

Model Building

Importing Required Model Building Library

```
In [1]:  
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
In [2]:  
  
# Training Datagen  
train_datagen =  
ImageDataGenerator(rescale=1/255, zoom_range=0.2, horizontal_flip=True, vertical  
_flip=False)  
# Testing Datagen  
test_datagen = ImageDataGenerator(rescale=1/255)
```

```
In [3]:  
  
import tensorflow as tf  
import os
```

Initialize the model

```
In [4]:  
  
import tensorflow as tf  
import os  
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Dense, Conv2D, Flatten, Dropout,  
MaxPooling2D  
from tensorflow.keras.preprocessing.image import ImageDataGenerator  
import numpy as np  
import matplotlib.pyplot as plt  
import IPython.display as display  
from PIL import Image  
import pathlib
```

```
In [5]:  
  
import numpy as np  
import matplotlib.pyplot as plt #to view graph in colab itself  
import IPython.display as display  
from PIL import Image  
import pathlib
```

Applying ImageDataGenerator to Training set and Test set

```
In [6]:  
  
x_train=train_datagen.flow_from_directory('/content/drive/MyDrive/Nalaiyathir  
an/Dataset/training_set', target_size=(64, 64), batch_size=200,
```

```
class_mode='categorical',color_mode="grayscale")
Found 15750 images belonging to 9 classes.
```

In [7]:

```
x_test=test_datagen.flow_from_directory('/content/drive/MyDrive/Nalaiyathiran
/Dataset/test_set',target_size=(64,64),batch_size=200,
```

```
class_mode='categorical',color_mode="grayscale")
Found 2250 images belonging to 9 classes.
```

Length of training and test sets

In [8]:

```
a=len(x_train)
b=len(x_test)
```

In [9]:

```
print(a)
79
```

In [10]:

```
print(b)
12
```

Add layers

In [11]:

```
model=Sequential()
```

In []:

```
model.add(Convolution2D(32,(3,3),input_shape=(64,64,1),activation='relu'))
```

Add Pooling Layer

In []:

```
model.add(MaxPooling2D(pool_size=(2,2)))
```

Add dense layer

In []:

```
#First hidden layer
model.add(Dense(units=512,activation='relu'))
```

In []:

```
#Second hidden layer
model.add(Dense(units=261,activation='relu'))
```

In []:

```
#output layer  
model.add(Dense(units=9,activation='softmax'))
```

Compile The Model

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

Fit the model

```
In [ ]:  
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
```

Save the model

```
In [ ]:  
model.save('aslpng2.h5')
```