Image Classification – Milestone Report-1

Objective:

- To reduce the burden for microscopists in resource-constrained regions and improve diagnostic accuracy
- Build an Image Classification model to obtain highest level of accuracy
- Making the model as small and computationally efficient as possible

Introduction:

Overview:

- The dataset contains 2 folders - Infected - Uninfected and it has total of 27,558 images.
- An instance of how the patient-ID is encoded into the cell name is shown herewith: “P1” denotes the patient-ID for the cell labeled “C33P1thinF_IMG_20150619_114756a_cell_179.png”

Source:

- This Dataset is taken from the official NIH Website: https://ceb.nlm.nih.gov/repositories/malaria-datasets/

Tools:

- Python -3
- Jupyter Notebook
- TensorFlow
- Keras

Approach:

- Import required libraries
- Pre-processing image data
- Train and test data
- Building Model
- Compile & Training Model

Importing Libraries

```
import  os
from keras.preprocessing.image import ImageDataGenerator, array_to_img,img_to_array,load_img
from keras.preprocessing.image import ImageDataGenerator
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D
from keras.layers import Activation, Dropout, Flatten, Dense
from keras import backend as K
import cvutils
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
import cv2
from sklearn.model_selection import train_test_split
import keras
from keras.utils import plot_model
from keras.layers import BatchNormalization
from pathlib import Path
from keras.preprocessing import image
from keras.models import model_from_json
```

```
Using TensorFlow backend.
```

Pre-processing image data

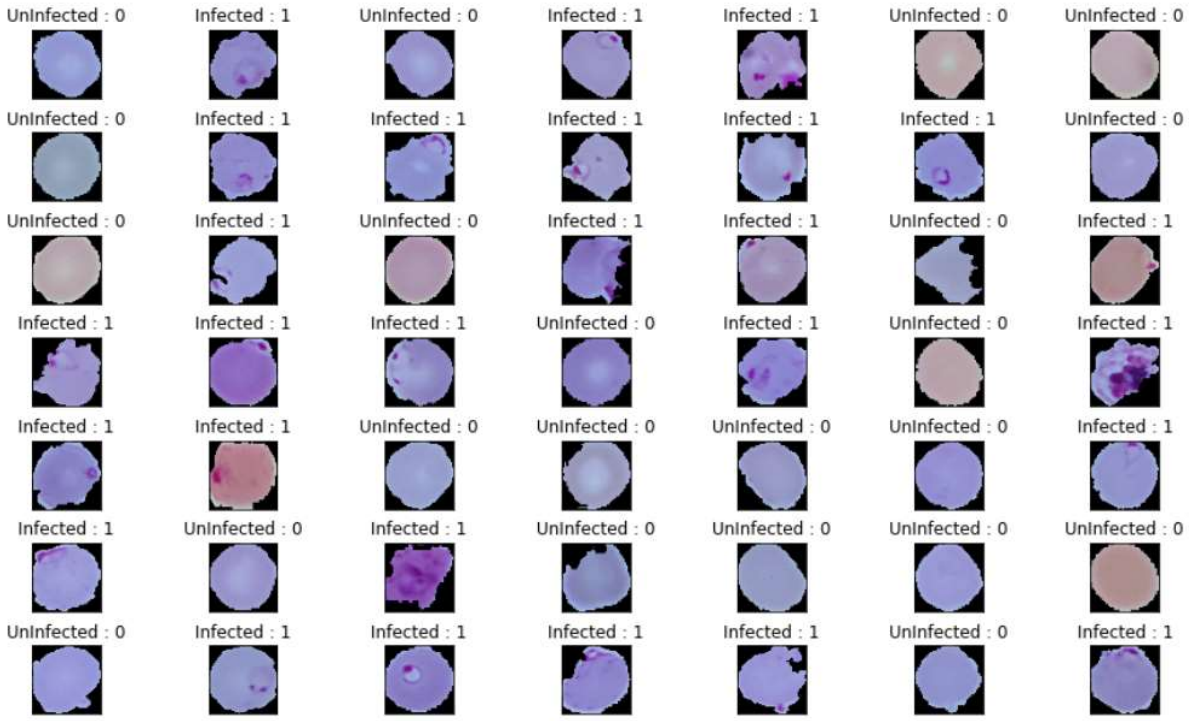
- Read the Image data using os.listdir
- Loop through all the images in both folders (Parasitized & Uninfected)
- Resize all the images with 50 x 50 pixel and convert the image into array
- Add all the resized image array to list

Train and test data

- Split the data in to train and test using sklearn.model_selection
- Normalize the data so that CNN process the data quickly and efficiently

Visualization:

Sample resized image Data



Building Model:

- Call keras Sequential model
  - Model.Sequential()
- Add convolution layer (Conv2D) to process image files
  - Params:
    - Kernel_size = 3 ,3 (window size)
    - Input shape = 50, 50, 3
    - Activation = relu
- Add max pooling so that only the higher weights will be passed on to next layer
  - Maxpooling2D: pool_size = 2,2
- Dropout is a technique where randomly selected neurons are ignored during training. They are “dropped-out” randomly. This means that their contribution to the activation of downstream neurons is temporally removed on the forward pass and any weight updates are not applied to the neuron on the backward pass
  - Dropout(0.25)

Compile and Train model:

- Compile the model and specify the loss, optimizer and metric
- Train the model and specify batch size, epochs and validation data