Assignment -3

Build CNN Model for Classification of Flowers

Assignment Date	29 September 2022
Student Name	Ms. Ishwarya G K
Student Roll Number	910619104029
Maximum Marks	

IMAGE AUGMENTATION

```
In [1]: 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 = r"C:\Users\hp\tensorflow\Assignment 3\Dataset\Training"
```

```
In [2]: train_datagen = ImageDataGenerator(rescale = 1./255, horizontal_flip = True, vertical_flip = True, zoom_range = 0.2)
```

```
In [3]: x_train  = train_datagen.flow_from_directory(r'C:\Users\hp\tensorflow\Assignment  3\Dataset\Training',target_size=(64,64),class_mode='categorical',batch_size=100)
```

```
Found 4317 images belonging to 5 classes.
```

```
In [4]: data_augmentation = Sequential(
          [
            layers.RandomFlip("vertical",input_shape=(img_height, img_width, 3)),
            layers.RandomRotation(0.1),
            layers.RandomZoom(0.1),
          ]
        )
```

```
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting RngReadAndSkip cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting Bitcast cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting StatelessRandomUniformV2 cause there is no registered converter for this op.
WARNING:tensorflow:Using a while_loop for converting ImageProjectiveTransformV3 cause there is no registered converter for this op.
```

CREATING MODEL

```
In [5]: from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense
        model = Sequential()
```

```
In [6]: 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)
```

```
Found 4317 files belonging to 5 classes.
Using 3454 files for training.
```

```
In [7]: 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)
```

```
Found 4317 files belonging to 5 classes.
Using 863 files for validation.
```

```
In [8]: training_ds.class_names
```

```
Out[8]: ['daisy', 'dandelion', 'rose', 'sunflower', 'tulip']
```

```
In [9]: 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")
```

ADD LAYERS

CONVOLUTION LAYER

```
In [10]: model.add(Convolution2D(32, (3,3), activation = "relu", input_shape = (64,64,3) ))
```

MAXPOOLING LAYER

```
In [11]: model.add(MaxPooling2D(pool_size = (2,2)))
```

FLATTEN

```
In [12]: model.add(Flatten())
```

HIDDEN/DENSE LAYER

```
In [13]: model.add(Dense(300, activation = "relu"))
         model.add(Dense(150, activation = "relu"))
```

OUTPUT LAYER

```
In [14]: model.add(Dense(5, activation = "softmax"))
```

COMPILE THE MODEL

```
In [15]: model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

FIT THE MODEL

```
In [16]: model.fit(x_train, epochs = 15, steps_per_epoch = len(x_train))
```

```
Epoch 1/15
44/44 [==============================] - 255s 6s/step - loss: 1.5912 - accuracy: 0.3678
Epoch 2/15
44/44 [==============================] - 153s 3s/step - loss: 1.1347 - accuracy: 0.5383
Epoch 3/15
44/44 [==============================] - 109s 2s/step - loss: 1.0550 - accuracy: 0.5791
Epoch 4/15
44/44 [==============================] - 69s 2s/step - loss: 0.9785 - accuracy: 0.6220
Epoch 5/15
44/44 [==============================] - 107s 2s/step - loss: 0.9251 - accuracy: 0.6412
Epoch 6/15
44/44 [==============================] - 53s 1s/step - loss: 0.8780 - accuracy: 0.6581
Epoch 7/15
44/44 [==============================] - 48s 1s/step - loss: 0.8600 - accuracy: 0.6620
Epoch 8/15
44/44 [==============================] - 34s 776ms/step - loss: 0.8493 - accuracy: 0.6785
Epoch 9/15
44/44 [==============================] - 36s 810ms/step - loss: 0.8076 - accuracy: 0.6894
Epoch 10/15
44/44 [==============================] - 36s 809ms/step - loss: 0.7879 - accuracy: 0.7019
Epoch 11/15
44/44 [==============================] - 52s 1s/step - loss: 0.7657 - accuracy: 0.7086
Epoch 12/15
44/44 [==============================] - 36s 819ms/step - loss: 0.7310 - accuracy: 0.7206
Epoch 13/15
44/44 [==============================] - 31s 711ms/step - loss: 0.7410 - accuracy: 0.7135
Epoch 14/15
44/44 [==============================] - 31s 699ms/step - loss: 0.7024 - accuracy: 0.7315
Epoch 15/15
44/44 [==============================] - 31s 692ms/step - loss: 0.6874 - accuracy: 0.7417
```

```
Out[16]: <keras.callbacks.History at 0x25c6755ad70>
```

SAVE THE MODEL

```
In [17]: model.save("flower.h1")
```

```
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op while saving (showing 1 of 1). These functions will not be directly callable after loading.
INFO:tensorflow:Assets written to: flower.h1\assets
INFO:tensorflow:Assets written to: flower.h1\assets
```

TEST THE MODEL

```
In [18]: from tensorflow.keras.models import load_model
         from tensorflow.keras.preprocessing import image
```

```
In [19]: tulip_img = image.load_img(r'C:\Users\hp\tensorflow\Assignment 3\Dataset\Training\tulip\112428919_f0c5ad7d9d_n.jpg',target_size=(64,64))
         x = image.img_to_array(tulip_img)
         x = np.expand_dims(x,axis=0)
         predicted_class=model.predict(x)

         1/1 [==============================] - 4s 4s/step
```

```
In [20]: labels = ['daisy','dandelion','roses','sunflowers','tulips']
         labels[np.argmax(predicted_class)]

Out[20]: 'tulips'
```

```
In [21]: tulip_img

Out[21]:
```