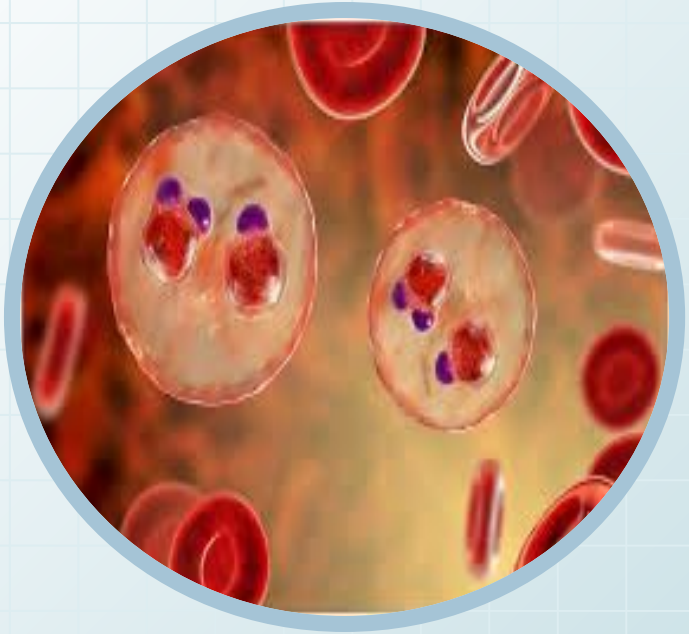


Cell Image Detection of Malaria

Group 7: Daniel Downs, Phillip Zhu, Shyam Goyal, Stavan Stah, Tiffany Dinh

Table of contents

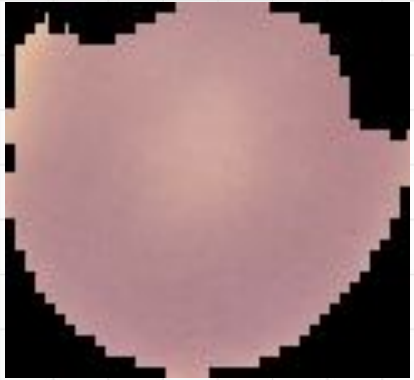
Explain the Problem

01

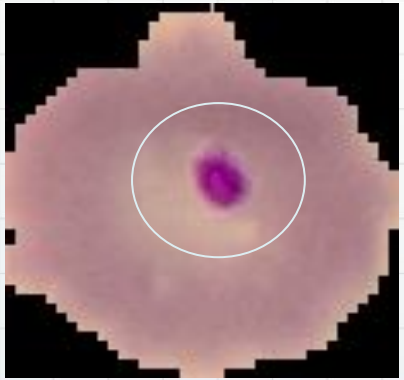
Problem Being Solved

- Classifying binary outcome: cell images that are infected with malaria or not
- No summary statistics since to data are images
- Developing a fast and reliable diagnostic test

Uninfected



Infected



Preprocessing Data

- Original dataset: 13,780 infected images & 13,780 uninfected images
- Created four copies of infected images for the entire dataset
- Created three copies of uninfected images for the entire dataset
- Normalization: rescaled images by dividing by 255

Data Set	Images in Data Set
Training Set	77,162
Validation Set	15,432
Test Set	19,291

Fit Training Set

02

Training Performance

- Evaluation Metric
  - Training Accuracy - Determine how well the model is classifying images
  - Binary Cross Entropy Loss - Determine the fit of the training and validation set
- Model is performing with a training accuracy of 96%
- Human-level accuracy could be similar or better, but would be time consuming and requires medical training
- Logistic Regression (without hidden layers) : 68%

Training Different Models

```
#Model 1
model = Sequential([
                    Conv2D(filters=16, kernel_size=(3,3), strides=(1,1), padding='valid', activation = 'relu',
                           input_shape=(64,64,3)),
                    MaxPooling2D(pool_size=(3,3), strides=(1,1), padding='valid'),
                    Flatten(),
                    Dense(64, activation = 'relu'),
                    Dense(1, activation = 'sigmoid')
```

Accuracy: 0.9645
Loss: 0.100

```
#Model 2
model4 = Sequential([
                    Conv2D(filters=16, kernel_size=(3,3), strides=(1,1), padding='valid', activation = 'relu',
                           input_shape=(64,64,3)),
                    MaxPooling2D(pool_size=(3,3), strides=(1,1), padding='valid'),
                    Flatten(),
                    Dense(64, activation = 'relu'),
                    Dense(10, activation = 'relu'),
                    Dense(1, activation = 'sigmoid')
```

Accuracy: 0.8080
Loss: 0.3943

```
#Model 3
model4 = Sequential([
                    Conv2D(filters=16, kernel_size=(3,3), strides=(1,1), padding='valid', activation = 'relu',
                           input_shape=(64,64,3)),
                    MaxPooling2D(pool_size=(3,3), strides=(1,1), padding='valid'),
                    Flatten(),
                    Dense(32, activation = 'relu'),
                    Dense(1, activation = 'sigmoid')
```

Accuracy: 0.9726
Loss: 0.0772

Training Different Optimizers

Optimizer	Accuracy	Loss
Adam	0.9726	0.0772
Gradient Descent with Momentum	0.5710	0.6831
RMSprop	0.5710	0.6841

Training Different Epochs

Epochs	Accuracy	Loss
8	0.9726	0.0772
12	0.9640	0.0979
16	0.9886	0.0323

Training Different Weights

Initializer	Accuracy	Loss
Zeros	0.9583	0.1062
Random	0.9803	0.0584
He	0.9473	0.1564

Best Model

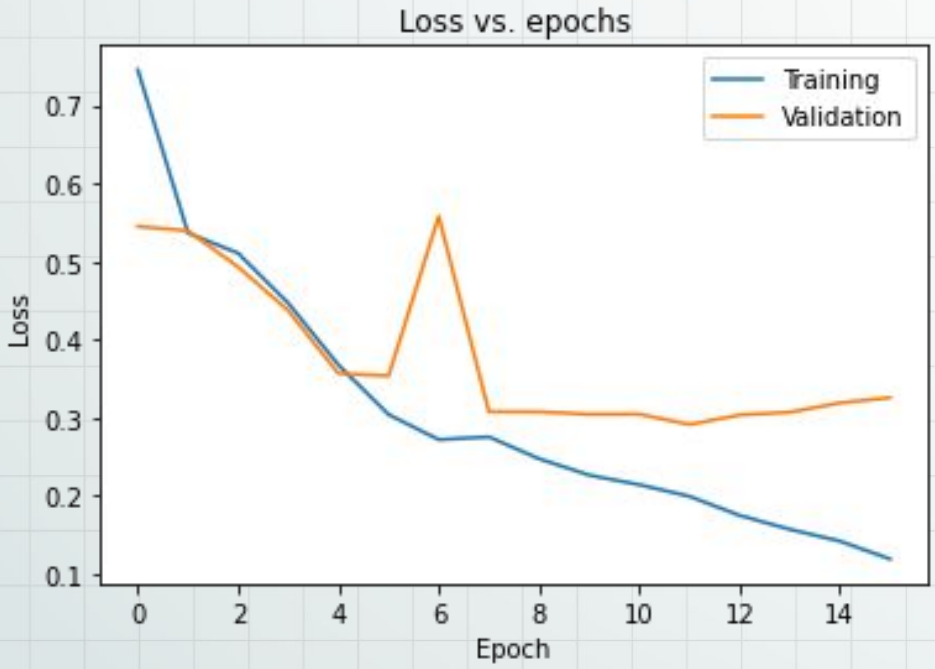
```
#Model 3
model4 = Sequential([
                    Conv2D(filters=16, kernel_size=(3,3), strides=(1,1), padding='valid', activation = 'relu',
                           input_shape=(64,64,3)),
                    MaxPooling2D(pool_size=(3,3), strides=(1,1), padding='valid'),
                    Flatten(),
                    Dense(32, activation = 'relu'),
                    Dense(1, activation = 'sigmoid')
```

Optimizer	Adam
Epochs	16
Weight Initializations	Random

Fit Validation Set

03

Model Fit on Validation Set



- Some overfitting
- Loss for validation set increases with more epochs
- Accuracy: 0.9014
- Loss: 0.3072

L2 Regularization

L2 Penalty Rate	Accuracy	Loss	Fit
0.1	0.7022	0.5519	No Overfit
0.05	0.7053	0.5454	Overfit
0.01	0.9432	0.2106	No Overfit
0.005	0.9451	0.1870	Overfit
0.001	0.9424	0.1889	Overfit

Dropout Regularization

Dropout Rate	Accuracy	Loss	Fit
0.1	0.9428	0.1859	Slight Overfit
0.05	0.5701	0.6833	No Overfit
0.01	0.5909	0.6908	Overfit
0.2	0.9419	0.1817	Slight Overfit
0.15	0.9422	0.2250	Overfit

L2 and Dropout Regularization

Dropout Rate	L2 Penalty Rate	Accuracy	Loss	Fit
0.1	0.005	0.9400	0.2087	Slight Overfit
0.1	0.001	0.9427	0.2201	Slight Overfit
~~0.2~~	~~0.005~~	~~0.9423~~	~~0.1765~~	~~No Overfit~~
~~0.2~~	~~0.001~~	~~0.5701~~	~~0.6857~~	~~No Overfit~~
0.1	0.01	0.5701	0.6899	No Overfit

Batch Normalization

	Accuracy	Loss	Fit
Default	0.9851	0.0417	Overfitting
Momentum = 0.95	0.7248	0.5152	No Overfit
Beta = random_normal Gamma = constant	0.9880	0.0336	Slight Overfit
Epsilon = 0.005 Beta = random_normal Gamma = constant	0.9825	0.0480	Overfitting
Epsilon = 0.01 Beta = random_normal Gamma = constant	0.9809	0.0531	Slight Overfit

Batch Normalization and Dropout Regularization

Batch Normalization	Dropout Regularization	Accuracy	Loss	Fit
Default	0.1	0.9726	0.0762	Slight overfit
Beta = random_normal Gamma = constant	0.1	0.9797	0.0564	Overfitting
Default	0.2	0.9687	0.0894	No Overfit
Beta = random_normal Gamma = constant	0.2	0.9706	0.0811	Slight Overfit
Epsilon = 0.01 Beta = random_normal Gamma = constant	0.2	0.9720	0.0802	Slight Overfit

Batch Normalization, Dropout Regularization, and L2 Regularization

Batch Normalization	Dropout Regularization	L2 Regularization	Accuracy	Loss	Fit
Default	0.2	0.005	0.5702	4.6547	Large Overfit
Default	0.1	0.005	0.9298	0.2163	Overfit
Beta = random_normal Gamma = constant	0.2	0.005	0.9230	0.2370	Overfit
Beta = random_normal Gamma = constant	0.2	0.001	0.9389	0.2236	Overfit
Beta = random_normal Gamma = constant	0.1	0.005	0.9092	0.3870	Overfit

Early Stopping

	Accuracy	Loss	Fit
Min_delta = 0.01 Patience = 5	0.8836	0.3126	Small Overfit
Min_delta = 0.05 Patience = 5	0.9387	0.2078	Small Overfit
Min_delta = 0.01 Patience = 7	0.9313	0.2521	Overfit
Min_delta = 0.05 Patience = 7	0.9388	0.2126	Small Overfit
Min_delta = 0.1 Patience = 5	0.9417	0.1845	No Overfit

What Worked Best

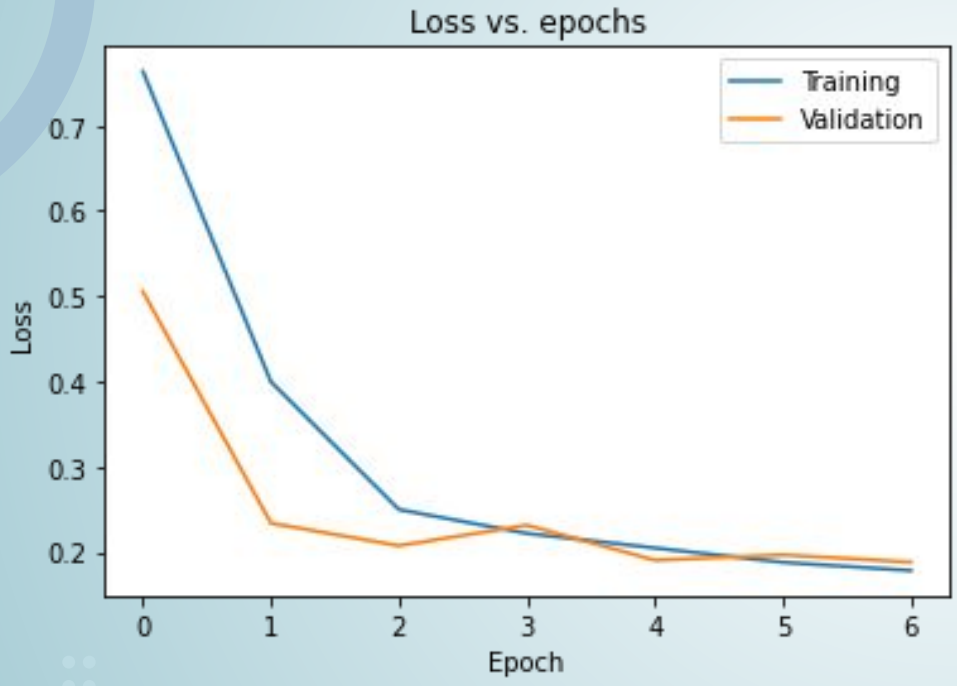
Model performed best when …

- Dropout Regularization
  - 0.1, 0.2
- L2 Regularization
  - 0.01
- Batch-Normalization
  - Beta_Initializer = RandomNormal & Gamma_Initializer = Constant
  - Epsilon = 0.01, Beta_Initializer = RandomNormal & Gamma_Initializer = Constant
- Early Stopping
  - Min_delta = 0.1
  - Patience = 5

Finding Best Model and Business Implications

04

Best Model



L2 Regularization = 0.005
Dropout Regularization = 0.2
Early Stopping

Business Implications

Cases & Deaths

There are about 200 million cases worldwide, and about 400,000 deaths caused by malaria

Diagnostic

~170 million blood films are examined every year to manually count for malaria

Hospital & Pharmaceutical

Reduce the cost of malaria diagnostic for medical and research purposes

Standardization

Standardizing malaria diagnostic to accurately diagnose patients in a timely manner

Source: National Library of Medicine

Impact

A study by the World Health Organization found that preventive measures such as early detection is proven to reduce deaths by 20%

As a result of our model, we are looking at an upwards estimate of 80,000 lives being saved with proper implementation

World Health Organization

Source: World Health Organization