JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

An Efficient Object Detection technique in Realtime and Noisy Environments

M. Nikhil Babu[1] **Dr. D. Lalitha Bhaskari[2]**

1. M.Tech Scholar, Dept. of CS & SE, Andhra University College of Engineering(A), Visakhapatnam, India, nikhil.maddinenichowdary@gmail.com
2. Professor, Dept. of CS & SE, Andhra University College of Engineering(A), Visakhapatnam, India, lalithabhaskari@yahoo.co.in

Abstract:

During the recent years there has been a great progress in Imaging technology. Image and video capturing devices are becoming cheaper and on the other hand computing power significantly increased with much more efficiency. The arena of Parallel Computing moved through different phases like multi-core processing, Graphical processing unit (GPU), Tensor processing units (TPU) and so on. Several proficient algorithms made the object detection process fine tuned. Also at the same time the processing power involves much cost. As a result, object detection has a high computational cost. We presented an object detection technique that may be executed on a standard CPU in this study. Due to this the computational cost drastically decreases. And at the same time we maintained an effective rate of FPS besides the time consumption to be low. Although we implemented the process computable at CPU level we did not let the accuracy to get deviated.

Keywords: Realtime object detection, Yolo, CPU based object detection.

1. Introduction:

In a few of milliseconds, we can choose out things in our field of vision. In fact if the machine does on behalf of us it would be very comfortable in many aspects. We should be thankful to the technology advancement breakthroughs such as computer vision, machine learning and deep learning.Every object has distinct features that help in classifying to which category it belongs to.To demonstrate, while searching for circles, one seeks items that are a certain distance from a point (i.e. the centre). A similar method is employed for facial recognition, in which the eyes, nose, and mouth are located via mathematical algorithms.

Object detection is a method for locating items inside a picture. If a picture contains a single item and we desire to detect it, this is referred to as image localization. Additionally, if a picture contains numerous items such as a vehicle, bus, or truck, as seen in Figure 1, object detection must be performed in such a manner that the correct thing is recognised while maintaining a high rate of detection. Recent object detection algorithms are lightning quick and very precise. Among the well-researched object detection fields are face detection and pedestrian identification. An extensive range of computer vision applications, including picture retrieval and video monitoring, rely on object detection to function well.

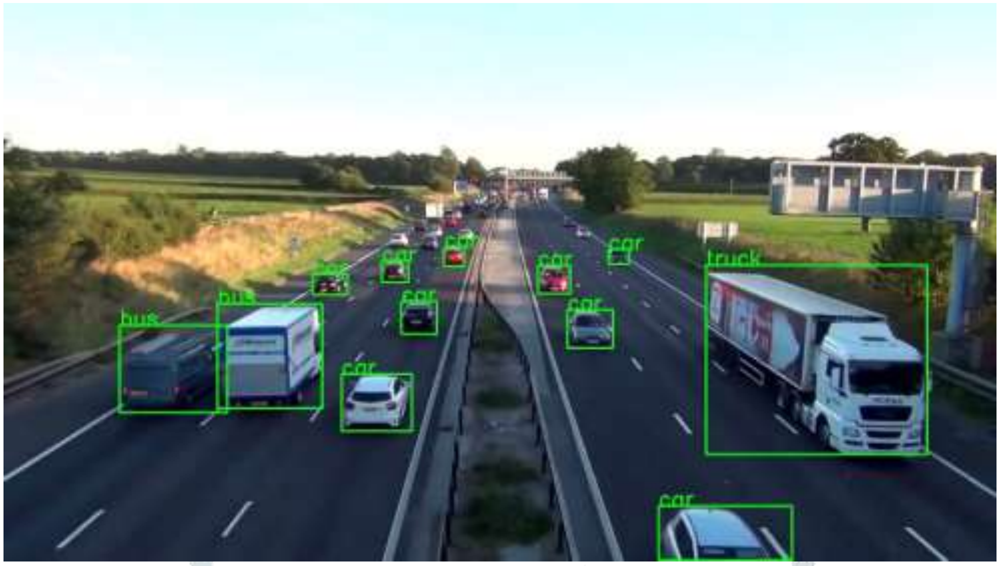

Fig 1: An image where multiple objects exist.

Classifiers and localizers are repurposed in prior detection approaches, which allows for more efficient detection. They use the model to apply it to various locations and sizes on a photograph. Detected areas of the image are those with a high score on the detection scale.

Generally, methods for object identification fall into two categories: neural network-based techniques and non-neural approaches. If you are utilising a non-neural methodology, it may be necessary to first construct features using one of the methods listed below, and then perform classification using a technique such as support vector machine (SVM). On the other hand, neural techniques, which are often based on convolutional neural networks, are capable of doing end-to-end object recognition without the need to define any features in the beginning. (CNN).

Self-driving automobiles are the most sophisticated use of object detection. Vehicles that are capable of moving independently without human intervention. To choose its next action, i.e. whether to accelerate or brake, or turn, a vehicle needs know the position of all things in its vicinity. The automobile can identify items like as other cars, pedestrians, and traffic lights by using Object Detection methods. In accomplishing it object detection plays a vital.

The evolution of object detection algorithms is as follows.

Prior to 2014 – Period of Traditional Object Detection

- DPM (2008), which introduced bounding box regression for the first time
- HOG Detector (2006), a well-known feature descriptor for object recognition in computer vision and image processing
- Viola-Jones Detector (2001), a seminal study that laid the groundwork for the development of conventional object detection algorithms.

After 2014 – Era of Deep Learning Detection

The most essential methods for two-stage object detection

- SPPNetand RCNN (2014)
- Faster RCNNandFast RCNN (2015)
- Pyramid Networks/FPN (2017)
- Mask R-CNN (2017)
- G-RCNN (2021)

The most essential methods for detecting objects in a single step

- SSD (2016)
- YOLO (2016)
- YOLOv3 (2018)
- RetinaNet (2017)
- YOLOR (2021)
- YOLOv4 (2020)

Introducing the object detection and the related concepts in this section, we discussed the related work in next section after which the third section consists of proposed work and results. In the final section we mentioned our conclusions.

2. Related Work:

Lewis et.al.[1], in their paper, proposed a DIY network called as SimpleNet, that performs deep object recognition without pre-processing or deep evaluations that are otherwise very costly. While SimpleNet's accuracy is far lower than that of the state-of-the-art, its power comes from well-designed loss functions with a restricted number of parameters, while other networks receive their power from the depth of the network's layers. The author analysed the performance of many CNN models, including OverFeat, VGG16, Fast R-CNN, and YOLO, against SimpleNet in order to offer the audience with a thorough grasp of the performance of all of these CNN models.

Redmon et.al. [2], in their paper, presented YOLOv3 which is an updated version of their revolutionary network YOLO. When it comes to processing speed and accuracy, this model outscored all other state-of-the-art networks. As a result, it is the best network for doing real-time detections and tracking while maintaining high precision, something that the other networks have failed to accomplish. Additionally, the YOLOv3 is capable of recognising tiny items due to its ability to recognise objects of three distinct sizes successfully.

Ren et al [3], in their paper, presented an updated version of ‘Fast R-CNN’ known as ‘Faster R-CNN’. As the name suggests, the updated version of the Region based Convolutional Neural Network, which showcased better computational speed and accuracy when compared to its previous version and many of the other state-of-the-art networks. The addition of a Region Proposal Network (RPN) has sped up computation by developing features and sharing them with the Detection Network, which is in charge of doing the final detection. While quicker R-CNN models are capable of real-time detection, they have difficulty distinguishing between small and large objects.

Yukui Luo et al. [4] created an OpenCL version of the Deep Convolutional Neural Network, one of the most complex frameworks for deep learning, which is one of the most advanced frameworks for deep learning. Three key advances were made by their framework: a real-time object recognition system, a framework with low power consumption that can be utilised on portable devices, and a framework that can be used on a range of computer platforms. Using the YOLO V2 benchmark, the performance of the framework was compared to that of the CUDA framework to see which performed better.

Alpaydin [5] presented an adaptable fuzzy network architecture that is used in conjunction with Deep Convolutional Neural Networks to produce very effective object detection for long range photos with poor contrast and changing, noisy backgrounds.

Daniel et al. [6] shown how to integrate LiDAR data and RGBD point clouds to develop a 3D Convolutional Neural Network (CNN) architecture known as 'VoxNet' for accurate and efficient item detection using a 3D Convolutional Neural Network (CNN) architecture. This approach was evaluated in real time against publicly available state-of-the-art benchmarks, and the researchers discovered that it outperformed the benchmarks in terms of accuracy while detecting objects in real time.

Girshick et al. [7] developed the 'Fast R-CNN,' which is a Convolutional Neural Network based on a region of interest. At the cost of computer performance, this network can identify objects with great accuracy, but at the price of processing performance. As a consequence, despite its great accuracy, the network has been judged inadequate for real-time item detection and identification applications.

Though the Faster R-CNN is faster than the Fast R-CNN by an order of magnitude, the CNN feature extraction and an expensive per-region computation which are the first and second stages in the Faster R-CNN network, hinder the speed of the network. Addressing this issue, Kim et al [8] made changes in the feature extraction stage by utilizing the cutting-edge technical innovations and presented a newer network known as PVANET. This network is capable of recognising items belonging to numerous categories with the same accuracy as its competitors but incurring a lower computational cost.

Dai et al. [9] created a fully convolutional network termed R-FCN by modifying the existing ResNet network, which is the state of the art in the field of object identification at the time of development. It was decided to replace the completely connected layers in Fast R-CNN with a series of position-sensitive score maps that can additionally store spatial information in order to increase object identification accuracy. As a consequence, R-FCN achieved the same level of accuracy as Faster R-CNN but at a faster processing speed.

The authors presented YOLO, a unified object detection paradigm, in [10]. The model is simple to construct and can be trained on whole pictures without any further steps. YOLO, in contrast to classifier-based approaches, is trained on a loss function that is directly proportional to detection performance, and the whole model is learned at the same time as the loss function. Yolo is the fastest general-purpose object detector currently accessible in the literature, and it is pushing the frontiers of real-time object detection to new heights. Also of note, YOLO is well-suited for generalisation to new domains, making it an excellent choice for applications that need speedy and reliable object recognition.

The authors announced YOLO9000 in [11], a cutting-edge, real-time object identification system capable of detecting over 9000 item types. To begin, the authors made many innovative and incremental enhancements to the YOLO detection approach. On typical identification tasks such as PASCAL VOC and COCO, the upgraded model, YOLOv2, performs at the cutting edge of the field. It is possible to run the same YOLOv2 model at several scales due to a new multi-scale training method, which allows for a simple exchange of one's preference for speed or accuracy. On VOC 2007, YOLOv2 reaches 76.8 mAP at a frame rate of 67 frames per second, which is a good result. In terms of performance, YOLOv2 outperforms state-of-the-art algorithms like as Quicker RCNN with ResNet and SSD, while staying much quicker than these approaches. At the end of the paper, the authors propose a strategy for training on object recognition and classification at the same time. In this manner, the YOLO9000 detection and classification models were both trained at the same time on the COCO and ImageNet datasets. This combined training allows the YOLO9000 to predict detections for item classes that do not yet have tagged detection data available to it. The approach was tested on the ImageNet detection task and found to be effective. Despite having detection data for just 44 of the 200 classes, the YOLO9000 obtains a 19.7 mAP on the ImageNet detection validation set, despite having only 44 classes of detection data. On the 156 classes that are not included in COCO, the YOLO9000 obtains a 16.0 mAP score. On the other hand, YOLO is capable of detecting over 200 different item classes and making predictions for over 9000 different object categories. In addition, it continues to function in real time.

Users of low-configuration PCs may profit from the implementation of CPU Based YOLO, which is a real-time object detection model that can be run on non-GPU computers, as shown by the authors in [12]. There are various improved object identification algorithms available, including "YOLO, Faster R-CNN, Fast R-CNN, R-CNN, Mask R-CNN, R-FCN, SSD, and RetinaNet." "YOLO, Faster R-CNN, Fast R-CNN, R-CNN, Mask R-CNN, R-FCN, SSD, and RetinaNet" are just a few examples. It is a Deep Neural Network-based item identification system that is far quicker and more accurate than the majority of previous object recognition systems. YOLO is designed to work best on GPU-based PCs with a graphics card that has at least 12GB of memory. Yolo has been improved in this model by including OpenCV, which allows real-time object recognition on CPU-based machines to be achieved. This model recognises objects in video at a pace of 10.12–16.29 frames per second and with an accuracy of 80–99 percent on a number of non-GPU-based PCs at a rate of 10.12–16.29 frames per second. YOLO on the CPU achieves a mAP of 31.05 percent.

3. Proposed work and results:

The proposed work is devising a CPU based object detection mechanism using YOLO. The algorithm is as follows.

Algorithm:

Strp 1: Create a Video Streaming Input.
Step 2: Load the model.
Step 3: While Input is available, read the next frame.
Step 4: Score the frame to get labels and coordinates.
Step 5: Plot the boxes over the objects detected.
Step 6: Write the processed frame onto the output video stream.

The performance evolution in terms of FPS could be observed from tables 1 and the comparison of computational requirements is given in table 2.

Table 1: Performance in terms of FPS

Model	**Performance evaluation**			
	Input	**Train set**	**Test set**	**FPS**
YOLOv2	608x608	COCO	COCO	40
YOLOv3	608x608	COCO	COCO	20
		COCO	COCO	5

Table 2: Computational requirements

Device	RAM	Dataset	FPS
Intel® Core™ i3-5010U CPU @ 2.10GHz	4 GB	COCO	5
Intel® Core™ i3-5010U CPU @ 2.10GHz	4 GB	COCO	7.7
AMD Ryzen™ 3 2200G CPU 3.50GHz	8 GB	COCO	16

The object detection from live capture and video streams could be observed from figures 2 to 9.

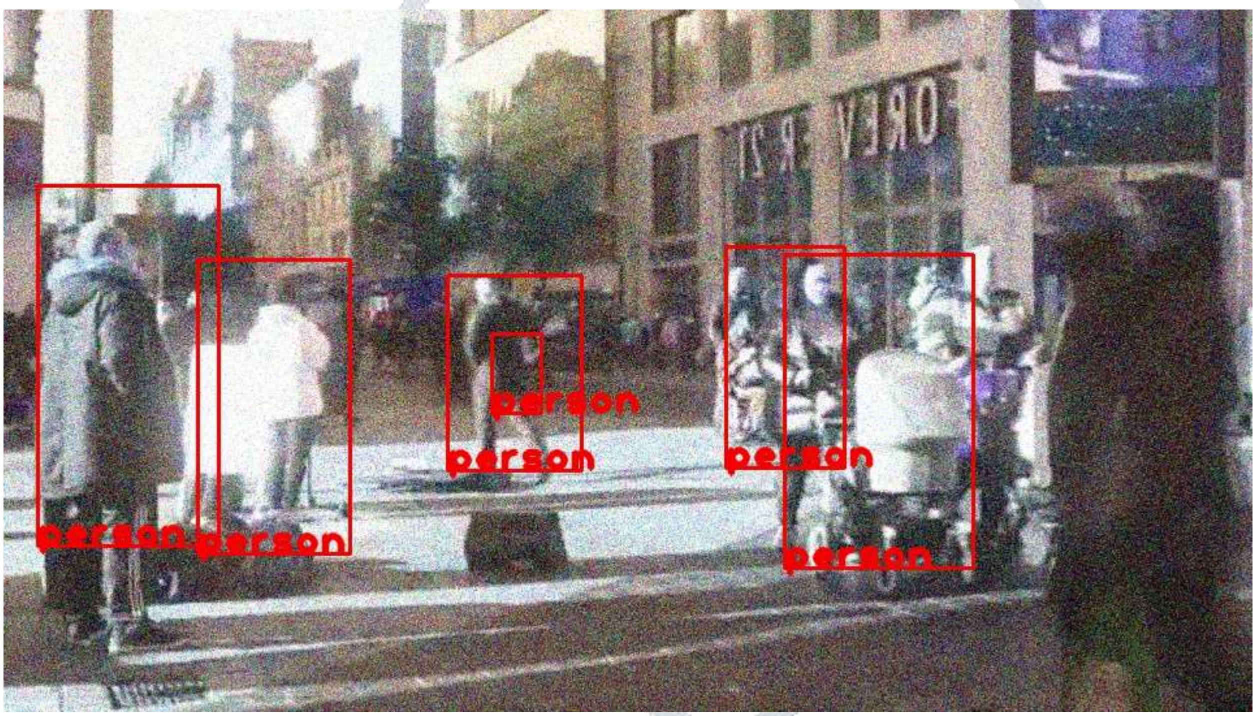

Fig 2: Video Stream Detection Frame 1

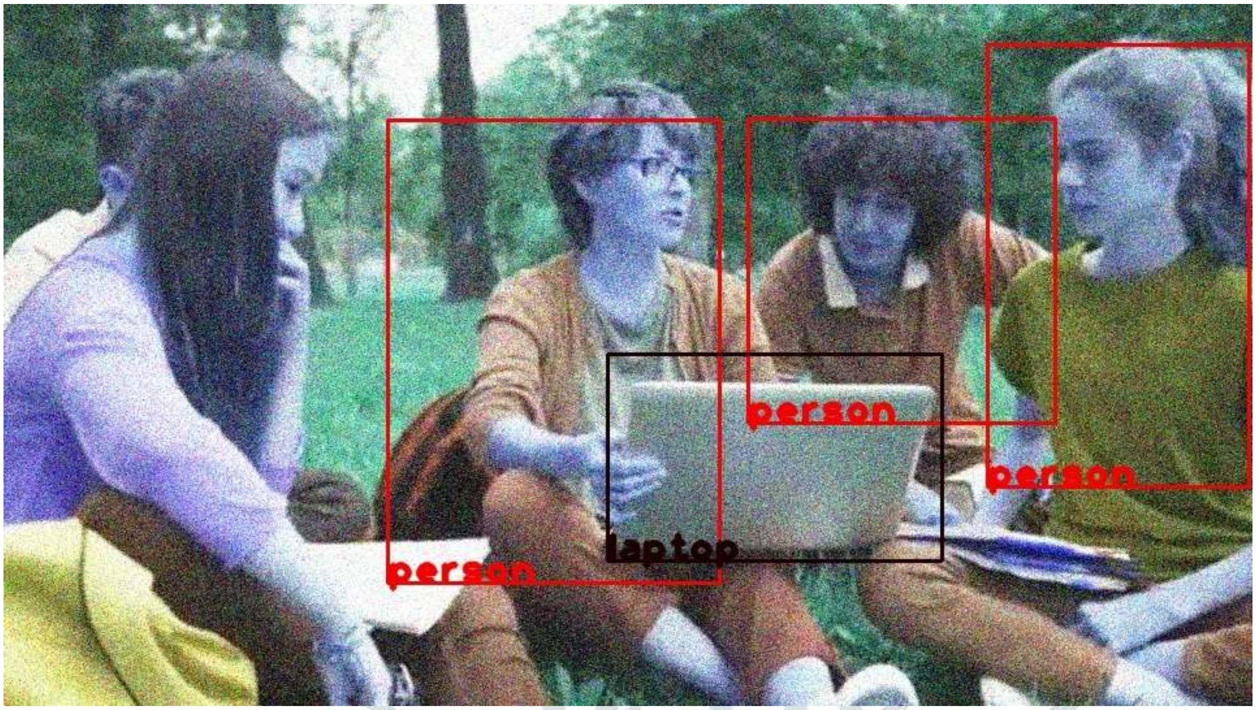

Fig 3: Video Stream Detection Frame 2

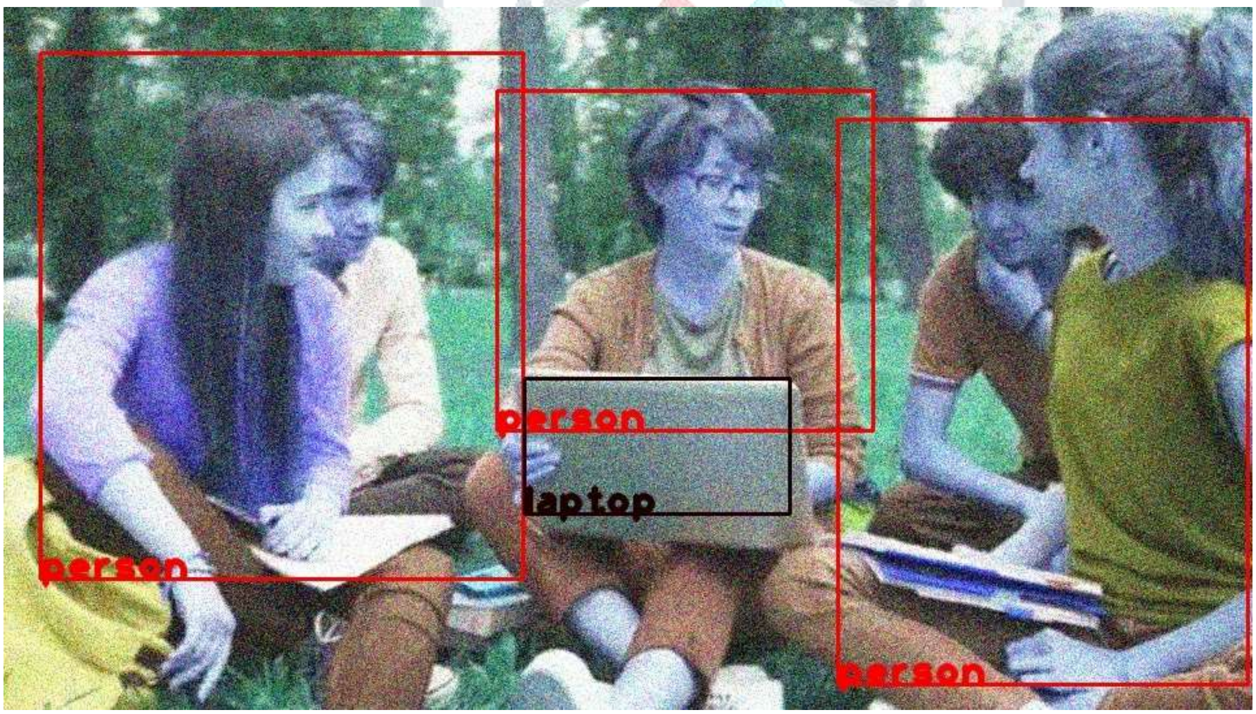

Fig 4: Video Stream Detection Frame 3

Fig 5: Live Camera Detection Frame 1

Fig 6: Live Camera Detection Frame 2

Fig 7: Live Camera Detection Frame 3

Fig 8: Live Camera Detection Frame 4

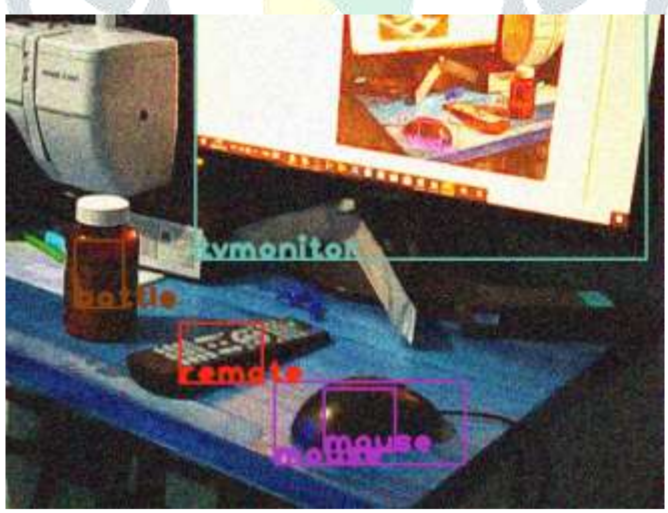

Fig 9: Live Camera Detection Frame 5

4. Conclusion:

There are several Object detection algorithms which are based on different architectures and computational requirements. Although there are various architectures belonging to deep learning which are used widely in object detection resulting in better performance, we made our effort by using CPU Based YOLO in object detection. The advantage of using this mechanism is that using a normal Desktop or Laptop we can accomplish the task of object detection. We are successful in attaining it with accuracy, minimum FPS rate and time. This approach will be very beneficial in a variety of applications, including video surveillance, traffic monitoring, and facial recognition. In the future, we may conduct the tasks using a custom dataset to train the machine and optimise the model in terms of mAP, time, and FPS.

References:

[1] G. Lewis, “Object detection for autonomous vehicles,” 2014.

[2] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” arXiv preprint arXiv:1804.02767, 2018.

[3] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in Advances in neural information processing systems, 2015, pp. 91–99.

[4] Y. Luo, S. Li, K. Sun, R. Renteria, and K. Choi, “Implementation of deep learning neural network for real-time object recognition in opencl framework,” in 2017 International SoC Design Conference (ISOCC). IEEE, 2017, pp. 298– 299.

[5] G. Alpaydin, “An adaptive deep neural network for detection, recognition of objects with long range auto surveillance.” in ICSC, 2018, pp. 316–317.

[6] D. Maturana and S. Scherer, “Voxnet: A 3d convolutional neural network for real-time object recognition,” in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015, pp. 922–928.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[8] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, and M. Park, “Pvanet: Deep but lightweight neural networks for real-time object detection,” arXiv preprint arXiv:1608.08021, 2016.

[9] J. Dai, Y. Li, K. He, and J. Sun, “R-fcn: Object detection via region-based fully convolutional networks,” in Advances in neural information processing systems, 2016, pp. 379–387.

[10] Redmon, Joseph & Divvala, Santosh & Girshick, Ross & Farhadi, Ali. (2016). You Only Look Once: Unified, Real-Time Object Detection. 779-788. 10.1109/CVPR.2016.91.

[11] Redmon, Joseph & Farhadi, Ali. (2017). YOLO9000: Better, Faster, Stronger. 6517-6525. 10.1109/CVPR.2017.690.

[12] M. B. Ullah, "CPU Based YOLO: A Real Time Object Detection Algorithm," 2020 IEEE Region 10 Symposium (TENSYMP), 2020, pp. 552-555, doi: 10.1109/TENSYMP50017.2020.9230778.