Name :Sowmya J

Register Number -711519bec309

Team Id -PNT2022TMID07418

# ImportandUnziptheDataset



↳*2cellshidden*

ImageAugmentation

fromtensorflow.keras.preprocessing.imageimportImageDataGenerator

#Data augmentation on training variable

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

zoom\_range=0.2,

horizontal\_flip=True)

#Data augmentation on testing variable

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

#Data augmentation on training data

xtrain=train\_datagen.flow\_from\_directory('/content/flowers',

target\_size=(64,64),

class\_mode='categorical',batch\_size=100)

Found 3384 images belonging to 5 classes.

# ImportLayers

fromtensorflow.keras.modelsimportSequential

fromtensorflow.keras.layersimportConvolution2D,MaxPooling2D,Flatten,Dense

# AddCNNLayers

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(150,activation='relu'))model.add(Dense(5,activation='softmax'))

# CompiletheModel

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

model.fit(xtrain,steps\_per\_epoch=len(xtrain),epochs=10)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Epoch  34/34 | 1/10  [==============================] | - 23s | 646ms/step | - loss: | 1.7948 | - accuracy: | 0 |
| Epoch | 2/10 |  |  |  |  |  |  |
| 34/34  Epoch34/34 | [==============================]3/10  [==============================] | - 22s  - 22s | 644ms/step  641ms/step | * loss: * loss: | 1.1695  1.0633 | * accuracy: * accuracy: | 0  0 |
| Epoch  34/34 | 4/10  [==============================] | - 26s | 759ms/step | - loss: | 1.0108 | - accuracy: | 0 |
| Epoch | 5/10 |  |  |  |  |  |  |
| 34/34  Epoch | [==============================]6/10 | - 22s | 637ms/step | - loss: | 0.9663 | - accuracy: | 0 |
| 34/34  Epoch34/34 | [==============================]7/10  [==============================] | - 22s  - 22s | 642ms/step  637ms/step | * loss: * loss: | 0.8937  0.8554 | * accuracy: * accuracy: | 0  0 |
| Epoch  34/34 | 8/10  [==============================] | - 22s | 644ms/step | - loss: | 0.8235 | - accuracy: | 0 |
| Epoch | 9/10 |  |  |  |  |  |  |
| 34/34  Epoch34/34 | [==============================]10/10  [==============================] | - 22s  - 22s | 643ms/step  639ms/step | * loss: * loss: | 0.7987  0.7775 | * accuracy: * accuracy: | 0  0 |

<keras.callbacks.History at 0x7fdc3ba7b090>

# SaveModel

model.save('flowers.h5')

# TestingModel

importnumpyas np

fromtensorflow.keras.preprocessing importimage

# Testdata:Daisy

img = image.load\_img('/content/flowers/rose/10090824183\_d02c613f10\_m.jpg',target\_size=(64,img



x = image.img\_to\_array(img)x

|  |  |  |  |
| --- | --- | --- | --- |
| array([[[14., | 22., | 7.], |  |
| [11., | 22., | 6.], |  |
| [ 8.,  ...,  [32., | 19.,  47., | 3.],  24.], |  |
| [30., | 48., | 22.], |  |
| [33., | 49., | 23.]], |  |
| [[13., | 20., | 12.], |  |
| [11., | 21., | 10.], |  |
| [11.,  ...,  [37., | 22.,  51., | 8.],  26.], |  |
| [35., | 49., | 26.], |  |
| [25., | 45., | 20.]], |  |
| [[19., | 30., | 16.], |  |
| [19., | 31., | 17.], |  |
| [16., | 29., | 12.], |  |
| ...,  [31., | 47., | 20.], |  |
| [28., | 49., | 18.], |  |
| [27., | 43., | 17.]], |  |
| ..., |  |  |  |
| [[15., | 17., | 6.], |  |
| [ 2., | 9., | 2.], |  |
| [ 2.,  ..., | 9., | 1.], |  |
| [ 8., | 21., | 11.], |  |
| [ 2., | 12., | 3.], |  |
| [ 9., | 16., | 9.]], |  |
| [[12., | 20., | 9.], |  |
| [ 1., | 8., | 1.], |  |
| [ 5.,  ..., | 10., | 3.], |  |
| [ 3., | 8., | 2.], |  |
| [ 6., | 16., | 5.], |  |
| [ 5., | 7., | 4.]], |  |
| [[24., | 27., | 18.], |  |
| [11., | 21., | 13.], |  |
| [ 8.,  ...,  [ 1., | 13.,  6., | 6.],  0.], |  |
| [ 2., | 9., | 1.], |  |
| [ 2., | 9., | 1.]]], | dtype=float32) |

x=np.expand\_dims(x,axis=0)

x

model.predict(x)

array([[1., 0., 0., 0., 0.]], dtype=float32)

xtrain.class\_indices

{'daisy': 0, 'dandelion': 1, 'rose': 2, 'sunflower': 3, 'tulip': 4}

outp = ['daisy','dandelion','rose','sunflower','tulip']pred= np.argmax(model.predict(x))

outp[pred]

'daisy'

# Testdata:Rose

img = image.load\_img('/content/flowers/rose/102501987\_3cdb8e5394\_n.jpg',target\_size=(64,64img



x=image.img\_to\_array(img)x=np.expand\_dims(x,axis=0)

pred = np.argmax(model.predict(x))outp[pred]

'rose'

# Testdata:Sunflower

img = image.load\_img('/content/flowers/sunflower/1022552036\_67d33d5bd8\_n.jpg',target\_size=img



x=image.img\_to\_array(img)x=np.expand\_dims(x,axis=0)

pred = np.argmax(model.predict(x))outp[pred]

'sunflower'

# Testdata:Tulip

img = image.load\_img('/content/flowers/tulip/10128546863\_8de70c610d.jpg',target\_size=(64,6img



x=image.img\_to\_array(img)x=np.expand\_dims(x,axis=0)

pred = np.argmax(model.predict(x))outp[pred]

'tulip'

# Testdata:Dandelion

img = image.load\_img('/content/flowers/dandelion/11405573\_24a8a838cc\_n.jpg',target\_size=(6img



x=image.img\_to\_array(img)x=np.expand\_dims(x,axis=0)

pred = np.argmax(model.predict(x))outp[pred]

'dandelion'

[Colabpaidproducts](https://colab.research.google.com/signup?utm_source=footer&utm_medium=link&utm_campaign=footer_links)-[Cancelcontractshere](https://colab.research.google.com/cancel-subscription)0s completedat10:46PM