In [481]:
```
pip install opencv-python
```

```
Requirement already satisfied: opencv-python in c:\users\hp\anaconda3\lib\site-packages (4.6.0.66)Note: you may need to rest
art the kernel to use updated packages.

Requirement already satisfied: numpy>=1.17.3 in c:\users\hp\anaconda3\lib\site-packages (from opencv-python) (1.21.5)
```

In [482]:
```
import numpy as np
import pandas as pd
```

In [483]:
```
import os
for dir_name,_,file_names in os.walk('C:/Users/hp/Downloads/traffic_Data'):
    for filename in file_names:
        print(os.path.join(dir_name, filename))
```

```
C:/Users/hp/Downloads/traffic_Data\traffic_sign_labels.csv
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0001.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0002.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0003.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0004.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0005.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0006.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0007.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0008.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0008_j.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0009.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0010.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0011.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0012.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0013.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0014.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0015.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0016.png
C:/Users/hp/Downloads/traffic_Data\DATA\0\000_0017.png
```

In [499]:
```
from pathlib import Path
images = list(Path('C:/Users/hp/Downloads/traffic_Data/DATA/').glob(r'**/*.png'))
labels = list(map(lambda path: os.path.split(os.path.split(path)[0])[1], images))
classes=len(set(labels))
train_images = pd.Series(images).astype(str)
train_labels = pd.Series(labels).astype(str)
print(classes)
```

```
58
```

In [500]:
```
import cv2
size=128
train_data=[]

for elem in range(len(train_images)):
    image = cv2.imread(train_images[elem])
    image_resized = cv2.resize(image, (size, size), interpolation = cv2.INTER_AREA)
    train_data.append(np.array(image_resized))
```

In [502]:
```
data = np.array(train_data)
data = data.reshape((data.shape[0], 128*128*3))
data_scaled = data.astype(float)/255
labels = np.array(train_labels)
from sklearn.model_selection import train_test_split
X_train, X_val, y_train, y_val = train_test_split(data_scaled, labels, test_size=0.20, random_state=65)
```

In [503]:
```
classifier = RandomForestClassifier(min_samples_split=10,n_estimators=1400, max_depth=10,criterion = 'entropy', random_state
classifier.fit(X_train, y_train)
```

Out[503]:
```
RandomForestClassifier(criterion='entropy', max_depth=10, min_samples_split=10,
                       n_estimators=1400, random_state=65)
```

In [504]:
```
y_pred = classifier.predict(X_val)
```

In [505]:
```
from sklearn.metrics import classification_report
print(classification_report(y_val, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        30
           1       1.00      1.00      1.00        11
          10       1.00      1.00      1.00        13
          11       1.00      1.00      1.00        30
          12       1.00      1.00      1.00        24
          13       1.00      1.00      1.00         9
          14       1.00      1.00      1.00        25
          15       1.00      1.00      1.00         7
          16       1.00      1.00      1.00        33
          17       0.90      1.00      0.95        18
          18       0.00      0.00      0.00         4
           2       0.89      1.00      0.94        17
          20       1.00      1.00      1.00         5
          21       1.00      1.00      1.00         2
          22       1.00      1.00      1.00         3
          23       1.00      0.33      0.50         3
          24       0.91      1.00      0.95        20
          26       1.00      1.00      1.00        25
          27       1.00      1.00      1.00         9
          28       1.00      1.00      1.00        81
          29       1.00      1.00      1.00         4
           3       0.89      1.00      0.94        47
          30       1.00      1.00      1.00        31
          31       1.00      1.00      1.00         8
          32       1.00      1.00      1.00         5
          33       1.00      1.00      1.00         2
          34       1.00      1.00      1.00         6
          35       1.00      1.00      1.00        34
          36       1.00      1.00      1.00         7
          37       1.00      1.00      1.00        10
          38       1.00      1.00      1.00         5
          39       1.00      1.00      1.00         3
           4       1.00      1.00      1.00        21
          40       1.00      1.00      1.00         4
          41       0.50      1.00      0.67         2
          42       1.00      0.56      0.71         9
          43       0.88      1.00      0.94        15
          44       1.00      1.00      1.00         5
          45       0.75      1.00      0.86         6
          46       1.00      0.25      0.40         8
          47       1.00      0.50      0.67         4
          48       1.00      1.00      1.00         4
          49       0.56      1.00      0.71         5
           5       1.00      1.00      1.00        38
          50       1.00      1.00      1.00        10
          51       1.00      0.50      0.67         4
          52       1.00      1.00      1.00        10
          54       1.00      1.00      1.00        63
          55       1.00      1.00      1.00        27
          56       1.00      1.00      1.00        20
          57       1.00      1.00      1.00         2
           6       1.00      0.88      0.93        16
           7       1.00      1.00      1.00        28
           8       1.00      1.00      1.00         2

    accuracy                           0.97       834
   macro avg       0.95      0.93      0.92       834
weighted avg       0.97      0.97      0.97       834

C:\Users\hp\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-sco
re are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this b
ehavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\hp\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-sco
re are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this b
ehavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\hp\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-sco
re are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this b
ehavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [506]:
```
from sklearn import metrics
print(metrics.accuracy_score(y_pred, y_val))
```

```
0.973621103117506
```

In [507]:
```
images = list(Path('C:/Users/hp/Downloads/traffic_Data/TEST/').glob(r'**/*.png'))
labels = list(map(lambda path: int(os.path.split(os.path.split(path)[1])[1].split('_')[0][1:]), images))
test_images = pd.Series(images).astype(str)
test_labels = pd.Series(labels).astype(str)
```

In [508]:
```
test_data=[]
for elem in range(len(test_images)):
    image = cv2.imread(test_images[elem])
    image_resized = cv2.resize(image, (size, size), interpolation = cv2.INTER_AREA)
    test_data.append(np.array(image_resized))
```

In [509]:
```
test_data = np.array(test_data)
test_data = test_data.reshape((test_data.shape[0], 128*128*3))
test_data = test_data.astype(float)/255
test_labels = np.array(test_labels)
```

In [510]:
```
test_pred=classifier.predict(test_data)
```

In [511]:
```
print(metrics.accuracy_score(test_pred, test_labels))
```

```
0.5175526579739218
```

In [512]:
```
print(classification_report(test_labels, test_pred))
```

```
              precision    recall  f1-score   support

           0       0.25      0.71      0.37        14
           1       1.00      0.17      0.29        12
          10       1.00      0.90      0.95        60
          11       0.85      0.77      0.81       130
          12       1.00      1.00      1.00        22
          13       1.00      0.17      0.30        92
          14       0.36      0.67      0.47        12
          15       0.00      0.00      0.00        36
          16       0.68      0.71      0.69        76
          17       0.62      0.90      0.74        84
           2       0.71      0.17      0.27        60
          20       0.00      0.00      0.00         2
          21       0.00      0.00      0.00        12
          22       0.00      0.00      0.00         8
          23       0.00      0.00      0.00        10
          24       0.18      0.31      0.23        26
          25       0.00      0.00      0.00         2
          26       0.87      0.70      0.78       134
          27       1.00      0.42      0.59        24
          28       0.67      0.53      0.59        68
          29       0.62      1.00      0.76        26
           3       0.39      0.86      0.53        84
          30       0.60      0.35      0.44        34
          31       0.50      0.22      0.31        18
          32       0.00      0.00      0.00         2
          34       0.00      0.00      0.00         8
          35       0.13      0.61      0.21        46
          36       1.00      0.17      0.29        12
          37       0.33      0.31      0.32        26
          38       0.00      0.00      0.00        40
          39       1.00      0.20      0.33        30
           4       0.63      0.41      0.50        58
          40       1.00      0.50      0.67         8
          41       0.00      0.00      0.00         8
          42       1.00      0.11      0.20        18
          43       0.23      0.14      0.17       116
          44       0.00      0.00      0.00        24
          45       0.03      1.00      0.06         2
          46       0.00      0.00      0.00        14
          47       1.00      0.40      0.57        10
          48       1.00      1.00      1.00         6
          49       0.40      0.10      0.15        42
           5       0.36      0.40      0.38        50
          50       0.00      0.00      0.00        20
          51       1.00      0.50      0.67         4
          52       0.75      0.20      0.32        30
          53       1.00      1.00      1.00         2
          54       0.60      0.94      0.73       176
          55       0.85      0.97      0.90        58
          56       0.23      0.50      0.32        40
          57       1.00      1.00      1.00         4
           6       0.63      0.80      0.71        30
           7       0.25      0.16      0.20        50
           8       1.00      0.29      0.44        14

    accuracy                           0.52      1994
   macro avg       0.51      0.41      0.39      1994
weighted avg       0.58      0.52      0.49      1994

C:\Users\hp\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-sco
re are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this b
ehavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\hp\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-sco
re are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this b
ehavior.
  _warn_prf(average, modifier, msg_start, len(result))
C:\Users\hp\anaconda3\lib\site-packages\sklearn\metrics\_classification.py:1318: UndefinedMetricWarning: Precision and F-sco
re are ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this b
ehavior.
  _warn_prf(average, modifier, msg_start, len(result))
```

In [513]:
```
for i in range(len(test_labels)):
    print('original: ', test_labels[i], ' predicted: ', test_pred[i])
```

```
original:  0  predicted:  56
original:  0  predicted:  0
original:  0  predicted:  4
original:  0  predicted:  0
original:  0  predicted:  0
original:  0  predicted:  0
original:  0  predicted:  0
original:  0  predicted:  56
original:  0  predicted:  0
original:  0  predicted:  4
original:  0  predicted:  0
original:  0  predicted:  0
original:  0  predicted:  0
original:  0  predicted:  0
original:  1  predicted:  6
original:  1  predicted:  1
original:  1  predicted:  4
original:  1  predicted:  5
original:  1  predicted:  56
```

In [514]:
```
#classifier.save("C:/Users/hp/Downloads/traffic_Data/traffic_recognition.h5")

import pickle
pickle.dump(classifier, open('C:/Users/hp/Downloads/traffic_Data/model.pkl', 'wb'))
```

In [515]:
```
pickled_model = pickle.load(open('C:/Users/hp/Downloads/traffic_Data/model.pkl', 'rb'))
test_pred=pickled_model.predict(test_data)
print(metrics.accuracy_score(test_pred, test_labels))
```

```
0.5175526579739218
```