Capstone Project – Malaria Detection: Final Report

Executive Summary

Exploratory data analysis of images was first performed to provide the initial exploration to gain insight information about parasitized and uninfected red blood cell images in both train and test datasets. The visualization of the original image from both datasets shows that they are many different colors of red blood cell images that might come from different laboratory and different people conducting the experiment. Red blood cell images that are labeled as parasitized have the purple color spot while the uninfected one did not have the color spot. Three image data preprocessing techniques consisting of **HSV** (hue, saturation, value), **Gaussian Blur** and **Data Augmentation** were created to be used as the training image in the neural network training step. **HSV** showed a clearer color image to human eyes and more robust to lighting change than RBG color. The **Gaussian blur** is reducing the noise of image and smoothed out the edge of an image. The **Data Augmentation** method is the method that create diversity of images without collecting new data by applying rotation, zoom, shear, horizontal flip, vertical flip, and many more on the images. These techniques have been studied, in some cases, to reduce error and provide higher accuracy for the training step using neural network.

The performance of ten different Convolutional Neural Networks (CNNs) were examined by studying the model accuracy on training, validation, and testing sets. The confusion matrix, precision, recall and F1-score are also studied to help decide on which CNN models are the best in detecting parasitized and uninfected red blood cell images. The **Model 6** was studied to be the best because of the highest accuracy on testing set with 98.57 %, low false negative (FN) error and the highest % recall leading to less error of model in detecting the images as uninfected instead of actual parasitized. This model has three convolutional layers, used Leaky ReLu as activation function, and used bath normalization trained on Data Augmentation images. Different layers, activation types, different input image preprocessing, and transfer learning from different pre-trained model of creating CNN models significantly affect the accuracy of malaria detection. The accuracy can also be improved by exploring more CNN models using different data preprocessing technique, changing transform parameters in Data Augmentation, using different number of neurons on each layer, adding bath normalization in between each layer, and using different pre-trained models for transfer learning technique.

Problem Statement

Malaria is one of contagious disease that is serious and sometimes can cause death and it is still a burden to global health. The cause of malaria is from the plasmodium parasites which transmit to human through the bite of female *Anopheles* mosquitoes. These parasites can damage the red blood cells (RBCs) that carry the oxygen. The symptoms of fever, headache and chills usually appear 10-15 days after the infective mosquito bite, which can be difficult to detect, and some parasites can cause deaths in 24 hours.

According to the latest report from World Health Organization (WHO),[1] In 2020, 50% of the world's population was at risk from malaria. There were also an estimated of 241 million cases of malaria worldwide and 627,000 related deaths reported in 2020. Children under 5 years old are the most vulnerable group of people.

Early state of malaria detection is needed, and traditional diagnosis needs experienced professional to manually detect whether the red blood cell images are parasitized or uninfected and it is also time consuming. Therefore, deep learning has played a key role in increasing detection time and removing human errors in malaria detection. Several deep learning models have been studied and developed. One of the most used deep learning models in image detection and computer vision is CNN. There are several ways and techniques of creating CNN architectures and they needed to be studied carefully. One CNN model might do better in one problem but might do worst in other problems. Therefore, in this project, several CNN models are studied and several metrics to measure accuracy of different CNN models are calculated.

Objective

For this project, several computer vision models are built to detect malaria by identifying the images of red blood cells if they are from the infected with malaria or not, which are identified as parasitized and uninfected, respectively. In this final report, the initial data exploration, and data preprocessing techniques are explored to gain insight into data and were used as trained image in the CNN training step to find the best model for malaria detection.

Solutions Designs

The images of parasitized and uninfected in both train and test datasets are first visualized to see how they are different from each other. All the images are converted to the same size and convert to 4-dimensional array for further analysis using CNN. The images are labeled as 1 for the parasitized red blood cells and 0 for uninfected red blood cells. The number of parasitized and uninfected images were counted to see if the data is balance or not. The data is balanced as we can see from the uniformly distributed of the number of images between parasitized and uninfected in both testing and training set that are shown in **Table1** and **Figure 1**.

Table 1: The number of parasitized and uninfected images in training and testing sets

Datasets	**Number of images**
Training set	**24958**
Parasitized	12582
Uninfected	12376
Testing set	**2600**
Parasitized	1300
Uninfected	1300

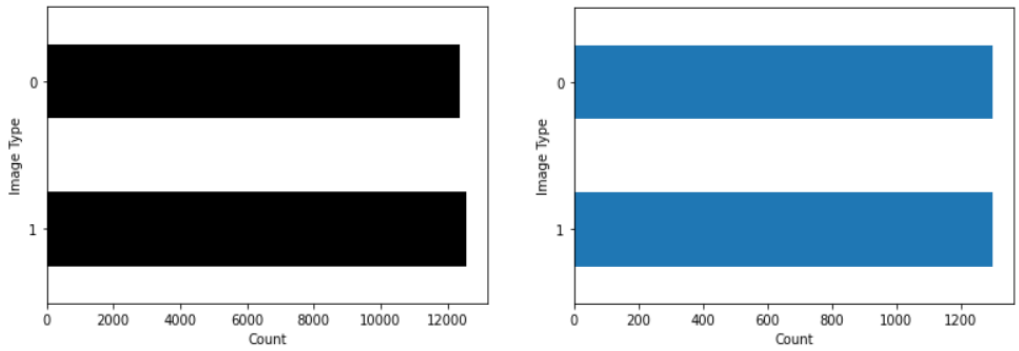

Figure 1. The count plot of the parasitized images (label as 1) and uninfected images (label as 0) in training set (left) and in testing set (right).

Before creating deep learning model to train data, the train and test images are normalized to ensure that each input pixel has similar data distribution. The one-hot encoding was applied to original data in both train labels and test labels. Ten computer vision models with different neural network architectures were studied.

The first model (**model1**) was built by artificial neural network (ANN) with only dense or fully connected layers are added with ReLu as an activation function. **Model2** was built on CNN model with three convolutional layers using ReLu as activation function, followed by max pooling and dropout layer in each of convolutional layer (Conv2D) in feature extraction part of CNN and flatten in feature maps and two dense layers in classification part of CNN. For the last dense layer or the output, softmax was used as activation function to transform the raw outputs of neural network into a vector of probabilities. The first two models have different number of filter (neurons) in each layer in feature extraction part. The first, two and three layers have 32, 64, and 128 neurons, respectively. For **model3**, the same number of filters with 32 neurons in all convolutional layer were studied to see if the accuracy will increase or decrease compared to **model2**. The number of parameters that are trained in **model3** are less than **model2**. To see whether the more convolutional layers will increase the model accuracy, **model4** was studied with one more Conv2D layer added onto **model2**. So far, we only studied ReLu activation function. Therefore, **model5** was studied by using Leaky ReLu as activation function and added the bath normalization layer before the flatten layer in **model2**. Batch normalization can solve the internal covariate shift and standardize the inputs variables or the output hidden layer. Different models were built to train images from different image preprocessing techniques, Data Augmentation for **model6**, HSV for **model7**, and Gaussian blur for **model8**. **Model9** and **model10** were built using transfer learning technique from existing model or pre-trained model called **VGG-16** and **ResNet-50** with 16 and 50 layers deep, respectively.[2] All models were trained with 20 epochs and the best epoch of each model was save using model checkpoint. The callback with early stopping is also used in each model to stop the training when validation loss of each model did not improve after a certain number of epochs by setting the keyword called “patience”. If patience equal to 2, the model will stope if there is no improvement in validation loss for two consecutive epochs.

The performance of each model is evaluated by training, validation and testing accuracy, precision, recall, F1-score, and confusion matrix. Precision, recall and F1-score are shown in equation (1), (2) and (3), respectively.

$$\text{Precision} = \frac{TP}{TP+FP} \tag{1}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{2}$$

$$\text{F1-score} = 2\ x\ \frac{Precion\ x\ Recall}{Precison+Recall} \tag{3}$$

Analysis and Discussion

The original red blood cell images are shown in **Figure 2**. All parasitized images show the purple color spot on red blood cell while all uninfected images do not have purple spot. We can see that each image has different red blood cell colors, which might come from different laboratory or different set of experiments. The average mean of parasitized and uninfected images is also plotted. They are very similar, which make it harder to distinguish them from the mean images. These average images can be used when deciding which images in the data are closer to parasitized or closer to uninfected by calculating the mean square distance between the image of interest to the mean of parasitized or uninfected image. Images from **HSV**,[3] **Gaussian blur**,[4] and **Data Augmentation**[5] preprocessing techniques using in the model training part in this project are shown in **Figure3**.

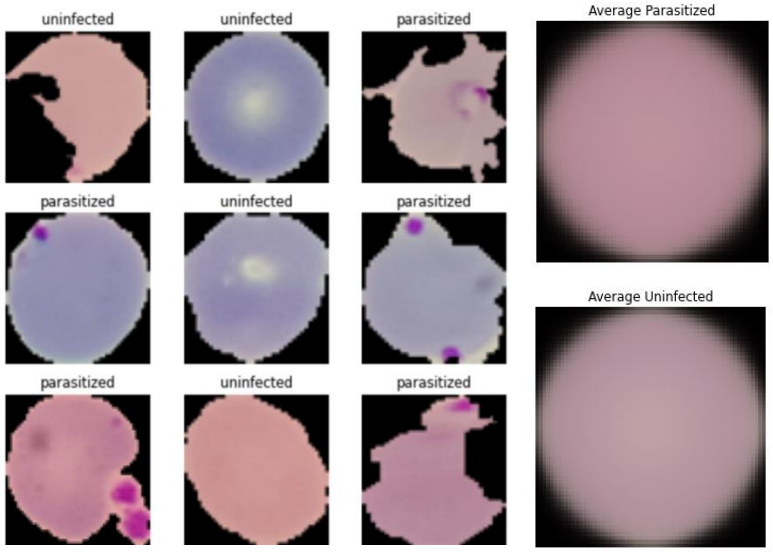

Figure 2. the original parasitized and uninfected images (left) and the mean of parasitized and uninfected image (right).

From the observation of HSV images with has label of 1 (parasitized), the red blood cell color of all images in both train and test sets are pink, which make them easier to be compared. The HSV is an alternative representation of RBG, and it is more robust toward a lighting change and used it when the color description plays a key role. Therefore, **HSV** might be preferred in many studies. The yellow spots can be seen in all images while some images, blue spots are present. These blue spots might just be the impurities. For the **Gaussian Blur**, it is using gaussian function to remove noise level and make the edge of the image smoother. The smoother edge means the images are smoothly changed between one side of an edge to another. The images are blurrier, and the edge are smoother compared to original images in **Figure 2.** This technique is use when reducing the size of an image because it reduces the amount of data in an image while preserving the structural properties of an image. I think this technique can be used in the pulling step of CNN, where the size of image is reduced. **Gaussian Blurring** is commonly used to avoid false edges detection and can be used to reduce the time in training using machine learning model.

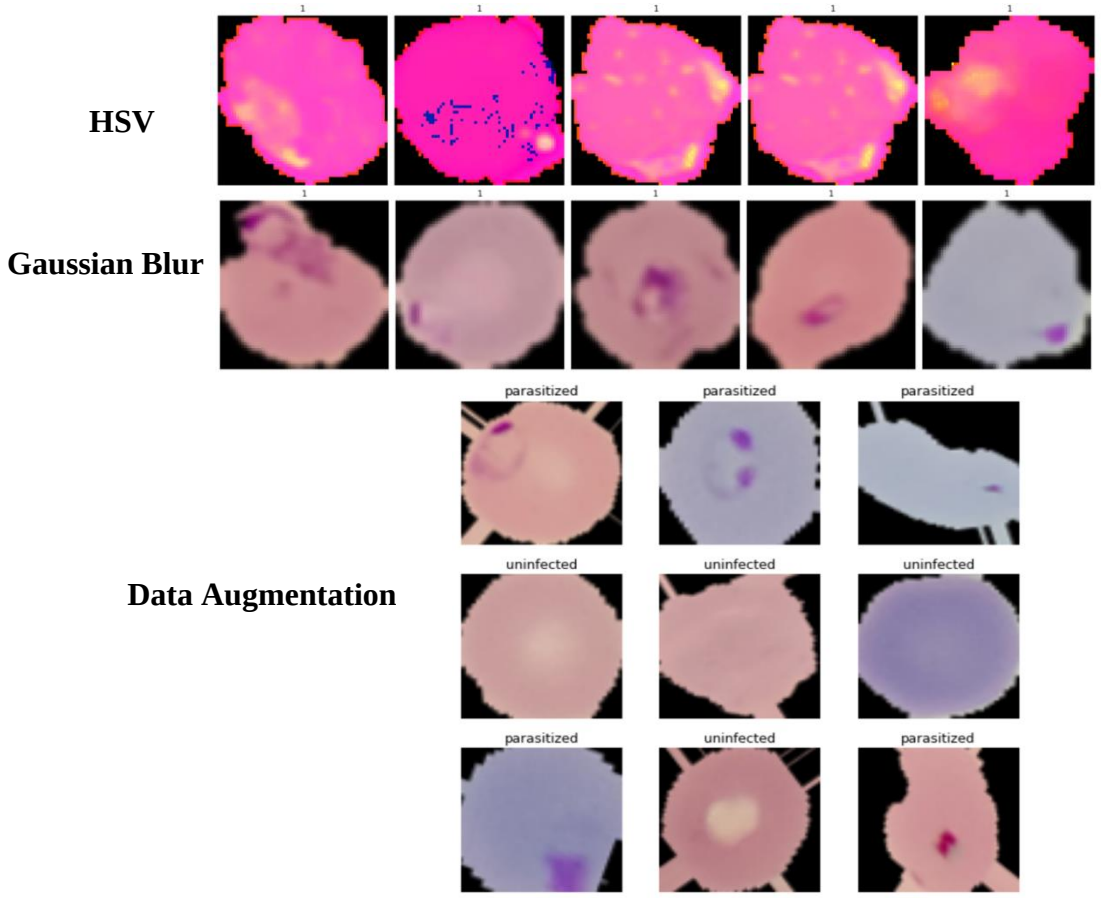

Figure 3. Training images from **HSV**, **Gaussian Blur**, and **Data Augmentation**.

The second part of the project is building computer vision models using neural networks. Ten models with different details are explained in the solution design part. The training, validation, and testing accuracy from the lowest validation loss of all 10 models are shown in **Table 2**. All models have different architectures in the feature learning part, which described in **Description** column. For the classification part, all models contain flatten, dense layer with 256 neurons,

dropout layer, and the output layer with dense layer of 2 neurons. First, focusing on testing accuracy, **Model2, 5** and **6** highlighted in blue fonts give the best results for the comparison of the model accuracy on testing images with 98.08%, 98.38%, and 98.58%, respectively. If we investigate more details of several models, we can see that using the same number of neurons (filter = 32) in **model 3** did not improve the testing accuracy compared to **model2** with different number of neurons in each Conv2D. **Model4** with more Conv2D layers also did not improve the testing accuracy compared to **model2**. Therefore, **model2** was used as the base model for **model5**. **Model5** with bath normalization layer added before classification part and with Leaky ReLu as activation function shows the improvement in testing accuracy from 98.08% in **model2** to 98.38%. With this result, **model5** was used as base model for **model6** to **model10**. **Model6** with **model5** trained on Data Augmentation images give the best accuracy on testing data of 98.58%. The other two models (**Model7**, and **Model8**) were trained on **HSV** and **Gaussian Blur** images, respectively and their accuracy on testing set are not improving from **model6**. **Model9** and **model10** used transfer learning from pre-trained model **VGG-16** and **ResNet50**, respectively also did not improve the testing accuracy. **Model10** with **ResNet-50** provides the lowest accuracy among all models. **ResNet-50** might not be an appropriate pre-trained model using in transfer learning for malaria detection case.

Table 2. The percent accuracy of training, validation, and testing sets from the lowest validation loss model

Model	Description	Training accuracy (%)	Validation accuracy (%)	Testing accuracy (%)
Model1	ANN with only dense layers	95.72	97.54	93.00
Model2	CNN (3 Conv2D layers) with **different** number of neurons	97.92	98.50	98.08
Model3	CNN (3 Conv2D layers) with **same** number of neurons	97.10	99.04	97.92
Model4	CNN (4 Conv2D layers) with **different** number of neurons	88.66	98.92	97.30
Model5	**Model2** with batch normalization and change ReLu to Leaky ReLu	97.41	98.70	98.38
Model6	**Model5** on Data Augmentation	97.38	98.00	98.58
Model7	**Model5** on HSV images	94.89	98.66	96.46
Model8	**Model5** on Gaussian blur image	97.79	99.46	97.53
Model9	VGG-16	95.24	98.54	95.58
Model10	ResNet-50	73.24	92.85	61.85

*__Note__: after each Conv2D, max pooling layer and dropout layer are added.

If we consider the effect of training and validation accuracy on the testing accuracy, higher accuracy from both could provide higher accuracy of model when study the testing set such as **model 2,5,** and **6**. However, both accuracy from training and validation set need to be high and close to each other based on the observation from **Table 2**. The highest accuracy in validation set or the highest accuracy from training set alone do not confirm that model can do the best for testing

set. For example, **model8** with the highest validation accuracy of 99.46% does not provide the highest testing accuracy because it is also a lot higher than training accuracy. **Model2** with the highest training accuracy of 97.92% also does not provide the highest testing accuracy. **Model6** with the highest testing accuracy has the closet accuracy between training and validation set as shown in the table and in the accuracy plot in **Figure 4**. This figure only shows the three best models that give the best testing accuracy. All models seem to have slightly higher validation plot than training plot, which might cause by the dropout layers that are added to reduce overfitting of the model making the accuracy of training set lower and the model might train better on the images of validation set making validation accuracy higher. **Model2'** validation accuracy is not as smooth as other two models. **Model6** shows the closest accuracy between validation and training, which might greatly enhance the accuracy of model on the test data.

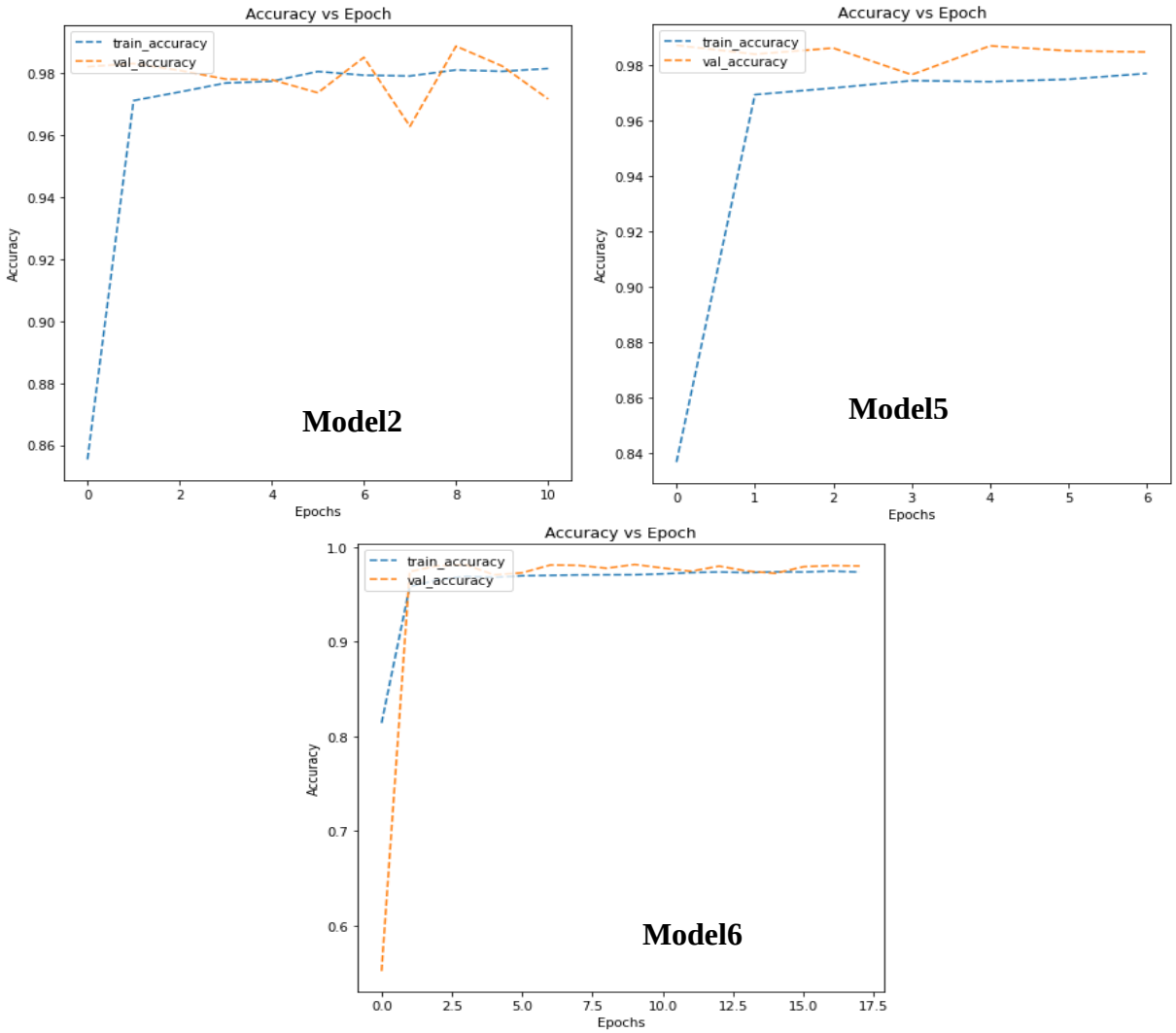

Figure 4. Plot of training accuracy vs validation accuracy of **Model2**, **Model5**, and **Model6**.

The accuracy from training, validation and testing set are not enough, other metrics need to also be considered in selecting which models are the best for using in malaria detection. These

metrics are classification report on precision, recall and F1-score, and confusion matrix showing how many images model predict parasitized and uninfected red blood cells correctly. The confusion matrix of three best models is shown in **Figure5**. Before explaining the results from confusion matrix, I would like to define what is negative and what is positive results. Negative is the uninfected image while positive is parasitized image. True positive (TP) is the term when the model correctly predicts parasitized image while true negative (TN) is the term when the model correctly predicts uninfected image. False positive (FP) is error term when model predict images as parasitized, but they are uninfected. The last term is false negative (FN) that identify model error detecting images as uninfected, but they are parasitized. FN is more concern because we would like to correctly detect patient who actually infected by malaria to provide fast and right treatment. It might cause death or more harm to people who are parasitized but are predicted as uninfected because they might not get treatment in time to get better. FP is also important but might be less concern; however, patient might be allergic to medicine when they did not actually get infected.

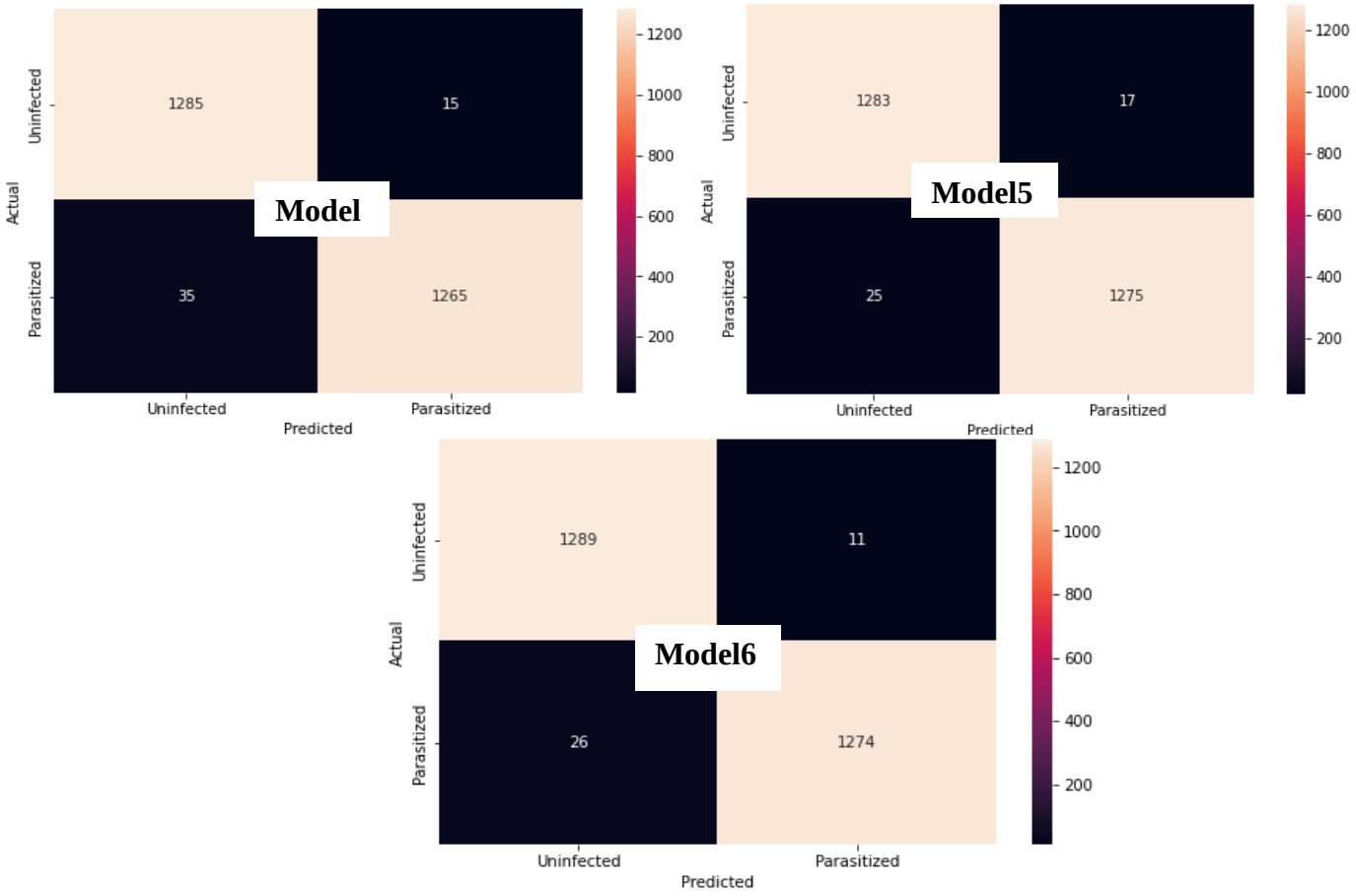

Figure 5. Confusion matrix of three best computer vison models; **Model2**, **Model5**, and **Model6**.

From confusion matrix, **model5** and **model6** clearly are better than **model2** for the comparison of error type FN. If FN is more concern, **model5** is better due to a smaller number of

FN than **model6**, but if the FP is more concern, **model6** and **model2** are better than **model5** due to a smaller number of FP. However, if we look at both **FN** and **FP**, **model6** would consider to be the best. To conclude which one is better between **model6** and **model 5**, the concern of error should be considered. If patient who wrongly detect in FP are not allergic to medicine and are less concern, **model5** would consider to be better even though it has one less image that are wrongly predict as uninfected than **model6** in FN error, but to protect at least one more patient to not be death is very important.

The precision, recall and F1-score are also calculated from the confusion matrix. Precision describes what proportion of positive identifications was correct while recall describes what proportion of actual positives was identified correctly. F1-score combines the precision and recall interpretating as the harmonic mean of precision and recall. If FN is more concern, % recall should be higher. From **Table 3**, % recall of **model5** and **model6** are the best with 98% recall corresponding from being the best in testing accuracy. If both FN and FP are concern, **model6** is the best due to the highest F1-score.

Table 3. The classification report of precision, recall and F1-score.

Model	**Precision (%)**		**Recall (%)**		**F1-score (%)**	
	Uninfected (0)	**Parasitized (1)**	**Uninfected (0)**	**Parasitized (1)**	**Uninfected (0)**	**Parasitized (1)**
Model1	91	95	96	90	93	93
Model2	97	99	99	97	98	98
Model3	97	99	99	97	98	98
Model4	96	99	99	96	97	97
Model5	98	99	99	98	98	98
Model6	98	99	99	98	99	99
Model7	94	99	99	94	97	96
Model8	96	100	100	96	98	97
Model9	94	97	97	94	96	96
Model10	58	78	91	33	70	46

Conclusion

The two best CNN architectures that should be selected to use in malaria detection from this study are **Model5** and **Model6** with the testing accuracy of 98.38%, and 98.58%, respectively. If false negative (FN) are very important and are more concerned, **model5** would do better due to less image are detect as uninfected even though it should be detected as parasitized. However, **model6**, if we consider % recall, we can see that it shows the equal % compared to **model5** and **model6** also show the best % precision leading to the best % F1-score. Therefore, **model6** is the best for this study and should be used in malaria detection when both error FN and FP are considered.

Recommendation and Future Improvement

The study of classification between parasitized and uninfected red blood cells by computer vision models should be in an early state of malaria detection (screening process) because we need to make sure for those people who are predicted as parasitized even though they are uninfected do not get allergic to any malaria treatment medicines and doctor can screen for those who are predicted as uninfected do not actually have parasitized red blood cell.

Exploration of other deep learning models detecting malaria from red blood cell images should still be study to improve accuracy and reduce error even more. Even though we did not get a good performance from pre-trained VGG-16 and ResNet-50 model, more transfer learning models from other pre-trained model such as AlexNet, LeNet, GoogLeNet, Xception, and DenseNet-121[6] can be considered. For VGG-16 and ResNet-50, we can use more trainable layers (not just the last layer) added in feature map and we can try different classification model to see if the model performance can be improved. For the best model like **model6** that are train on Data Augmentation images can also be improved by playing with more transform parameters using ImageDataGenerator. Other modifications such as more bath normalization layers between each convolutional layer, different number of neurons in each layer, and less or more dropout layers can also be changed to improve model performance. Another thing to try is using Data Augmentation on HSV and Gaussian Blur images before training with neural networks.

References

[1] World Health Organization (2022, April 6). **Malaria**. https://www.who.int/news-room/fact-sheets/detail/malaria#:~:text=According%20to%20the%20latest%20World,deaths%20over%20the%20previous%20year.

[2] K. Simonyan, A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. ICLR, 2015

[3] JeefriAMoke (2017, October 16). **Image Classification with HSV Color Model Processing**. https://www.datasciencecentral.com/image-classification-with-hsv-color-model-processing/

[4] Digital Image Processing (2005). **Gaussian Blur the Image to Reduce the Amount of Noise and Remove spackles within the image**. https://www.sciencedirect.com/topics/engineering/gaussian-blur

[5] Arun Gandhi (2020). Data Augmentation | How to use Deep Learning when you have Limited Data - Part2 https://nanonets.com/blog/data-augmentation-how-to-use-deep-learning-when-you-have-limited-data-part-2/

[6] S. Rajaraman, S. K. Antani, et al. pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. The Open Access Journal for Life & Environmental Research.