main.py - E:\Internship\29_Covid19Detection_CNN\main.py (3.7.8)

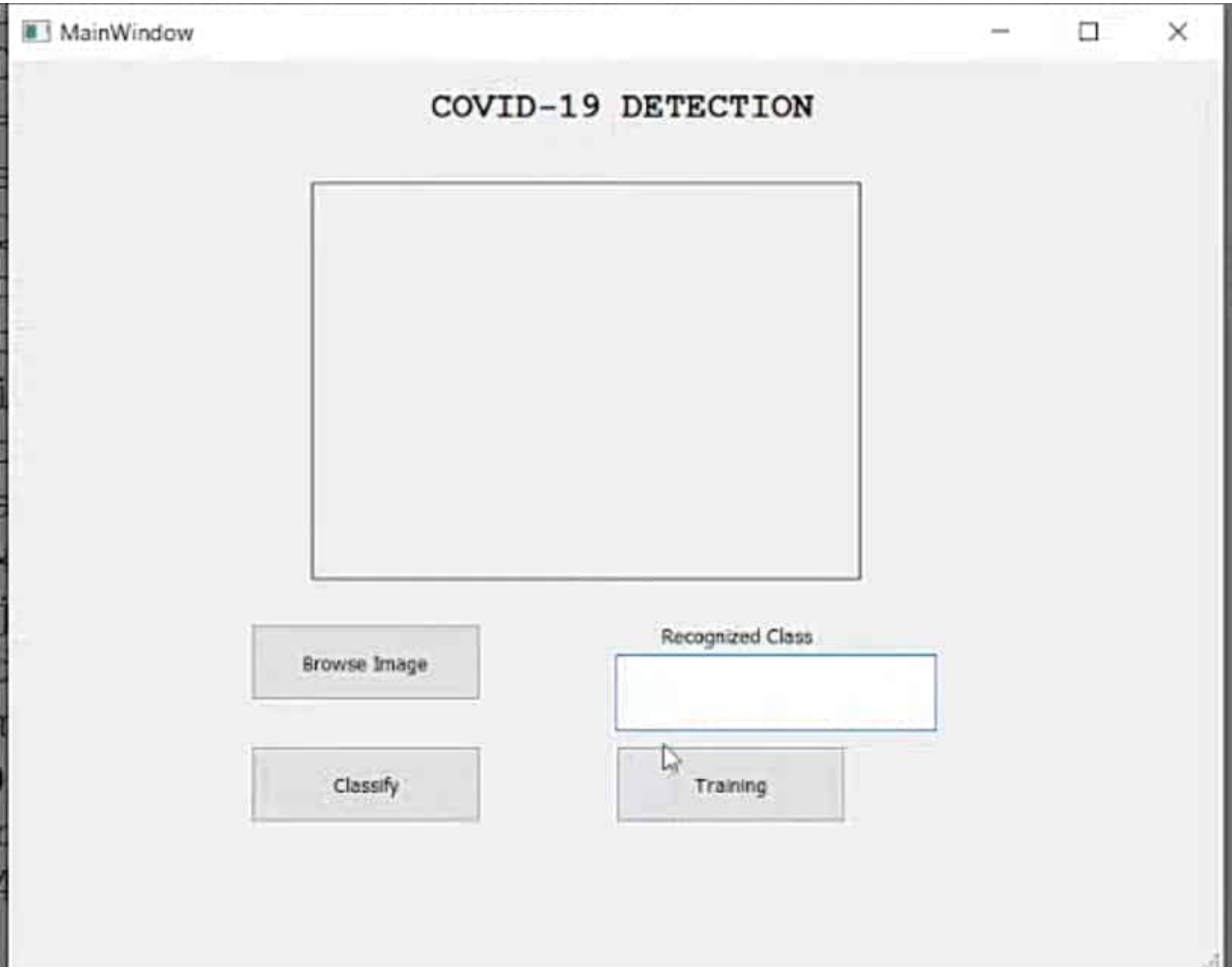
File Edit Format Run Options Window Help

```
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'D:\opencv\ui\Gujarathi_lang_recognition\
#
# Created by: PyQt5 UI code generator 5.11.3
#
# WARNING! All changes made in this file will be lost!

from PyQt5 import QtCore, QtGui, QtWidgets
import numpy as np
from keras.preprocessing import image
from keras.models import Sequential
from keras.layers import Dense
from keras.models import model_from_json
from keras.models import Sequential
#initialize nn
from keras.layers import Conv2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
#convert pooling features space to large feature vector for fully
#connected layer
from keras.preprocessing.image import ImageDataGenerator
from keras.layers import BatchNormalization
from keras.layers import Dropout

import os
```

main.py - E:\Internship\29_Covid19Detection_CNN\main.py (3.7.8)

File Edit Format Run Options Window Help

```
import os
import cv2
####################
from imutils import contours
from sklearn.cluster import KMeans
from sklearn.cluster import spectral_clustering
from sklearn.neural_network import MLPClassifier
import csv
import scipy
import scipy.io as sio
import imutils
import os
import mahotas as mt
p=1;
#########################

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(800, 600)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.BrowseImage = QtWidgets.QPushButton(self.centralwidget)
        self.BrowseImage.setGeometry(QtCore.QRect(160, 370, 151, 51))
        self.BrowseImage.setObjectName("BrowseImage")
        self.imageLbl = QtWidgets.QLabel(self.centralwidget)
```


MainWindow
COVID-19 DETECTION
Browse Image
Recognized Class
Classify
Training

```
        self.imageLbl = QtWidgets.QLabel(self.centralwidget)
        self.imageLbl.setGeometry(QtCore.QRect(200, 80, 361, 261))
        self.imageLbl.setFrameShape(QtWidgets.QFrame.Box)
        self.imageLbl.setText("")
        self.imageLbl.setObjectName("imageLbl")
        self.label_2 = QtWidgets.QLabel(self.centralwidget)
        self.label_2.setGeometry(QtCore.QRect(110, 20, 621, 20))
        font = QtGui.QFont()
        font.setFamily("Courier New")
        font.setPointSize(14)
        font.setBold(True)
        font.setWeight(75)
        self.label_2.setFont(font)
        self.label_2.setObjectName("label_2")
        self.Classify = QtWidgets.QPushButton(self.centralwidget)
        self.Classify.setGeometry(QtCore.QRect(160, 450, 151, 51))
        self.Classify.setObjectName("Classify")
        self.label = QtWidgets.QLabel(self.centralwidget)
        self.label.setGeometry(QtCore.QRect(430, 370, 111, 16))
        self.label.setObjectName("label")
        self.Training = QtWidgets.QPushButton(self.centralwidget)
        self.Training.setGeometry(QtCore.QRect(400, 450, 151, 51))
        self.Training.setObjectName("Training")
        self.textEdit = QtWidgets.QTextEdit(self.centralwidget)
        self.textEdit.setGeometry(QtCore.QRect(400, 390, 211, 51))
        self.textEdit.setObjectName("textEdit")
```

main.py - E:\Internship\29_Covid19Detection_CNN\main.py (3.7.8)

File Edit Format Run Options Window Help

```
        self.textEdit.setObjectName("textEdit")
        MainWindow.setCentralWidget(self.centralwidget)
        self.menubar = QtWidgets.QMenuBar(MainWindow)
        self.menubar.setGeometry(QtCore.QRect(0, 0, 800, 26))
        self.menubar.setObjectName("menubar")
        MainWindow.setMenuBar(self.menubar)
        self.statusbar = QtWidgets.QStatusBar(MainWindow)
        self.statusbar.setObjectName("statusbar")
        MainWindow.setStatusBar(self.statusbar)

        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

        self.BrowseImage.clicked.connect(self.loadImage)

        self.Classify.clicked.connect(self.classifyFunction)

        self.Training.clicked.connect(self.trainingFunction)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
        self.BrowseImage.setText(_translate("MainWindow", "Browse Image"))
        self.label_2.setText(_translate("MainWindow", "            COVID-19 DET
        self.Classify.setText(_translate("MainWindow", "Classify"))
        self.label.setText(_translate("MainWindow", "Recognized Class"))
```

Type here to search

27°C Partly cloudy

main.py - E:\Internship\29_Covid19Detection_CNN\main.py (3.7.8)

File Edit Format Run Options Window Help

```
        self.label.setText(_translate("MainWindow", "Recognized Class"))
        self.Training.setText(_translate("MainWindow", "Training"))

    def loadImage(self):
        fileName, _ = QtWidgets.QFileDialog.getOpenFileName(None, "Select Image", "", "Image F
        if fileName: # If the user gives a file
            print(fileName)
            self.file=fileName
            pixmap = QtGui.QPixmap(fileName) # Setup pixmap with the provided image
            pixmap = pixmap.scaled(self.imageLbl.width(), self.imageLbl.height(), QtCore.Qt.Ke
            self.imageLbl.setPixmap(pixmap) # Set the pixmap onto the label
            self.imageLbl.setAlignment(QtCore.Qt.AlignCenter) # Align the label to center

    def classifyFunction(self):
        json_file = open('model.json', 'r')
        loaded_model_json = json_file.read()
        json_file.close()
        loaded_model = model_from_json(loaded_model_json)
        # load weights into new model
        loaded_model.load_weights("model.h5")
        #loaded_model.load_weights("ResNet50-ft-10.model")
        print("Loaded model from disk");
        label=["Covid","Normal"]
        path2=self.file
        print(path2)
        ########################
```

```
    ########################
    test_image = image.load_img(path2, target_size = (128, 128))
    test_image = image.img_to_array(test_image)
    test_image = np.expand_dims(test_image, axis = 0)
    result = loaded_model.predict(test_image)
    print(result)
    fresult=np.max(result)
    label2=label[result.argmax()]
    print(label2)
    self.textEdit.setText(label2)

def trainingFunction(self):
    self.textEdit.setText("Training under process...")
    #basic cnn
    model = Sequential()
    model.add(Conv2D(32, kernel_size = (3, 3), activation='relu', input_shape=(128,128, 3)))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(BatchNormalization())
    model.add(Conv2D(64, kernel_size=(3,3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(BatchNormalization())
    model.add(Conv2D(64, kernel_size=(3,3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
    model.add(BatchNormalization())
    model.add(Conv2D(96, kernel_size=(3,3), activation='relu'))
    model.add(MaxPooling2D(pool_size=(2,2)))
```

```
model.add(BatchNormalization())
model.add(Conv2D(64, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(96, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Conv2D(32, kernel_size=(3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(BatchNormalization())
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(2, activation = 'softmax'))

model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accur

train_datagen = ImageDataGenerator(rescale = None,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)
```

main.py - E:\Internship\29_Covid19Detection_CNN\main.py (3.7.8)

File Edit Format Run Options Window Help

```
test_datagen = ImageDataGenerator(rescale = 1./255)

training_set = train_datagen.flow_from_directory('G:\__Technology___Beyond_Dreams\Deep_
                                                 target_size = (128, 128),
                                                 batch_size = 8,
                                                 class_mode = 'categorical')
#print(test_datagen);
labels = (training_set.class_indices)
print(labels)


test_set = test_datagen.flow_from_directory('G:\__Technology___Beyond_Dreams\Deep_Learn
                                            target_size = (128, 128),
                                            batch_size = 8,
                                            class_mode = 'categorical')

labels2 = (test_set.class_indices)
print(labels2)
#self.textEdit.setText(labels2)

model.fit_generator(training_set,
                    steps_per_epoch = 100,
                    epochs = 10,
                    validation_data = test_set,
                    validation_steps = 125)
```

main.py - E:\Internship\29_Covid19Detection_CNN\main.py (3.7.8)

File Edit Format Run Options Window Help

```
                                    validation_data = test_set,
                                    validation_steps = 125)


        # Part 3 - Making new predictions

        model_json=model.to_json()
        with open("model.json", "w") as json_file:
            json_file.write(model_json)
        # serialize weights to HDF5
            model.save_weights("model.h5")
            print("Saved model to disk")
            self.textEdit.setText("Saved model to disk")


if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())
```

Type here to search

27°C Partly cloudy