Detection of Malaria in Red Blood Cells using Deep Learning

Abhinav Ram Mohan

MIT Applied Data Science Program

December 2022

Executive Summary:

Disease Overview and Diagnostics:

Malaria is a parasitic disease caused through transmission by mosquito bites in developing nations. The parasite of interest is plasmodium. In 2019 alone, there were more than 229M cases of malaria with 400K reported deaths globally. As a blood-borne disease, the diagnosis is typically by blood smear testing and manual microscopic evaluation for parasitic infection. Hence, an automated method would alleviate overhead from having to perform a task that can become tedious quickly.

Key Methodologies and Evaluation Metrics:

A machine learning model utilizing a convolutional neural network was able to automate the binary classification of infected and uninfected images of histologically stained blood cells. This framework and model utilized image pre-processing and data augmentation to feed the model. Callbacks were deployed to prevent model overfitting and the model accuracy and loss were evaluated, along with metrics such as precision, recall and f1-score as the metrics of success for the model. The classification matrix was plotted as a means of visualizing the outcomes of the classification.

Key Outcomes:

The model boasts a high accuracy and low loss and an f1-score very close to 1.0, indicating a high classification outcome. Out of a total of 2800 images, 15 uninfected images were misclassified as parasitized and 26 parasitized images were misclassified as uninfected. These results give a high degree of confidence for the preliminary deployment of this algorithm for classifying malarial infected cells.

Next Steps:

There is still room for significant improvement in the model evaluation. In terms of the present model, further layering and method and hyperparameter tuning (i.e., adjusting the number of nodes, the activation function, use of batch normalization), and deploying methods such as grid search, random search, and Bayesian optimization for hyperparameter tuning are considerations. Various pre-existing models such as Mask R-CNN, YOLOR and YOLOv7 as the latest in image classification can be looked into for improving image classification. In short, the model can be fine-tuned in various ways. Key stakeholders would be interested in seeing what exactly is being tagged while classifying the images, so an important output would be to show markup images of the infected cells to show that the parasite is what is being tagged (i.e. segmentation). Additionally, physicians and pathologists interested in understanding the gravity of infection would be interested in the parasitic density, which is the amount of parasite per cell(s). This would be an important factor to grade the severity of infection. Furthermore, the blood contains white blood cells and others among red blood cells, so a robust algorithm would need to differentiate these cells out, leaving the focus on red cells. All of these considerations should be taken into account in the future to build a more improved and robust model for evaluating blood samples.

Problem and Solution Summary

Disease Overview:

Malaria is a parasitic disease caused through transmission by mosquito bites in developing nations. The parasite of interest is plasmodium. Complications of malaria include infection of the nervous system as well as respiratory distress, both of which can be fatal if not given immediate attention. In 2019 alone, there were more than 229M cases of malaria with 400K reported deaths globally[1]. As a blood-borne disease, the diagnosis is typically by blood smear testing and microscopic evaluation for parasitic infection, which is the present gold standard for evaluation[2] (Figure 1).

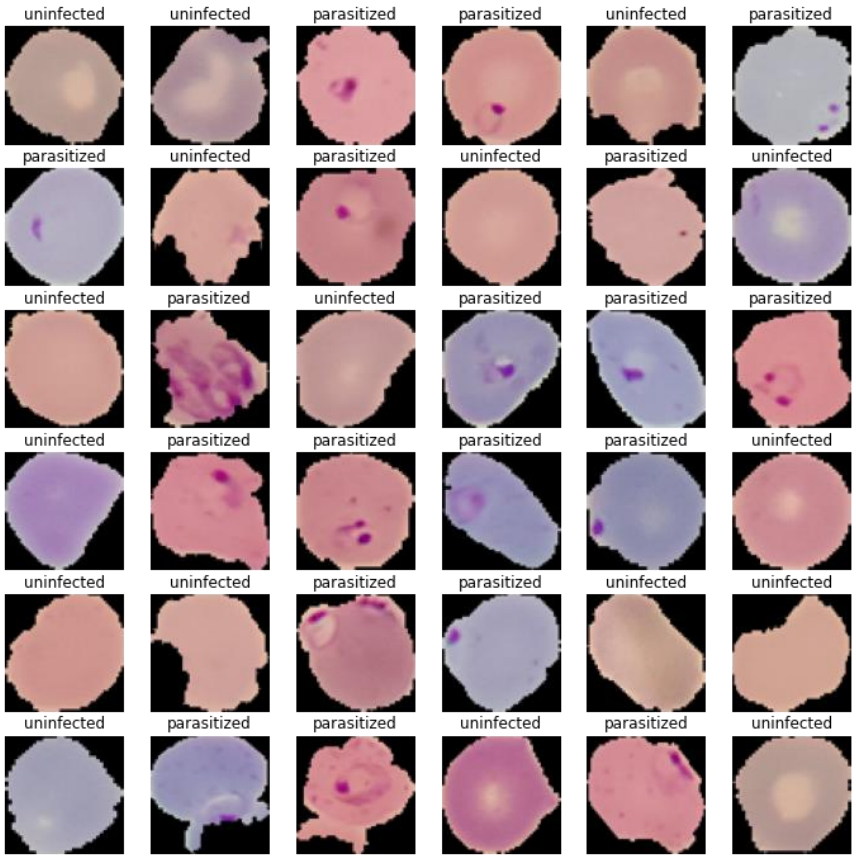

Figure 1: Sample of red blood cells that are Giemsa stained to identify plasmodium infection. Bright pink/magenta concentrations indicate the presence of malarial infection.

Problem Overview:

While other diagnostic tools such as PCR and rapid antigen detection tests exist, the grading of parasitic density requires microscopic evaluation[1]. This is a time-consuming process as it requires an analyst to spend valuable time at the microscope to manually count each cell and estimate the parasitic density. Furthermore, for diagnostic laboratories that rely heavily on hospitals and samples for data and revenue,

[1] UNICEF: Malaria - https://data.unicef.org/topic/child-health/malaria/#:~:text=In%202019%2C%20there%20were%20229,an%20urgent%20public%20health%20priority.

[2] CDC: Malaria Diagnostic Tests - https://www.cdc.gov/malaria/diagnosis_treatment/diagnostic_tools.html#:~:text=The%20gold%20standard%20for%20the,the%20infected%20red%20blood%20cells

having an automated process for evaluating the samples would cut any additional overhead cost required for hiring an analyst and put that money back towards progressing the business.

The algorithm presented here gives insight into how convolutional neural networks (CNNs) can be used to evaluate histologically stained red blood cell images showing malarial infected cells. The methodology and results can be used as a framework to further add on additional analyses such as whole blood analysis, blood cell differentiation, and densitometric evaluation for pathological diseases. This framework could also be expanded for other blood pathogens as well.

Data Preprocessing:

The images presented in the dataset were pre-grouped as infected or uninfected and labels were placed accordingly before the images were all pooled together and randomized for model training and evaluation. Prior to image processing, the dataset was augmented using ImageDataGenerator from TensorFlow to augment each dataset, training and testing by 8%. Before the images were evaluated by the CNN, image preprocessing was done to convert the images from the standard RGB color scheme to HSV, so that there would be normalization of the image saturation and the key feature that would stand out would be the malaria infection. From Figure 1, we see that through just binary classification, there are potentials for errors and misclassification just based on color given that some of the uninfected cells have the same coloration as the parasite infecting the cells. With the HSV conversion, because of the saturation of the parasite within the cells, a distinction can be made (Figure 2).

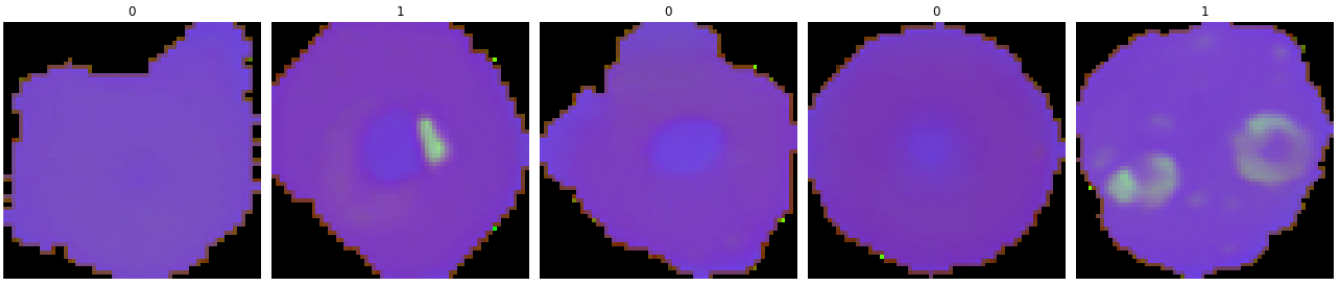

Figure 2: HSV representation of the Giemsa-stained red blood cells. 0: uninfected cells; 1: infected cells. Note how the infected regions appear as green, while the rest of the cell is rendered blue/purple.

The images were also normalized and converted to HSV to determine whether normalization of images would provide benefit towards classifying the plasmodium infected cells; however, due to the enhanced brightness and color scheme of the images (Figure 3), the decision to use the non-normalized, non-standardized images in HSV format was made.

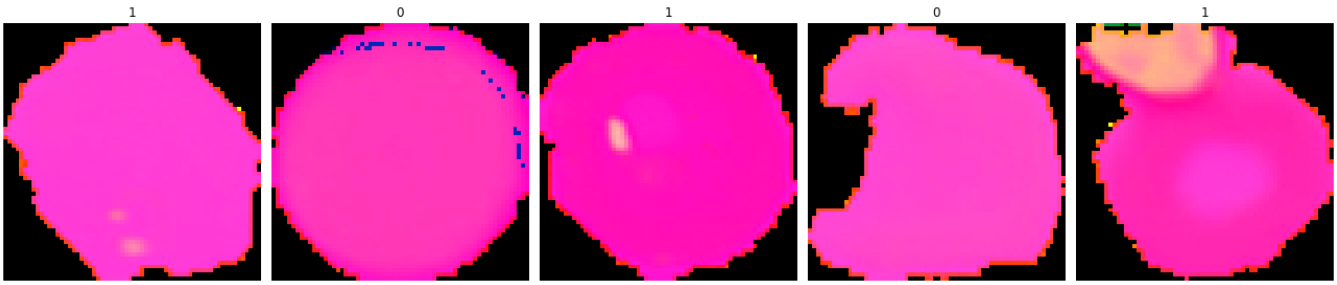

Figure 3: HSV representation of the normalized Giemsa-stained red blood cells. 0: uninfected cells; 1: infected cells. Note how the infected regions appear as yellow, while the rest of the cell is rendered bright pink.

Model Framework and Evaluation Metrics:

A number of CNN architectures were tested as part of the solution design, with differences in number of layers to hyperparameter tuning and also determining the training/test split. The final CNN was built with 5 hidden layers and utilizing the leaky ReLU activation function to perform computations as shown in Figure 4.

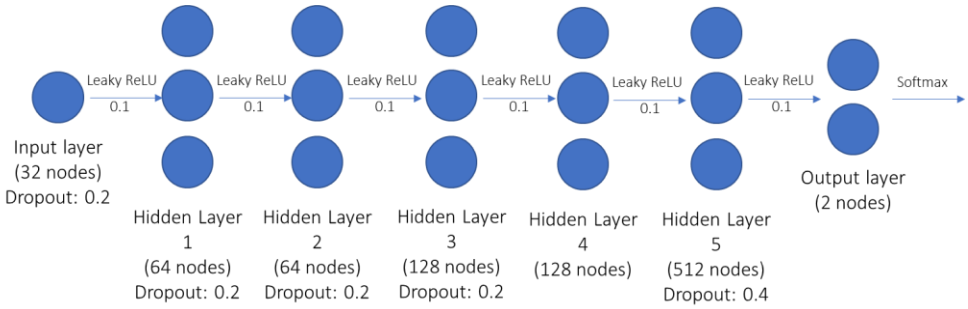

Figure 4: Architecture of the CNN used for binary classification of images of malarial infected cells. Leaky ReLU was the activation function used, with a slope of 0.1 and softmax as the output function.

In this CNN, dropout was used to randomly drop nodes in given layers by random in order to help prevent overfitting of the model. The model was built sequentially with convolution layers and a dense layer towards the end. The number of trainable parameters was determined at each convolutional or dense layer (Figure 5).

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 64, 64, 32)	416
leaky_re_lu (LeakyReLU)	(None, 64, 64, 32)	0
max_pooling2d (MaxPooling2D)	(None, 32, 32, 32)	0
dropout (Dropout)	(None, 32, 32, 32)	0
conv2d_1 (Conv2D)	(None, 32, 32, 64)	8256
leaky_re_lu_1 (LeakyReLU)	(None, 32, 32, 64)	0
max_pooling2d_1 (MaxPooling 2D)	(None, 16, 16, 64)	0
dropout_1 (Dropout)	(None, 16, 16, 64)	0
conv2d_2 (Conv2D)	(None, 16, 16, 64)	16448
leaky_re_lu_2 (LeakyReLU)	(None, 16, 16, 64)	0
max_pooling2d_2 (MaxPooling 2D)	(None, 8, 8, 64)	0
dropout_2 (Dropout)	(None, 8, 8, 64)	0
conv2d_3 (Conv2D)	(None, 8, 8, 128)	32896
leaky_re_lu_3 (LeakyReLU)	(None, 8, 8, 128)	0
max_pooling2d_3 (MaxPooling 2D)	(None, 4, 4, 128)	0
dropout_3 (Dropout)	(None, 4, 4, 128)	0
conv2d_4 (Conv2D)	(None, 4, 4, 128)	65664
leaky_re_lu_4 (LeakyReLU)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 512)	1049088
leaky_re_lu_5 (LeakyReLU)	(None, 512)	0
dropout_4 (Dropout)	(None, 512)	0
dense_1 (Dense)	(None, 2)	1026

Total params: 1,173,794
Trainable params: 1,173,794
Non-trainable params: 0

Figure 5: Architecture description of the CNN used in the classification of plasmodium infected red blood cells. Trainable parameters are shown in the right column, increasing with the number of layers and nodes in each layer. Trainable parameters are shown only for each layer, either a convolution or dense.

As part of model evaluation metrics, the accuracy, loss, f1-score and Matthews Correlation Coefficient (MCC) score were evaluated. The formula for MCC is given as follows:

$$\textbf{MCC} = (TP*TN - FP*FN) / \sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}$$

Where TP is the True Positive, FP is the False Positive, TN is the True Negative, and FN is the False Negative. An f1-score close to 1.0, corresponds to a high accuracy of classification and lower loss value and an MCC score close to 1.0 also indicates a high level of classification and prediction. Different learning rates were evaluated as well and are shown in the appendix for a comparison between the different learning rates for the models.

To prevent overfitting of the model, callbacks were implemented, to monitor whether validation loss was changing significantly or not during the model training regimen. A train/validation split of 0.1 was used when training the model before being deployed on the test dataset.

Analysis and Insights

Given that the utility of the model is to be for clinical and potentially industrial purposes, it is important that the given model perform maximally. The current model described was able to classify the plasmodium stained images with an accuracy of 98.54% and a loss value of 0.0452, which is close to 0 to indicate a high functionality on the test dataset. The training and validation accuracy during the training phase reached a plateau around 97.5 and 95.0% respectively. However, there was no significant fluctuation in either system (Figure 6), and although the training and validation accuracies appeared to be low with a higher loss value, the model performed relatively well on the testing dataset (Figure 7).

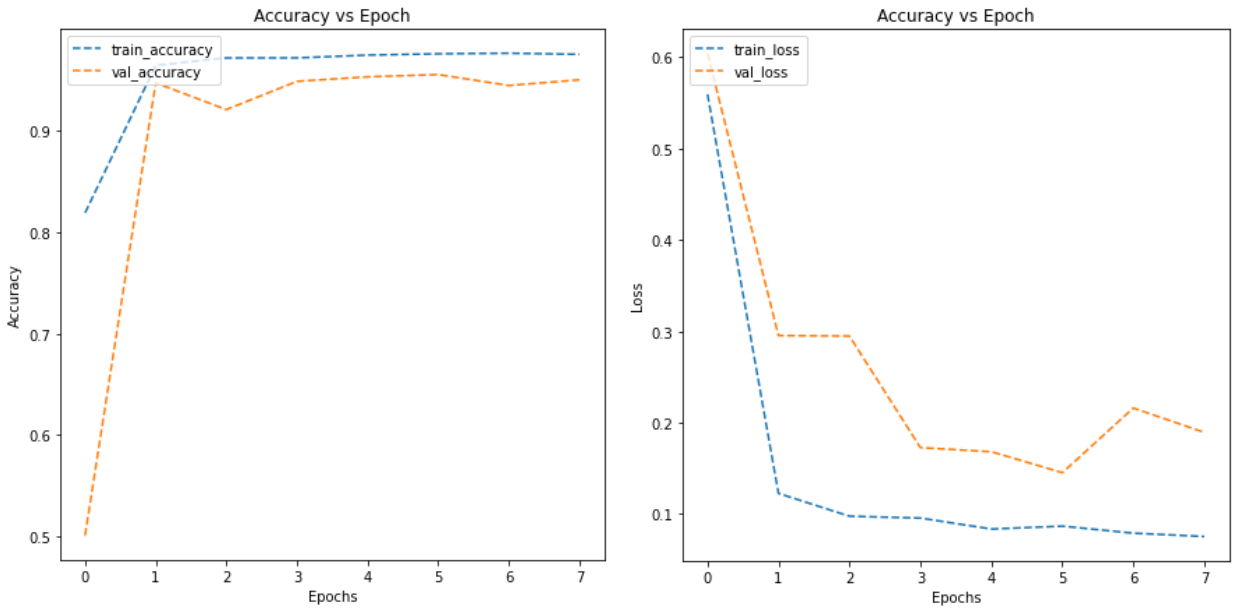

Figure 6: Plot of the accuracy and loss curves against the epochs iterating through the algorithm. Both the accuracy and loss start steep for the first couple epochs while as epochs increase, there is a more steadying of the accuracy and loss in their relative scales.

The f1-score achieved was 0.99 (Figure 7) and MCC score was 0.97, indicating a high classification rate given that these scores are both close to 1.0. Of the 2800 images evaluated, a total of 41 (less than 2%) were misclassified. Based on the confusion matrix presented in Figure 7, the number of misclassified plasmodium infected and uninfected images are shown.

	precision	recall	f1-score	support
0	0.98	0.99	0.99	1385
1	0.99	0.98	0.99	1415
accuracy			0.99	2800
macro avg	0.99	0.99	0.99	2800
weighted avg	0.99	0.99	0.99	2800

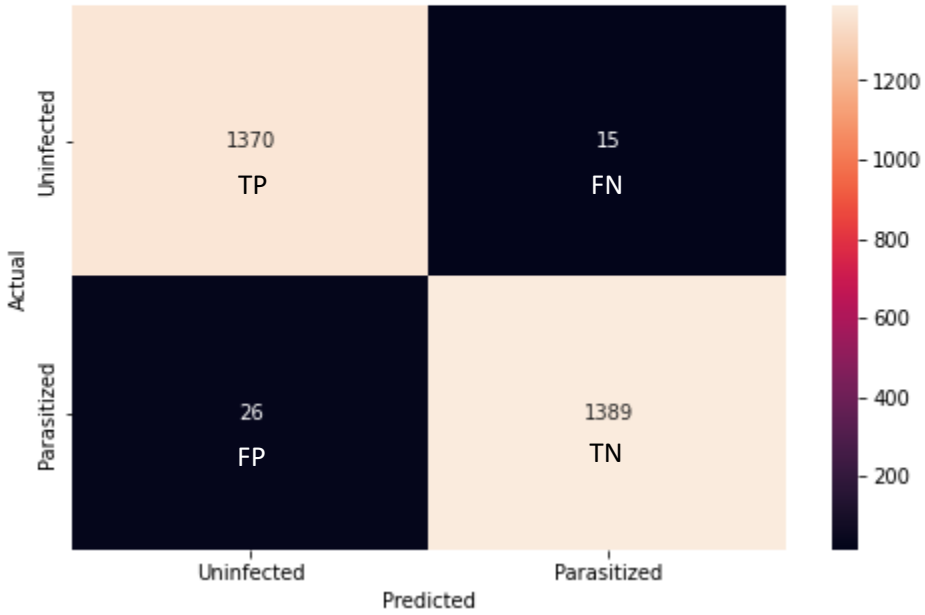

Figure 7: Classification report and confusion matrix for the outcomes of the image evaluation by the presented CNN model. TP: True Positive; FP: False Positive; TN: True Negative; FN: False Negative.

Limitations and Recommendations for Future Analysis and Final Conclusions

While this model proved to be robust for performing binary classification of images that are plasmodium infected, this is not typically how the real world scenario will be. Many limitations occur in this data presentation and model. In terms of the data presentation, typically images will be presented as having a mix of many infected and uninfected cells while also having present other cells such as white blood cells. A limitation of this algorithm is in simply binary classifying plasmodium infected and uninfected cells in a population that has already been stratified. Thus recommendations for future models would be as follows:

Data Presentation:

The data would need to be presented in a manner that is available in the real world, such as having light microscopy images and then annotating the images to identify key features to be designated before putting the data through the CNN. Having a proper data preprocessing is important in order to ensure reliability of the model outcomes. Annotating the images to differentiate different cell types and further the plasmodium infection would be vital for a model that is to be used in a clinically relevant setting.

Data Analysis by CNN:

The data analysis by CNN would need to be done taking into consideration all the different features that would need to be evaluated. For example, while binary classification is useful, the model should be able

to differentiate different cell types as well as determine whether or not red blood cells are infected by plasmodium. With respect to the current model, different CNN architectures can be explored. Furthermore, different models that are used for image analysis, such as Mask R-CNN and the YOLO models can be explored as well[3]. Having different models that utilize different concepts and properties will help determine which model(s) best define the solution to the problem of automating pathological identification.

Overall, it appears that the model performed relatively well on binary classification of plasmodium infected images. Given that the automation using Deep Learning is significantly faster than manual counting and classification, there is a lot of promise for the use case of ML operations in the healthcare industry. Granted that there are barriers to overcome such as HIPAA regulations and patient data privacy, however there is momentum in the case of big data analytics to move towards data anonymity for population data analysis and also cross-identifying patients by tags instead of using identifying information such as their name, DOB, etc. if this information is not needed for analysis. The scope for deep learning, as a result is much wider. It can be used in a variety of cases such as cancer cell classification and gaining insights into potential mechanisms that cancer cells utilize to proliferate, and also for identifying the degree and nature of fibrotic tissues for lung and kidney fibrosis and making a prognosis for the longevity of the patient. Long term implications of implementing such a model on a clinical or industrial scale would mean reduced overhead costs just for an analyst and increased return on investment for driving clinical and research efforts forward to advance science for the society.

[3] A Complete Guide to Image Classification in 2022 - https://viso.ai/computer-vision/image-classification/

APPENDIX 1: Effect of Learning Rate on f1-score as a metric of Model Success

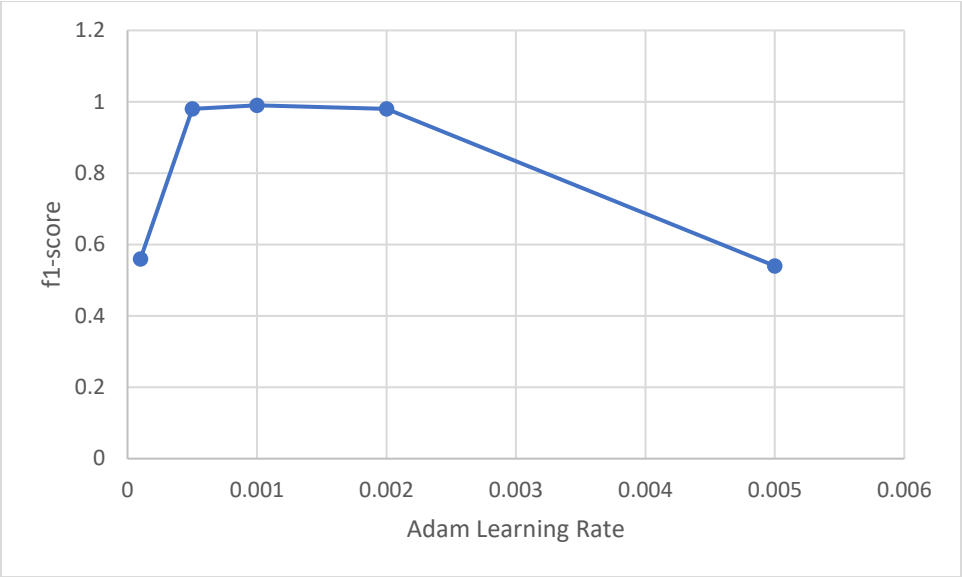

Figure A1: f1-score vs. Adam Learning Rate. The graph shows a curve with a local maxima at 0.001 and an f1-score of 0.99.