MODEL BULDING

Step 1: Importing The Model Building Libraries

Importing libraries

```
#This library helps add support for large, multi-dimensional arrays and matrices
import numpy as np
#open source used for both ML and DL for computation
import tensorflow as tf
#it is a plain stack of layers
from tensorflow.keras.models import Sequential
#Dense layer is the regular deeply connected neural network layer
from tensorflow.keras.layers import Dense,Flatten, Dropout
#Faltten-used fot flattening the input or change the dimension, MaxPooling2D-for downsampling the image for Convolutional layer
from tensorflow.keras.layers import Convolution2D,MaxPooling2D
#Its used for different augmentation of the image
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

Step 2: Initializing The Model

Model Creation

```
#Initializing the CNN
model = Sequential()
```

Step 3: Adding Dense Layers

```
#Adding a fully connected layer, i.e. Hidden Layer
model.add(Dense(units=512 , activation='relu'))
```

```
# softmax for categorical analysis, Output Layer
model.add(Dense(units=6, activation='softmax'))
```

```
model.summary()#summary of our model
```

```
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
_________________________________________________________________
```

Step 4: Configure The Learning Process

Model Compilation

```
Compiling the CNN
# categorical_crossentropy for more than 2
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Step 5: Train The Model

Model fitting

```
It will generate packets of train and test data for training
model.fit_generator(x_train,
                    steps_per_epoch = 594/3 ,
                    epochs = 25,
                    validation_data = x_test,
                    validation_steps = 30/3 )
```

Step 6: Save The Model

Saving model

```
Save the model
model.save('Tested_gesture.h5')
```

Step 7: Test The Model

Importing Libraries

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
#loading the model for testing
model = load_model("D:\Python\gesture based control of raidiology images\Flask\Tested_gesture.h5")
path = "D:\\Python\\gesture based control of raidiology images\\Dataset\\test\\1\\1.jpg"
```

```
#loading of the image
img = image.load_img(path,
                     color_mode='grayscale',
                     target_size= (64,64))
x = image.img_to_array(img)
mage to array
x.shape
```

```
dex=['0','1','2','3','4','5']
result=str(index[pred[0]])
result
```