

AI-based localization and classification of skin disease with erythema

ASSIGNMENT-3

NAME : KARTHIKEYAN.P

REGISTER NUMBER : 412519106060

BUILD CNN MODEL FOR CLASSIFICATION OF FLOWERS

ls

Volume in drive C is OS

Volume Serial Number is 08F5-B770

Directory of C:\Users\aruna\flowers

06-10-2022	19:23	.
06-10-2022	21:40	..
06-10-2022	18:10	daisy
06-10-2022	18:11	dandelion
06-10-2022	19:23	111,871,896 flower1.h5
06-10-2022	18:11	rose
06-10-2022	18:11	sunflower
06-10-2022	18:12	tulip
1 File(s)		111,871,896 bytes
7 Dir(s)		839,051,767,808 bytes free

IMAGE AUGMENTATION

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

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

```
test_datagen=ImageDataGenerator(rescale=1./255)
```

```
x_train=train_datagen.flow_from_directory(r"C:\Users\aruna\flowers",target_size=(64,64),
```

```
class_mode="categorical",batch_size=24)
```

Found 4317 images belonging to 5 classes.

```
x_test=test_datagen.flow_from_directory(r"C:\Users\aruna\flowers",target_size=(64,64),
```

```
class_mode="categorical",batch_size=24)
```

Found 4317 images belonging to 5 classes.

CREATE MODEL: Add Layers (Convolution,MaxPooling,Flatten,Dense-(Hidden Layers),Output)

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

```
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
```

```
model=Sequential()
```

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

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

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

```
model.add(Dense(300,activation='relu'))
```

```
model.add(Dense(300,activation='relu'))
```

```
model.add(Dense(5,activation="softmax"))
```

COMPILE THE MODEL

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

```
len(x_train)
```

```
180
```

```
FIT THE MODEL
```

```
model.fit(x_train,epochs=5,validation_data=x_test,steps_per_epoch=len(x_train),validation_steps=len(x_test))
```

```
Epoch 1/5
```

```
180/180 [=====] - 250s 1s/step - loss: 1.2670 - accuracy: 0.4772 - val_loss: 1.1097 - val_accuracy: 0.5552
```

```
Epoch 2/5
```

```
180/180 [=====] - 34s 187ms/step - loss: 1.0532 - accuracy: 0.5770 - val_loss: 0.9905 - val_accuracy: 0.6034
```

```
Epoch 3/5
```

```
180/180 [=====] - 31s 171ms/step - loss: 0.9811 - accuracy: 0.6125 - val_loss: 0.9617 - val_accuracy: 0.6308
```

```
Epoch 4/5
```

```
180/180 [=====] - 30s 166ms/step - loss: 0.9225 - accuracy: 0.6370 - val_loss: 0.8994 - val_accuracy: 0.6535
```

```
Epoch 5/5
```

```
180/180 [=====] - 30s 169ms/step - loss: 0.8758 - accuracy: 0.6630 - val_loss: 0.9469 - val_accuracy: 0.6389
```

```
SAVE THE MODEL
```

```
model.save("flower1.h5")
```

```
TEST THE MODEL
```

```
from tensorflow.keras.models import load_model
```

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

```

import numpy as np

model=load_model("C:/Users/aruna/flowers/flower1.h5")

img=image.load_img("C:/Users/aruna/flowers/tulip/11746276_de3dec8201.jpg",target_size=(64,64))

img

x=image.img_to_array(img)

x

array([[202.,  94., 206.],
       [203., 136., 207.],
       [ 73.,  13.,  39.],
       ...,
       [164.,  42., 129.],
       [128.,  42., 113.],
       [149.,  42., 112.]],

      [[148.,  41., 111.],
       [168.,  37., 141.],
       [ 41.,  10.,  28.],
       ...,
       [126.,  15.,  81.],
       [112.,  24.,  74.],
       [101.,  16.,  57.]],

      [[ 81.,   0.,  36.],
       [215., 110., 203.],
       [182.,  67., 220.],

```

...,

[127., 48., 103.],

[85., 19., 47.],

[77., 9., 30.]],

...,

[[6., 38., 0.],

[39., 93., 33.],

[50., 3., 19.],

...,

[33., 20., 37.],

[42., 138., 12.],

[61., 158., 45.]],

[[5., 44., 0.],

[11., 55., 4.],

[39., 117., 42.],

...,

[70., 192., 21.],

[210., 57., 171.],

[31., 147., 14.]],

[[25., 74., 27.],

[7., 71., 11.],

[40., 108., 47.],

```
...,
[173.,  44., 137.],
[136.,  24., 100.],
[155.,   0., 115.]], dtype=float32)
```

```
x.ndim
```

```
3
```

```
x=np.expand_dims(x,axis=0)
```

```
x.ndim
```

```
4
```

```
pred=model.predict(x)
```

```
1/1 [=====] - 0s 286ms/step
```

```
pred
```

```
array([[0.000000e+00, 0.000000e+00, 6.961726e-27, 0.000000e+00,
        1.000000e+00]], dtype=float32)
```

```
labels=["daisy","dandelion","rose","sunflower","tulip"]
```

```
np.argmax(pred)
```

```
4
```

```
labels[3]
```

```
'sunflower'
```

```
labels[np.argmax(pred)]
```

```
'tulip'
```