Artificial & Computational Intelligence / Published Online : June 2020
https://acors.org/ijacoi/VOL1_ISSUE3_02.pdf

Artificial & Computational Intelligence

Product Purity Analytic Tool Using Image Detection and Processing

Vartika Singh[1], Ekank Rana[2]
vsingh3@amity.edu[1], ekank.rana@gmail.com[2]
Amity University Noida, Uttar Pradesh, India

Abstract

Python is becoming increasingly popular programming language. It is a free, high-level language that has a very flat learning curve. It has a wide set of freely available libraries. In this paper computer vision libraries are first discussed. Then Image Detection and Image recognition capabilities of libraries available are analyzed. The basic description of the algorithm used in the libraries is given.
For each major step an example of the resulting image is provided. Although just two sample images are given in the paper, the algorithm was analyzed on many images. The analysis confirmed that Python is really the tool of choice for image detection and recognition tasks.

Keywords: Python, Image Processing, OpenCV2, Image Detection, Image Recognition, TensorFlow

1. Introduction

Python is a high-level general-purpose programming language created by Guido van Rossum in 1991. It has a design philosophy that puts emphasis on code readability. It supports multiple programming paradigms including object-oriented, imperative, functional and procedural and has a large standard and comprehensive library. The first release was followed by Python 2.0 in 2000 and Python 3.0 in 2008. At the time of writing this paper the latest version is Python 3.7. Python is a good choice for all the researchers in the scientific community because it is [1]:

• Free and open source
• A scripting language, meaning that it is Interpreted
• A modern language (object oriented, exception handling, dynamic typing etc.)
• Concise, easy to read and quick to learn
• Full of freely available libraries, in particular scientific ones (linear algebra, visualization tools, plotting, image analysis, differential equations solving, symbolic computations, statistics etc.)
• Useful in a wider setting: scientific computing, scripting, web sites, text parsing, Etc.
• Widely used in industrial applications

In comparison with other programming languages such as C/C++, Java, and Fortran, Python is a higher-level language. The computation time is therefore typically a little longer, but it is much easier to program in. In the case of C and Fortran, wrappers are also available. PHP and Ruby on the other side are high level languages as well. Ruby can be compared to Python but lacks scientific libraries. PHP on the other hand is a more web-oriented language. Python can also be compared to MATLAB, which has a really extensive scientific library. It is however not open source and free. Scilab and Octave are open source environments similar to MATLAB. Their language features are however inferior to the ones available in Python. People in general tend to think that complex problems demand complex processes in order to produce complex solutions. Python was developed with exactly the opposite philosophy. It has an extremely flat learning curve and development process for soft-ware engineers [4]. It is used for system administration tasks, by NASA both for development and as a scripting language in several of its systems, Industrial Light & Magic uses Python in its production of special effects for large-budget feature films, Yahoo! Uses it (among other things) to manage its discussion groups and Google has used it to implement many components of its web crawler and search engine [3]. As Python is also a language that is easy to learn and both powerful and convenient from the start [2], we might soon be asking, who is not using it. As already mentioned Python has an extensive set of libraries which can be imported into a project in order to perform specific tasks. The ones that really should be mentioned in any scientific paper dealing with mathematics are NumPy and SciPy. NumPy is a library which provides support for large, multi-dimensional arrays. As images are in fact large two (greyscale) or three (color) dimensional matrices, this library is essential in all image processing tasks. It should also be emphasized that many other libraries (not limited to image processing) use NumPy array representation. SciPy is a library built on the NumPy array object and contains modules for signal and image processing, linear algebra, fast Fourier transform, etc. The last library mentioned in this introductory section is Matplotlib. As the name suggests this library is a plotting library. Although it is used a lot in all areas of science, Image processing relies heavily on it. In this paper the libraries and their capabilities in the area of image processing, image analysis and computer vision in general are discussed. The execution of some of the most common algorithms in the field is also demonstrated together with the resulting images.

2. Python's image processing L1-Braries

There are several Python libraries related to Image processing and Computer vision. The ones that will be presented in this paper are:

• PIL/Pillow
This library is mainly appropriate for simple image manipulations (rotation, resizing, etc.) and very basic image analysis (histogram for example)

• OpenCV2
It's a library intended (as the name suggests) to be a simplified version of OpenCV. It doesn't offer all the possibilities of OpenCV, but it is easier to learn and use.

• OpenCV
It is by far the most capable and mostcommonly used computer vision library.
It is written in C/C++, but Python bindings are added during the installation. It also gives emphasis on real time image processing. Among the ones, which will not be presented, it might be worth mentioning Ilastik. It is a simple, user-friendly tool for interactive image classification, segmentation and analysis.

2.1 Python Imaging Library (PIL)

Python Imaging Library (PIL) is a library originally written by Fredrik Lundh. As PIL's last release dates back to 2009 it is a little outdated[5]. It's successor that also supports Python3 is called Pillow [6]. Consequently, not both of them can be installed at the same time. At the time of writing the paper, the last version of Pillow is 5.1.0. The most trivial program (showing an image) can be written as follows:

```
from PIL import Image
im=Image.open('/path/to/the/image/')
im.show()
```

Pillow is able to extract a lot of informationfrom an image and makes it possible to execute several standard procedures for image manipulation, including:

• per-pixel manipulations,
• masking and transparency handling,
• image filtering, such as blurring, contour-
ing, smoothing, or edge finding,
• image enhancing, such as sharpening, adjusting brightness, contrast or color

Some of them will be demonstrated. Image can for example be easily rotated for specific angle (in our case 45 ◦) and then saved by the following code:

```
rotated_image=im.rotate(45)
rotated_image.save('rotated.jpg')
```

A color image can also be split into differentcomponents (red, green and blue).

```
r, g, b = im.split()
r.show()
```

Image can also easily be sharpened or blurred.In this case it is however important that wealso import the ImageFilterlibrary. The codeneeded is shown below.

```
from PIL import ImageFilter
sharp=im.filter(ImageFilter.SHARPEN)
blur=im.filter(ImageFilter.BLUR)
```

Image can for example also be easily croppedwith the following command

```
cropped_im=im.crop((100,100,400,400))
```

It was demonstrated that PIL/Pillow is veryeasy if only basic image processing task areneeded. For more detailed analysis and computer vision OpenCV and OpenCV2 are moreappropriate.

2.2 OpenCV

OpenCV is an open source computer vision library available written in C and C++ which runs under Linux, Windows, Mac OS X, iOS, and Android. Interfaces are available for Python, Java, Ruby, Matlab, and other languages. A very simple program used just to show an image can be written as follows:

```
import numpy as np
import cv2
img = cv2.imread('lena-color.jpg')
cv2.imshow('image',img)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

As OpenCV is nowadays the most suitable library for computer vision we will use it in the remaining part of the paper.

3. Object Detection

Given an image or a video stream, an object detectionmodel can identify which of a known set of objectsmight be present and provide information about their positions within the image. Figure 1 shows the object detection.

For example, this screenshot of our example application shows how two objects have been recognized and their positions annotated:

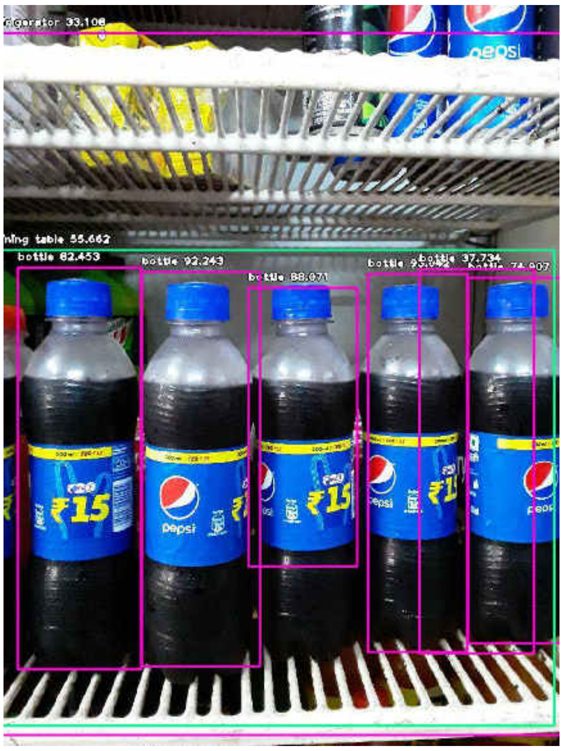

Figure 1 shows the object detection

```
execution_path = os.getcwd()
detector = ObjectDetection()
detector.setModelTypeAsRetinaNet()
detector.setModelPath( os.path.join(execution_path , "resnet50_coco_best_v2.0.1.h5"))
detector.loadModel()
detections = detector.detectObjectsFromImage(input_image=os.path.join(execution_path , "img/d51.jpg"), output_image_path=os.path.join(execution_path , "initial/initial.jpg"), minimum_percentage_probability=30)
count =0
for eachObject in detections:
  if(eachObject["percentage_probability"] > 55 and eachObject["name"] =='bottle'):
      print(eachObject["name"] , " : ", eachObject["box_points"] )
      print("--------------------------------")
      count =count+1
print("Initial count =",count)
```

An object detection model is trained to detect the presence and location of multiple classes of objects. For example, a model might be trained with images that contain various pieces of fruit, along with a *label* that specifies the class of objects they represent (e.g. chair, bottle, car, bike), and data specifying where each object appears in the image.

When we subsequently provide an image to the model, it will output a list of the objects it detects, the location of a bounding box that contains each object, and a score that indicates the confidence that detection was correct.

4. Object Recognition

Object detection is a technology that falls under the broader domain of Computer Vision. It deals with identifying and tracking objects present in images and videos. Object detection has multiple applications such as object detection, vehicle detection, pedestrian counting, self-driving cars, security systems, etc.
The two major objectives of object detection include:

- To identify all objects present in an image
- Filter out the object of attention

In this article, you will see how to perform object detection in Python with the help of the ImageAI library.

ImageAI

ImageAI is a Python library built to empower developers to build applications and systems with self-contained deep learning and Computer Vision capabilities using a few lines of straight forward code. ImageAI contains a Python implementation of almost all of the state-of-the-art deep learning algorithms like RetinaNet, YOLOv3, and TinyYOLOv3.ImageAI makes use of several APIs that work offline - it has object detection, video detection, and object tracking APIs that can be called without internet access. ImageAI makes use of a pre-trained model and can easily be customized. The Object Detection class of the ImageAI library contains functions to perform object detection on any image or set of images, using pre-trained models. With ImageAI, you can detect and recognize 80 different kinds of common, everydayobjects.

Figure 2 shows the Sample Image

The network architecture used for object recognition is based on the **resnet50_coco_best_v2.0.1.h5neural network**. The Python's object recognition library however has fewer layers and the number of filters is reduced by half. The network was trained on a dataset of approximately 3 million images (mainly the VGG dataset and the scrubs dataset).
The algorithm is composed of four steps:
1. Finding all the objects:
In the first step we could use the object detection algorithm described in the previous section , which is the most commonly applied one. The object recognition library however uses a more advanced Histogram of Oriented Gradients (HOG) method .Color images must first be converted to grayscale ones. Then for each pixel in the image we look at the direction in which the image is getting
Darker.

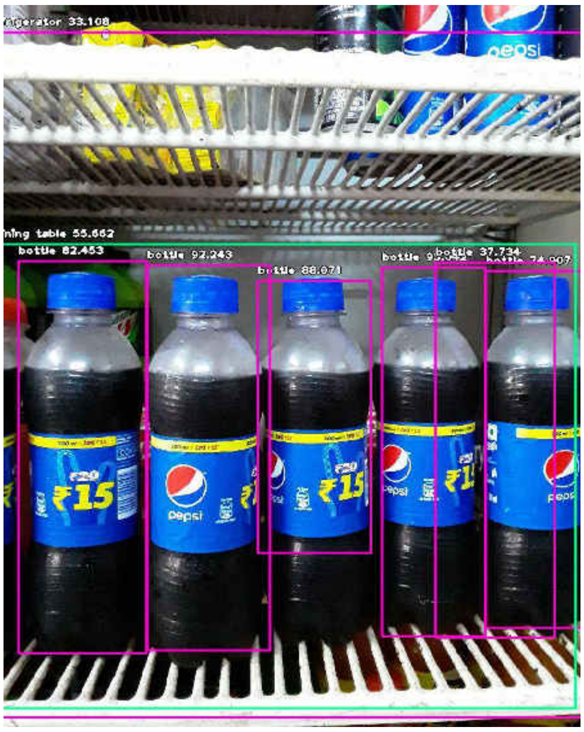

Fig 2.1

2. Matching the objects with sample:

This step deals with matching all the objects on basis of

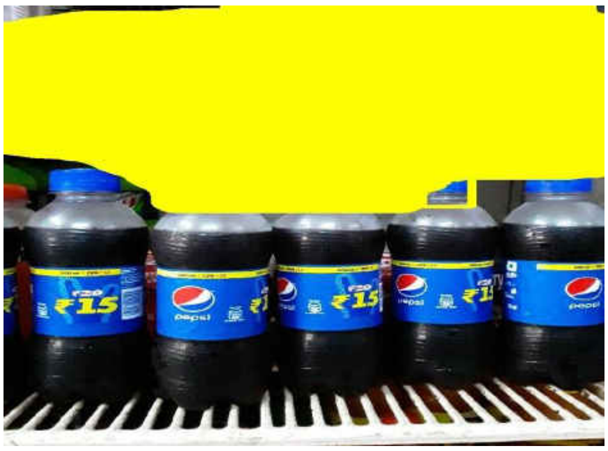

Fig 2.2

color, text, logo, shape with the sample image and mask the unmatched objects in the image like in the below figure.

```
res=cv2.matchTemplate(gray,template,cv2.TM_SQDIFF_NORMED)
min_val,max_val, min_loc, max_loc = cv2.minMaxLoc(res)
min_thresh = (min_val + 1e-6) * 1.5
match_locations = np.where(res<=min_thresh)
w, h = template.shape[::-1]
for (x, y) in zip(match_locations[1], match_locations[0]):
   cv2.rectangle(img, (x, y), (x+w, y+h), [0,255,255], 2)
```

3. Detect the object on masked image:

Now,again the masked image (Fig 2.1) is under pass through Object Detection Algorithm to determine the final count of objects matched with sample.

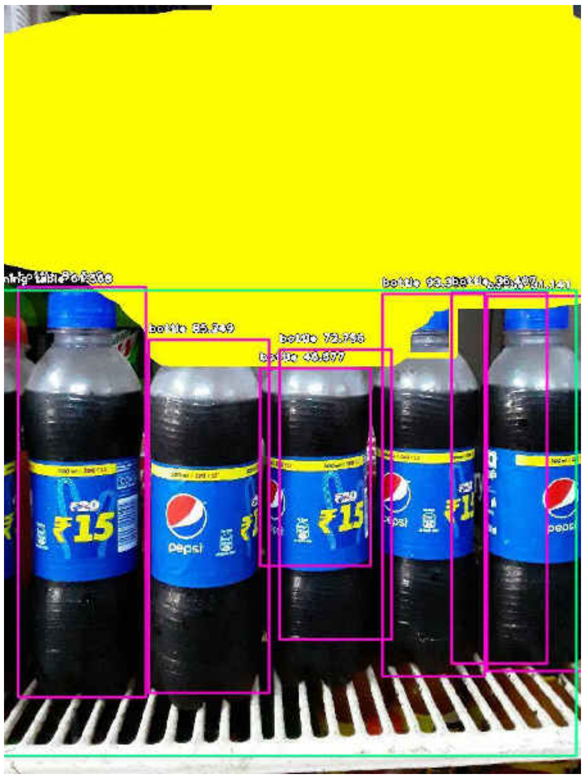

Fig 3.1

These objects are determined by the rectangle boxes as shown in fig 3.1

5. Conclusion

The paper is divided into two parts. In the first one a short overview of the most common Python's libraries related to image processing and computer vision is given. The second part uses the OpenCV library. In this part the libraries for object detection and object recognition are described and analyzed. object detection and object recognition are namely two areas of intensive research, because they enable a better interaction between computer system or robots on one side and humans on the other. Python's object recognition library turns out to be a fast and very reliable tool for object detection and object recognition. As Python is a high level programming language the library is well suited to be used just as an object detection (recognition) procedure in a wider project without the need for a detailed knowledge of the theoretical background of the employed algorithms. So, in our opinion it has a bright future.

In future it would be very interesting to investigate the possibilities of Python and its libraries for emotion detection. This field is a very hot topic in human machine inter object research. Using the results of this research it is possible to vastly improve the social aspects of robots or software packages that can adapt to the user. It namely gives a very reliable feedback in human machine interaction.

References

[1] https://pypi.org/project/object-detection/

[2] https://pypi.org/project/opencv-python/

[3] http://python-pillow.org/

[4] H. W. Ng, and S. Winkler, "A data-driven approach to cleaning large face datasets," IEEE International Conference on Image Processing (ICIP), pp. 343-347, 2014.

[5] N. Dalal, and B. Trigs, "Histograms of oriented gradients for human detection," IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 886-893, 2005.

[6] V. Kazemi, and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1867-1874,2014.

[7] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 815-823, 2015.

[8] S.Nagar, Introduction to Python: For Scientists and Engineers, Bookmuft, 2016.

[9] M. L. Hetland, Beginning Python: from novice to professional, 3rd Ed., Apress, 2017.