phone with low memory. Here are the results of testing the application for memory and CPU usage (at beginning of profiling the phone had 93.71MB of unused memory):

Device Name	**Start of image upload**	**End of Image Upload**	**Before classification**	**After classification**
CPU Usage	~0%	5%	~0%	%13
Memory Usage	93.71MB	74.666	91.06 MB	111.05MB

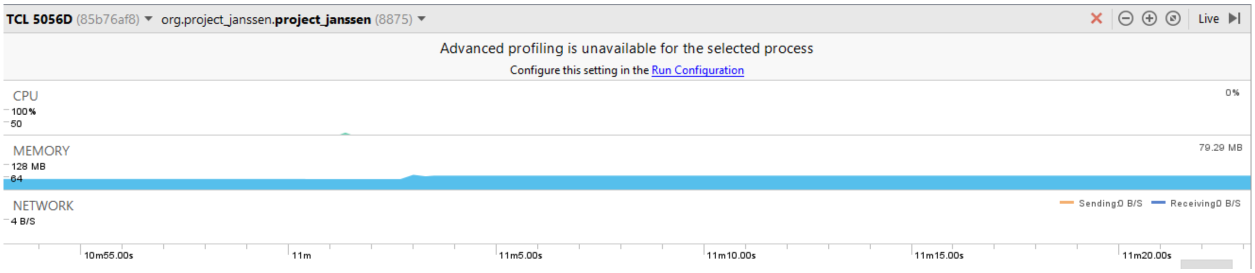
When a user is uploading an image to the application there is decrease in memory usage as he is leaving the context of the application to pick a file from the file system (see Figure 10). On the other hand, there is an increase in memory usage by 10.01MB when a user calls the model in the app. This is seen in Figure 11.



Advanced profiling is unavailable for the selected process

Configure this setting in the Run Configuration

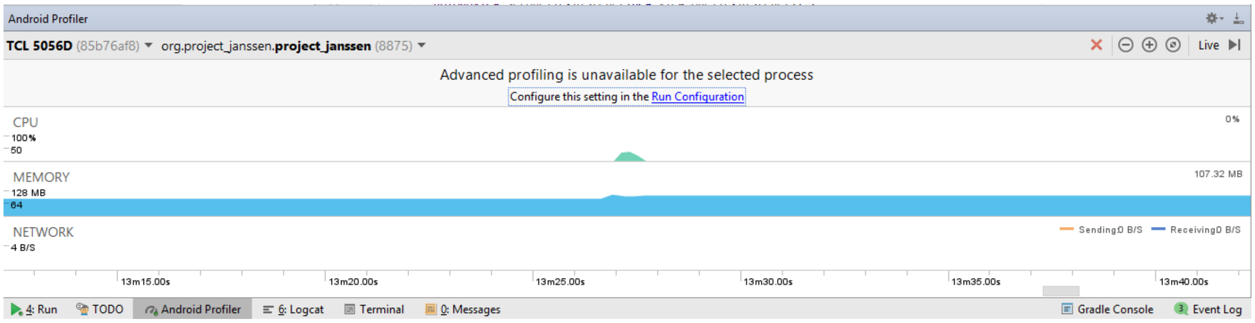
Figure 10



Advanced profiling is unavailable for the selected process

Configure this setting in the Run Configuration

Figure 11



Advanced profiling is unavailable for the selected process

Configure this setting in the Run Configuration

Figure 12

With regards to CPU usage, there is an increase of 5% in CPU usage on image upload and an increase of 13% during classification. This 13% increase in CPU usage

Chapter 6: Conclusion and Recommendations

The machine learning model built through this project shows how computer vision can be applied to the field of medical diagnosis to make the work of pathologists and doctors easier. With this model, we can do a binary classification of malaria blood smear images by uploading these images to a phone and then passing these images through a classifier to predict if a person is malaria positive or malaria negative. This project's classifier was trained using Google's Mobilenet convolutional neural network, this model has been optimized for mobile devices which makes the model suitable to work in areas where there is no internet connectivity and on low memory devices. While working on the classifier, our negative examples where not representative of a person without malaria. To make sure that our model was learning the right features, we trained a generative adversarial network(GAN) on data where the discriminator was to discriminate between positive and negative images instead of real and fake images. The project explored using GAN to create classifier rather than using it to generate data as it is usually the case. Despite these great milestones, much is still left to be done.

6.1 Future Work:

This project sets the stage for very interesting improvements and explorations as it has proven that machine learning could be used on low memory devices with no connectivity to detect malaria. The following are my recommendations for future work:

- For a start, this system needs to be tested in clinics and hospitals to document how well the system is doing in practice compared to humans. Based on the model's practical performance, it could be improved by tuning hyperparameter during training.
- This project could also be used as foundation for developing a real time malaria parasite detection system. The advantage for such a system is that it could help pathologists focus

on the right spots when looking at a malaria blood smear thereby speeding up the malaria detection process. This could also be very helpful in computing malaria parasites to red blood cells which is the determining factor used by pathologists for malaria diagnosis.

- While working on this project, our greatest challenge was getting data that was representative of the problem and that was of high resolution. To retrain our classifier for higher performance, there is need to collect data with higher resolution and better labeling from hospitals and clinics. For this to happen, the images need to be taken with microscopes of better camera quality to get images of higher resolution.
- The techniques used in this project could easily be applied to solve similar diagnosis problem for tuberculosis and for other medical procedures where classifier could be good at producing medical results.

References

Andrew G. Howard, M. Z. (2018). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *Cornell University Library*.

Apple. (2017). Improving the Realism of Synthetic Images . *Machine Learning Journal*.

Boray Tek, A. D. (2009). Computer vision for microscopy diagnosis of malaria. *Malaria Journal*.

Mosquera, D. G. (2018, February 1). Retrieved from https://medium.com/ai-society/gans-from-scratch-1-a-deep-introduction-with-code-in-pytorch-and-tensorflow-cb03cdcdba0f

Arnon Houri-Yafin, Y. E.-S. (2016). An Enhanced Computer Vision Platform for Clinical Diagnosis of Malaria. *Malaria Control and Elimination*.

Luıs Rosadoa, J. M. (2016). Automated detection of malaria parasites on thick blood smears via. *Procedia Computer Science*.

Mehrjou, A. (2018). Towards life cycle identification of malaria parasites using machine learning. *Cornell University Library*.