```
import os
import cv2
import numpy as np
```

Reading Input Images

```
images = []
labels = []
trial = []
```

Image Augmentation

Daisy Inputs

```
for img in os.listdir('/home/wintermute/Downloads/flowers/daisy/') :
    label = 0
    path = os.path.join('/home/wintermute/Downloads/flowers/daisy/',
img)
    img = cv2.imread(path, cv2.IMREAD_COLOR)
    img = cv2.resize(img, (150, 150))
    aug_img = cv2.flip(img,1)
    images.append(np.array(img))
    images.append(np.array(aug_img))
    labels.append(str(label))
    labels.append(str(label))
```

Dandelion Inputs

```
for img in os.listdir('/home/wintermute/Downloads/flowers/dandelion/')
:
    label = 1
    path =
os.path.join('/home/wintermute/Downloads/flowers/dandelion/', img)
    img = cv2.imread(path, cv2.IMREAD_COLOR)
    img = cv2.resize(img, (150, 150))
    aug_img = cv2.flip(img,1)
    images.append(np.array(img))
    images.append(np.array(aug_img))
    labels.append(str(label))
    labels.append(str(label))
```

Rose Inputs

```
for img in os.listdir('/home/wintermute/Downloads/flowers/rose/') :
    label = 2
    path = os.path.join('/home/wintermute/Downloads/flowers/rose/',
img)
    img = cv2.imread(path, cv2.IMREAD_COLOR)
    img = cv2.resize(img, (150, 150))
    aug_img = cv2.flip(img,1)
    images.append(np.array(img))
    images.append(np.array(aug_img))
```

```
    labels.append(str(label))
    labels.append(str(label))
```

Sunflower Inputs

```
for img in os.listdir('/home/wintermute/Downloads/flowers/sunflower/')
:
    label = 3
    path =
os.path.join('/home/wintermute/Downloads/flowers/sunflower/', img)
    img = cv2.imread(path, cv2.IMREAD_COLOR)
    img = cv2.resize(img, (150, 150))
    aug_img = cv2.flip(img,1)
    images.append(np.array(img))
    images.append(np.array(aug_img))
    labels.append(str(label))
    labels.append(str(label))
```

Tulip Inputs

```
for img in os.listdir('/home/wintermute/Downloads/flowers/tulip/') :
    label = 4
    path = os.path.join('/home/wintermute/Downloads/flowers/tulip/',
img)
    img = cv2.imread(path, cv2.IMREAD_COLOR)
    img = cv2.resize(img, (150, 150))
    aug_img = cv2.flip(img,1)
    images.append(np.array(img))
    images.append(np.array(aug_img))
    labels.append(str(label))
    labels.append(str(label))

print(len(images))
print(len(labels))
```

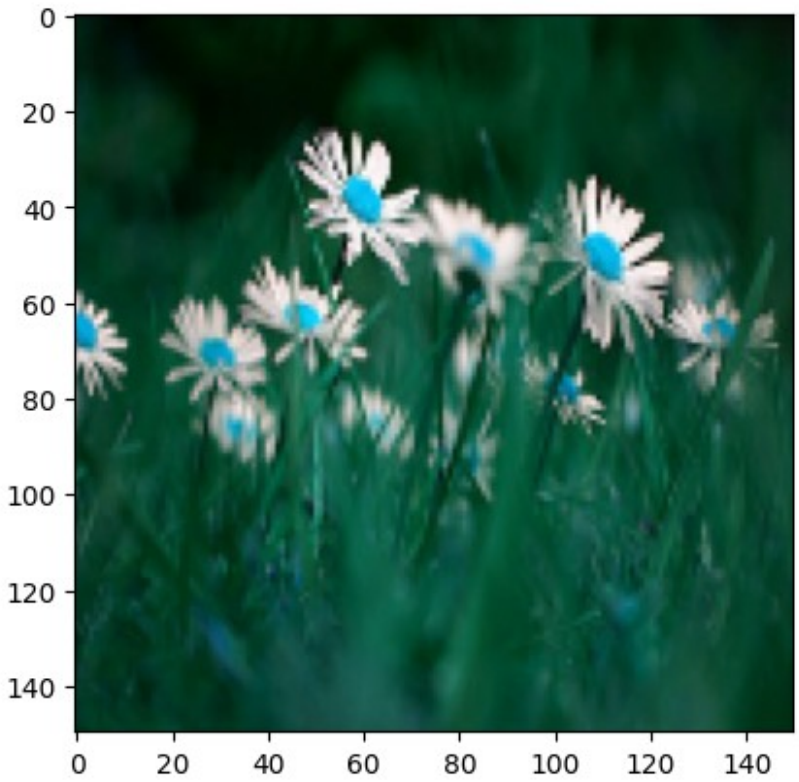
```
8634
8634
```

```
images = np.array(images)
```

Visualizing Image

```
import matplotlib.pyplot as plt

plt.imshow(images[6])
plt.show()
```



Data Preparation

```
from sklearn.model_selection import *
import keras
```

```
2022-10-03 12:57:10.016905: I
tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow
binary is optimized with oneAPI Deep Neural Network Library (oneDNN)
to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the
appropriate compiler flags.
2022-10-03 12:57:11.122756: E
tensorflow/stream_executor/cuda/cuda_blas.cc:2981] Unable to register
cuBLAS factory: Attempting to register factory for plugin cuBLAS when
one has already been registered
2022-10-03 12:57:14.551792: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could
not load dynamic library 'libnvinfer.so.7'; dlerror: libnvinfer.so.7:
cannot open shared object file: No such file or directory;
LD_LIBRARY_PATH: /home/wintermute/.local/lib/python3.10/site-
packages/cv2/../../lib64:
2022-10-03 12:57:14.552228: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could
not load dynamic library 'libnvinfer_plugin.so.7'; dlerror:
```

```
libnvinfer_plugin.so.7: cannot open shared object file: No such file
or directory; LD_LIBRARY_PATH:
/home/wintermute/.local/lib/python3.10/site-packages/cv2/../../lib64:
2022-10-03 12:57:14.552245: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning:
Cannot dlopen some TensorRT libraries. If you would like to use Nvidia
GPU with TensorRT, please make sure the missing libraries mentioned
above are installed properly.
```

```
labels = keras.utils.to_categorical(labels,num_classes = 5)

print(labels)
```

```
[[1. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0.]
 ...
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 1.]]
```

```
x_train,x_test,y_train,y_test =
train_test_split(images,labels,test_size = 0.2,random_state = 42)

x_train.shape
```

```
(6907, 150, 150, 3)
```

```
x_train,x_val,y_train,y_val =
train_test_split(x_train,y_train,test_size = 0.15,random_state = 42)

import keras
from keras.models import Sequential
from keras.layers import Dense, Activation, Dropout, Flatten
from keras.layers import Conv2D
from keras.layers import MaxPooling2D,MaxPool2D
from keras.optimizers import Adam
from tensorflow.keras.layers import Normalization
from tensorflow.keras.layers import LeakyReLU
from tensorflow.keras.layers import Conv2D, MaxPooling2D,
GlobalMaxPooling2D

model = Sequential()

model.add(Conv2D(128, (3, 3), input_shape=x_train.shape[1:]))
model.add(LeakyReLU(alpha=0.02))

model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, (3, 3)))
model.add(LeakyReLU(alpha=0.02))
```

```
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(GlobalMaxPooling2D())

model.add(Dense(512))
model.add(LeakyReLU(alpha=0.02))
model.add(Dropout(0.5))

model.add(Dense(5))
model.add(Activation('softmax'))
```

```
2022-10-03 12:57:30.868525: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful
NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero
2022-10-03 12:57:30.998271: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful
NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero
2022-10-03 12:57:30.999508: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful
NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero
2022-10-03 12:57:31.003081: I
tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow
binary is optimized with oneAPI Deep Neural Network Library (oneDNN)
to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the
appropriate compiler flags.
2022-10-03 12:57:31.012538: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful
NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero
2022-10-03 12:57:31.013477: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful
NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero
2022-10-03 12:57:31.013641: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful
NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero
2022-10-03 12:57:33.469275: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful
NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero
2022-10-03 12:57:33.470573: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful
```

```
NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero
2022-10-03 12:57:33.472435: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful
NUMA node read from SysFS had negative value (-1), but there must be
at least one NUMA node, so returning NUMA node zero
2022-10-03 12:57:33.473338: I
tensorflow/core/common_runtime/gpu/gpu_device.cc:1616] Created
device /job:localhost/replica:0/task:0/device:GPU:0 with 2121 MB
memory:  -> device: 0, name: NVIDIA GeForce RTX 3050 Laptop GPU, pci
bus id: 0000:02:00.0, compute capability: 8.6
```

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

```
history = model.fit(x_train, y_train, epochs=25, batch_size=32,
validation_data=(x_val, y_val))
```

```
2022-10-03 12:57:42.446183: W
tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of
396225000 exceeds 10% of free system memory.
2022-10-03 12:57:43.224683: W
tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of
396225000 exceeds 10% of free system memory.

Epoch 1/25

2022-10-03 12:57:52.783676: I
tensorflow/stream_executor/cuda/cuda_dnn.cc:384] Loaded cuDNN version
8500
2022-10-03 12:57:55.865772: I
tensorflow/stream_executor/cuda/cuda_blas.cc:1614] TensorFloat-32 will
be used for the matrix multiplication. This will only be logged once.

184/184 [==============================] - 53s 203ms/step - loss:
5.4625 - accuracy: 0.4342 - val_loss: 1.0550 - val_accuracy: 0.6316
Epoch 2/25
184/184 [==============================] - 36s 197ms/step - loss:
1.0916 - accuracy: 0.5644 - val_loss: 0.9466 - val_accuracy: 0.6577
Epoch 3/25
184/184 [==============================] - 41s 222ms/step - loss:
1.0087 - accuracy: 0.6060 - val_loss: 0.9776 - val_accuracy: 0.6528
Epoch 4/25
184/184 [==============================] - 41s 222ms/step - loss:
0.9426 - accuracy: 0.6390 - val_loss: 0.8582 - val_accuracy: 0.6818
Epoch 5/25
184/184 [==============================] - 44s 241ms/step - loss:
0.9008 - accuracy: 0.6508 - val_loss: 0.8867 - val_accuracy: 0.6606
Epoch 6/25
184/184 [==============================] - 63s 342ms/step - loss:
0.8671 - accuracy: 0.6681 - val_loss: 0.8365 - val_accuracy: 0.6644
```

```
Epoch 7/25
184/184 [==============================] - 82s 445ms/step - loss:
0.8294 - accuracy: 0.6794 - val_loss: 0.7286 - val_accuracy: 0.7406
Epoch 8/25
184/184 [==============================] - 89s 485ms/step - loss:
0.8116 - accuracy: 0.6894 - val_loss: 0.7453 - val_accuracy: 0.7387
Epoch 9/25
184/184 [==============================] - 89s 485ms/step - loss:
0.8066 - accuracy: 0.6918 - val_loss: 0.7996 - val_accuracy: 0.7040
Epoch 10/25
184/184 [==============================] - 89s 485ms/step - loss:
0.7702 - accuracy: 0.7089 - val_loss: 0.7026 - val_accuracy: 0.7387
Epoch 11/25
184/184 [==============================] - 89s 483ms/step - loss:
0.7501 - accuracy: 0.7121 - val_loss: 0.7037 - val_accuracy: 0.7512
Epoch 12/25
184/184 [==============================] - 64s 345ms/step - loss:
0.7370 - accuracy: 0.7208 - val_loss: 0.7229 - val_accuracy: 0.7367
Epoch 13/25
184/184 [==============================] - 45s 246ms/step - loss:
0.7489 - accuracy: 0.7155 - val_loss: 0.7112 - val_accuracy: 0.7358
Epoch 14/25
184/184 [==============================] - 43s 235ms/step - loss:
0.7067 - accuracy: 0.7358 - val_loss: 0.7007 - val_accuracy: 0.7338
Epoch 15/25
184/184 [==============================] - 47s 256ms/step - loss:
0.7255 - accuracy: 0.7249 - val_loss: 0.7051 - val_accuracy: 0.7464
Epoch 16/25
184/184 [==============================] - 53s 287ms/step - loss:
0.7180 - accuracy: 0.7242 - val_loss: 0.5987 - val_accuracy: 0.7772
Epoch 17/25
184/184 [==============================] - 89s 484ms/step - loss:
0.6625 - accuracy: 0.7450 - val_loss: 0.6584 - val_accuracy: 0.7551
Epoch 18/25
184/184 [==============================] - 89s 485ms/step - loss:
0.6918 - accuracy: 0.7378 - val_loss: 0.6580 - val_accuracy: 0.7647
Epoch 19/25
184/184 [==============================] - 89s 484ms/step - loss:
0.6697 - accuracy: 0.7514 - val_loss: 0.6395 - val_accuracy: 0.7512
Epoch 20/25
184/184 [==============================] - 89s 484ms/step - loss:
0.6560 - accuracy: 0.7537 - val_loss: 0.7107 - val_accuracy: 0.7358
Epoch 21/25
184/184 [==============================] - 89s 484ms/step - loss:
0.6440 - accuracy: 0.7537 - val_loss: 0.7112 - val_accuracy: 0.7107
Epoch 22/25
184/184 [==============================] - 89s 484ms/step - loss:
0.6325 - accuracy: 0.7549 - val_loss: 0.6934 - val_accuracy: 0.7329
Epoch 23/25
184/184 [==============================] - 89s 485ms/step - loss:
```

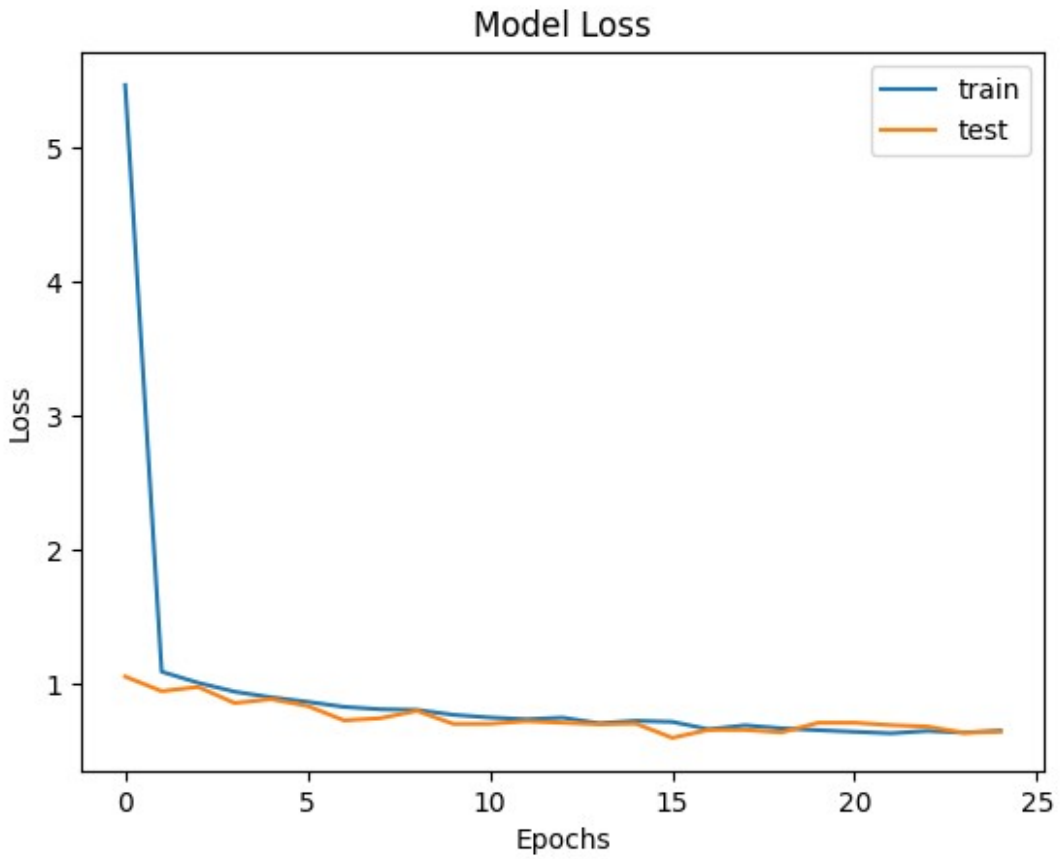
```
0.6503 - accuracy: 0.7506 - val_loss: 0.6827 - val_accuracy: 0.7531
Epoch 24/25
184/184 [==============================] - 89s 484ms/step - loss:
0.6375 - accuracy: 0.7629 - val_loss: 0.6367 - val_accuracy: 0.7541
Epoch 25/25
184/184 [==============================] - 89s 484ms/step - loss:
0.6524 - accuracy: 0.7560 - val_loss: 0.6443 - val_accuracy: 0.7676

print("Test Accuracy: {0:.2f}%".format(model.evaluate(x_test,y_test)
[1]*100))

54/54 [==============================] - 6s 103ms/step - loss: 0.6752
- accuracy: 0.7678
Test Accuracy: 76.78%

model.save_weights('Assignment_3_Final_Weights.h5')

plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epochs')
plt.legend(['train', 'test'])
plt.show()
```



```
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epochs')
plt.legend(['train', 'test'])
plt.show()
```