

Project report

CS419-Introduction to Machine Learning

Covid19 Detection Using CNN

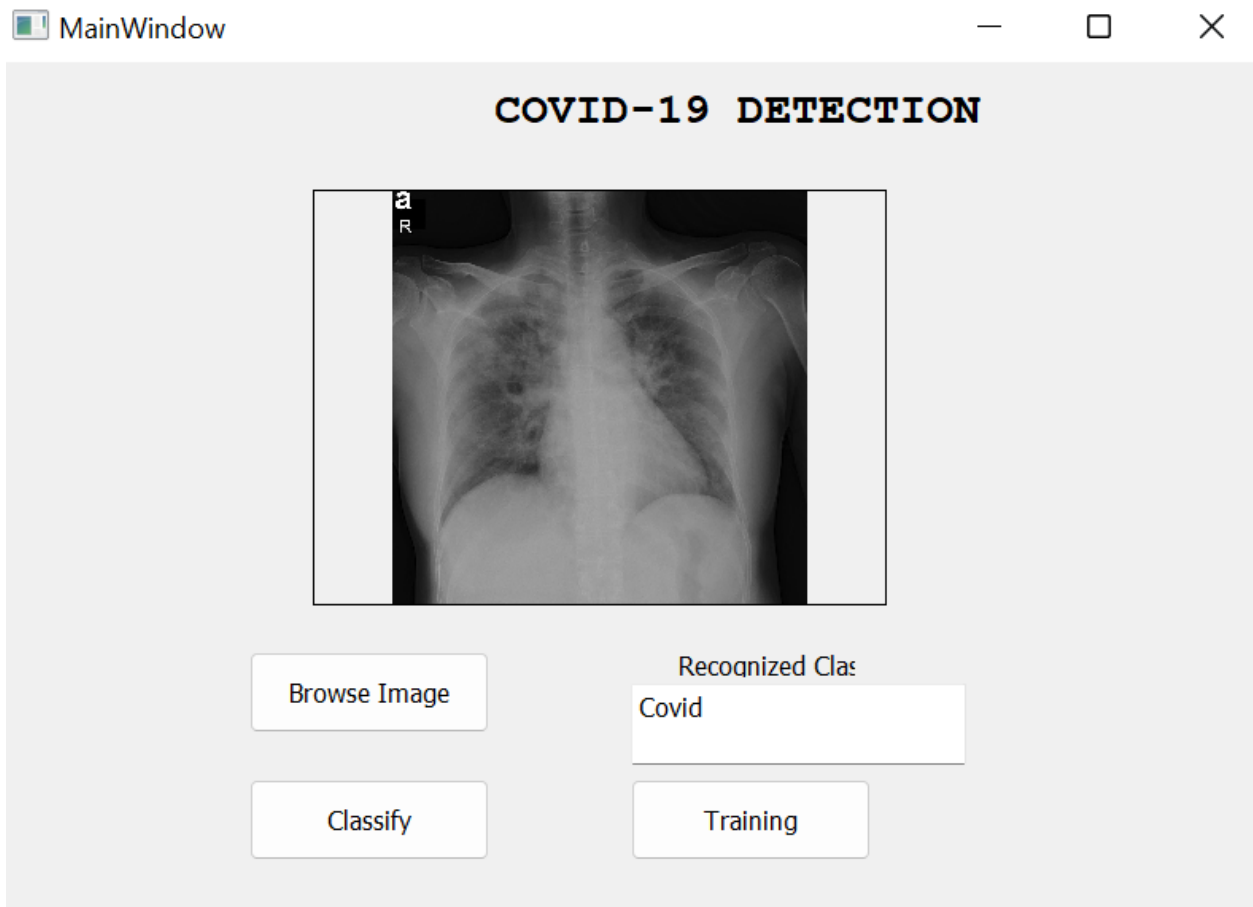
Group - 26
Utkarsh Gupta - 20040158

Introduction.....	2
Dataset details.....	3
Approach.....	3
1. Training Function:.....	3
2. Classify Function:.....	4
3. Load Image:.....	4
4. Retranslate:.....	4
5. Main setup UI:.....	4
Experiments and Results.....	4
References.....	5

Introduction

I have prepared a deep learning model which uses the dataset containing the chest's X-rays and classifies it as covid or normal. I have also coded a GUI that allows the user to browse through the images, select a particular image and then the model classifies it. The output we get is the probabilities of the image being covid or not in an array and the max of this array classifies our image.

I used Keras and the numpy library to make the CNN model. The GUI was made using the PyQt5 library.



Dataset details

I used the dataset which consists of jpeg images of chest x-rays available online. The testing dataset has 50 images, 25 of each kind.

Approach

The code has been divided into 5 functions:

1. Training Function:

This function is responsible for the training of the model. A basic CNN is formed using layers along with other functions like batch normalization, drop out, flatten and dense. The model is saved as a json file. The weights and model are then used by the classify function for classification.

2. Classify Function:

This takes the input from the user and then predicts the class using the model generated by the training function.

3. Load Image:

This function accepts the image file to be classified from the user. It saves the path and the image file and sends it to the classify function.

4. Retranslate:

This function is used to rename the buttons from their global definition. Each button has its own use.

5. Main setup UI:

This is used to define the GUI and all the buttons. It forms a new window anytime the code runs. The 3 buttons each have a specific role:

1. Browse: select the image from the folder to be classified (Load Image)
2. Classify: Gives us the final output (Classify function)
3. Training: If there are new images in the training dataset then using this we can re-train our model and see the results in the terminal.

```
def setupUi(self, MainWindow): ...  
  
def retranslateUi(self, MainWindow): ...  
  
def loadImage(self): ...  
  
def classifyFunction(self): ...  
  
def trainingFunction(self): ...
```

Experiments and Results

1. Step wise increased the number of layers in the model. Then tested the model on the dataset after adding the fifth layer, most of the images were classified correctly. Hence avoided the 6th layer to avoid any overfitting.
2. Changing the learning rate had very little effect.
3. Tried 2 optimizers sgd and adam. Adam gave better results.

Most (**23/25**) of the images of the testing dataset were classified as covid. On average the correct probability of the image being correctly identified was around **75%**(A probability of more than **50%** was required for an image to be classified as either) which ensured correct classification.

References

1. <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>
2. <https://www.hindawi.com/journals/complexity/2021/6621607/>