



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No 10:

Aim: Program to demonstrate use of numpy array for working with images

Theory

Images are an easier way to represent the working model. In Machine Learning, Python uses the image data in the format of Height, Width, Channel format. i.e. Images are converted into Numpy Array in Height, Width, Channel format.

Modules Needed:

NumPy: By default in higher versions of Python like 3.x onwards, NumPy is available and if not available(in lower versions), one can install by using
pip install numpy

Pillow: This has to be explicitly installed in later versions too. It is a preferred image manipulation tool. In Python 3, Pillow python library which is nothing but the upgradation of PIL only. It can be installed using
pip install Pillow

One can easily check the version of installed Pillow by using the below code

```
import PIL

print('Installed Pillow Version:', PIL.__version__)
```

Output:

Installed Pillow Version: 7.2.0

Loading the images via Pillow Library

Let us check for an image that is in the PNG or JPEG format. The image can be referred via its path. Image class is the heart of PIL. It has open() function which opens up an image and digital file format can be retrieved as well as pixel format.

Converting an image into NumPy Array

Python provides many modules and API's for converting an image into a NumPy array. Let's discuss a few of them in detail.

Using NumPy module

Numpy module in itself provides various methods to do the same. These methods are –

Method 1: Using asarray() function



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

asarray() function is used to convert PIL images into NumPy arrays. This function converts the input to an array

Method 2: Using numpy.array() function

By using numpy.array() function which takes an image as the argument and converts to NumPy array

In order to get the value of each pixel of the NumPy array image, we need to print the retrieved data that got either from asarray() function or array() function.

Getting back the image from converted Numpy Array

Image.fromarray() function helps to get back the image from converted numpy array. We get back the pixels also same after converting back and forth. Hence, this is very much efficient

PROGRAM:

```
from PIL import Image
import numpy as np

# Load the image
image = Image.open('cars..jpg')

# Convert image to numpy array
data = np.asarray(image)
print(type(data))

# Summarize shape
print(data.shape)
print(data)

# Convert numpy array back to image
image2 = Image.fromarray(np.uint8(data))
print(type(image2))

# Summarize image details
print(image2.mode)
print(image2.size)
```



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Output:

Original array:

```
[[1 2 3 4]
 [5 2 4 2]
 [1 2 0 1]]
```

Reshaped array:

```
[[[1 2 3]
   [4 5 2]]
```

```
[[4 2 1]
 [2 0 1]]]
```

Original array:

```
[[1 2 3]
 [4 5 6]]
```

Flattened array:

```
[1 2 3 4 5 6]
```

Array elements in sorted order:

```
[-1 0 1 2 3 4 4 5 6]
```

Row-wise sorted array:

```
[[ 1  2  4]
 [ 3  4  6]
 [-1  0  5]]
```

Column-wise sort by applying merge-sort:

```
[[ 0 -1  2]
 [ 1  4  5]
 [ 3  4  6]]
```

x.base: None

y.base: [1 2 3 4 5]

PS C:\Users\Lenovo\Downloads\Pvthon Prgs> █



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Conclusion:

The experiment effectively showcased the practical application of numpy arrays in imagemanipulation, underscoring their efficiency and versatility in processing visual data. By implementing fundamental numpy functions, the experiment demonstrated how these arrays can be harnessed to perform various image operations with ease and precision, providing a solid foundation for further exploration and development in the field of image processing.

