Int. J. Res. Ind. Eng. Vol. 9, No. 1 (2020) 46–64

International Journal of Research in Industrial Engineering

www.riejournal.com

Comparative Analysis on YOLO Object Detection with OpenCV

H. Deshpande*, A. Singh, H. Herunde

Department of Computer Application, Jain (Deemed to-be) University, Bengaluru, Karnataka, India.

A B S T R A C T

Computer Vision is a field of study that helps to develop techniques to identify images and displays. It has various features like image recognition, object detection and image creation, etc. Object detection is used for face detection, vehicle detection, web images, and safety systems. Its algorithms are Region-based Convolutional Neural Networks (RCNN), Faster-RCNN and You Only Look Once Method (YOLO) that have shown state-of-the-art performance. Of these, YOLO is better in speed compared to accuracy. It has efficient object detection without compromising on performance.

Keywords: YOLO, Faster-RCNN, Convolutional neural network, COCO.

Article history: *Received: 13 December 2019* *Revised: 23 January 2020* *Accepted: 17 March 2020*

1. Introduction

The development of software and hardware image processing systems was mainly limited to the development of the user interface. In the current era, computer science is a major subject. It has many real-life applications such as Internet of Things (IoT) [1-8], SPP [9-16], TP [17-19], PowerShell [20], uncertainty [21-23], cloud computing [24], artificial intelligence [25], internet security [26], and so on. The image processing plays a vital role in object recognition applications. Object recognition consists of the related computer vision tasks, including functions such as identifying objects in digital photographs. The region-based convolutional neural network like Fast R-CNN and Faster R-CNN are the state-of-the-art detection systems that have shown more definitions [27-29]. At the same time, many methods have been proposed in terms of feature map applications [28-33] which have achieved sophisticated performance. To overcome the problem of object detection, there are many methods introduced, namely the Viola-Jones detector with Haar Cascades [34], the HOG gradient-based approaches [35], and the

* Corresponding author
E-mail address: harideeps96@gmail.com
DOI: 10.22105/riej.2020.226863.1130

section and template matching approaches. In this manuscript, we discuss the comparative analysis of YOLO object detection with OpenCV. Here there are some methods used in this paper as follows:

First Method R-CNN. To overcome the problem of choosing many regions, Girshick et al. [27] introduced a method that uses selective search. The region with (R-CNN) (***Figure 1***) [36] has tried to improve the sliding window method. R-CNN first creates boundary boxes called regional schemes using the selected search process before insertion [37].

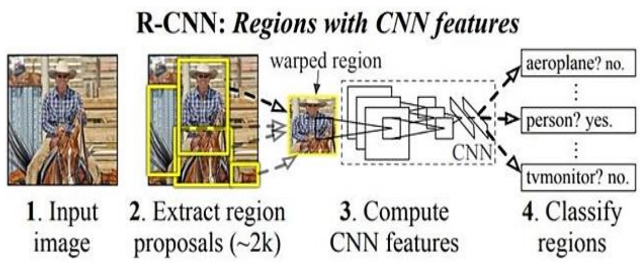

Figure 1. Selected search process in R-CNN [36].

Second Method ResNet. It is to train the network model in a more effective manner, then we adopt the same strategy as that used for DSSD (the performance of the residual network is better than that of the VGG network).

The Third Method is Fast R-CNN [38]. Here, the input image to CNN to generate a convolutional feature map. We can identify the region and transform them to a fixed size using a Region of Interest Pool (RoIPool) pooling layer (***Figure 2***). Fast R-CNN [38] brought (RoIPool) which tightens the extraction features and boundary boxes joined.

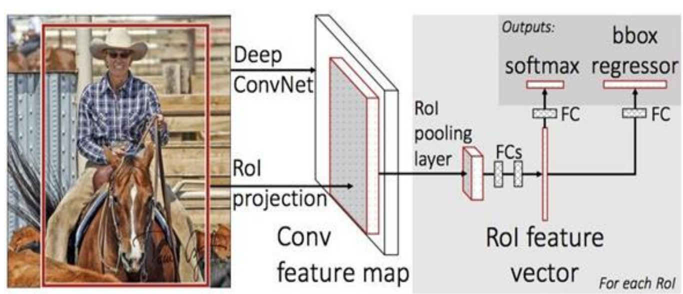

Figure 2. Fast R-CNN [38].

The Fourth Method is Faster R-CNN [39]. It's easy to understand and consumes less time to detect the object. The speed of training, detection, and selective search process are improved [40] (***Figure 3***).

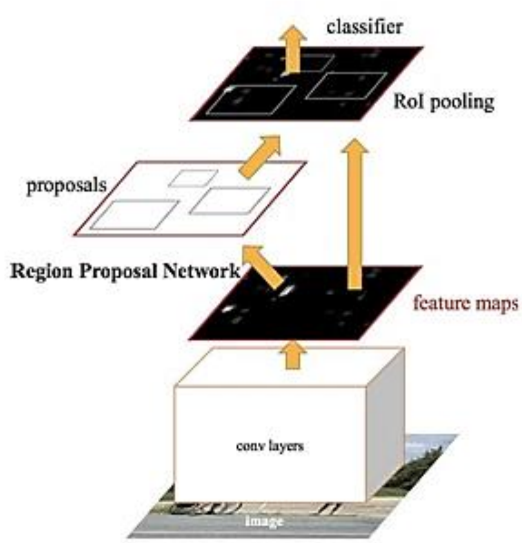

Figure 3. Faster R-CNN [39].

The Fifth Method is YOLO. The YOLO [41] that shows accurate results and looks at the highest probability of having the object. YOLO and its second amendment YOLOv2 [42] are the same as R-CNN in that they use probable boundary boxes (***Figure 4***), which are capable of extracting convolutional features, but differ from the Faster R-CNN systems.

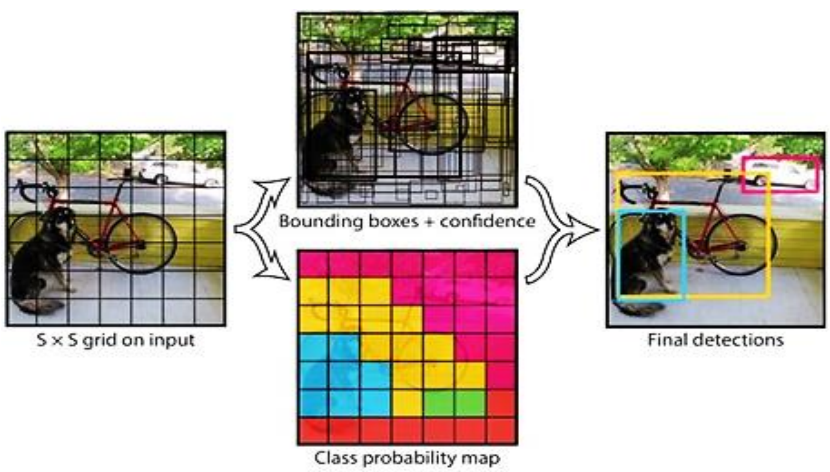

Figure 4. YOLO object detection model [32].

Publicly, the network is trained in an integrated dataset consisting of the MS COCO detection dataset [43].

2. Object Detection Methods

Object Detection has both object classification and the location issues of objects. The last step is to have the boundary boxes and labeled images. In the past, most used materials were a special face detector developed by Viola and Jones [44] to detect multiple object types. In image classification, researchers often focus on Convolutional Neural Networks (CNNs), which are influenced by the author's results [45]. First, CNN's such as VGNET [46] and Inception [47] were used for classification purposes. The image classification process takes input and returns the output of a class or multiple classes during multiple label classification [48].

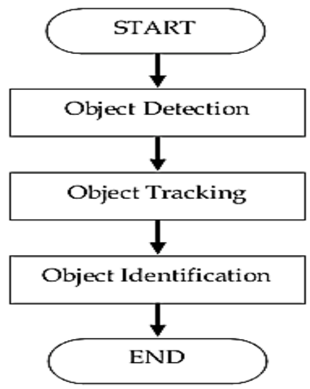

Figure 5. Flow of procedure [49].

Importantly, the object detection system has three main steps. In the first step, it detects the object. The second step is to track the necessary points. And the last step is identifying the object shown in the flowchart (***Figures (5)-(6)***).

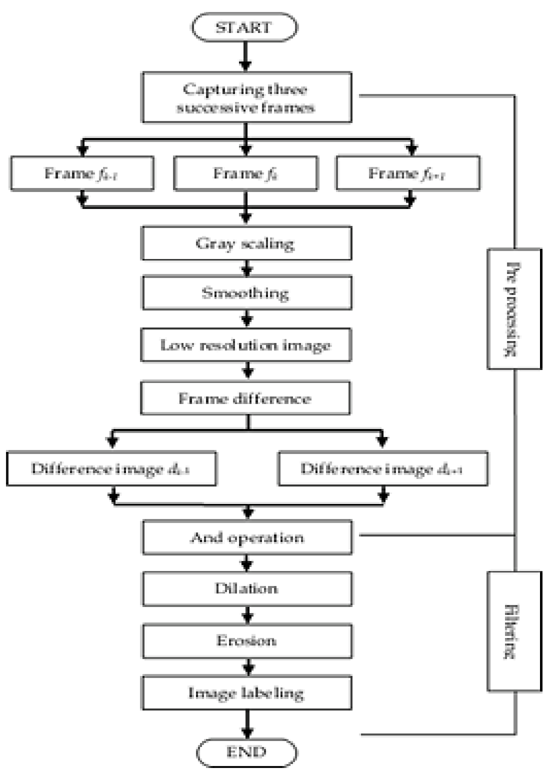

Figure 6. Flow of object detection [49].

2.1. Background Subtraction

The background subtraction method introduced by Horprasert et al. [50] was able to cope with local illumination changes, such as shadows and highlights, even globe illumination changes. In this method, the background model was statistically modeled on each pixel.

2.2. Template Matching

Template matching is a process of finding small parts of an image that match a template image. It slides the template from the top left to the bottom right of the image and compares for the best match with the template.

3. Literature Survey

Object tracking has many more technical unified spaces in the background subtraction methods. There are some works discussed for frame differencing that use the pixel-wise differences between two frame images that are background subtraction for detecting moving regions and background model for object detection by a Gaussian mixture model.

Lipton et al. [51] have proposed frame differences for using pixel-wise differences to get the motion objects. In another work, Stauffer and Grimson et al. [52] has proposed a Gaussian mixture model on the basis of a background model to identify the object. Liu et al. [53] have proposed background subtraction to detect the motion of objects in an image by obtaining the difference between current and reference background images in a pixel-by-pixel. Desa and Salih et al. [54] has proposed and improvised both background subtraction and frame difference. Sungandi et al. [55] proposed and introduced object detection using frame difference in low-resolution images. Jacques et al. [56] have proposed a new background model and shadow detection in grayscale video clips. Satoh et al. [57] proposed a new concept for object tracking using the PISC image-based block matching algorithm. Sugandi et al. [55] proposed tracking techniques for moving individuals using a camera peripheral signaling contact image. Authors [58] proposed in stereo camera-based object tracking, used Kalam filter to predict the position and speed of objects in the x-2 dimension [59] that proposes application of extended Kalam filter to calculate the 3D path of an object from 2D motion.

3.1. Motivation

After studying the literature, it has been found that tracking the object from the image sequence is really a challenging task.

- Tracking objects is a hard task because it takes more time to detect large data.
- In earlier days, detecting objects was an unsolvable task. But today, there is a number of methods to detect in reasonable accuracy.

- A literature survey has proposed different frameworks and methods to detect an object in high accuracy and revealed that many background subtraction algorithms perform efficiently.
- Object detection detects and localizes all known objects in a scene.
- In the YOLO method, the objects are identified very quickly, and results immediately and good for the real-time processing. This method has excellent results on the COCO dataset.

3.2. Objective of the Research

This article aims to improve the performance of object detection and tracking by increasing the speed of detecting objects and the accuracy of moving objects. At present, it detects only proper objects present in an image, if the objects are smaller or hidden, then it's difficult to detect. To overcome this problem, we choose another method called Faster R-CNN which identifies each and every object even if it's smaller, but its accuracy of identifying is slow compared to YOLO. Though, it does not recognize hidden objects, it recognizes every kind of object and also, needs to improve the segmentation. If speed is absolutely paramount, then we use YOLO.

The major objectives are:

- Analyzing grid images at a higher speed.
- Analyzing tracking methods for detecting hidden or smaller objects in an image.

4. Limitation or Shortcoming of Existing System

Here we discuss the limitations of some of the existing models:

4.1. R-CNN (2014)

It takes more time to train the network as we have to classify 2000 region proposals per image [36]. It cannot be implemented in real-time because it takes around 47 seconds for each and every test image. R-CNN was introduced in the year of 2014 combining region proposals with a CNN. Major drawbacks are that it was slow, hard to train, and consumes large memory. This method uses a selective search to generate regions and to detect the object.

4.2. ResNet (2015)

It is the most powerful deep neural networks which have achieved state-of-the-art performance on the ILSVRC 2015 classification challenge [60]. The first implemented was the VGG network. Suppose the input size is given as 300 and 320, even though the ResNet-101 layer is deeper than the VGG-16 layer, the main disadvantage is that it decreases the accuracy. The main thing is that it has the capacity to undergo a deeper layer.

4.3. Fast R-CNN (2015)

Fast R-CNN uses a single method that extracts features from the regions, then divides them into different classes, and returns boundary boxes for the identified classes. It takes time to concentrate on increasing accuracy and decreasing time. This method was introduced in the year of 2015 [38]. It has high accuracy compared to the previous method and detects the object in a faster way.

4.4. Faster R-CNN (2015)

In the above methods, a selective search is used to detect the region proposals, which is time-consuming and slow in the process. To overcome these problems, a new method was introduced, i.e. RoI Pool layer which is used to classify the image within the proposed region and can find the values for the boundary boxes. This method also consumes time and detects smaller or hidden objects that introduced in the year of 2015 [40].

4.5. YOLO (2015)

YOLO is a single-stage detector. The first breakthrough was in 2015 by Redmon et al. [32], it detects the real-time object and is very fast, better, and stronger compared to other methods. Its accuracy is very high. It has a COCO dataset to store the data of images and videos and has excellent results on the COCO dataset. YOLO helps to detect moving objects, recognizes and helps to display in a rectangular bounding box with a provided caption. The major advantage is that the fast-moving objects are captured very quickly compared to the rest of the methods. This method is mainly used for speed. It is faster compared to any other method.

5. Description of the Research Work

5.1. Research Problem

The problem with computer vision, image processing, and machine vision is that it determines whether or not the image data has a specific object, feature, or function. One of the major problems was the image classification. Image classification involves labeling an image based on the content of the image. For example, the objects in a particular image such as tray, fork, spoon, etc. It is only detected by fork and spoon, but not by the tray because inside the data set, tray keyword is not inserted. Therefore, the image classification will not be classified. There are some databases used as shown in (***Table 1***).

Table 1. Data sets.

Databases	Total Images
Test set 1	20

5.2. Solution Methodologies

There are many ways to detect objects. The best way for detecting object methods are convolutional neural network, RCNN, Fast-RCNN, Faster-RCNN, YOLO, OpenCV, etc. In this project, we use YOLO and OpenCV method for object detection which detects each and every object clearly. The last step is to have the boundary boxes and labeled images. It is easy to understand and consumes less time to detect the object. In table below it says about how the objects are detected and check the process goes through to detect an object. There are four steps to follow to get a proper object detected image. The steps involved in this method are as follows (***Table 2 & Figure 7***):

Table 2. Algorithm for proposed method.

Steps	Description
Step I	Consider an image and we need to create a grid that will give us the features of an object.
Step II	In this step, we make using of OpenCV which will read the input image and data points and specify the file path to an image in a Numpy array.
Step III	Detecting an image in a grid view after the process of reading image by OpenCV and Numpy and converting grid to rectangular boxes.
Step IV	The final step consists of displaying the image with the rectangular box along with the caption on the window. This is done using YOLO and COCO dataset.

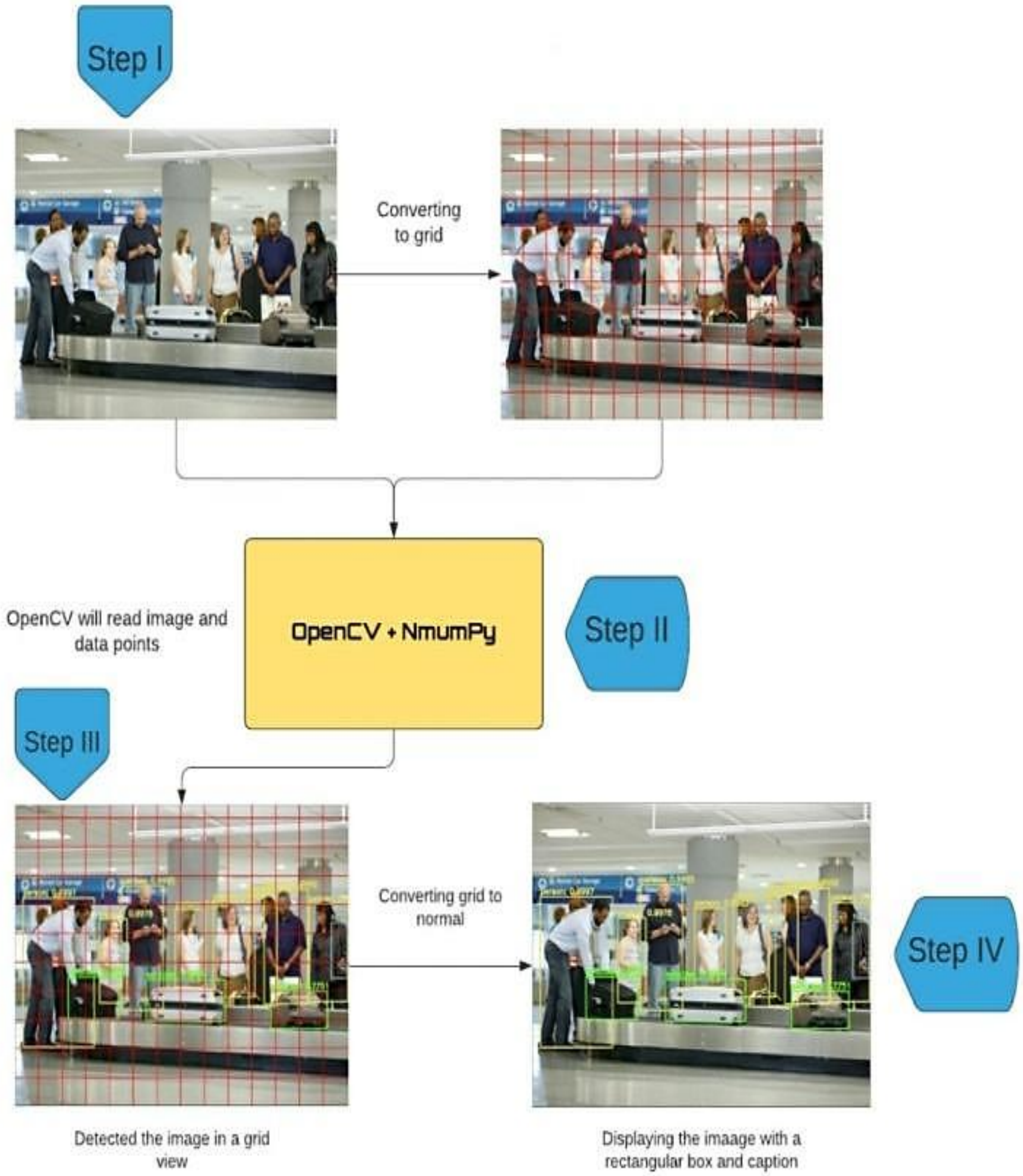

Figure 7. Flowchart of the proposed algorithm**.**

6. Result and Discussions

Figure 8. Before detection [61].

In the above-inserted image to the algorithm, we expect to detect the objects and label them.

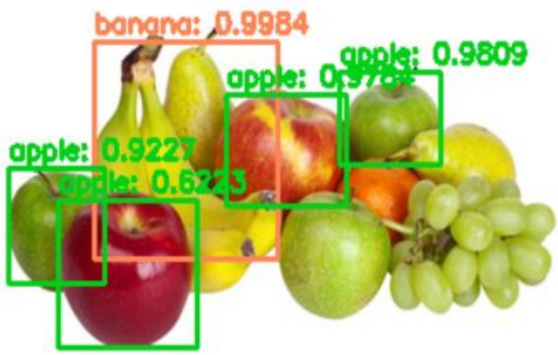

Figure 9. After detection.

Table 3. Comparative analysis based on time.

Methods	Fast R-CNN	Faster R-CNN	YOLO
Time (Sec)	41.64326	38.53267	1.929929

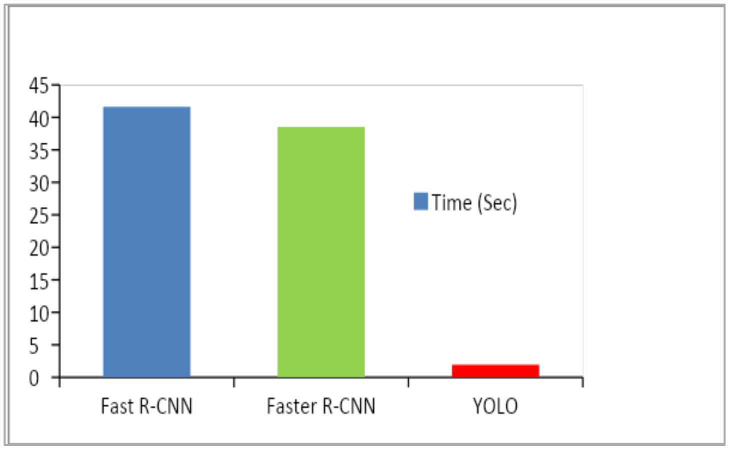

Figure 10. Time is taken to detect.

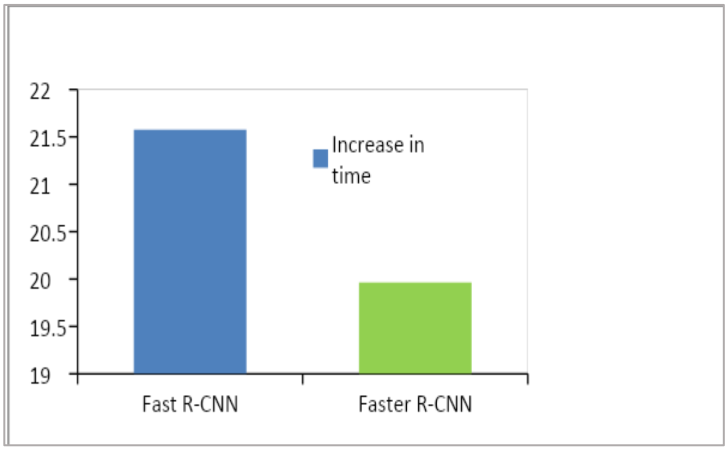

Figure 11. Comparison of time.

Discussion based on ***Figures (8)-(9)***, YOLO not only detects bags and person but also detects fruits. In the above image, the detected object are banana and apple. As expected, we have an output with the labeled object. Finally, the objects are correctly detected. This takes less time to detect a particular object. Because of a number of objects are less and the size of the object as well, so, it has detected in a lesser time compared to the two methods mentioned in ***Table 3***. Detecting this kind of images helps children to recognize fruits along with names, also helps to learn in a faster way.

As we see in the graph (***Figure 10***), the Fast R-CNN is 21 times slower than YOLO and Faster R-CNN is 19 times slower than YOLO and another observation is the Faster R-CNN is better than Fast R-CNN but not better than YOLO (see ***Figure 11***).

Figure 12. Before detection [62].

The above image inserted into the algorithm. We expect the algorithm to detect, identify, and label them according to the class assigned to it.

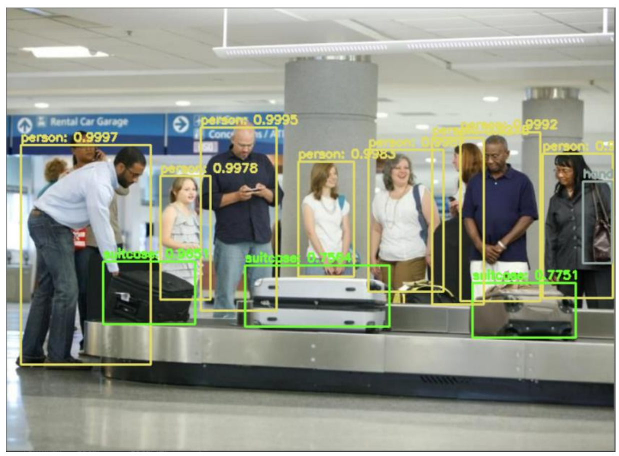

Figure 13. After detection.

Table 5. Comparative analysis based on time (example 2).

Methods	Fast R-CNN	Faster R-CNN	YOLO
Time (Sec)	55.769754	48.45637	7.366887

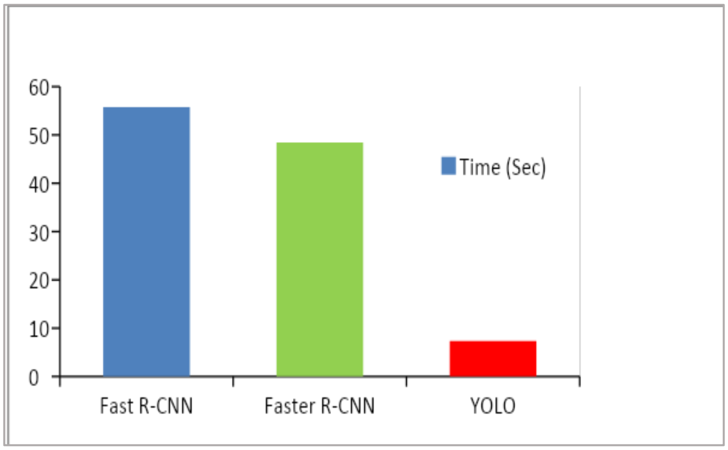

Figure 14. Time is taken to detect.

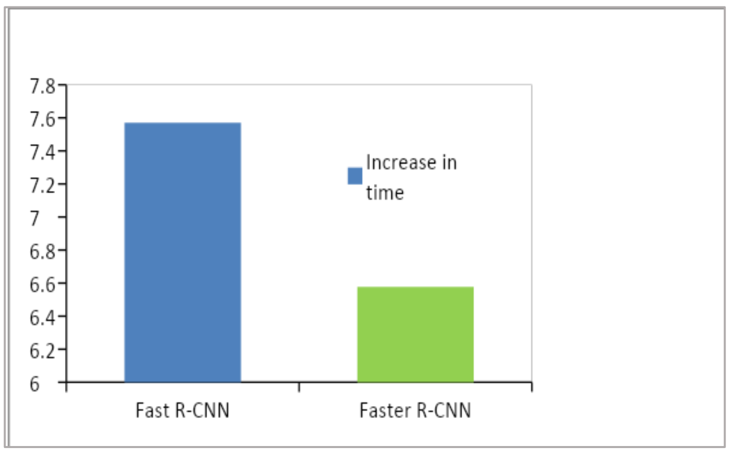

Figure 15. Comparison of time.

Discussion based on ***Figures (12)-(13)***, YOLO with OpenCV has detected person and suitcase. In deeper observation, we can also see there's another object detected at the right corner of an image, i.e. handbag. It has consumed more time compared to ***Figure 9***, because the size of the image and the number of objects are more so, it took more time compared to the other two methods as shown in ***Table 5***. This kind of detection helps people to find their lost bags or kids in an airport or any other places and also was used to obstacle avoidance.

As we see in the graph (***Figure 14***), YOLO is 7 times faster than Fast R-CNN and 6 times faster than Faster R-CNN and also Faster R-CNN is faster than Fast R-CNN but not faster than YOLO (see ***Figure 15***).

Figure 16. Before detection [63].

Above image is a live Bangalore traffic signal which consists of all kinds of vehicle, we expect the algorithm to detect each and every vehicle and label them.

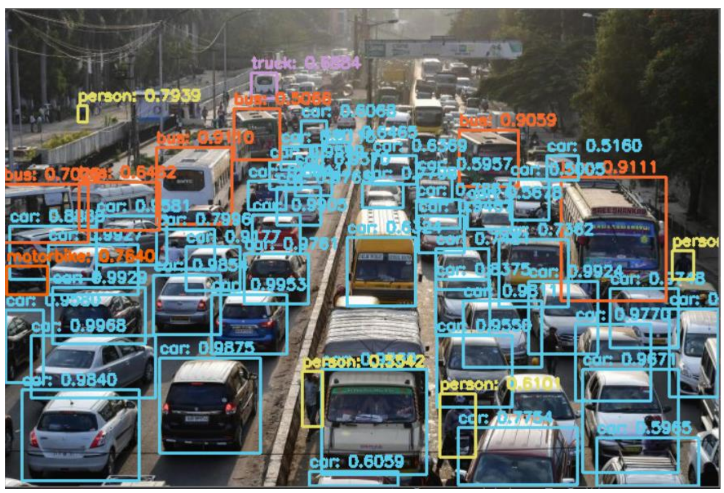

Figure 17. After detection.

Table 7. Comparative analysis based on time.

Methods	Fast R-CNN	Faster R-CNN	YOLO
Time (Sec)	83.81045	73.88907	19.389161

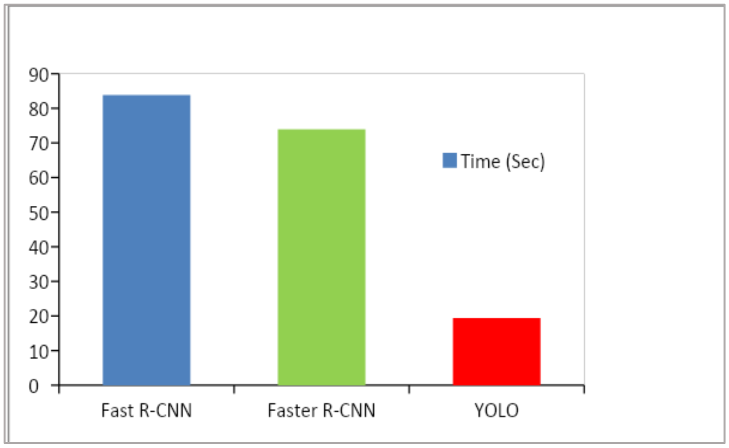

Figure 18. Time is taken to detect.

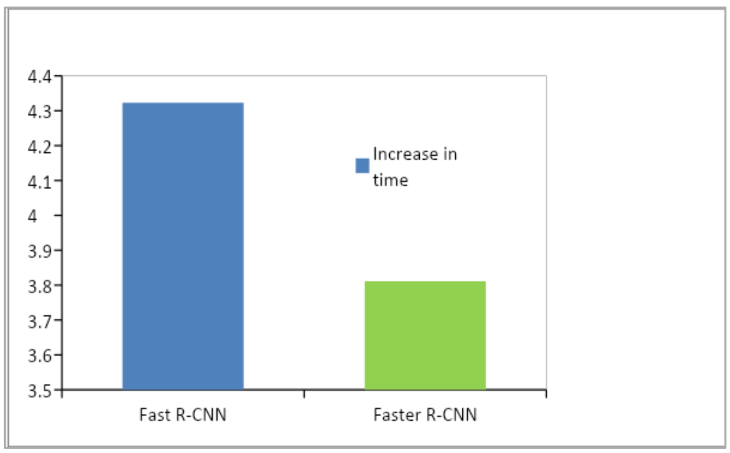

Figure 19. Comparison of time.

Discussion based on ***Figures (16)-(17)***, YOLO is able to correctly detect each car, bus, motorbike, truck, and person shown in an image as expected. We can notice that person at the right corner detected is slightly blurred and partially obscured that's a positive point of this method. Also in some parts of the image, persons are not detected because it's very small to detect this YOLO method. It struggles to detect small objects, which need to be improved. This method helps detecting an object at high speed compared to the other two methods mentioned in ***Table 7***. The use of detecting this inserted image helps to control the traffic signal. This image has consumed more time compared to ***Figure (9)-(13)***. Because of there are many objects seen on image and also it has to detect smaller objects, it takes time to understand and compute to give accurate results.

As we see in the graph (***Figure 18***), YOLO is 4 times faster than Fast R-CNN and 3 times faster than Faster R-CNN and also Faster R-CNN is better than Fast R-CNN but not better than YOLO and (see ***Figure 19***).

The final decision is, as the size of the image and number of objects increases, the time taken also increases and vice versa. Therefore, the number of objects is directly proportional to the time taken in detecting those objects.

7. Conclusion

Based on test results, the object can be detected more accurately and individually identified with the exact location of an object in the image along with the x and y-axis. This paper provided experimental results on different methods for object detection and identification and compared each method for their efficiencies, also performed efficient object detection while not compromising on the performance.

References

[1] Mohapatra, Hitesh. (2015). HCR (English) using neural network. *International journal of advance research and innovative ideas in education*, *1*(4), 379385.

[2] Mohapatra, H., & Rath, A. K. (2019). Detection and avoidance of water loss through municipality taps in India by using smart taps and ICT. *IET Wireless sensor systems*, *9*(6), 447-457.

[3] Mohapatra, H., & Rath, A. K. (2019). Fault tolerance in WSN through PE-LEACH protocol. *IET wireless sensor systems*, *9*(6), 358-365.

[4] Mohapatra, H., Debnath, S., & Rath, A. K. (2019). *Energy management in wireless sensor network through EB-LEACH* (No. 1192). Easy Chair.

[5] Nirgude, V., Mahapatra, H., & Shivarkar, S. (2017). Face recognition system using principal component analysis & linear discriminant analysis method simultaneously with 3d morphable model and neural network BPNN method. *Global journal of advanced engineering technologies and sciences*, *4*(1), 1-6.

[6] Panda, M., Pradhan, P., Mohapatra, H., & Barpanda, N. K. (2019). Fault tolerant routing in heterogeneous environment. *International journal of scientific & technology research, 8(8), 1009-1013.*

[7] Mohapatra, H., & Rath, A. K. (2019). Fault-tolerant mechanism for wireless sensor network. *IET wireless sensor systems*, *10*(1), 23-30.

[8] Swain, D., Ramkrishna, G., Mahapatra, H., Patr, P., & Dhandrao, P. M. (2013). A novel sorting technique to sort elements in ascending order. *International journal of engineering and advanced technology*, *3*(1), 212-126.

[9] Broumi, S., Dey, A., Talea, M., Bakali, A., Smarandache, F., Nagarajan, D., ... & Kumar, R. (2019). Shortest path problem using Bellman algorithm under neutrosophic environment. *Complex & intelligent systems*, *5*(4), 409-416.

[10] Kumar, R., Edalatpanah, S. A., Jha, S., Broumi, S., Singh, R., & Dey, A. (2019). A multi objective programming approach to solve integer valued neutrosophic shortest path problems. *Neutrosophic sets and systems*, *24*, 134-149.

[11] Kumar, R., Dey, A., Broumi, S., & Smarandache, F. (2020). A study of neutrosophic shortest path problem. In *Neutrosophic graph theory and algorithms* (pp. 148-179). IGI Global.

[12] Kumar, R., Edalatpanah, S. A., Jha, S., & Singh, R. (2019). *A novel approach to solve gaussian valued neutrosophic shortest path problems*. Infinite study.

[13] Kumar, R., Edalatpanah, S. A., Jha, S., Gayen, S., & Singh, R. (2019). Shortest path problems using fuzzy weighted arc length. *International journal of innovative technology and exploring engineering*, *8*, 724-731.

[14] Kumar, R., Edaltpanah, S. A., Jha, S., & Broumi, S. (2018). Neutrosophic shortest path problem. *Neutrosophic sets and systems*, *23*(1), 2.

[15] Kumar, R., Jha, S., & Singh, R. (2020). A different approach for solving the shortest path problem under mixed fuzzy environment. *International journal of fuzzy system applications (IJFSA)*, *9*(2), 132-161.

[16] Kumar, R., Jha, S., & Singh, R. (2017). Shortest path problem in network with type-2 triangular fuzzy arc length. *Journal of applied research on industrial engineering*, *4*(1), 1-7.

[17] Kumar, R., Edalatpanah, S. A., Jha, S., & Singh, R. (2019). A pythagorean fuzzy approach to the transportation problem. *Complex & intelligent systems*, *5*(2), 255-263.

[18] Smarandache, F., & Broumi, S. (Eds.). (2019). *Neutrosophic graph theory and algorithms*. Engineering science reference.

[19] Singh, R., & Saxena, V. (2017). A new data transfer approach through fuzzy Vogel's approximation method. *International journal of advanced research in computer science*, *8*(3), 515-519.

[20] Mohapatra, H., Panda, S., Rath, A., Edalatpanah, S., & Kumar, R. (2020). A tutorial on powershell pipeline and its loopholes. *International journal of emerging trends in engineering research*, *8*(4).

[21] Gayen, S., Smarandache, F., Jha, S., & Kumar, R. (2020). Interval-valued neutrosophic subgroup based on interval-valued triple T-Norm. In *Neutrosophic sets in decision analysis and operations research* (pp. 215-243). IGI Global.

[22] Gayen, S., Smarandache, F., Jha, S., Singh, M. K., Broumi, S., & Kumar, R. (2020). Introduction to plithogenic subgroup. In *Neutrosophic graph theory and algorithms* (pp. 213-259). IGI Global.

[23] Gayen, S., Jha, S., Singh, M., & Kumar, R. (2019). On a generalized notion of anti-fuzzy subgroup and some characterizations. *International journal of engineering and advanced technology*.

[24] Xu, X. (2012). From cloud computing to cloud manufacturing. *Robotics and computer-integrated manufacturing*, *28*(1), 75-86.

[25] Haenlein, M., & Kaplan, A. (2019). A brief history of artificial intelligence: on the past, present, and future of artificial intelligence. *California management review*, *61*(4), 5-14.

[26] Sakhnini, J., Karimipour, H., Dehghantanha, A., Parizi, R. M., & Srivastava, G. (2019). Security aspects of Internet of Things aided smart grids: a bibliometric survey. *Internet of things*, 100111.

[27] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587). http://openaccess.thecvf.com/menu.py

[28] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: towards real-time object detection with region proposal networks. *Advances in neural information processing systems* (pp. 91-99). Neural Information Processing Systems Foundation, Inc.

[29] Roh, M. C., & Lee, J. Y. (2017, May). Refining faster-RCNN for accurate object detection. *2017 fifteenth IAPR international conference on machine vision applications (MVA)* (pp. 514-517). IEEE.

[30] Zeng, X., Ouyang, W., Yang, B., Yan, J., & Wang, X. (2016, October). Gated bi-directional CNN for object detection. *European conference on computer vision* (pp. 354-369). Cham: Springer.

[31] Erhan, D., Szegedy, C., Toshev, A., & Anguelov, D. (2014). Scalable object detection using deep neural networks. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2147-2154). http://openaccess.thecvf.com/menu.py

[32] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788). http://openaccess.thecvf.com/menu.py

[33] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. *European conference on computer vision* (pp. 21-37). Cham: Springer.

[34] Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001* (Vol. 1, pp. I-I). IEEE.

[35] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* (pp. 1097-1105). Neural Information Processing Systems Foundation, Inc.

[36] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587). http://openaccess.thecvf.com/menu.py

[37] Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, *104*(2), 154-171.

[38] Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448). http://openaccess.thecvf.com/menu.py

[39] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* (pp. 91-99). Neural Information Processing Systems Foundation, Inc.

[40] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* (pp. 91-99). Neural Information Processing Systems Foundation, Inc.

[41] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788). http://openaccess.thecvf.com/menu.py

[42] Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263-7271). http://openaccess.thecvf.com/menu.py

[43] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014, September). Microsoft coco: Common objects in context. *European conference on computer vision* (pp. 740-755). Cham: Springer.

[44] Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition (CVPR)* (Vol. 1, pp. I-I). IEEE.

[45] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* (pp. 1097-1105). Neural Information Processing Systems Foundation, Inc.

[46] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *Computer vision and pattern recognition*. https://arxiv.org/abs/1409.1556

[47] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9). http://openaccess.thecvf.com/menu.py

[48] Pobar, M., & Ivasic-Kos, M. (2017, August). Multi-label poster classification into genres using different problem transformation methods. *International conference on computer analysis of images and patterns* (pp. 367-378). Cham: Springer.

[49] Bishop, G., & Welch, G. (2001). An introduction to the kalman filter. *Proc of SIGGRAPH, Course*, *8*(27599-23175), 41. Los Angeles: SIGGRAPH.

[50] Horprasert, T., Harwood, D., & Davis, L. S. (1999, September). A statistical approach for real-time robust background subtraction and shadow detection. In *IEEE ICCV* (pp. 1-19). Citeseer.

[51] Lipton, A. J., Fujiyoshi, H., & Patil, R. S. (1998, October). Moving target classification and tracking from real-time video. *Proceedings fourth IEEE workshop on applications of computer vision. WACV'98 (Cat. No. 98EX201)* (pp. 8-14). IEEE.

[52] Stauffer, C., & Grimson, W. E. L. (1999, June). Adaptive background mixture models for real-time tracking. *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)* (Vol. 2, pp. 246-252). IEEE.

[53] Liu, Y., Ai, H., & Xu, G. Y. (2001, September). Moving object detection and tracking based on background subtraction. *Proc. SPIE 4554, object detection, classification, and tracking technologies* (Vol. 4554, pp. 62-66). https://doi.org/10.1117/12.441618

[54] Desa, S. M., & Salih, Q. A. (2004, July). Image subtraction for real time moving object extraction. *Proceedings. International conference on computer graphics, imaging and visualization, 2004. CGIV 2004.* (pp. 41-45). Penang, Malaysia, Malaysia: IEEE.

[55] Sungandi, B., Kim, H., Tan, J. K., & Ishikawa, S. (2009). Real time tracking and identification of moving persons by using a camera in outdoor environment. *International journal of innovative computing, information and control, 5,* 1179-1188.

[56] Jacques, J. C. S., Jung, C. R., & Musse, S. R. (2005, October). Background subtraction and shadow detection in grayscale video sequences. *XVIII Brazilian symposium on computer graphics and image processing (SIBGRAPI'05)* (pp. 189-196). IEEE.

[57] Satoh, Y., Kaneko, S. I., & Igarashi, S. (2004). Robust object detection and segmentation by peripheral increment sign correlation image. *Systems and computers in Japan, 35*(9), 70-80.

[58] Beymer, D., & Konolige, K. (1999, September). *Real-time tracking of multiple people using continuous detection*. Paper presented at the IEEE frame rate workshop (pp. 1-8). Retrieved from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.46.7987&rep=rep1&type=pdf

[59] Rosales, R., & Sclaroff, S. (1999, June). 3D trajectory recovery for tracking multiple objects and trajectory guided recognition of actions. *Proceedings. 1999 IEEE computer society conference on computer vision and pattern recognition (Cat. No PR00149)* (Vol. 2, pp. 117-123). IEEE.

[60] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). ImageNet: a large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition* (pp. 248-255). IEEE.

[61] Anonymous. (2017). Nutrition & exercise - timing is everything. Retrieved October 12, 2019 from https://blog.nasm.org/workout-and-nutrition-timing

[62] Rosebrock, A. (2018). *YOLO object detection with OpenCV*. Retrieved December 25, 2019 from https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/

[63] Kappan, R. (2019). *Average commute speed in Bengaluru drops 18.7 kmph*. Retrieved December 21, 2019 from https://www.deccanherald.com/city/life-in-bengaluru/average-commute-speed-in-bengaluru-drops-to-187-kmph-757943.html