

# CV & TL for Human Malaria Detection

Michael Sam, Abbie Kinaro, Kimberly Llajaruna, Emilia Mazzolenis, Hope Neveux

# Problem Statement

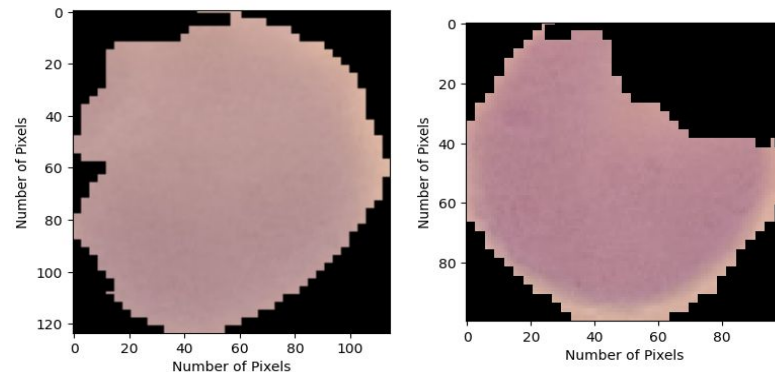
- Develop a computer vision-based model to accurately **identify malaria infection** in human cells
- Aims to improve the efficiency and accuracy of malaria diagnosis by classifying cell images as infected or clear

**How can machine learning algorithms be utilized to precisely identify malaria parasites in human cell images and classify them as infected or not?**

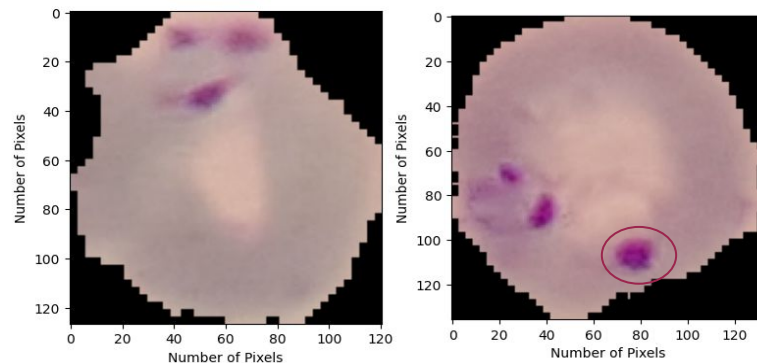
# Dataset Overview

- 27,558 total cell images
- 2 classes
  - parasitized (infected)
  - uninfected
- 193 patients
  - 148 infected
  - 45 uninfected
- Balanced
  - 13,779 images per class
- Contains only the images in folders
- RGB coded
- Infected cells have distinct visual cues

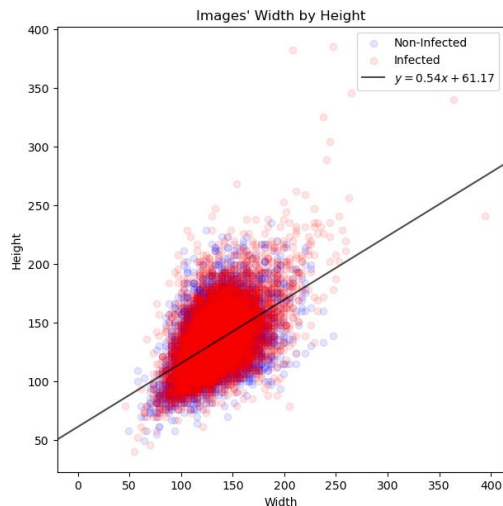
First 10 Uninfected Cell Images



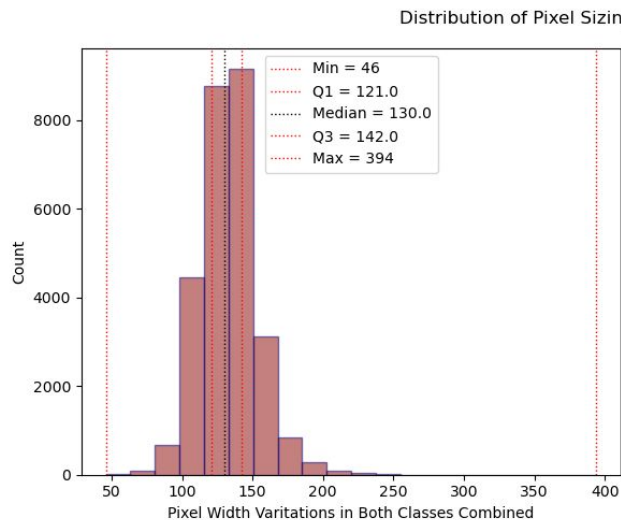
First 10 Malaria Infected Cell Images



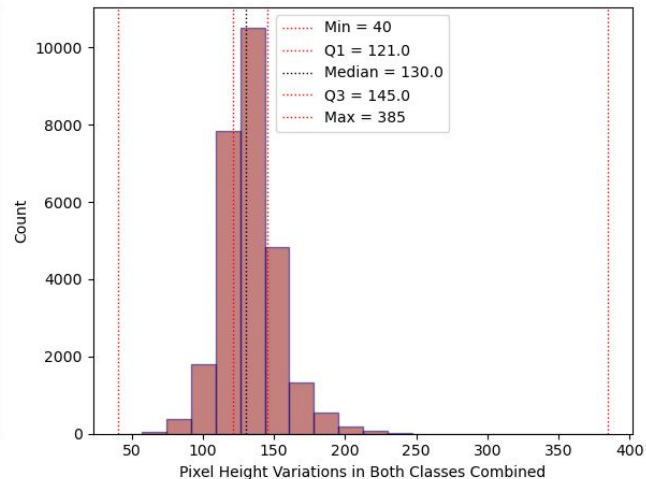
# EDA: Image Sizing



- Most images are relatively square



- Width ranges from 46 to 394 pixels
- Height ranges from 40 to 385 pixels



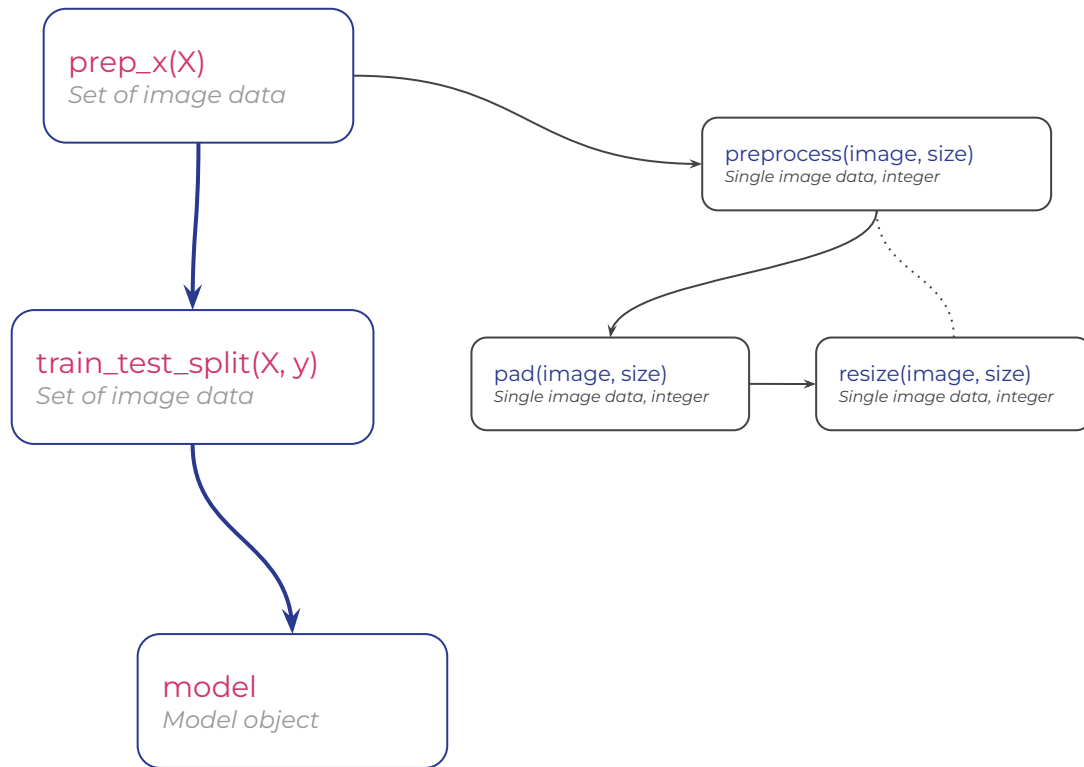
# Visual Findings & Problems To Address

- Images are relatively square, but very different sizes
  - Upsampling likely to cause infection sparsity during vision
  - Downsampling likely to cut images and lose infection at cell borders
  - Combination of up and down sampling
- Scale image values
  - Some images are more RGB saturated
  - Some cells have a different dye and may become latent confounder during training

# Preprocessing

- `prep_x` takes an array of images or single image, returning the padded and resized image
- `preprocess` calls to `resize` then `pad` each of the passed images
- Split the cleaned images 70/30 train, test
- Pass the final data to the `model` to fit and predict

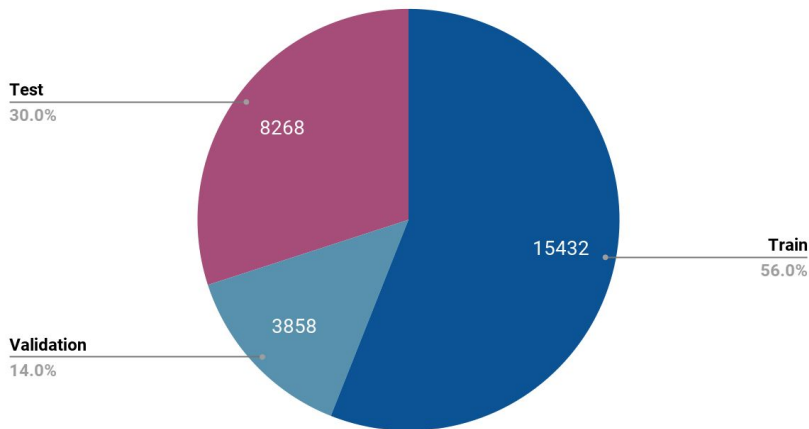
# Pipeline



# Modeling

## Commonalities

Dataset Split



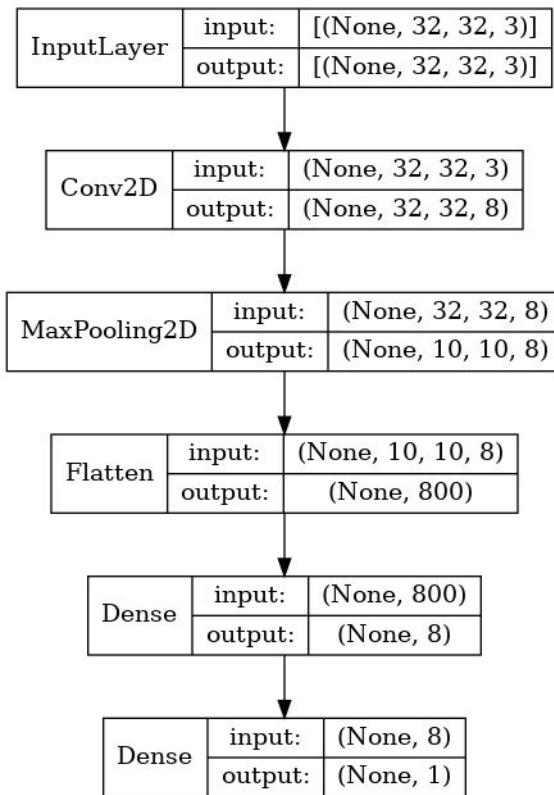
- Data
  - Prepped with pipeline
  - Train-Test Split: 30%
  - Train-Val Split: 20%
- Images
  - (32, 32, 3)
  - “Same” padding
- Batch Size
  - 2,756 images: 10% of the data
- Loss
  - Binary Crossentropy
- Metrics
  - Accuracy
- Optimizer
  - Adam\*

---

\* Differing learning rates

# Baseline Model

Train Time: 4.03 s

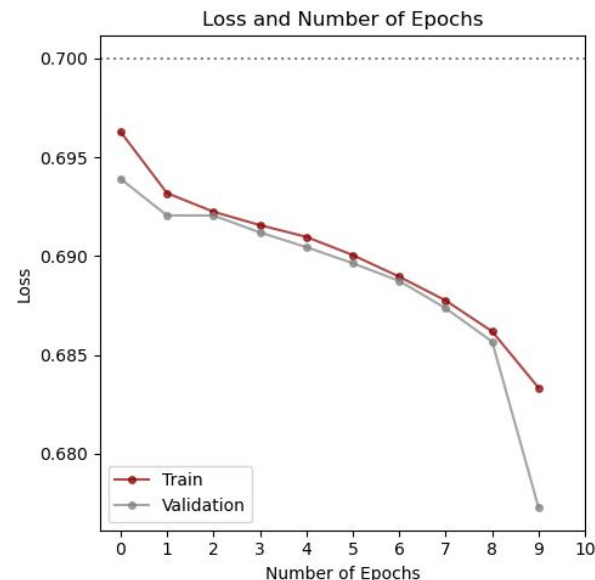
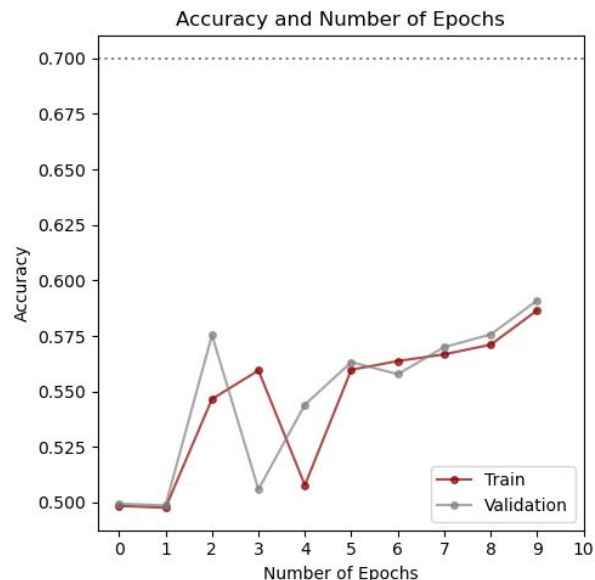


- 6,641 total (+ trainable) params
- 8 filters, kernel size 3
- Default Adam learning rate (1e-3)
- 10 epochs with 2,756 batch size
- Single Convolution + Maxpooling



# Analysis: Baseline Model

- This model is **not converging**
- Significant accuracy increase and loss decrease in both sets
- Needs to **train for longer**
- **Additional layers** allows it to learn more “complicated” features such as the ridges and colors of infectious regions



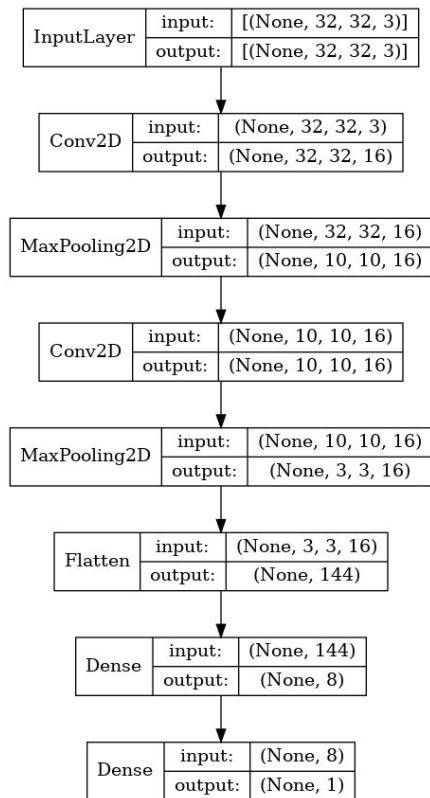
Train – 58.65%

Val – 59.10%

Test – 59.08%

# Model Improvement

*Train Time: 30.7 s*

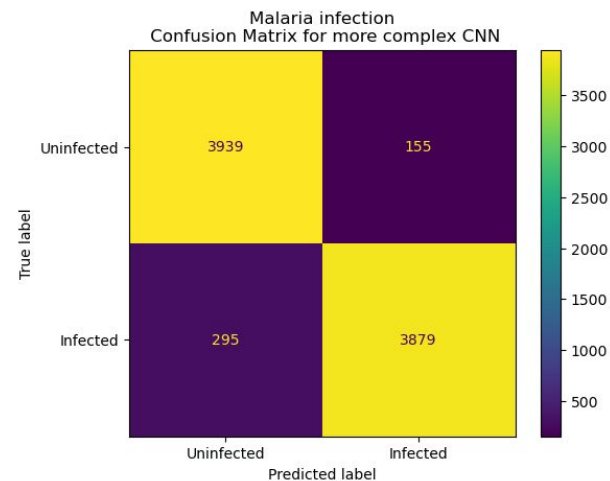
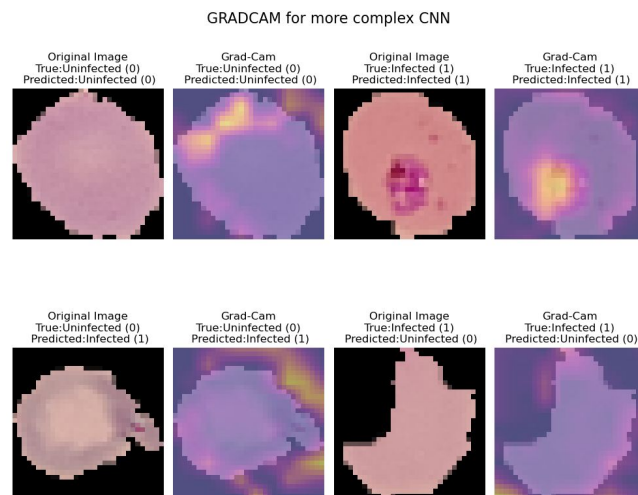
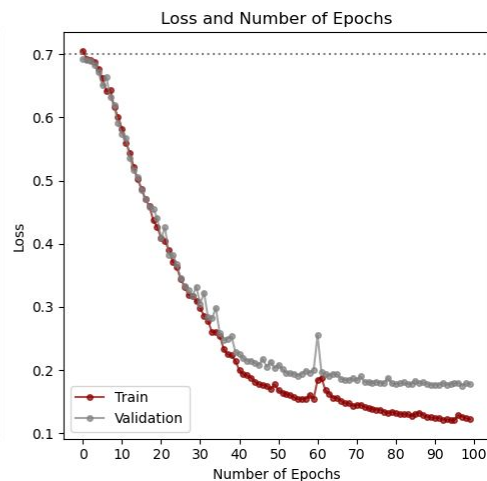
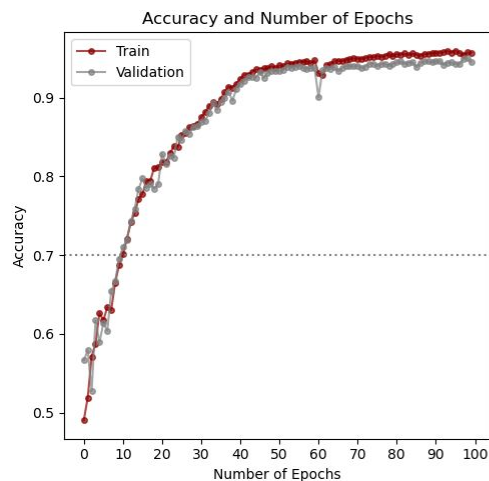


- 3, 937 total (+ trainable) params
- 16 filters, kernel size 3
- Higher Adam Learning Rate (1e-2)
- 100 epochs with 2,756 batch size
- Additional Convolution + Maxpooling

# Analysis: Complex CNN

Train – 95.70%    Val – 94.50%    Test – 94.56%

- Balancing bias and variance
- Looking at areas of high color saturation



# ResNet50V2

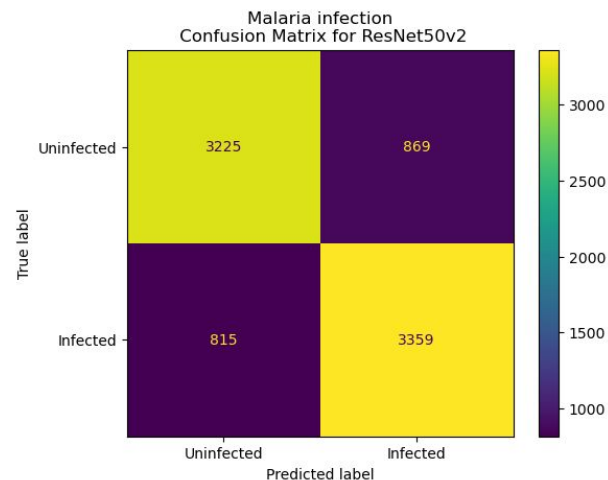
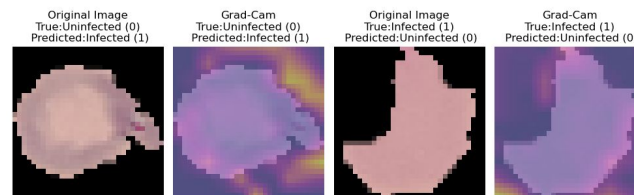
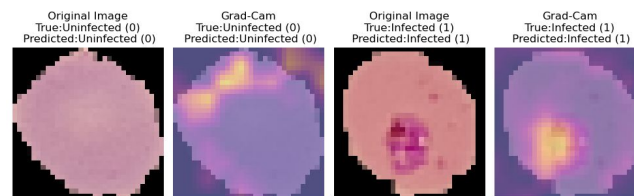
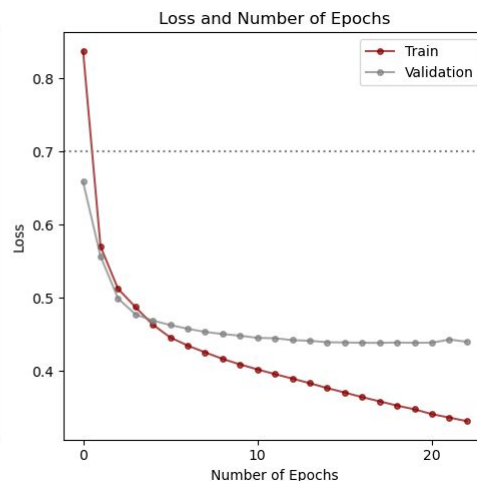
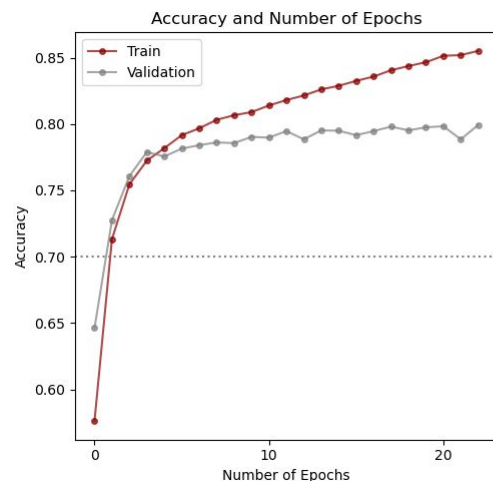
*Train Time: 37.1 s*

- 25,664,001 total parameters
  - 2,099,201 trainable
  - 23,564,800 non-trainable
- 64 filters, 193 layers
- Early Stopping with 5 epoch patience, keep best weights
- Default Adam learning rate (1e-3)
- 100 epochs with 2,756 batch size

# Analysis: ResNet50v2

Train – 85.52%    Val – 79.94%    Test – 79.63%

- Discrepancy in performance: ResNet overfits given the robust data it was trained on



# False Negative GradCAM Study

- Cell images have different stains, and in some cases are looking at cell edge instead of raised characteristics
- Previous papers\* on tumor cell imaging showed **hue/saturation augmentation increased detection**
- Willing to sacrifice accuracy for better False Negative performance

## METHODS

- Create function to adjust image hue and saturation
  1. CNN2 with function as data preprocessing
  2. CNN2 with function as data augmentation
  3. More Complex CNN with function as data augmentation

\* Dzubay, J.A., Jerusalmi, A., Jesudason, R. et al. An image analysis algorithm based on the hue saturation density transformation, an important tool for melanoma immunotherapy research. j. immunotherapy cancer 2 (Suppl 3), P134 (2014). <https://doi.org/10.1186/2051-1426-2-S3-P134>

## MODEL 1

CNN2 with function as  
data preprocessing

*Train: 92.28%*

*Val: 89.71%*

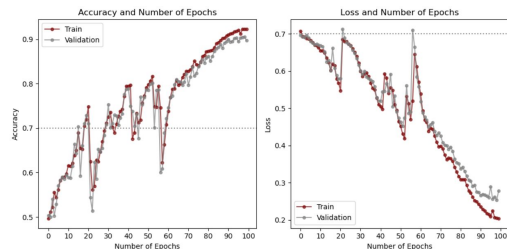
*Test: 87.99%*

**Train set:** x\_train\_mod

**Test set:** x\_test\_mod

**Test TPR:** 91.59%

**Train Time:** 27.7 s



# AC 209B: Hue and saturation processing

Original CNN TPR: 92.93%

## MODEL 1

CNN2 with function as  
data preprocessing

*Train: 92.28%*

*Val: 89.71%*

*Test: 87.99%*

**Train set:**

x\_train\_mod

**Test set:**

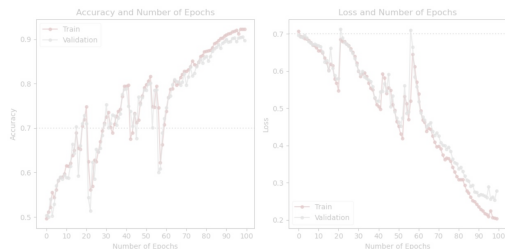
x\_test\_mod

**Test TPR:**

91.59%

**Train Time:**

27.7 s



## MODEL 2

CNN2 with function as  
data augmentation

*Train: 94.75%*

*Val: 91.33%*

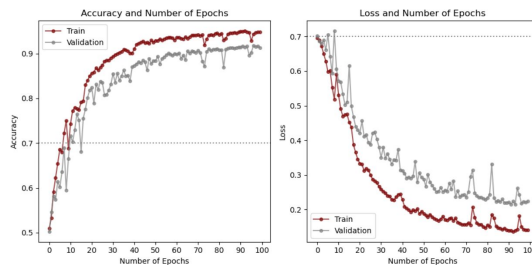
*Test: 93.30%*

x\_train + x\_train\_mod

x\_test

93.79%

52.6 s





# AC 209B: Hue and saturation processing

Original CNN TPR: 92.93%

## MODEL 1

CNN2 with function as  
data preprocessing

*Train: 92.28%*

*Val: 89.71%*

*Test: 87.99%*

**Train set:**

x\_train\_mod

**Test set:**

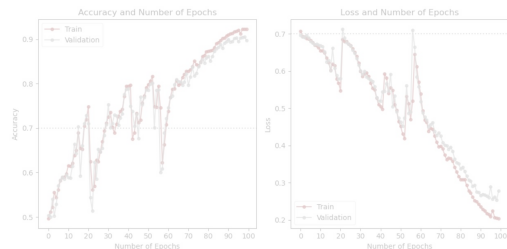
x\_test\_mod

**Test TPR:**

91.59%

**Train Time:**

27.7 s



## MODEL 2

CNN2 with function as  
data augmentation

*Train: 94.75%*

*Val: 91.33%*

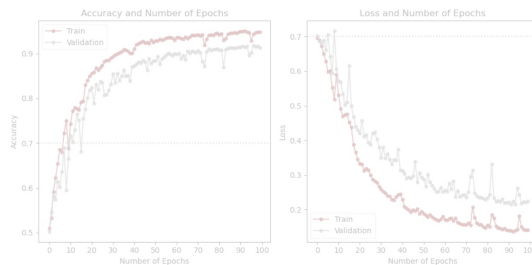
*Test: 93.30%*

x\_train + x\_train\_mod

x\_test

93.79%

52.6 s



## MODEL 3

More complex CNN  
with function as data  
augmentation

*Train: 94.34%*

*Val: 91.28%*

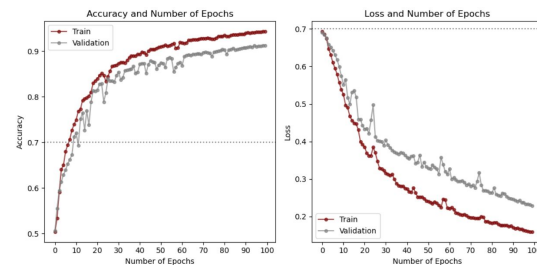
*Test: 94.01%*

x\_train + x\_train\_mod

x\_test

92.76%

56.3s

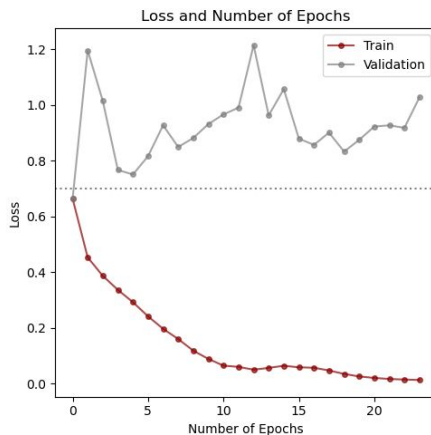
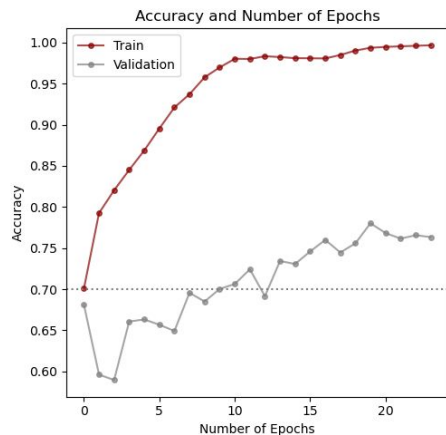


# AC 209B: Fine-tune pre-trained model

## Train Head + Final BatchNorm & Convo

Train – 99.66%    Val – 76.33%    Test – 74.73%

- Unfreeze last 7 layers (3.6%) of the model
- 5,514,241 trainable parameters
- 100 epochs + Early Stopping
- Batch Size 2,756
- Default Adam Learning Rate (1e-3)
- *Train Time: 42.5 s*

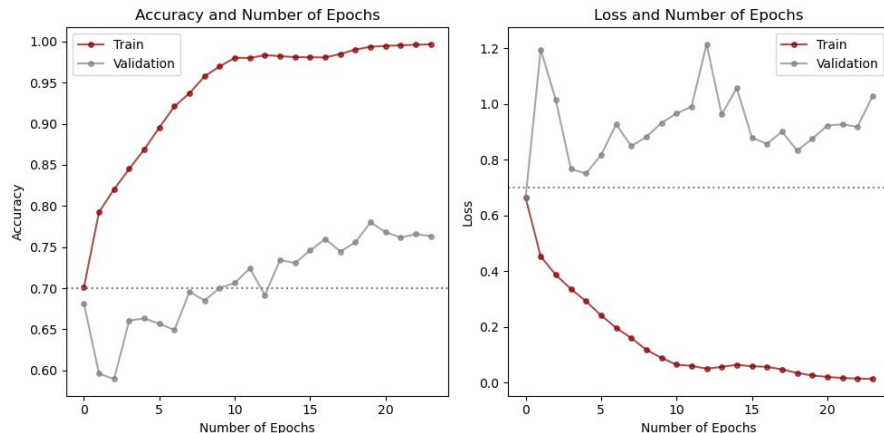


# AC 209B: Fine-tune pre-trained model

## Train Head + Final BatchNorm & Convo

Train – 99.66%    Val – 76.33%    Test – 74.73%

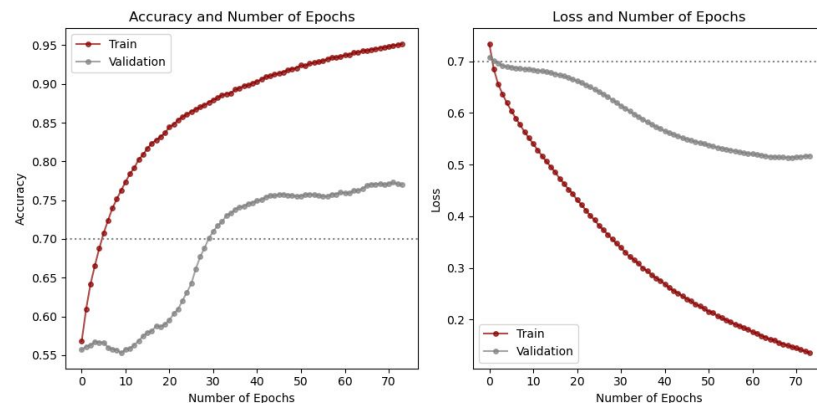
- Unfreeze last 7 layers (3.6%) of the model
- 5,514,241 trainable parameters
- 100 epochs + Early Stopping
- Batch Size 2,756
- Default Adam Learning Rate (1e-3)
- Train Time: 42.5 s



## Train Head + Last 2 BatchNorm & Convo

Train – 95.17%    Val – 77.01%    Test – 76.54%

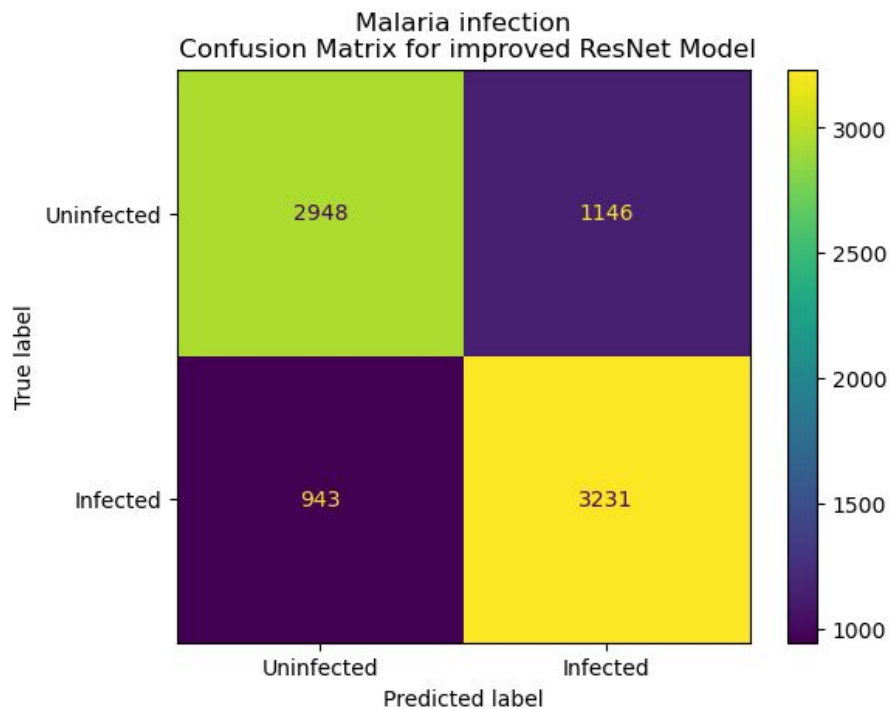
- Unfreeze last 10% (19 layers) of the model
- 9,978,881 trainable parameters
- 100 epochs + Early Stopping
- Batch Size 2,756
- Lower Adam Learning Rate 1e-5
- Train Time: 2 min 25 s



## AC 209B: Fine-tune pre-trained model

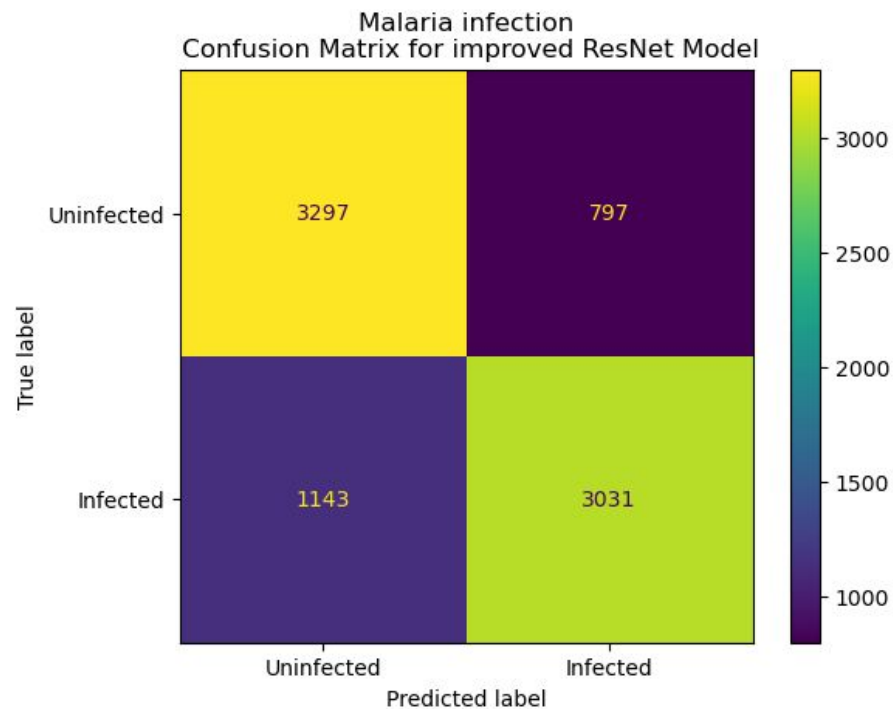
### Train Head + Final BatchNorm & Convo

Train – 99.66%    Val – 76.33%    Test – 74.73%



### Train Head + Last 2 BatchNorm & Convo

Train – 95.17%    Val – 77.01%    Test – 76.54%



# Conclusions and Inferences

	Train	Validation	Test
<b>Baseline</b>	0.59	0.59	0.59
<b>CNN2</b>	0.96	0.95	0.95
<b>ResNet</b>	0.86	0.80	0.80
<b>Saturation ResNet*</b>	0.94	0.91	0.94
<b>Fine-Tune ResNet*</b>	0.95	0.77	0.77

- CNN2 has the best, stable performance
- **Simple is better**: ResNet not needed for simple images and task
- **Color and HSI** are important features of microscopic samples when dye is involved

# Improvements & Future Work

- Fine-tuning ResNet with differential learning rates
- Use a different SOTA model
  - Other papers\* found similar ResNet results when classifying simple cell images
- Visual inspection indicates different stages of infection, which could be learned via mixture models or general clustering

\* Ma, L., Shuai, R., Ran, X. et al. Combining DC-GAN with ResNet for blood cell image classification. Med Biol Eng Comput 58, 1251–1264 (2020). <https://doi.org/10.1007/s11517-020-02163-3>