

MODEL BUILDING

Team ID	PNT2022TMID33016
Project Name	Real time communication for specially abled powered by AI

Import the libraries:

Model Building –Sprint 3

```
[ ] from tensorflow.keras.models import Sequential
    from tensorflow.keras.layers import Dense
    from tensorflow.keras.layers import Convolution2D
    from tensorflow.keras.layers import MaxPooling2D
    from tensorflow.keras.layers import Flatten
```

Creating the model:

```
[ ] #Creating the model
    model=Sequential()
```

Adding the layers:

```
▶ #Adding the layers
model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(64,64,1),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.summary()
```

Model: "sequential_5"

Layer (type)	Output Shape	Param #
conv2d_10 (Conv2D)	(None, 62, 62, 32)	320
max_pooling2d_4 (MaxPooling 2D)	(None, 31, 31, 32)	0
flatten_4 (Flatten)	(None, 30752)	0

=====
Total params: 320
Trainable params: 320
Non-trainable params: 0
=====

Adding the hidden layers:

```
▶ #adding hidden layers  
model.add(Dense(512, activation='relu'))  
model.add(Dense(9, activation='relu'))
```

Add the output layers:

```
[ ] #Adding the output layer  
model.add(Dense(9, activation='softmax'))
```

Compile the model:

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

53

```
[ ] len(x_test)
```

8

Fit the model:

```
[ ] ### model.fit_generator(x_train,steps_per_epoch=len(x_train),validation_data=x_test,validation_steps=len(x_test),epochs=10)
# 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))
#model.fit(x_train, epochs=100, verbose=1)
```

/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:3: UserWarning: 'Model.fit_generator' is deprecated and will be removed in a future version. Please use 'Model.fit'. This is separate from the 'ipykernel' package so we can avoid doing imports until

Epoch 1/10

53/53 [=====] - 98s 2s/step - loss: 1.5098 - accuracy: 0.4139 - val_loss: 1.2455 - val_accuracy: 0.6098

Epoch 2/10

53/53 [=====] - 98s 2s/step - loss: 0.9909 - accuracy: 0.6395 - val_loss: 0.7839 - val_accuracy: 0.8364

Epoch 3/10

53/53 [=====] - 96s 2s/step - loss: 0.6332 - accuracy: 0.8344 - val_loss: 0.6295 - val_accuracy: 0.8342

Epoch 4/10

53/53 [=====] - 96s 2s/step - loss: 0.4576 - accuracy: 0.8549 - val_loss: 0.5360 - val_accuracy: 0.8333

Epoch 5/10

53/53 [=====] - 98s 2s/step - loss: 0.2958 - accuracy: 0.8976 - val_loss: 0.3582 - val_accuracy: 0.9498

Epoch 6/10

53/53 [=====] - 95s 2s/step - loss: 0.1532 - accuracy: 0.9635 - val_loss: 0.3361 - val_accuracy: 0.9724

Epoch 7/10

53/53 [=====] - 97s 2s/step - loss: 0.1040 - accuracy: 0.9747 - val_loss: 0.3673 - val_accuracy: 0.9649

Epoch 8/10

53/53 [=====] - 98s 2s/step - loss: 0.0681 - accuracy: 0.9828 - val_loss: 0.3307 - val_accuracy: 0.9627

Epoch 9/10

53/53 [=====] - 97s 2s/step - loss: 0.0542 - accuracy: 0.9855 - val_loss: 0.3669 - val_accuracy: 0.9747

Epoch 10/10

53/53 [=====] - 99s 2s/step - loss: 0.0426 - accuracy: 0.9895 - val_loss: 0.3932 - val_accuracy: 0.9627

<keras.callbacks.History at 0x7fb73f56f0d0>

Save the model:

```
[ ] model.save('zxcvb.h5')
```