

Delivery of Sprint 2

Date	31 October 2022
Team ID	PNT2022TMID21808
Project Name	Real time Communication System Powered by AI for Specially Abled

Model Building

Import The Required Model Building Libraries

```
In [6]: from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Dropout
from keras.layers import Flatten
```

Initialize The Model

```
In [7]: model=Sequential()
```

Add The Convolution Layer

```
In [10]: model.add(Convolution2D(32,(3,3),activation="relu",input_shape=(64,64,3)))
#No of feature detectors, size of feature detector, image size, activation function
```

Add The Pooling Layer

```
In [11]: model.add(MaxPooling2D(pool_size=(2,2)))
```

Add The Flatten Layer

```
In [12]: model.add(Flatten())
```

Adding The Dense Layers

```
In [13]: model.add(Dense(200,activation='relu'))
```

```
In [15]: model.add(Dense(9,activation="softmax"))
```

Compile The Model

```
In [16]: model.compile(loss="categorical_crossentropy",metrics=["accuracy"],optimizer='adam')
```

```
In [17]: len(x_train)
```

```
Out[17]: 525
```

```
In [18]: len(x_test)
```

```
Out[18]: 75
```

```
In [32]: model.fit(x_train,epochs=8,validation_data=x_test,steps_per_epoch=len(x_train),validation_steps=len(x_test))
```

Epoch 1/8
525/525 [=====] - 620s 1s/step - loss: 0.6116 - accuracy: 0.7691 - val_loss: 0.3006 - val_accuracy: 0.9329
Epoch 2/8
525/525 [=====] - 223s 425ms/step - loss: 0.1041 - accuracy: 0.9683 - val_loss: 0.0779 - val_accuracy: 0.9858
Epoch 3/8
525/525 [=====] - 132s 250ms/step - loss: 0.0592 - accuracy: 0.9829 - val_loss: 0.1236 - val_accuracy: 0.9760
Epoch 4/8
525/525 [=====] - 104s 198ms/step - loss: 0.0431 - accuracy: 0.9879 - val_loss: 0.2067 - val_accuracy: 0.9742
Epoch 5/8
525/525 [=====] - 107s 204ms/step - loss: 0.0322 - accuracy: 0.9912 - val_loss: 0.0713 - val_accuracy: 0.9800
Epoch 6/8
525/525 [=====] - 113s 216ms/step - loss: 0.0348 - accuracy: 0.9895 - val_loss: 0.1267 - val_accuracy: 0.9787
Epoch 7/8
525/525 [=====] - 101s 193ms/step - loss: 0.0293 - accuracy: 0.9926 - val_loss: 0.1558 - val_accuracy: 0.9751
Epoch 8/8
525/525 [=====] - 107s 205ms/step - loss: 0.0222 - accuracy: 0.9940 - val_loss: 0.1998 - val_accuracy: 0.9769

```
Out[32]: <keras.callbacks.History at 0x20f394a4e20>
```

```
In [33]: model.save("C:/Users/rajes/Downloads/signlanguage-new.h5")
```

Test the Model

Test the Model

Import The Packages And Load The Saved Model

```
In [35]: from keras.models import load_model
import numpy as np
import h5py
import cv2
```

```
In [36]: from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import numpy as np
```

```
In [37]: model = load_model("C:/Users/rajes/Downloads/signlanguage-new.h5")
```

Load The Test Image, Pre-Process It And Predict

```
In [39]: img = image.load_img(r"C:\Users\rajes\Desktop\Dataset\test_set\A\8.png",target_size = (64,64,1))
img
```

```
Out[39]:
```



```
In [41]: from skimage.transform import resize
def detect(frame):
    img=image.img_to_array(frame)
    img = resize(img,(64,64,1))
    img = np.expand_dims(img,axis=0)
    pred=np.argmax(model.predict(img))
    op=['A','B','C','D','E','F','G','H','I']
    print("THE PREDICTED LETTER IS ",op[pred])
```

```
In [42]: from skimage.transform import resize
def detect(frame):
    img=resize(frame,(64,64,1))
    img=np.expand_dims(img,axis=0)
    if(np.max(img)>1):
        img=img/255.0
        prediction=model.predict(img)
        print(prediction)
        prediction=model.predict_classes(img)
        print(prediction)
```

```
In [43]: frame=cv2.imread(r"C:\Users\rajes\Desktop\Dataset\test_set\A\8.png")
data=detect(frame)
```

```
In [44]: type(img)
```

```
Out[44]: PIL.Image.Image
```

```
In [45]: x = image.img_to_array(img)
x
```

```
Out[45]: array([[0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.],
                ...,
                [0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.]],

               [[0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.],
                ...,
                [0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.]],

               [[0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.],
                ...,
                [0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.]],

               ...,

               [[0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.],
                ...,
                [0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.]],

               [[0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.],
                ...,
                [0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.]],

               [[0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.],
                ...,
                [0., 0., 0.],
                [0., 0., 0.],
                [0., 0., 0.]]], dtype=float32)
```

```
In [46]: x.shape
```

```
Out[46]: (64, 64, 3)
```

```
In [47]: x=np.expand_dims(x,axis=0)
x.shape
```

```
Out[47]: (1, 64, 64, 3)
```

```
In [48]: pred_prob = model.predict(x)

1/1 [=====] - 1s 1s/step
```

```
In [49]: pred_prob
```

```
Out[49]: array([[9.9954236e-01, 7.8000909e-13, 7.1030300e-08, 7.4072335e-07,
3.7532591e-04, 2.8473270e-12, 1.7780074e-05, 6.3732426e-05,
7.7890165e-09]], dtype=float32)
```

```
In [50]: class_name=["A","B","C","D","E","F","G","H","I"]
pred_id = pred_prob.argmax(axis=1)[0]
```

```
In [51]: pred_id
```

```
Out[51]: 0
```

```
In [52]: print("the alphabet is ",str(class_name[pred_id]))

the alphabet is  A
```