Developed a CNN based Deep Learning model to Detect Malaria

Preparing Data Set

Data set - https://www.kaggle.com/datasets/iarunava/cell-images-for-detecting-malaria

Created Folders with name “Cell_images”, which contains Uninfected and Parasitized folders

Import CV2

Using CV2 try to convert the image into Numpy arrays so that deep learning model can understand.

```
image_exts =['jpg','bmp','png','jpeg']
```

```
for image_class in os.listdir(data_dir):
    for image in os.listdir(os.path.join(data_dir,image_class)):
        image_path = os.path.join(data_dir,image_class,image)
        try:
            img=cv2.imread(image_path)
            tip=imghdr.what(image_path)
            if tip not in image_exts:
                print('Image not in ext list {}'.format(image_path))
                os.remove(image_path)
        except excetions as ex:
            print('Issue with image'.format(image_path))
```

Using this code to convert all images to Numpy array if they are of valid file type.

Utilized the tf.kera.dataset pipeline to create Data set for the collected images.

```
data=tf.keras.utils.image_dataset_from_directory('cell_images')
```

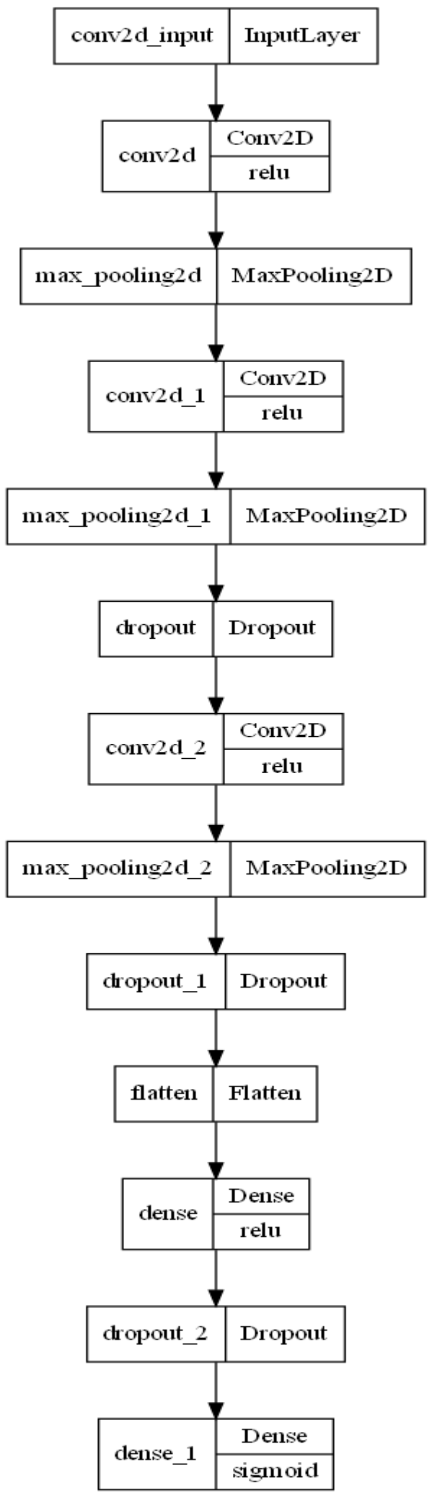
```
Found 27558 files belonging to 2 classes.
```

Pre-processing Data

We resize the image to 255*255 pixels, to standardize image size.

Assigned 60% of total data length to Train dataset, 20% for Validation dataset, 20% for Testing Dataset

Convolutional Neural Network Model



The First layer is the Input layer with Relu activation.

Relu Activation -All the negative values are defaulted to zero, and the maximum for the positive number is taken into consideration.

Next we have the Maxpooling Layer

Maxpooling - The pooling operation involves sliding a two-dimensional filter over each channel of feature map and summarising the features lying within the region covered by the filter.

Hidden Layer

Next we have another Convolutional layer to increase model complexity

Followed by a Maxpooling Layer.

We have added a Dropout layer to remove some parameters to reduce model complexity and avoid overfitting

Next we have another Convolution Layer followed by Maxpooling and Dropout Layers

Next we have Flatten –

Flatten - It is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector.

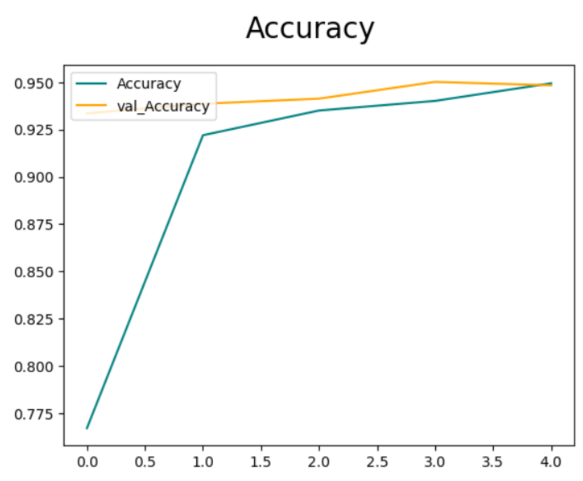
Output Layer

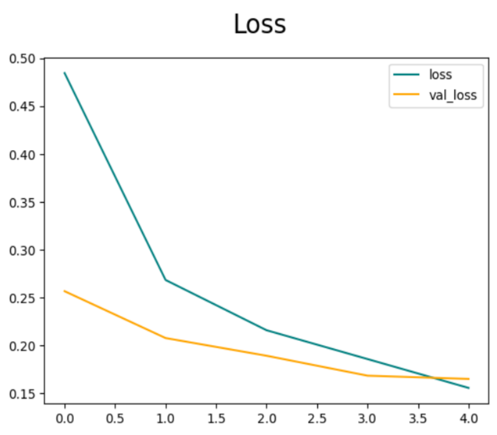
Next we have Dense Layer with 256 units followed by a Dropout.

At last we have Dense Layer with single unit and sigmoid Activation function.

Dense Layer forms the end of the model from which we can classify the image.

Train model with train and Val data. Optimizer =Adam, Loss= BinaryCrossEntropy, metrics = [‘accuracy’]



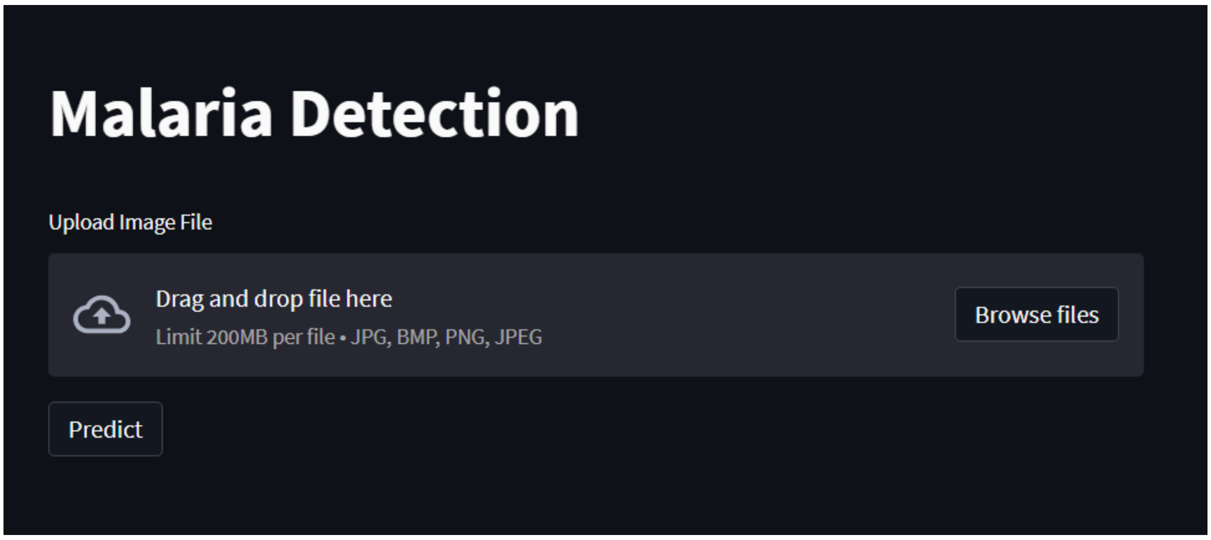


I was able to train the model with 95% Accuracy.

Save the model to use later for StreamLit App.

Streamlit App –

Designed the Streamlit App that takes input from the User and detects whether the person was infected or not.

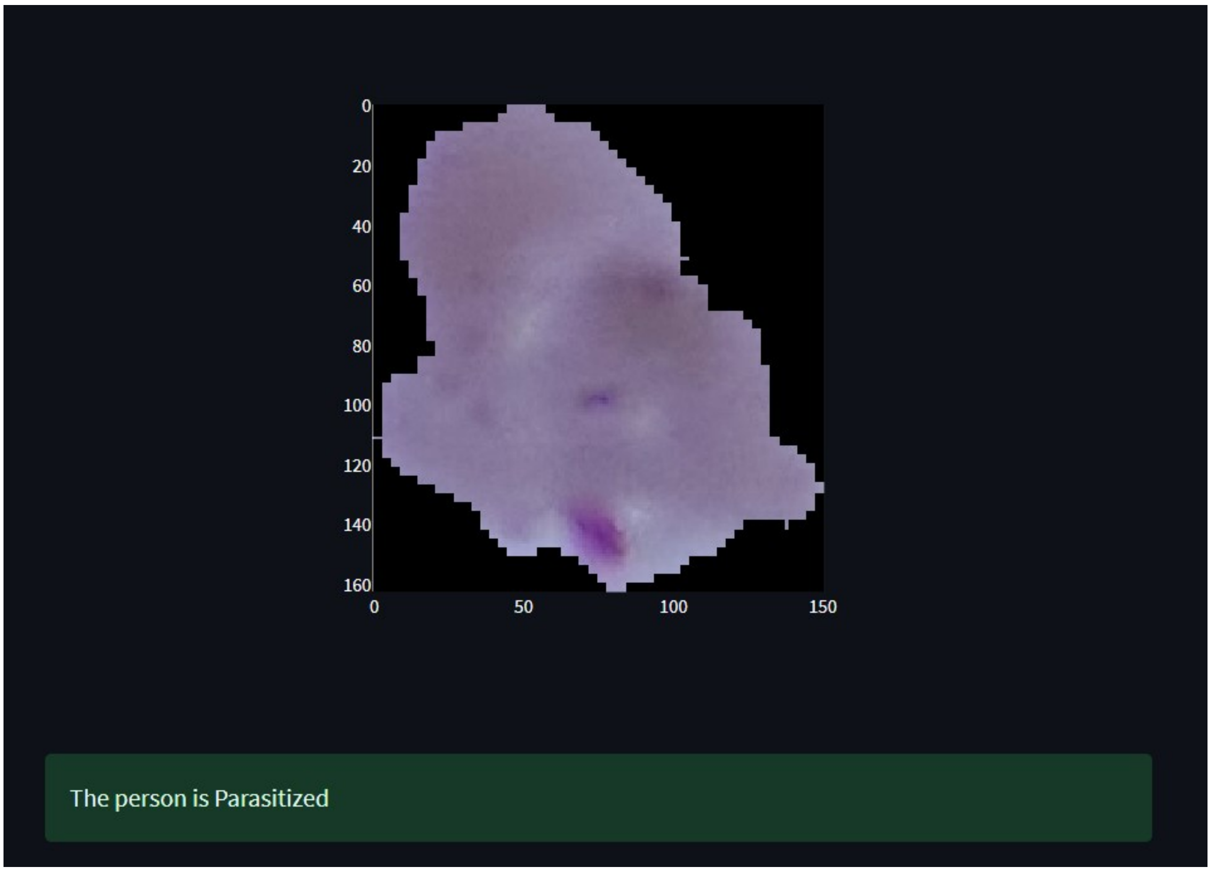


Malaria Detection

Upload Image File

Drag and drop file here

Limit 200MB per file • JPG, BMP, PNG, JPEG



The person is Parasitized

The model was correctly able to detect if the person was infected or not.

Could Not deploy to Heroku cloud as it was no longer FREE.