

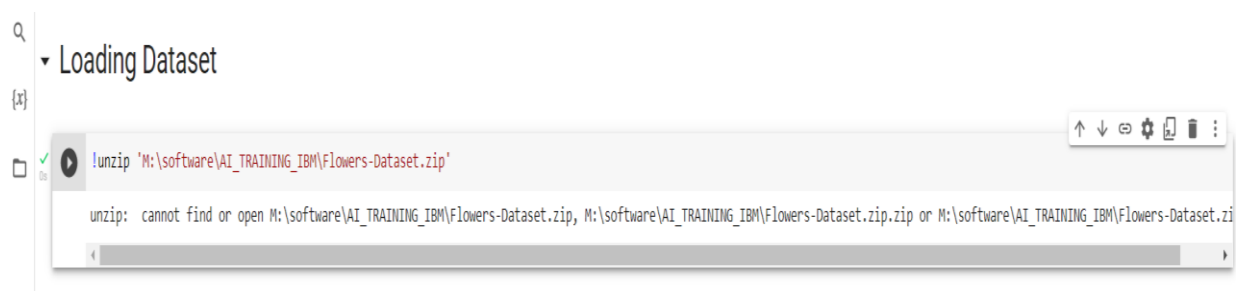
ASSIGNMENT – 3

Assignment Date	06- October- 2022
Student's Name	KAVIYA A
Student's Roll No.	510919106701
Maximum marks	2 marks

PROBLEM STATEMENT: Build CNN Model for Classification of Flowers.

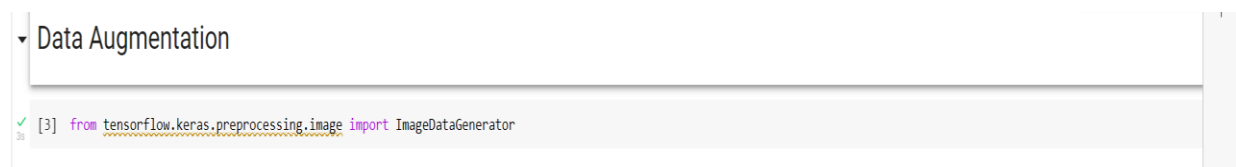
QUESTION – 1:

DOWNLOAD THE DATASET



QUESTION – 2:

DATA/ IMAGE AUGMENTATION



QUESTION – 3:

TRAINING & TESTING

▾ Training and Testing

✓
1s



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

In [9]:

```
xtrain=train_datagen.flow_from_directory('M:\\software\\AI_TRAINING_IBM\\flowers', class_mode='categorical', target_size=(64,64), batch_size=100)
```

Found 4317 images belonging to 5 classes.

In [10]:

```
xtest=test_datagen.flow_from_directory('M:\\software\\AI_TRAINING_IBM\\flowers', class_mode='categorical', target_size=(64,64), batch_size=100)
```

Found 4317 images belonging to 5 classes.

QUESTION – 4 &

QUESTION -5:

CREATE MODEL:

ADD LAYERS

▾ Importing the models and the layers

✓
0s

```
[7] from tensorflow.keras.models import Sequential
    from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten, Dense
```

✓
1s




```
model=Sequential()
model.add(Convolution2D(64,(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'))
```

QUESTION – 6:

COMPILE THE MODEL:

▼ Compile

0s  `model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])`

QUESTION – 7:

FIT THE MODEL:

Fit the model.

```
model.fit_generator(xtrain,
                    steps_per_epoch=len(xtrain),
                    epochs=20,
                    validation_data=xtest,
                    validation_steps=len(xtest))
```

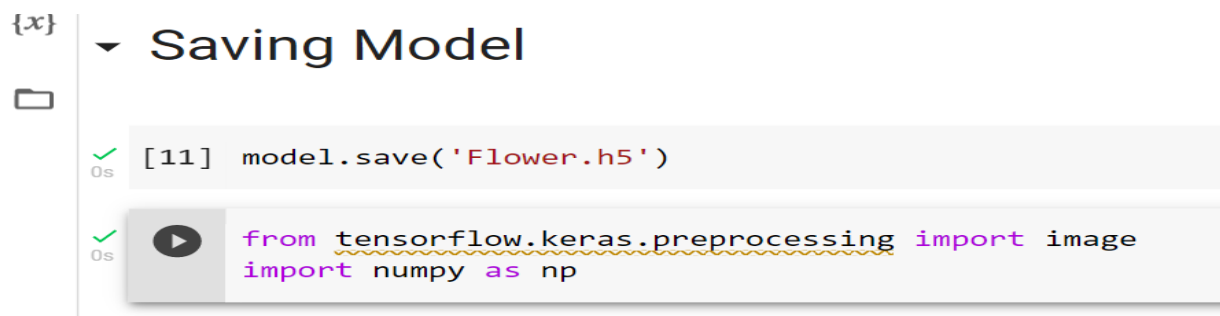
C:\Users\mm\AppData\Local\Temp\ipykernel_6696\312721451.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
model.fit_generator(xtrain,

```
Epoch 1/20
44/44 [=====] - 37s 835ms/step - loss: 1.9038 - ac
curacy: 0.3836 - Val loss: 1.1672 - Val accuracy: 0.5219
Epoch 2/20
44/44 [=====] - 34s 779ms/step - loss: 1.0908 - ac
curacy: 0.5606 - Val loss: 1.0398 - Val accuracy: 0.5965
Epoch 3/20
44/44 [=====] - 36s 815ms/step - loss: 1.0262 - ac
curacy: 0.5925 - Val loss: 1.0038 - Val accuracy: 0.6185
Epoch 4/20
44/44 [=====] - 36s 823ms/step - loss: 0.9335 - ac
curacy: 0.6410 - Val loss: 0.8923 - Val accuracy: 0.6560
Epoch 5/20
44/44 [=====] - 36s 809ms/step - loss: 0.8781 - ac
curacy: 0.6604 - Val loss: 0.8886 - Val accuracy: 0.6646
Epoch 6/20
44/44 [=====] - 34s 764ms/step - loss: 0.8512 - ac
curacy: 0.6713 - Val loss: 0.8784 - Val accuracy: 0.6771
Epoch 7/20
44/44 [=====] - 33s 758ms/step - loss: 0.7922 - ac
curacy: 0.6931 - Val loss: 0.7586 - Val accuracy: 0.7121
Epoch 8/20
44/44 [=====] - 35s 811ms/step - loss: 0.7471 - ac
curacy: 0.7107 - Val loss: 0.6955 - Val accuracy: 0.7262
Epoch 9/20
```

Out[19]:

QUESTION – 8:

SAVING THE MODEL



QUESTION – 9:

TEST THE MODEL

Testing the model

```
In [22]: img=image.load_img('M:\\software\\AI_TRAINING_IBM\\flowers\\sunflower\\6953297_8576bf4ea3.jpg',target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
prediction=np.argmax(model.predict(x))
op=['daisy','dandelion','rose','sunflower','tulip']
op[prediction]

1/1 [=====] - 0s 22ms/step
Out[22]: 'sunflower'
```

QUESTION -10:

TESTING THE MODEL

```
In [24]: img=image.load_img('M:\\software\\AI_TRAINING_IBM\\download.jpg',target_size=(64,64))#randomly downloaded testing
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
prediction=np.argmax(model.predict(x))
op=['daisy','dandelion','rose','sunflower','tulip']
op[prediction]

1/1 [=====] - 0s 22ms/step
Out[24]: 'sunflower'
```

In []: