

Team Id: PNT2022TMID22447

Project: Real-Time Communication system powered by AI for specially abled

Import The Required Model Building Libraries

```
#import imagedatagenerator
from keras.preprocessing.image import ImageDataGenerator

#training datagen
train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_

#testing datagen
test_datagen=ImageDataGenerator(rescale=1./255)
```

IMPORTING tensorflow

```
import tensorflow as tf
import os
```

IMPORTING LIBRARIES TO INITIALIZE NEURAL NETWORK LAYER

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

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
```

Unzipping the dataset

```
!unzip '/content/conversation engine for deaf and dumb.zip'

inflating: Dataset/training_set/I/947.png
inflating: Dataset/training_set/I/948.png
inflating: Dataset/training_set/I/949.png
inflating: Dataset/training_set/I/95.png
```

```
inflating: Dataset/training_set/I/950.png
inflating: Dataset/training_set/I/951.png
inflating: Dataset/training_set/I/952.png
inflating: Dataset/training_set/I/953.png
inflating: Dataset/training_set/I/954.png
inflating: Dataset/training_set/I/955.png
inflating: Dataset/training_set/I/956.png
inflating: Dataset/training_set/I/957.png
inflating: Dataset/training_set/I/958.png
inflating: Dataset/training_set/I/959.png
inflating: Dataset/training_set/I/96.png
inflating: Dataset/training_set/I/960.png
inflating: Dataset/training_set/I/961.png
inflating: Dataset/training_set/I/962.png
inflating: Dataset/training_set/I/963.png
inflating: Dataset/training_set/I/964.png
inflating: Dataset/training_set/I/965.png
inflating: Dataset/training_set/I/966.png
inflating: Dataset/training_set/I/967.png
inflating: Dataset/training_set/I/968.png
inflating: Dataset/training_set/I/969.png
inflating: Dataset/training_set/I/97.png
inflating: Dataset/training_set/I/970.png
inflating: Dataset/training_set/I/971.png
inflating: Dataset/training_set/I/972.png
extracting: Dataset/training_set/I/973.png
inflating: Dataset/training_set/I/974.png
inflating: Dataset/training_set/I/975.png
inflating: Dataset/training_set/I/976.png
inflating: Dataset/training_set/I/977.png
inflating: Dataset/training_set/I/978.png
inflating: Dataset/training_set/I/979.png
inflating: Dataset/training_set/I/98.png
inflating: Dataset/training_set/I/980.png
inflating: Dataset/training_set/I/981.png
inflating: Dataset/training_set/I/982.png
extracting: Dataset/training_set/I/983.png
inflating: Dataset/training_set/I/984.png
inflating: Dataset/training_set/I/985.png
inflating: Dataset/training_set/I/986.png
inflating: Dataset/training_set/I/987.png
inflating: Dataset/training_set/I/988.png
inflating: Dataset/training_set/I/989.png
inflating: Dataset/training_set/I/99.png
inflating: Dataset/training_set/I/990.png
inflating: Dataset/training_set/I/991.png
inflating: Dataset/training_set/I/992.png
extracting: Dataset/training_set/I/993.png
inflating: Dataset/training_set/I/994.png
inflating: Dataset/training_set/I/995.png
extracting: Dataset/training_set/I/996.png
inflating: Dataset/training_set/I/997.png
inflating: Dataset/training_set/I/998.png
inflating: Dataset/training_set/I/999.png
```

Applying ImageDataGenerator to training set

```
x_train=train_datagen.flow_from_directory('/content/Dataset/training_set',target_size=(64,64),
                                         class_mode='categorical',color_mode="grayscale")
```

Found 15750 images belonging to 9 classes.

Applying ImageDataGenerator to test set

```
x_test=test_datagen.flow_from_directory('/content/Dataset/test_set',target_size=(64,64),batch_size=32,
                                         class_mode='categorical',color_mode="grayscale")
```

Found 2250 images belonging to 9 classes.

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

Length of training set

```
print(a)
```

79

Length of test set

```
print(b)
```

12

Add Layers

```
#create model
model=Sequential()
```

Add The Convolution Layer

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

Add Pooling Layer

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

Add The Flatten Layer

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

Adding The Dense Layers

```
#1st hidden layer
model.add(Dense(units=512,activation='relu'))
#2nd hidden layer
model.add(Dense(units=261,activation='relu'))

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

Compile The Model

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

[Colab paid products](#) - [Cancel contracts here](#)

✓ 0s completed at 1:34 PM

