International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 07, July-2023

Real Time Object Detection based on YOLOv3 using Python

Dr.J. Malla Reddy
Dept. of Computer Science and Engineering
Mahaveer Institute of Science and Technology
Hyderabad, India

M. Kamal Nadh
Dept. of Computer Science and Engineering
Mahaveer Institute of Science and Technology
Hyderabad, India

M. Harsha Vardhan
Dept. of Computer Science and Engineering
Mahaveer Institute of Science and Technology
Hyderabad, India

N. Priyanka
Dept. of Computer Science and Engineering
Mahaveer Institute of Science and Technology
Hyderabad, India

Abstract— **Object Detection is one of the stimulating tasks in the applications of Computer Vision. Computer Vision is an AI discipline, empowering computers and systems to extract significant insights from digital images, videos, and various forms of visual data. Object detection is the identification of objects by the help of properties like size ,shape, color etc. in our environment .It is gaining a lot of attention in many real time applications such as surveillance, self-driving cars, detection of number plates at the traffic signals, detection of vehicles in the parking slot and detection of animals in agriculture farm etc. . It can recognize objects in different formats like images, video, and live stream. In this presentation we are using the YOLO algorithm which uses Convolutional Neural Network which can process Single forward direction. In the implementation we are creating and designing YOLOv3 model for the detection of objects. You Only Look Once (YOLO) based model comes under the Deep Learning approaches. This model contains three main files: COCO (names dataset), YOLO (configuration file) and WEIGHTS (measurement dataset). These files are added to python modules through the present directory. The YOLO model can present better computer vision and give the best output through images, videos, or live stream.**

Keywords—Object Detection, YOLO, Convolutional Neural Network, YOLOv3 model, COCO dataset, WEIGHTS dataset, Neural Network, image, video, live stream, objects.

I. INTRODUCTION

In the realm of computer vision, object detection refers to the process of recognizing and precisely localizing objects within images or videos. It is an important part of many applications, such as surveillance, self-driving cars, or robotics and it also includes detecting people, cars, chairs, stones, and buildings. By utilizing abundant annotated visuals during the training phase, object detection models are equipped to effortlessly process new data, allowing users to input visuals and receive output visuals with comprehensive annotations.

A. Computer Vision

Computer vision, an AI discipline, empowers computers and systems to extract valuable insights from digital images, videos, and various visual inputs, enabling them to make informed decisions or provide recommendations based on the derived information. While AI grants computers the ability to think, computer vision equips them with the power to visually perceive, observe, and comprehend their surroundings.

By training machines to fulfill these functions, computer vision enables a system to efficiently examine products or monitor production assets, rapidly analyzing thousands of items or processes per minute. This allows for the detection of imperceptible defects or issues that may surpass the capabilities of human observers.

B. YOLO

YOLO is short form of "You Only Look Once" which refers to an algorithm that performs real-time object detection and recognition in images. In YOLO, object detection is approached as a regression problem, yielding the probabilities of different classes for the detected objects.

Utilizing convolutional neural networks (CNN), the YOLO algorithm excels in delivering real-time object detection. True to its name, this algorithm accomplishes object detection by performing a single forward propagation through a neural network, eliminating the need for additional iterations.

This paper presents the development of YOLOv3 model using Python libraries and YOLO datasets. The organization of paper as follows. Section 2 describes the related work focusing on prior work on Object Detection based on YOLOv3. Section 3 states the Existing System of Object Detection and traditional approaches. Section 4 presents the Proposed System of Object Detection. Section 5 exposes the System Architecture of Object Detection. Section 6 presents System Implementation of Object Detection. Finally, concluded with future extension in Section 7 and Section 8.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

II. Literature Survey

In the realm of engineering and scientific research, the relentless pursuit of innovation and progress has been the driving force behind the development of many concepts regarding object detection. These endeavors owe their success not only to the tireless efforts of engineers and scientists but also to their collective contributions in shaping and advancing key concepts within their respective fields.

Dumitru Erhan [1], “Scalable Object Detection using Deep Neural Networks”. We used a real-time object detection algorithm, YOLO, to train our machine learning model for object detection. Erhan, D et al., 2014[2] proposed research on object detection using neural networks. In the object detection method, a deep convolutional network was used, whereas region based conventional network increases the accuracy of the network and decreases the time.

Eric. K. W 2019 [3] techniques have brought much more easiness to train large and deeper networks and shown enhanced performance. Recent advancements have introduced novel techniques that leverage deep convolutional neural networks (DCNN), such as Girshick, to efficiently detect vehicles and other objects from both videos and static images The Faster R-CNN algorithm[4]suggests potential regions as object candidates and verifies their validity using CNN. On the other hand, YOLO employs an end-to-end unified and fully convolutional network architecture that simultaneously predicts the presence of objects and their corresponding bounding boxes across the entire image.

Han, C., Liu et al., 2018[5] proposed research on a novel approach to identify the shape of the clustered image. The central concept of image transformation lies in converting a queried shape into a cluttered image. PAD (Pyramid of arc length descriptor), a point-based approach, is utilized to establish correspondences between the queried shape and the local shape image."YOLOv3: An Incremental Improvement" by Joseph Redmon and Ali Farhadi [6] discusses how YOLOv3 is an incremental improvement over earlier versions of the algorithm.

In 2015, Kamate conducted a study focusing on the utilization of unmanned moving vehicles (UAVs) to safeguard the United States from illegal border crossings by tracking and detecting moving objects[7]. The significance of UAVs extends beyond border security, playing a vital role in various industries. However, detecting moving objects poses a challenging task. Several methods, including histogram-oriented gradients and background subtraction, have been employed for object detection. Latharani, (2011).[8] Object detection achieved excellent performance in computer vision that can be described from the following four aspects: bottom feature extraction, feature coding, feature aggregation, and classification.

In comparison to YOLO, D. G. Lowe's SSD (Single Shot MultiBox Detector) [9] surpasses it by dividing the space for bounding boxes into avoidance boxes across various feature ratios and scales per feature map location, resulting in improved performance. The study conducted by He. K., Zhang et al., 2016[10] presents an investigation into both classical and deep learning approaches for object detection. The research primarily emphasizes the operational principles of the model and its real-time performance and accuracy in detecting objects. The challenges of object detection are based on the deep learning solutions.

Zhang Li, (2018, March) [11] proposed research on the object detector based on deep learning of small samples. The deepest learning and YOLO structure were used in a deep convolution neural network. Zheng Y [12] The technique represents the traversing of the post preserving of the graph and that will improve the computational time of the graph. Such algorithm is tested through a specific database and that demonstrates the two problems which are object class recognition and similar image retrieval.

III. Existing System

Object Detection exist multiple approaches to implement object detection, including fast R-CNN (Regional Convolutional Neural Network), Retina Net, and Single-Shot MultiBox Detector (SSD). These traditional approaches successfully address the challenges associated with data limitations and modeling in object detection. However, they typically rely on multiple algorithms to detect objects rather than a single unified solution.

By using a single pass of the input image, single-shot object detection can make predictions about the presence and placement of objects within it. By processing the entire image in a single pass, it demonstrates computational efficiency. However, it is important to note that single-shot object detection methods generally exhibit lower accuracy compared to alternative approaches and may be less effective in detecting smaller objects.

The technique of two-shot object detection employs a two-pass strategy on the input image to generate predictions concerning the existence and spatial properties of objects. In the first pass, a set of proposals or potential object locations is generated, while the second pass refines these proposals and provides final predictions. This approach offers higher accuracy compared to single-shot object detection, but it does come at the cost of increased computational complexity.

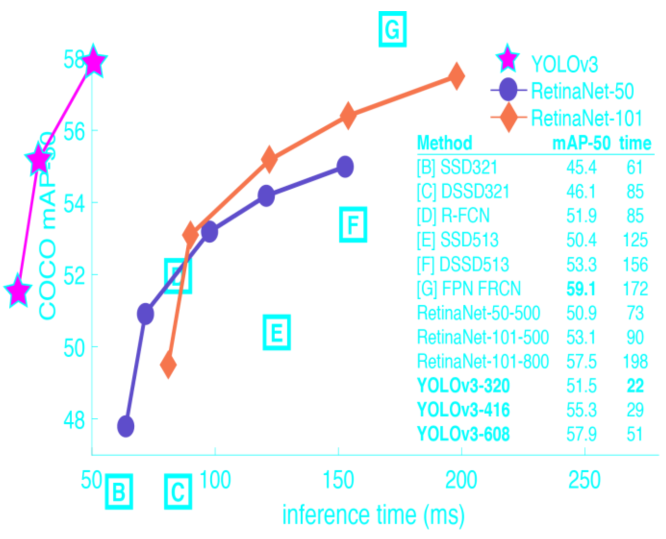
R-CNNs, as an early instance of deep learning-based object detectors, exemplify a two-stage detection approach. While R-CNNs exhibit remarkable accuracy, their major drawback lies in their speed. In the past, R-CNN networks suffered from slow performance, achieving a mere 5 frames per second (FPS) on a GPU.

IV. Proposed System

In our proposed system, we aim to enhance the speed of deep learning-based object detectors by adopting a one-stage detection strategy that combines the strengths of Single Shot Detectors (SSDs) and YOLO. Typically, single-stage detectors sacrifice some accuracy compared to two-stage detectors but offer notable gains in terms of speed.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 07, July-2023



Method	mAP-50	time
[B] SSD321	45.4	61
[C] DSSD321	46.1	85
[D] R-FCN	51.9	85
[E] SSD513	50.4	125
[F] DSSD513	53.3	156
[G] FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	**22**
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51

Fig. 1: Performance of YOLO

The You Only Look Once (YOLO) approach introduces a neural network structure that predicts bounding boxes and class probabilities simultaneously, achieving an end-to-end solution in a single pass. YOLO serves as a prime illustration of a single-stage object detector. YOLO is an algorithm that uses convolutional neural networks (CNN) to provide real-time object detection. Which can represent the single stage propagation for detecting objects in a single image or frame.

While algorithms like Faster RCNN work by detecting possible regions of interest using the Region Proposal Network and then performing recognition on those regions separately, YOLO performs all its predictions with the help of a single fully connected layer. You Only Look Once object detector capable of super real-time object detection, it is very fast and obtaining 45 FPS on a GPU.

V. System Architecture

Designed specifically for real-time object detection, YOLOv3 (You Only Look Once, Version 3) algorithm possesses the ability to detect and classify distinct objects within videos, live streams, or images. By leveraging a deep convolutional neural network, the YOLO machine learning algorithm utilizes learned features to effectively detect objects.

In this architecture, we can explore the YOLO algorithm, which encompasses the methodology of neural networks. The initial step involves partitioning the data into multiple grids and generating object boundary boxes. Each boundary box consists of five elements: (x, y, w, h) along with a box confidence score.

The confidence score indicates the likelihood of the box containing an object and the accuracy of the boundary box. Next, the bounding box width (w) and height (h) are normalized by the image's width and height. The values of x and y represent offsets relative to the corresponding cell. Consequently, all values of x, y, w, and h fall within the range of 0 to 1.

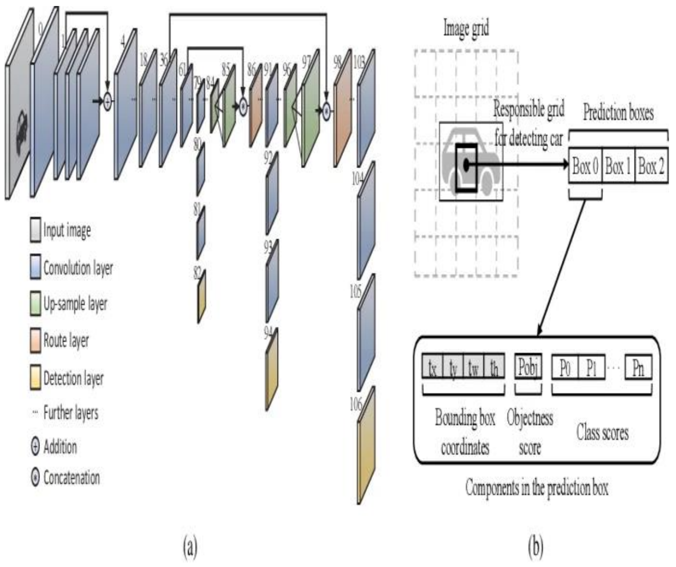


Fig. 2:YOLOv3 Architecture

The conditional class probability corresponds to the likelihood that an identified object belongs to a specific category. Each grid cell contains a set of conditional class probabilities, with each category having its own probability value. For instance, in YOLO, each cell encompasses 20 conditional class probabilities. The shape of a grid cell is determined as C + B * 5, where C represents the number of classes and B represents the number of predicted bounding boxes. The multiplication of B by 5 arises from including the (x, y, w, h, confidence) values for each box. Given that there are S × S grid cells in an image, the minimum grid divisions for input data in YOLO are typically 19x19. The model's overall prediction is a tensor of shape S × S × (C + B * 5). For example, YOLO's prediction may have a shape of (S, S, B×5 + C) = (5, 5, 2×5 + 20) = (5, 5, 30).

$$\text{Intersection Over Union (IOU)}=\frac{AREA\ OF\ INTERSECTION}{AREA\ OF\ UNION} > 0.5$$

To generate the ultimate prediction, select the predictions with box confidence scores exceeding 0.50 and consider them as the final predictions.

VI. System Implementation

In the system we are implementing the input data to be processed using python code. It can test the data, evaluate the data, and determine the accuracy. In this implementation we are considering some modules to determine the object detection and we are using some predefined datasets and python libraries.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

A. *Static Object Detection*

In this unit the static object represents the unmovable or motionless object. The detection procedure and its implementation can be done in the format of Image. In theory, object detection involves several key phases, including recognition, localization, tracking, and extracting object information. During the process, the image is partitioned into multiple grids, and bounding boxes are generated for the input data. Whenever an object is detected, it is visually indicated by a bounding box. Subsequently, the predictions of each grid cell are computed based on the intersection and union areas. This calculation helps refine the output, resulting in a more accurate and refined representation of the detected objects. To implement python programming, we are using predefined datasets which can represent the YOLO algorithm. So, we are datasets and an image as input data that can be configured by the python libraries and can get good output.

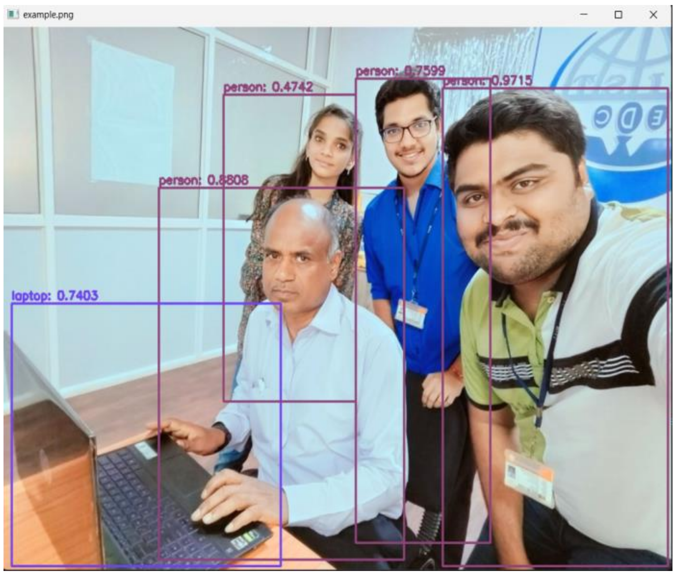


Fig. 3:Static Object Detection

B. *Moving Object Detection*

In this unit the Moving object represents the shifting or movable object. Detection procedure and its implementation can be done in format of Video. In the realm of object detection, there are distinct phases involved, including recognition, localization, tracking, and extracting object information. In the process, videos are divided into multiple frames, and bounding boxes are generated for each frame as part of the input data. Whenever an object is detected, a bounding box is displayed around it, indicating its location. By calculating predictions for each frame using intersection and union areas, a refined and accurate output is obtained. Notably, when considering the Graphical Processing Unit (GPU), R-CNN achieves a frame rate of 5 frames per second (FPS), whereas YOLO operates at a significantly faster rate of 45 FPS. To implement python programming, we are using predefined datasets which can represent the YOLO algorithm. So, we are datasets and an image as input data that can be configured by the python libraries and can get good output.

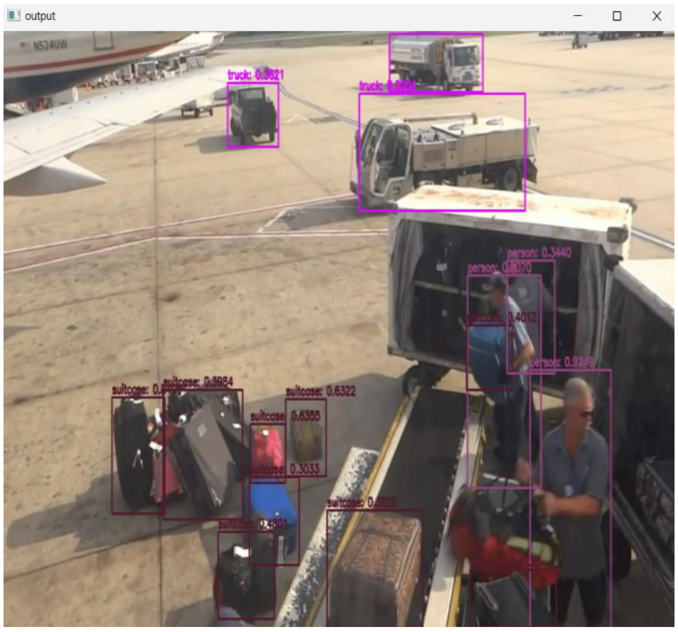


Fig. 4: Frame 1

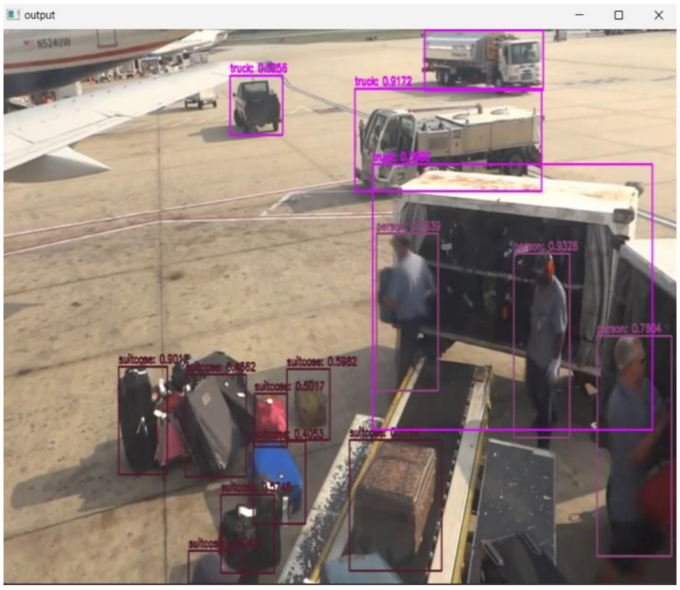


Fig. 5: Frame 2

C. *Live Object Detection*

In this unit the Live object represents the shifting or movable object. It is very similar to the Moving Object Detection Module. The Detection procedure and its implementation can be done in the format of Livestream or webcam. The concept of object detection can be theoretically defined by various phases such as recognition, localization, tracking, and extracting object information. In the process, the live streaming screen is divided into multiple frames, and bounding boxes are applied to identify objects within each frame. When an object is detected, it is visually represented by a bounding box. By evaluating predictions for each frame using intersection and union areas, a refined output is obtained. Notably, considering the Graphical Processing Unit (GPU), R-CNN achieves a frame rate of 5 frames per second

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

(FPS), whereas YOLO operates at a significantly higher frame rate of 45 FPS. To implement python programming, we are using predefined datasets which can represent the YOLO algorithm. So, we are datasets and an image as input data that can be configured by the python libraries and can get good output.

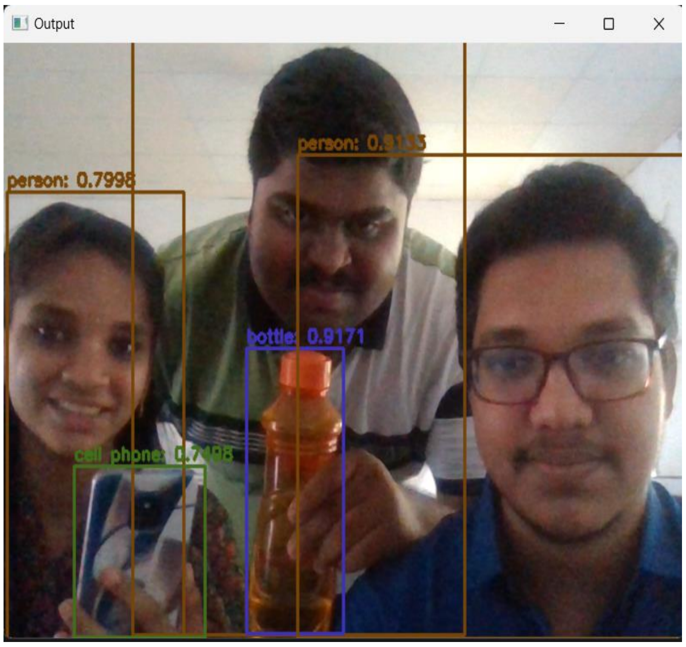


Fig. 6: Live Object Detection

D. *Predefined Datasets*

- COCO Dataset (Names dataset): The Common Objects in Context (COCO) database serves as a valuable resource for advancing research in areas such as object detection, instance segmentation, image captioning, and person key points localization. It is a comprehensive dataset designed to facilitate diverse studies within the field. COCO encompasses a vast collection of annotated images, allowing for large-scale object detection, segmentation, and captioning tasks. This extensive data set comprises various object categories, including but not limited to people, cars, buses, cats, dogs, bottles, and more. The dataset can be represented as (. names) i.e., coco.names .
- WEIGHTS Dataset: It is the predefined dataset which includes all outputs and properties after the model has been trained . It contains properties of each object like shape ,size ,colour ,height ,width etc. The dataset can be represented as (.weights) i.e., yolo.weights .
- YOLO Configuration File: This configuration file includes network definitions, hyper-parameters, anchor settings and describes the layout of the network, block by block. It is the configuration file which represents YOLO algorithm, and it can be configured to the python libraries. It can be represented as (.cfg) i.e., yolo.cfg.

E. *Python Libraries*

- OpenCV**:** OpenCV is a vast open-source library, holds significant importance in the realm of computer vision, machine learning, and image processing. In modern systems, it plays a crucial role by enabling real-time operations. Leveraging OpenCV, individuals can effectively analyse images and videos to detect objects, recognize faces, and even decipher human handwriting. To install this library, we use command as " pip install opencv-python"**.** We can import this library in python programming this can be configured to the YOLO predefined datasets.
- NumPy**:** Generally, NumPy is a Python library used for working with arrays and memory allocation but in object detection NumPy reshape and transpose to avoid unnecessary memory allocation and can reduce the memory. To install this library, we use command as " pip install numpy" . We can import this library in python programming this can be configured to the YOLO predefined datasets.
- Imutils: This library offers a set of handy functions designed to simplify fundamental video processing tasks, including translation, rotation, resizing, skeletonization, and displaying. Primarily utilized for video and live streaming applications, it facilitates frame-by-frame analysis for object detection purposes. To install this library, we use command as " pip install imutils" . We can import this library in python programming this can be configured to the YOLO predefined datasets.

VII. Conclusion

In this project, presenting the Object Detection methodology developed by YOLOv3 algorithm which is based on Convolutional Neural Network (CNN). There are several approaches available for object detection, including R-CNN, fast R-CNN, faster R-CNN, YOLO (You Only Look Once), and SSD (Single Shot Detector). These approaches, while offering high accuracy, often suffer from slower detection speeds. The R-CNN family of networks faced challenges with their speed, achieving a modest 5 frames per second (FPS) on a GPU.

To overcome the speed of detection with normal accuracy the YOLO has been considered the best algorithm for problem solving which can compute the problems with Single Shot Detection. You Only Look Once object detector capable of super real-time object detection, it is very fast and obtaining 45 FPS on a GPU. We are developing a YOLOv3 model for implementing python programming and considering the inputs as predefined datasets and sample images, videos, and livestreaming. After importing the python libraries those are configured to datasets which detect the objects in input

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

data and gives best output. The results of this study are developing a model that can increase the speed, accuracy, and reduce the time in object detection.

VIII. Future Work

YOLO is the best at generalizing Object representation compared with other object detection models and can be recommended for real-time environment. We have different versions of YOLO like YOLOv1 , YOLOv2,YOLOv3.The upcoming versions can be developed by Artificial Intelligence it should consider the detection of smaller objects and multiple scales. It also should consider the high accuracy in object detection. YOLO should develop updated datasets for the objects and accurate information.

Acknowledgement

We sincerely acknowledge the Department of Computer Science and Engineering (R&D), Mahaveer Institute of Science and Technology for their resources, support, guidance, and cooperation during the period of project development.

References

[1] Dumitru Erhan “Scalable Object Detection using Deep Neural Networks”, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 237-248.

[2] Erhan, D et al (2014). Scalable object detection using deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition vol 3, no 4,pp. 2147-2154.

[3] Eric. K. W, Applied Science has developed an innovative method called deep fusion feature-based approach, which significantly boosts the efficiency of YOLO in detecting objects in high-resolution optical remote sensing images., vol. 34, 2019.

[4] Girshick. R., In the Proceedings of the IEEE International Conference on Computer Vision, the paper "Fast r-CNN" presents a method for object detection., pp. 1440–1448, Berlin, Germany, 2015.

[5] Han, C., Liu (2018). TransHist: Occlusion-robust shape detection in cluttered images. Computational Visual Media, vol4,no. 2,pp. 161-172.

[6] Joseph Redmon: At the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), the paper "Better, Faster, Stronger" introduces advancements in object detection. 2017, pp. 7263-7271.

[7] Kamate, S.,(2015). Application of object detection and tracking techniques for Procedia Computer Science features an article highlighting the application of object detection and tracking techniques specifically tailored for unmanned aerial vehicles (UAVs). vol61,no.3, pp. 436-441

[8] Latharani, (2011). Various object recognition techniques for computer vision. Journal of Analysis and Computation, vol7, no. 1,pp. 39-47

[9] Lowe. D. G., The International Journal of Computer Vision presents a study focusing on the extraction of distinctive image features through scale-invariant key points., vol. 60, no. 2, pp. 91–110, 2004.

[10] Zhang He, K., (2016). Deep residual learning for image recognition. The paper is published in the conference proceedings of the IEEE conference on computer vision and pattern recognition. vol 2, no.3, pp. 770-778.

[11] Zhang Li, (2018, March). The Tenth International Conference on Advanced Computational Intelligence (ICACI), organized by IEEE, explores the utilization of deep learning techniques for object detection, even with limited training samples.2018,vol2, no.3, pp. 449-454.

[12] Zheng Y .,“Towards a deep learning framework for unconstrained face detection,” in Proceedings of the 2016 IEEE 8th International Conference on BiometricsAeory, Applications and Systems (BTAS), IEEE, New York, NY, USA, pp. 1–8, 2016.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)