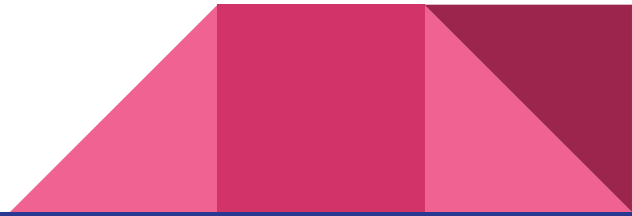
How to Automate Malaria Diagnosis

By Michael Scognamiglio and Amir Edris
nyc-mnhtn-ds-080320

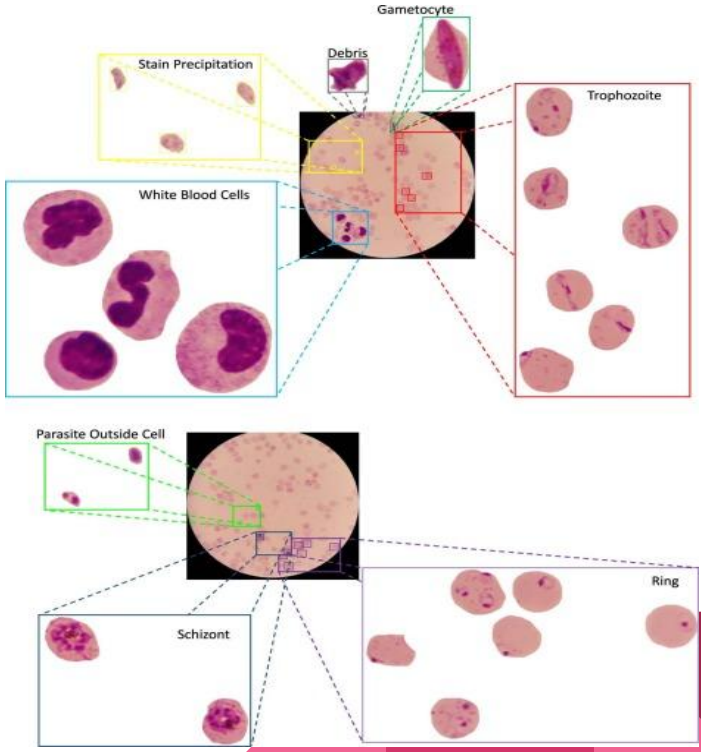
Presentation Structure

- ➔ What is this Project about and What is the Business Case?
- ➔ What's the Dataset and Process like?
- ➔ What's the structure of a KNN like?
- ➔ How was modeling selection?
- ➔ How well did your final model perform?
- ➔ What observations were misclassified and why?
- ➔ What can you conclude from this project?
- ➔ What's next?



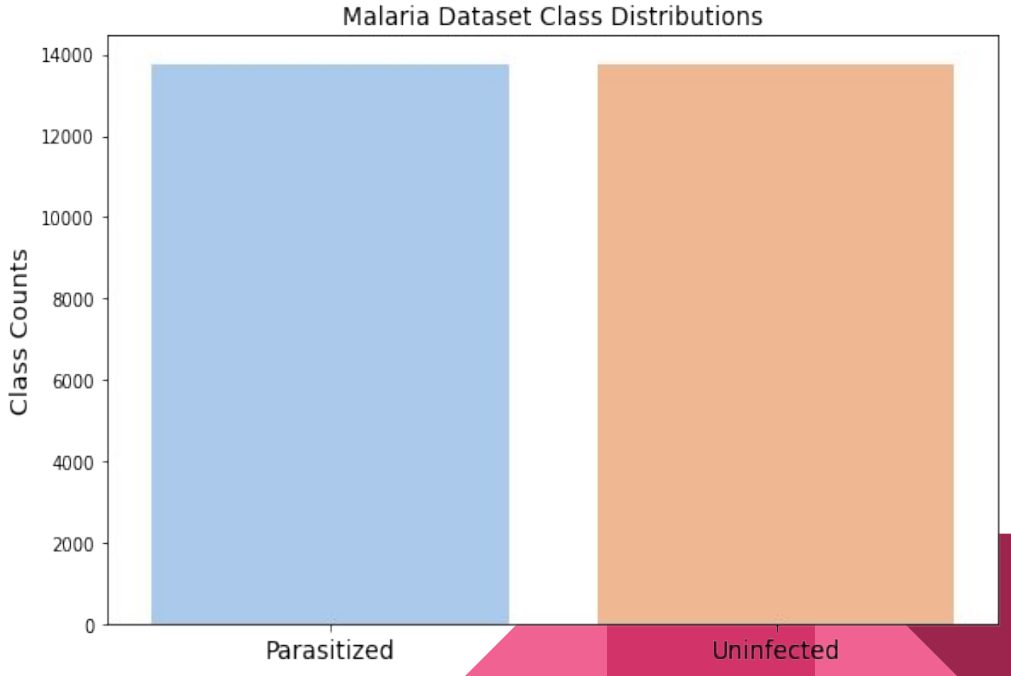
What is this Project Anyway?

- ❏ Accurately Diagnosing Malaria
- ❏ 200 million cases worldwide every year
- ❏ 400,000 deaths worldwide every year
- ❏ Manual counting is standard diagnosis tool
- ❏ Manual counting is not standardized and accuracy varies
- ❏ Automatic Counting (CNN Model) is faster, standardizable, and possibly much more accurate



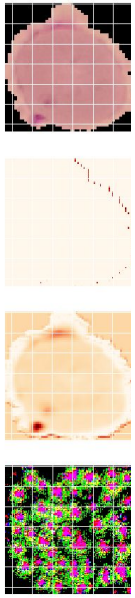
Dataset and Process

- Kaggle Dataset of about 28,000 images of cells
- Class imbalance non existent
- Normalized Data
- Compared uninfected and infected images
- Looked to determine patterns to improve classification



Model Architecture

Exaggerating Input Image Features



Input Layer shape
(128,128, rgb)

Conv Layer 1
16 filters for 3 inputs

Pool Layer 1
128,128 - 64, 64

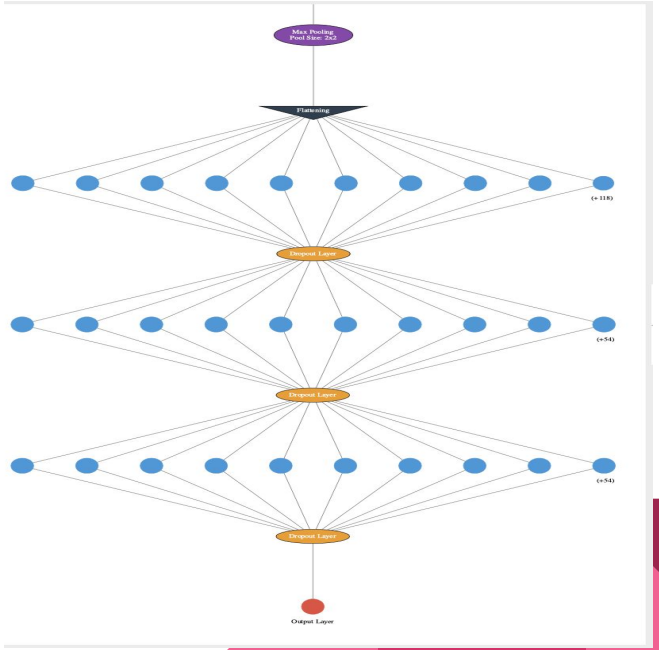
Conv Layer 2
32 filters for 3 inputs

Pool Layer 2
64,64 - 32, 32

Conv Layer 3
64 filters for 3 inputs

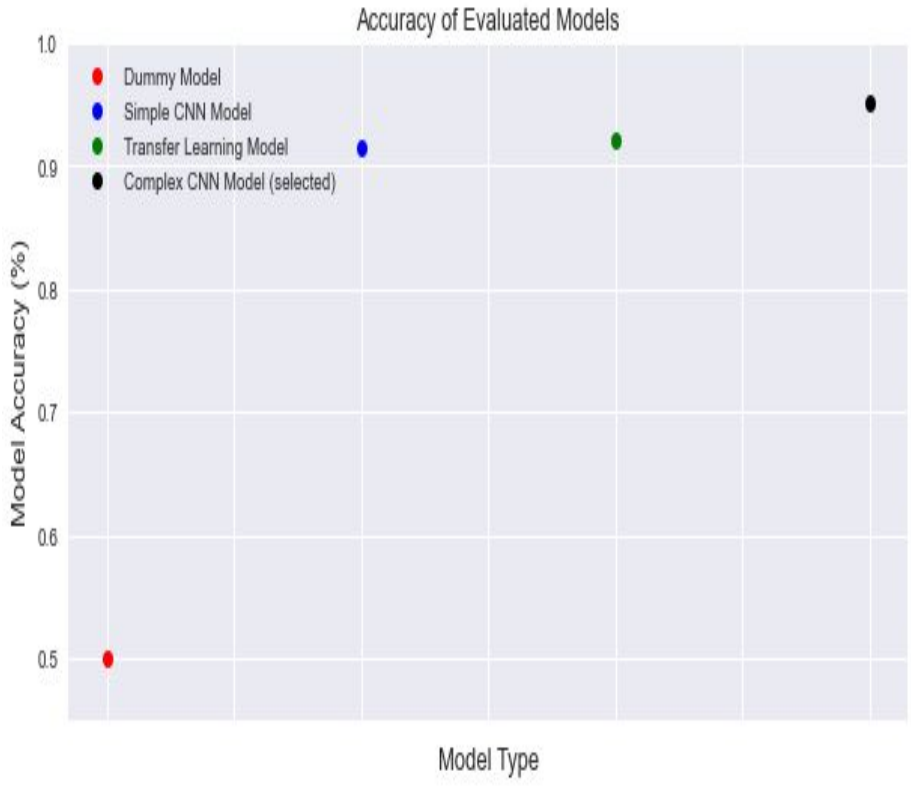
Pool Layer 3
32,32 - 16, 16

Making A Classification From Patterns

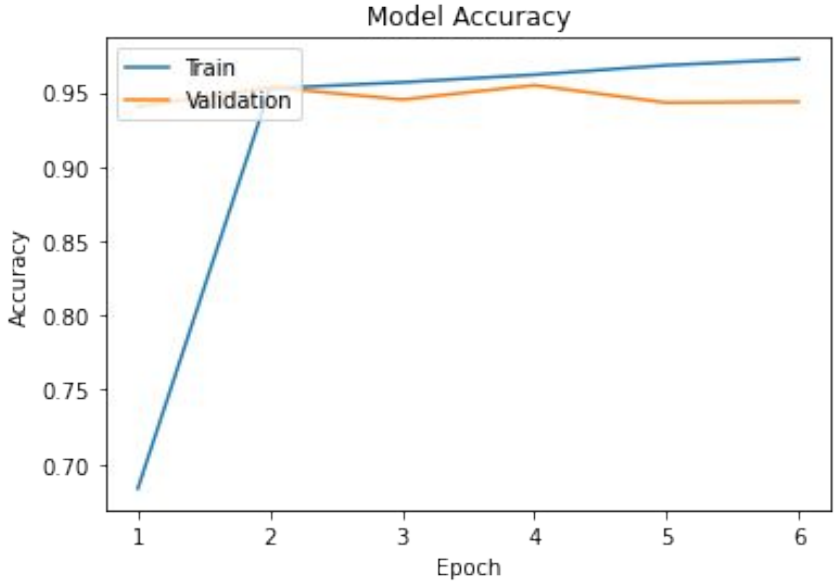


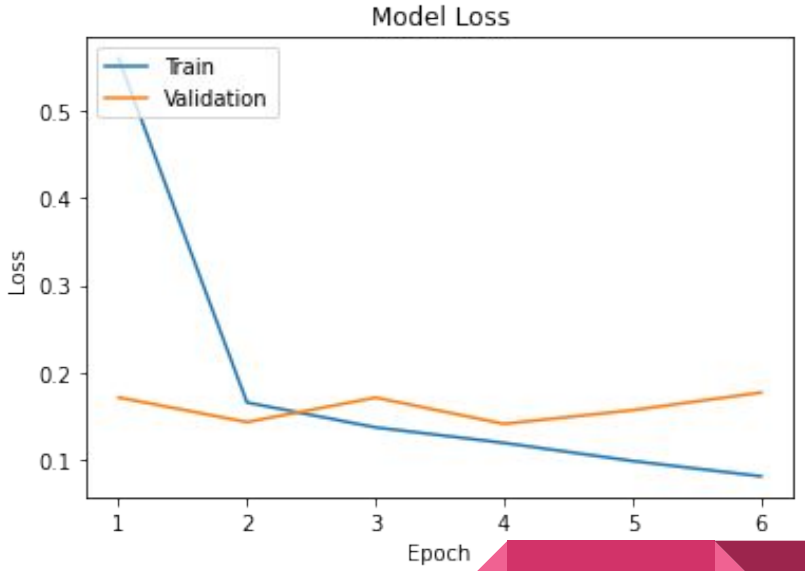
Modeling Selection

Model Type	Accuracy Score (Test Sets)
Dummy Classifier Model	50%
Simple CNN Model	93%
Complex CNN Model	95%
Transfer Learning Model	92%



Final Modeling Accuracy and Loss Curves



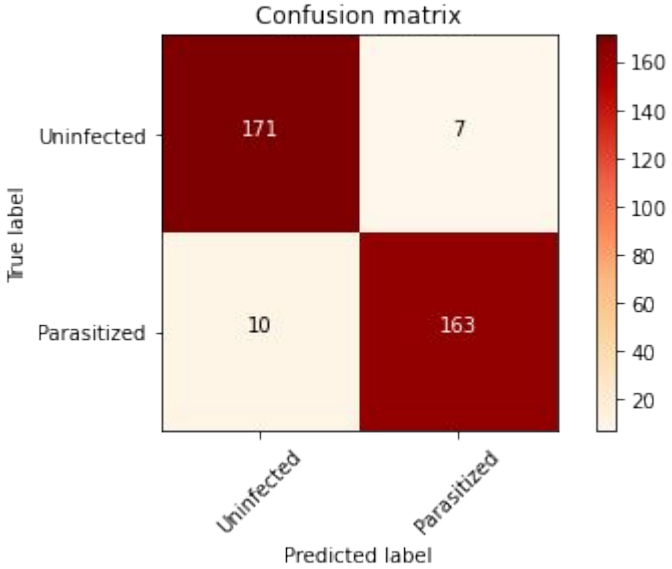


Final Modeling Classification Report

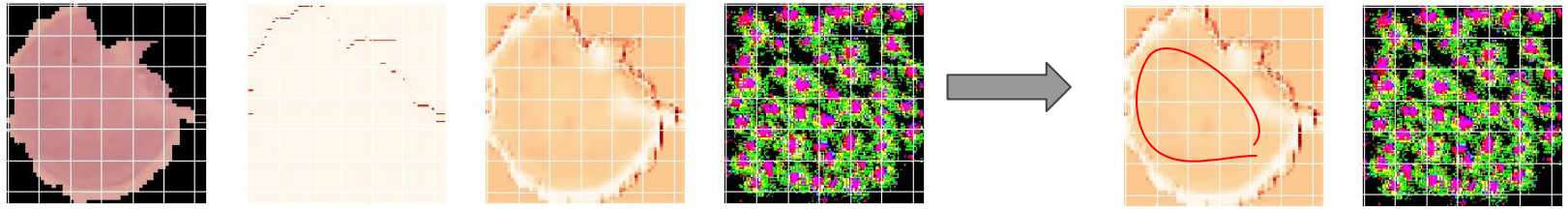
	precision	recall	f1-score	support
Parasitized	0.9447513812154696	0.9606741573033708	0.9526462395543176	178.0
Uninfected	0.9588235294117647	0.9421965317919075	0.9504373177842566	173.0
accuracy	0.9515669515669516	0.9515669515669516	0.9515669515669516	0.9515669515669516
macro avg	0.9517874553136172	0.9514353445476391	0.951541778669287	351.0
weighted avg	0.9516872263378602	0.9515669515669516	0.9515575117303275	351.0

- Accuracy was main evaluation metric used, since no class imbalance there was no concern for misleadingness.
- Recall was second most important, as minimizing false negatives was a high priority.

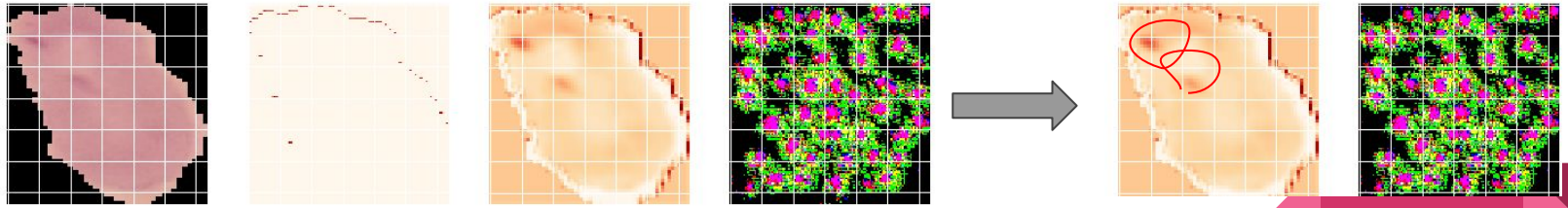
Final Modeling Confusion Matrix



Why False Negatives? Why False Positives?



False Negative - Model Predicted Uninfected When Patient Was Infected (Worst Case!)



False Positive - Model Predicted Infected When Patient Was Uninfected

Conclusions/Recommendations/Next Steps

- Non parasite impurities were often misrecognized as parasites thus causing the misclassification.
- Thus, we would recommend screening through that data as much as possible before training your model. If possible, try to minimize the ambiguity of cells by using the highest image quality possible.
- The two hyperparameters that were most effective in boosting model performance were the complexity of the CNN Model and the order of convolutional layers.
- We would recommend to begin with a simpler CNN but iteratively build its complexity until model performance is sufficient. Aslo, try testing different orders for the convolutional layers to find which best works for your models.
- One area we would like to experiment with in the future is images containing multiple cells. In terms of data collection, this method would be much quicker. It would also provide the model with more information. In the case of a CNN model, this is likely to result in the model finding more patterns it can use to help its classification ability. Thus, we believe this type of model could boost performance even further

Any Questions?