

FIT AND SAVE THE MODEL

Team ID : PNT2022TMID06334

Title: Real-Time Communication System Powered by AI for Specially Abled

Loading the Dataset & Image Data Generation

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

```
# 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)
```

```
# Training Dataset
```

```
x_train=train_datagen.flow_from_directory(r'content/drive/MyDrive/Dataset/training_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=(64,6
```

```
4), class_mode='categorical', batch_size=900)
```

```
Found 15760 images belonging to 9 classes.
```

```
Found 2250 images belonging to 9 classes.
```

```
print('Len x-train : ', len(x_train))
```

```
print('Len x-test : ', len(x_test))
```

```
Len x-train : 18
```

```
Len x-test : 3
```

```
# The Class Indices in Training Dataset
```

```
x_train.class_indices
```

```
{&#39;A&#39;: 0, &#39;B&#39;: 1, &#39;C&#39;: 2, &#39;D&#39;: 3, &#39;E&#39;: 4, &#39;F&#39;: 5, &#39;G&#39;: 6, &#39;H&#39;: 7, &#39;I&#39;: 8}
```

Model Creation

Importing Libraries

```
from tensorflow.keras.models import Sequential
```

```
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

Creating Model

```
model=Sequential()
```

Adding Layers

```
model.add(Convolution2D(32,(3,3),activation=&#39;relu&#39;,input_shape=(64,64,3)))
```

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

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

Adding Dense Layers

```
model.add(Dense(300,activation=&#39;relu&#39;))
```

```
model.add(Dense(150,activation=&#39;relu&#39;))
```

```
model.add(Dense(9,activation=&#39;softmax&#39;))
```

Compiling the Model

```
model.compile(loss=&#39;categorical_crossentropy&#39;,optimizer=&#39;adam&#39;,metrics=[&#39;accuracy&#39;])
```

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 - accuracy: 0.9994 - val_loss: 0.2635 - val_accuracy: 0.9773

Epoch 2/10

18/18 [=====] - 90s 5s/step - loss: 0.0040 - accuracy: 0.9995 - val_loss: 0.2074 - val_accuracy: 0.9773

Epoch 3/10

18/18 [=====] - 87s 5s/step - loss: 0.0041 - accuracy: 0.9995 - val_loss: 0.2460 - val_accuracy: 0.9773

Epoch 4/10

18/18 [=====] - 91s 5s/step - loss: 0.0041 - accuracy: 0.9992 - val_loss: 0.2470 - val_accuracy: 0.9782

Epoch 5/10

18/18 [=====] - 88s 5s/step - loss: 0.0037 - accuracy: 0.9993 - val_loss: 0.2439 - val_accuracy: 0.9782

Epoch 6/10

18/18 [=====] - 88s 5s/step - loss: 0.0024 - accuracy: 0.9997 - val_loss: 0.2852 - val_accuracy: 0.9782

Epoch 7/10

18/18 [=====] - 91s 5s/step - loss: 0.0023 - accuracy: 0.9997 - val_loss: 0.2589 - val_accuracy: 0.9782

Epoch 8/10

18/18 [=====] - 93s 5s/step - loss: 0.0014 - accuracy: 1.0000 - val_loss: 0.2523 - val_accuracy: 0.9782

Epoch 9/10

18/18 [=====] - 92s 5s/step - loss: 0.0013 - accuracy: 0.9999 - val_loss: 0.2269 - val_accuracy: 0.9778

Epoch 10/10

18/18 [=====] - 91s 5s/step - loss: 0.0012 - accuracy: 0.9999 - val_loss: 0.2968 - val_accuracy: 0.9782

Saving the Model

model.save('asl_model_84_54.h5')

TEAM ID: PNT2022TMID06334