**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

InÂ [2]:

train\_datagen **=** ImageDataGenerator(rescale**=**1.**/**255,

shear\_range**=**0.2,

zoom\_range**=**0.2,

horizontal\_flip**=True**)

test\_datagen**=**ImageDataGenerator(rescale**=**1.**/**255)

InÂ [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.

InÂ [5]:

model **=** Sequential()

InÂ [6]:

model**.**add(Convolution2D(32, (3, 3), input\_shape**=**(64, 64, 1), activation**=**'relu'))

model**.**add(MaxPooling2D(pool\_size**=**(2, 2)))

InÂ [7]:

model**.**add(Convolution2D(32, (3, 3), activation**=**'relu'))

model**.**add(MaxPooling2D(pool\_size**=**(2,2)))

InÂ [8]:

model**.**add(Flatten())

InÂ [9]:

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

InÂ [10]:

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

InÂ [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

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

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

InÂ [12]:

model**.**compile(optimizer**=**'adam', loss**=**'categorical\_crossentropy', metrics**=**['accuracy'])

InÂ [13]:

model**.**fit\_generator(x\_train,

steps\_per\_epoch **=** 594**/**3 ,

epochs **=** 25,

validation\_data **=** x\_test,

validation\_steps **=** 30**/**3 )

<ipython-input-13-b82b2aea274a>: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 - accuracy: 0.4428 - val\_loss: 0.7625 - val\_accuracy: 0.7333

Epoch 2/25

198/198 [==============================] - 14s 70ms/step - loss: 0.6228 - accuracy: 0.7374 - val\_loss: 0.4547 - val\_accuracy: 0.9000

Epoch 3/25

198/198 [==============================] - 14s 69ms/step - loss: 0.4162 - accuracy: 0.8350 - val\_loss: 0.2619 - val\_accuracy: 0.9000

Epoch 4/25

198/198 [==============================] - 13s 68ms/step - loss: 0.3183 - accuracy: 0.8771 - val\_loss: 0.2557 - val\_accuracy: 0.9000

Epoch 5/25

198/198 [==============================] - 13s 67ms/step - loss: 0.2563 - accuracy: 0.9024 - val\_loss: 0.2768 - val\_accuracy: 0.9000

Epoch 6/25

198/198 [==============================] - 14s 69ms/step - loss: 0.1513 - accuracy: 0.9495 - val\_loss: 0.3705 - val\_accuracy: 0.9333

Epoch 7/25

198/198 [==============================] - 14s 71ms/step - loss: 0.1610 - accuracy: 0.9360 - val\_loss: 0.3457 - val\_accuracy: 0.8667

Epoch 8/25

198/198 [==============================] - 14s 70ms/step - loss: 0.1166 - accuracy: 0.9545 - val\_loss: 0.3367 - val\_accuracy: 0.9000

Epoch 9/25

198/198 [==============================] - 14s 69ms/step - loss: 0.1109 - accuracy: 0.9646 - val\_loss: 0.2659 - val\_accuracy: 0.9000

Epoch 10/25

198/198 [==============================] - 14s 68ms/step - loss: 0.0823 - accuracy: 0.9663 - val\_loss: 0.5025 - val\_accuracy: 0.9333

Epoch 11/25

198/198 [==============================] - 13s 67ms/step - loss: 0.0799 - accuracy: 0.9747 - val\_loss: 0.2626 - val\_accuracy: 0.9333

Epoch 12/25

198/198 [==============================] - 9s 45ms/step - loss: 0.0919 - accuracy: 0.9714 - val\_loss: 0.4762 - val\_accuracy: 0.8333

Epoch 13/25

198/198 [==============================] - 14s 70ms/step - loss: 0.0357 - accuracy: 0.9899 - val\_loss: 0.2444 - val\_accuracy: 0.9000

Epoch 14/25

198/198 [==============================] - 13s 65ms/step - loss: 0.0341 - accuracy: 0.9899 - val\_loss: 0.2582 - val\_accuracy: 0.9333

Epoch 15/25

198/198 [==============================] - 9s 44ms/step - loss: 0.0964 - accuracy: 0.9646 - val\_loss: 0.1478 - val\_accuracy: 0.9333

Epoch 16/25

198/198 [==============================] - 8s 43ms/step - loss: 0.0482 - accuracy: 0.9832 - val\_loss: 0.1486 - val\_accuracy: 0.9667

Epoch 17/25

198/198 [==============================] - 7s 37ms/step - loss: 0.0564 - accuracy: 0.9815 - val\_loss: 0.2217 - val\_accuracy: 0.9333

Epoch 18/25

198/198 [==============================] - 7s 37ms/step - loss: 0.0493 - accuracy: 0.9832 - val\_loss: 0.4553 - val\_accuracy: 0.9333

Epoch 19/25

198/198 [==============================] - 8s 42ms/step - loss: 0.0419 - accuracy: 0.9832 - val\_loss: 0.3088 - val\_accuracy: 0.9000

Epoch 20/25

198/198 [==============================] - 14s 71ms/step - loss: 0.0206 - accuracy: 0.9933 - val\_loss: 0.2250 - val\_accuracy: 0.9333

Epoch 21/25

198/198 [==============================] - 13s 68ms/step - loss: 0.0200 - accuracy: 0.9983 - val\_loss: 0.2326 - val\_accuracy: 0.9667

Epoch 22/25

198/198 [==============================] - 14s 69ms/step - loss: 0.0590 - accuracy: 0.9815 - val\_loss: 0.2914 - val\_accuracy: 0.9667

Epoch 23/25

198/198 [==============================] - 14s 71ms/step - loss: 0.0365 - accuracy: 0.9916 - val\_loss: 0.2424 - val\_accuracy: 0.9667

Epoch 24/25

198/198 [==============================] - 9s 46ms/step - loss: 0.0299 - accuracy: 0.9882 - val\_loss: 0.2493 - val\_accuracy: 0.9667

Epoch 25/25

198/198 [==============================] - 8s 38ms/step - loss: 0.1112 - accuracy: 0.9697 - val\_loss: 0.2215 - val\_accuracy: 0.9667

Out[13]:

<keras.callbacks.History at 0x2a66cdbfeb0>

InÂ [71]:

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

InÂ [70]:

model\_json **=** model**.**to\_json()

**with** open("model-bw.json", "w") **as** json\_file:

json\_file**.**write(model\_json)