Malaria Detection: Deep Learning Capstone Project Final Report

Erma C.A.

EXECUTIVE SUMMARY

The World Health Organization (WHO) reports over 400,000 people die of malaria each year worldwide. It is estimated that 2/3 of the deaths are children under the age of five. Malaria is the leading cause of death and disease in developing countries. The WHO reported 229 million infections and 409,000 deaths from malaria in 2019. Malaria is considered one of the most severe public health problems worldwide, though it is less prevalent in the U.S. The Centers for Disease Control & Prevention (CDC) reports only 2000 cases of malaria are diagnosed in the U.S. each year. The majority of the cases in the U.S. are travelers returning from Africa and South Asia where malaria is endemic.

Malaria is a bloodborne disease caused by plasmodium parasites that infect red blood cells (RBCs). The parasites enter the blood stream via the bite of an infected female Anopheles mosquito then grow within the RBCs causing damage. Malaria is curable if diagnosed and treated promptly. Delays in diagnosis are the leading cause of death in patients. Microscopic examination of blood smears remains as the "gold standard" for malaria diagnosis. Diagnosis is a time-consuming process that involves examination of multiple blood smear samples. Also, in the U.S. where malaria is not endemic, lab personnel fail to detect the parasite due to lack of experience. This project aims to improve malaria diagnosis by bridging the gap in lack of lab personnel experience and works towards eliminating the need for lab personnel by automating the diagnosis process using machine learning techniques.

The objective of this project is to build an efficient computer vision model that can help with the early and accurate diagnosis of malaria by identifying whether the image of an RBC is that of one infected with malaria and classifying as parasitized or not infected and classifying as uninfected. A convolutional neural network (CNN) deep learning model was built, trained, validated and tested with an image dataset of human RBCs. The dataset consisted of 27,558 singular RBC color images equally divided into two mutually exclusive classes, uninfected and parasitized. The images of varying sizes were resized to 64 x 64 pixels to account for limited computing capacity. Seven different models were created. Of those seven models one used Keras turner HyperBand to try and optimize the model hyperparameters and one used transfer learning with the VGG-16 pretrained model. Performance was good for all seven models with Model 4 as the best model out of all based on the highest accuracy 95 % and the highest recall 94% for the parasitized class. Recall for the parasitized class is more important due to the high risk of misclassification. A false negative can ultimately lead to the death of a patient if the patient is not treated for malaria.

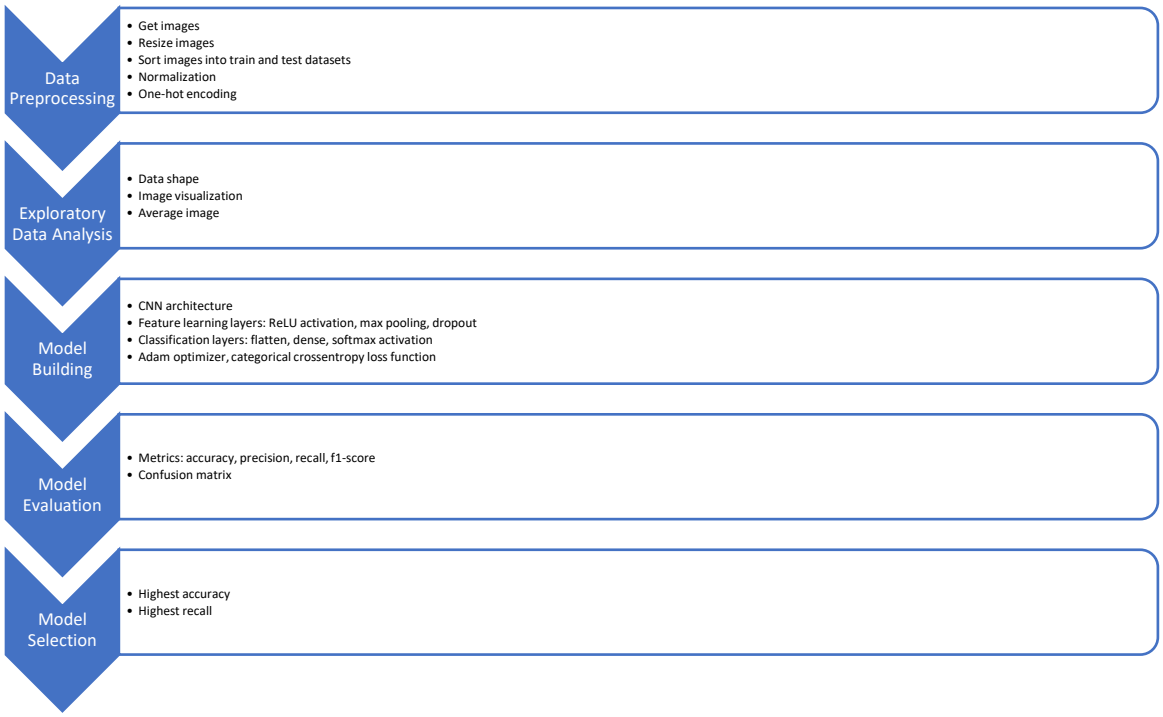
Model 4 performed well, though more work can be done to increase accuracy and recall. Data augmentation can be utilized to generate more images for feature learning. Also, images with more pixels can be utilized for feature learning if there are additional computing resources.

PROBLEM AND SOLUTION SUMMARY

The objective of this project is to build an efficient computer vision model that can help with the early and accurate diagnosis of malaria by identifying whether the image of an RBC is that of one infected with malaria and classifying as parasitized or not infected and classifying as uninfected.

There were five steps to the final solution design: (1) data preprocessing, (2) exploratory data analysis, (3) model building, (4) model evaluation and (5) model selection.

Figure 1: Solution Design



The dataset consisted of 27,558 images of varying sizes. The images were resized into 64 x 64 pixels. The images were divided into train and test datasets which were subsequently divided into two classes, uninfected and parasitized. The images were normalized and the labels were one-hot encoded.

Exploratory data analysis revealed the train image dataset contained 24,958 color images and the test dataset contained 2,600 color images. The images were equally divided into two classes, uninfected and parasitized. Visualization of the images show singular RBCs. Uninfected class, “normal”, RBCs show no abnormalities within the cell and parasitized class RBCs show some abnormalities within the cell. Figure 2 shows random images in the train image dataset. Figure 3 shows the average image of the uninfected class, “normal’, in the train image dataset. Figure 4 shows the average image of the parasitized class in the train image dataset.

Figure 2: Train Image Dataset

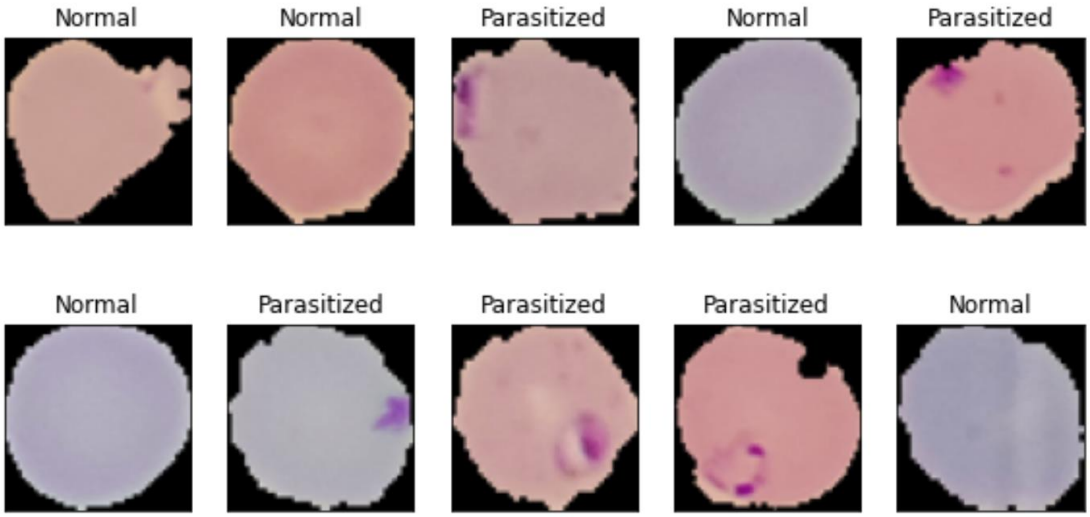


Figure 3: Average Uninfected Image

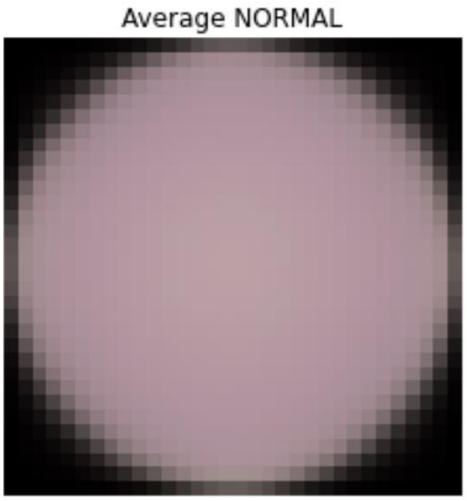
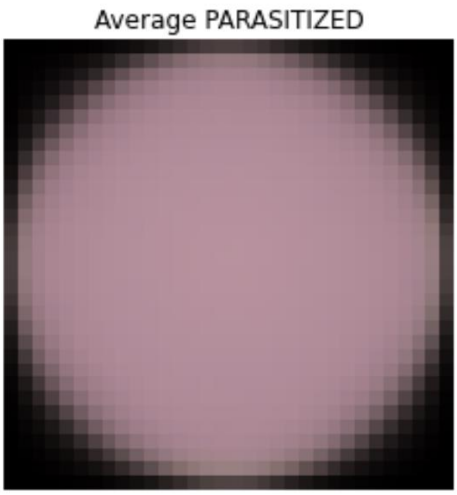


Figure 4: Average Parasitized Image



Seven different CNN models were built using TensorFlow Keras Sequential layers. Model 1 was the baseline model with 2 CNN layers. Model 2 added dropout layers to address overfitting. Model 3 added 2 additional CNN layers. Model 4 added max pooling layers. Model 5 added a cropping and padding layer with an additional CNN layer. Model 6 utilized Keras tuner to try and optimize the hyperparameters from Model 4. Model 7 utilized transfer learning with the VGG-16 pretrained model. The ReLU activation function was used in feature learning and softmax activation function was used in the final output binary classification layer. Categorical cross-entropy was used for the loss function. 80% - 90% of the train image dataset was used for training and 10% - 20% of the train image dataset was used for validation. Batch sizes ranged from 32 to 100 and epochs was set to 20. Comparison of model train accuracy, validation accuracy, train loss and validation loss are depicted in the following graphs, Figure 5 – Figure 8.

Figure 5: Train Accuracy

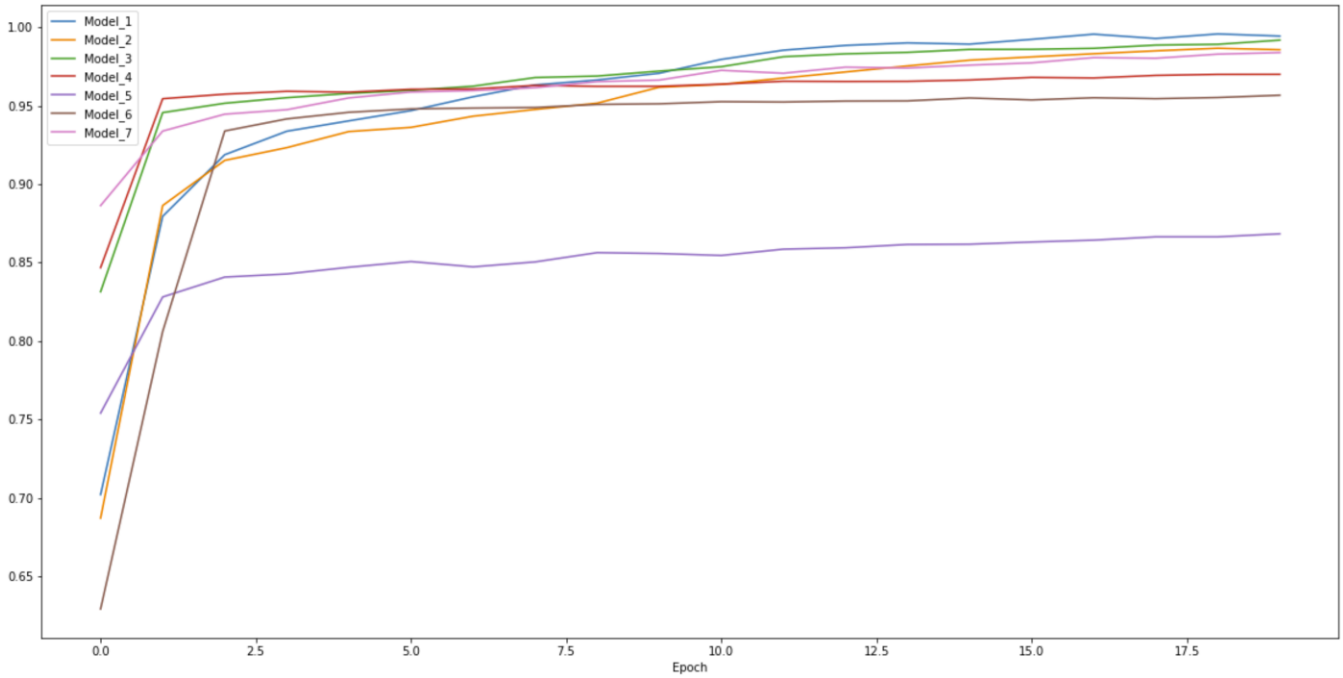


Figure 6: Validation Accuracy

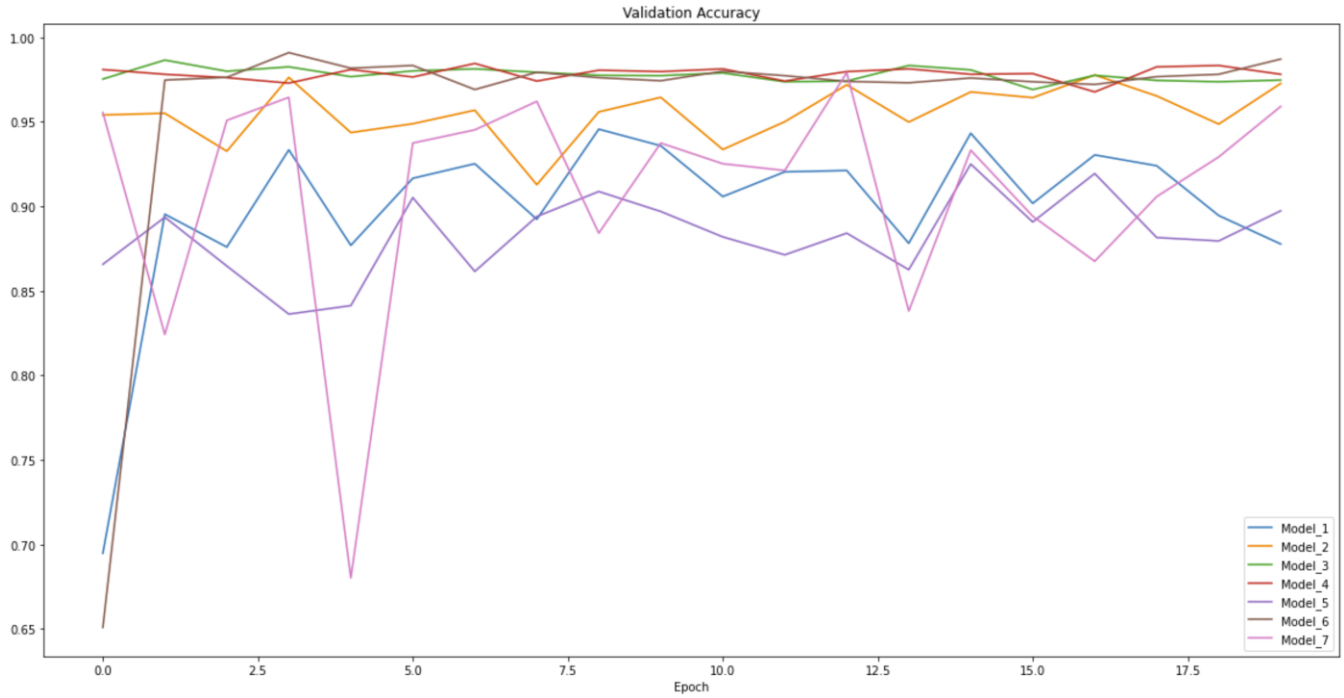


Figure 7: Train Loss

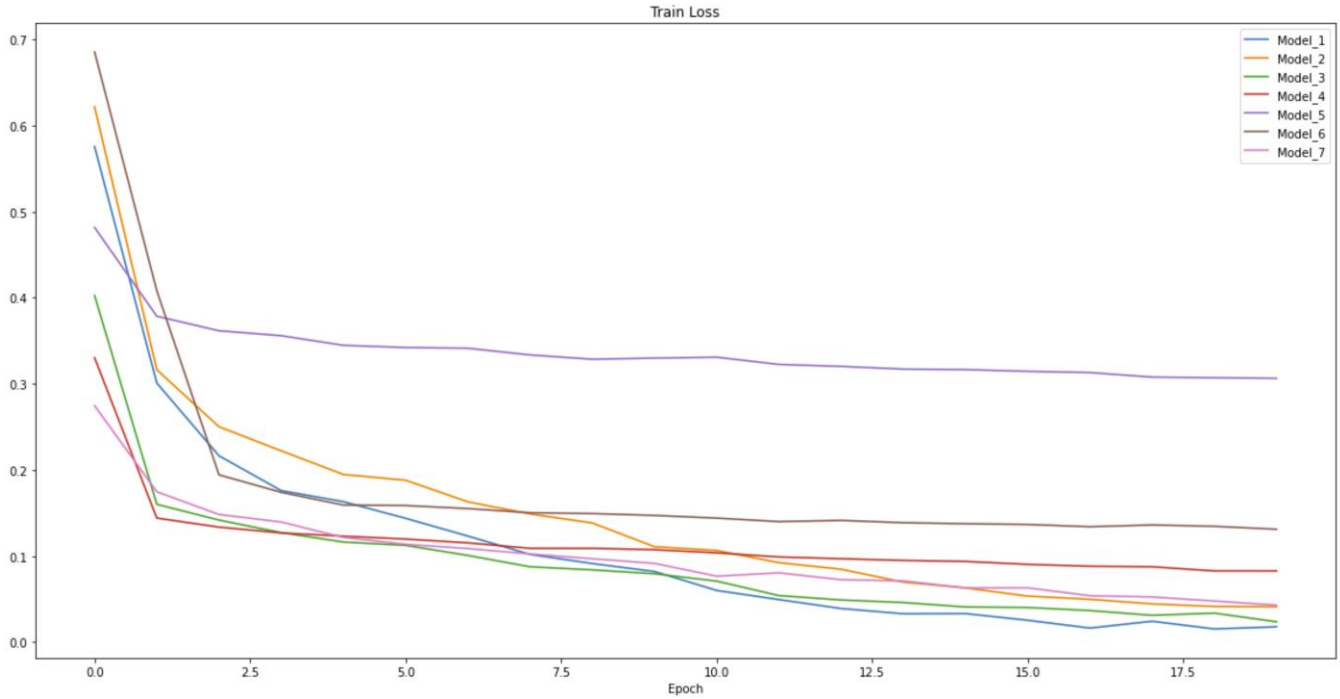
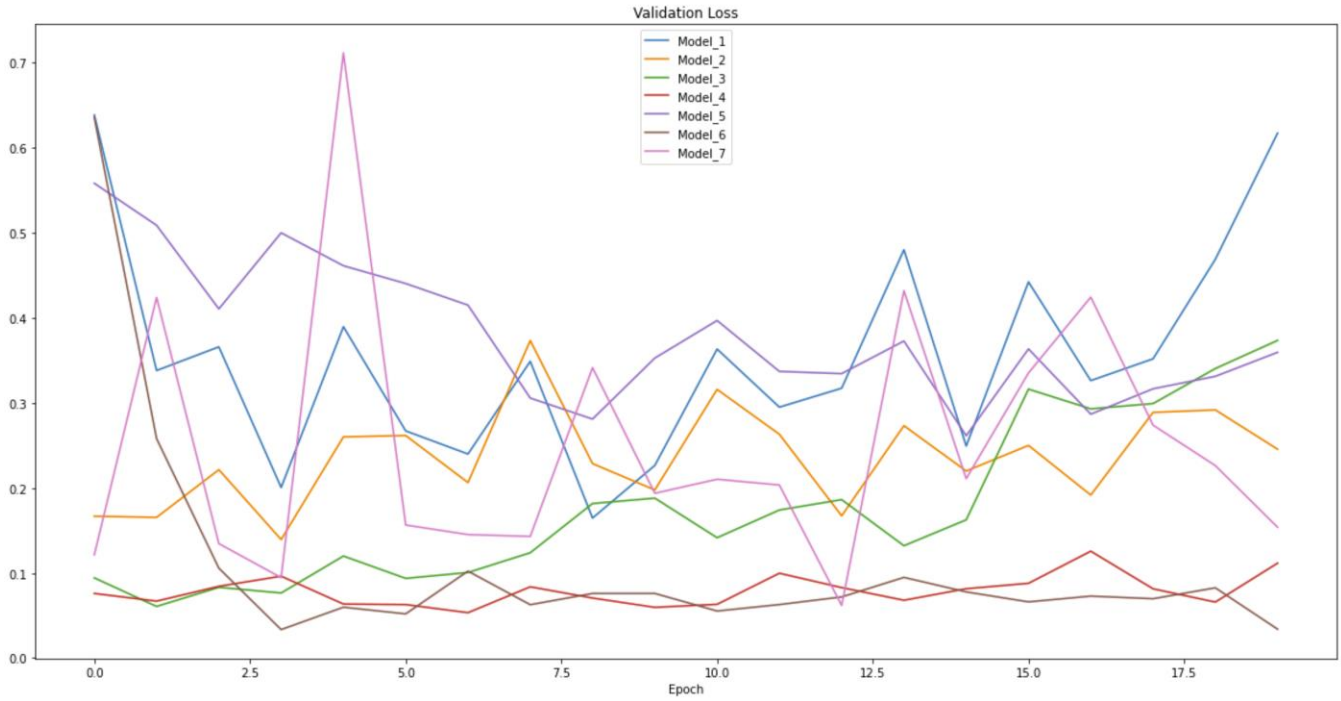


Figure 8: Validation Loss



The models were evaluated using the 2,600 test images. Performance of the models were good with numbers in the high 80s and into the 90s. The number of CNN layers and performance of the different models are tabulated below in Table 1. The confusion matrix results are tabulated below in Table 2.

Table 1: Model Performance Metrics

			Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
		CNN Layers	2	2	4	4	5	3	VGG-16
		Accuracy	0.91	0.93	0.94	0.95	0.88	0.94	0.95
Class	Uninfected	Precision	0.94	0.91	0.92	0.94	0.86	0.91	0.93
		Recall	0.87	0.97	0.97	0.97	0.9	0.98	0.96
		f1-score	0.9	0.94	0.95	0.96	0.88	0.94	0.95
	Parasitized	Precision	0.88	0.97	0.97	0.97	0.89	0.98	0.96
		Recall	0.94	0.9	0.92	0.94	0.85	0.9	0.93
		f1-score	0.91	0.93	0.94	0.95	0.87	0.94	0.95

Table 2: Confusion Matrix Results

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
True Negative (TN)	1134	1259	1265	1266	1168	1279	1249
False Positive (FP)	166	41	35	34	132	21	51
True Positive (TP)	1224	1171	1190	1217	1109	1172	1211
False Negative (FN)	76	129	110	83	191	128	89

Of all the seven different models, Model 4 was chosen as the best of all the models based on having the highest accuracy 95% and the highest recall 94% for the parasitized class. Model 4 architecture is shown below in Figure 9. Emphasis was placed on recall for the parasitized class, also known as true positive rate (TPR), due to the high risk of misclassification. A false negative can ultimately lead to the death of a patient if the patient is not treated for malaria. There were 83 false negatives out of the 1,300 images of the parasitized class. The model was able to correctly classify 1,217 images of the parasitized class. The model misclassified a total of 117 images out of the 2,600 images in the test dataset. Figure 10 is the confusion matrix for Model 4. Figure 11 shows the false negative images classified by Model 4.

Figure 9: Model 4 Architecture

```
model_4.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 64, 64, 16)        448
_________________________________________________________________
leaky_re_lu (LeakyReLU)      (None, 64, 64, 16)        0
_________________________________________________________________
dropout (Dropout)            (None, 64, 64, 16)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 64, 64, 32)        4640
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 64, 64, 32)        0
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 32, 32, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 32, 64)        18496
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 32, 32, 64)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 16, 16, 64)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 16, 16, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 16, 16, 32)        18464
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 16, 16, 32)        0
_________________________________________________________________
dropout_3 (Dropout)          (None, 16, 16, 32)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 8, 8, 32)          0
_________________________________________________________________
flatten (Flatten)            (None, 2048)              0
_________________________________________________________________
dense (Dense)                (None, 32)                65568
_________________________________________________________________
dropout_4 (Dropout)          (None, 32)                0
_________________________________________________________________
leaky_re_lu_4 (LeakyReLU)    (None, 32)                0
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 66
=================================================================
Total params: 107,682
Trainable params: 107,682
Non-trainable params: 0
```

Figure 10: Model 4 Confusion Matrix

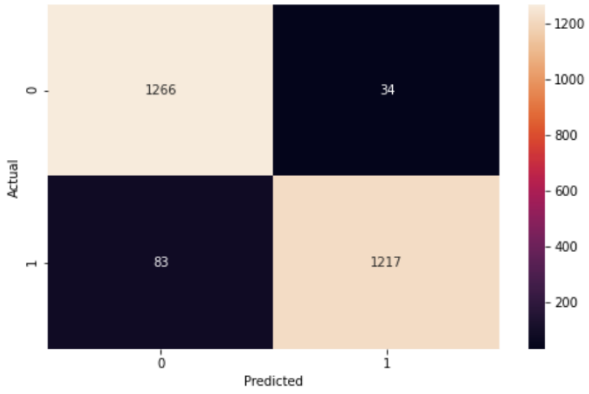
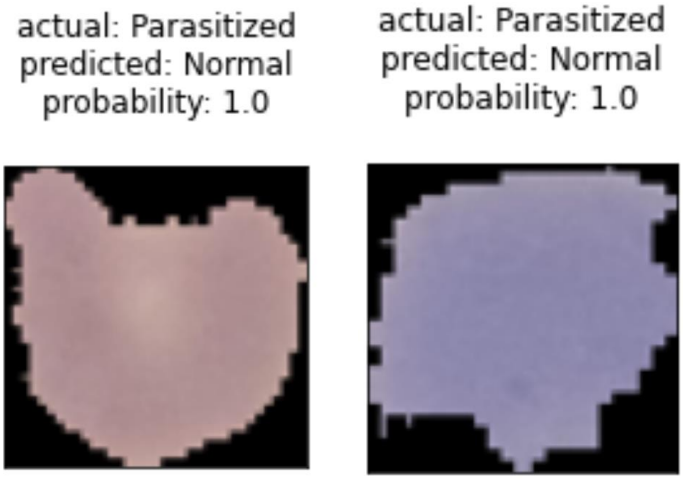


Figure 11: Model 4 False Negative Images



RECOMMENDATIONS FOR IMPLEMENTATION

The ideal scenario is to have a computer vision system that will accurately diagnose malaria and eliminate the need for lab personnel to examine blood smear samples for possible infection. Computer vision system classification ideally would trigger additional tests if classified uninfected or treatment of the patient for malaria if classified parasitized. At this time Model 4 is the best performing model for use in the computer vision system. A more robust model can be built with the availability of additional computational power, which will allow for data augmentation and images resized with larger pixels. Currently, images must be in color and 64 x 64 pixels before the model is applied.

The expected benefits of implementation are reduction in diagnosis time, improvement in diagnosis accuracy for labs with inexperienced lab personnel, possible elimination of lab personnel and most importantly reduction of the number of deaths from malaria worldwide. Reduction in diagnosis time will lessen the delays in diagnosis and allow for earlier patient treatment. Improved accuracy will lower the deaths from malaria due to misdiagnosis. Implementation in a lab setting would alleviate the need to train inexperienced lab personnel in proper identification of parasitized RBCs. Implementation in rural areas of developing countries would alleviate the need for lab personnel to examine blood smears. Cost avoidance for lab personnel training and hiring lab personnel is dependent on the country. In the U.S. the average salary for a lab technician is $36,710 and the average salary of a pathologist is $283,900.

Mitigating steps should be applied as this model is not 100% accurate. This is critical for reducing the high risk of a false negative. A false negative could mean the death of a patient and a false positive can result in unnecessary treatment and associated costs.

Figure 12: Ideal Scenario