```
import numpy as np
import tensorflow
import keras
```

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

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

```
from tensorflow.keras import layers
```

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

```
from tensorflow.keras.layers import Conv2D, MaxPooling2D
```

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

```
#setting parameter for Image Data agumentation to the traing data
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
#Image Data agumentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
```

```
#performing data agumentation to train data
x_train = 
train_datagen.flow_from_directory(r'/content/drive/MyDrive/ibm/train',
                                            target_size=(64, 64),
                                            batch_size=3,
                                            color_mode='grayscale',
                                            class_mode='categorical')
#performing data agumentation to test data
x_test = 
test_datagen.flow_from_directory(r'/content/drive/MyDrive/ibm/test',
                                          target_size=(64, 64),
                                          batch_size=3,
                                          color_mode='grayscale',
                                          class_mode='categorical')
```

```
Found 594 images belonging to 6 classes.
Found 30 images belonging to 6 classes.
```

```
classifier  = Sequential()
```

```
import numpy as np
from keras.preprocessing import image
```

```
classifier.add(Conv2D(32,(3,3), input_shape=(64,64,1), 
activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2,2)))
classifier.add(Conv2D(32,(3,3), activation='relu'))
classifier.add(MaxPooling2D(pool_size=(2,2)))
classifier.add(Flatten())
```

```
classifier.add(Dense(units=128, activation='relu'))
classifier.add(Dense(units=6, activation='softmax'))
classifier.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_3 (Conv2D)           (None, 62, 62, 32)        320

 max_pooling2d_3 (MaxPooling  (None, 31, 31, 32)       0
 2D)

 conv2d_4 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_4 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten_1 (Flatten)         (None, 6272)              0

 dense (Dense)               (None, 128)               802944

 dense_1 (Dense)             (None, 6)                 774

 dense_2 (Dense)             (None, 128)               896

 dense_3 (Dense)             (None, 6)                 774

=================================================================
Total params: 814,956
Trainable params: 814,956
Non-trainable params: 0
_________________________________________________________________
```

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

```
classifier.fit_generator(
    generator = x_train, steps_per_epoch = len(x_train),
    epochs = 20, validation_data = x_test, validation_steps =
len(x_test)
)
```

```
/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`, which supports
generators.
```

```
  This is separate from the ipykernel package so we can avoid doing
imports until

Epoch 1/20
198/198 [==============================] - 160s 807ms/step - loss:
1.8016 - accuracy: 0.1684 - val_loss: 1.7928 - val_accuracy: 0.1667
Epoch 2/20
198/198 [==============================] - 7s 35ms/step - loss: 1.7978
- accuracy: 0.1347 - val_loss: 1.7919 - val_accuracy: 0.1667
Epoch 3/20
198/198 [==============================] - 7s 35ms/step - loss: 1.7981
- accuracy: 0.1582 - val_loss: 1.7923 - val_accuracy: 0.1667
Epoch 4/20
198/198 [==============================] - 7s 35ms/step - loss: 1.7958
- accuracy: 0.1465 - val_loss: 1.7918 - val_accuracy: 0.1667
Epoch 5/20
198/198 [==============================] - 7s 35ms/step - loss: 1.7961
- accuracy: 0.1549 - val_loss: 1.7920 - val_accuracy: 0.1667
Epoch 6/20
198/198 [==============================] - 7s 36ms/step - loss: 1.7971
- accuracy: 0.1397 - val_loss: 1.7918 - val_accuracy: 0.1667
Epoch 7/20
198/198 [==============================] - 7s 35ms/step - loss: 1.7977
- accuracy: 0.1465 - val_loss: 1.7922 - val_accuracy: 0.1667
Epoch 8/20
198/198 [==============================] - 7s 35ms/step - loss: 1.7978
- accuracy: 0.1229 - val_loss: 1.7919 - val_accuracy: 0.1667
Epoch 9/20
198/198 [==============================] - 7s 36ms/step - loss: 1.7968
- accuracy: 0.1397 - val_loss: 1.7923 - val_accuracy: 0.1667
Epoch 10/20
198/198 [==============================] - 7s 36ms/step - loss: 1.7958
- accuracy: 0.1616 - val_loss: 1.7921 - val_accuracy: 0.1667
Epoch 11/20
198/198 [==============================] - 7s 35ms/step - loss: 1.7955
- accuracy: 0.1431 - val_loss: 1.7920 - val_accuracy: 0.1667
Epoch 12/20
198/198 [==============================] - 7s 36ms/step - loss: 1.7972
- accuracy: 0.1380 - val_loss: 1.7922 - val_accuracy: 0.1667
Epoch 13/20
198/198 [==============================] - 7s 36ms/step - loss: 1.7955
- accuracy: 0.1549 - val_loss: 1.7921 - val_accuracy: 0.1667
Epoch 14/20
198/198 [==============================] - 7s 37ms/step - loss: 1.7968
- accuracy: 0.1229 - val_loss: 1.7926 - val_accuracy: 0.1667
Epoch 15/20
198/198 [==============================] - 7s 36ms/step - loss: 1.7969
- accuracy: 0.1330 - val_loss: 1.7920 - val_accuracy: 0.1667
Epoch 16/20
198/198 [==============================] - 7s 35ms/step - loss: 1.7956
```

```
- accuracy: 0.1549 - val_loss: 1.7918 - val_accuracy: 0.1667
Epoch 17/20
198/198 [==============================] - 7s 35ms/step - loss: 1.7959
- accuracy: 0.1498 - val_loss: 1.7918 - val_accuracy: 0.1667
Epoch 18/20
198/198 [==============================] - 7s 36ms/step - loss: 1.7952
- accuracy: 0.1515 - val_loss: 1.7918 - val_accuracy: 0.1667
Epoch 19/20
198/198 [==============================] - 7s 36ms/step - loss: 1.7952
- accuracy: 0.1414 - val_loss: 1.7920 - val_accuracy: 0.1667
Epoch 20/20
198/198 [==============================] - 7s 36ms/step - loss: 1.7948
- accuracy: 0.1431 - val_loss: 1.7919 - val_accuracy: 0.1667
```

```
<keras.callbacks.History at 0x7f2490208ed0>
```

```python
classifier.save('gesture.h5')
```

```python
model_json = classifier.to_json()
with open("model-bw.json", "w") as json_file:
    json_file.write(model_json)
```