International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Comparative Study of the Different Object Detection Algorithms: YOLOv4, SSD, and RCNN based on Accuracy and Speed

Saurish Reddy Dirisala[1], Dr. Dileepkumar Padidem[2]

[1]The International School Bangalore, Bangalore, India
Email: *saurish456[at]gmail.com*

[2]RSREC, Jawaharlal Nehru Technological University, AP, India
Email: *padidemdileep[at]gmail.com*

Abstract: ***Computer Vision is an interesting and active area of research from past couple of decades. Object detection is a computer vision technique for locating instances of objects in images or videos. It is extensively utilized in various fields such as Self driving, Security surveillance, health care, robotics and so on. The major motivation of this research paper is to test the accuracy and performance of the various popular machine learning algorithms on simple objects under different conditions. Creation of sample images is done using some images found on the internet and some captured at home. Analysis is done for various situations like blurred images, shadows and reflections. This paper explores the features of single-shot detectors (YOLOv4 and SSD) and two-shot detectors (R-CNN) based on speed and accuracy.***

Keywords: Object detection, Computer vision, YOLO V4, SSD, RCNN, Artificial intelligence, Machine learning

1. Introduction

Identifying and classifying a specific object in digital photographs is done using the object detection, a computer vision technique. By scanning digital images for instantaneous predictions, object detection is used to recognize and track particular objects. To be more explicit, object detection, in simplest terms, uses computer vision to locate the stated things in the provided environment by drawing a bounding box around the input image.

Understanding picture categorization and item identification is essential for object detection. Image classification entails tasks like predicting an object's class among many classes using the feature set that was employed by the algorithm during the pre-training period. To create a bounding box around the extent of one or more objects in a picture, object localization must be done first. Finding an object's position is referred to as object localization. Object detection combines these two tasks to identify and categorize one or more items in a picture.

The most noticeable feature of computer vision is object detection. It serves as a starting point for a variety of computer vision tasks that come later, such as image segmentation, object tracking, and image captioning. Additionally, its applications, such as number-plate detection, face detection, vehicle detection, and self-driving, are amply demonstrated in practice.

Object detection techniques
Object detection can be performed in two different techniques

1. Image processing technique:
This technique is unsupervised in nature since it does not require any historical training data.

Pros: Reduces dependence on humans who are required to annotate images in supervised learning. As a result, the models that use this particular technique do not require high GPUs or large data sets, thus being less computationally intensive.
Cons: Due to the model being based on such a technique, it is restricted to factors such as complex scenarios, occlusion, illumination, shadows, and clutter effects.

2. Deep learning techniques:
Models based on this technique use either supervised or unsupervised learning, with supervised learning being the preferred method in the majority of objection detection applications.
Pros: It can handle occlusion, complex settings, and illumination far better, yielding much more accurate results.
Cons: Because traditional computer vision techniques involve supervised learning, a large amount of training data is necessary, which is time-consuming and costly when considering the number of required annotated images.

Challenges

1. Dual synchronization: The initial task that developers face is object localization, which involves classifying the image and determining the position of the object under consideration in the image.
Solution: Regional-based Convolutional neural networks are an object detection framework that consists of general proposal regions where the desired object is most likely to be located, followed by CNN model processing to categorize and rectify the object positions. The fast-R CNN model can improve the original R-CNN results. Because of its capacity to optimize picture localization and object recognition using a multi-task loss function, this fast-R CNN model boosts detection speed while also boosting accuracy.

Licensed Under Creative Commons Attribution CC BY

2. Real-time detection speed: Object detection algorithm detection speed has increased over time, from 0.02 fps to 155 fps. Experts have always found it challenging to effectively categorize and localize the target object while meeting real-time video processing requirements.
Solution: Models such as Faster R-CNN and Fast R-CNN are designed to accelerate the basic R-CNN approach. A large bottleneck is usually produced by the R-CNN's selective search function, which is employed to generate 2000 candidate regions of interest. This is then processed individually by each CNN-based model. The Fast R-CNN model, on the other hand, transmits the entire image through the CNN base once and then compares the ROIs obtained with selective search to the feature map, resulting in a 20-fold reduction in processing time.

3. Multiple aspect ratios and spatial scales: The desired object can be seen in a wide variety of aspect ratios and sizes.
Solution: The Faster R-CNN uses a regional proposal network that generates the desired region under consideration by sliding a window rather than doing a selective search over the image's convolutional feature map. As a result, the desired regions under examination are also known as regions of interest, and their positions can be specified in relation to the anchor boxes. These anchor boxes' size and shape are chosen to accommodate a variety of aspect ratios and scales.

4. Limited data:A key hurdle encountered during the development of an application is a lack or a restricted amount of annotated data.
Solution: Pre-built datasets, such as the COCO dataset, which contains 300,000 segmented images with 80 different object categories, might be employed. Furthermore, YOLO9000 can be employed because it is trained in parallel with both ImageNet and COCO and has tens of thousands of object classes.

Applications of Computer Vision

Medical Imaging: Visual pattern recognition through computer vision enables modern technologies to give accurate and speedy diagnosis in the domains of pathology, radiology, and ophthalmology.

Autonomous Vehicles: Using a 360-degree and multi-camera configuration allows the vehicle to recognize and classify things using computer vision, which then participates in complex processes such as path planning, driving scene perception, and behaviour arbitration.

Facial Recognition: Deep learning and machine vision are used for facial recognition. Algorithms recognize and capture photos of people, which can then be utilized to detect and deter criminal activity as well as track specific individuals for security purposes.

Object Recognition Technology based on Deep Learning
These CNN models can be further classified as a single-shot detector or a two-shot object detection models. The model you require for a certain activity is determined by the task's predicted outcome.

Single-shot detectors
- This class uses a single network to predict the object border-box and category likelihood score from a single image. It is an end-to-end object identification and recognition approach based on regression.
- The computation of these is faster with compromised accuracy.

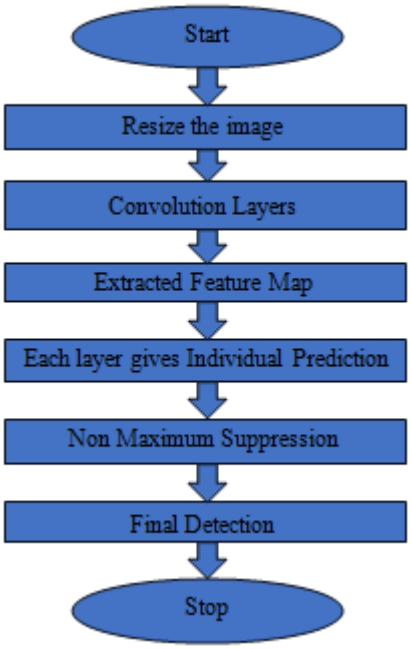
Two-shot detectors
- Detection is done in two stages: the first is a regional suggestion, followed by a classification of those regions and refinement of the site forecast.
- These are heavy on computation but highly accurate

This paper investigates the speed and accuracy characteristics of single-shot detectors (YOLOv4 and SSD) and two-shot detector (R-CNN).

As previously stated, there are fundamentally two types of typical object detection systems. Algorithms such as R-CNN and Fast(er) R-CNN use a two-step approach to detect objects only in the expected regions. First, they select the areas where things are likely to be discovered. Fully convolutional methods, on the other hand, such as YOLO (You Only Look Once) and SSD (Single-Shot Detector), find all objects in an image in a single pass (hence, "single-shot" or "look once" via the convnet). Single-shot algorithms are more efficient and have the same accuracy as region proposal approaches, which are frequently somewhat more accurate but slower to execute.

SSD (Single-shot detector)
Without the usage of a delegated region proposal network (RPN), it predicts border boxes and classes directly from feature maps in a single pass. The Single Shot Multibox Detector (SSD) is a single-stage detector that use multiscale features and default boxes rather than RPN. SSD extracts feature maps for image classification using VGG16, a deep convolutional neural network architecture created at the University of Oxford. The convolutional layer is then used to detect objects. In this case, the convolutional neural networks (CNN) are organised into three layers: base convolutions, auxiliary convolutions, and prediction convolutions. While auxiliary convolutions help extract higher-level feature maps, base convolutions help extract lower-level feature maps. Prediction convolutions can help you detect and locate the things in these feature maps. For each cell or place in the image, four predictions of the object are made, and each one produces a score of 21 for each class plus one border box. SSD uses several layers to detect items individually, and it uses lower resolution layers to recognise larger scale objects. Ground truth boundary boxes can be divided into clusters based on the user-specified default boxes. This is separated into clusters, each of which is represented by a pre-selected bounding box.SSD creates both positive and negative matches while making predictions, but only the positive matches are used for determining why the bounding boxes do not match. If the related default boundary box's IoU with the ground truth is larger than 0.5, the match is regarded successful. Finally, SSD makes use of Non-maximum Suppression (NMS) to eliminate redundant predictions that result in the same item.

Licensed Under Creative Commons Attribution CC BY

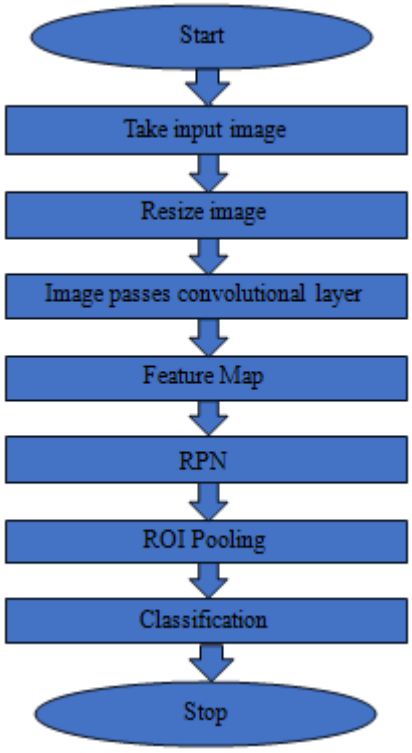


This process includes:
1) Extraction of feature maps
2) Application of convolution filters for object detection

Features include:
1) To predict object classes and offsets to default border boxes, small convolutional filters are used.
2) Filters for default boxes that are specifically designed to accommodate the various aspect ratios.
3) Maps with many scales of features for object detection.

SSD can be trained from start to finish to increase accuracy. SSD covers more locations, sizes, and aspect ratios, resulting in more predictions. By removing the assigned region proposal and using lower-quality images, the model can work at real-time speed while outperforming the accuracy of the most advanced Faster R-CNN.

R-CNN (Region-Based Convolutional Neural Network)
The user perceives the R-CNN, a deep convolutional network for object identification, as a single, end-to-end, unified network. In Faster Region based Convolutional Neural Networks (RCNN), we combine the Fast RCNN detector with a Region proposal Network (RPN). To find the area where the items are visible in the picture, the RPN first creates region recommendations. The anchor boxes are used by RPN to capture objects of varied sizes. Faster RCNN by default places 9 anchors at an image location. The foreground and background class zones are formed after the intersection over union, or IoU, calculation. If IoU> 0.5, the bounding box is categorised as foreground class; otherwise, it is regarded as background class. The feature maps of the anchor boxes are produced by the deep convolutional network employed in this case, the Region Proposal Layer, which may be either a VGG net or a Lx net. These output anchor boxes then go through the Region of Interest Pooling step. In ROI Pooling, all feature maps are shrunk to the same size. By dividing the feature maps into a set number of nearly equal areas and applying Max-Pooling to each region created, this decrease is accomplished using ROI pooling. As a result, regardless of the size of the input, ROI Pooling always produces a consistent output. Once finished, the picture is given to the classifier and regressor for processing. The classifier identifies the item in a picture and determines whether it is an object or the background. The regressor is responsible for drawing a bounding box around the classified item.



This process includes:
1) Inputting an image
2) Extracting potential objects using the selective search algorithm
3) Transferring Learning Feature extraction to compute features
4) Classifying them

The issue with the original CNN is that it employs no deep neural networks for classification and is exceedingly slow.

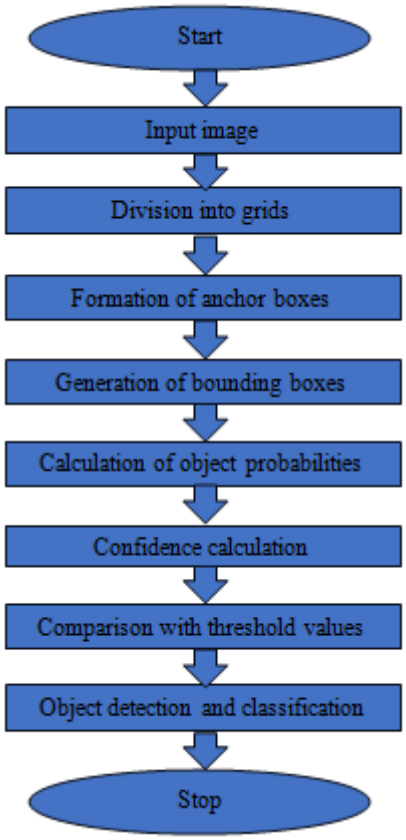
Features of the selective search algorithm include:

Selective search is an algorithm native to R-CNN that identifies the patterns in objects and proposes various regions based on varying scales, colors, enclosures, and textures.
1) Takes an input image
2) Generation of initial sub-segmentations such that multiple regions can be created in the image
3) Combination of similar regions formed to form one larger region. This is done based on color, size, shape, texture, and similarity.
4) Finally, sending it to CovNet.

YOLO (You Only Look Once)
It is a real-time object detection model created in order to overcome the shortcomings of earlier YOLO versions, such as YOLOv3 and other object detection models. Unlike other convolutional neural network (CNN) based object detectors, YOLOv4 may be used for independent process management

Licensed Under Creative Commons Attribution CC BY

and the reduction of human input in addition to recommendation systems. Each individual box prediction's YOLO loss is made up of the words coordinate loss, objectness loss, and classification loss. When the projected box does not completely enclose the dominant object in a picture, coordinate loss occurs, whereas objectness loss happens when an inaccurate box-object (IoU) prediction is displayed. Classification loss happens when there are variations or changes in the precise classes that are predicted, i.e. when the item in the box is predicted to have the proper classes as '1' and all other classes as '0'. Darknet53 and CSPDenseNet are combined to form CSPDarknet53. It is the foundation for object identification and a convolutional neural network. There are 53 convolutional layers in Darknet53. It has 53 convolution layers, with sizes ranging from 1x1 to 3x3. 29 of them have three-by-three convolution layers. A batch normalization (BN) layer and a Mish activation layer are coupled to each convolution layer.



This process includes:

1. Organizing the images along with their annotations into a directory such that each annotation file contains object coordinates and labels.
2. Configuring the model using parameters such as backbone network, input size, number of classes, anchor sizes, and training hyperparameters.
3. Initializing the backbone network either from scratch or using pre-trained weights.
4. Training the model on the labeled dataset using stochastic gradient descent.
5. Evaluating the trained model on a separate validation set to measure performance metrics.

Features include:

1) Backbone network to extract meaningful features from the input image.
2) Neck network to fuse features at different scales, enhancing object detection performance
3) The head network is made up of several convolutional layers, forecasts bounding boxes, class probabilities, and objectness scores for numerous anchor boxes.
4) Feature pyramid enable detection at varying scales, improving accuracy.
5) Improved loss function to handle class imbalance and improve bounding box regression
6) Data augmentation to enhance model generalization

2. Methodology

Dataset Preparation

Microsoft COCO is a well-known dataset for object detection in the picture classification area. A sizable dataset that is accessible for picture recognition and classification serves as support for the research. In this research, we have evaluated and contrasted the performances of the different methods and their implementation using the COCO dataset as a common element. In our research study, we applied the algorithms RCNN, SSD, and YOLOv4.

Software

The software configuration used is the Python 3 Google Compute Engine Backend integrated with Google Colab. It offers 12.72 GB of RAM, of which 3.54 were on average utilised. Additionally, it offers 107.77 GB of disk space, of which 74.41 GB was used to store the training and validation datasets. The artificial GPU from Google Colab was the tool utilized as a hardware accelerator.

Factors the models were tested on

Speed: Speed is one of the most important factors when it comes to comparing various models. While not only giving the fastest output, it highlights the computational intensity of a given dataset.

Accuracy: Along with speed, accuracy plays a vital role in choosing the models for the desired task. This is extremely important when computer vision is used in sensitive situations such as medical image testing for tumors.

Images used

To have an equal comparison between all three models, the same images were used to test the models. Firstly, normal images were used which depicted day-to-day situations such as a cup, bottle, chair, etc. In the second set, we used a set of blurred images to see whether these models can perform with the same efficiency if the image has poor resolution. Lastly, a set of random images was used depicting real-world scenarios such as shadows, reflection, partial objects, etc.

3. Results

Normal pictures:

The 3 algorithms were run on normal 500X667 resolution images.

Licensed Under Creative Commons Attribution CC BY

The following is the time taken in seconds by each algorithm to detect the objects.

Table 1

Object	RCNN	YOLO	SSD
Banana	1.88	0.883	1.11
Chair	1.23	0.904	1.12
Scissors	1.23	0.904	1.11
Cup	1.21	0.687	1.08
Bottle	1.2	0.852	1.07
2 Cycles	1.21	0.728	1.12
Multiple	1.2	0.683	1.16

The following is the accuracy of the 3 algorithms.

Table 2

Object	RCNN	YOLO	SSD
Banana	Detected	Detected	Detected
Chair	Detected	Detected	Detected 2 chairs
Scissors	Detected	Detected	Detected
Cup	Detected	Detected	Detected
Bottle	Detected	Detected bottle and imaginary refrigerator	Detected
2 Cycles	Failed, Detected cycle as dog	Detected two cycles & imaginary sofa, person & chair	Detected only one cycle
Multiple Objects	Detected only 3 out of 6 objects	Detected all 6 objects	Detected only 1 out of 6 objects

Blurred pictures:

The 3 algorithms were run on blurred images 500X667 resolution.

The following is the time taken in seconds by each algorithm to detect the objects.

Table 3

Object	RCNN	YOLO	SSD
Banana	1.14	0.7	0.96
Chair	1.72	0.9	1.02
Scissors	1.78	0.7	0.78
Cup	1.08	0.813	0.72
Bottle	1.08	0.697	1.05
2 Cycles	1.73	0.706	0.72
Multiple	1.09	0.702	0.74

The following is the accuracy of the 3 algorithms.

Table 4

Object	RCNN	YOLO	SSD
Banana	Failed	Failed	Failed
Chair	Failed	Failed	Failed
Scissors	Detected	Failed	Detected
Cup	Detected	Failed	Failed
Bottle	Detected	Failed	Detected
2 Cycles	Failed -detected as dog	Failed	Failed -detected one wheel as train
Multiple Objects	Failed	Failed -detected as person	Failed

Special Cases:

The 3 algorithms were run on images to check for the accuracy of detecting partial imagesand ignoring shadows and reflections.

The following is the time taken in seconds by each algorithm to detect the objects.

Table 5

Object	RCNN	YOLO	SSD
Bottle reflection	1.56	0.703	1.08
Partial man	1.39	0.704	1.85
Partial cyclist	1.18	0.711	1.05
Cup reflection	1.15	0.735	1.05
Dog reflection	0.97	0.886	1.85
Dog shadow-1	1.24	0.72	1.9
Dog shadow-2	1.2	0.713	1.04

The following is the accuracy of the 3 algorithms.

Table 6

Object	RCNN	YOLO	SSD
Bottle reflection	Failed	Failed	Failed
Partial man	Failed	Detected	Failed
Partial cyclist	Detected	Detected	Detected
Cup reflection	Detected cup and bottle and ignored reflection of cup but not bottle	Failed-Detected the original and reflection as well.	Detected cup and bottle and ignored reflection of cup but not bottle
Dog reflection	detected as horse, but ignored the reflection	Failed - Detected as horse	Failed to detect any
Dog shadow-1	Detected, ignored the shadow	Detected, ignored the shadow	Failed to detect any
Dog shadow-2	Detected the dog and failed to ignore the shadow	Detected the dog and detected shadow as sheep	Detected both as one dog

4. Discussion

Directly from Tables 1, 3, and 5 we can notice that YOLOv4 is the fastest, which is followed by SSD and RCNN comes at the last. On the other hand, when it comes to accuracy (from tables 2,4 and 6) YOLOv4 is the most accurate which is separated from the worst, SSD, by RCNN.

It is clear that because to their considerably bigger violet regions, region-based detectors like SSD and R-CNN both have low accuracy. R-CNN, on the other hand, is more accurate than SSD among itself, although SSD, because of its greater mAP values, is more effective for real-time processing applications. As can be seen from its nearly nonexistent violet zones, YOLO is perhaps the most effective of all.

SSD

From last row of Table-1 it can be concluded that SSD performs significantly worse than R-CNN when it comes to tiny objects. This disadvantage is mostly due to the fact that in SSD, tiny object detection is handled by higher resolution layers. These layers contain lower-level information like color patches or edges, making them less effective for classification and lowering SSD performance. The intricacy of SSD's data augmentation suggests another drawback of

Licensed Under Creative Commons Attribution CC BY

this approach, which is that SSD needs a lot of data for training. Depending on the application, this might be quite expensive and time-consuming.

RCNN
This algorithm's accuracy comes at the expense of temporal complexity. It moves far more slowly than similar apps like YOLO. Contrary to YOLOv4, it still needs numerous passes over a single picture, despite advancements over RCNN and Fast RCNN. The convolutional network, Regions of Interest (ROI) pooling layer, and Region Proposal Network (RPN) are just a few of the numerous parts that make up the FRCNN. One of them might prevent the others from moving forward.

YOLOv4
CSPDarknet-53, introduced in YOLOv4, is superior than Darknet-53 since it requires just 66% of the parameters that version 3 did, but produces better results with improved speed and accuracy.

Single-shot detector advantages:
Quicker: Since single-shot detectors do not need a separate proposal generating stage, they are often quicker than multi-shot detectors. For real-time applications or when processing a lot of photos, this might be very crucial.
Simpler: Compared to multi-shot detectors, single-shot detectors are frequently more straightforward and have fewer components, which might make them simpler to use and keep up.

5. Limitations

1) The images used are very few due to limitation on hardware provided by Google colab.
2) Only a few objects were tested that were part of the dataset
3) Due to unavailability of dedicated GPU for the research, the data collected about the speed may not be accurate.

6. Future Exploration

1) The research can further be continued on videos.
2) We can explore the performance of these algorithms with respect to different sized images from very high resolution to very low resolution.
3) We can also explore the performance of these algorithms with respect to the size of the object varying from large to tiny.
4) Research on the performance of these algorithms with respect to refraction of images, low light images etc would throw more light on the ability of these algorithms in real world situations.

Links:
Input and result images:
https://drive.google.com/drive/folders/1nOuhTzcf_QqeYAmz04Lgi7aUsmp7QV6e?usp=drive_link

References

[1] AllegroAdmin1. "Single-Shot vs Two-Shot Detection Meta-Architecture." *ClearML*, 29 Oct. 2022, clear.ml/blog/the-battle-of-speed-accuracy-single-shot-vs-two-shot-detection.
[2] Boesch, Gaudenz. "Object Detection in 2023: The Definitive Guide." *Viso.Ai*, 27 Feb. 2023, viso.ai/deep-learning/object-detection/.
[3] Devanathan, Hari. "The Basics of Object Detection: Yolo, SSD, R-CNN." *Medium*, Towards Data Science, 11 Oct. 2022, towardsdatascience.com/the-basics-of-object-detection-yolo-ssd-r-cnn-6def60f51c0b#:~:text=SSD%20is%20a%20family%20of,detected%20(region%20based%20network). https://medium.com/ibm-data-ai/faster-r-cnn-vs-yolo-vs-ssd-object-detection-algorithms-18badb0e02dc.
[4] Kumari, Kajal. "A Basic Introduction to Object Detection." *Analytics Vidhya*, 5 June 2023, www.analyticsvidhya.com/blog/2022/03/a-basic-introduction-to-object-detection/.
[5] Muhammad Tamjid Rahman. *DRIVING-SCENE IMAGE CLASSIFICATION USING DEEP LEARNING NETWORKS: YOLOV4 ALGORITHM*. spring 2022, http://uu.diva-portal.org/smash/get/diva2:1689752/FULLTEXT01.pdf.
[6] "Object Detection Using Yolov4." *Auriga IT*, 22 May 2023, aurigait.com/blog/object-detection-using-yolov4/#:~:text=Feature%20pyramid%3A%20YOLOv4%20utilizes%20a,improving%20bounding%20box%20regression%2C%20respectively.
[7] Parab, Chinmay U., et al. "Comparison of Single-Shot and Two-Shot Deep Neural Network Models for Whitefly Detection in IOT Web Application." *MDPI*, Multidisciplinary Digital Publishing Institute, 10 June 2022, www.mdpi.com/2624-7402/4/2/34#:~:text=Whereas%20two%2Dshot%20object%20detectors,refinement%20of%20the%20location%20prediction.
[8] Patel, Ashish. "What Is Object Detection?" *Medium*, ML Research Lab, 11 June 2020, medium.com/ml-research-lab/what-is-object-detection-51f9d872ece7.
[9] *What Is Object Detection? Importance, Models and Types - G2*, www.g2.com/articles/object-detection. Accessed 12 Oct. 2023.
[10] Writer, Hossein Ashtari Technical, et al. "Computer Vision Meaning, Examples, Applications." *Spiceworks*, www.spiceworks.com/tech/artificial-intelligence/articles/what-is-computer-vision/. Accessed 12 Oct. 2023.

Licensed Under Creative Commons Attribution CC BY