SOURCE CODE

Importing the data :

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import cv2
import tensorflow as tf
from PIL import Image
import os
os.chdir('D:/Traffic_Sign_Recognition')
from sklearn.model_selection import train_test_split
from keras.utils import to_categorical
from keras.models import Sequential, load_model
from keras.layers import Conv2D, MaxPool2D, Dense, Flatten,
Dropout
classes = 43
cur_path = os.getcwd()
cur_path
```

Processing the image :

```
for i in range(classes):
    path = os.path.join(cur_path,'train',str(i))
images = os.listdir(path)
    for a in images:
        try:
            image = Image.open(path + '\\'+ a)
```

```
            image = image.resize((30,30))
            image = np.array(image)
            data.append(image)
            labels.append(i)
        except Exception as e:
            print(e)
```

Converting lists into numpy arrays :

```
data = np.array(data)
labels = np.array(labels)
```

Save Labels & Data for future use

```
# os.mkdir('training')
np.save('./training/data',data)
 np.save('./training/target',labels)
```

Load data & Labels :

```
data=np.load('./training/data.npy')
labels=np.load('./training/target.npy')
print(data.shape, labels.shape)
```

(78418, 30, 30, 3) (78418,)

```
X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.2, random_state=0)

print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
```

(62734, 30, 30, 3) (15684, 30, 30, 3) (62734,) (15684,)

Convert labels to one hot encoding :

```
y_train = to_categorical(y_train, 43)
```

y_test = to_categorical(y_test, 43)

Creating Model :

model = Sequential()

model.add(Conv2D(filters=32, kernel_size=(5,5), activation='relu', input_shape=X_train.shape[1:]))

model.add(Conv2D(filters=32, kernel_size=(5,5), activation='relu'))

model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Dropout(rate=0.25))

model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))

model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))

model.add(MaxPool2D(pool_size=(2, 2)))

model.add(Dropout(rate=0.25))

model.add(Flatten())

model.add(Dense(256, activation='relu'))

model.add(Dropout(rate=0.5))

 # We have 43 classes that's why we have defined 43 in the dense

model.add(Dense(43, activation='softmax'))

Compiling the model :

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

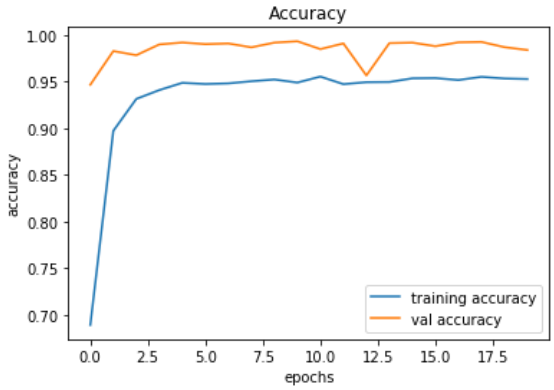
epochs = 20

history = model.fit(X_train, y_train, batch_size=32, epochs=epochs, validation_data=(X_test, y_test))

Accuracy :

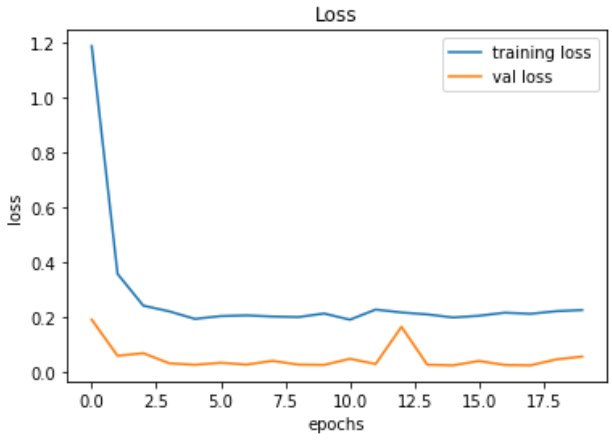
plt.figure(0)

plt.show()



Loss :

```
plt.plot(history.history['loss'], label='training loss')
plt.plot(history.history['val_loss'], label='val loss')
plt.title('Loss')
plt.xlabel('epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```



LEt's do testing on Test data :

```
def testing(testcsv):
    y_test = pd.read_csv(testcsv)
```

```
    label = y_test["ClassId"].values
    imgs = y_test["Path"].values
    data=[]
    for img in imgs:
        image = Image.open(img)
        image = image.resize((30,30))
        data.append(np.array(image))
    X_test=np.array(data)
    return X_test,label
X_test, label = testing('Test.csv')
Y_pred = model.predict_classes(X_test)
Y_pred
array([16,  1, 38, ...,  5,  7, 10], dtype=int64)
```

Accuracy with the test data

```
from sklearn.metrics import accuracy_score
print(accuracy_score(label, Y_pred))
0.9403008709422012
```

Save the model

```
model.save("./training/TSR.h5")
```

Load the Model

```
import os
os.chdir(r'D:\Traffic_Sign_Recognition')
from keras.models import load_model
 model = load_model('./training/TSR.h5')
from PIL import Image
```

```
import numpy as np
import matplotlib.pyplot as plt
def test_on_img(img):
    data=[]
    image = Image.open(img)
    image = image.resize((30,30))
    data.append(np.array(image))
    X_test=np.array(data)
    Y_pred = model.predict_classes(X_test)
    return image,Y_pred
plt.show()
```

OUTPUT

Predicted traffic sign is: Bumpy road