Modelt.py - Notepad

File Edit Format View Help

```
import warnings
warnings.filterwarnings('ignore') # suppress import warnings

import os
import cv2
import tflearn
import numpy as np
import tensorflow as tf
from random import shuffle
from tqdm import tqdm
from tflearn.layers.conv import conv_2d, max_pool_2d
from tflearn.layers.core import input_data, dropout, fully_connected
from tflearn.layers.estimator import regression

''' <global actions> '''

TRAIN_DIR = r'F:\Development project\100% Module ready\100% pothholes ready\Dataset\train'
TEST_DIR = r'F:\Development project\100% Module ready\100% pothholes ready\Dataset\test'
IMG_SIZE = 50
LR = 1e-3
MODEL_NAME = 'Potholesdetection-{}-{}.model'.format(LR, '2conv-basic')
tf.logging.set_verbosity(tf.logging.ERROR) # suppress keep_dims warnings
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3' # suppress tensorflow gpu logs
tf.reset_default_graph()

''' </global actions> '''

def label_leaves(brain):
    braintype = brain[0]
    ans = [0,0]

    if braintype =="Y": ans = [1,0]
    elif braintype =="N": ans = [0,1]


    return ans

def create_training_data():
```

Unix (LF) Ln 1, Col 1 100%

Model Training-1

Modelt.py - Notepad

File Edit Format View Help

```
def create_training_data():

    training_data = []

    for img in tqdm(os.listdir(TRAIN_DIR)):
        label = label_leaves(img)
        path = os.path.join(TRAIN_DIR,img)
        img = cv2.imread(path,cv2.IMREAD_COLOR)
        img = cv2.resize(img, (IMG_SIZE,IMG_SIZE))
        training_data.append([np.array(img),np.array(label)])

    shuffle(training_data)
    np.save('Brain.npy', training_data)

    return training_data

def main():

    train_data = create_training_data()

    convnet = input_data(shape=[None, IMG_SIZE, IMG_SIZE, 3], name='input')

    convnet = conv_2d(convnet, 32, 3, activation='relu')
    convnet = max_pool_2d(convnet, 3)

    convnet = conv_2d(convnet, 64, 3, activation='relu')
    convnet = max_pool_2d(convnet, 3)

    convnet = conv_2d(convnet, 128, 3, activation='relu')
    convnet = max_pool_2d(convnet, 3)

    convnet = conv_2d(convnet, 32, 3, activation='relu')
    convnet = max_pool_2d(convnet, 3)

    convnet = conv_2d(convnet, 64, 3, activation='relu')
    convnet = max_pool_2d(convnet, 3)

    convnet = fully_connected(convnet, 1024, activation='relu')
```

Unix (LF) Ln 38, Col 1 100%

Model Training-2

Modelt.py - Notepad

File Edit Format View Help

```
    model = tflearn.DNN(convnet, tensorboard_dir='log')


    if os.path.exists('{}.meta'.format(MODEL_NAME)):
        model.load(MODEL_NAME)
        print('Model Loaded')

    train = train_data[:-500]
    test = train_data[-20:]

    X = np.array([i[0] for i in train]).reshape(-1,IMG_SIZE,IMG_SIZE,3)
    Y = [i[1] for i in train]

    test_x = np.array([i[0] for i in test]).reshape(-1,IMG_SIZE,IMG_SIZE,3)
    test_y = [i[1] for i in test]

    model_info = model.fit({'input': X}, {'targets': Y}, n_epoch=100, validation_set=({'input': test_x}, {'targets': test_y}), snapshot_step=40, show_metric=True, run_i

    print(model_info)
    model.save(MODEL_NAME)

    from sklearn.metrics import classification_report

    pred = model.predict(test_x)

    predicted = np.argmax(pred, axis=1)
    report = classification_report(np.argmax(test_y, axis=1), predicted)
    print(report)




    # # plot model history after each epoch
    # from plt_graph import plot_model_history
    # plot_model_history(model_info)
    #
    # return Str
if __name__ == '__main__':
```

Unix (LF) Ln 81, Col 27 100%

Model Training-3

GUI_Master.py - Notepad

File Edit Format View Help

```
import tkinter as tk
from tkinter import ttk, LEFT, END
from PIL import Image , ImageTk
from tkinter.filedialog import askopenfilename
import cv2
import numpy as np
import time
#import tfModel_CPU
import Modelt as MT
import tfModel_test as tf_test
global fn
fn=""
###############################################+=============================================================
root = tk.Tk()
root.configure(background="seashell2")
#root.geometry("1300x700")


w, h = root.winfo_screenwidth(), root.winfo_screenheight()
root.geometry("%dx%d+0+0" % (w, h))
root.title("Potholes  Detection")


#430
#+++++++++++++++++++++++++++++++++++++++++++++
#####For background Image
image2 =Image.open('a2.jpg')
image2 =image2.resize((w,h), Image.ANTIALIAS)

background_image=ImageTk.PhotoImage(image2)

background_label = tk.Label(root, image=background_image)

background_label.image = background_image

background_label.place(x=0, y=0) #, relwidth=1, relheight=1)
#
lbl = tk.Label(root, text="Road Condition Detection ", font=('times', 30,' bold '), height=1, width=35,bg="seashell2",fg="indian red")
```

Windows (CRLF) Ln 1, Col 1 100%

GUI Py – 1

GUI_Master.py - Notepad

File Edit Format View Help

```
#
lbl = tk.Label(root, text="Road Condition Detection ", font=('times', 30,' bold '), height=1, width=35,bg="seashell2",fg="indian red")
lbl.place(x=250, y=0)


frame_display = tk.LabelFrame(root, text=" --Display-- ", width=950, height=300, bd=5, font=('times', 10, ' bold '),bg="lightblue4")
frame_display.grid(row=0, column=0, sticky='nw')
frame_display.place(x=200, y=60)



frame_alpr = tk.LabelFrame(root, text=" --Process-- ", width=200, height=650, bd=5, font=('times', 10, ' bold '),bg="lightblue4")
frame_alpr.grid(row=0, column=0, sticky='nw')
frame_alpr.place(x=5, y=0)


################################$%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
def clear_img():
    img11 = tk.Label(frame_display, background='DarkGoldenrod1', width=160, height=120)
    img11.place(x=0, y=0)

def update_label(str_T):
    clear_img()
    result_label = tk.Label(frame_display, text=str_T, width=50, font=("bold", 25), bg='DarkGoldenrod1', fg='black')
    result_label.place(x=0, y=0)

def train_model():
        update_label("Model Training Start...............")

        start = time.time()

        X = MT.main()

        end = time.time()

        ET = "Execution Time: {0:.4} seconds \n".format(end - start)

        msg = "Model Training Completed.."  # + '\n' + X + '\n' + ET
```

Windows (CRLF) Ln 1, Col 1 100%

GUI Py – 2

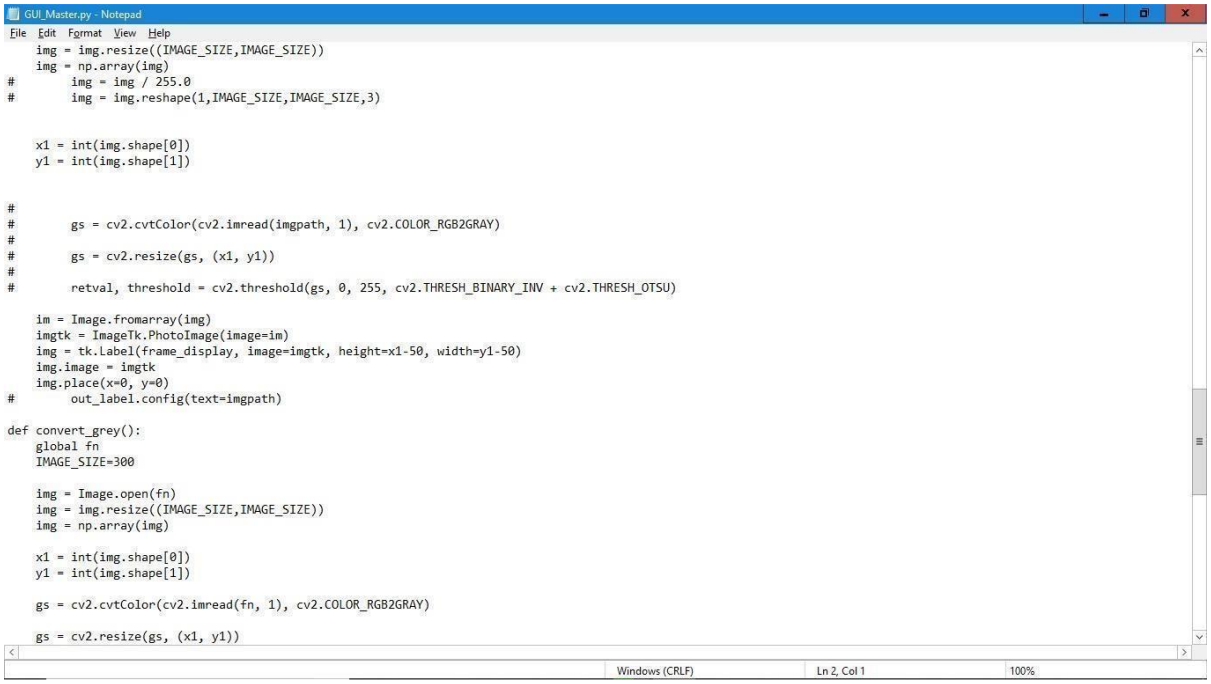
GUI_Master.py - Notepad

File Edit Format View Help

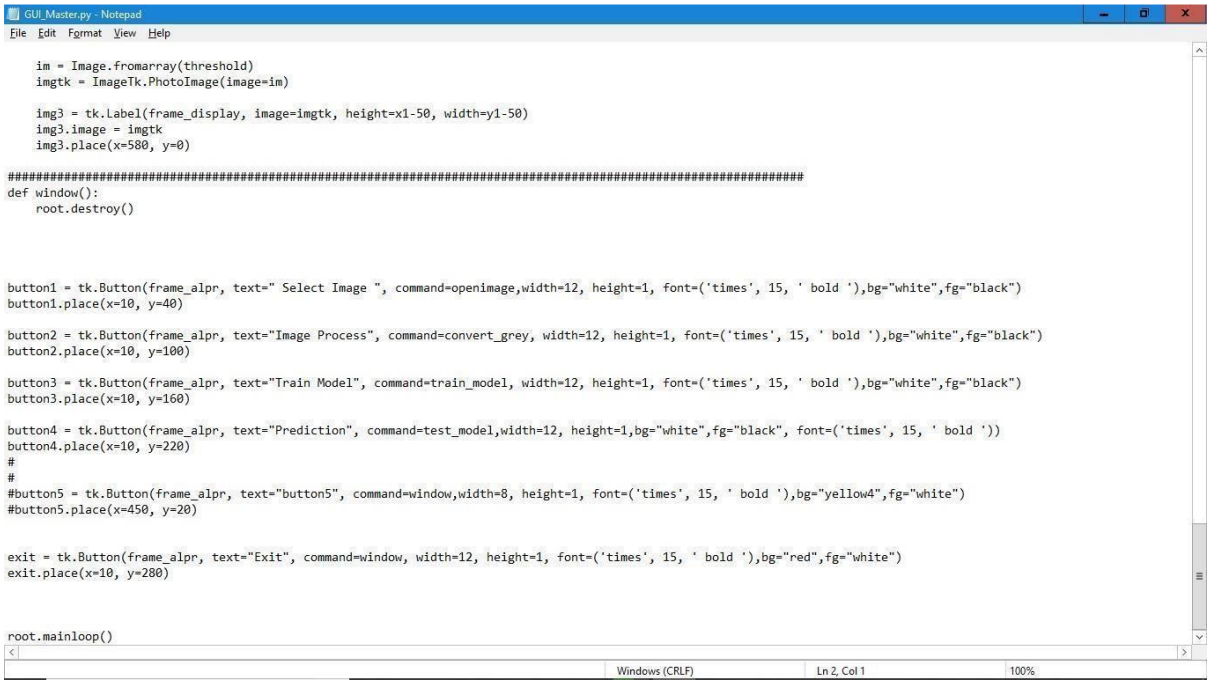
```
def test_model():
    global fn
    if fn != "":
        update_label("Model Testing Start...............")

        start = time.time()

        X = tf_test.analysis(fn)
        X1 = "Selected Image is {0}".format(X)
        end = time.time()

        ET = "Execution Time: {0:.4} seconds \n".format(end - start)

        msg = "Image Testing Completed.." + '\n' + X1 + '\n' + ET
        fn = ""
    else:
        msg = "Please Select Image For Prediction...."

    update_label(msg)





def openimage():

    global fn
    clear_img()
    fileName = askopenfilename(initialdir='/dataset', title='Select image for Aanalysis ',
                           filetypes=[("all files", "*.*")])
    IMAGE_SIZE=300
    imgpath = fileName
    fn = fileName


#        img = Image.open(imgpath).convert("L")
```

Windows (CRLF) Ln 1, Col 1 100%

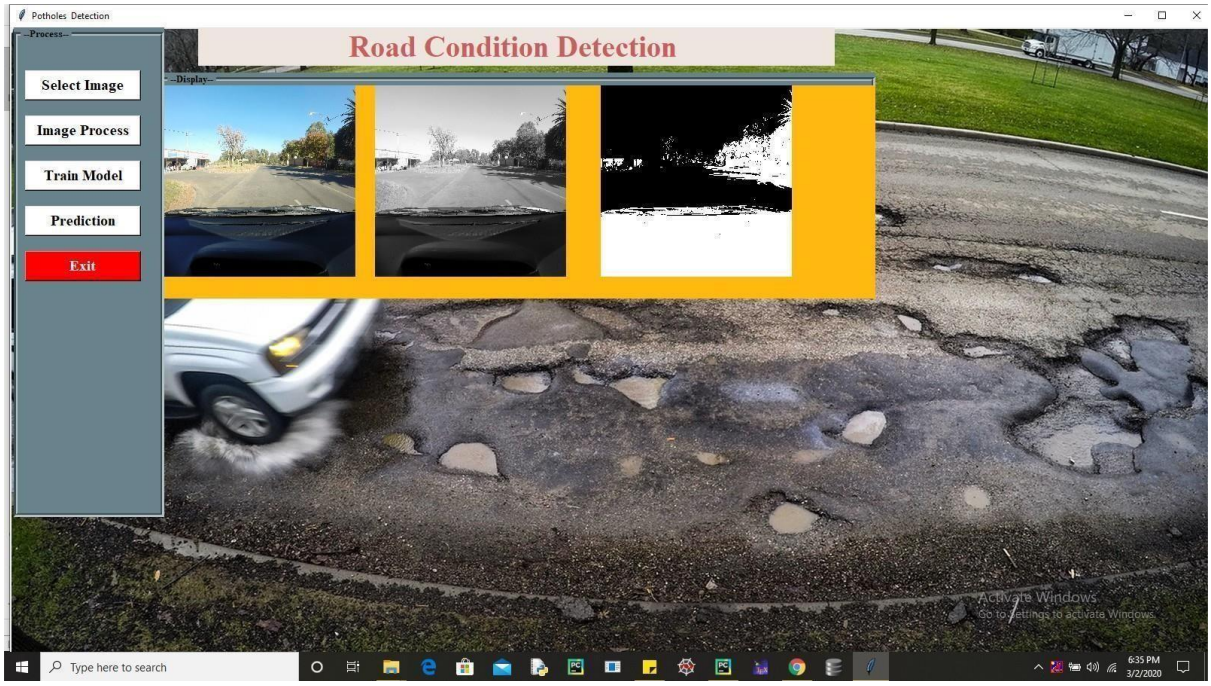
GUI Py – 3



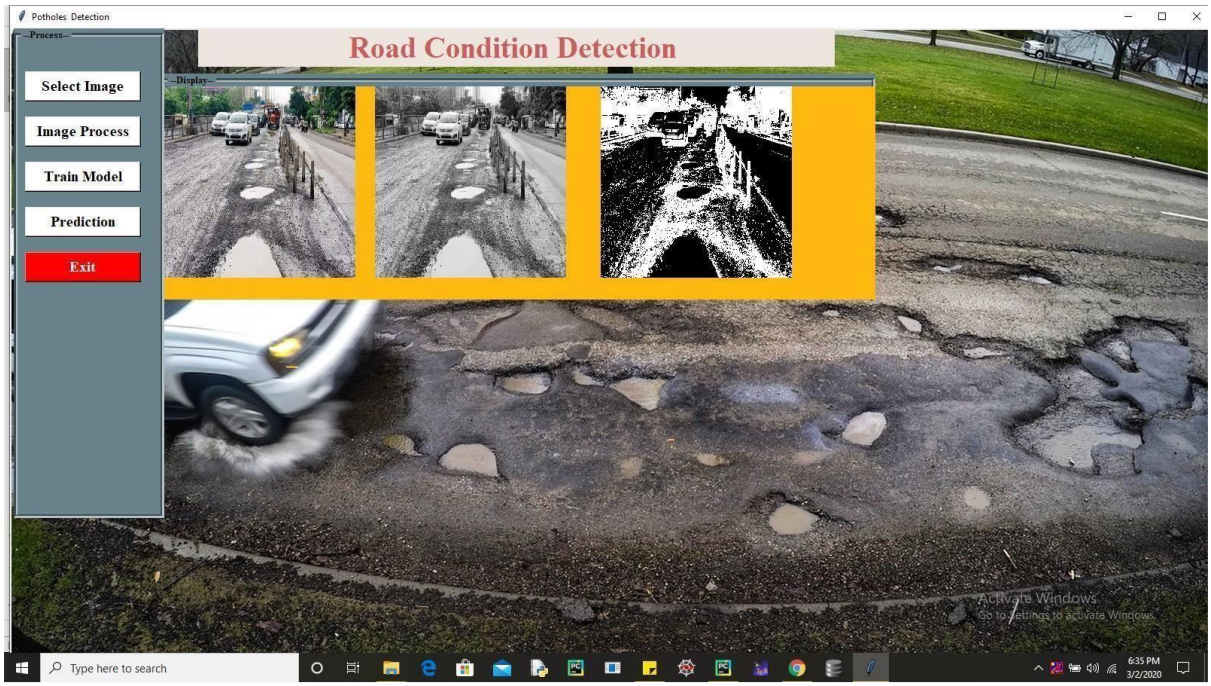
GUI Py – 4



GUI Py – 5


Potholes Detection
Process
Select Image
Image Process
Train Model
Prediction
Exit
Road Condition Detection
Display
Activate Windows
Go to Settings to activate Windows.
Type here to search
6:35 PM
3/2/2020

GUI – 1


Potholes Detection
Process
Select Image
Image Process
Train Model
Prediction
Exit
Road Condition Detection
Display
Activate Windows
Go to Settings to activate Windows.
Type here to search
6:35 PM
3/2/2020

GUI – 2