

A COMPARATIVE STUDY OF VARIOUS OBJECT DETECTION AND FEATURE EXTRACTION ALGORITHMS

Santhosh S Nayak¹

K Gaurav Shet², Jayasoorya³, Sourav K⁴, Nandakumar D N⁵

¹Assistant Professor, Dept. Of ECE

^{2,3,4,5}Students, Department of Electronics and Communication Engineering

Alva's Institute of Engineering and Technology

Affiliated to VTU, Belgaum, Approved by AICTE, New Delhi, Recognized by Govt. of Karnataka

Moodbidri 574225, D.K

To Cite this Article

Santhosh S Nayak, K Gaurav Shet, Jayasoorya, Sourav K, Nandakumar D N "A COMPARATIVE STUDY OF VARIOUS OBJECT DETECTION AND FEATURE EXTRACTION ALGORITHMS", *Journal of Science and Technology*, Vol. 07, Issue 02, Mar-April 2022.

Article Info

Received: 24-04-2022

Revised: 15-04-2022

Accepted: 22-04-2022

Published: 30-04-2022

Abstract- Agriculture is the most important sector in the Indian economy's development. In today's agriculture, there is a high demand to transition from tedious time-consuming manual harvesting to a fully automated operation. The combination of both autonomous harvesting and segregation of tomato fruit and three different phases or conditions of tomato is classified in this study. This project uses an image processing technique to classify the tomato's condition using a sample OpenCV tool. A Raspberry Pi camera is used to capture video continuously and this R-pi converts real time input video into frame, then object detection and image processing algorithm is performed on those image frame to detect shape of tomato, color filter is applied on tomato image to detect color of tomato. This paper suggests various important approaches for image processing and also suggests the most popular and widely used algorithms for object detection.

Keywords- Fruit harvesting, Segregation, Image processing, Object detection, OpenCV, R-pi

I. INTRODUCTION

Agricultural play a main role in economy, as well as it is considered to be the backbone of economic system for developing countries, the agriculture industry faces numerous challenges, including a lack of field employees and increasing fruit harvesting costs. Saving labour and scale up in agriculture is necessary in solving these problems. Agriculture automation has advanced in recent years, allowing for labour savings and large-scale farming. However, much of the work in the field of fruit harvesting is manually done. Today's whole population living in this 21st century, yet million tons of fruits and vegetables are plucked off manually, with this traditional method of manual harvesting and segregation leads to time consuming process, less accuracy and it may cause physical damage to the plants, an autonomous fruit plucking and segregation robot would help to overcome these issues thereby, fully replacing the manual harvesting and segregation method. Fruit harvesting by a robot

entails two major tasks: fruit detection and localization on trees using computer vision with a sensor and robot arm motion to the detected fruit's position, and fruit harvesting by end effectors without harming the target fruit or its tree.



Fig 1 Harvesting using Hand



Fig 1.2 Manual Harvesting tools

On the other side, it is also included with the autonomous segregation, there is no lot of implemented systems which contains both the harvesting and segregation unit, but in this proposed system, it provides an additional feature, where the robot itself segregates the plucked fruits based on three major factors like size, color and quantity. Fruit segregation plays an important role because, there are some fruits where one rotted fruit can damage or can rot other ripened fruits. So, in order to overcome all these challenges and to fully replace the manual plucking and harvesting system, this system is introduced in vertical farming.



Fig 1.3, 1.4 Manual Plucking in Vertical farming

The modern concept of vertical farming was proposed in 1999 by Professor Dickson Despommier. The concept was to grow the food in urban areas itself utilizing less distance and saving the time in bringing the food produced in rural areas to the cities. Vertical farming has the potential to help the environment by allowing more food to be produced with less resources. Minimization of water requirements through water recycling. The need for vertical farming is to ensure a consistent supply of products to demand centers, reducing the need for storage and refrigeration.

II. PROPOSED SYSTEM

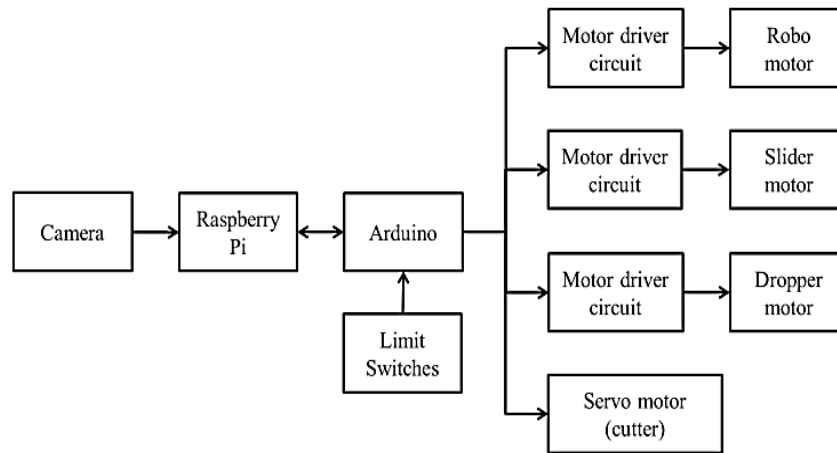


Fig 2.1 Block diagram of proposed system

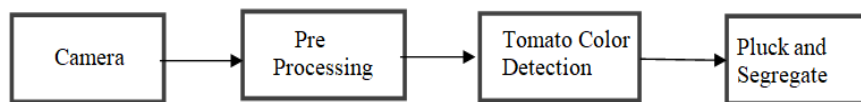


Fig 2.2 Block diagram of video processing

In this proposed system, a tomato fruit is considered as an example for experimentation, system consists of Raspberry pi, Arduino along with camera module, motors and motor driver circuits shown in fig 2.1 camera captures video continuously, Raspberry Pi converts real time input video into frame and then perform circle Hough Transform on image to detect shape of tomato. If tomato circular then it is considered as tomato is ripe. Then color filter is applied on tomato image to detect color of tomato shown in fig 2.2. The video captured from the camera. For the image processing purpose OpenCV library is used which allows image processing operations in python. Image is preprocessed to de-noise it, extract region of interest from image and extract features. Classify the color of tomato into green, yellow and red. After detecting color of tomato, Raspberry Pi sends information about tomato color and shape to Arduino as output. Arduino compares input from R-pi then Arduino command motor as follows, if tomato is red color, then Arduino command robo motor to go close to tomato, servo motor to cut tomato, dropper motor to drop tomato in slider, slider motor to drop tomato in box2. If tomato is yellow, command robo motor to go close to tomato, servo motor to cut tomato, dropper motor to drop tomato in slider, slider motor to drop tomato in box1. If tomato is green command robo motor to move forward.

III. DISCUSSION

Most of the reviewed papers use the concept of Open-Source Computer Vision Library (OpenCV), which is most used library in robotics to detect, track and understand the surrounding world captured by image sensors or cameras. Different types of Image processing algorithms are presented for object detection purpose; they are Region-Based Convolutional Network (R-CNN), Fast Region-Based Convolutional Network (Fast R-CNN), Faster Region-Based Convolutional Network (Faster R-CNN), Single Shot Detector (SSD), Random Sample Consensus (RANSAC), Iterative Relief (I-RELIEF), K-means clustering and YOLO algorithm etc.

Algorithm 1- Region-Based Convolutional Network (R-CNN).

Ross Girshick developed R-CNN in 2014. Regions with Convolutional Neural Networks is a deep learning strategy that combines box-sized region proposals with convolutional neural network properties. The R-CNN algorithm will first look for regions in the image that could contain an item, known as region proposals. The CNN features will then be calculated from the region proposals. It will then categorize the objects based on the extracted features in the last stage. Selective search is used by Areas with Convolutional Neural Networks to locate regions in an image, and it generates 2000 region proposals for each image, i.e., we get the region of interest (RoI). R-CNN will only view small regions and regions with good output during the classification of objects in the image. The image is cropped to remove the region proposals, and all regions are reshaped to a fixed size. All regions are categorized with special class specific linear support vector machines (SVMs) using the bounding box regressor. The total of positive and negative numbers is used to determine the region proposals. To produce the bounding boxes, the counted regions are subjected to bounding box regression, which is subsequently filtered with non-maximum suppression (NMS).

Kuang-Wen et al. [1] proposes R-CNN algorithm for object detection On the VOC 2010 dataset, the R-CNN gives a mean average precision (mAPs) of 53.7 percent. It has a mAP of 31.4 percent on the 200-class ILSVRC 2013 object detection dataset, which is a significant improvement over the previous high of 24.3 percent. However, this architecture is difficult to learn and generate test results on a single image from the VOC 2007 dataset takes 49 seconds.

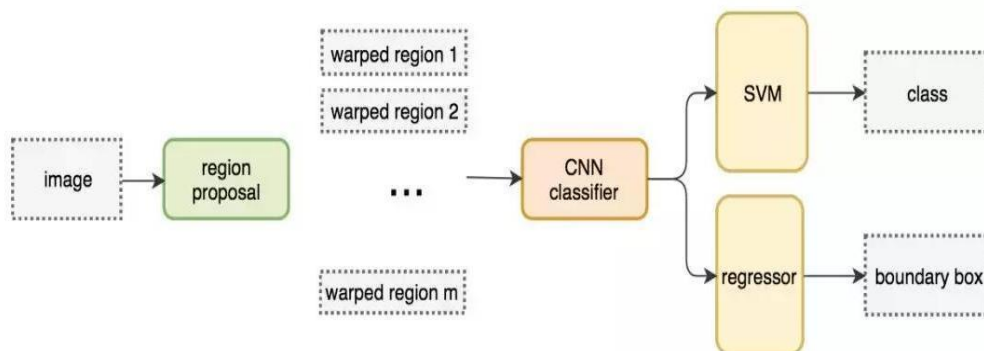


Fig. 3.1 – Architecture of R-CNN

There were major developments and significance when using R-CNN for object detection over manual mathematical approaches, although there are still limitations.

1. It should extract 2K regions for each image using selective search approach, which is a time-consuming and complex process.
2. Using an R-CNN model is slow and expensive. It lasts much time to complete even a small work set for testing, and it requires large storage memory requirement by its characteristics.
3. The use of an R-CNN model is a multi-step process. Initially, region proposals are passed through a convolutional network. The classifier is then substituted by SVMs to adjust with the attributes of convolutional network. Finally, the object is classified using bounding-box regressor. As a result, it becomes a lengthy procedure.
4. Furthermore, to achieve the results in this model, extra boxes are required.
5. Detecting objects in an image on a GPU takes 49 seconds.

Algorithm 2 – Fast Region-Based Convolutional Network (Fast R-CNN).

By including classification and bounding box regression, Girshick was able to overcome the drawbacks of R-CNN and designed a new CNN structure known as fast R-CNN. The following is the gain of the fast RCNN method:

1. It is more exact than R-CNN in locating objects.
2. This method is a one-step approach with little task loss.
3. This technique completely alters the network.
4. Extra memory is not required for storing the calculation.

Ross Girshick has provided overview on Fast R-CNN, the image is processed with a deep convolutional network and max pooling layers in the model fast R-CNN to produce a convolutional image layer with different region proposals. The pooling layer then pulls the features of a fixed-length object from the convolutional layer for each region proposal. The properties of each object are organized into a hierarchy of fully connected layers that yield two common output levels. One output layer creates four selection box positions values for each item, while the second output layer produces SoftMax probability between the object and background values. Fig. 3.2 demonstrates the fast R-CNN structure which combines different parts such as convolutional network, region of interest pooling, and classification layer in a single structure. The Region of interest uses max pooling to transform the object's characteristics into a smaller attribute value which makes the calculation faster [3].

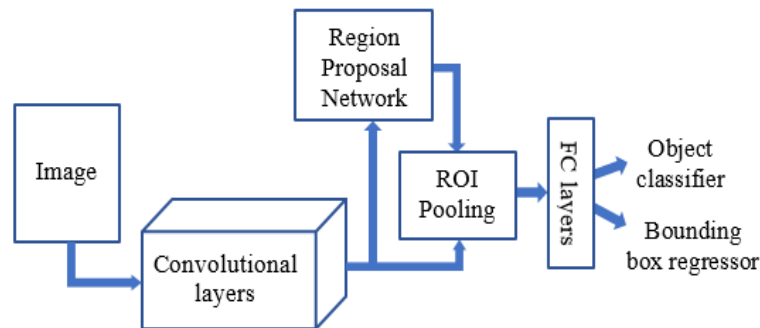


Fig. 3.2 – Architecture of Fast R-CNN

Fast R-CNN drastically improves the training (8.75 hrs vs 84 hrs) and detection time from R-CNN. It also improves Mean Average Precision (mAP) marginally as compare to R-CNN. Problems with Fast R-CNN:

1. Majority of the time taken by Fast R-CNN during detection is a selective search region proposal generation algorithm. As a result, the bottleneck of this architecture which was dealt with in Faster R-CNN.

Algorithm 3 - Faster Region-Based Convolutional Network (Faster R-CNN)

In both the method, R-CNN and fast R-CNN the object is detected by performing selective thorough search. This thorough search is a long process and takes larger time affecting the performance of the object detection method. This selective thorough search technique was a bottleneck in the efficiency of the object finding algorithm. Hence, from Microsoft a research team consisting of haoqing Ren,

Kaiming He, Ross Girshick and Jian Sun, in 2015, developed a faster RCNN object detection system that eliminates the selective search technique. Shaoqing Ren and his colleagues proposed an additional area proposal network in quicker R-CNN, which computes convolutional characteristics from the convolutional network to detect the object instead of searching again for object in the image. There are two sections to the faster R-CNN. The first section contains a fully convolutional neural network that creates region proposal networks, while the second contains a fast R-CNN detector that calculates proposed regions for object classification. This algorithm is a combination of processes for finding the object [4]. Figure 3 depicts the structure of a faster R-CNN.

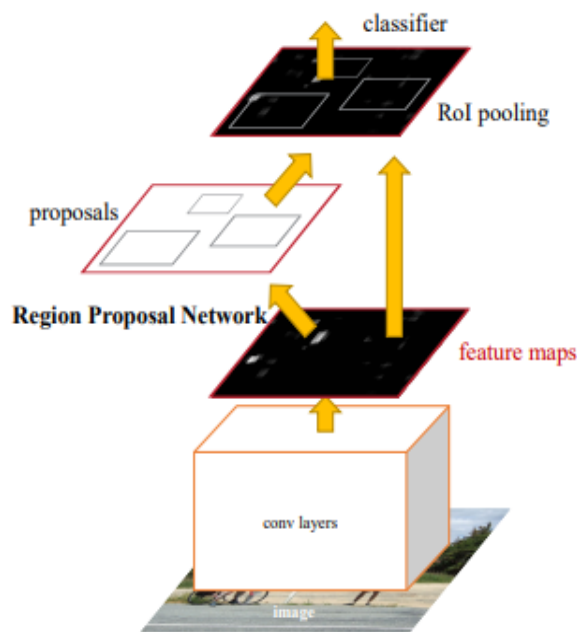


Fig. 3.3 – Architecture of Faster R-CNN

Faster R-CNN outperforms both R-CNN and Fast R-CNN in terms of detection time. The mAP of the Faster R-CNN is likewise higher than that of the preceding two [6].

Algorithm 4 - Single Shot MultiBox Detector (SSD)

The Single Shot MultiBox Detector (SSD) is one of Christian Szegedy's most popular item detection methods, by employing SSD, we are only required to take a single shot to see many objects within the image, but regional proposal network (RPN)-based techniques like the R-CNN series require two shots, one for producing region proposals and the other for policing the content of each proposal. As a result, SSD is much faster than two-shot RPN-based techniques. To better recognize objects of any size, SSD employs a variety of grid sizes rather than just one, which is presented in a distinct way in the SSD paper [5]. SSD outperforms approaches that use an additional object proposal phase in terms of accuracy and speed, while also providing a unified framework for both training and inference. SSD achieves 74.3 percent mAP on the VOC2007 test at 59 frames per second on an Nvidia Titan X, and 76.9% mAP with 512 x 512 inputs, surpassing a comparable state-of-the-art Faster R-CNN model. SSD gives far more accuracy than previous single stage techniques, even with smaller input image sizes. [7].

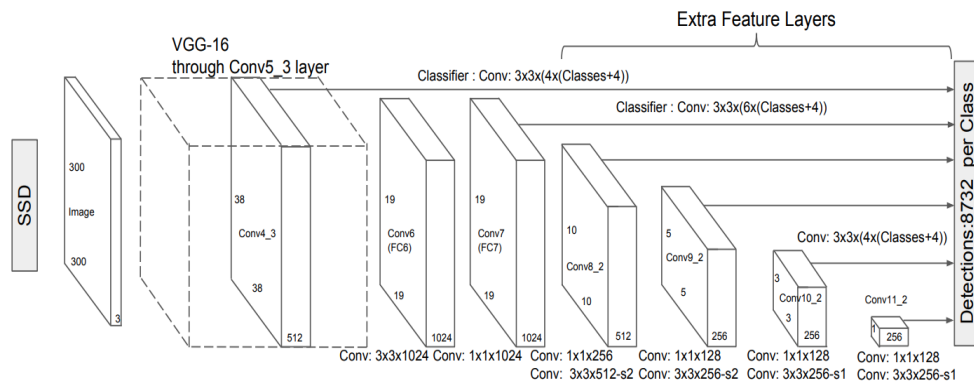


Fig. 3.4 – Architecture of SSD

Algorithm 5 - Random Sample Consensus (RANSAC)

RANSAC stands for Random Sample Consensus. It is the best type of algorithm. It is very basic but really effective and useful. When a dataset contains a large number of outliers (e.g., half of the points, or even more), it is very well suited for fitting models. The RANSAC approach is quite versatile, and it can be utilized in a variety of applications, including curve fitting, classification, state estimation, and a variety of computer vision tasks. The RANSAC algorithm comprised of two steps that are iteratively repeated:

1. From the input dataset, a sample subset containing minimal data items is randomly selected in the first step. Only the elements of this sample subset are used to create a fitting model and the associated model parameters. The sample subset's cardinality is the smallest that allows the model parameters to be determined.
2. In the second phase, the algorithm then checks whatever elements of the entire dataset are consistent with the model instantiated by the estimated model parameters from the first step. If a data element does not fit the fitting model instantiated by the set of estimated model parameters within some error threshold that determines the maximum deviation attributed to the effect of noise, it will be considered an outlier.

A set of observed data values, a method of fitting some form of model to the observations, and some confidence parameters are fed into the RANSAC algorithm. RANSAC achieves its objective by repeating the steps below:

1. Choose a subset of the original data at random. This group is known as the hypothetical inliers.
2. The set of hypothetical inliers is fitted with a model.
3. After that, the fitted model is compared to the rest of the data. The consensus set includes those points that, according to some model-specific loss function, match the estimated model well.
4. The estimated model appears to be reasonable. If the consensus set contains a sufficient number of points.
5. The model can then be enhanced by reestimating it with all of the members of the consensus set.

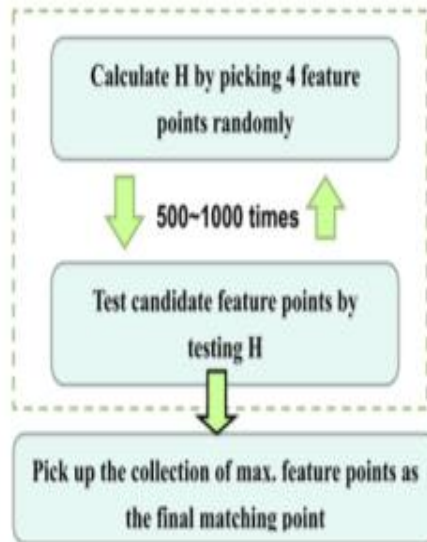


Fig. 3.5 – Flow diagram of RANSAC

Algorithm 6 – You Only Look Once (YOLO v3)

YOLO developed by Joseph Redmon's, is a combined architecture model that is extremely speedy. The YOLO algorithm detects objects in an image at 45 frames per second. It outperforms other methods of detection including R-CNN and SSD. But YOLO v3 is faster than previous YOLO. YOLO v3 runs in 22 ms at 28.2 mean access precision, which is three times faster than SSD. YOLO v3 is also good at detecting small objects in an image. To find the object, YOLO has a set of algorithms called YOLO-based convolutional neural networks [5].

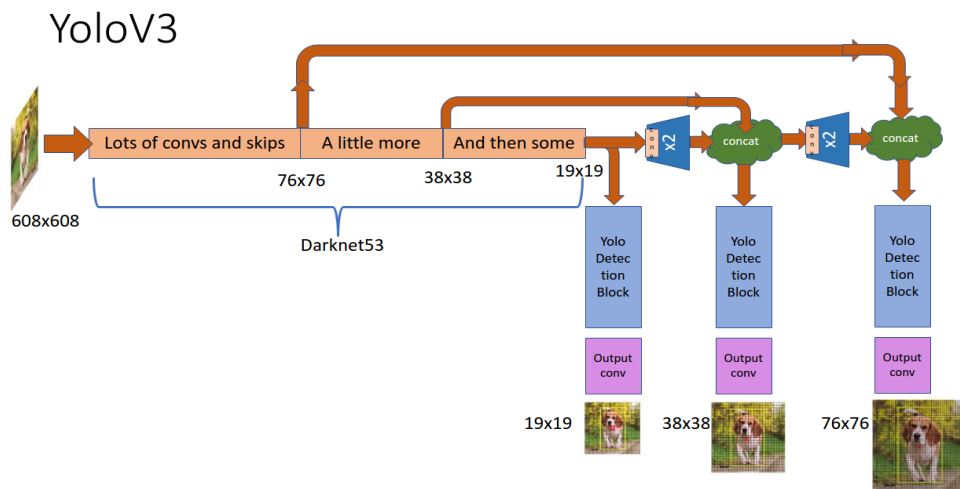


Fig. 3.6 – Architecture of YOLO v3

YOLO v3 is the most recent version. Because it exclusively uses convolutional layers, so it is considered a fully convolutional network. For the implementation of the YOLO v3 algorithm, the Keras library will be introduced in the TensorFlow, which also has several open-source libraries. In YOLO v3, the learned weights are utilized to find the object in the images provided. To form an accurate object detection algorithm, YOLO v3 removes the region proposal approach and combines all processes into one network. The YOLO v3 approach divides the image into a tiny net of cells, with each cell providing selection box offsets and forward convolutional classification of the objects. Following a post-processing step by the algorithm, the bounding boxes are combined to detect the object. If a grid cell is the center of an item, that grid cell is taken into account while locating the object. The exact area of the selection boxes is determined by each grid cell [11].

IV. CONCLUSION

In this paper, several object detection algorithms are discussed and compared, namely R-CNN, fast RCNN, faster R-CNN, single shot detector (SSD), YOLO v3, and others. According to the discussions, the model's speed and accuracy have improved and increased. Fast R-CNN is better than R-CNN, but Faster R-CNN is significantly improved than fast R-CNN. Furthermore, single shot detector outperforms faster R-CNN, whereas YOLO v3 outperforms single shot detector. Prior to the development of YOLO v3, SSD was the best option. However, the most recent greatest approach discovered is YOLO v3, which is far better than SSD and significantly faster than SSD. YOLOv3 is incredibly fast and precise. As a result, we can use Tensorflow to recognize numerous objects faster and add our own images and labels to the datasets using the YOLO v3 model. This YOLO v3 model is advantageous since it can detect object directly and all objects are detected single time only in this mode. The adaption of YOLO v3 algorithm in object detection is very convenient method, because YOLO is one of the popular algorithms in object detection used by researchers around the globe. The base YOLO model processes images in real-time at 45 frames per second, while the smaller version of the network, Fast YOLO processes an astounding 155 frames per second. OpenCV is the popular library for computer vision operations such as object detection, face recognition etc., This OpenCV, and along with the YOLO algorithm implementation using Raspberry Pi makes overall system work efficiently and accurately with high speed.

V. REFERENCE

- [1] Kuang-Wen Hsieh, Bo-Yu Huang, Kai-Ze Hsiao, Yu-Hao Tuan and Fu-Pang Shih "Fruit maturity and location identification of beef tomato using R-CNN and binocular imaging technology" *Journal of Food Measurement and characterization* 15, 5170-5180. doi:10.1007/s11694-021-01074-7, 2021.
- [2] Y. LeCun, Y. Bengio, G. Hinton, "Deep Learning", *Nature*, Vol.521, pp.436-444, 2015.
- [3] R. Girshick, "Fast R-CNN", *IEEE International Conference on Computer Vision (ICCV)*, Santiago, USA pp.1440-1448, 2015.

- [4] Z. Zhao, X. Wu, S. Xu, P. Zheng, “Object Detection with Deep Learning: A Review”, IEEE Transactions on Neural Networks and Learning Systems, Vol.30, Issue.11, pp.3212-3232, 2019
- [5] J. Redmon, A. Farhadi, “YOLOv3: An Incremental Improvement”, ResearchGate, 2018.
- [6] S. Ren, K. He, R. Girshick, J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.39, Issue.6, pp.1137-1149, 2017.
- [7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, A.C. Berg, “SSD: Single Shot Multibox Detector”, Springer, Vol.9905, pp.21-37, 2016.
- [8] J. Y. Lu, C. Ma, L. Li, X.Y. Xing, Y. Zhang, Z.G. Wang, J.W. Xu, “A Vehicle Detection Method for Aerial Image Based on YOLO”, Journal of Computer and Communications, Vol.6, Issue.11, pp.98-107, 2018.
- [9] L. Zhao, S. Li, “Object Detection Algorithm Based on Improved YOLOv3”, Electronics, Vol.9, Issue.3, pp.537, 2020.
- [10] N. Raviteja, M. Lavanya, S. Sangeetha, “An Overview on Object Detection and Recognition”, International Journal of Computer Sciences and Engineering, Vol.8, Issue.2, pp.42-45, 2020.
- [11] A. Kaur, D. Kaur, “Yolo Deep Learning Model Based Algorithm for Object Detection”, International Journal of Computer Sciences and Engineering, Vol.8, Issue.1, pp.174-178, 2020.
- [12] Yi-Qi Huang, Jia-Chun Zheng, Shi-Dan Sun, Cheng-Fu Yang, and Jing Liu 2 “Optimized YOLOv3 Algorithm and Its Application in Traffic Flow Detections”
- [13] K G Shreyas Dixit, Mahima Girish Chadaga, Sinchana S Savalgimath, G Ragavendra Rakshith, Naveen Kumar M R “Evaluation and Evolution of Object Detection Techniques YOLO and R-CNN” International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-2S3, July 2019
- [15] R. Girshick, J. Donahue, T. Darrell, J. Malik, “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”, IEEE Conference on Computer Vision and Pattern Recognition, Columbus, USA, pp. 580-587, 2014.