

# Comparison of Object Detection Algorithms

## Research Report

Team

Arpita Chatterjee [17BCA048]

K.V.Sanjana[17BCA051]

Mentor

Prof Dr Dhara Bhatt

Instructor

Prof Nithya Nadar

Department of Computer Science  
St Xavier's College (Autonomous), Ahmedabad  
India

## Abstract

Object detection involves detecting objects in a particular image using computer vision and image processing technology. The research has been done in Python programming language as it has the required packages and properties available when compared with other programming languages. This research aims towards comparing three object detecting algorithms, in order to find the best of the three algorithm which can be used further for making a computer perform better so as to contribute a bit to the ocean of artificial intelligence specifically in the area which uses object detection for their working, the future is moving towards making the computers do the work instead of humans.

## Introduction

Object detection involves detecting objects in a particular image using computer vision and image processing technology. For humans, it is easy to see and feel objects around us but it's not the same with computers, computers need instruction or blocks of code to see the objects and feel them, that's how robots are coded. Now to make a computer understand an object, that's where computer vision and image processing technology play the vital roles. Computer vision helps computer detect the objects whereas the image processing technology refers to processing of image in order to obtain useful information. This helps computer to see objects and categorize them accordingly, that is differentiating between an object and the background.

In the research following 3 algorithms have been compared:

1. Joint Detection
2. Contour Detection
3. Lineation

## Literature Review

The rapid increase in the usage of Object Detection algorithms in many important aspects of our lives such as Medical Diagnosis and Robotics etc. urges the need of a fast and accurate Object Detection algorithm that would be efficient and effective enough to carry out its expected work. For that our Research compares three algorithms namely Joint Detection, Contour Detection and Lineation on the basis of time and accuracy and formulate a result stating the best out of the three. The functional working of the three algorithms used in the Research can be briefed as below-

### Joints Detection

The function is created in this algorithm, to do joint detection. By default it uses CornerMinEigenVal algorithm. Similar characteristics can also be seen in Corner Harris Algorithm [1][22], but the difference between them is that you should specify the minimum distance between each point, the quality level and the number of corners to detect.

### Contours detection

To perform this algorithm, OpenCv package used in python provides a function called FindContours which helps in countouring the image. we first use the MorphologyEx function [16] with CV\_MOP\_OPEN method and CV\_MOP\_CLOSE to see the contours. Then we apply the FindContours function to find contours and print them on the colour image. [23]

### Lineation

In this algorithm, we used the cv.MorphologyEx. CV\_MOP\_GRADIENT to dilate and subtract the result to get the outline of the object in the image. [24]

The more we increase our threshold, the more harder to get out the pixels from our image.

## Problem Formulation

Most of the problems were faced when extracting packages from OpenCv[12]. OpenCv is an open source computer vision library that has a library of programming functions aimed mainly at real time computer vision especially for python, which helps solve the computer vision problems with the help of Numpy[7] and Scipy [9] packages. Python has been used as a language for object detection because, the packages available in it are more advanced and use latest technology in detecting objects either from an image or from a video. Compared to other languages like java which is a platform independent language, does not have enough packages to accurately detect objects in image when compared to python programming language.

Every one of the algorithm used in this research has faced initial execution problems due to lack of proper versions of packages. Packages from OpenCv [12] help solve the problems faced.

The contour detection algorithm faces problems when an image of multiple objects is placed. Whereas similar problems are faced by remaining algorithms but not as much as the contour detection faces.

## Mathematical Background

In the mathematical background, we can describe that the package Numpy and Scipy belong to the OpenCv- python programming language[6], has helped us a lot with its array assembling function used to detect the object in an image.

The Numpy has a huge library supporting large and multidimensional Arrays and matrices along with a large collection of mathematical function to operate on these arrays[7]. As the package contains most of the things it has made easy for the programmer to import them into the python IDLE(code editor).

Similar for Scipy, this package is open source and is mostly used for its modules like linear algebra, integration, special function etc

## Methodology

Our research comparison is carried out mainly in 3 phases. First we have assembled the 3 algorithms namely Joint Detection, Contour Detection and Lineation respectively. The second phase comprises of testing the algorithms on computer . Finally in the third step we have generated the result based on a comparison test between the three algorithms on the basis of time and accuracy. The image selected should be of high definition quality with just single object present to get the best out of each algorithm. The first phase that is the assembly phase includes gathering the codes for object detection with the help of algorithms. The second phase that is the testing phase includes running the algorithms different algorithm codes and the final phase, that is considerably the main phase called the comparison phase is carried out by carefully observing the three algorithms and generating a result which reveals the name of best of the three algorithms . The comparison is based on the time taken and the level of accuracy by the algorithms. According to the comparison a result is formulated.[20]

## Objective

Objective of object detection algorithm comparison which algorithm performs better in terms to find the location of an object in a given picture accurately and mark the object with the appropriate category. To be precise, the problem that object detection seeks to solve involves determining where the object is, and what it is with utmost accuracy.

However, solving this problem is not easy. Unlike the human eye, a computer processes images in two dimensions. Furthermore, the size of the object, its orientation in the space, its attitude, and its location in the image can all vary greatly.

Technically, the above task involves image recognition and positioning.

Image recognition (classification)

1. Input: High definition image
2. Output: the type of object
3. Method of evaluation: accuracy

Positioning

1. Input: High definition image
2. Output: the location of the box in the image
3. Method of evaluation: evaluation function intersection-over-union (IOU)[17]



## Result and Analysis

All the three algorithms have been worked upon a high definition image that has an equivocate object.

We have considered fig 1.0 as our original image on which we would execute our algorithms.



Fig: 1.0

After comparing the three object detection algorithms namely:

### 1. Joint Detection

. Joint detection is an image processing algorithm used for finding the joints of object within image .In this algorithm, we can see that the object is in the image is being detected by the algorithm in such a way that all the joints in the image are being detected, collectively only to detect an object in the given image [21].The algorithm, it manages to detect as many joints as possible.

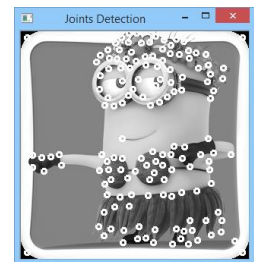


Fig: 2.0

### 2. Contour Detection

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same colour or intensity. The contours are a useful tool for shape analysis and object detection and recognition.[26]



Fig: 3.0

### 3. Lineation

Then the image is used by a threshold. At length, the objects are detected through morphology. Experimental results show that the proposed approach to object detection is feasible with Morphology.



Fig: 4.0

We were able to conclude that Lineation algorithm works the best among the three with its capability to detect the object accurately with minimum time of execution along with its threshold and MorphologyEx function being perfectly designed when compared to the rest of the provided algorithms

While the rest of the two algorithms, we were able to conclude that after Lineation algorithm, Joint detection algorithms works the best and then followed by the contour detection.

### Implementation Details

The project is implemented in python 2.4. OpenCV was used for image processing along with Numpy and Scipy python packages. The system specifications on which the model is trained and evaluated are mentioned as follows: CPU - Intel Core i3-4005U 1.70 GHz, RAM – 4.00 Gb, GPU - Nvidia .

## Conclusion

The research paper compares the different methods of detecting objects using algorithms. It basically compares three algorithms namely Joint Detection, Contour Detection and Lineation . Apart from these, we have other Object Detecting algorithms like sliding-window , morphology etc which are not mentioned here.

Although, there are many existing researches like AK Jain, NK ratha(Object Detection using Gabon filters)[18], A Borji ,MM Cheng(Salient Object Detection) [19]etc. but this research is different in the sense that it provides a comparison for the Object Detection algorithms namely Joint detection, Contour Detection and Lineation to find the best out of the three.

They are compared on the basis of the characteristics of the Object Detection algorithms such as time and accuracy. And hence as a conclusion we can declare Lineation Algorithm working at its best when compared to rest of the algorithms.

## Scope for Future Work

This research has wide scope in various areas and it can possibly increase the efficiency of its present utilization. It can help to convince switching to a better algorithm for Object Detection from present algorithm used. Moreover, Object is detected but also tries to find out its location.

In Future we will achieve accuracy in motion analysis the segmented moving object from tracking can be further analyzed with the statistics of each motion to verify whether a car is speeding or not, or whether a person is running, walking, or jumping. Processing time need to produce searching time by searching only in some parts of the image. Here we can use motion trigger and search only in the moving region. Searching algorithm such as hierarchical search or block matching algorithm might be able to make this program faster because it reduces number of pixels to be searched.

Use of object detection:

>Medical Diagnose: Use of object detection and recognition in medical diagnose to detect the X-Ray report, brain tumours.

>Shapes recognition: Recognize the shape from whole region in images.

>Cartography: The cartography as the discipline dealing with the conception, production dissemination and study of maps.

>Robotics: In robotics use of object detection is movement of body parts and motion sensing.

## Reference

1. [https://docs.opencv.org/2.4/modules/imgproc/doc/feature\\_detection.html?highlight=cornerharris](https://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html?highlight=cornerharris)
2. [https://docs.opencv.org/2.4/modules/imgproc/doc/feature\\_detection.html](https://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html)
3. [https://docs.opencv.org/2.4/modules/imgproc/doc/feature\\_detection.html](https://docs.opencv.org/2.4/modules/imgproc/doc/feature_detection.html)
4. [https://docs.opencv.org/2.4/modules/core/doc/drawing\\_functions.html](https://docs.opencv.org/2.4/modules/core/doc/drawing_functions.html)
5. <http://www.robindavid.fr/opencv-tutorial/>
6. <https://sourceforge.net/projects/opencvlibrary/>
7. <https://www.numpy.org/>
8. <http://cs231n.github.io/python-numpy-tutorial/>
9. <https://www.scipy.org/>
10. <https://www.tutorialspoint.com/scipy>
11. <https://pdfs.semanticscholar.org/6bca/fdf33445585966ee6fb3371dd1ce15241a62.pdf>
12. [https://docs.opencv.org/master/d9/df8/tutorial\\_root.html](https://docs.opencv.org/master/d9/df8/tutorial_root.html)
13. <https://www.analyticsvidhya.com/blog/2018/12/practical-guide-object-detection-yolo-framework-python/>
14. Artificial Intelligence: A Modern Approach Textbook by Peter Norvig and Stuart J. Russell
15. Learning OpenCV Book by Adrian Kaehler and Gary Bradski
16. [https://docs.opencv.org/3.0beta/doc/py\\_tutorials/py\\_imgproc/py\\_morphological\\_ops/py\\_morphological\\_ops.html](https://docs.opencv.org/3.0beta/doc/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html)
17. <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>
18. <https://www.semanticscholar.org/paper/Object-detection-using-gabor-filters-Jain-Ratha/cc2ec1863f7b7ce16b78bfdd8b5c7bd0f056c99e>
19. [https://www.researchgate.net/publication/268689513\\_Salient\\_Object\\_Detection\\_A\\_Survey](https://www.researchgate.net/publication/268689513_Salient_Object_Detection_A_Survey)
20. <http://ijcset.net/docs/Volumes/volume6issue1/ijcset2016060110.pdf>
21. <https://www.learnopencv.com/hand-keypoint-detection-using-deep-learning-and-opencv/>
22. [https://opencv-python-tutroals.readthedocs.io/en/latest/py\\_tutorials/py\\_feature2d/py\\_features\\_harris/py\\_features\\_harris.html](https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_features_harris/py_features_harris.html)

23. <https://www.superdatascience.com/opencv-face-recognition/>
24. <http://maniraj.xyz/.well-known/acme-challenge/blogs/sotpie/qoviaj2.php?hytsefe=opencv-pose-detection>
25. <https://www.cse.iitb.ac.in/~pratikm/projectPages/objectDetection/downloads/report.pdf>
26. [https://docs.opencv.org/3.1.0/d4/d73/tutorial\\_py\\_contours\\_begin.html](https://docs.opencv.org/3.1.0/d4/d73/tutorial_py_contours_begin.html)
27. <https://stackoverflow.com/questions/41486770/how-to-detect-and-draw-contours-using-opencv-in-python>
28. <http://opencvpython.blogspot.com/2012/06/hi-this-article-is-tutorial-which-try.html>