In [1]:
```
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
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
```

```
WARNING:tensorflow:From C:\Users\arnak\anaconda3\Lib\site-packages\keras\src
\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecate
d. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.
```

In [2]:
```
data = []
labels = []
classes = 43
cur_path = os.getcwd()

#Retrieving the images and their labels
for i in range(classes):
    path = os.path.join(cur_path, 'D:\\NIT\\DATASCIENCE\\ARNAK TASK\\traffic s
    images = os.listdir(path)

    for a in images:
        try:
            image = Image.open(path + '\\'+ a)
            image = image.resize((30,30))
            image = np.array(image)
            #sim = Image.fromarray(image)
            data.append(image)
            labels.append(i)
        except:
            print("Error loading image")
```

In [3]:
```
#Converting lists into numpy arrays
data = np.array(data)
labels = np.array(labels)

print(data.shape, labels.shape)
#Splitting training and testing dataset
X_train, X_test, y_train, y_test = train_test_split(data, labels, test_size=0.

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

#Converting the labels into one hot encoding
y_train = to_categorical(y_train, 43)
y_test = to_categorical(y_test, 43)
```

```
(39209, 30, 30, 3) (39209,)
(31367, 30, 30, 3) (7842, 30, 30, 3) (31367,) (7842,)
```

In [4]:

```
#Building the model
model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(5,5), activation='relu', input_shape
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))
model.add(Dense(43, activation='softmax'))

#Compilation of the model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['acc

epochs = 15
history = model.fit(X_train, y_train, batch_size=32, epochs=epochs, validation
```

```
WARNING:tensorflow:From C:\Users\arnak\anaconda3\Lib\site-packages\keras\src
\backend.py:873: The name tf.get_default_graph is deprecated. Please use tf.c
ompat.v1.get_default_graph instead.

WARNING:tensorflow:From C:\Users\arnak\anaconda3\Lib\site-packages\keras\src
\layers\pooling\max_pooling2d.py:161: The name tf.nn.max_pool is deprecated.
Please use tf.nn.max_pool2d instead.

WARNING:tensorflow:From C:\Users\arnak\anaconda3\Lib\site-packages\keras\src
\optimizers\__init__.py:309: The name tf.train.Optimizer is deprecated. Pleas
e use tf.compat.v1.train.Optimizer instead.

Epoch 1/15
WARNING:tensorflow:From C:\Users\arnak\anaconda3\Lib\site-packages\keras\src
\utils\tf_utils.py:492: The name tf.ragged.RaggedTensorValue is deprecated. P
lease use tf.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From C:\Users\arnak\anaconda3\Lib\site-packages\keras\src
\engine\base_layer_utils.py:384: The name tf.executing_eagerly_outside_functi
ons is deprecated. Please use tf.compat.v1.executing_eagerly_outside_function
s instead.

981/981 [==============================] - 31s 30ms/step - loss: 1.9460 - acc
uracy: 0.4954 - val_loss: 0.5745 - val_accuracy: 0.8625
Epoch 2/15
981/981 [==============================] - 28s 29ms/step - loss: 0.8173 - acc
uracy: 0.7619 - val_loss: 0.3108 - val_accuracy: 0.9206
Epoch 3/15
981/981 [==============================] - 29s 29ms/step - loss: 0.5270 - acc
uracy: 0.8448 - val_loss: 0.1759 - val_accuracy: 0.9531
Epoch 4/15
981/981 [==============================] - 29s 29ms/step - loss: 0.3572 - acc
uracy: 0.8928 - val_loss: 0.1175 - val_accuracy: 0.9662
Epoch 5/15
981/981 [==============================] - 30s 31ms/step - loss: 0.3031 - acc
uracy: 0.9123 - val_loss: 0.0866 - val_accuracy: 0.9755
Epoch 6/15
981/981 [==============================] - 30s 30ms/step - loss: 0.2669 - acc
uracy: 0.9225 - val_loss: 0.1013 - val_accuracy: 0.9725
Epoch 7/15
981/981 [==============================] - 29s 29ms/step - loss: 0.2426 - acc
uracy: 0.9293 - val_loss: 0.0899 - val_accuracy: 0.9731
Epoch 8/15
981/981 [==============================] - 29s 29ms/step - loss: 0.2368 - acc
uracy: 0.9322 - val_loss: 0.0681 - val_accuracy: 0.9793
Epoch 9/15
981/981 [==============================] - 29s 29ms/step - loss: 0.2193 - acc
uracy: 0.9374 - val_loss: 0.0652 - val_accuracy: 0.9819
Epoch 10/15
981/981 [==============================] - 29s 30ms/step - loss: 0.2140 - acc
uracy: 0.9418 - val_loss: 0.0571 - val_accuracy: 0.9844
Epoch 11/15
981/981 [==============================] - 29s 29ms/step - loss: 0.2301 - acc
uracy: 0.9393 - val_loss: 0.0729 - val_accuracy: 0.9805
Epoch 12/15
981/981 [==============================] - 29s 30ms/step - loss: 0.2029 - acc
uracy: 0.9436 - val_loss: 0.0661 - val_accuracy: 0.9833
```

```
Epoch 13/15
981/981 [==============================] - 28s 29ms/step - loss: 0.1926 - acc
uracy: 0.9477 - val_loss: 0.0775 - val_accuracy: 0.9799
Epoch 14/15
981/981 [==============================] - 29s 30ms/step - loss: 0.2233 - acc
uracy: 0.9418 - val_loss: 0.0460 - val_accuracy: 0.9879
Epoch 15/15
981/981 [==============================] - 29s 29ms/step - loss: 0.1949 - acc
uracy: 0.9479 - val_loss: 0.0764 - val_accuracy: 0.9773
```

In [5]:
```
model.save("my_model.h5")
```

```
C:\Users\arnak\anaconda3\Lib\site-packages\keras\src\engine\training.py:3103:
UserWarning: You are saving your model as an HDF5 file via `model.save()`. Th
is file format is considered legacy. We recommend using instead the native Ke
ras format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(
```

In [6]:
```
#testing accuracy on test dataset
from sklearn.metrics import accuracy_score

y_test = pd.read_csv(r"D:\NIT\DATASCIENCE\ARNAK TASK\traffic sign\New folder\T

labels = y_test["ClassId"].values
imgs = y_test["Path"].values

data=[]
for img in imgs:
    print(img)
    image = Image.open(img)
    image = image.resize((30,30))
    data.append(np.array(image))
```

```
Test/00000.png
Test/00001.png
Test/00002.png
Test/00003.png
Test/00004.png
Test/00005.png
Test/00006.png
Test/00007.png
Test/00008.png
Test/00009.png
Test/00010.png
Test/00011.png
Test/00012.png
Test/00013.png
Test/00014.png
Test/00015.png
Test/00016.png
Test/00017.png
Test/00018.png
```

In [8]:
```
X_test=np.array(data)

pred = model.predict_generator(X_test)


# Get the predicted probabilities for each class
pred_probabilities = model.predict(X_test)

# Get the class with the highest probability for each sample
pred = np.argmax(pred_probabilities, axis=1)
```

```
C:\Users\arnak\AppData\Local\Temp\ipykernel_24588\3375125404.py:3: UserWarnin
g: `Model.predict_generator` is deprecated and will be removed in a future ve
rsion. Please use `Model.predict`, which supports generators.
  pred = model.predict_generator(X_test)

395/395 [==============================] - 3s 9ms/step
```

In [9]:
```
#Accuracy with the test data
from sklearn.metrics import accuracy_score
print(accuracy_score(labels, pred))
```

```
0.9381631037212985
```

In [ ]:
```

```