Assignment- 3

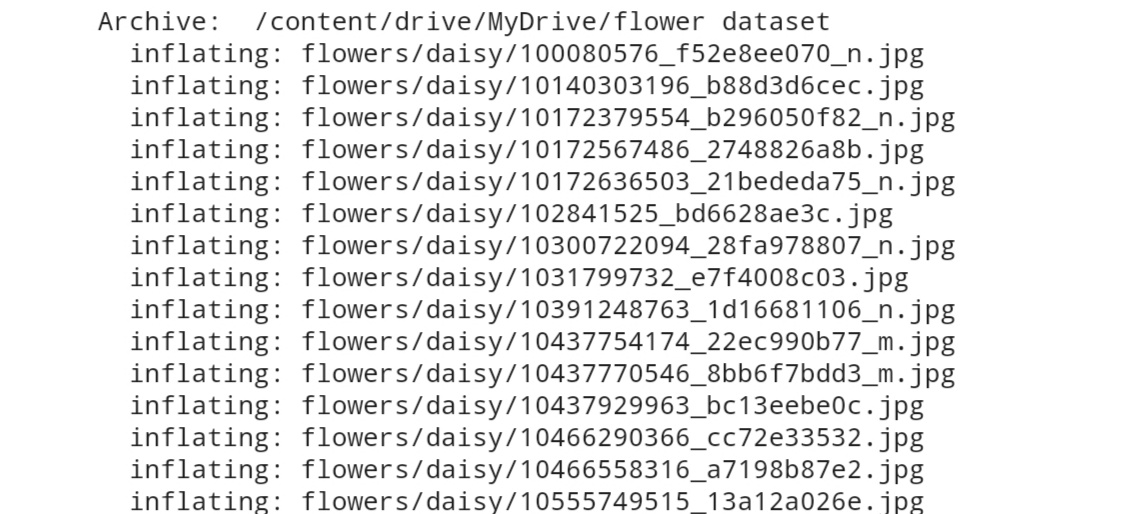
|  |  |
| --- | --- |
| Student name | Yuvaraj .S |
| Roll no/Reg no | 411419104040 |
| College Name | New Prince Shri Bhavani Clg Of Engg & Tech |
| Assesment No | 03 |

# ****Tasks****

## ****1.Downloading and unzipping dataset****

Solution:

!unzip '/content/drive/MyDrive/flower dataset'









## ****2. Image Augmentation****

Solution:

**import** numpy **as** np

**import** tensorflow **as** tf

**from** tensorflow.keras **import** layers

**from** tensorflow.keras.models **import** Sequential

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

**import** matplotlib.pyplot **as** plt

batch\_size **=** 32

img\_height **=** 180

img\_width **=** 180

data\_dir **=** "/content/flowers"

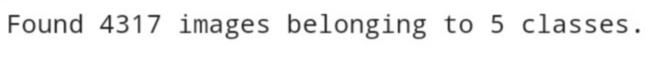
train\_datagen **=** ImageDataGenerator(rescale **=** 1.**/**255, horizontal\_flip **=** **True**, vertical\_flip **=** **True**, zoom\_range **=** 0.2)

x\_train **=** train\_datagen**.**flow\_from\_directory('/content/flowers',

target\_size**=**(64,64),

class\_mode**=**'categorical',

batch\_size**=**100)



data\_augmentation **=** Sequential(

[

layers**.**RandomFlip("vertical",input\_shape**=**(img\_height, img\_width, 3)),

layers**.**RandomRotation(0.1),

layers**.**RandomZoom(0.1),

]

)

## ****3. Creating Model****

Solution:

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

model **=** Sequential()

training\_ds **=** tf**.**keras**.**utils**.**image\_dataset\_from\_directory(

data\_dir,

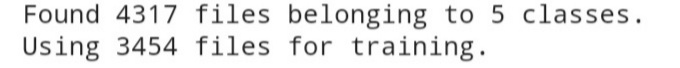
validation\_split**=**0.2,

subset**=**"training",

seed**=**57,

image\_size**=**(img\_height, img\_width),

batch\_size**=**batch\_size)



validation\_ds **=** tf**.**keras**.**utils**.**image\_dataset\_from\_directory(

data\_dir,

validation\_split**=**0.2,

subset**=**"validation",

seed**=**107,

image\_size**=**(img\_height, img\_width),

batch\_size**=**batch\_size

training\_ds**.**class\_names

Output:



plt**.**figure(figsize**=**(7, 7))

**for** data, labels **in** training\_ds**.**take(1):

**for** i **in** range(6):

ax **=** plt**.**subplot(2, 3, i **+** 1)

plt**.**imshow(data[i]**.**numpy()**.**astype("uint8"))

plt**.**title(training\_ds**.**class\_names[labels[i]])

plt**.**axis("off")





## ****4.Add Layers****

### ****4a. Convolution layer****

Solution:

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

### ****4b. Maxpooling layer****

Solution:

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

### ****4c. Flatten****

Solution:

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

### ****4d. Hidden/dense layers****

Solution:

model**.**add(Dense(300, activation **=** "relu"))

model**.**add(Dense(150, activation **=** "relu"))

### ****4e. Output layer****

### **Solution:**

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

## ****5. Compiling Model****

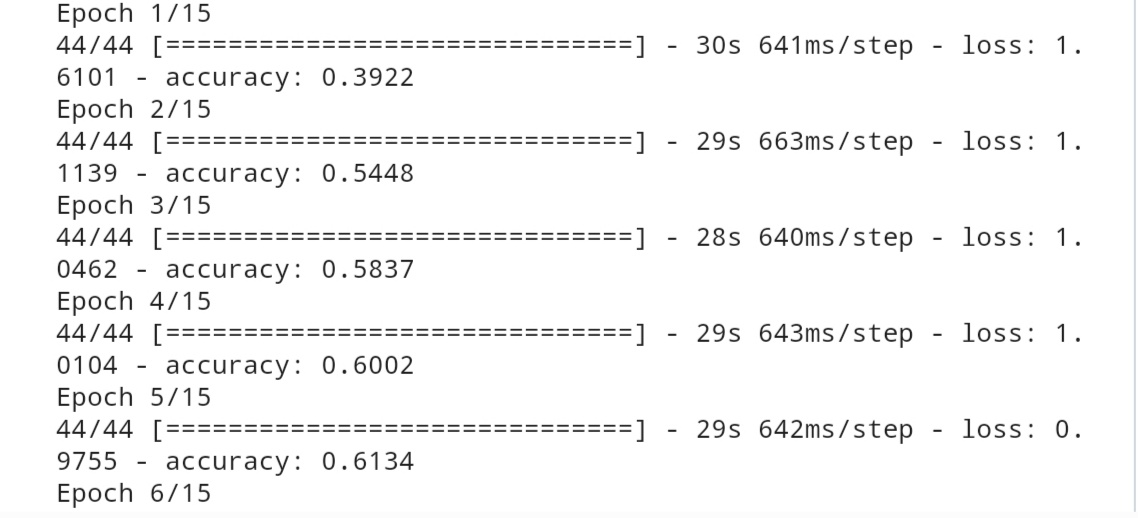
Solution:

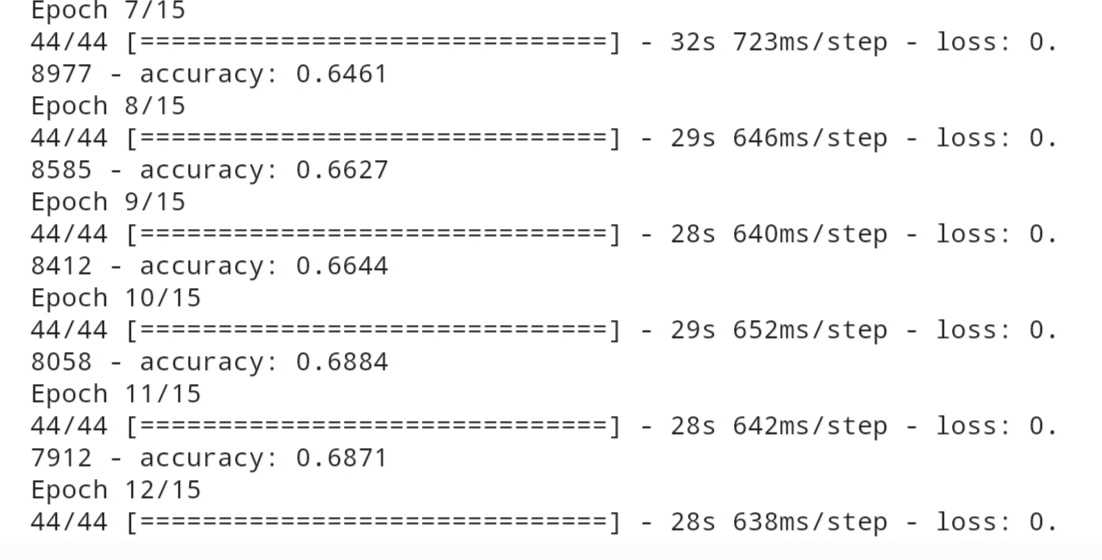
model**.**compile(optimizer**=**'adam',loss**=**'categorical\_crossentropy',metrics**=**['accuracy'])

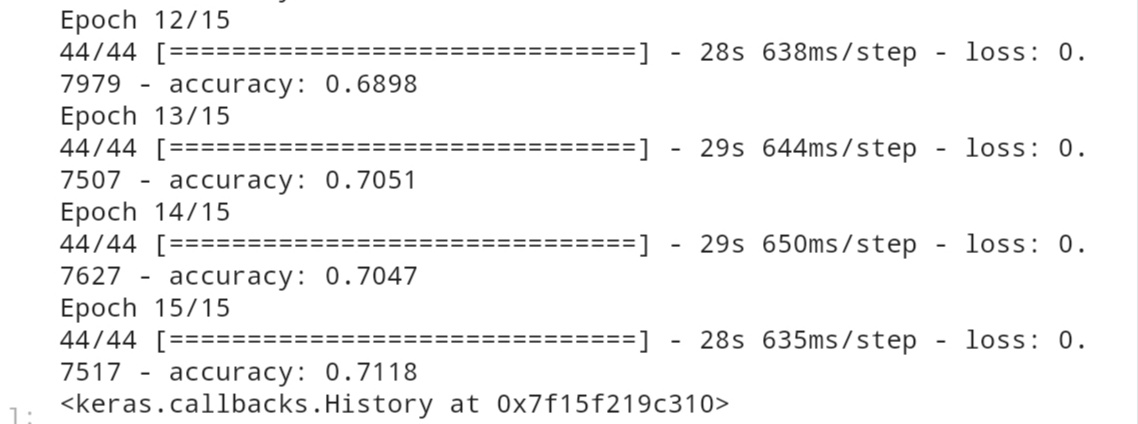
## ****6. Fit The Model****

Solution:

model**.**fit(x\_train, epochs **=** 15, steps\_per\_epoch **=** len(x\_train))







## ****7. Save The Model****

Solution:

model**.**save("flowers.h1")

## ****8. Test The Model****

Solution:

**from** tensorflow.keras.models **import** load\_model

**from** tensorflow.keras.preprocessing **import** image

model **=** load\_model("/content/flowers.h1")

daisy\_img **=** image**.**load\_img('/content/flowers/daisy/100080576\_f52e8ee070\_n.jpg',target\_size**=**(64,64))

x **=** image**.**img\_to\_array(daisy\_img)

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

predicted\_class**=**model**.**predict(x)

labels **=** ['daisy','dandelion','roses','sunflowers','tulips']

labels[np**.**argmax(predicted\_class)]

Output:



Solution:

daisy\_img

Output:

