Analysis of Image Processing Techniques in Malaria Cell Classification using Convolutional Neural Networks

•••

By: Rayhaan Rasheed, William Noone, Samuel Aboagye
DATS 6203 - Machine Learning II

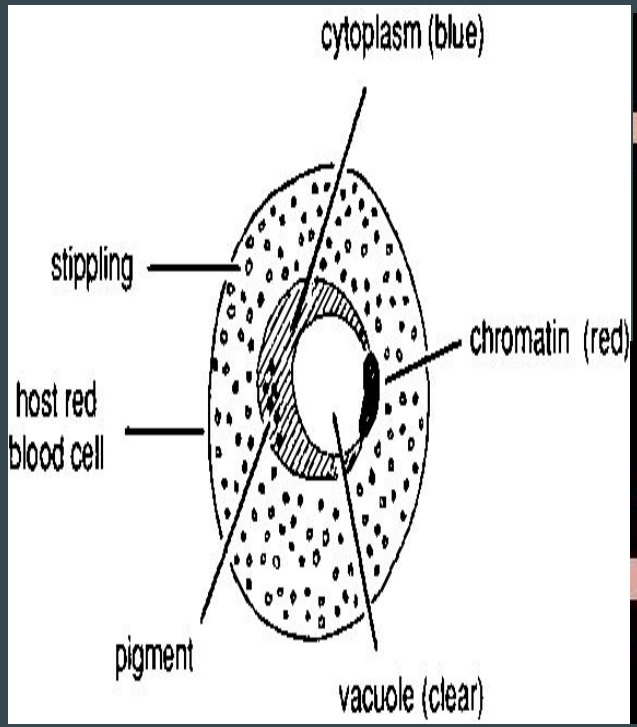
Overview

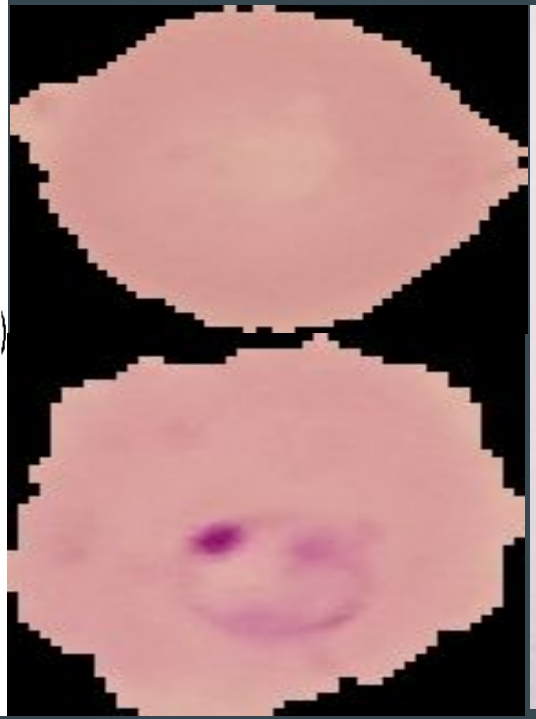
1. Introduction
2. Background
3. Data
4. Image Processing
5. Caffe Setup and Preprocessing
6. Network Architecture
7. Results
8. Conclusion

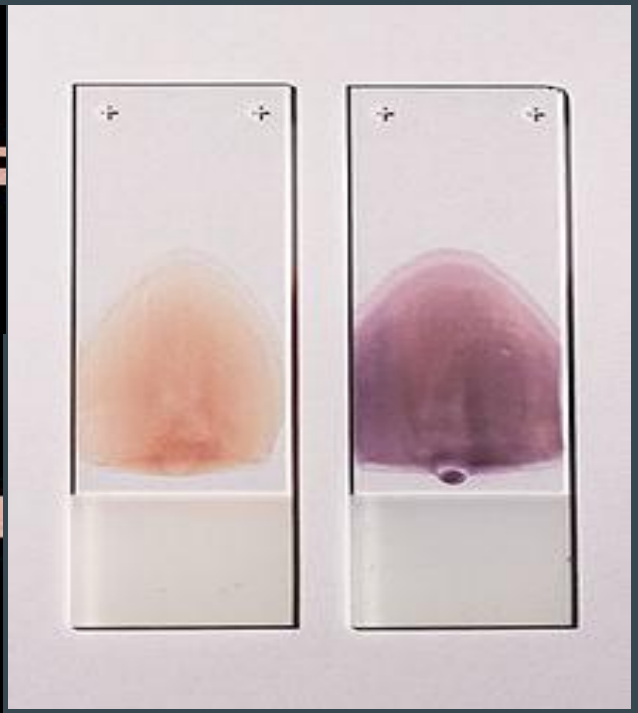
Introduction

- What is Malaria?
  - blood borne disease caused by a parasite
- Why is it important?
  - 216 Million people infected and 445,000 died in 2016
  - Most cases are found in underdeveloped areas such as Sub-Saharan Africa and India
- How can we detect it?
  - Presence is confirmed by examining infected blood under a microscope
  - Diagnosis is made based on the physician/technician's assessment
  - Long process with low accuracy
  - Certain techniques are used to help but it is still up to the person looking through the microscope

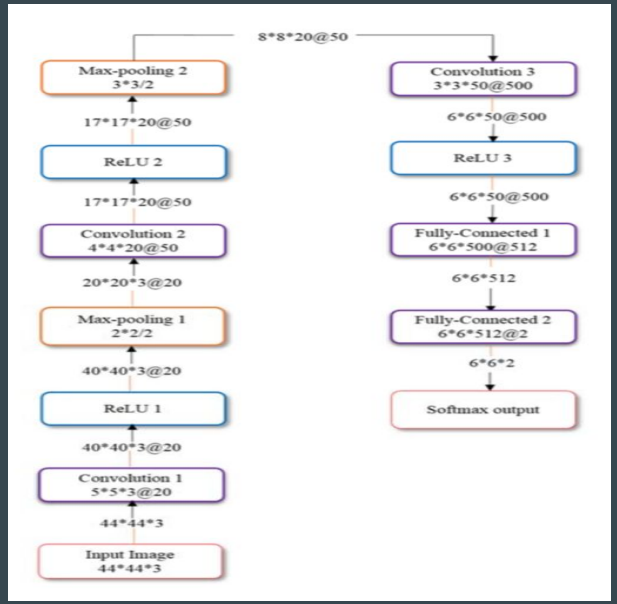
Cell Image


cytoplasm (blue)
stippling
chromatin (red)
host red blood cell
pigment
vacuole (clear)

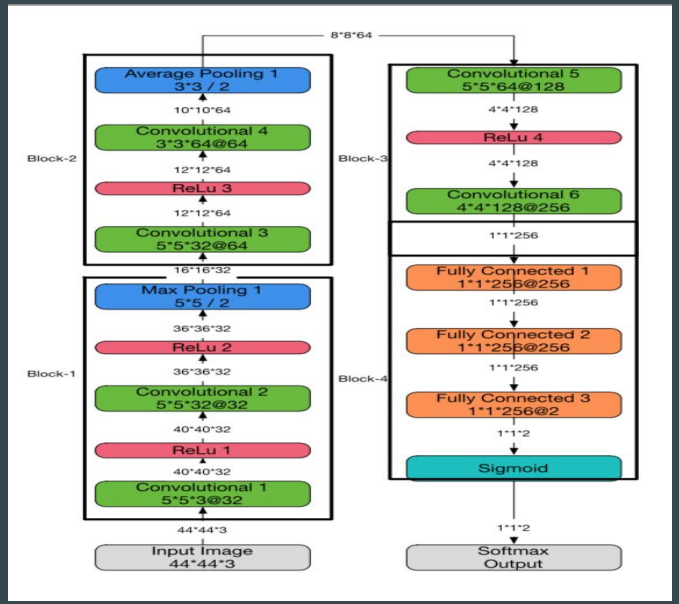




Background



Krishnan et al 2017



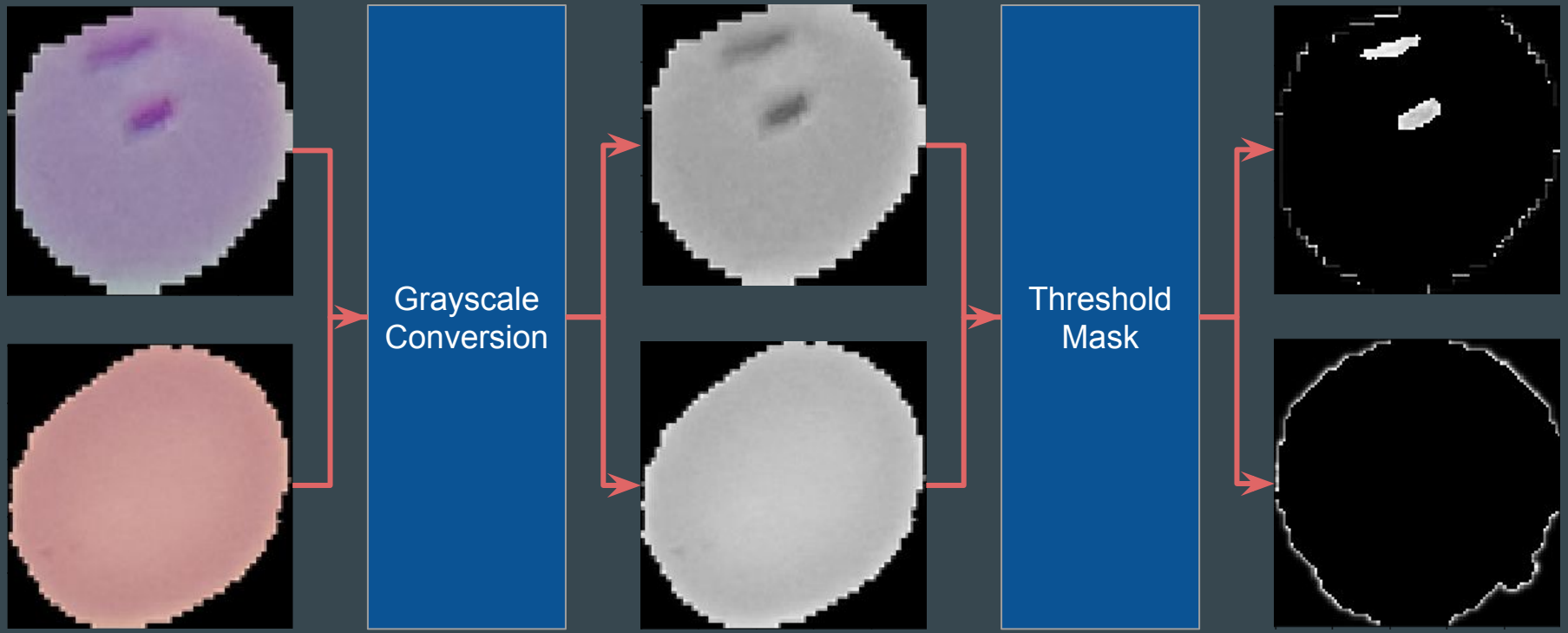
Liang et al 2016

Data

- Images pulled from the U.S. National LIbrary of Medicine's Communications Engineering Branch
- 27,558 images of single red blood cells from Giemsa stained thin blood slides
  - Chittagong Medical College hospital in Bangladesh
  - *P. falciparum* parasite
  - Each image is roughly the same size and is given one of
  - Two classes: "Uninfected" or "Parasitized"

Source: NIH Website: https://ceb.nlm.nih.gov/repositories/malaria-datasets

Image Processing



Implementation: Preprocessing & Caffe Build

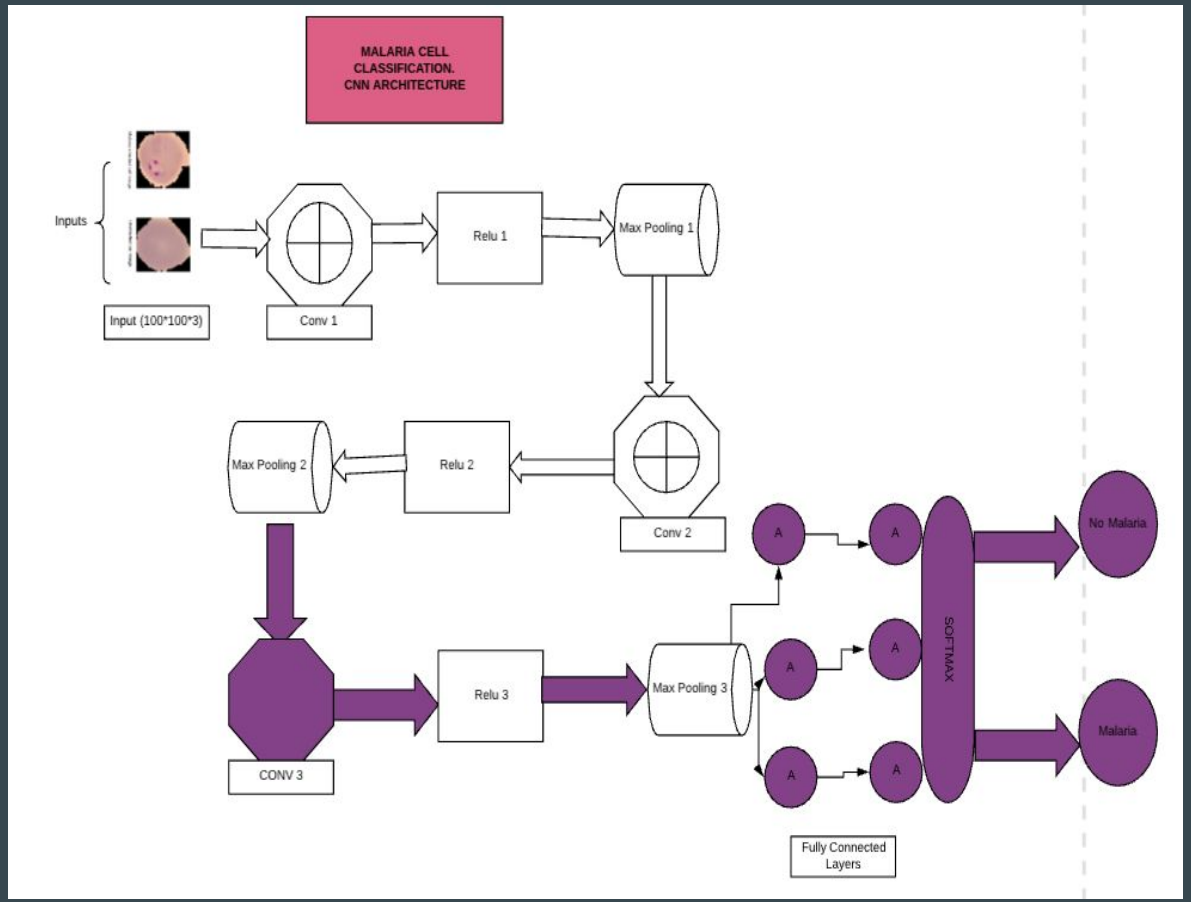
Preprocessing and System Admin Tasks

1. NIH Data Download & File Directory Operations
2. Format Unprocessed Images
3. Format Processed Images
4. Create Training and Validation Directories (70/30)
5. Create .lmdb Packages

CNN Build & Training in Caffe

1. Implement CNN for alternative layer & blob configurations
2. Research hyper-parameter settings for performance tuning.
3. Compare results between models and processed vs unprocessed image groups

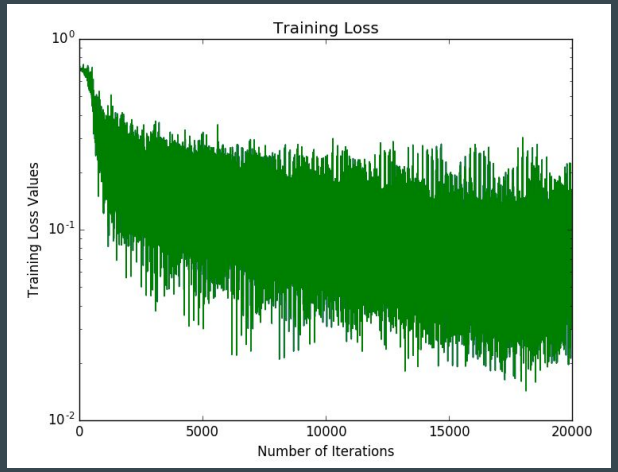
3 LAYER CNN

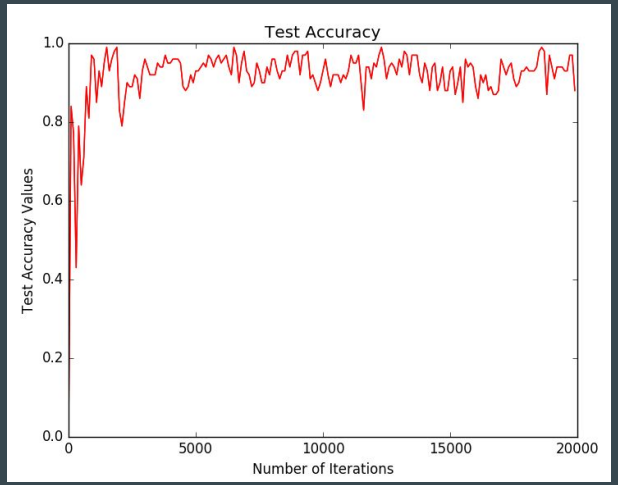

MALARIA CELL CLASSIFICATION. CNN ARCHITECTURE
Inputs
Input (100*100*3)
Conv 1
Relu 1
Max Pooling 1
Conv 2
Relu 2
Max Pooling 2
CONV 3
Relu 3
Max Pooling 3
A
SOFTMAX
No Malaria
Malaria
Fully Connected Layers

Model 1 - Unprocessed Images

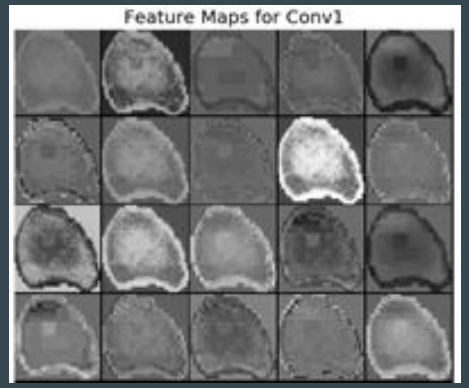
- 3 layer CNN
- 20 FMs per layer
- kernel_size = 2, stride =2 across all layers
- (2) fully connected layers
- Soft-max output function.

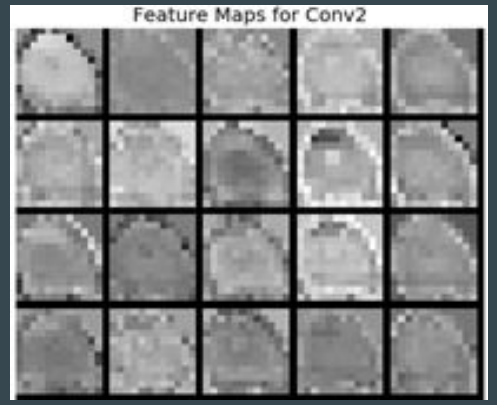
Layer	Shape	Kernel_size	Stride
data	(100, 3, 100, 100)	-	-
label	(100,)	-	-
conv1	(100, 20, 50, 50)	2	2
relu1	(100, 20, 50, 50)	-	-
pool1	(100, 20, 25, 25)	2	2
conv2	(100, 20, 12, 12)	2	2
relu2	(100, 20, 12, 12)	-	-
pool2	(100, 20, 6, 6)	2	2
conv3	(100, 20, 3, 3)	2	2
relu3	(100, 20, 3, 3)	-	-
pool3	(100, 20, 2, 2)	2	2
fc1	(100, 10)	-	-
relu4	(100, 10)	-	-
fc2	(100, 2)	-	-
loss	()	-	-
Weight, Bias	**Shape**	**Measure**	**Value**
conv1	(20, 3, 2, 2) (20,)	Mean Train_Loss	0.13354048
conv2	(20, 20, 2, 2) (20,)	Std Dev Train_Loss	0.109245778
conv3	(20, 20, 2, 2) (20,)	**Mean Test_Acc**	**0.917450002**
fc1	(10, 80) (10,)	Std Dev Test_Acc	0.087166495
fc2	(2, 10) (2,)		

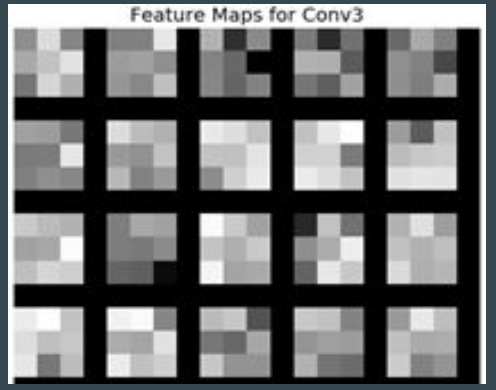


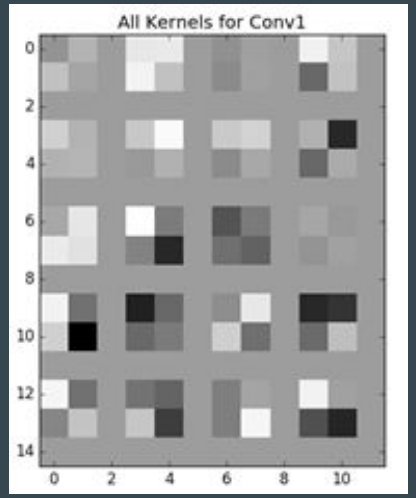


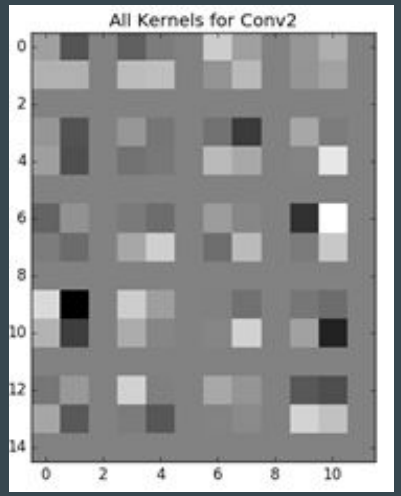
Model 1 - Unprocessed Images FM & Kernels

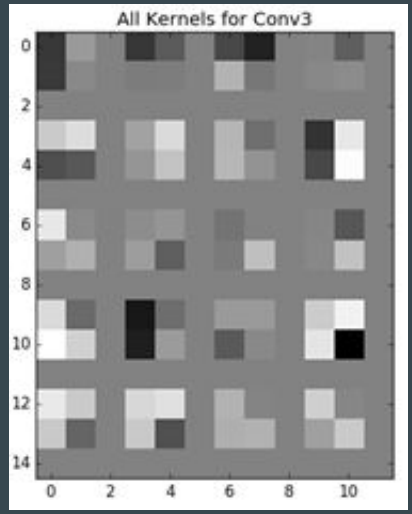








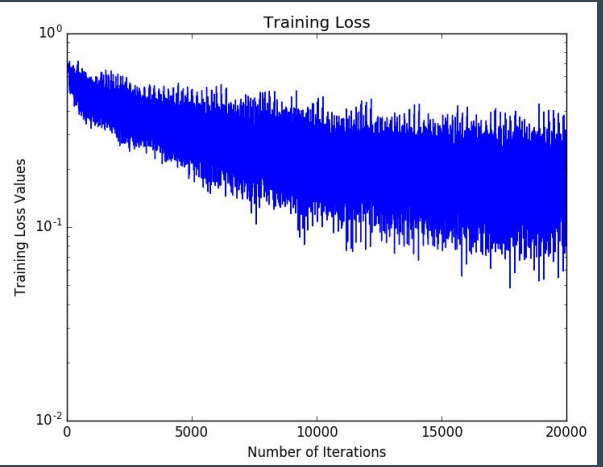


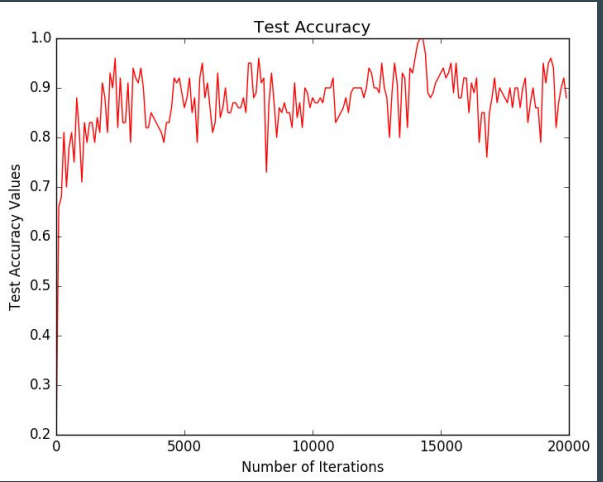


Model 1 - Processed Images

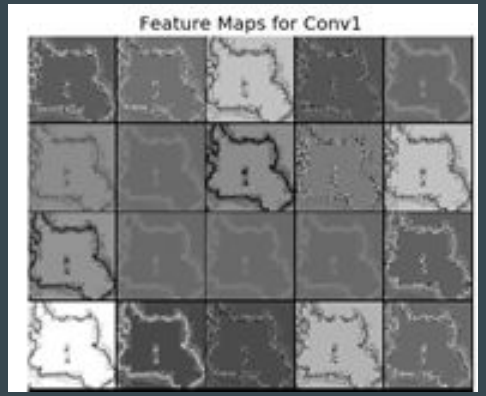
- 3 layer CNN
- 20 FMs per layer
- kernel_size = 2, stride =2 across all layers
- (2) fully connected layers
- Soft-max output function.

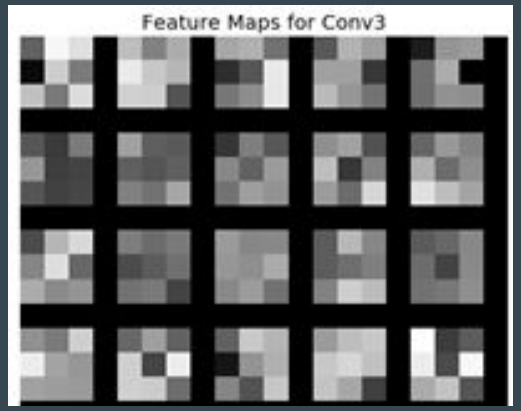
Layer	Shape	Kernel_size	Stride
data	(100, 1, 100, 100)	-	-
label	(100,)	-	-
conv1	(100, 20, 50, 50)	2	2
relu1	(100, 20, 50, 50)	-	-
pool1	(100, 20, 25, 25)	2	2
conv2	(100, 20, 12, 12)	2	2
relu2	(100, 20, 12, 12)	-	-
pool2	(100, 20, 6, 6)	2	2
conv3	(100, 20, 3, 3)	2	2
relu3	(100, 20, 3, 3)	-	-
pool3	(100, 20, 2, 2)	2	2
fc1	(100, 10)	-	-
relu4	(100, 10)	-	-
fc2	(100, 2)	-	-
loss	()	-	-
Weight, Bias	**Shape**	**Measure**	**Value**
conv1	(20, 1, 2, 2) (20,)	Mean Train_Loss	0.265589719
conv2	(20, 20, 2, 2) (20,)	Std Dev Train_Loss	0.111700857
conv3	(20, 20, 2, 2) (20,)	**Mean Test_Acc**	**0.8718**
fc1	(10, 80) (10,)	Std Dev Test_Acc	0.072916115
fc2	(2, 10) (2,)		

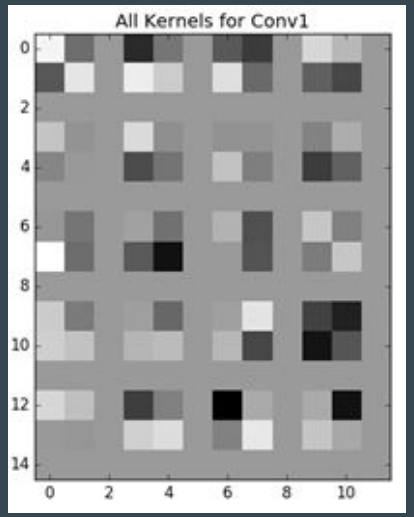


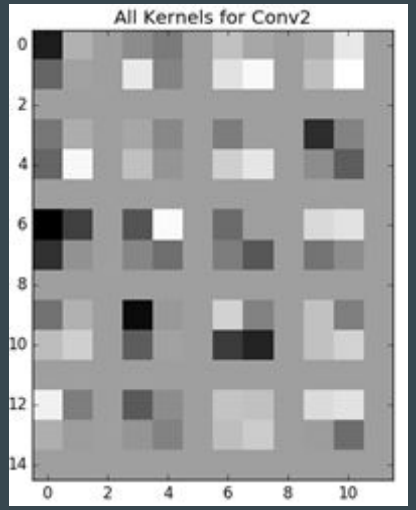


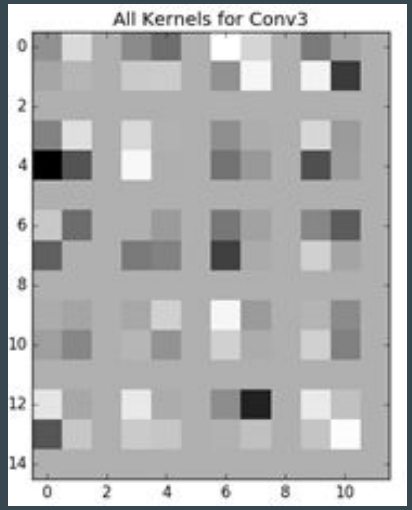
Model 1 - Processed Images FM & Kernels



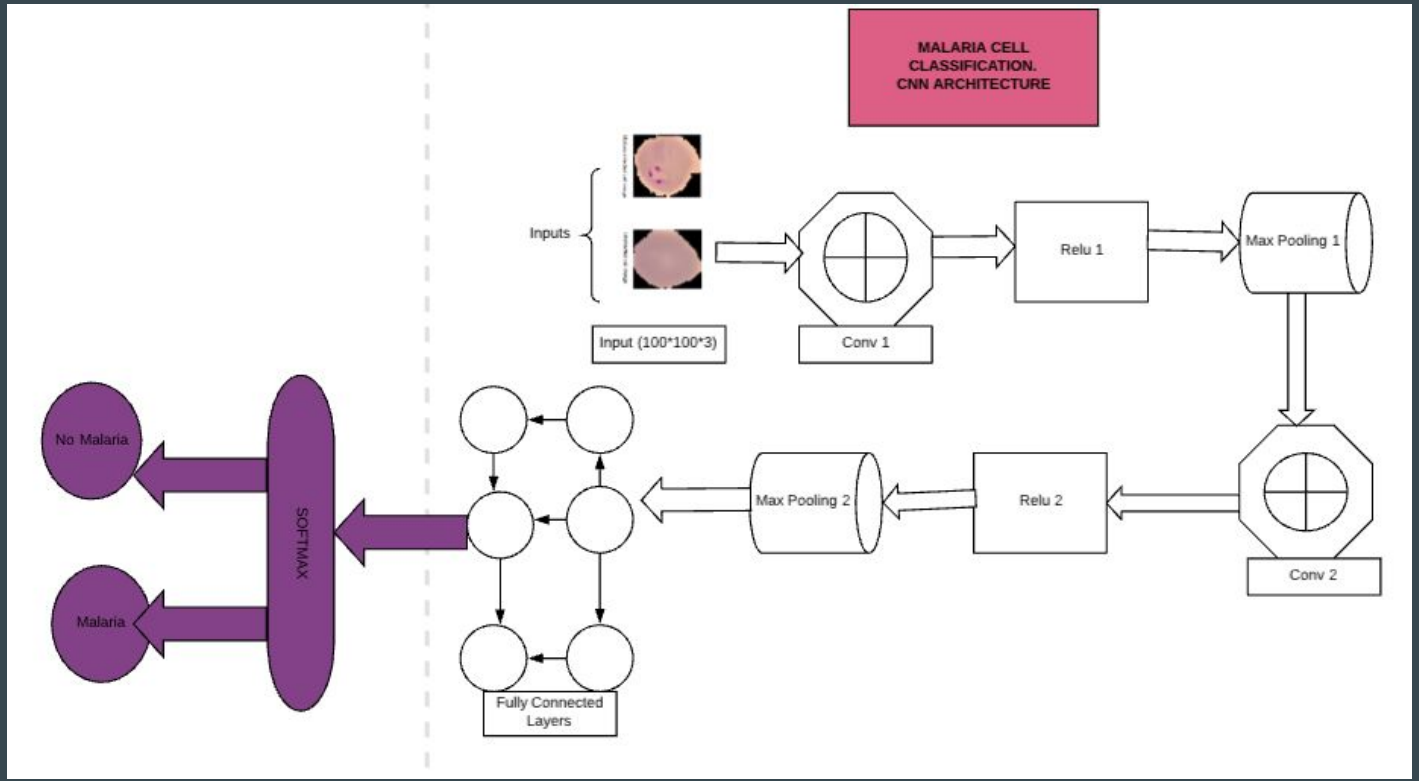








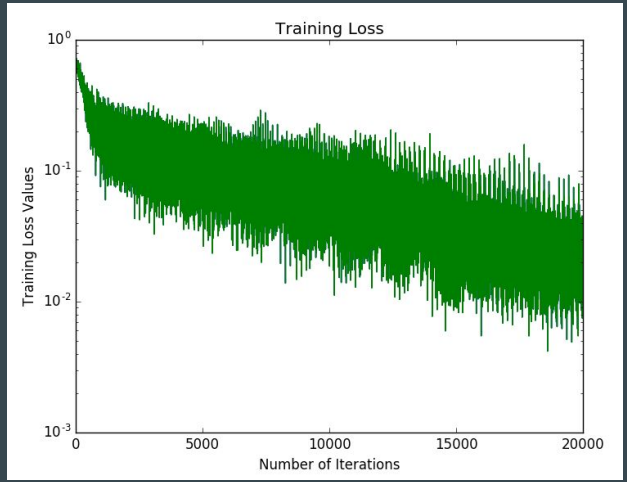
2 LAYER CNN

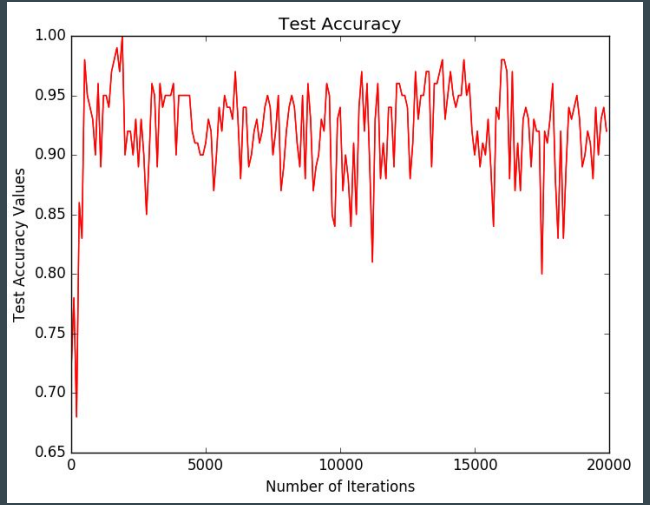

MALARIA CELL CLASSIFICATION. CNN ARCHITECTURE
Inputs
Input (100*100*3)
Conv 1
Relu 1
Max Pooling 1
Conv 2
Relu 2
Max Pooling 2
Fully Connected Layers
SOFTMAX
No Malaria
Malaria

Model 2 - Unprocessed Images

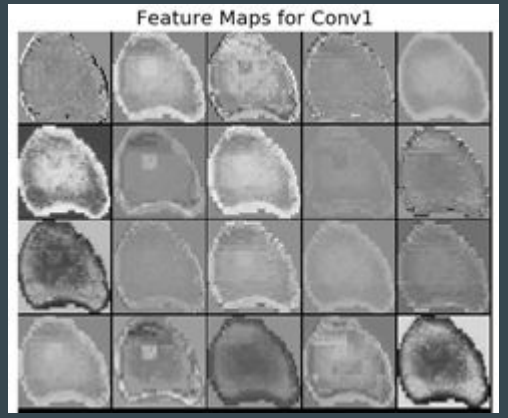
- 3 layer CNN
- 20 FMs per layer
- kernel_size = 2, stride =2 across all layers
- (2) fully connected layers
- Soft-max output function.

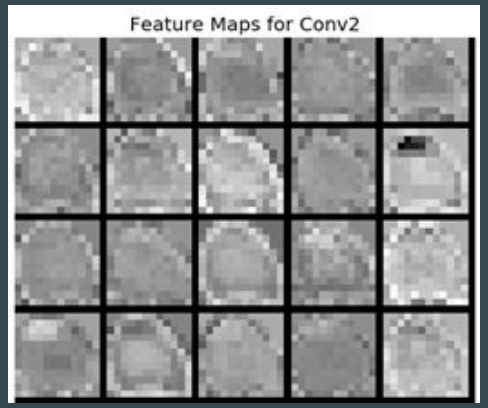
Layer	Shape	Kernel_size	Stride
data	(100, 3, 100, 100)	-	-
label	(100,)	-	-
conv1	(100, 30, 50, 50)	2	2
relu1	(100, 30, 50, 50)	-	-
pool1	(100, 30, 25, 25)	2	2
conv2	(100, 30, 12, 12)	2	2
relu2	(100, 30, 12, 12)	-	-
pool2	(100, 30, 6, 6)	2	2
fc1	(100, 10)	-	-
relu3	(100, 10)	-	-
fc2	(100, 2)	-	-
loss	()	-	-
Weight, Bias	**Shape**	**Measure**	**Value**
conv1	(30, 3, 2, 2) (30,)	Mean Train_Loss	0.08991731
conv2	(30, 30, 2, 2) (30,)	Std Dev Train_Loss	0.087200941
fc1	(10, 1080) (10,)	**Mean Test_Acc**	**0.919999999**
fc2	(2, 10) (2,)	Std Dev Test_Acc	0.043370498

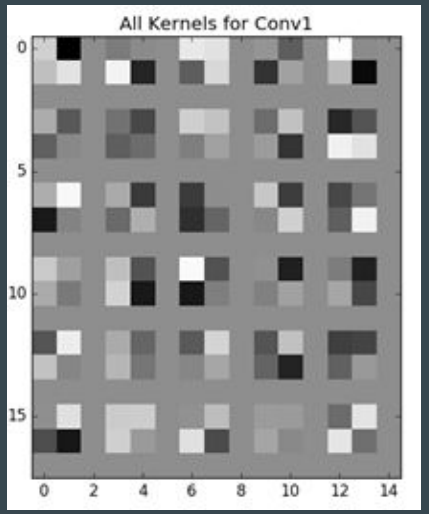


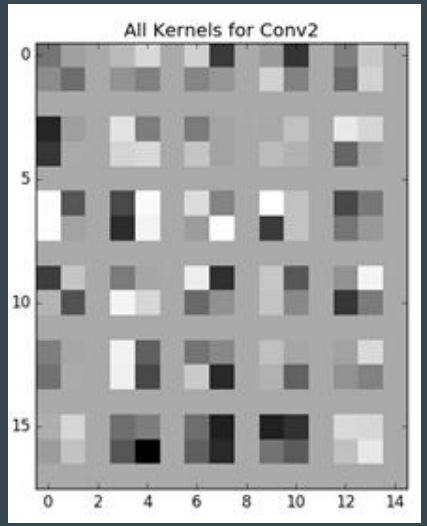


Model 2 - Unprocessed Images FM & Kernels





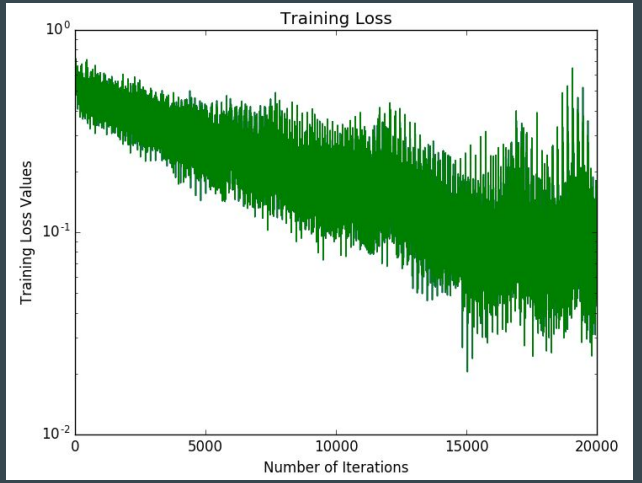


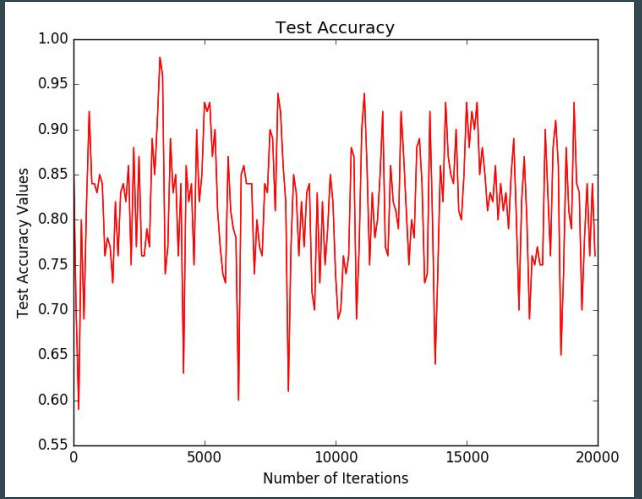


Model 2 - Processed Images

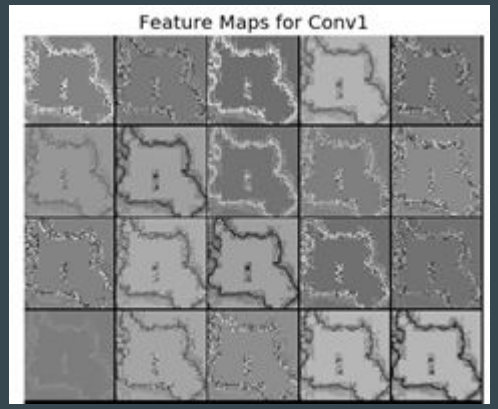
- 2 layer CNN
-30 FMs per layer
-kernel_size = 2, stride =2 across all layers
-2 fully connected layers
-SoftMax output function.

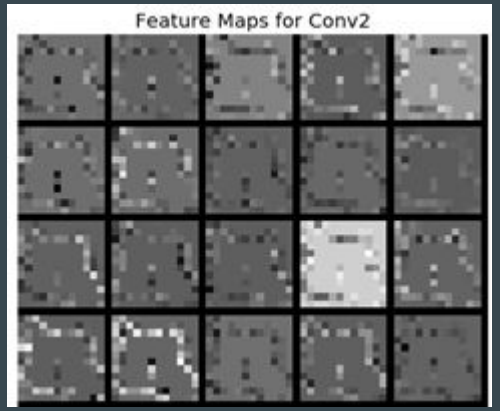
Layer	Shape	Kernel_size	Stride
data	(100, 1, 100, 100)	-	-
label	(100,)	-	-
conv1	(100, 30, 50, 50)	2	2
relu1	(100, 30, 50, 50)	-	-
pool1	(100, 30, 25, 25)	2	2
conv2	(100, 30, 12, 12)	2	2
relu2	(100, 30, 12, 12)	-	-
pool2	(100, 30, 6, 6)	2	2
fc1	(100, 10)	-	-
relu3	(100, 10)	-	-
fc2	(100, 2)	-	-
loss	()	-	-
Weight, Bias	**Shape**	**Measure**	**Value**
conv1	(30, 1, 2, 2) (30,)	Mean Train_Loss	0.213966045
conv2	(30, 30, 2, 2) (30,)	Std Dev Train_Loss	0.124579427
fc1	(10, 1080) (10,)	**Mean Test_Acc**	**0.814999998**
fc2	(2, 10) (2,)	Std Dev Test_Acc	0.07026379

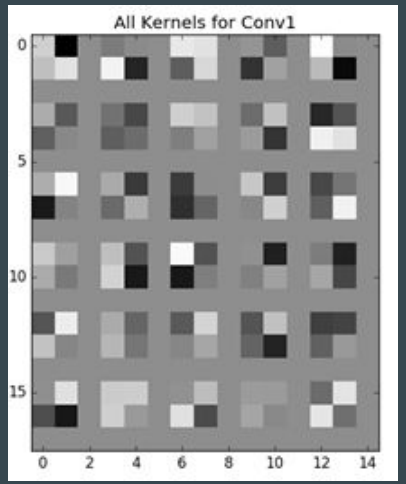


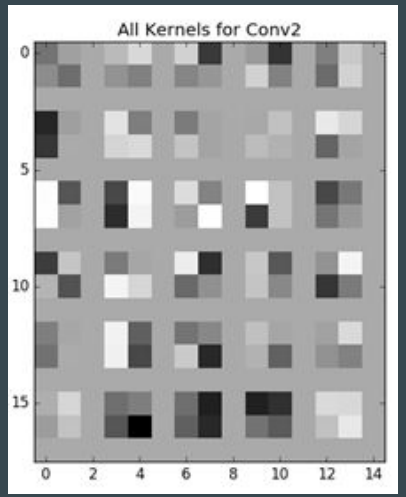


Model 2 - Processed Images FM & Kernels





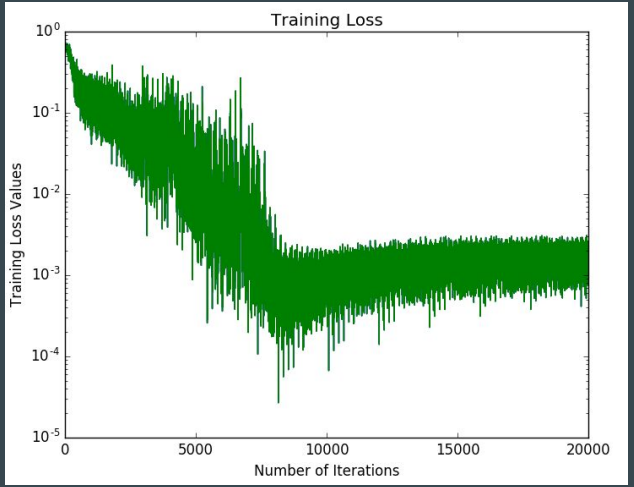


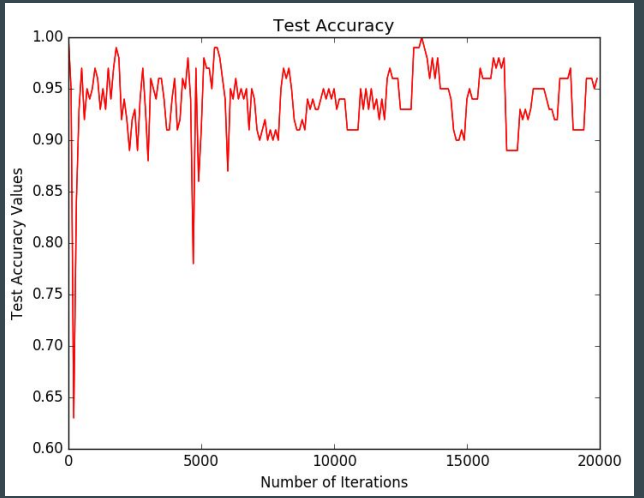


Model 3 - Unprocessed Images

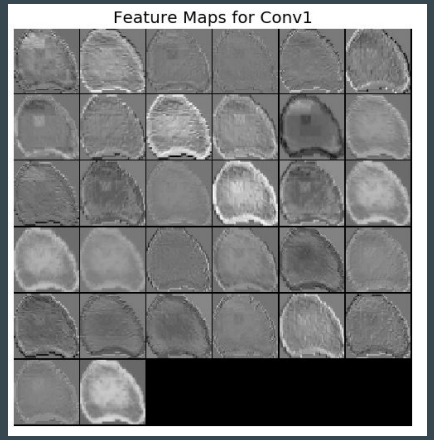
- 3 layer CNN
- L1 = 32 FMs, L2 = 32 FMs, L3 = 64 FMs
- kernel_size = 4, stride =2 across all layers
- (2) fully connected layers
- Soft-max output function.

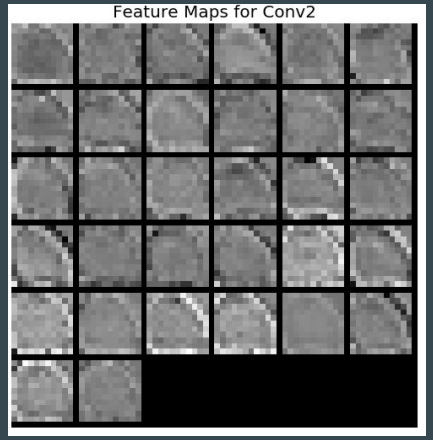
Layer	Shape	Kernel_size	Stride
data	(100, 3, 100, 100)	-	-
label	(100,)	-	-
conv1	(100, 32, 49, 49)	4	2
relu1	(100, 32, 49, 49)	-	-
pool1	(100, 32, 25, 25)	2	2
conv2	(100, 32, 11, 11)	4	2
relu2	(100, 32, 11, 11)	-	-
pool2	(100, 32, 6, 6)	2	2
conv3	(100, 64, 2, 2)	4	2
relu3	(100, 64, 2, 2)	-	-
pool3	(100, 64, 1, 1)	2	2
fc1	(100, 64)	-	-
relu4	(100, 64)	-	-
fc2	(100, 2)	-	-
loss	()	-	-
Weight, Bias	**Shape**	**Measure**	**Value**
conv1	(32, 3, 4, 4) (32,)	Mean Train_Loss	0.031756144
conv2	(32, 32, 4, 4) (32,)	Std Dev Train_Loss	0.081596197
conv3	(64, 32, 4, 4) (64,	Mean Test_Acc	0.937650002
fc1	(64, 64) (64,)	Std Dev Test_Acc	0.036850747
fc2	(2, 64) (2,)		

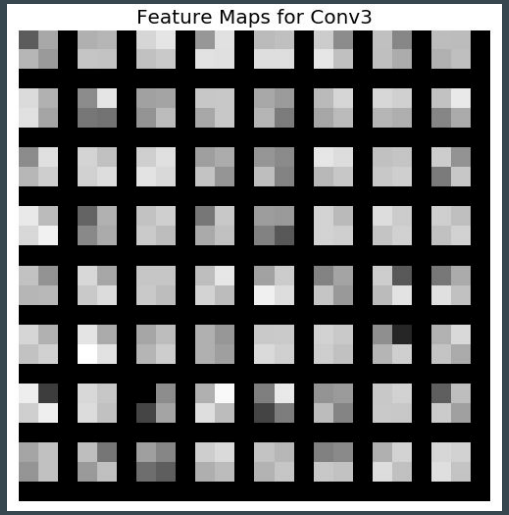


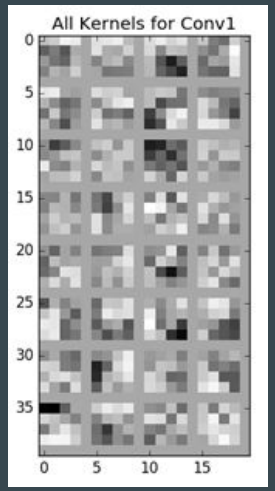


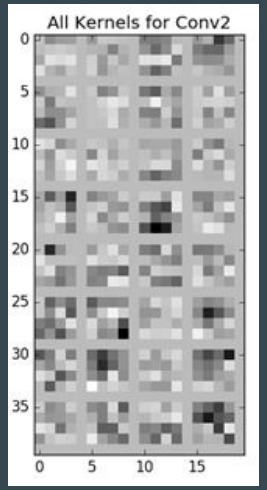
Model 3 - Unprocessed Images FM & Kernels

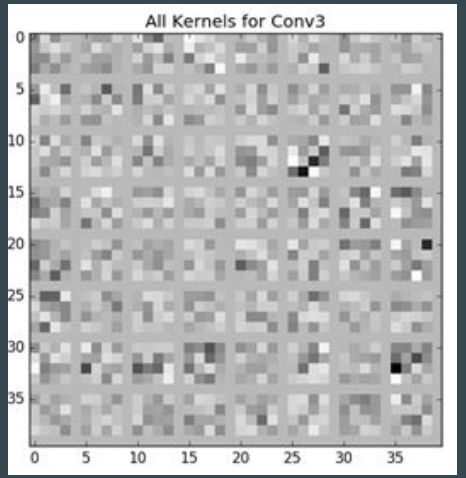












Conclusion

- Image processing techniques had a negative effect upon CNN predictive performance
    - This could be improved upon by modifying the parameters of the image filtering and masking steps.
    - This study underscores the importance of image processing on image classification tasks.
- More specialized image analysis approaches are needed
- Proper classification of malarial cells is vital for classification of specific strains.
    - a priori for identifying various strains of malaria, unique treatment can be synthesized to help save lives.

Resources

[1] Center for disease Control and Prevention (CDC). https://www.cdc.gov/malaria/diagnosis_treatment/clinicians1.html

[2] Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15 (June 2014), 1929–1958.

[3] Nasir AA, Mashor M, Mohamed Z. Segmentation based approach for detection of malaria parasites using moving k-means clustering. In: EMBS conference on biomedical engineering and sciences (IECBES). IEEE; 2012:653–8.

[4] Poostchi, M., Silamut, K., Maude, R. J., Jaeger, S., Thoma, G. Image analysis and machine learning for detecting malaria. Journal of Translational Research 2017:194-33

[5] Sampathila, N., Shet, N., Basu, A. Computational approach for diagnosis of malaria through classification of malaria parasite from microscopic image of blood smear. Journal of Biomedical Research 2018: 29-970

[6] LeCun Y, Bengio Y, Hinton G. 2015. Deep learning. Nature 521:436–444

[7] Rajaraman et al. (2018), Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. PeerJ 6:e4568; DOI 10.7717/peerj.4568

[8] Krishnan S., Antani, S., Jaeger, S., Visualizing Deep Learning Activations for Improved Malaria Cell Classification.

[9] Liang et al. (2016), CNN-Based Image Analysis for Malaria Diagnosis. In: EMBS conference on biomedical engineering and sciences (IECBES). IEEE