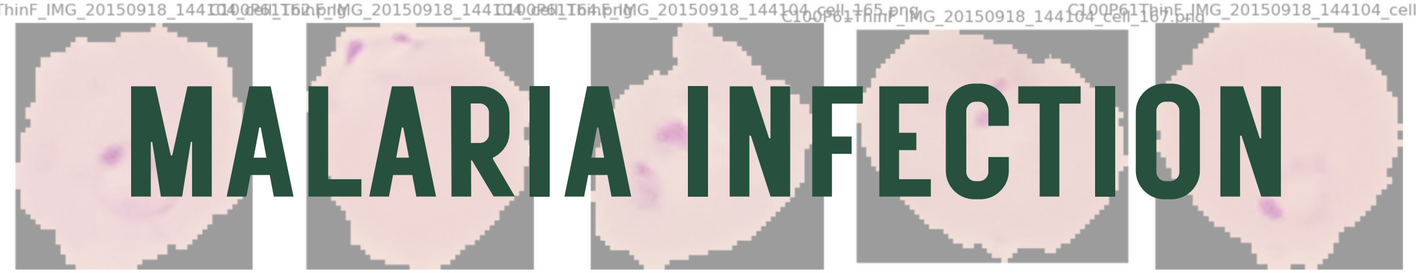
# MACHINE LEARNING: AUTOMATED CELL IMAGE ANALYSIS FOR MALARIA INFECTION DETECTION




GitHub *https://github.com/jnliou/project4*

# TABLE OF CONTENT

# INTRODUCTION

## What is Malaria?

Malaria is a severe and **occasionally lethal illness** caused by a parasite that commonly infects mosquitos that feeds on humans. Individuals who contract malaria typically experience severe symptoms such as high fevers, intense chills, and symptoms resembling the flu (CDC).

# PREVALENCE

247 million

documented clinical cases of malaria

619,000

fatalities as of April 2023

South Africa is home to

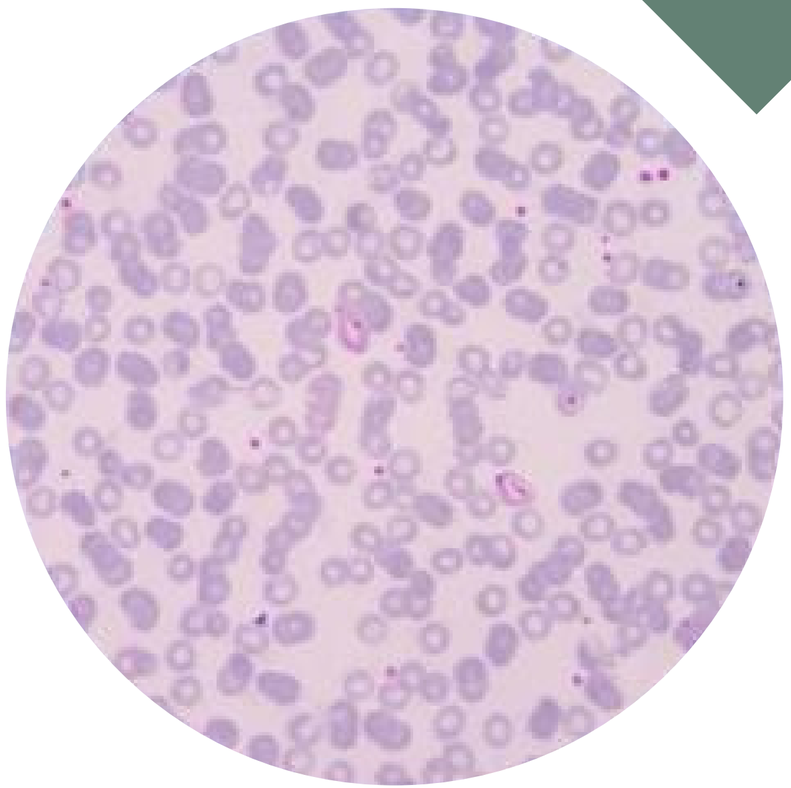
96%

victims out of which 76% are children

according to the WHO...

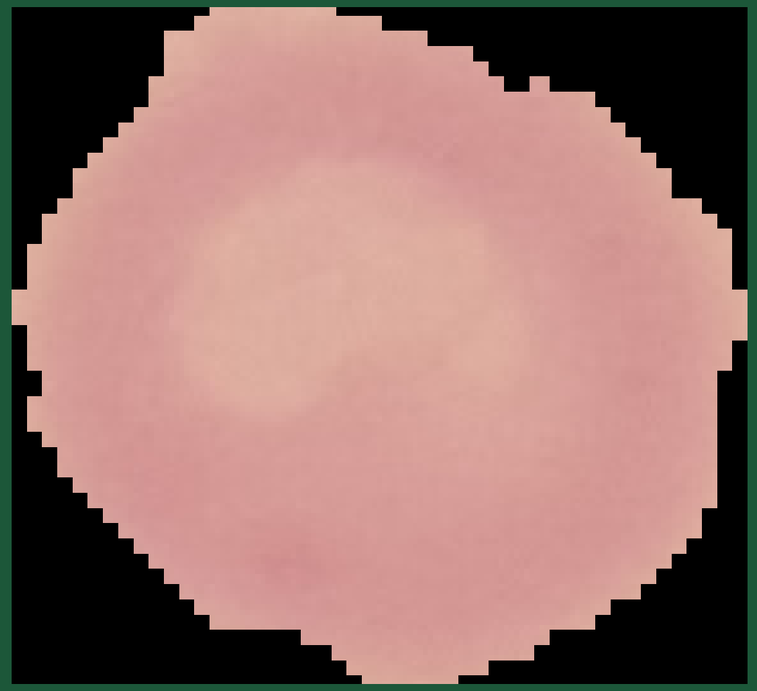
# THE WHY?

- Leveraging Machine Learning Principles for Malaria Detection
- Develop predictive model using cell images
- Streamline healthcare diagnosis
- Empower public for timely assistance
- Facilitate learning for medical and science students

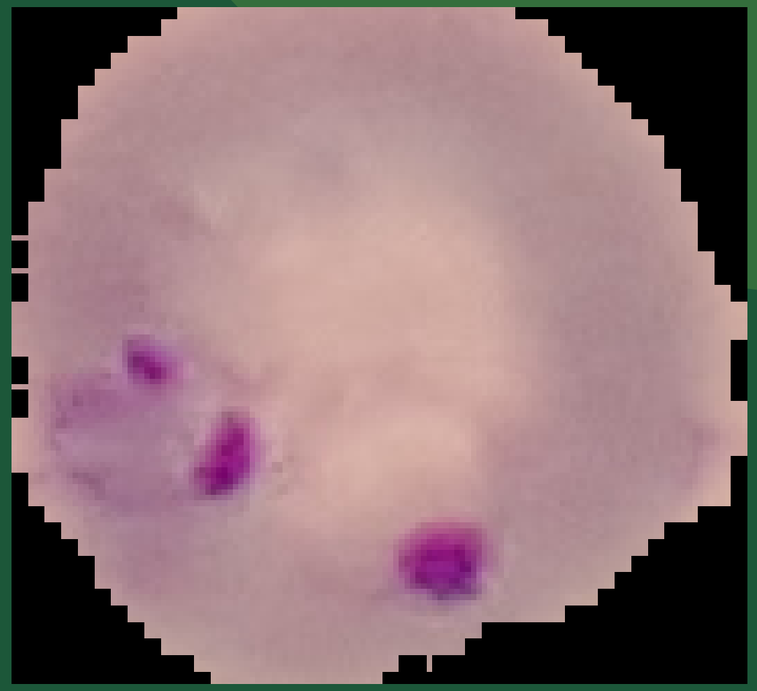


# How is Malaria diagnosed?

According to the Center for Disease Control (CDC), Malaria parasites can be identified by examining **under the microscope** a drop of the patient's blood, spread out as a “**blood smear**” on a microscope slide.



Uninfected



Infected

U.S. National Library of Medicine

# Data Source

- Kaggle: Malaria Cell Images Dataset
- National Library of Medicine

kaggle

# Solution Architecture | Project Timeline

Julia

## DATA CLEANING & PREPROCESSING

- Data Reduction
- Image Resizing
- *Python*

Julia

## EXPLORATORY DATA ANALYSIS

- Image Analysis
- *Tableau* Data Visualizations
- Statistical Analysis
- *Python: SciPyStats, Matplotlib, CV2, OS, etc.*
- PCA Clustering over Image characteristics & Raw Image Data

Srinivas

Manpreet

## DATA SCIENCE MACHINE LEARNING

- Xception Accuracy

*- Base Model : 80%*

*-Top Model: 79%*

- (CNN +RF) Accuracy

*- Optimized Model: 94% -96%*

- *Various other models*

Kevin

## DATABASE

- S3 Bucket
- Flask
- Python
- Excel
- SQLite/Database

Kevin

## WEB APPLICATION

- HTML
- CSS
- JavaScript

# PREPROCESSING

## Data Reduction

- 27600 to 5000 images
- 2500 infected 2500 uninfected
- 70% of images go into training folder 30% go into testing folder

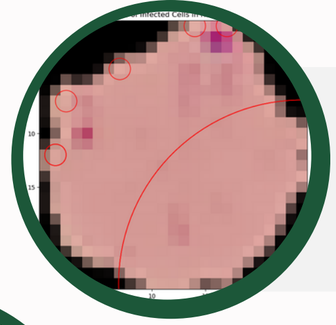
## Image Resizing

- Converting image size from 150x150 to 25x25

150x150

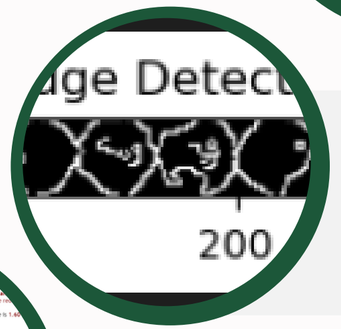
25x25

# Exploratory Data Analysis



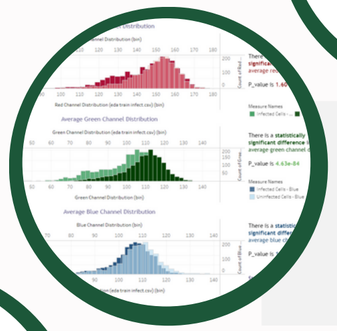
## Assessing Blobs

Mean Blob Size &
Max Blob Size



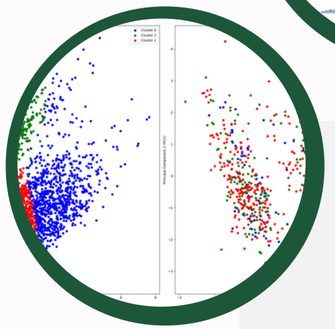

## Edges

Edge Detection &
Average Edge
Density



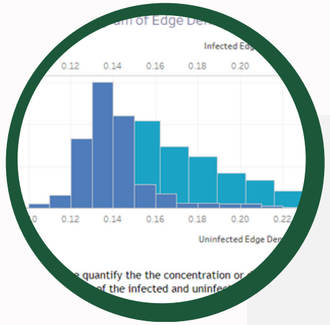
## RGB Channel Distribution

Average Red, Green
& Blue Color Channel
Distrubtions



## PCA Clustering of Image Classification

Utilizing Machine Learning
Principles of PCA
Clustering for EDA &
Image Pixels



## Statistical Analysis & Tableau

Mann Whitney U T-Test,
Data Visualizations via
Matplotlib and Tableau

# MODEL SELECTION

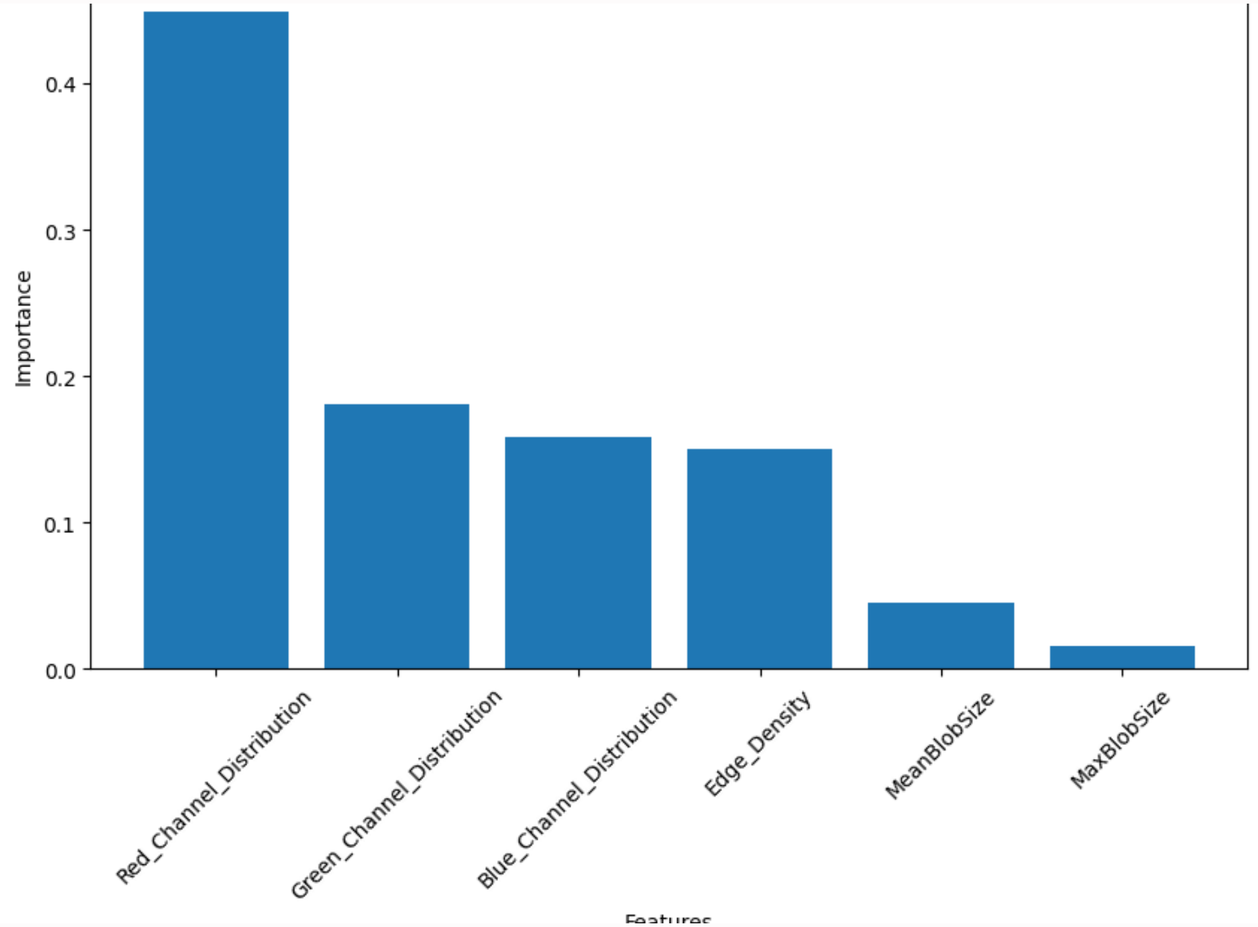
EDA DATASET

EDA Features

| Types of Models ⟶ | Random Forest | Random Forest + Hyp | RF + Gradient boosting | Liner Regression Model | SVC model | SVC+Hyp | SVC + PCA | SVC+RF+NN | Decision Tree |
|---|---|---|---|---|---|---|---|---|---|
| **Hypter parameter** | | 0.10 | | | | 0.10 | | | |
| **Parameters** | | | | | | | | | |
| **Uninfected** | | | | | | | | | |
| **Accuracy** | **83%** | **89%** | **83%** | | **79%** | **85%** | **79%** | **81%** | **74%** |
| **Precision** | 0.81 | 0.88 | 0.80 | | 0.78 | 0.83 | 0.72 | 0.77 | 0.72 |
| **Recall** | 0.83 | 0.90 | 0.86 | | 0.85 | 0.83 | 0.88 | 0.84 | 0.70 |
| **F1-score** | 0.82 | 0.89 | 0.83 | | 0.82 | 0.83 | 0.79 | 0.8 | 0.71 |
| **Infected** | | | | | | | | | |
| **Precision** | 0.85 | 0.90 | 0.87 | | 0.79 | 0.86 | 0.87 | 0.85 | 0.75 |
| **Recall** | 0.83 | 0.88 | 0.81 | | 0.70 | 0.86 | 0.71 | 0.78 | 0.77 |
| **F1-score** | 0.84 | 0.89 | 0.84 | | 0.74 | 0.86 | 0.78 | 0.81 | 0.76 |
| **R2** | | | | 0.32 | | | | | |
| **MSE** | | | | 0.17 | | | | | |

# MODEL SELECTION

## IMPORTANT FEATURES




This graph shows which features of the dataset are significant in determining whether the image is infected or not infected.

Here we see that Mean and Max Blob size are less significant wrt to other features.

# MODEL SELECTION

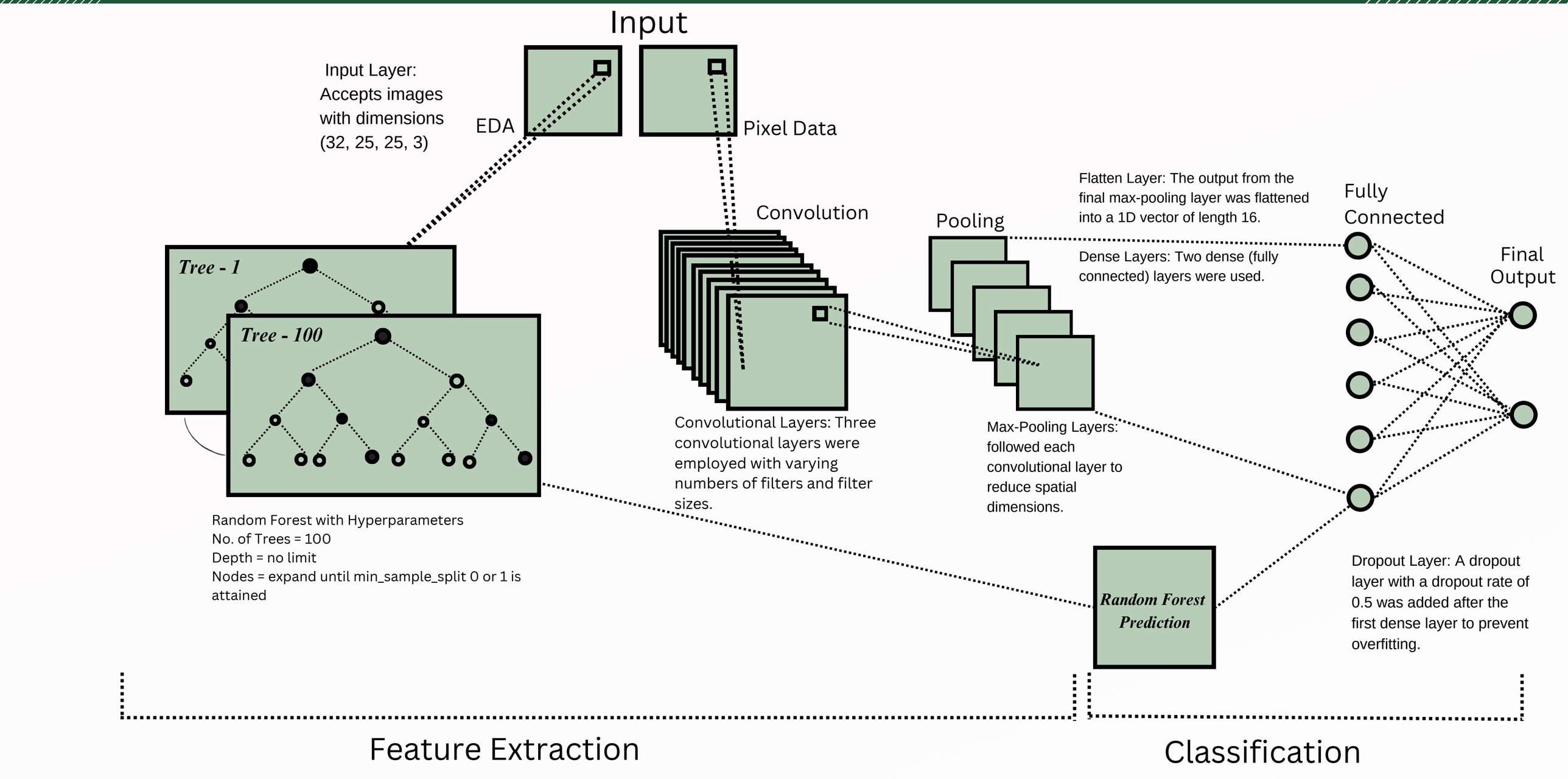
**IMAGE DATASET**

| Image pixel ML | | |
|---|---|---|
| | | |
| | CNN | K-NN |
| Accuracy | 94% | 60% |

**FINAL MODEL**

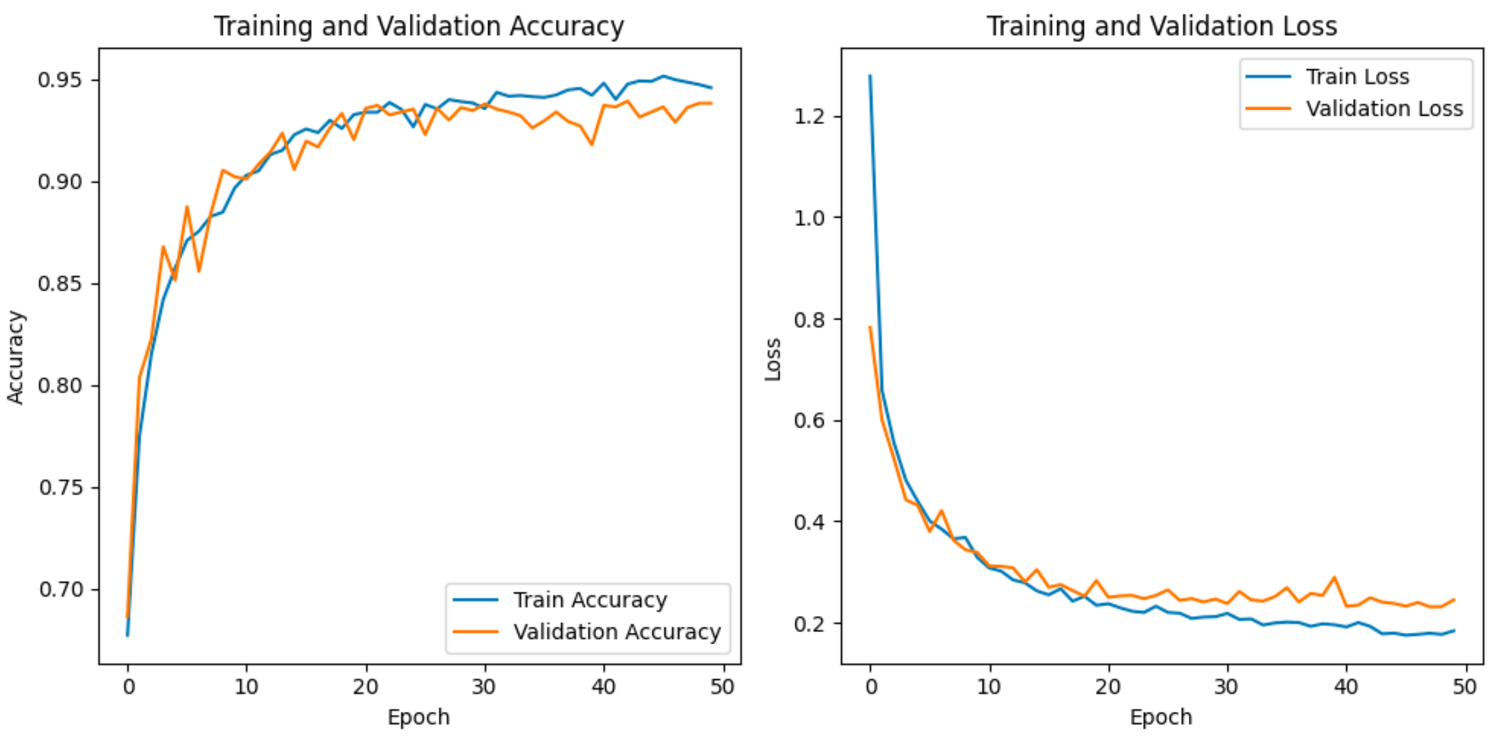
**CNN(PIXEL) + RF(EDA)**

# MODEL ARCHITECTURE



# CNN+ RF(Hyp)

## ACCURACY







```
Accuracy with selected features: 0.89
              precision    recall  f1-score   support

           0       0.88      0.90      0.89       750
           1       0.90      0.88      0.89       750

    accuracy                           0.89      1500
   macro avg       0.89      0.89      0.89      1500
weighted avg       0.89      0.89      0.89      1500
```

## FINAL PREDICTIONS ->

```
Effciency of predicting uninfected by both models is 96.13 %
Effciency of predicting infected by both models is 96.53 %
```

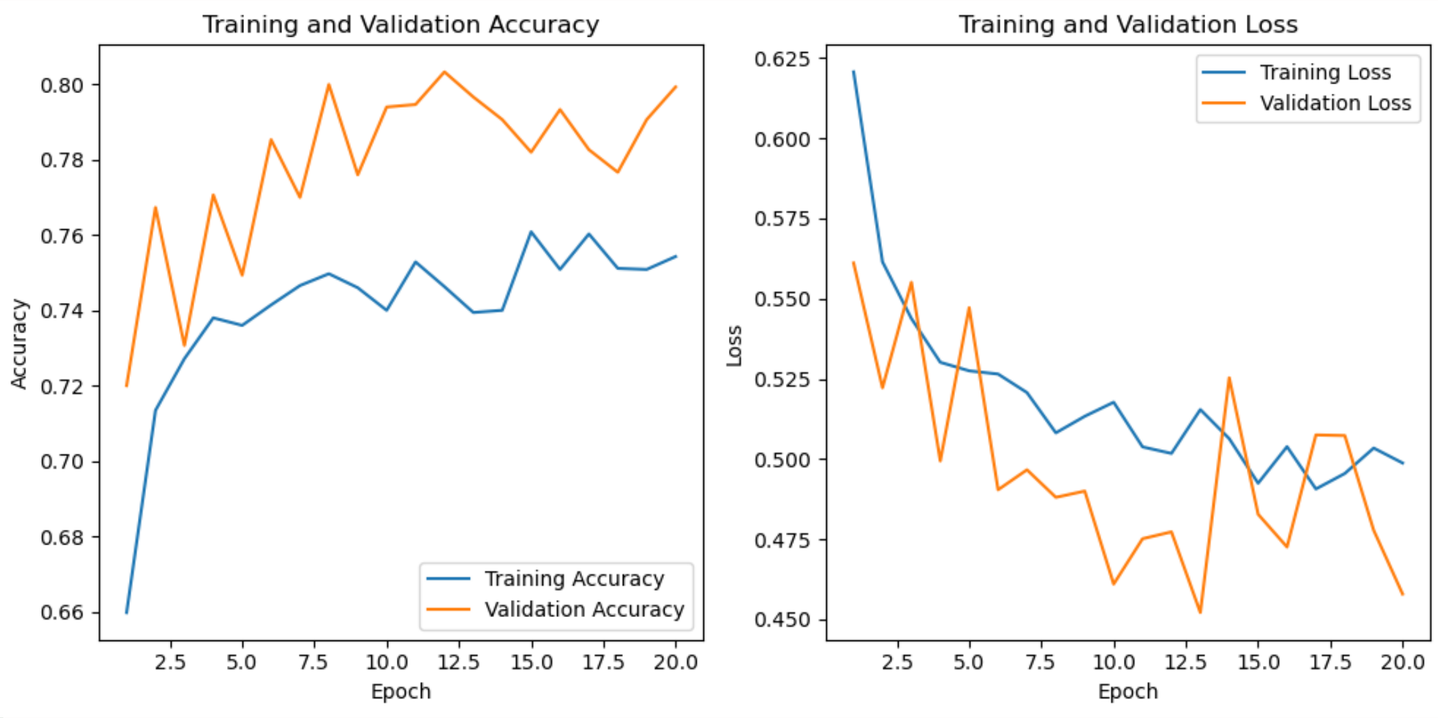
# Xception

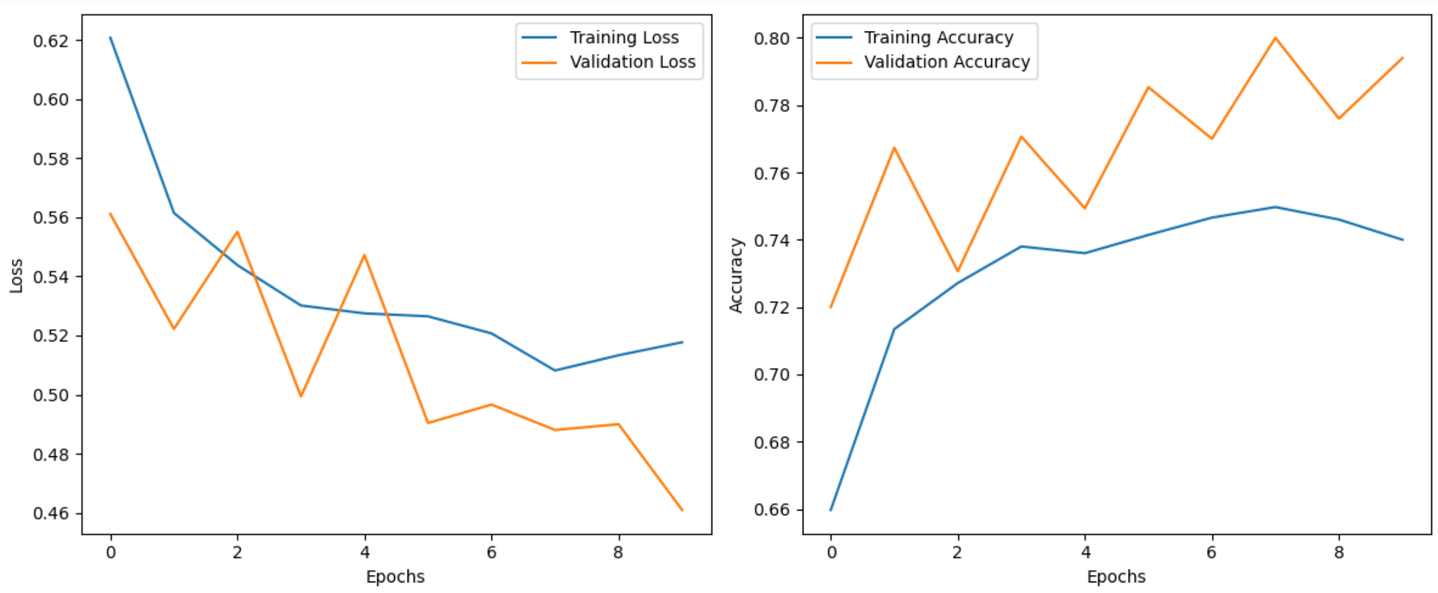
01 Accuracy

02 Loss

03 Fine Tuning

04 Limitations



# DATABASE

Connecting the ML model to our webpage!

## AWS & S3 Bucket

Using the cloud service from AWS and S3 storages to store our images, later on pulling these images using special API.

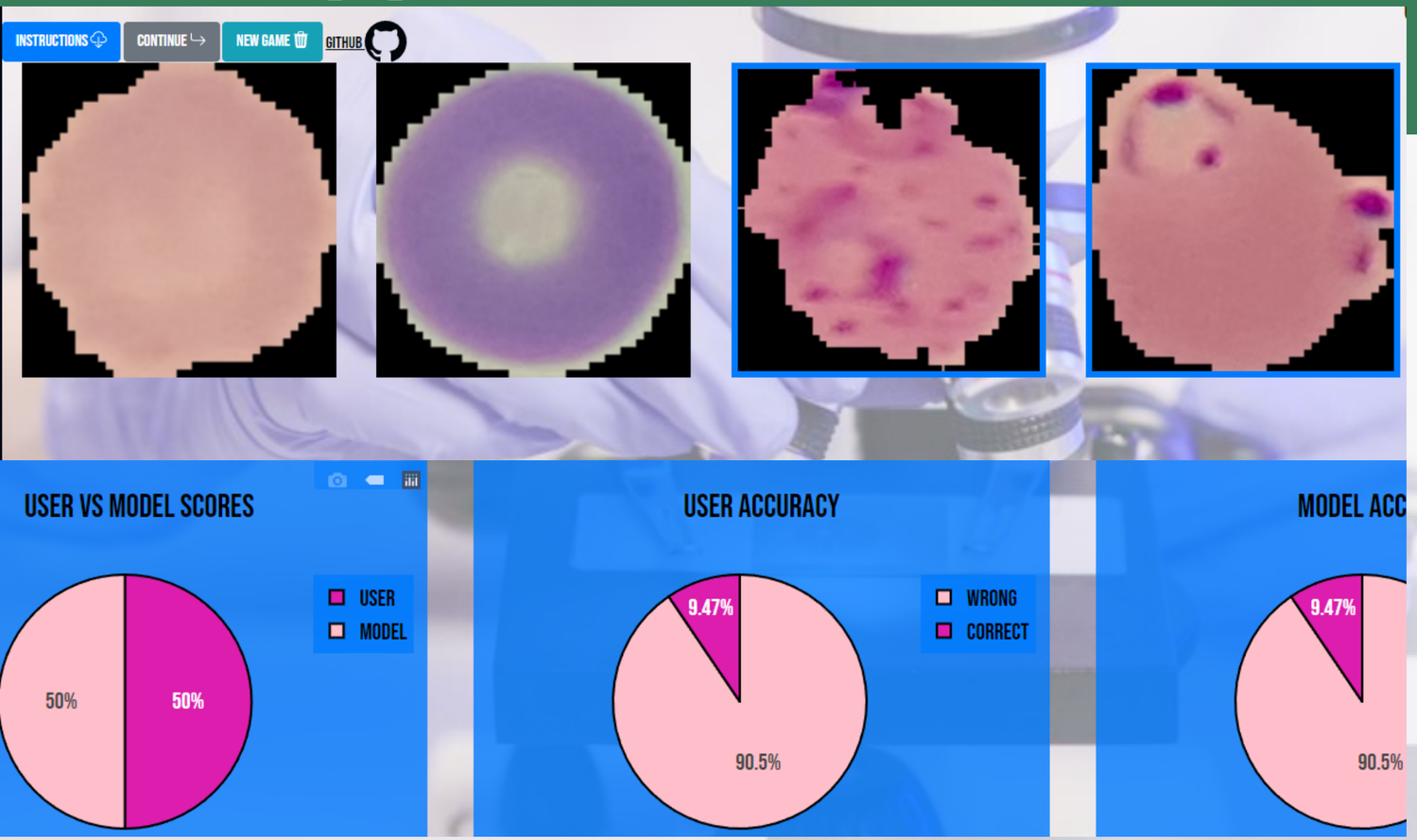
## Python & SQLite

Using python with pandas to filter and create the structure for our database in SQLite, smoothing the process for our web app to read.

## Python & Shutil

Reading through  the data base to see which images will be needed to be store on S3.
Using shutil library to move files to folders for easier management.

# Web Application




## User Interaction

Users can **select** infected cells on the platform.
Once they **submit** their selections, the data is sent to our backend for processing.

**Frontend:** HTML, CSS and JavaScript

**Visualization**: Plotly - pie charts

**Styling & Fonts:** Bootstrap and Google Fonts

**AWS S3:** Storage

# LIMITATIONS

- Limited Diversity
- Quality Control
- Varied Image Dimensions
- Limited Computing Resources

# CONCLUSION + REFLECTION

SIGNIFICANT STEP FORWARD

OPTIMIZED MODEL(CNN +RF)
ACCURACY: 94%- 96%

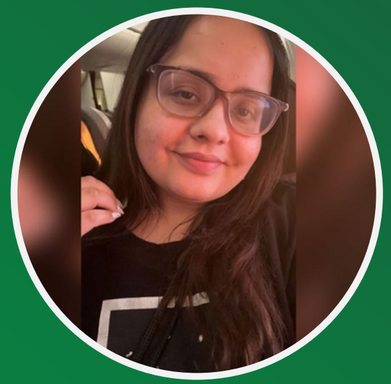
TECHNOLOGY‘S IMPACT ON REAL-WORLD HEALTH CHALLENGES

# Our Dream Team

## Julia Liou

**Data Engineer/Analyst**
**Product Manager**

Linkedin
Github



## Manpreet Sharma

**Data Scientist**

Linkedin
Github

## Srinivas Jayaram

**Machine Learning Engineer**

Linkedin
Github

## Kevin Wan

**Flask & Web Developer**

Linkedin
Github

# Happy Graduation!

UNIVERSITY OF TORONTO
SCHOOL OF CONTINUING STUDIES

# Thank You

Bharat Janarthanan
Anthony Inthavong
Mahesh Peiris
Nayan Saxena
Mike Gonzalez

# Appendix

## Data Source

**https://www.kaggle.com/datasets/iarunava/cell-images-for-detecting-malaria**
**https://ceb.nlm.nih.gov/repositories/malaria-datasets/**

## Github

**https://github.com/jnliou/project4**

## Tableau

**https://public.tableau.com/app/profile/julia.liou6123/viz/EDAonCellImagesofMalaria-Tableau/ExploratoryDataAnalysisonMalariaCellImages**

## Project Management

**https://trello.com/b/ZZba331m/project-4**
**https://miro.com/app/board/uXjVMgjTsg0=/?share_link_id=288261535764**

# Appendix Cont’d

## References


Byeon, E. (2020, September 11). Exploratory data analysis ideas for image classification. Medium. https://towardsdatascience.com/exploratory-data-analysis-ideas-for-image-classification-d3fc6bbfb2d2

Centers for Disease Control and Prevention. (2019, October 18). CDC - Malaria - diagnosis & treatment (United States). Centers for Disease Control and Prevention. https://www.cdc.gov/malaria/diagnosis_treatment/index.html

Google. (n.d.). Google fonts. https://fonts.google.com/

Microscope photos, download the best free microscope stock ... - pexels. (n.d.). https://www.pexels.com/search/microscope/

Team, K. (n.d.). Keras Documentation: Transfer Learning & Fine-tuning. https://keras.io/guides/transfer_learning/

Tian, Y. (2020, June 17). Integrating image and tabular data for Deep Learning. Medium. https://towardsdatascience.com/integrating-image-and-tabular-data-for-deep-learning-9281397c7318


DEMO TIME