

```
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Flatten, Dropout
from tensorflow.keras.layers import Convolution2D, MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [2]:

```
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
```

In [4]:

```
x_train =
train_datagen.flow_from_directory(r'C:\Users\mrith\Desktop\dataset\train',
                                  target_size=(64, 64),
                                  batch_size=3,
                                  color_mode='grayscale',
                                  class_mode='categorical')

x_test =
test_datagen.flow_from_directory(r'C:\Users\mrith\Desktop\dataset\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.
```

In [5]:

```
model = Sequential()
```

In [6]:

```
model.add(Convolution2D(32, (3, 3), input_shape=(64, 64, 1),
                        activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

In [7]:

```
model.add(Convolution2D(32, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
```

In [8]:

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

In [9]:

```
model.add(Dense(units=512 , activation='relu'))
```

In [10]:

```
model.add(Dense(units=6, activation='softmax'))
```

In [11]:

```
model.summary()
```

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
=====		
conv2d (Conv2D)	(None, 62, 62, 32)	320

max_pooling2d (MaxPooling2D	(None, 31, 31, 32)	0
)		
conv2d_1 (Conv2D)	(None, 29, 29, 32)	9248
max_pooling2d_1 (MaxPooling	(None, 14, 14, 32)	0
2D)		
flatten (Flatten)	(None, 6272)	0
dense (Dense)	(None, 512)	3211776
dense_1 (Dense)	(None, 6)	3078

```

=====
Total params: 3,224,422
Trainable params: 3,224,422
Non-trainable params: 0

```

```

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

```

In [12]:

```

model.fit_generator(x_train,
                    steps_per_epoch = 594/3 ,
                    epochs = 25,
                    validation_data = x_test,
                    validation_steps = 30/3 )

```

In [13]:

```

:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in
a future version. Please use `Model.fit`, which supports generators.

```

```

    model.fit_generator(x_train,
Epoch 1/25
198/198 [=====] - 15s 71ms/step - loss: 1.3935 - a
ccuracy: 0.4428 - val_loss: 0.7625 - val_accuracy: 0.7333
Epoch 2/25
198/198 [=====] - 14s 70ms/step - loss: 0.6228 - a
ccuracy: 0.7374 - val_loss: 0.4547 - val_accuracy: 0.9000
Epoch 3/25
198/198 [=====] - 14s 69ms/step - loss: 0.4162 - a
ccuracy: 0.8350 - val_loss: 0.2619 - val_accuracy: 0.9000
Epoch 4/25
198/198 [=====] - 13s 68ms/step - loss: 0.3183 - a
ccuracy: 0.8771 - val_loss: 0.2557 - val_accuracy: 0.9000
Epoch 5/25
198/198 [=====] - 13s 67ms/step - loss: 0.2563 - a
ccuracy: 0.9024 - val_loss: 0.2768 - val_accuracy: 0.9000
Epoch 6/25
198/198 [=====] - 14s 69ms/step - loss: 0.1513 - a
ccuracy: 0.9495 - val_loss: 0.3705 - val_accuracy: 0.9333
Epoch 7/25
198/198 [=====] - 14s 71ms/step - loss: 0.1610 - a
ccuracy: 0.9360 - val_loss: 0.3457 - val_accuracy: 0.8667
Epoch 8/25
198/198 [=====] - 14s 70ms/step - loss: 0.1166 - a
ccuracy: 0.9545 - val_loss: 0.3367 - val_accuracy: 0.9000
Epoch 9/25

```

```

198/198 [=====] - 14s 69ms/step - loss: 0.1109 - a
ccuracy: 0.9646 - val_loss: 0.2659 - val_accuracy: 0.9000
Epoch 10/25
198/198 [=====] - 14s 68ms/step - loss: 0.0823 - a
ccuracy: 0.9663 - val_loss: 0.5025 - val_accuracy: 0.9333
Epoch 11/25
198/198 [=====] - 13s 67ms/step - loss: 0.0799 - a
ccuracy: 0.9747 - val_loss: 0.2626 - val_accuracy: 0.9333
Epoch 12/25
198/198 [=====] - 9s 45ms/step - loss: 0.0919 - ac
curacy: 0.9714 - val_loss: 0.4762 - val_accuracy: 0.8333
Epoch 13/25
198/198 [=====] - 14s 70ms/step - loss: 0.0357 - a
ccuracy: 0.9899 - val_loss: 0.2444 - val_accuracy: 0.9000
Epoch 14/25
198/198 [=====] - 13s 65ms/step - loss: 0.0341 - a
ccuracy: 0.9899 - val_loss: 0.2582 - val_accuracy: 0.9333
Epoch 15/25
198/198 [=====] - 9s 44ms/step - loss: 0.0964 - ac
curacy: 0.9646 - val_loss: 0.1478 - val_accuracy: 0.9333
Epoch 16/25
198/198 [=====] - 8s 43ms/step - loss: 0.0482 - ac
curacy: 0.9832 - val_loss: 0.1486 - val_accuracy: 0.9667
Epoch 17/25
198/198 [=====] - 7s 37ms/step - loss: 0.0564 - ac
curacy: 0.9815 - val_loss: 0.2217 - val_accuracy: 0.9333
Epoch 18/25
198/198 [=====] - 7s 37ms/step - loss: 0.0493 - ac
curacy: 0.9832 - val_loss: 0.4553 - val_accuracy: 0.9333
Epoch 19/25
198/198 [=====] - 8s 42ms/step - loss: 0.0419 - ac
curacy: 0.9832 - val_loss: 0.3088 - val_accuracy: 0.9000
Epoch 20/25
198/198 [=====] - 14s 71ms/step - loss: 0.0206 - a
ccuracy: 0.9933 - val_loss: 0.2250 - val_accuracy: 0.9333
Epoch 21/25
198/198 [=====] - 13s 68ms/step - loss: 0.0200 - a
ccuracy: 0.9983 - val_loss: 0.2326 - val_accuracy: 0.9667
Epoch 22/25
198/198 [=====] - 14s 69ms/step - loss: 0.0590 - a
ccuracy: 0.9815 - val_loss: 0.2914 - val_accuracy: 0.9667
Epoch 23/25
198/198 [=====] - 14s 71ms/step - loss: 0.0365 - a
ccuracy: 0.9916 - val_loss: 0.2424 - val_accuracy: 0.9667
Epoch 24/25
198/198 [=====] - 9s 46ms/step - loss: 0.0299 - ac
curacy: 0.9882 - val_loss: 0.2493 - val_accuracy: 0.9667
Epoch 25/25
198/198 [=====] - 8s 38ms/step - loss: 0.1112 - ac
curacy: 0.9697 - val_loss: 0.2215 - val_accuracy: 0.9667

```

Out[13]:

In [71]:

```
model.save('Gesture.h5')
```

In [70]:

```

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

```

```
json_file.write(model_json)
```