International Journal of Scientific Research in Science, Engineering and Technology
Print ISSN: 2395-1990 | Online ISSN : 2394-4099 (www.ijsrset.com)
doi : https://doi.org/10.32628/IJSRSET231039

Attendance System Using Face Detection and Face Recognition in Python Programming Language

R. Jeya Malar, Reenathangam G, Ranjani V
Department of ECE, Kings Engineering College, Mevalurkuppam, Tamil Nadu, India

ARTICLEINFO

Article History:

Accepted: 20 April 2023
Published: 10 May 2023

Publication Issue
Volume 10, Issue 3
May-June-2023

Page Number
52-58

ABSTRACT

The programming language Python is gaining popularity. It has a fairly flat learning curve and is a free, high-level language. It features a large selection of free libraries. The first topic covered in this study is computer vision libraries. The capabilities of the existing libraries for face detection and identification are next examined. The algorithm utilised in the libraries is provided a basic description. An example of the generated image is given for each significant stage. Even though the study only includes two sample photographs, the technique was tested on a large number of images. The investigation proved that Python is the go-to programme for tasks requiring face recognition and detection.

Keywords : Python, Image Processing, OpenCV, Face Detec- tion, Face Recognition

I. INTRODUCTION

Guido van Rossum developed the high- level general-purpose computer language Python in 1991. Its design philosophy places a strong emphasis on the readability of the code. It includes a sizable, comprehensive library and supports a variety of programming paradigms, including imperative, functional, procedural, and object-oriented. Python 2.0 and Python 3.0 were released after the original one, respectively, in 2000 and 2008. The most recent version of Python was 3.7 at the time this essay was being written. Every researcher in the scientific community should utilise Python since it is [1]:

Open source software
- ✓ a scripting language, which is an interpretive language
- ✓ handling exceptions, dynamic typing, etc.)
- ✓ short, simple to read, and easy to understand
- ✓ packed with freely accessible libraries, especially those related to science (linear algebra, visualisation tools, graphing, image analysis, solving differential equations, symbolic calculations, statistics, etc.).
- ✓ helpful in a variety of contexts, including scientific computing, scripting, websites, text parsing, etc.
- ✓ frequently used in commercial applications

Python is a higher-level language in compared to other programming languages like C/C++, Java, and Fortran. This results in slightly longer computation times but is much simpler to programme in. Wrappers are also offered for the programming languages C and Fortran. On the other hand, high-level languages like PHP and

Ruby are also used. Ruby is comparable to Python, however it doesn't have any scientific libraries. PHP, on the other hand, is more focused on the web. Additionally, Python is comparable to Matlab, which has a very sizable scientific library.However, it is not both free and open source.

Open source programmes that resemble Matlab include Scilab and Octave. However, compared to Python's, their language features are lacking. Most people have a tendency to believe that complicated problems require complex processes to develop complex answers.

The exact opposite philosophy guided the development of Python. For software programmers, the development process and learning curve are incredibly flat [4]. Industrial Light & Magic and NASA both utilise it as a scripting language in some of their systems for system management work. utilises Python to create special effects for high-budget feature films, Yahoo! uses it to run its discussion forums, among other things, and Google has utilised it to develop a number of its web crawler and search engine components [3]. Who isn't using Python? might be a question we ask in the near future given that it is also a language that is simple to learn and powerful and practical from the start [2]. As was already noted, Python includes a sizable library system that may be loaded into a project to carry out particular tasks.a project to carry out particular tasks. NumPy and SciPy are the two that ought to be specifically referenced in any academic publication that discusses mathematics. A library called NumPy offers support for big, multi-dimensional arrays. This library is necessary for all image processing jobs because images are actually big two- or three-dimensional matrices (greyscale or colour). It should be emphasised that NumPy array representation is used by numerous different libraries, not just those for image processing. The NumPy array object is the foundation of the SciPy library, which includes modules for signal and image processing, linear algebra, quick Fourier transform, etc.In this introduction, Matplotlib was the final library mentioned. This library is one for plotting, as its name suggests. Image processing strongly depends on it, despite the fact that it is employed extensively across all scientific disciplines. It is addressed in this study how libraries can be used for image processing, image analysis, and computer vision in general.Alongwith the generated photographs, the execution of some of the most popular field algorithms is also shown.

II. IMAGERY PROCESSING LI-BRARIES IN PYTHON

Python has a number of libraries for image processing and computer vision. The following will be discussed in this essay

- Pillow/PIL

This library is most suitable for basic picture analysis (such as histogram) and straightforward image editing (such as rotation, scaling, etc.).

- SimpleCV

Its a library intended (as the name sug- gests) to be a simplified version of OpenCV. It doesnt offer all the possibil- ities of OpenCV, but it is easier to learn and use.

- OpenCV

It is by far the most capable and most commonly used computer vision library. It is written in C/C++, but Python bind- ings 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 men- tioning Ilastik. It is a simple, user-friendly tool for interactive image classification, segmentation and analysis.

2.1 The PIL (Python Imaging Library)

The Python Imaging Library (PIL) was created by Fredrik Lundh in the beginning. PIL's most recent release was in 2009, so it is somewhat out of date [5].

Pillow [6] is its replacement, which also supports Python 3. As a result, they cannot both be installed at the same time. The most recent version of Pillow at the time the paper was being written was 5.1.0. The simplest programme (which displays an image) can be created as follows:

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

Pillow allows the execution of a number of common image editing techniques and can extract a lot of information from an image, including:

- ✓ pixel-by-pixel adjustments,
- ✓ picture enhancement techniques including sharpening, changing brightness, contrast, or colour; masking and transparency handling; image filtering techniques like blurring, contouring, smoothing, or edge finding; and A few of them will be shown. For instance, with the following code, an image may be simply rotated to a particular angle (in this case, 45 degrees) and saved. rotated_image=im.rotate(45) rotated_image.save('rotated.jpg')

A colour image can also be divided into its red, green, and blue component parts.
= im.split() r, g, and b r.show()
Additionally, blurring or sharpening an image is simple.But in this instance, it's crucial that we additionally import the ImageFilter library. The required code is displayed below.
sharp=im.filter(ImageFilter.SHARPEN) import ImageFilter from PIL
blur=im.filter(ImageFilter.BLUR)
For instance, cropping an image with the following command: cropped_im=im.crop((100,100,400,400))It was shown that PIL/Pillow is relatively simple to use if only elementary image processing tasks are required. Computer vision and analysis for more detail OpenCV and SimpleCV are better options.

2.2 SimpleCV

The Python interface for open-source machine vision libraries is called SimpleCV. As implied by the name, it is a condensed version of OpenCV.
Due to its concision, understandable camera interface, and ability to carry out image modification, feature extraction, and format conversion, it is relatively simple to learn and use.
But as of this writing, Python 2 is still the only supported language. SimpleCV won't be covered in detail because most Python users have already switched to Python 3. A straightforward programme is shown below.
from SimpleCV import Image img = Image('lena.jpg') img.show()
Since OpenCV, which is still under development, can perform every function featured in SimpleCV, it will be highlighted.

2.3 OpenCV

With support for Linux, Windows, Mac OS X, iOS, and Android, OpenCV is a free and open- source computer vision library developed in C and C++. Matlab, Python, Java, Ruby, and other languages all have interfaces available. This is how to write a very basic programme that is only used to display images:
import numpy as np import cv2
img = cv2.imread('lena-color.jpg') cv2.imshow('image',img) cv2.waitKey(0) cv2.destroyAllWindows()
Considering that OpenCV is currently the best library for computer vision, we will use it in the next sections of the study.

III. DETECTION OF FACE

Even more difficult jobs are pretty simple for us to complete using OpenCV. For instance, there are procedures that can identify faces and eyes in an image. The commands in the following list accomplish that.
face_cascade=cv2.CascadeClassifier
('C:\\Users\\...\\haarcascade

```
_frontalface_default.xml')
eye_cascade=cv2.CascadeClassifier
('C:\\Users\\...\\haarcascade_eye.xml')
img=cv2.imread('lena.jpg') gray=cv2.cvtColor
(img, cv2.COLOR_BGR2GRAY)
faces=face_cascade.detectMultiScale (gray, 1.3, 5)
for (x,y,w,h) in faces: cv2.rectangle
(img,(x,y),(x+w,y+h),(255,0,0),2)
roi_gray = gray[y:y+h, x:x+w] roi_color = img[y:y+h,
x:x+w]
eyes = eye_cascade.detectMultiScale (roi_gray)
for (ex,ey,ew,eh) in eyes:
cv2.rectangle(roi_color,(ex,ey),
(ex+ew,ey+eh),(0,255,0),2) cv2.imshow('img',img)
cv2.waitKey(0) cv2.destroyAllWindows()
```

The outcome is displayed in Fig. 1. Algorithm performs flawlessly for the image in Figure 1. However, the outcome is not as favourable when a more complex image is used, especially for the eyes. See, for instance, Fig. 2. The actual approach uses what are known as Haar feature-based cascade classifiers. Paul Viola and Michael Jones proposed it as an efficient object detection technique [9]. These features can be extremely many. However, most of them are unnecessary.

For instance, the fact that the area around the eyes is typically darker than the area around the nose and cheeks, is a positive trait. A second good

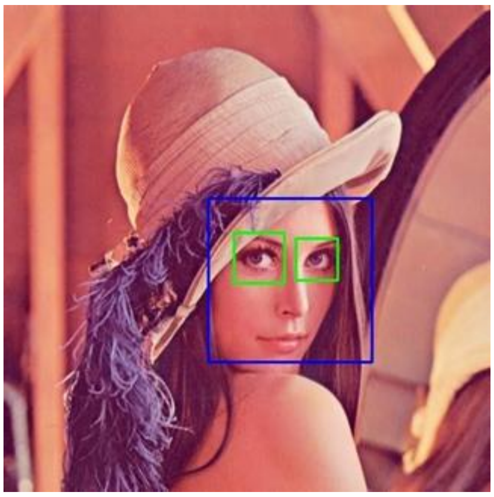
Figure 1. An example of face detection

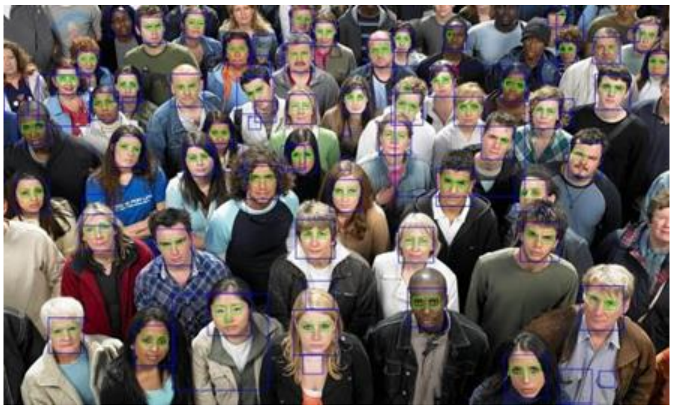
Figure 2. An example of face detection

The fact that the eyes are typically darker than the bridge of the nose, for instance, could serve as a basis for a second appealing characteristic. We can boost the algorithm's dependability by adding more of these features. Of course, misclassifications are always a possibility. It should be mentioned that when the number of pixels in the facial area reduces, the reliability also lowers.

IV. FACE RECOGNITION

Face recognition research can be summed up as the study of how to classify facial photos into sets that each represent an individual.Using Facebook as an example might make it simpler to comprehend. Facebook used to be able to identify faces (see previous section), but at that point, the user had to tag the person by clicking on the image and entering their name. Facebook may now automatically tag every person in the photo. Face recognition techniques are used to do this.Pretrained convolutional neural networks and OpenCV can be used in Python to complete this assignment.The whole program relies heav-

The programme as a whole largely depends on many libraries. The Python project must import the modules paths, face recognition, argparse, picle, and os.We must first gather several photographs of the individual we want to identify. Either manually entering the information or using Microsoft's Bing API can accomplish this. The dataset should ideally include at least 30 photos of each individual. The training photos shouldn't contain any other people.Two sample images (one for each person) are shownin Fig. 3. The network architecture used for.

Figure 3. Sample images of Bob Kelso and John Dorian

Fig. 3 displays two sample pictures (one for each person). ResNet-34 neural network is the foundation of the network architecture utilised for facial recognition [10]. The facial recognition library for Python, on the other hand, has less layers and the number of filtering is cut in half. Approximately 3 million rows of data were used to train the network.pictures, primarily from the scrubs dataset [12] and the VGG dataset [11]

The four steps of the algorithm are as follows:

1. Identifying every face

The face detection method [9], which is the one that is most frequently used, might be used as the initial step. The face recognition library, on the other hand, employs the more sophisticated Histogram of Oriented Gradients (HOG) technique [13]. Grayscale images must first be created from colour ones. Next, we examine which direction the image is getting darker for each individual pixel. As a result, we obtain a gradient matrix (see Fig. The brightness differences in the original image are largely unrelated to this matrix. However, it is too large to be manipulated.

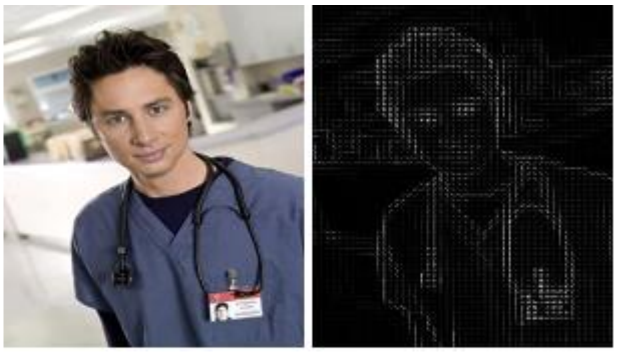
Figure 4. A sample of an original image and its histogram of gradients

Submatrices of size 16x16 are thus created. The dominating direction for each submatrix is then discovered.

2. Posing and Facial Projections

This phase addresses the issue that faces in an image might be gazing away from the camera and not directly at it. There are numerous approaches to this issue. The technique with 68 landmarks that are present on every face is used by the Python library. Figure 5 displays an illustration of one such picture.

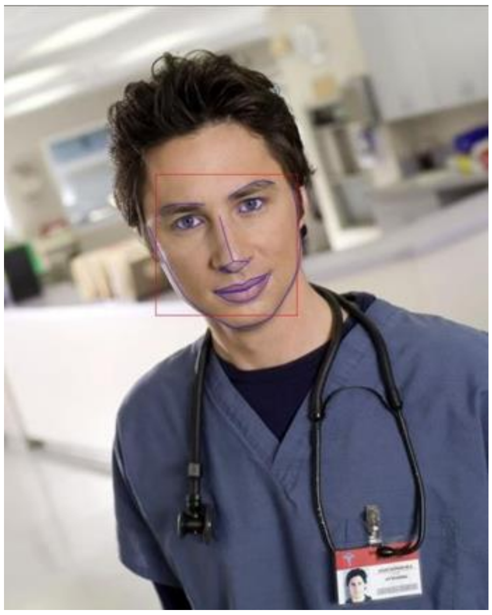
Figure 5. A face with the landmarks

The 68 landmarks are then used to train a machine learning system to identify them on any face. The face is then altered using affine transformations to centre the eyes and mouth as closely as possible.

3. Face encoding
For each face, the Deep Convolutional Neural Network is trained to produce 128 measures. Three images are used at a time during the training process: one of a familiar person, a second image of that person, and a third image of a different individual . A sizable dataset and a lot of computing power are needed for this stage. However, it only needs to be done once. There are also some pre-trained neural networks online.
4. Decoding the encoding to reveal the person's name
The final step is really rather simple.The faces in our database are contrasted with the face under analysis.Support vector machine (SVM) is used in Python's library to accomplish it. Any other categorization algorithm may theoretically be employed.

The performance of the algorithm is presented on the image shown in Fig. 6. It can be seen

Figure 6. An example of the Face detection algorithm

The image in Fig. 6 displays the performance of the algorithm. Although neither Bob Kelso nor John Dorian are looking directly at the camera, it is still clear that the technique is effective. In the case of John Dorian, the head is virtually perpendicular to the camera axis. Additionally, the entire image is noticeably darker than the norm. Nevertheless, the algorithm was flawlessly able to distinguish between the two of them. Figure 7 provides another another illustration. In this instance, there is no issue with brightness.John Dorian is again looking into the direction

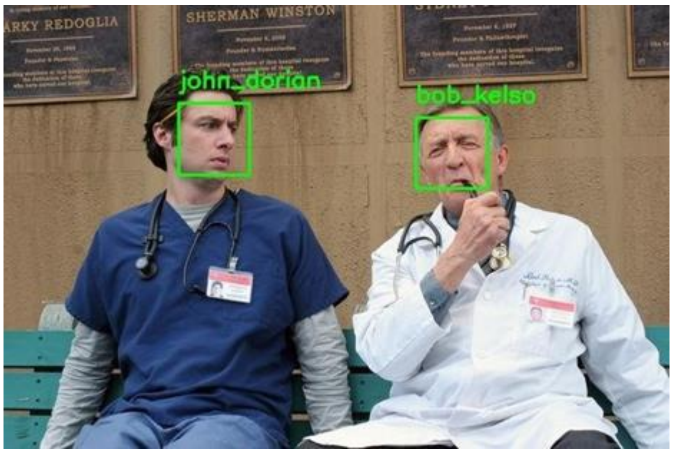

Figure 7. Another example of the Face detection algorithm

John Dorian is once more glancing towards Bob Kelso's general way. His facial expression also conveys emotion. In the case of Bob Kelso, we can observe that he is puffing on a pipe and sporting an odd smile on his face. Despite this, the system was able to accurately identify both of them.

V. CONCLUSION

The paper is split into two halves. An overview of the most popular Python libraries for image processing and computer vision is provided in the first one. OpenCV is utilised in the second section. The libraries for face detection and face recognition are explained and analysed in this section. Due to their ability to improve communication between computer systems or robots on the one hand and people on the other, face detection and face recognition are specifically two areas of intense research.The Python face recognition module proves to be a quick and dependable tool for face identification and face detection. Python is a high level programming language, therefore the library can be used in a larger project only as a face detection (or recognition) method without requiring in- depth understanding of the theory behind the algorithms utilised. Consequently, we believe it has a promising future.Future research on the capabilities of Python and its modules for emotion detection would be quite fascinating. In the realm of human-machine interface research, this area is particularly popular.Us- ing the results of this research it is possible to It is conceivable to significantly enhance the social capabilities of robots or software solutions that can adapt to the user using the findings of this research.

It specifically provides very trustworthy feedback in human-machine interactions.

VI. REFERENCES

[1]. C. Fuhrer, J. E. Solem, and O. Verdier, Scientific Computing with Python 3, Packt Publishing Ltd, 2016. (references)

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

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

[4]. R. V. Hattem, Mastering Python: master the art of writing beautiful and powerful Python by using all of the features that Python 3.5 offers, Packt Publish- ing, 2016.

[5]. http://www.pythonware.com/products/pil/

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

[7]. https://en.wikipedia.org/wiki/Python Imaging Library

[8]. A. Kaehler, and G. Bradski, Learning OpenCV: computer vision in C++ with the OpenCV library, 2nd Ed.," O'Reilly, 2016.

[9]. P. Viola, and M. Jones, "Rapid object detection us- ing a boosted cascade of simple features," Proceed- ings of the 2001 IEEE Computer Society Confer- ence on Computer Vision and Pattern Recognition, vol. 1, pp. I-511-I-518, 2001.

[10]. K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition," In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016.

[11]. O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," In British Machine Vision Conference, vol. 1, no. 3, p. 6, September, 2015.

[12]. 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.

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

Cite this article as :

R. Jeya Malar, Reenathangam G, Ranjani V, "Attendance System Using Face Detection and Face Recognition in Python Programming Language", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 10 Issue 3, pp. 52-58, May-June 2023. Available at doi : https://doi.org/10.32628/IJSRSET231039

Journal URL : https://ijsrset.com/IJSRSET231039