

Team Id:PNT2022TMID46656

Model Building

Fit And Save The Model

In []:

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In []:

```
# 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 []:

```
# Training Dataset  
x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/train_set', target_size=(64,64), class_mode='categorical', batch_size=900)  
# Testing Dataset  
x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/test_set', target_size=(64,64), class_mode='categorical', batch_size=900)
```

Found 15760 images belonging to 9 classes.

Found 2250 images belonging to 9 classes.

In []:

```
# Save Model Using Pickle import pandas from  
sklearn import model_selection from  
sklearn.linear_model import LogisticRegression  
import pickle
```

In []:

```
url =  
"https://raw.githubusercontent.com/jbrownlee/Datasets/master/pimaindians-diabetes.data.csv" names = ['preg', 'plas', 'pres', 'skin', 'test',  
'mass', 'pedi', 'age',  
'class'] dataframe = pandas.read_csv(url,  
names=names) array = dataframe.values  
X = array[:,0:8] Y  
= array[:,8]  
test_size = 0.33  
seed = 7  
X_train, X_test, Y_train, Y_test = model_selection.train_test_split(X, Y,  
test_size=test_size, random_state=seed)
```

In []:

```
# Fit the model on training set  
model = LogisticRegression()  
model.fit(X_train, Y_train) # save the  
model to disk filename =
```

```
'finalized_model.sav' pickle.dump(model,
open(filename, 'wb'))
```

```
# load the model from disk
loaded_model = pickle.load(open(filename, 'rb')) result
= loaded_model.score(X_test, Y_test) print(result)
```

```
In []:
```

```
print("Len x-train : ", len(x_train)) print("Len
x-test : ", len(x_test))
```

```
Len x-train : 18 Len
x-test : 3
```

```
In []:
```

```
# The Class Indices in Training Dataset x_train.class_indices
```

```
Out[]:
```

```
{'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8}
```

Model Creation

```
In []:
```

```
# Importing Libraries from
tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

```
In []:
```

```
# Creating Model model=Sequential()
```

```
In []:
```

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

```
In []:
```

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

```
In []:
```

```
model.add(Flatten())
```

```
In []:
```

```
# Adding Dense Layers model.add(Dense(300,activation='relu'))
model.add(Dense(150,activation='relu'))
model.add(Dense(9,activation='softmax'))
```

```
In []:
```

```
# Compiling the Model
model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['acc
uracy'])
```

```
In []:
```

```
# Fitting the Model Generator
model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_data=x_test,validation_steps=len(x_test))
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: UserWarning:
`Model.fit_generator` is deprecated and will be removed in a future version.
Please use `Model.fit`, which supports generators.
```

```
Epoch 1/10 18/18 [=====] - 92s 5s/step - loss:
0.0049 - accurac y: 0.9994 - val_loss: 0.2635 - val_accuracy: 0.9773
Epoch 2/10 18/18 [=====] - 90s 5s/step - loss:
0.0040 - accurac y: 0.9995 - val_loss: 0.2074 - val_accuracy: 0.9773
Epoch 3/10
18/18 [=====] - 87s 5s/step - loss: 0.0041 -
accurac y: 0.9995 - val_loss: 0.2460 - val_accuracy: 0.9773
Epoch 4/10 18/18 [=====] - 91s 5s/step - loss:
0.0041 - accurac y: 0.9992 - val_loss: 0.2470 - val_accuracy: 0.9782
Epoch 5/10 18/18 [=====] - 88s 5s/step - loss:
0.0037 - accurac y: 0.9993 - val_loss: 0.2439 - val_accuracy: 0.9782
Epoch 6/10 18/18 [=====] - 88s 5s/step - loss:
0.0024 - accurac y: 0.9997 - val_loss: 0.2852 - val_accuracy: 0.9782
Epoch 7/10 18/18 [=====] - 91s 5s/step - loss:
0.0023 - accurac y: 0.9997 - val_loss: 0.2589 - val_accuracy: 0.9782
Epoch 8/10
18/18 [=====] - 93s 5s/step - loss: 0.0014 -
accurac y: 1.0000 - val_loss: 0.2523 - val_accuracy: 0.9782
Epoch 9/10 18/18 [=====] - 92s 5s/step - loss:
0.0013 - accurac y: 0.9999 - val_loss: 0.2269 - val_accuracy: 0.9778
Epoch 10/10
18/18 [=====] - 91s 5s/step - loss: 0.0012 -
accurac y: 0.9999 - val_loss: 0.2968 - val_accuracy: 0.9782
```

Out[]:

Saving the Model

In []:

```
model.save('asl_model_84_54.h5')
```