Milestone 2

Following our data exploration in milestone 1, we began our model building phase. We initially started with a base model using RelU activation functions on the convolutional layers. With a strong performance of approximately 95% accuracy, we set out to not only improve this accuracy but also to improve the recall score. This means that we want to minimize the number of infected cells getting misclassified as an uninfected cell. Given the risk of further damage a severity of the disease, we are more interested in this metric compared to misclassifying healthy cells as infected ones.

To achieve this, we tried another model by adding another convolutional layer as well as a dropout and a max pooling layer. While we didn't see improvement in the accuracy, we did see an improvement in the recall score.

We tried other methods such as changing the activation function to LeakyRelU, Batch Normalization, augmenting the images, as well as using a pre-trained model. None of these methods saw improvement over our improved initial model. Thus, we decided to improve this model by using the Keras Tuner which allows us to trial different hyperparameters to find the optimal values. We specifically ran the tuner on the number of filters of each convolutional layer and found the best values to further improve our model. This resulted in an improved recall score, making our model even more efficient at avoiding misclassification errors for infected cells.

After tuning our model, we looked at other methods to improve performance. Namely, instead of further alterations to the model itself, we looked at pre-processing the images with various transformations to observe the results when fed into our model. We began by grayscaling, and after feeding the modified images to our model, we saw further improvement in our recall score. We also tried converting to HSV, however, this resulted in a significant drop in performance.

To conclude, we obtained our best performance through our hyperparameter-tuned model and grayscale image inputs. Therefore, we will move forward with this model and explore further methods to improve performance. For example, we can try running the keras tuner on different learning rates for the adam optimizer to see if we find a difference there. Additionally, it would be worth trialing other image preprocessing methods to see if we see more accuracy like we saw with grayscaling. Finally, we can also look into trying other pre-trained models such as ResNet and compare the results to our best model. For future considerations, we could also attempt to correct image labels if there are any errors by consulting with a medical expert or we can try obtaining more data to see if we can get our accuracy score higher.