Alzheimer Detection Through MRI Scans
Deep Learning Project

Project Presentation

Basit Budroo	B208027
Saurabh Gawade	B208054
Om S Gaye	B208060
Tejaswini Ugale	B204106

Guide: Diptee Ghusse

DataSets

Ref: **https://www.kaggle.com/tourist55/alzheimers-dataset-4-class-of-images**

Dataset: 6400 Images (Actually 5000 images divided into train and test)

1. Trian:

   MildDemented - 717

   ModerateDemented - 52
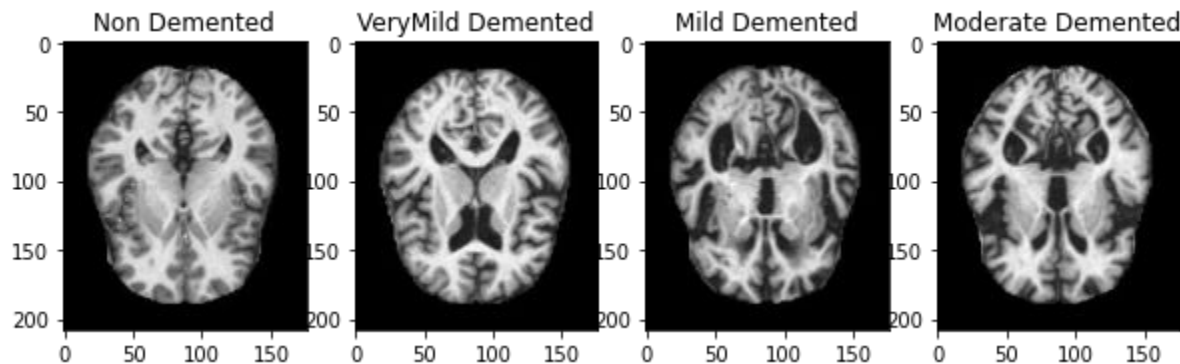
   NonDemented - 2560

   VeryMildDemented - 1792

2. Test:

   MildDemented - 179

   ModerateDemented - 12

   NonDemented - 640

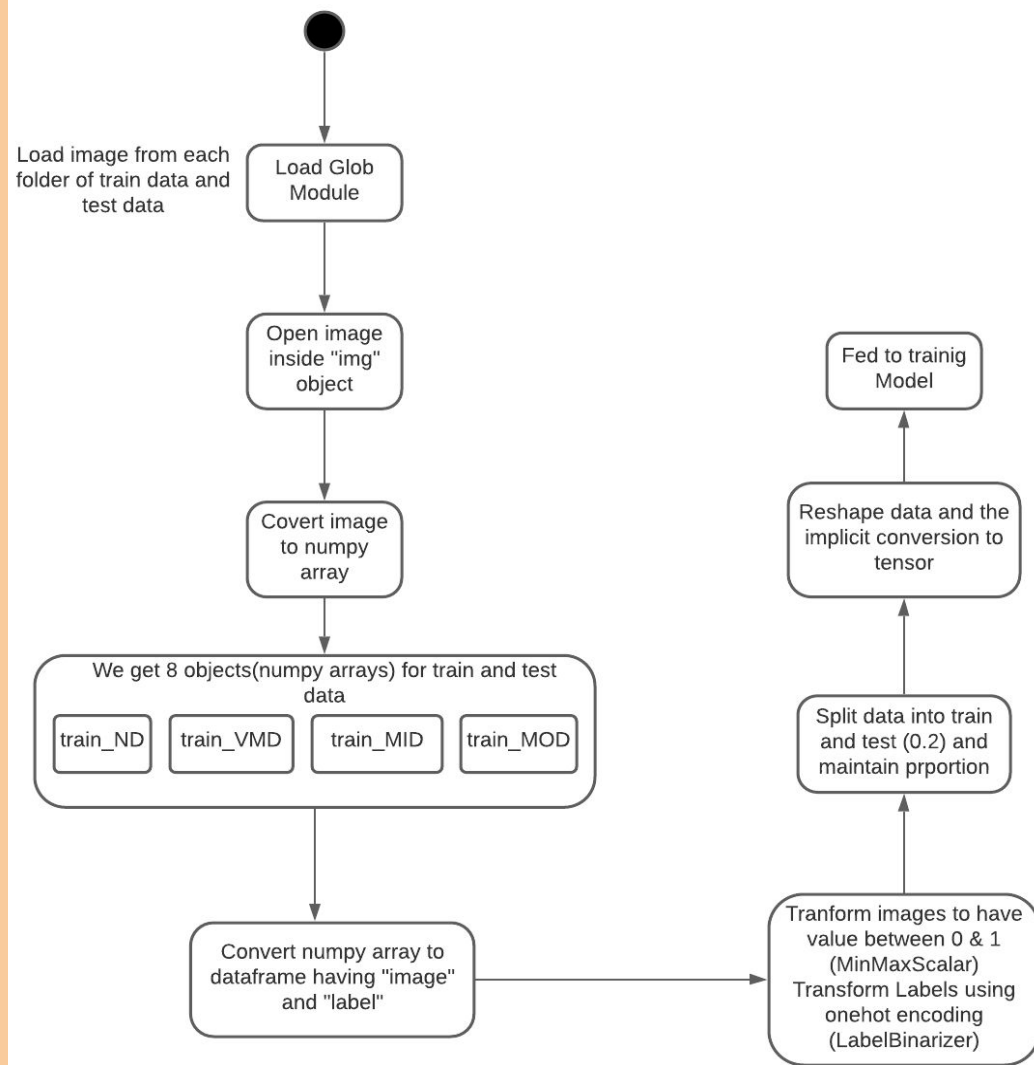   VeryMildDemented - 448

Example from "train" data
Each image has single channel and pixel value of 0-255 and dimension 176 x 208
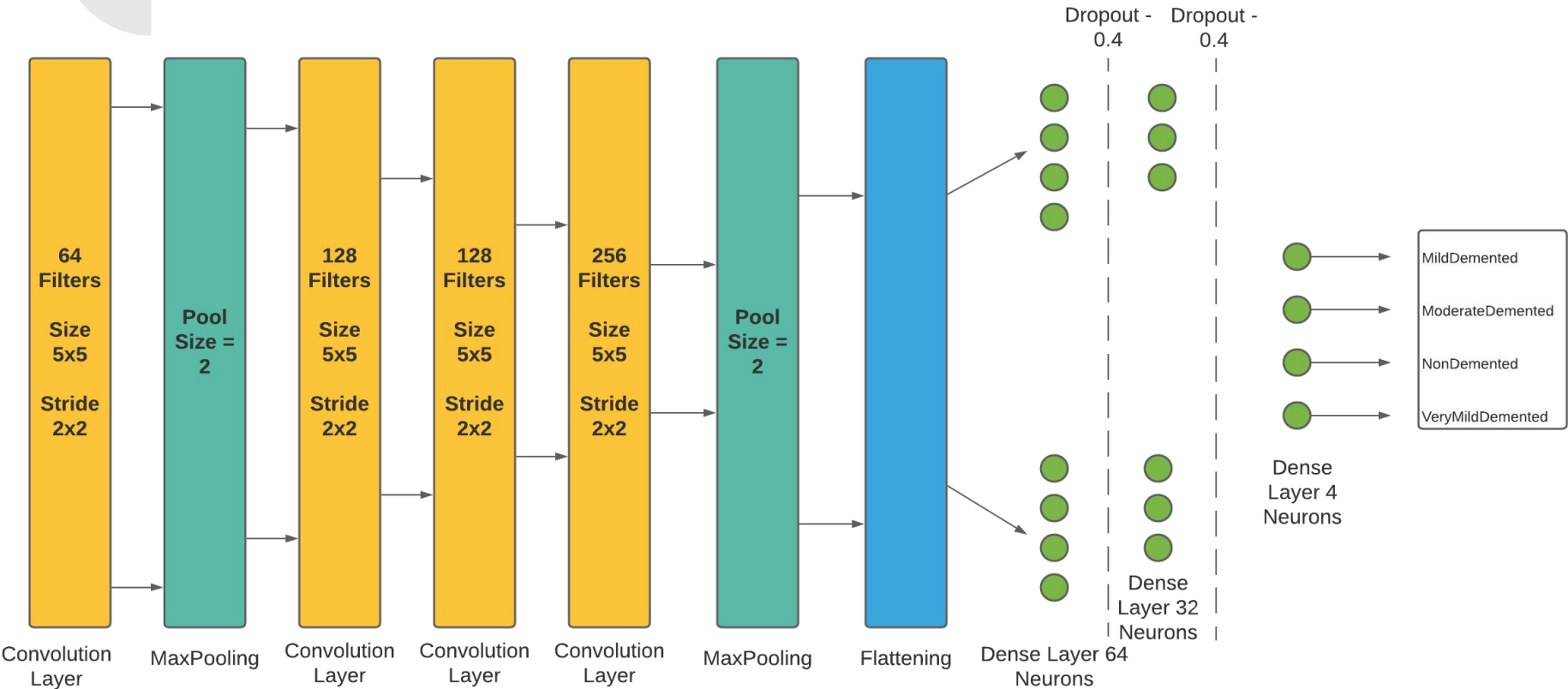
Data Pre-processing

Libraries Used:

Numpy
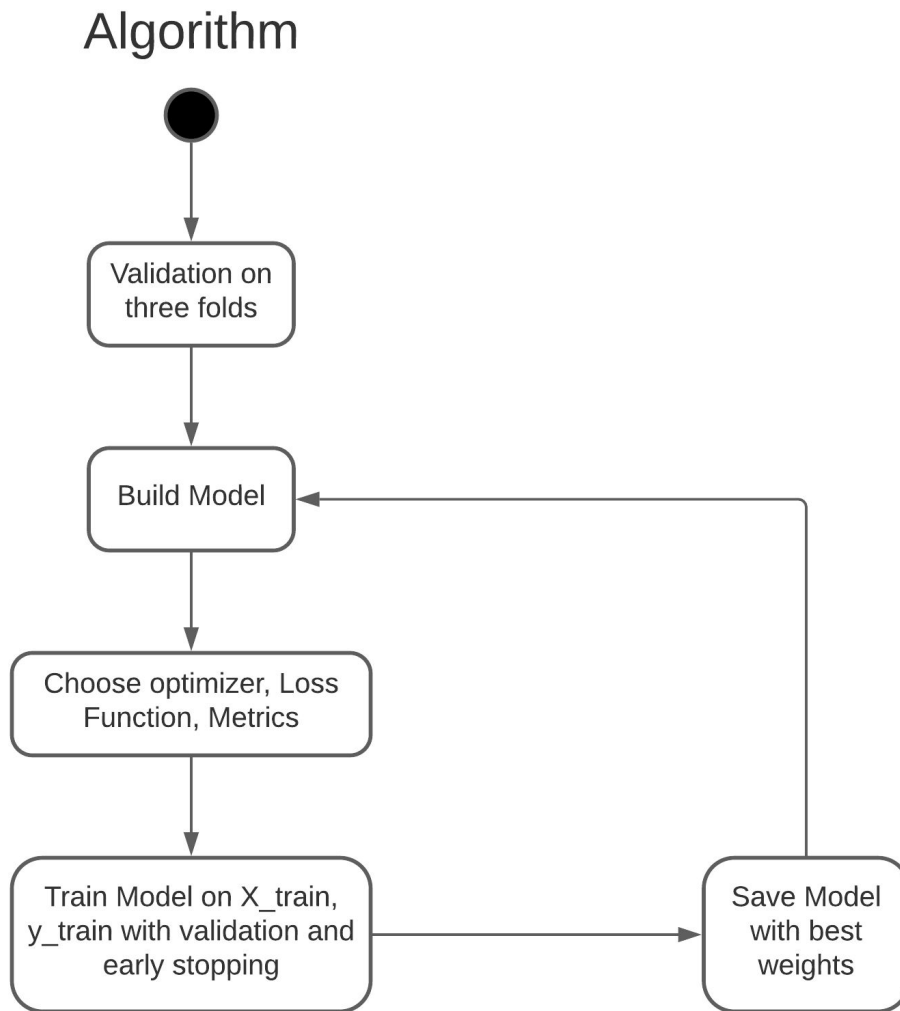Pandas
Scikit Learn
Glob
MinMaxScalar
LabelBinarizer

Load image from each folder of train data and test data

Load Glob Module

Open image inside "img" object

Covert image to numpy array

We get 8 objects(numpy arrays) for train and test data

| train_ND | train_VMD | train_MID | train_MOD |

Convert numpy array to dataframe having "image" and "label"

Tranform images to have value between 0 & 1 (MinMaxScalar) Transform Labels using onehot encoding (LabelBinarizer)

Split data into train and test (0.2) and maintain prportion

Reshape data and the implicit conversion to tensor

Fed to trainig Model

CNN-Architecture

CNN Architecture

Algorithm

```
Validation on
three folds
        │
        ▼
   Build Model  ◄──────────────────┐
        │                          │
        ▼                          │
Choose optimizer, Loss             │
  Function, Metrics                │
        │                          │
        ▼                          │
Train Model on X_train,    Save Model
y_train with validation ──▶ with best
  and early stopping          weights
```

Some Other Characteristics

Dataset is unbalanced:

```
Counter({'MID': 896, 'MOD': 64, 'ND': 3200, 'VMD': 2240})
```

To balance dataset we compute class weights:

```
{0: 1.7852161785216178, 1: 25.098039215686274, 2: 0.5, 3: 0.7142857142857143}
```
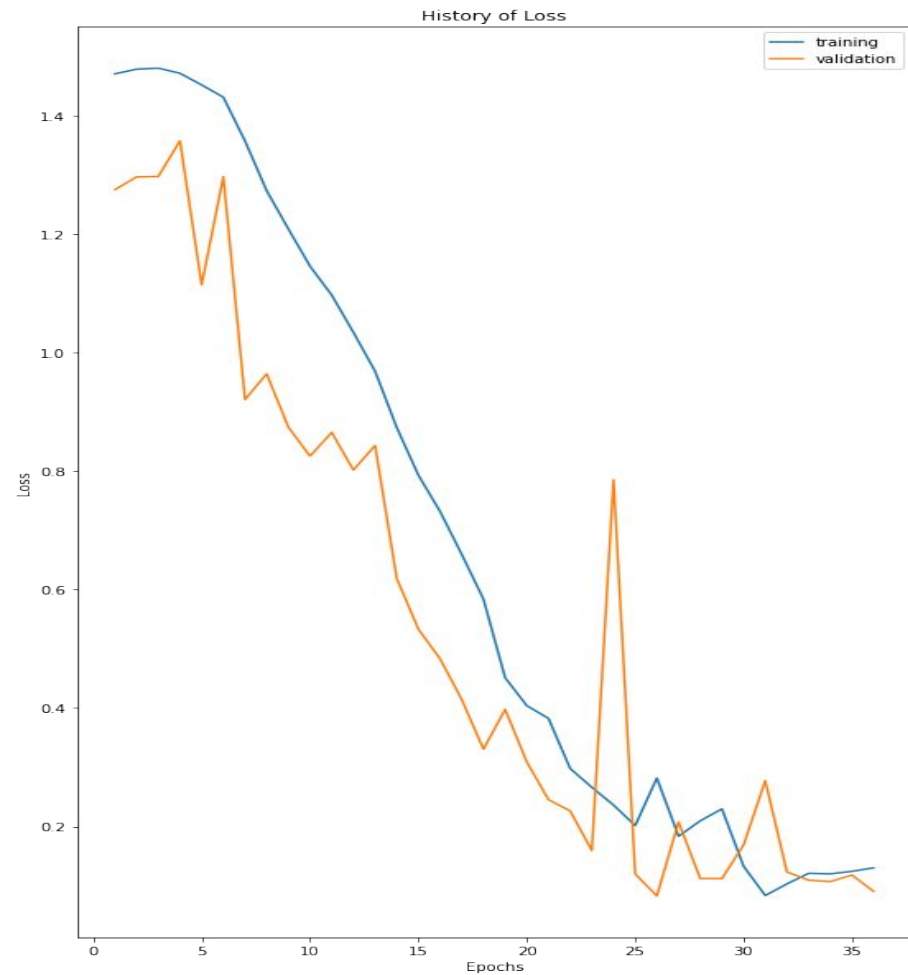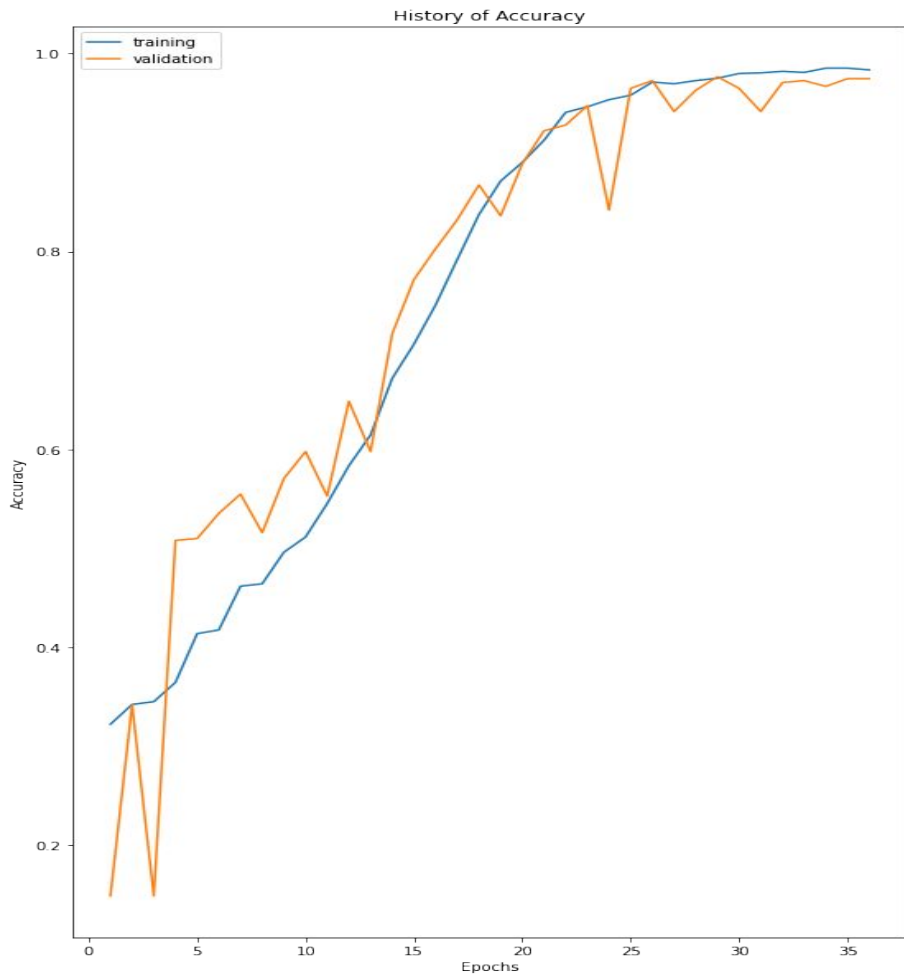
Optimizer: RMSprop

Loss Function: categorical_crossentropy

Metrics: Accuracy & Area under ROC Curve(ROC Curve → Graph between true positive and false positive rate)

EarlyStopping: When model starts to overfit (val_loss starts to increase)

ValidationSet = 0.1, Epochs = 100, BatchSize = 10

Performance

Confusion Matrix