

Assignment -3
Python Programming

Assignment Date	15 October 2022
Student Name	Vasu Deva Krishna Rayan K
Student Roll Number	953719104058
Maximum Marks	2 Marks

Question-1: Image

Augmentation

Solution:

1. Image Augmentation

```
In [ ]: from keras.utils import load_img, img_to_array
        from keras.preprocessing.image import ImageDataGenerator, image_utils
        from keras.models import Sequential
        from keras.layers import Convolution2D, MaxPooling2D, Flatten, Dense,
        import numpy as np
```

2.1 Augmenting the Train Variables

```
In [ ]: train_datagen = ImageDataGenerator(
        rotation_range = 40,
        shear_range = 0.2,
        zoom_range = 0.2,
        horizontal_flip = True,
        brightness_range = (0.5, 1.5)
    )
```

2.2 Augmenting the Test variables

```
In [ ]: test_datagen = ImageDataGenerator(rescale= 1./255)
```

```
In [ ]: ftrain = train_datagen.flow_from_directory(
        '../Datasets/flowers/train/',
        target_size = (64,64),
        class_mode = 'categorical',
        batch_size = 100
    )
```

Question-2: Create

Model Solution:

```
In [ ]: ftest = test_datagen.flow_from_directory(
        '../Datasets/flowers/test/',
        target_size = (64,64),
        class_mode = 'categorical',
        batch_size = 100
    )
```

1. Creating the Model

```
In [ ]: model = Sequential()
```

QUESTION -3

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

1. Adding Layers (Convolution Layers, MaxPooling, Flatten, Dense)

```
In [ ]: 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(Dropout(0.25))
        model.add(Dense(400, activation = 'relu'))
        model.add(Dropout(0.25))
        model.add(Dense(200, activation = 'relu'))
        model.add(Dropout(0.25))
        model.add(Dense(5, activation = 'softmax'))
```

QUESTION-4

Compile, Fit and save the model, SOLUTION:

1. Compiling the Model

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

1. Fitting the model

```
In [ ]: model.fit(
        ftrain,
        steps_per_epoch = len(ftrain),
        epochs = 10,
        validation_data = ftest,
        validation_steps = len(ftest)
    )
```

1. Saving the model

```
In [ ]: model.save('./flowers.h5')
```

QUESTION -5

Test the model **SOLUTION:**

1. Testing the model

8.1 Test 1

```
[ ]: img = image_utils.load_img(
    '../Datasets/flowers/test/daisy/1150395827_6f94a5c6e4_n.jpg',
    target_size = (64,64)
)
f = image_utils.img_to_array(img)
f = np.expand_dims(f, axis = 0)
pred = np.argmax(model.predict(f))
op = ['daisy', 'dandelion', 'rose', 'sunflower', 'tulip']
op[pred]
```

8.2 Test 2

```
[ ]: img = image_utils.load_img(
    '../Datasets/flowers/test/dandelion/33907694863_f7c0f23ef3_n.jpg',
    target_size = (64,64)
)
f = image_utils.img_to_array(img)
f = np.expand_dims(f, axis = 0)
pred = np.argmax(model.predict(f))
op = ['daisy', 'dandelion', 'rose', 'sunflower', 'tulip']
op[pred]
```

2020-07-15 10:45:45.100000000: 2020-07-15 10:45:45.100000000