AI/ML Project

Covid-19/Breast-Cancer/Alzheimer prediction

Team - 2

TEAM MEMBERS

Aakriti Singh
20190802006

20190802140
Ashwin Jawalikar

Krutika Deolapure
20190802054

20190802063
Maitri Surve

Priyansh Bakhtani
20190802079

Introduction

1. Coronavirus disease is an infectious disease caused by the sars-cov-2 virus.

2. Breast cancer is a disease in which cells in the breast grow out of control. A tumour can be benign or malignant. Benign are not cancerous tumors and Malignant are cancerous tumors.

3. Alzheimer’s disease is a neurodegenerative disorder that affects the brain. The symptoms are mild at first and become more severe over time.

PROJECT OVERVIEW

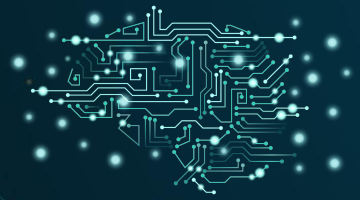
Disease Prediction System (DPS) is based on the Data Science And Aiml techniques which makes use of for predicting the risk level of diseases i.e. Breast Cancer, Covid-19 and Alzheimer Based on their symptoms and some reports.

The Covid-19 prediction uses 15 responses including some symptoms and several questions to predict the chances of the disease.

The breast cancer prediction is based on the measurements of the tumor.

The Alzheimer’s disease prediction uses MRI scans of a patient's brain to determine the severity of the patient’s Dementia.

MACHINE LEARNING ALGORITHMS



Supervised

COVID-19 (Decision Tree Classifier), Breast Cancer (Random Forest Classifier)

Unsupervised

Alzheimer's Disease (Convolutional Neural Network)

ARCHITECTURE FOR COVID-19 & BREAST CANCER PREDICTION

1.

Exploratory Data analysis

2.

Split the dataset to training data and testing data.

3.

Using Decision tree(Covid) /Random forest (Breast Cancer) to train the model with training data.

4.

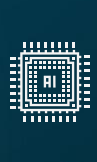
Test and validate the model with the test data.

5.

Compare the accuracy of the models.

Tested Algorithms

Decision Tree



K Nearest Neighbors

SVM

Logistic Regression

Random Forest

GaussianNB

Covid-19 EDA

- Finding Unique Values
- Checking for missing values
- Label Encoding
- Feature importance

Decision Tree Classifier

- uses a set of rules to make decisions
- perform both classification and regression tasks
- splits the dataset based on the value of a column, and creates new nodes
- visualize the model

Breast Cancer EDA

- Feature encoding
- Check Correlation
- Dropping dependent variable
- Feature Scaling

Random Forest Classifier

- Can be used for both Classification and Regression problems
- Combines multiple classifiers to solve a complex problem
- Enhances the accuracy of the model

Alzheimer Disease - Convolutional Neural Network

Input preprocessed images

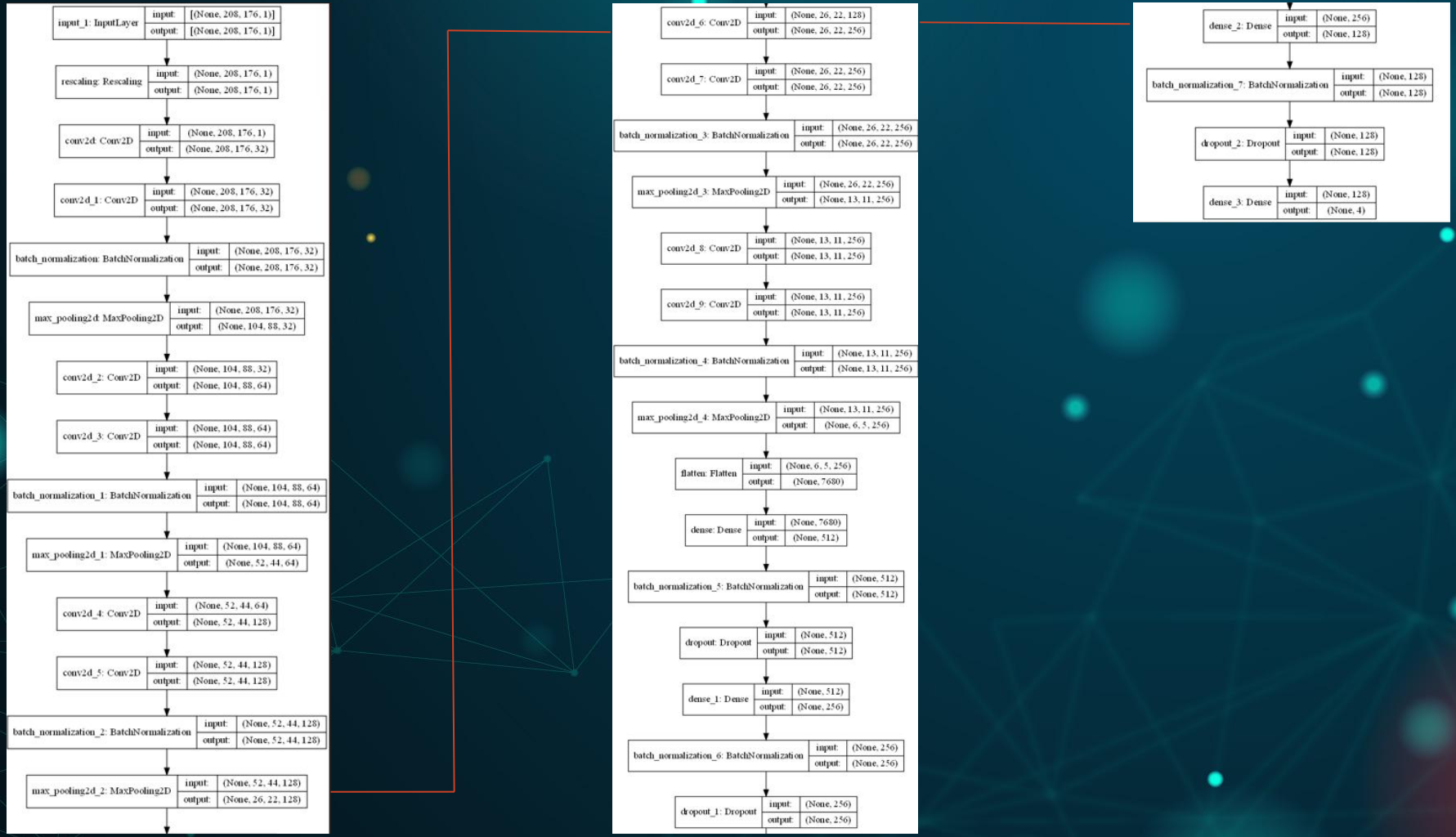
Scaling and image preprocessing

Model extracts features to find

Convolution and Pooling layers extract features and create smaller images

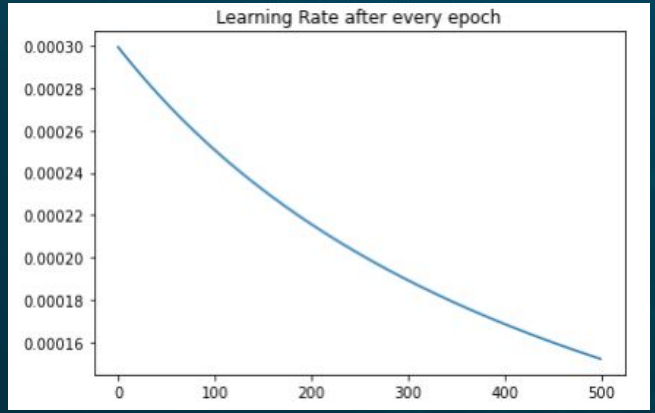
Finds aforementioned features in test images

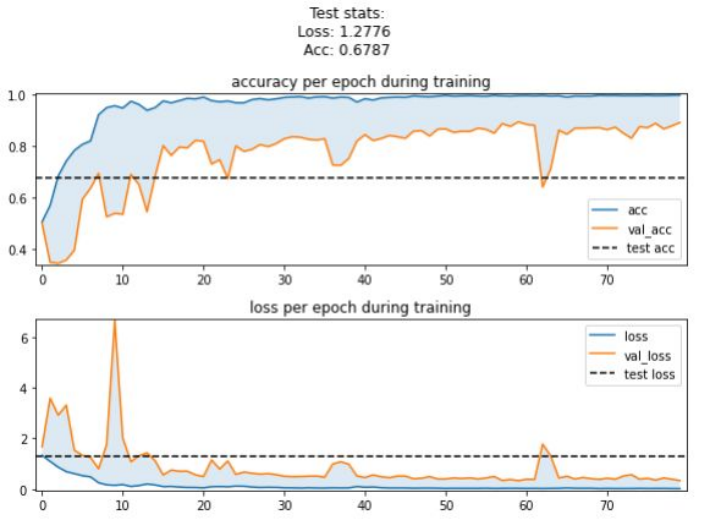
Identifies said features in test images after going through model


input_1: InputLayer | input: [(None, 208, 176, 1)] | output: [(None, 208, 176, 1)]
rescaling: Rescaling | input: (None, 208, 176, 1) | output: (None, 208, 176, 1)
conv2d: Conv2D | input: (None, 208, 176, 1) | output: (None, 208, 176, 32)
conv2d_1: Conv2D | input: (None, 208, 176, 32) | output: (None, 208, 176, 32)
batch_normalization: BatchNormalization | input: (None, 208, 176, 32) | output: (None, 208, 176, 32)
max_pooling2d: MaxPooling2D | input: (None, 208, 176, 32) | output: (None, 104, 88, 32)
conv2d_2: Conv2D | input: (None, 104, 88, 32) | output: (None, 104, 88, 64)
conv2d_3: Conv2D | input: (None, 104, 88, 64) | output: (None, 104, 88, 64)
batch_normalization_1: BatchNormalization | input: (None, 104, 88, 64) | output: (None, 104, 88, 64)
max_pooling2d_1: MaxPooling2D | input: (None, 104, 88, 64) | output: (None, 52, 44, 64)
conv2d_4: Conv2D | input: (None, 52, 44, 64) | output: (None, 52, 44, 128)
conv2d_5: Conv2D | input: (None, 52, 44, 128) | output: (None, 52, 44, 128)
batch_normalization_2: BatchNormalization | input: (None, 52, 44, 128) | output: (None, 52, 44, 128)
max_pooling2d_2: MaxPooling2D | input: (None, 52, 44, 128) | output: (None, 26, 22, 128)
conv2d_6: Conv2D | input: (None, 26, 22, 128) | output: (None, 26, 22, 256)
conv2d_7: Conv2D | input: (None, 26, 22, 256) | output: (None, 26, 22, 256)
batch_normalization_3: BatchNormalization | input: (None, 26, 22, 256) | output: (None, 26, 22, 256)
max_pooling2d_3: MaxPooling2D | input: (None, 26, 22, 256) | output: (None, 13, 11, 256)
conv2d_8: Conv2D | input: (None, 13, 11, 256) | output: (None, 13, 11, 256)
conv2d_9: Conv2D | input: (None, 13, 11, 256) | output: (None, 13, 11, 256)
batch_normalization_4: BatchNormalization | input: (None, 13, 11, 256) | output: (None, 13, 11, 256)
max_pooling2d_4: MaxPooling2D | input: (None, 13, 11, 256) | output: (None, 6, 5, 256)
flatten: Flatten | input: (None, 6, 5, 256) | output: (None, 7680)
dense: Dense | input: (None, 7680) | output: (None, 512)
batch_normalization_5: BatchNormalization | input: (None, 512) | output: (None, 512)
dropout: Dropout | input: (None, 512) | output: (None, 512)
dense_1: Dense | input: (None, 512) | output: (None, 256)
batch_normalization_6: BatchNormalization | input: (None, 256) | output: (None, 256)
dropout_1: Dropout | input: (None, 256) | output: (None, 256)
dense_2: Dense | input: (None, 256) | output: (None, 128)
batch_normalization_7: BatchNormalization | input: (None, 128) | output: (None, 128)
dropout_2: Dropout | input: (None, 128) | output: (None, 128)
dense_3: Dense | input: (None, 128) | output: (None, 4)

Training and testing performance

- Used variable learning rate that decreases by 5% after every epoch





- Training performance every epoch and final testing performance

THANKS!