IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 13, No. 1, March 2024, pp. 827~838
ISSN: 2252-8938, DOI: 10.11591/ijai.v13.i1.pp827-838

You only look once model-based object identification in computer vision

Shiva Shankar Reddy[1], Venkata Rama Maheswara Rao[2], Priyadarshini Voosala[1], Silpa Nrusimhadri[2]
[1]Department of Computer Science and Engineering, Sagi Rama Krishnam Raju Engineering College (A), Bhimavaram, India
[2]Department of Computer Science and Engineering, Shri Vishnu Engineering College for Women (A), Bhimavaram, India

Article Info

Article history:

Received Jul 2, 2023
Revised Oct 1, 2023
Accepted Nov 7, 2023

Keywords:

Convolutional neural network
Deep learning
Feature extraction
Machine learning
You only look once

ABSTRACT

You only look once version 4 (YOLOv4) is a deep-learning object detection algorithm. It is used to decrease parameters and simplify network structures, making it suited for mobile and embedded device development. The YOLO detector can foresee an object's Class, bounding box, and probability of that Object's Class being found inside that bounding box. A probability value for each bounding box represents the likelihood of a given item class in that bounding box. Global features, channel attention, and special attention are also applied to extract more compelling information. Finally, the model combines the auxiliary and backbone networks to create the YOLOv4's entire network topology. Using custom functions developed upon YOLOv4, we get the count of the objects and a crop around the objects detected with a confidence score that specifies the probability of the thing seen being the same Class as predicted by YOLOv4. A confidence threshold is implemented to eliminate the detections with low confidence.

Corresponding Author:

Shiva Shankar Reddy
Department of Computer Science and Engineering, Sagi Rama Krishnam Raju Engineering College
Bhimavaram, West Godavari (District), Andhra Pradesh, India
Email: shiva.srkr@gmail.com

1. INTRODUCTION

The computer vision enhancement identifies and locates at least one compelling target from still picture or video information. Image processing, pattern acknowledgment, and artificial intelligence (AI) are among the techniques covered. The technology has a wide range of potential uses, including traffic management, accident prediction, crowd analysis, detection of dangerous substances in factories, optical character recognition, autonomous vehicles, facial and iris recognition for verification, robotics, object tracking and counting, monitoring restricted military areas, and advanced human-computer collaboration [1]. Given the intricate and unpredictable nature of identifying many target application scenarios, achieving an optimal balance between accuracy and computational costs in real-world situations is challenging. Several approaches have been proposed to overcome this problem, notably using computer vision and deep learning methodologies [2].

You only look once (YOLO) is a realtime item identification system. It recognizes objects faster and more accurately. It can estimate up to 80 kinds of visible and invisible objects [3]. The realtime recognition system could frame a confined-edge box around adjacent items, recognize numerous things from a single picture, and be quickly taught and deployed in a production system. It improves, speeds up, and adapts computer vision algorithms by advancing object detection research. In Figure 1, the objects belonging to different categories, like person and train, are detected along with their confidence score. You only look once

version 4 (YOLOv4) surpasses prior methods in terms of detection accuracy, performance, and speed [4]. It is a rapidly functioning component that can be readily identified and taught for use in industrial processes.

In addition to identifying different objects, they are trimmed and stored in their respective directory. The individual and the locomotive shown in Figure 2 have been trimmed and stored inside a designated directory. The major objective was to enhance the efficiency of the neural network detector for simultaneous computations. Additionally, it encompasses a range of potential designs and architectural decisions, taking into account the impact on the performance of different detectors, as suggested by previous YOLO models [5]. This technique predicts the classes and bounding boxes for the whole image instead of picking an attractive region of interest (ROI), allowing for faster detection.

Figure 1. Object detection along with confidence score

Figure 2. Cropping of different objects present in the image

2. RELATED WORK

Yang *et al*. [1] suggested AIRCRAFT-YOLOv4 object localization computation that replaces the unit convolution with a depth-wise separable convolution. They examined the AIRCRAFT-YOLOv4 computation using the UCAS-AOD dataset. Aircraft YOLOv4 can recognize airplanes in remote sensing photos with 86.92% mAP and 29.62 FPS. The model indicated that Aircraft-YOLOv4 is better for military remote sensing picture aircraft object locating jobs because of its high speculation. Kumar *et al*. [2] used tiny YOLO v4 with a spatial pyramid pooling (SPP) module to construct a face coverings identification network model. They assured that the provided network model is ready for precise cloak area on the face district, increasingly reconnaissance applications where the detectable quality of the entire face locality is a criterion. Wang *et al*. [3] trained and tested on Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI) and berkeley deep drive (BDD) datasets to reach their objective. They employed a YOLOv4-based object recognition computation with a single step to boost identification accuracy and maintain support for continuing action. Content security policy (CSP) structure in highlight fusion and the remote frame buffer (RFB) module in the realtime object detector increases accuracy. Realtime object detector has 92.5% accuracy for KITTI and 93.01% for BDD. Lee *et al*. [4] utilized multiple object tracking (MOT) 17-05. The transparencies object identification model considers the video stream's features and designs a low-overhead scheduling approach to choose the optimum deep neural networks (DNN) on the fly for each video outline to improve recognition accuracy.

Bochkovskiy *et al*. [5] built a production system object detector with a high operational speed and optimization for parallel calculations instead of the low calculation volume theoretical indication (BFLOP). Steady-state genetics selected the optimum hyperparameters. YOLOv3-SPP trained the genetic algorithm with generalized intersection over union (GioU) loss to find min-val 5k sets for 300 epochs. They computed 43.5% AP (65.0% AP50) on the Tesla V100 at 65 FPS. Zhao *et al*. [6] limited example-based and block-based trimming to a wide variety of convolutional (CONV) and Fully-Connected (FC) layers. Pruning algorithms helped them investigate. They evaluated YOLOv4 on the COCO dataset for 2D object recognition. They considered 3D identification on the KITTI dataset using Point Pillars. Experiments showed that the suggested method consistently achieves 55ms guessing times for YOLOv4-based 2D item discovery and 99ms for Point Pillars-based 3D identification on a commercially available mobile device, with only slightly decreased accuracy. The DL object finder with micro fluidic image-activated droplet sorting (DL-IADS) was suggested by Howell *et al*. [7] to execute the adaptable, name-free game plan, counting, and limitation of diverse smaller-than-anticipated articles at high throughput. YOLOv4-small was used with good accuracy and speed for multi-class counting of cells, cell totals, and polyacrylamide (PA) globules.

To achieve high detection accuracy, Liu *et al*. [8] compared three cutting-edge object recognition techniques: RetinaNet, fully convolutional one-stage object detection (FCOS), YOLOv3, and YOLOv4. They handled YOLOv4's shortcut layer and convolutional channel to create thinner and shallower models. Parico and Ahamed [9] used the RGB dataset. They built the real time pear fruit detection model to increase accuracy given time, equipment, and dataset size, and to examine the determination speed of the YOLOv4 family and identify which one has allowance speed near to advancing (>24 FPS). Since YOLOv4 had a low misleading negative rate when differentiating pear organic items, summing with deep simple online realtime tracking (SORT), the exceptional ID was not established to be more reliable, with an F1 count of 87.85%. Kumari *et al*. [10] introduced the mobile eye-tracking model to distribute flexible eye-following data to accurate items using object recognition algorithms. They showed that combining YOLOv4 with an optical stream evaluation yields the fastest outcomes with the highest accuracy of 90% for object recognition. It allows continual framework replies to the client's look and reduces portable eye-following data review time.

Chen *et al*. [11] used pictures to tackle the scale pest problem using an AI-based pest-finding framework. Object recognition methods were used to analyze the data. The adaptive image scaling approach effectively minimizes computation and redundancy, and the updated CSPDarkNet53 used in the Yu and Zang [12] studied trunk FE network reduces the network's processing expenses while increasing the model's learning capacity. According to the studies, the face mask identification method has a mAP of 98.3% and a frame rate of 54.57 FPS, quicker than the current approach. Region-based fully convolutional network (R-FCN), mask region-based convolutional neural network (R-CNN), a single-shot detector (SSD), RetinaNet, and YOLOv4 were examined by Haris and Glowacz [13]. They used the Berkeley deep drive 100K dataset for these comparisons. Their strengths and limitations are assessed by accuracy, computation time, and precision-recall curve. YOLOv4 outperformed in recognizing challenging road target objects under various road settings and weather conditions. Babu *et al*. [14] worked on identifying the facial expression using bezier curves, and Image segmentation based on scanned document detection was done by using neural network (NN) [15]. Shankar *et al*. [16] developed a tool to remove noise images on portable gray map (PGM) and designed a framework using the YOLO model [17]. Shankar *et al*. [18] used a noise reduction filter using social group optimization (SGO).

Roy *et al*. [19] suggested a high-accuracy single-stage object placement model that turns item identification into a relapse problem by creating jumping box arrangements and assigning class confidence. They also developed a high-performing continuous fine-grain object identification framework to overcome several plant sickness localization obstacles that prevent conventional methods from working.

The Microsoft common objects in context (MS COCO) dataset were utilized by Cai *et al*. [20] as their train and test dataset. They use pruning methodologies for inference to achieve realtime mobile object detection. Specifically, a regularisation pruning technique was applied. The results showed that the YOLOv4 model is 92.4% accurate. Guo *et al*. [21] suggested YOLOv4-tiny to differentiate electronic components and present the model on an electronic part dataset for approval. Electronic components are tiny, hard to discern, and move on a transit line, making objective discovery harder. Compared to faster RCNN, SSD, RefneDet, EfcientDet, and YOLOv4, YOLOv4-tiny has the highest location accuracy and quickest speed and may be used to build electronics industry assembly robots. The initial calculation's accuracy increased from 93.74 to 98.6% based on trial data. Liu *et al*. [22] conducted several removal tests for the updated YOLO v4 in SeaShips and SeaBuoys. They developed a residual depth-wise separable convolution (RDSC) model and applied it to the YOLO v4 spine and component combination organizations. The improved YOLO v4 had a 25% increase in identification speed, 1.78% in mAP %, and 0.95% in the two information arrangements.

The 2010 and 2012 ImageNet large scale visual recognition challenge (ILSVRC) subsets were generated by Krizhevsky [23]. They created one of the biggest CNNs using these datasets. Two new, massive datasets are LabelMe, with hundreds of thousands of well-segmented photographs, and ImageNet, with over 15 million marked high-resolution photos in 22,000 categories. They won the ILSVRC-2012 competition with a 15.3% test mistake rate, compared to 26.2% for the second-best passage, using a variant of their algorithm. Fast YOLO, a smaller variant of the organization presented by Redmon *et al*. [24], examines 155 frames per second (FPS) and doubles detector mAP. Convolutional layers are trained using ImageNet 1,000-class competition data. According to the data, YOLO learns extensive representations with 92.83% accuracy. A region proposal network (RPN) by Ren *et al*. [25] communicates full-picture convolutional properties with the recognition organization for almost-free location identification. Region proposal networks (RPNs) predict local object limitations and objectness scores. The GPU-detected VGG-16 model runs at 5FPS. Pattern analysis, statistical modelling, and computational learning (PASCAL) achieved 5 FPS (all stages) on a GPU with state-of-the-art object detection accuracy. The light detection and ranging (LiDAR) sensor were employed in the realtime object detection model by Fan *et al*. [26], which can offer 360° ambient depth information with a detection range of 120 meters. Datasets from PASCAL and KITTI visual object classes (VOC) were used. KITTI provides inside-out information for LiDAR segmentation (LS) of objects from

LiDAR point clouds, while PASCAL VOC was used to train the YOLOv4 neural network for object identification. They found 91.27% accuracy in Lidar segmentation.

Ganesh *et al*. [27] suggested a realtime object detection on edge GPU model that improves accuracy and execution performance on edge GPU devices. YOLO-ReT with MobileNetV 20.75% backbone runs at 3.05 FPS on Jetson Nano and scores 68.75 mean average precision (mAP) on Pascal VOC and 34.91 mAP on COCO, outperforming its competitors by 3.05 and 0.91 mAP, respectively. They also introduced a multi-scale inclusion cooperation module in YOLOv4-tiny, which improved their presentation by 1.3 and 0.9 mAP on COCO. Li *et al*. [28] suggested a calibrated part affinity fields technique to evaluate pedestrian posture based on YOLOv4 structure. Explainable artificial intelligence (XAI) was employed in the risk assessment phase to interpret and estimate results. YOLOv4's total parameters were decreased by 74%, indicating it can run in real-time. Li *et al*. [29] worked on forward location prediction using a Siamese network to reduce false positives from noisy detections, whereas reverse forecast check reduces false positives from forward expectation. The remaining tracks are identified and have future expectation certainty via weighted consolidation. Results showed that the suggested technique beats the state-of-the-art on the UA-DETRAC vehicle in the following dataset and maintains continuous processing at 20.1 FPS. Li *et al*. [30] proposed YOffleNet. This additional object detection model limits accuracy loss while compressing information rapidly for ongoing and safe driving applications on autonomous cars. Using the KITTI dataset as a test bed, experiments revealed that the proposed YOffleNet is 4.7 times more compressed than the YOLOv4-s, which could produce 46 FPS using a coordinated graphics processing unit (GPU) system (NVIDIA Jetson AGX Xavier). To 85.8% mAP, which is just 2.6% less accurate than YOLOv4, the accuracy is considerably reduced compared to the high compression percentage. Thus, the suggested network can reliably identify objects on an autonomous system's implanted system.

Gao *et al*. [31] added channel attention mechanism to the YOLOv4 algorithm and created an object recognition method with channel attention mechanism to improve visual feature representation. The module initially performed global average pooling on the features recovered by YOLOv4, then performed local cross-channel interaction operation on the feature channels using one-dimensional convolution to increase the correlation between channel features to improve placement accuracy. Guo *et al*. [32] developed a deep learning (YOLO model) based, real-time object recognition system for mixed reality devices. Using the YOLO paradigm, they presented a HoloLens-Ubuntu real-time communication system for object identification. The experiment results indicated that HoloLens realtime object identification using the suggested model is quick and accurate at 92.8%. They believe it makes Microsoft HoloLens a robot vision device and improves human-robot collaboration. To organize 24 geo-referenced RGB images on an 8-ha grape plantation and to determine the number of packs, Sozzi *et al*. [33] suggested that the Grape yield spatial inconstancy model was employed. This has been done in light of several target images (320-1,280 pixels) and varied certainty edges (0.25-0.35). Subsequently, the number of packs that were detected was compared to the actual number, together with the total weight obtained from the plants that were the subject of the collected images.

3. METHODOLOGY

Digitally detecting semantic entities like people, buildings, automobiles, and animals in images and films is called object detection. It involves image processing and computer vision. YOLOv4, a state-of-the-art (SOTA) realtime object detection model, is used. YOLOv4 is the fourth YOLO game. It performed SOTA on the 80-category common objects in context (COCO) dataset. The YOLOv4 detector is single-stage. One-stage object detection prioritizes inference speeds. One-stage detector models predict picture classes and bounding boxes but not ROIs. Thus, they are quicker than detectors with two stages.

3.1. Dataset

The data was collected from the Microsoft-published MS COCO dataset, an enormous-scale object detection, segmentation, and inscribing dataset. AI and PC vision researchers generally use the COCO dataset for some PC vision projects. The YOLO model applied to these datasets achieved the objectives of the work.

3.2. Objectives

To fulfill this gap, the objectives of the Computer Vision Enhancement using YOLOv4 are to Count the total number of objects in the image, Count the things per Class in the image, and finally. Crop the detected objects and save them as a new idea in a new folder. The user can easily identify the objects from the images using these objectives.

3.3. Proposed method (YOLOv4) architecture

The inner-workings of the YOLOv4 system are broken down into its component parts and categorized according to their architecture. In the architecture shown in Figure 3, we can see that the YOLOv4 strategy is being used. We can understand that the YOLOv4 technique's five steps involved in computer vision enhancement are input, backbone, neck, head, and custom functions.

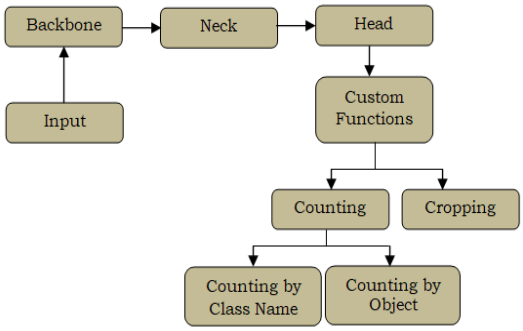


Figure 3. YOLOv4 architecture

3.3.1. Input

The first Input is just our collection of training photographs, which will be taken care of into the network in batches and processed by the GPU. The Input is given primarily in the Yolov4 technique, and the entire flow is shown in Figure 4. The provided Input can be of any form, such as images, videos, patches, and image pyramids.

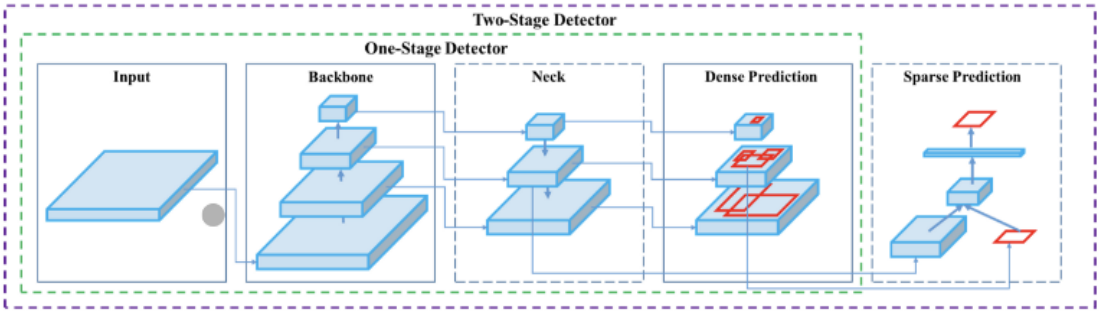


Figure 4. Processing an input

3.3.2. Backbone network

The primary purpose of the backbone is to extract the most relevant information; selecting the proper spine is an essential step in increasing object detection speed. Its goal is to locate important features in an input, but this would be improved in the new role of object detection. CSPResNext50, CSPDarknet53, and EfficientNet-B3 were believed to be the backbone networks. After much research and experimentation, CSPDarknet53 CNN was finally chosen. The DenseNet architecture is used in CSPDarkNet53. Before moving into the dense layers, it joins the previous inputs with the current one. This is known as the dense connection pattern. Mainly, CSPDarkNet53 is made up of two components: i) convolutional base layer and ii) cross stage partial (CSP) block

The cross stage partial technique separates the base layer feature map into two halves. It combines them using a cross-stage hierarchy to avoid "VanishingGradient" and increase gradient flow between layers. The base convolutional layer is made up of the full-size feature map input. As previously explained, the convolutional base layer is next to the CSP block. It breaks the information into two parts, one transmitted via the dense block and the other without processing, sent to the following stage. CSP keeps fine-grained features for more effective transmission, promotes network reusability, as well as lowering the number of network parameters. Only the backbone network's final convolutional block, which can extract more semantically rich data, is dense, as evidenced by more densely coupled convolutional layers, which can slow down detection.

3.3.3. Neck

Features converge near the neck. It compiles feature maps from different backbone stages and merges them to prepare them for the next phase. The channel has several top-down and bottom-up routes. spatial pyramid pooling (SPP) is added between the feature aggregator network and the PANet backbone. It improves the receptive field and filters out essential context items without affecting network performance. It links to the highly CSPDarkNet's last convolutional layers. Only one kernel or filter is applied to an image's receptive field. When we develop dilated convolutions, it rises exponentially, causing non-linearity. A modified route aggregation network is utilized to make YOLOv4 more suitable for single GPU training, as illustrated in Figure 5. The path aggregation network's (PANet's) primary function is to increase the segmentation efficiency by preserving space data, which aids in appropriate pixel localization for mask prediction. The main qualities that make them precise for ask prediction are path augmentation from the bottom up, adaptive feature pooling, and fully connected fusion.

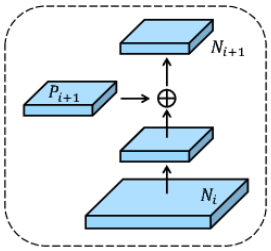


Figure 5. Feature aggregator network

3.3.4. Head

The head's primary task in YOLOv4 is prediction, which comprises classification and regression of bounding boxes. The primary goal of this software is to find bounding boxes and categorize them. The bounding box coordinates (x, y, height, width) and the scores are recognized. The b-center box's x and y coordinates are at the grid cell's border. The width and size of the image are computed to the whole. A YOLOv4 head can be installed in any anchor box. As shown in Figure 6, anchor boxes hold many objects of varied sizes in a single frame with the center in the same cell. In contrast to the preceding illustration, a grid was utilized to recognize a single object in a frame.

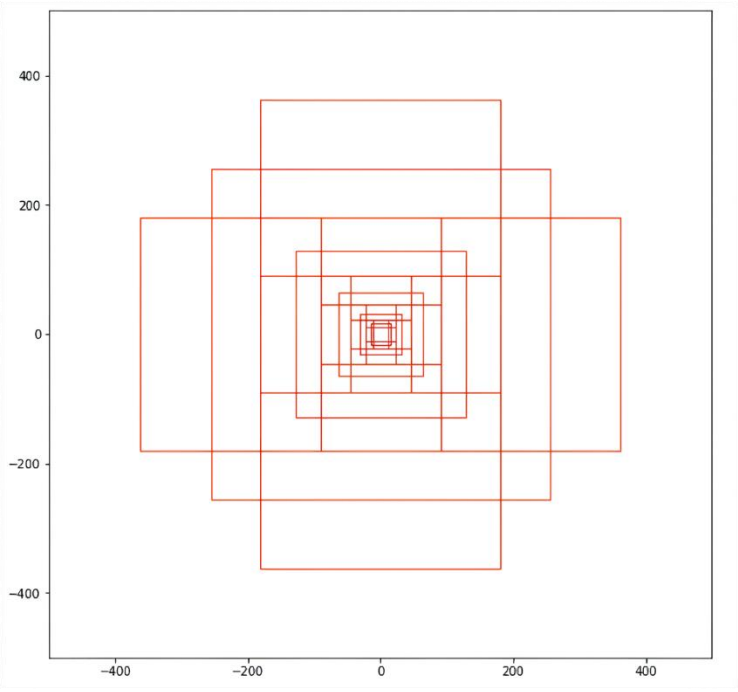


Figure 6. Example of anchor box plated around (0,0) of different scales

3.3.5. Custom functions

a) Counting objects

A custom function is constructed inside the file core/functions.py. It may be used to enumerate and monitor the number of items identified in each picture or video at any given moment. It has the capability to monitor the overall count of identified objects or the quantity of items detected by class. Add to count real objects, and the command above counts the total count of objects found and displays on your command prompt or shell and the saved detection, as shown in Figure 7. The algorithm to count the number of things present in the image is as Algorithm 1 - counting objects:

Step 1: Define a dictionary count in the count_objects function to hold the count of objects detected.

Step 2: Since we need to count the objects per Class, we use an if condition as if by_class to read the class name data from yolo_classes.

class_names=read_class_names(cfg.YOLO.CLASSES)

Step 3: The number of objects detected in the image of the input data is assigned to the variable num_objects as follows:

num_objects=data

Step 4: We traverse the num_objects variable to grab the class_index and convert it into the corresponding class name.

class_index =int(classes[j])class_name=class_names[class_index]

Step 5: The class name is assigned to the count's dictionary if it is present in the allowed_classes list it continues.

ifclass_nameinallowed_classes:
counts[class_name]=counts.
get(class_name,0)+1 else:
continue

Step 6: If step 2 is false, we assign the count of the total number of objects to the count's dictionary.

counts['totalobject']=num_objects

Step 7: Finally, we return the count dictionary.

Return counts

b) Counting objects per class

To enable the counting of multiple items for each class in your object detector, modify a single line in either the detect.py or detect_video.py script. Use the custom flag "--count" as shown in Table 1. The count objects method has a default value of False for the by-class argument. When the value of this option is set to True, the count is calculated for each individual class. For counting per class, rewrite the FLAGS. count command. Figure 8 shows the total number of objects per class.

Table 1. Function for counting classes

If FLAGS.count:
count object found counted_classes = count_objects (pred_bbox, by_class = True)

Figure 7. Counting objects

Figure 8. Counting objects per class

c) Crop detections and save themes. new images

The YOLOv4 detections may be cropped and saved as new photos by using a custom function that can be found in the file core/functions.py. This function can be used to any detect.py or video.py command. Simply appending the - crop flag to any order will result in crop detections being generated. After counting the total number of items in the picture, the detections that have been cropped are stored in the folder titled detections/crop. The items may be trimmed down in the manner shown in Figure 9. Algorithm 2 - crop the detection of an image:

Step 1: The name of the Class is assigned to the counts dictionary if it is present in the allowed_classes list else it continues. If class_name in allowed_classes:

```
counts[class_name]=counts.get( class_name,0)+1
else:
    continue
```

Step 2: The box coordinates for each detection on a 2-D plane is stored as

```
xmin, ymin, xmax, ymax = boxes [1]
```

Step 3: The detection is cropped from the image and saved as an image in cropped_img variable.

```
Cropped_image=img[int(ymin): int(ymax), int(xmin): int(xmax)]
```

Step 4: The image obtained through cropping must have the class name and the detection number as its image name.

```
img_name=class_name+'_'+str(counts[class_name])+'.png'
```

Step 5: Save the cropped detection from the image

```
cv2.imwrite (img_path, cropped_img).
```

Figure 9. Crop detection of the objects in an image

d) Calculating the confidence score

YOLO defines the confidence score as P(Object) * (intersection/union). The confidence score per bounding box is one of the neural network's outputs at test time; it is not recomputed but used to determine which boxes have the highest confidence. As shown in Figure 10, the confidence ratings of various items are calculated using multiple parameters and displayed above the top corner of each Object. The conditional class probabilities and the individual box confidence predictions are multiplied as (1) at test time:

```
Pr(Classi|Object) * Pr(Object) * IOUtruthpred =
Pr(Classi) * IOUtruthpredPr (Classi|Object) * Pr(Object) * IOUpredtruth =
Pr(Classi) * IOUpredtruth                                                    (1)
```

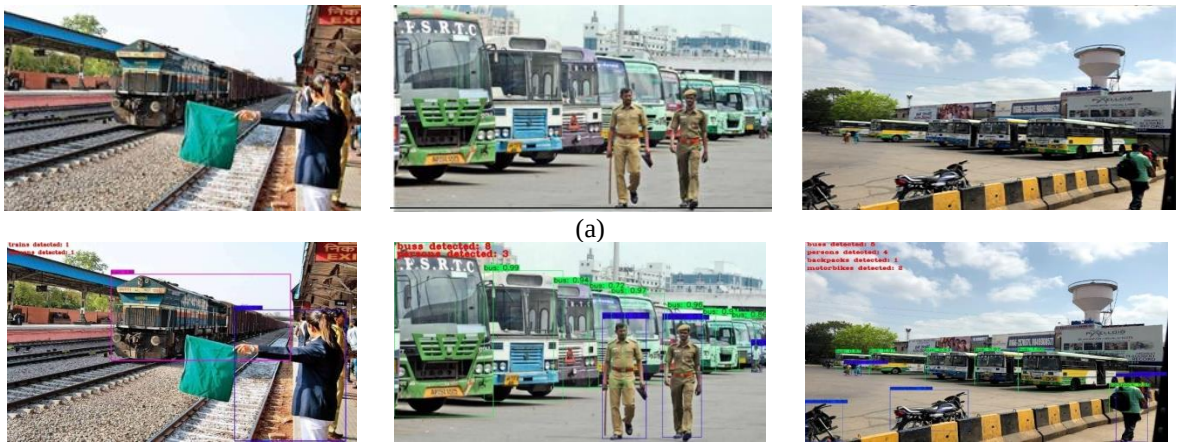
Figure 10. Confidence score of the objects

This is done for each bounding box separately. For each bounding box, acquire a numerical result used as the confidence score. They get two such outcomes for the two bounding boxes per grid square that they employed in their experiment. That output corresponds to two terms on the left-hand side of the equation above. Then, they multiply by the conditional chance that a grid square includes a specific class if it contains an object. This yields a confidence score for each frame and class.

4. RESULTS

The system detects the objects from the image according to their class and displays it on the output image at the top left corner, as shown in Figure 11. Here, in Figure 11(a), the images were taken as Input, and in Figure 11(b) it will display the image class in the left corner. Here we consider Blur images to track the objects. Here, we have shown three different images to track the things.

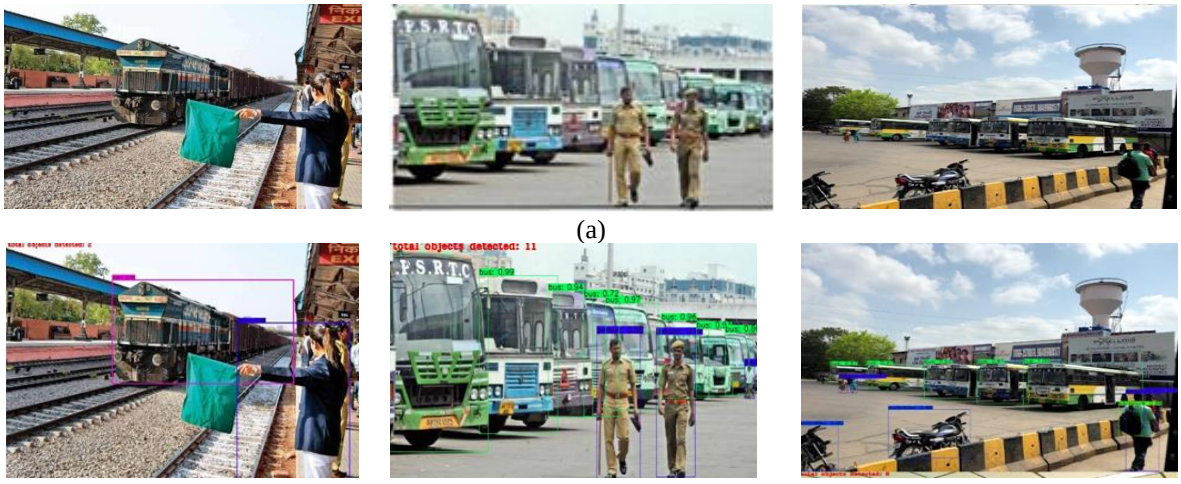
The system detects the total number of objects in the input image and displays it on the output image at the top left corner, as shown in Figure 12. In Figure 12(a), the images were taken as Input. In Figure 12(b), they were cropped and saved as new images in the given folder if observed that Figure 11 images are detected from per class, and Figure 12 images are the total no of object.



(a)

(b)

Figure 11. Results of the object detected per class as given in (a) input images and (b) output images

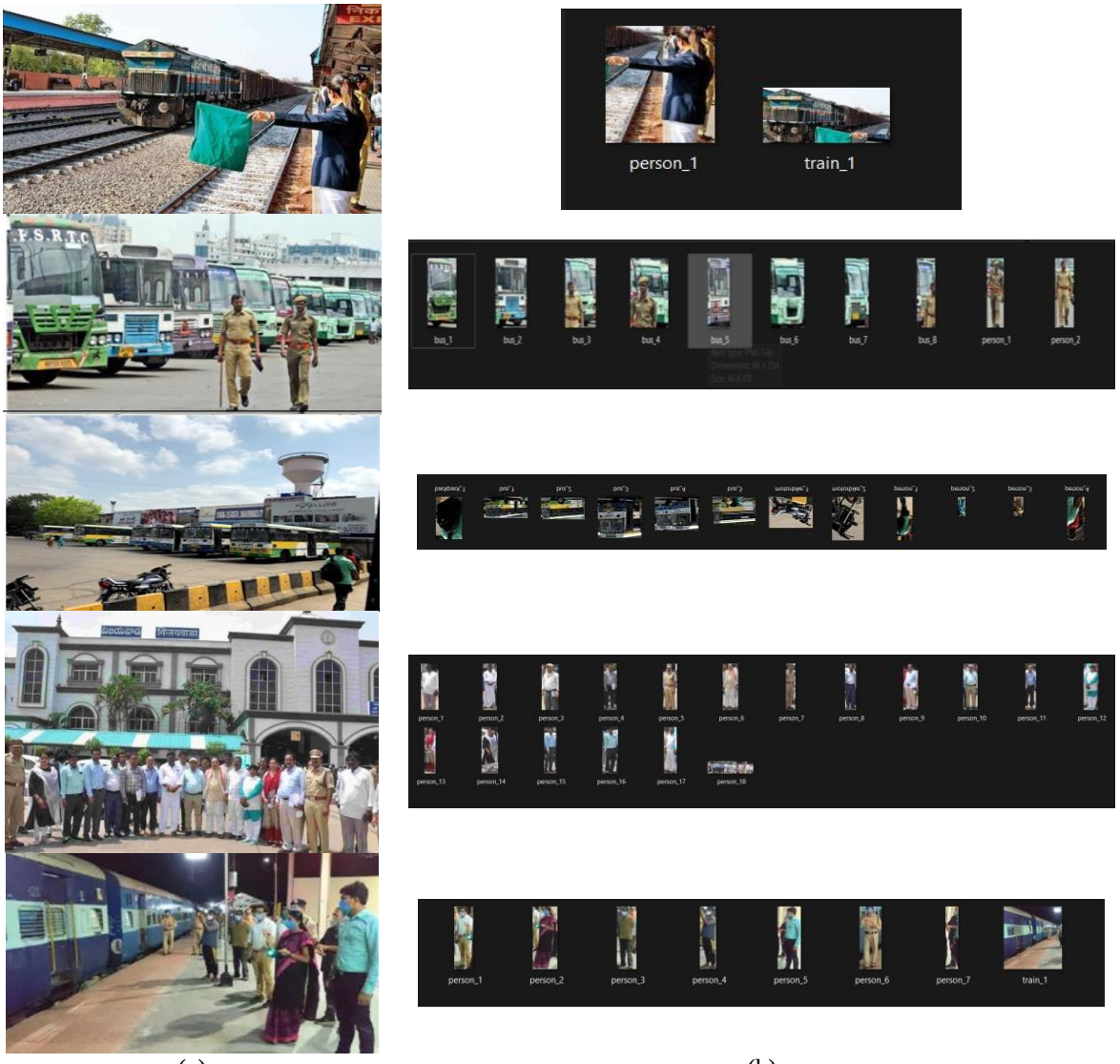


(a)

(b)

Figure 12. Results for objects detected as given in (a) input images and (b) output images

After the results of Figure 12, the images were considered cropped. The detected objects are cropped and saved as new images with their class name as their image name in a new folder called crop, as shown in Figure 13. In Figure 13(a), the images were considered input images, and Figure 13(b) shows the images after cropping. Here, we have viewed some blurred images as Input and cropped them from them. After cropping, the images are considered as output, as shown in Figure 13. The custom functions are built on top of the YOLOv4 framework. The output includes the number of items identified, as well as a bounding box around each object. The confidence score indicates the likelihood that the detected object belongs to the predicted class according to YOLOv4. A confidence threshold is implemented to eliminate the detections with low confidence.



(a) (b)

Figure 13. Results after cropping the images as given in (a) input images and from that and (b) output images

5. CONCLUSION

This research was effective in determining the presence of the things that are visible in the picture. In general, YOLOv4 is an advanced object identification model that can identify the many things that may be seen in a picture. We built custom procedures to count items per Class, crop the picture, and store it in a different folder since YOLOv4's information is not fully used. With the help of these custom functions, we are able to do an analysis of the data more quickly. In addition to recognizing visual objects, a confidence

score is produced to provide the user a likelihood. The monitoring industry may find this technology useful. We are able to assert that the newly installed system likewise has the same degree of precision. The graph above and result analysis show that YOLOv4 is more accurate than YOLOv3 and other real-time object identification methods.

ACKNOWLEDGEMENTS

The authors have done their work individually and declared no conflicts of interest.

REFERENCES

[1] Y. Yang *et al*., "Realtime detection of aircraft objects in remote sensing images based on improved YOLOv4". In 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC) 2021 Mar 12 (Vol. 5, pp. 1156-1164). IEEE.DOI: 10.1109/IAEAC50856.2021.9390673.

[2] A. Kumar *et al*., "A hybrid tiny YOLO v4-SPP module based improved face mask detection vision system. Journal of Ambient Intelligence and Humanized Computing". 2021 Oct 20:1-4. DOI: https://doi.org/10.1007/s12652-021-03541-x.

[3] R. Wang *et al*., "A Realtime Object Detector for Autonomous Vehicles Based on YOLOv4". Computational Intelligence and Neuroscience. 2021 Dec 10;2021. DOI:https://doi.org/10.1155/2021/9218137.

[4] J. Lee *et al*., "TOD: Transprecise object detection to maximise realtime accuracy on the edge". In2021 IEEE 5th International Conference on Fog and Edge Computing (ICFEC) 2021 May 10 (pp. 53-60). IEEE. DOI: 10.1109/ICFEC51620.2021.00015.

[5] A. Bochkovskiy *et al*., "Yolov4: Optimal speed and accuracy of object detection". arXiv preprint arXiv:2004.10934. 2020 Apr 23. DOI: https://doi.org/10.48550/arXiv.2004.10934.

[6] P. Zhao *et al*.,"Neural Pruning Search for Realtime Object Detection of Autonomous Vehicles". In2021 58th ACM/IEEE Design Automation Conference (DAC) 2021 Dec 5 (pp. 835-840). IEEE. DOI: 10.1109/DAC18074.2021.9586163.

[7] L.Howell *et al*., "Multi-Object Detector YOLOv4-Tiny Enables High-Throughput Combinatorial and Spatially-Resolved Sorting of Cells in Microdroplets". Advanced Materials Technologies. 2022 May;7(5):2101053. DOI: https://doi.org/10.1002/admt.202101053.

[8] H. Liu *et al*., "Realtime small drones detection based on pruned yolov4. Sensors". 2021 May 12;21(10):3374. DOI: https://doi.org/10.3390/s21103374.

[9] AI. Parico AI and T. Ahamed, "Real time pear fruit detection and counting using YOLOv4 models and deep SORT. Sensors". 2021 Jul 14;21(14):4803. DOI: https://doi.org/10.3390/s21144803.

[10] N. Kumari *et al*., 'Mobile Eye-Tracking Data Analysis Using Object Detection via YOLO v4. Sensors". 2021 Nov 18;21(22):7668. DOI: https://doi.org/10.3390/s21227668.

[11] JW. Chen *et al*.,"A smartphone-based application for scale pest detection using multiple-object detection methods". Electronics. 2021 Feb 3;10(4):372. DOI: https://doi.org/10.3390/electronics10040372.

[12] J. Yu and W. Zhang, " Face mask wearing detection algorithm based on improved YOLO-v4. Sensors". 2021 Jan;21(9):3263. DOI: https://doi.org/10.3390/s21093263.

[13] M. Haris and A. Glowacz, "Road object detection: A comparative study of deep learning-based algorithms". Electronics. 2021 Aug 11;10(16):1932. DOI: https://doi.org/10.3390/electronics10161932.

[14] D. R. Babu *et al*., "Facial expression recognition using bezier curves with hausdorff distance". In2017 International Conference on IoT and Application (ICIOT) 2017 May 19 (pp. 1-8). IEEE. DOI: 10.1109/ICIOTA.2017.8073622.

[15] R. B. Devareddi *et al*.,"Image segmentation based on scanned document and hand script counterfeit detection using neural network". InAIP Conference Proceedings 2022 Dec 9 (Vol. 2576, No. 1, p. 050001). AIP Publishing LLC. DOI: https://doi.org/10.1063/5.0105808.

[16] R. S. Shankar *et al*.,"Object oriented fuzzy filter for noise reduction of Pgm images". In2012 8th International Conference on Information Science and Digital Content Technology (ICIDT2012) 2012 Jun 26 (Vol. 3, pp. 776-782). IEEE.

[17] R. S. Shankar *et al*.,"A Framework to Enhance Object Detection Performance by using YOLO Algorithm". In2022 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS) 2022 Apr 7 (pp. 1591-1600). IEEE. DOI: 10.1109/ICSCDS53736.2022.9760859.

[18] V. M. Gupta *et al*., "A Novel Approach for Image Denoising and Performance Analysis using SGO and APSO". InJournal of Physics: Conference Series 2021 Nov 1 (Vol. 2070, No. 1, p. 012139). IOP Publishing. DOI: 10.1088/1742-6596/2070/1/012139.

[19] A. M. Roy *et al*., "A fast accurate fine-grain object detection model based on YOLOv4 deep neural network". Neural Computing and Applications. 2022 Mar;34(5):3895-921. DOI: https://doi.org/10.1007/s00521-021-06651-x.

[20] Y. Cai *et al*., "Yolobile: Realtime object detection on mobile devices via compression-compilation co-design". InProceedings of the AAAI Conference on Artificial Intelligence 2021 May 18 (Vol. 35, No. 2, pp. 955-963). DOI:https://doi.org/10.1609/aaai.v35i2.16179.

[21] C. Guo *et al*., "Improved YOLOv4-tiny network for realtime electronic component detection". Scientific Reports. 2021 Nov 23;11(1):1-3. DOI: https://doi.org/10.1038/s41598-021-02225-y.

[22] T. Liu *et al*., "Sea Surface Object Detection Algorithm Based on YOLO v4 Fused with Reverse Depthwise Separable Convolution (RDSC) for USV". Journal of Marine Science and Engineering. 2021 Jul 7;9(7):753. DOI: https://doi.org/10.3390/jmse9070753.

[23] A. Krizhevsky *et al*., "Imagenet classification with deep convolutional neural networks". Communications of the ACM. 2017 May 24;60(6):84-90.

[24] J. Redmon *et al*., "You only look once: Unified, realtime object detection". InProceedings of the IEEE conference on computer vision and pattern recognition 2016 (pp. 779-788).

[25] S. Ren *et al*., "J. Faster r-cnn: Towards realtime object detection with region proposal networks". Advances in neural information processing systems. 2015;28.

[26] Y. C. Fan *et al*.,"Realtime Object Detection for LiDAR Based on LS-R-YOLOv4 Neural Network". Journal of Sensors. 2021 May 26;2021. DOI: https://doi.org/10.1155/2021/5576262.

[27] P. Ganesh *et al*., "YOLO-ReT: Towards high accuracy realtime object detection on edge GPUs". InProceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision 2022 (pp. 3267-3277).

[28] Y. Li *et al*., "A deep learning-based hybrid framework for object detection and recognition in autonomous driving". IEEE Access. 2020 Oct 23; 8:194228-39. DOI: 10.1109/ACCESS.2020.3033289.

[29] Y. Li *et al*., "Crop pest recognition in natural scenes using convolutional neural networks". Computers and Electronics in Agriculture. 2020 Feb 1;169: 105174.DOI: https://doi.org/10.1016/j.compag.2019.105174.
[30] I. Sim *et al*., "Developing a Compressed Object Detection Model based on YOLOv4 for Deployment on Embedded GPU Platform of Autonomous System". arXiv preprint arXiv:2108.00392. 2021 Aug 1.DOI: https://doi.org/10.48550/arXiv.2108.00392.
[31] C. Gao,"YOLOv4 object detection algorithm with efficient channel attention mechanism". In2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE) 2020 Dec 25 (pp. 1764-1770). IEEE. DOI: 10.1109/ICMCCE51767.2020.00387.
[32] J. Guo *et al*., "Realtime Object Detection with Deep Learning for Robot Vision on Mixed Reality Device". In2021 IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech) 2021 Mar 9 (pp. 82-83). IEEE. DOI: 10.1109/LifeTech52111.2021.9391811.
[33] M. Sozzi *et al*., "Grape yield spatial variability assessment using YOLOv4 object detection algorithm". Proceedings of the Precision Agriculture '21, ECPA. 2021 Jul 19.

BIOGRAPHIES OF AUTHORS

Shiva Shankar Reddy is an Assistant Professor at the Department of Computer Science and Engineering in Sagi Rama Krishnam Raju Engineering College, Bhimavaram, Andhrapradesh, INDIA. He is pursuing PhD degree in Computer Science and Engineering with a specialization in Medical Mining and Machine Learning. His research areas are Image Processing, Medical Mining, Machine Learning, Deep Learning and Pattern Recognition. He published 30+ papers in International Journals and Conferences. S.S. Reddy has filed 05 patents. His research interests include image processing, medical mining, machine learning, deep learning and pattern recognition. He can be contacted at email: shiva.csesrkr@gmail.com.

Dr. Venkata Rama Maheswara Rao is a leading Researcher & Academician in Computer Science & Engineering and holds Ph.D. degree. He is currently working as a Professor in the Dept. of Computer Science & Engineering at Shri Vishnu Engineering College for Women (A), Andhra Pradesh, India. He is actively involved and successfully implemented three projects funded by DST. He has 45 research papers, 17 of which are Scopus-indexed and 7 of which are Web of Science-indexed. He has 23 years of experience that include 6 years of Industry experience, 19 years of teaching experience and 15 years of Research experience. His Research interests include data mining, web mining, cloud computing, big data analytics, data science, artificial intelligence, and machine learning. He can be contacted at email: mahesh_vvr@yahoo.com.

Priyadarshini Voosala is Assistant Professor at Sagi Rama Krishnam Raju Engineering College, Department of Computer Science and Engineering, India. She Received a B.Tech. degree from Sri Vishnu Engineering College for Women, Department of Computer Science and Engineering, in 2005. She holds an M.Tech. degree in Sagi Rama Krishnam Raju Engineering College, Department of Computer Science and Engineering, in 2010. Her research areas are cloud computing, fog computing, edge computing, machine learning, and image processing. She has 05+ research Scopus-indexed papers. She can be contacted at email: priyavoosala@gmail.com.

Silpa Nrusimhadri is working as Assistant Professor in the Department of Computer Science & Engineeringat Shri Vishnu Engineering College for Women (A), Andhra Pradesh, India.Presently she is pursuing her Ph.D. in Computer Science and Engineering at Centurion University of Technology and Management (CUTM), Odissa, India. She has 13 years of teaching experience and 8 years of Research Experience. Her Research interests include data mining, web mining, big data analytics, text mining, data science, artificial intelligence, and machine learning. She is actively involved and successfully implemented two projects funded by DST. She has 11 research Scopus-indexed papers. She can be contacted at email: nrusimhadri.silpa@gmail.com.