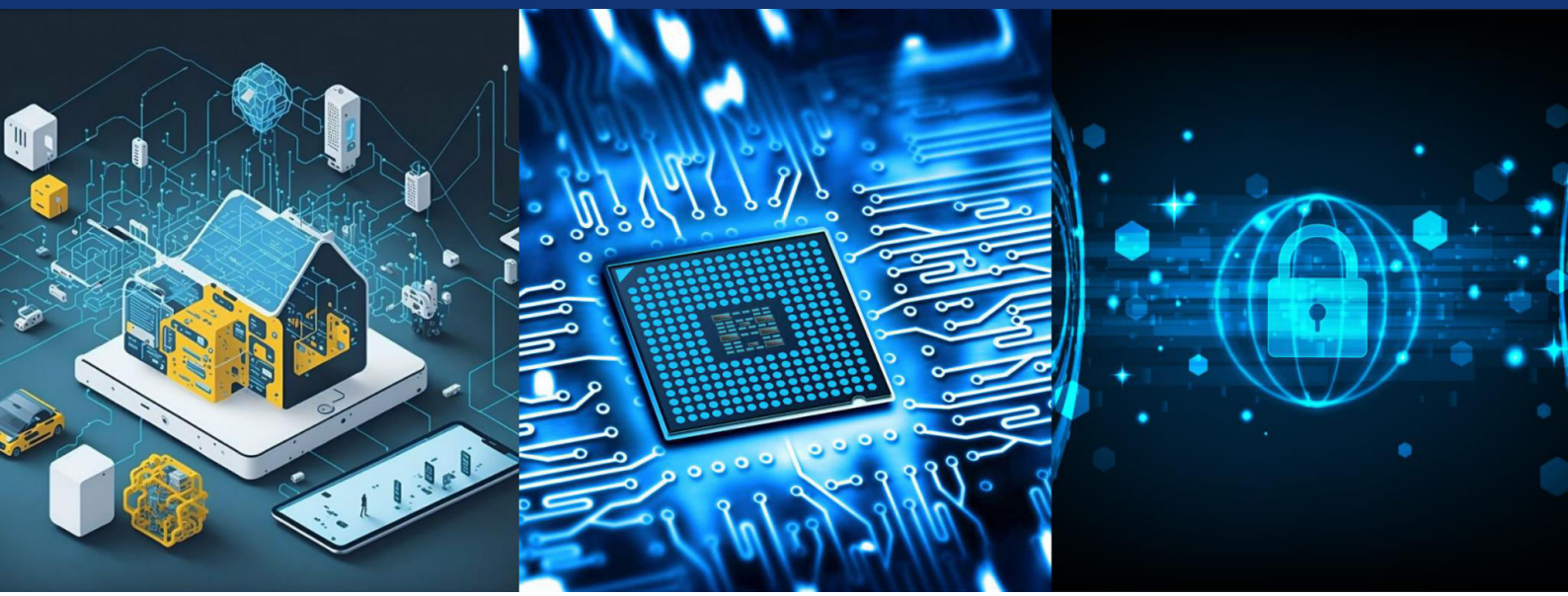
IJIRCCE

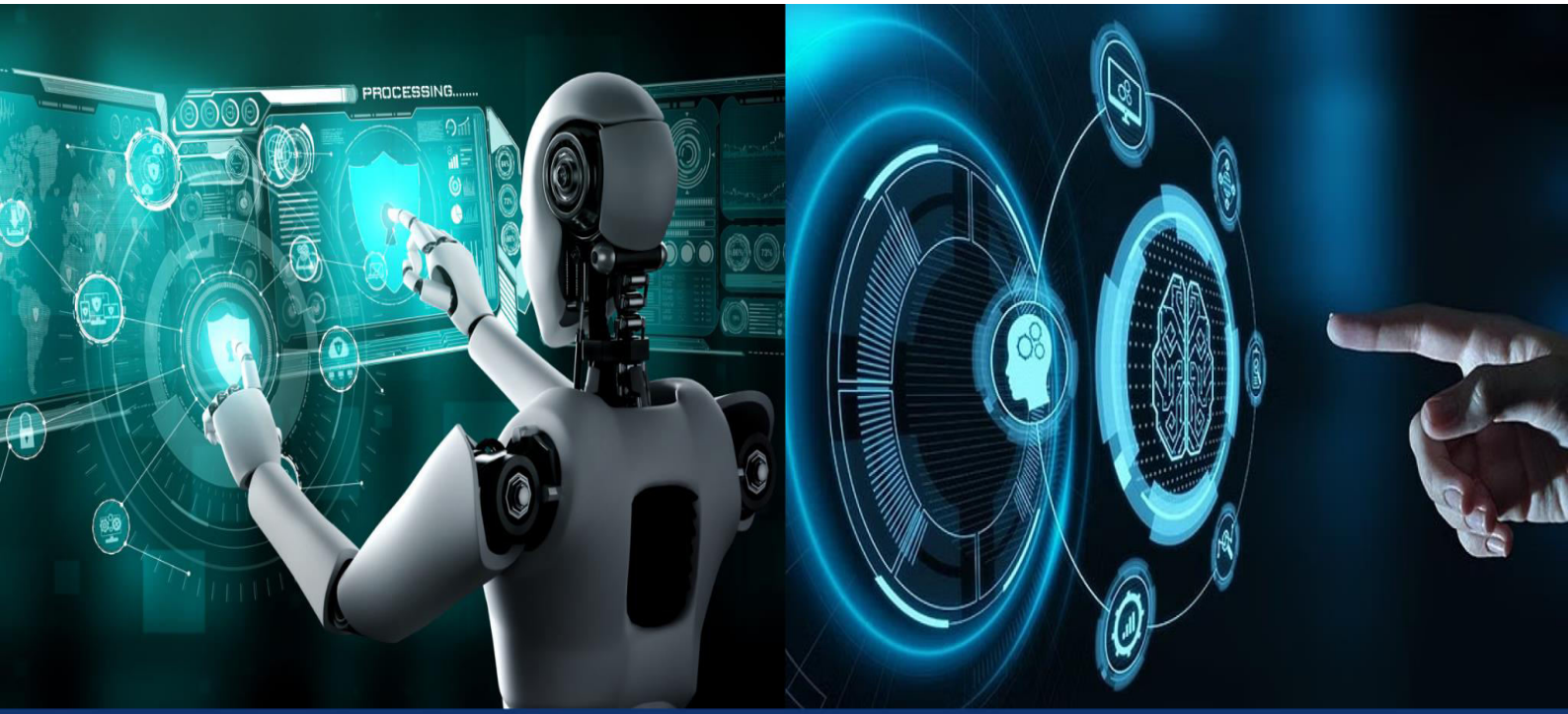
ISSN(O): 2320-9801

ISSN(P): 2320-9798



International Journal of Innovative Research in Computer and Communication Engineering

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



Impact Factor: 8.771

Volume 13, Issue 4, April 2025

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304036

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Real-Time Vehicle Object Detection using YOLO-11 (You Only Look Once) Algorithm for Intelligent Transportation Systems

Subhash Mokase, Dr. B.M.Patil

Department of Computer Science & Engineering, Shreeyash College of Engineering and Technology, Ch. Sambhajinagar, India

Principal, Shreeyash College of Engineering and Technology, Ch. Sambhajinagar, India

ABSTRACT: This paper presents a cutting-edge real-time vehicle detection system utilizing the YOLO-v11 algorithm, achieving 97.25% accuracy and detection speeds of up to 100 frames per second. By leveraging advanced deep learning techniques, the system effectively identifies vehicles in various lighting conditions, orientations, and sizes, enhancing intelligent transportation systems. Performance is rigorously evaluated using precision, recall, and Intersection over Union (IoU) metrics, highlighting the system's potential for improving traffic monitoring and safety in dynamic urban environments.

Unlike traditional methods that adapt classifiers for detection, our approach treats the problem as a regression task, predicting bounding boxes and class probabilities directly from images in a single pass. The YOLOv11 architecture allows for end-to-end optimization, delivering exceptional speed and performance. The optimized Fast YOLOv11 variant processes up to 100 frames per second while achieving double the mean Average Precision (mAP) of other real-time systems. Although YOLO may have a higher localization error rate, it generates fewer false positives and excels in learning generalized object representations, outperforming traditional methods across diverse domains.

KEYWORDS: Convolution neural networks, Computer vision, Deep learning, Object detection, vehicular detection, YOLO (You Only Look Once).

I. INTRODUCTION

Vehicle detection plays a crucial role in various applications, including public safety, security, surveillance, intelligent traffic management, and autonomous driving. However, it poses significant challenges due to the wide variations in vehicle appearance, camera angles, and the presence of severe occlusions. Moreover, weather and lighting conditions further complicate the detection process. Earlier research on vehicle detection primarily focused on specialized designs, such as hand-crafted features and modelling for parts and occlusions [1]. While the proposed methods show reasonable performance, the leading techniques in object detection are now all based on deep neural networks.Intelligent transportation systems (ITS) play a crucial role in enhancing road safety and efficiency by providing real-time vehicle detection capabilities. As urban areas continue to grow, traffic congestion and safety concerns have become pressing challenges that demand effective monitoring solutions. Traditional vehicle detection methods often struggle with accuracy and speed, particularly under varying lighting conditions and vehicle orientations. These limitations hinder the ability of existing systems to respond promptly to dynamic traffic scenarios.

This paper proposes a robust vehicle detection system based on the YOLOv11 algorithm, a state-of-the-art object detection model known for its exceptional speed and accuracy. By leveraging the advanced architecture of YOLOv11, our approach aims to significantly improve real-time vehicle detection on one of the busiest roads in Chhatrapati Sambhaji Nagar, specifically Samrudhi Road. We focus on addressing the unique challenges posed by local traffic conditions, such as fluctuating light levels and diverse vehicle sizes and orientations.

The deployment of this innovative YOLOv11-based system is anticipated to reduce response times for traffic management and enhance overall safety measures. By providing a more reliable and efficient solution for real-time

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304036

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

vehicle detection, our research contributes to the ongoing efforts to optimize traffic flow and ensure safer roads in urban environments. Through rigorous evaluation of the system's performance, we aim to demonstrate its effectiveness as a transformative tool for intelligent transportation systems.

Object detection is a crucial aspect of computer vision, utilizing various machine learning (ML) and deep learning (DL) models to improve performance in detecting and recognizing objects. Traditionally, two-stage object detectors were highly effective and popular. However, recent advancements in single-stage object detection and their associated algorithms have led to significant enhancements, often outperforming many two-stage models. The emergence of YOLO (You Only Look Once) has further transformed the landscape, with numerous applications successfully employing YOLO for object detection and recognition, demonstrating remarkable results compared to two-stage detectors. This motivates us to conduct a focused review of YOLO and its architectural successors, highlighting their design features, optimizations in later versions, and their strong competition against two-stage detectors. This section will provide an overview of deep learning and computer vision fundamentals, explain object detection and relevant terminology, address key challenges, and outline the stages involved in implementing object detection algorithms. We will also discuss the evolution of various object detection methods, review popular datasets in the field, and summarize the primary contributions of this review.

Computer vision within deep learning networks is one of the fastest-growing areas in artificial intelligence. We are now in an era dominated by images and videos, driven by the information explosion facilitated by the Internet and the increasing prevalence of various types of sensors.

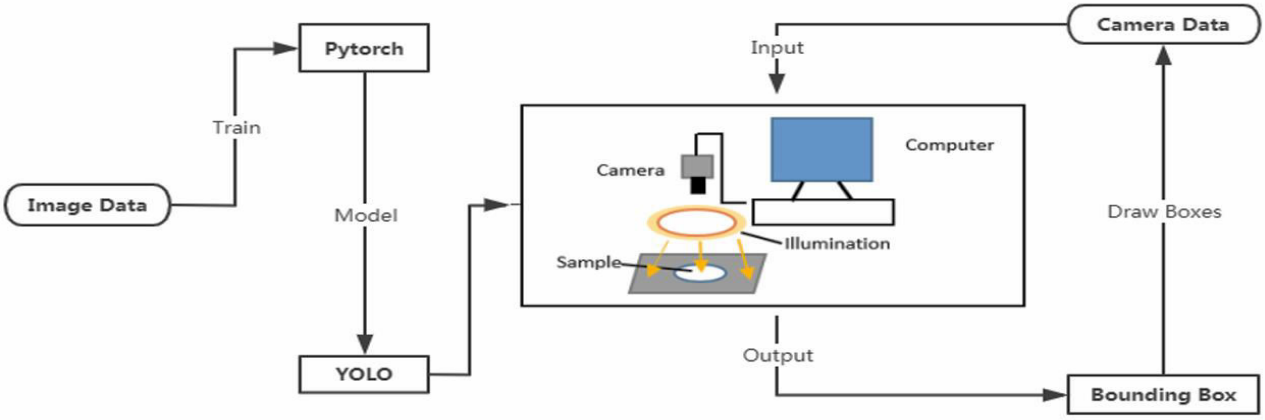


Figure 1. System structure Real time Object Detection using YOLO.

1.1. Deep learning and computer vision

Deep Learning (DL) emerged in the early 2000s, following the rise of Support Vector Machines (SVM), Multilayer Perceptron (MLP), Artificial Neural Networks (ANN), Convolution Neural Network (CNN), Recurrent Neural Networks (RNN) and other shallower neural networks. Researchers have often classified it as a subset of Machine Learning (ML), which in turn falls under the broader category of Artificial Intelligence (AI). Initially, deep learning struggled to attract attention due to challenges such as scalability and the significant computational power required. However, after 2006, it experienced a shift in popularity compared to other machine learning algorithms, driven by two main factors: (i) the vast amounts of data now available for processing and (ii) the emergence of advanced computational resources. Deep learning has since achieved remarkable success across various domains, including weather forecasting [2], stock market prediction [3], speech recognition [4], object detection [5], character recognition [6], intrusion detection [7], automatic landslide detection [8], time series prediction [9], text classification [10], unstructured text data mining with fault classification [11], video processing such as caption generation [12], and many more.

Computer vision is a branch of artificial intelligence (AI) and computer science that aims to enable machines to interpret and understand visual data from the world, similar to how humans perceive it. It involves the development of algorithms and models that allow computers to process, analyze, and derive meaningful insights from images and

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304036

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

videos. Key subfields within computer vision include scene and Image Processing, Image Segmentation, object recognition, object detection, video tracking, object segmentation, pose and motion estimation, scene modelling, and image restoration.[13]. In this review, we concentrate on object detection and its related subfields, including object localization and segmentation, which are among the most significant and widely studied tasks in computer vision. Common deep learning models that can be applied to various computer vision tasks include Convolutional Neural Networks (CNN), Deep Belief Networks (DBN), Deep Boltzmann Machines (DBM), Restricted Boltzmann Machines (RBM), and Stacked Autoencoders [14].

1.2 Object classification and localization Image classification

It involves categorizing an image or an object within an image into one of several predefined categories. This task is typically addressed using supervised machine learning or deep learning algorithms, which require training the model on a substantial labelled dataset. Commonly used models for image classification include Artificial Neural Networks (ANN), Support Vector Machines (SVM), Decision Trees, and K-Nearest Neighbours (KNN) [15]. **Object localization** involves identifying the position of one or more objects within an image or frame by enclosing them in rectangular boxes, commonly referred to as bounding boxes. In contrast, image segmentation is the process of dividing an image into multiple segments, where each segment may represent a complete object or part of an object. This technique is often used to identify objects, lines, and curves, such as the boundaries of an object or segment within an image. Typically, the pixels in a segment share common characteristics like intensity and texture. The primary goal of image segmentation is to create a meaningful representation of the image. Additionally, object detection can be viewed as a combination of classification, localization, and segmentation, involving the accurate classification and efficient localization of one or more objects in an image, usually through supervised algorithms trained on sufficiently large labelled datasets.
Figure 2 presents the clear understanding of classification, localization, and segmentation for single and multiple objects in an image in the context of object detection.

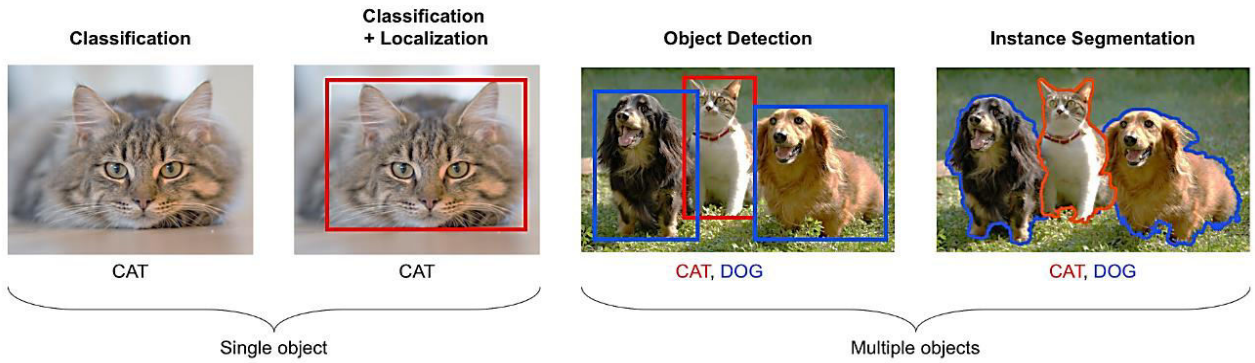


Figure. 2 Classification, Localization, and Segmentation in Single and Multiple Objects image [20].

1.3. Challenges in object detection

Object detection has a wide array of applications, including autonomous driving, aerial object detection, text recognition, surveillance, rescue operations, robotics, face detection, pedestrian monitoring, visual search engines, identifying objects of interest, brand recognition, and much more[16, 17]. The main challenges in object detection include: (i) the inherent variation in object occupancy within an image, where objects may cover a significant portion of the pixels (70% to 80%) or only a small fraction (10% or less); (ii) processing low-resolution visual content; (iii) managing multiple objects of varying sizes within an image; (iv) the availability of labelled data; and (v) dealing with overlapping objects in visual scenes. Many machine learning and deep learning-based object detectors struggle to address these common challenges, which can be summarized as follows:

Viewpoint variation: Objects look different from different angles, making them hard to detect [18].
Deformation: Some objects can change shape, which makes recognition tricky[18].
Occlusion: Partially hidden objects are harder to identify [18].
Illumination Conditions: Different lighting can change how objects appear, affecting their color[18].
Cluttered or Textured Background: Busy backgrounds can obscure objects, making them hard to see[18].
Intra-Class Variation: Different types of the same object can look very different[18].

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Multi-scale training: Detection systems often struggle with objects of varying sizes.
Foreground-Background class imbalance: Imbalance or disproportion among the instances of different categories can majorly affect the model performance.
smaller objects: Models trained on larger objects may perform poorly on smaller ones.
Necessity of large datasets: Deep learning requires extensive datasets, labor-intensive annotations, and strong computational resources [19].
Smaller sized datasets: Deep learning models excel over traditional methods but struggle with small datasets, leading to inaccuracies and localization errors [19].

II. RELATED WORK

Early research in the field primarily relied on traditional machine learning methods. These approaches typically involved manually designing feature extraction techniques, where the accuracy of target detection was heavily influenced by the choice of features and the selected classifier. Additionally, traditional object detection methods often employed computationally expensive sliding window techniques, which made real-time performance challenging.

Histogram of Oriented Gradients (HOG) was introduced by Robert K. McConnell of Wayland Research Inc. in 1986. First, the image is segmented into smaller regions called cells. The Histogram of Oriented Gradients (HOG) is then computed for each pixel within a cell, and the collective set of histograms forms the descriptor. Convolutional Neural Networks (CNNs), the most widely used deep learning algorithms, apply multiple convolutional layers to perform convolutional computations. They are highly effective in feature extraction and offer a more efficient approach for solving object detection problems.

. The CNNs used for target object detection, such as Faster R-CNN, SSD, and YOLO v3,YOLO v4 to YOLO v10 architectures, incorporate the structure of the above-mentioned CNNs used for image classification, and can accomplish both image classification and localization& Segmentation.

Object Recognition Technology based on Deep Learning
Current deep learning-based approaches for target classification and regression can be divided into two main categories. The first is the two-stage algorithm, represented by architectures such as R-CNN, Fast R-CNN, and Faster R-CNN. These methods typically involve two steps: first, generating Region Proposals using selective search or a Region Proposal Network (RPN), and then performing classification and regression on the proposed regions. While this approach offers high accuracy, it often results in slower detection speeds. The second category is the one-stage algorithm, represented by models like SSD and YOLO. These are end-to-end, regression-based object detection algorithms that use a single network to directly predict both the object's bounding box and category probability from the image. Without the need for RPN, these algorithms achieve faster detection speeds, but their performance in detecting small objects may not be as strong as that of two-stage methods. Ultimately, both detection accuracy and speed are crucial for the feasibility of real-time object detection.

YOLO introduces a novel approach to target detection by framing it as a regression problem. Its framework utilizes a simple convolutional neural network (CNN) to directly predict the bounding box positions and the class of objects. YOLOv3 employs a backbone network structure without pooling and fully connected layers, relying instead on adjusting the convolutional kernel's stride for image transformation. It uses Darknet-53 as its network architecture, which deepens the model and enhances feature extraction, leading to improved accuracy compared to YOLOv1 and YOLOv2. Darknet-53 incorporates the ResNet residual structure, effectively mitigating the vanishing gradient problem in deep networks [27].

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304036

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

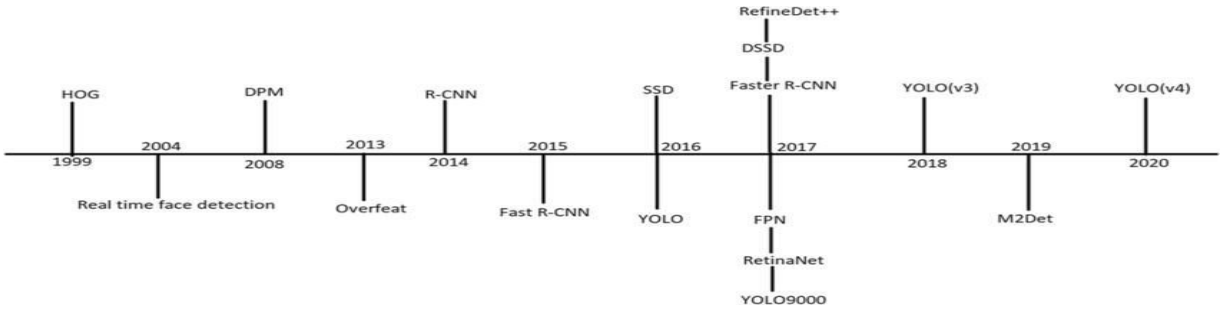


Figure. 3 Year wise evolution of object detection algorithms.

Figure 3 shows the evolution of key object detection algorithms over the years. Because two-stage object detectors are complex and require a lot of resources, researchers have shifted their focus to single-stage detectors, especially the YOLO (You Only Look Once) models. Deep learning is not only successful in object detection but also in many other areas. In healthcare, deep learning is helping to process and analyze medical data. Since the COVID-19 pandemic began, imaging techniques like X-rays, CT scans, and MRIs have been widely used to detect potential viral infections. A summary of how various deep learning models are being used to detect COVID-19 can be found in [28]. Among all the deep learning methods, Convolutional Neural Networks (CNNs) are particularly popular for extracting features from images. For example, a CNN-based hand recognition system has achieved 100% accuracy in both training and testing by using the Crow Search Algorithm (CSA) to find the best hyperparameters[28].

III. MATERIALS & METHODS

3.1 YOLO Architecture

The YOLO algorithm processes an image to detect objects using a straightforward deep convolutional neural network. The architecture of the CNN that serves as YOLO's backbone is illustrated below.

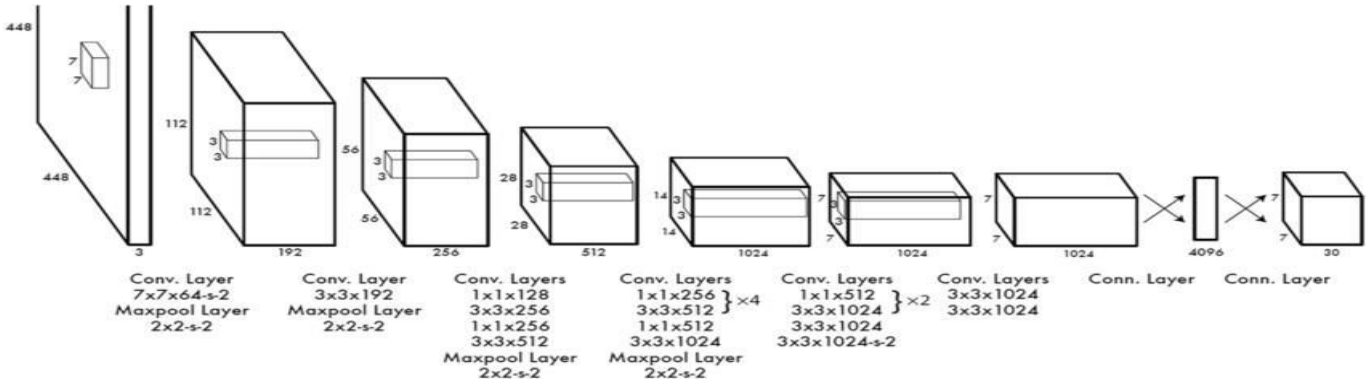


Figure 4. YOLO architecture for object detection and localization[23].

Figure 4: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1 × 1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224 × 224 input image) and then double the resolution for detection.

The first 24 convolution layers of the model are pre-trained using ImageNet by plugging in a temporary average pooling and fully connected layer. Then, this pre-trained model is converted to perform detection since previous research showcased that adding convolution and connected layers to a pre-trained network improves performance. YOLO's final fully connected layer predicts both class probabilities and bounding box coordinates. YOLO divides an input image into an S × S grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object. Each grid cell predicts B bounding boxes and confidence

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304036

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

scores for those boxes. These confidence scores reflect how confident the model is that the box contains an object and how accurate it thinks the predicted box is.

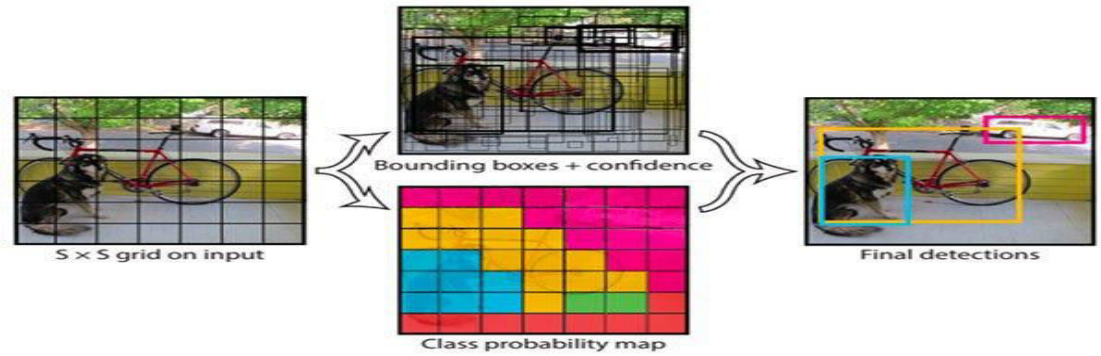


Figure 5. Shows the working of YOLO model [24].

Each bounding box has 5 additional values apart from class probabilities which are as follows:
Pc: confidence of an object being present in the bounding box
bx, by: Center coordinates of the object present if any. This value is "do not care" if no object is present.
bh , bw : height and width of the object present. This value is "do not care" if no object is present.
C: Class of the object being detected.

Non-max Suppression
NMS is a post-processing technique that enhances detection accuracy by removing redundant bounding boxes. It consolidates overlapping predictions to provide a single bounding box for each object in the image, improving efficiency.The concept of non-max suppression comes when the same object is present in several grid cells. The algorithm might then detect several bounding boxes around the same object with different confidence scores. Non-max suppression suppresses the values (detection) where Pc values are below a certain threshold. First, it selects the highest Pc or IOU value and suppress all the values less than that. Then selects the second highest and continue this process.

3.2 Model Architecture of YOLOv11 Variants

YOLOv11 has several variants to cater to different computational resources and application needs. These variants are labeled as N, S, M, L, and X, each corresponding to different model sizes and levels of complexity:

- **YOLOv11N (Nano):** Very small and lightweight, perfect for devices with limited resources.
- **YOLOv11S (Small):** Good for mobile devices; balances speed and accuracy for real-time use.
- **YOLOv11M (Medium):** A versatile model that works well for many tasks, striking a balance between performance and resource use.
- **YOLOv11L (Large):** A powerful model designed for high accuracy; great for detailed detection.
- **YOLOv11X (Extra Large):** Built for challenging environments with many objects; focuses on achieving the highest accuracy.

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed T4 TensorRT10 (ms)	params (M)	FLOPs (B)
YOLO11n	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304036

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

3.3 Methodology

NMS-Free Training: Uses a dual label assignment strategy (one-to-many and one-to-one) to avoid Non-Maximum Suppression (NMS), improving model supervision and performance.
Consistent Matching Metric: Combines bounding box overlap (IoU) and spatial priors to assess how well predictions match ground truth, aligning different branches of the model for better optimization.
Lightweight Classification Head: Employs depthwise separable convolutions to reduce computational load, making the model faster and more efficient for real-time applications.
Spatial Channel Decoupled Downsampling: Uses pointwise and depthwise convolutions to efficiently reduce image size without significantly increasing parameters or computations.
Rank Guided Block Design: Adapts block design by adjusting redundant parts of the model based on performance, ensuring efficiency across different stages.
Large Kernel Convolutions: Utilizes larger kernels in deeper layers to enhance performance while managing latency and feature quality.
Partial Self Attention (PSA): Incorporates self-attention selectively on parts of the feature map, improving global understanding without heavy computation.

IV. RESULTS AND DISCUSSION

YOLOv1: Groundbreaking for speed and simplicity; struggled with small objects and localization accuracy.
YOLOv2 (YOLO9000): Recognized over 9000 object categories; improved accuracy and performance.
YOLOv3:Enhanced feature pyramids for better detection accuracy; slower than later versions but good for accuracy.
YOLOv4: Maximized speed and accuracy, ideal for real-time applications; balanced performance improvements.
YOLOv5: Not officially released by original creators; popular for ease of use and lightweight efficiency.
YOLOv6 and YOLOv7:Further improvements in architecture and training methods; enhanced capabilities.
YOLOv8 and YOLOv10: Introduced sophisticated methods for various object detection challenges; latest versions with notable advancements.

YOLOv11:
Efficiency and Performance: YOLOv11 shows significant improvements with 28% to 57% fewer parameters and 23% to 38% fewer calculations compared to its predecessors.

Average Precision (AP): The model variants (N/S/M/L/X) achieve an AP increase of 1.2% to 1.4%, enhancing overall detection accuracy.

Real-Time Application: It boasts 37% to 70% shorter latencies, making YOLOv11 particularly suitable for real-time applications.

Cost vs. Accuracy: YOLOv11 outperforms earlier YOLO models, with YOLOv11N and S improving AP by 1.5 and 2.0 over YOLOv63.0N and S while using fewer parameters.

Comparison with GoldYOLOL: YOLOv11L demonstrates a 1.4% AP improvement with 32% less latency and 68% fewer parameters compared to GoldYOLOL.

Against RTDETR: YOLOv11S and X outperform RTDETRR18 and R101 by 1.8× and 1.3×, respectively, while maintaining comparable performance levels.

These results demonstrate the state-of-the-art performance and efficiency of YOLOv11 across several model scales, highlighting its supremacy as a real-time end-to-end detector. The impact of our architectural designs is confirmed when this effectiveness is further validated by utilizing the original one-to-many training approach.

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304036

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

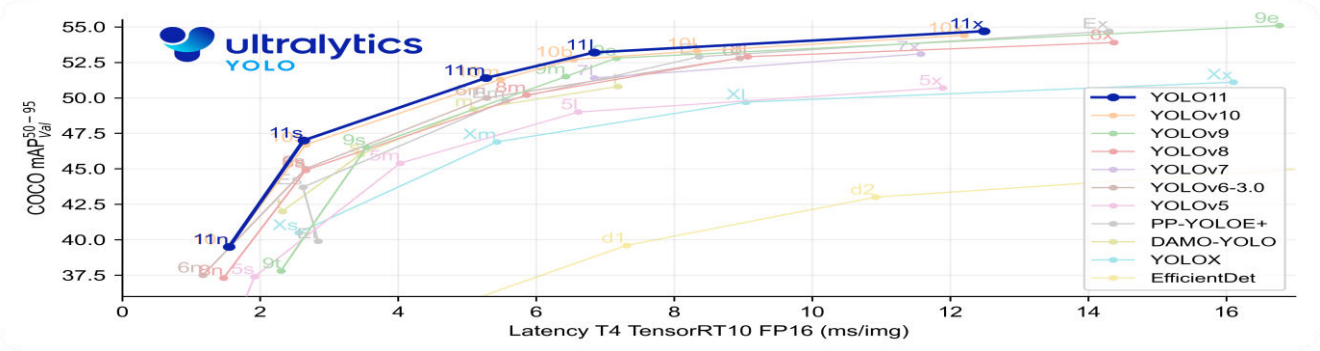


Figure 7: Comparisons with others in terms of latency-accuracy and size-accuracy trade-offs. We measure the end-to-end latency using the official pre-trained models.

4.1 Convolutional neural networks and pretrained models

Traditional machine learning algorithms depend on manually designed feature extraction and selection for making predictions and classifications. This process often consumes a lot of time in finding the most effective feature extraction techniques. To overcome these limitations, researchers, industry experts, and academics are turning their attention to deep learning. Popular deep learning models include deep neural networks, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and their variants such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), along with Generative Adversarial Networks (GANs) and different types of auto encoders. Because of scaling challenges in deep neural networks, CNNs are commonly used to effectively capture spatial and contextual information with fewer parameters. In the case of high-dimensional inputs like images, it is not practical to connect every neuron in one layer to all neurons in the previous layer; instead, each neuron is connected only to a subset of the previous layer's neurons.

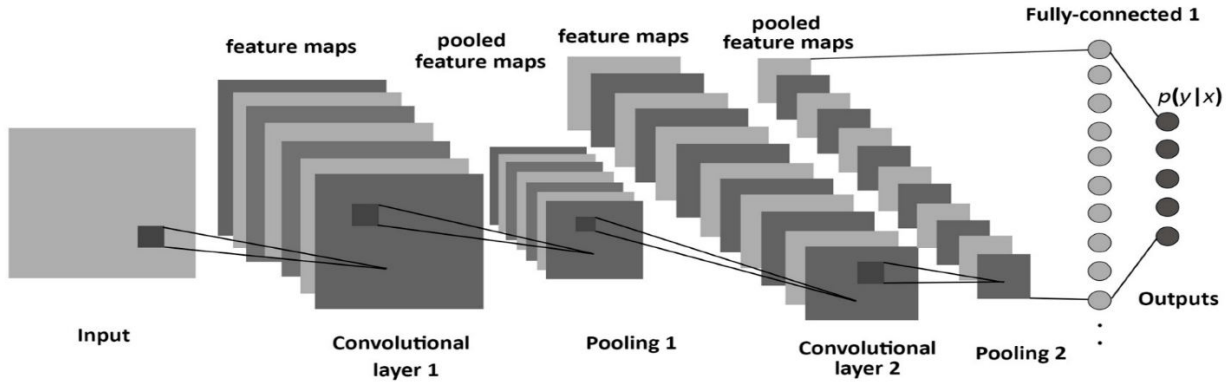


Figure 8: Architecture of Convolutional neural networks[21].

CNN models generally follow a well-defined structure that alternates convolutional and pooling layers, with pooling often placed after convolutional layers. The architecture typically ends with a few fully connected layers, culminating in a softmax classifier. These networks are trained using backpropagation along with Stochastic Gradient Descent (SGD) to optimize weights and biases, aiming to minimize a specific loss function and effectively map inputs to the desired outputs.

In convolutional layers, a set of learnable kernels or filters is employed to extract local features from the input. Each kernel produces a feature map by performing a dot product—similar to a convolution—while sliding over the input. This is followed by applying a non-linear activation function to introduce complexity into the model. The units in each feature map connect only to a small region of the input known as the receptive field. By sharing weights across all units in a feature map, CNNs not only reduce the number of parameters but also enable the model to recognize the same features regardless of their position within the input[22].

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304036

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

4.2. System Designing

Figure 1 shows the architecture diagram of the whole system. We trained the model based on the collected data, obtained the training file of the model using YOLO in the PyTorch (Tenserfolw&keras) environment, and used the model for commodity detection and identification in the actual production environment.

Preparation of training sets

The performance of object target detection during training is closely tied to the size of the training dataset. A small dataset can lead to a decrease in training error but an increase in testing error, a situation referred to as "overfitting." To mitigate this, an initial training set consisting of 1,000 items, each with 12 images taken from various angles, was used. A Python script was employed to augment the dataset by applying rotations of 90°, 180°, and 270°, along with horizontal flips, effectively tripling the number of images. This resulted in a total of 1,000 * 12 * 5 = 60,000 training samples.

Preparation of training set tags

After generating the training set for real-time vehicle object detection, we also need to create the corresponding LABEL tags for each vehicle. During the preparation of the training set, we collected barcode labels for each vehicle simultaneously. The 60,000 images are named using their vehicle identification codes, and we utilize Python scripts to compile this information. The result is a TXT file where each line contains the path to an image along with its associated barcode label, facilitating accurate classification during detection.

Preparation of the test suite

The purpose of the test set in real-time vehicle object detection is to evaluate the recognition performance of the generated training set. It is crucial that the test set does not contain any images from the training set, as this would compromise the validity of the evaluation. Using Python's built-in random functions, we randomly allocate the initial images to create the test set, ensuring a division of 30% for testing and 70% for training. This process maintains the integrity of the testing requirements.

Preparation of test set tags

To prepare the test set for real-time vehicle object detection, we need to ensure that the corresponding labels are divided alongside the test items. This means that for each object in the test set, there should be a matching label stored in a file named label_test.txt. These labels serve as classification markers, created through manual annotation, to accurately identify and classify the detected vehicles in real time.

The training process

The objective of training for real-time vehicle object detection is to enhance the mean Average Precision (mAP), a key metric for evaluating prediction accuracy in object detection. mAP assesses how well the model performs across various vehicle categories by generating precision-recall curves for each category. The Average Precision (AP) represents the area under these curves, while mAP aggregates the performance across all categories.

After preparing the training set, images are resized to 416x416 pixels and processed through a CNN to create a 13x13 feature map. Each cell in this feature map contains parameters and confidence scores (x, y, h, z, conf) that indicate the presence and type of vehicles within that region.

The training dataset comprises 60,000 JPG images paired with 60,000 XML annotation files, formatted similarly to the VOC dataset. These XML files follow a consistent naming convention from 00001.xml to 60000.xml, containing manually annotated relative position information for each vehicle (e.g., 0.5 0.6 0.3 0.2).

The dataset is randomly split into 70% for training (42,000 images) and 30% for testing (18,000 images), covering a total of 1,000 vehicle categories. Once training is complete, the YOLOv2 model outputs pixel-level detection information. For example, in the image 00011.jpg, a detected vehicle might have a confidence score of 0.0075 and be located at the coordinates 336.25684, 125.2356, 268.6589, 165.2547.

To accurately evaluate mAP, the system retrieves the corresponding information from the XML files to ensure alignment between predicted and actual data. The initial training uses a subset of 5,000 images, with a default iteration count of 45,000. Utilizing an Intel i7 processor and a GTX 2080 Ti graphics card, the training typically spans about three days.

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304036

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Following this, a file containing 1,000 barcodes is processed, allowing the system to label vehicle information directly on the images, with Bounding Boxes displaying product information in the upper left corner.

The testing process

The testing process is quite simple. After setting up the required environment, you just need to run the test program once the weights file has been generated from training. The results are displayed in

Table 1. Training process parameters.

ID	Rps/Img	IOU	Recall	Precision
101	1.07	82.00%	95.50%	97.25%
102	1.07	82.11%	95.54%	97.27%

We define the parameters as follows: **ID** represents the iteration number, indicating how frequently the training program updates the weights file; **Correct** denotes the number of correctly selected boxes. After processing the data for an image, IMGnet predicts the box selections for different objects, compares the ground truth for each object with all predicted box selections, and calculates the Intersection over Union (IoU). If the maximum IoU value exceeds a preset threshold, one is added to the **Correct** parameter. **RPS/IMG** stands for Region Proposal per Image, which indicates the average number of frames predicted per map. This parameter's value is influenced by a threshold, set to 0.001 in the YOLO_RECALL function. Consequently, more boxes are selected, leading to some non-object elements being incorrectly identified as objects. However, this approach increases the recall value, which enhances recognition accuracy and reduces the chances of failing to recognize objects in the image. The Precision value is calculated as the ratio.

Precision = TP/(TP+FP)

Meaning for TP(True positives)& FP(False positives), that is, the number of network consensus out of the object; TP :True positives, predict the number of correct positions; Recall: The ratio of the numberof correctly identified objects of a certain class to the total number of objects of the same type in the test set. The denominator is True positives + False negatives, which can be understood as the number of a certain type of object.

Recall = TP/(TP+FN)

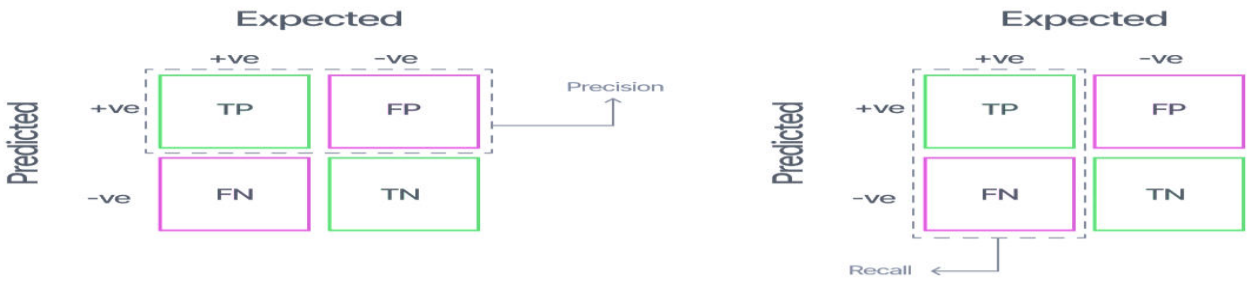


Figure9: Regions of the confusion matrix that help compute precision and recall [25].

Intersection Over Union (IOU): refers to the coincidence degree between the predicted object position obtained through the network and the original manually marked object position. The value of IoU can be obtained through the set operation of DetectionResult and GroundTruth:

$$\mathrm{IoU} = \frac{\mathrm{DetectionResult} \cap \mathrm{GroundTruth}}{\mathrm{DetectionResult} \cup \mathrm{GroundTruth}}$$

The IoU measures the accuracy of our detections. Given a ground-truth bounding box and a detected bounding box, we compute the IoU as the ratio of the overlap and union areas**:**

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304036

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Figure10: Intersection Over Union for Object Detection[26].

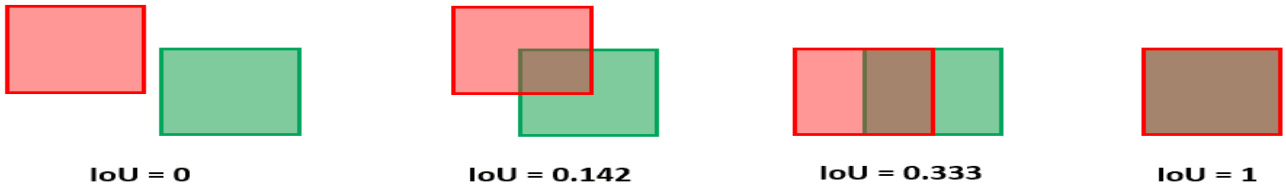


Figure11: Intersection over Union for Object DetectionResults [26].

The IoU can have any value between 0 and 1. If two boxes do not intersect, the IoU is 0. On the other hand, if they completely overlap, the intersection and the union areas are equal. So, in that case, the IoU is 1.Therefore, the higher the IoU, the better the prediction of an object detection system.

V. PERFORMANCE EVALUATION AND IMPROVEMENTS

Adjusting the threshold allows for maintaining an overall object recognition rate of approximately 98%, with potential for further enhancement, as indicated in Table 2. A lower threshold results in more successful predictions, thereby increasing the recall rate. The precision-recall curve assists in finding a balanced compromise between precision and recall for optimal system effectiveness. Changes in precision and recall values become evident when the threshold is modified. An effective classification strategy seeks to maximize correct identifications while minimizing false positives, achieving a significant increase in recall without compromising precision. In contrast, some less effective approaches may sacrifice precision to boost recall, but the precision-recall curve effectively mitigates this issue. By leveraging this curve, it is possible to achieve high accuracy while keeping the recall rate around 40%. However, at a 100% recall rate, accuracy may decline to 50%.

Table 2. Training confidence threshold.

Retrieval Cutoff	Precision	Recall
Train_confidence=0.25	98.85%	20%
Train_confidence=0.30	95.50%	40%
Train_confidence=0.35	66.34%	40%
Train_confidence=0.40	73.87%	60%
Train_confidence=0.45	60.38%	60%
Train_confidence=0.50	65.65%	62%
Train_confidence=0.55	55.32%	80%

© 2025 IJIRCCE | Volume 13, Issue 4, April 2025| DOI: 10.15680/IJIRCCE.2025.1304036

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Test results:

```
Found 10 boxes for img.jpg
person 0.69 (1027, 270) (1091, 343)
car 0.70 (587, 352) (817, 539)
car 0.74 (390, 82) (509, 194)
car 0.74 (507, 193) (653, 327)
car 0.75 (259, 455) (490, 621)
car 0.77 (243, 265) (431, 432)
truck 0.78 (634, 30) (915, 269)
car 0.80 (139, 58) (255, 160)
car 0.83 (216, 146) (369, 267)
car 0.83 (771, 216) (945, 351)
```

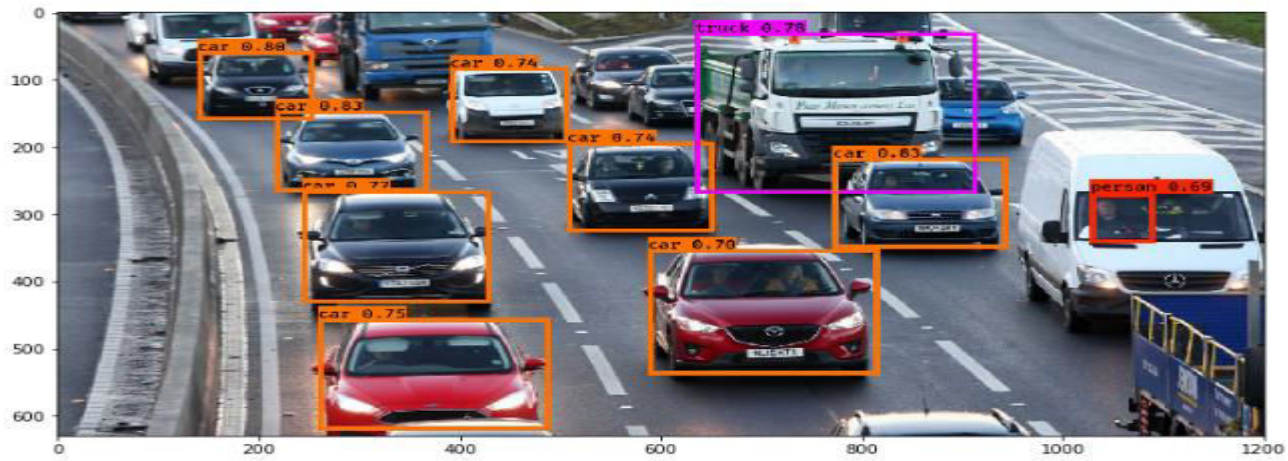


Figure 12. Test results.

Final Results:

Figure 13. Final Results.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Result of figure 13:
image 1/1 **C:\Users\Subhash_Mokase\OneDrive\Desktop\Object_detection\samruddhi_highway.webp:** 384x640 3 persons, 9 cars, 1 motorcycle, 813.6ms
Speed: 43.5ms pre-process, 813.6ms inference, 10.8ms postprocess per image at shape (1, 3, 384, 640)
Future developments aim to improve detection speed and accuracy, also detecting unauthorized vehicles and send fine to them using their registered vehicle numbers.

VI. CONCLUSIONS AND FUTURE SCOPE

The YOLOv11 algorithm is a powerful solution for real-time vehicle detection, surpassing earlier versions in both accuracy and speed. Its innovative techniques make it ideal for applications in autonomous driving and traffic management. Our vehicle detection system operates on Samruddhi Road in Chhatrapati Sambhajinagar, utilizing YOLO, a widely recognized computer vision algorithm.Developed in Python, the system combines advanced object recognition with practical AI applications. While Python's high-level nature can slow execution, its readability and flexibility are advantageous as computational power improves. Integrating computer vision and natural language processing is expected to boost productivity across industries, despite ongoing challenges in recognition speed. By leveraging YOLO's efficiency, we enhance the accuracy and speed of vehicle identification in various conditions.YOLOv11 represents a significant advancement in real-time object detection, achieving top-tier performance while maintaining efficiency. Its versatility makes it suitable for diverse applications, including driverless vehicles and healthcare solutions. Understanding YOLOv10's capabilities and limitations opens new opportunities for researchers and industry professionals.

Future work aims to enhance the vehicle detection system through several key initiatives. This includes integrating tracking algorithms to monitor vehicles and animals across frames, expanding detection capabilities to include pedestrians and cyclists, and optimizing the system for deployment on embedded devices for real-time processing. Developing a multi-camera fusion system and improving robustness in adverse weather conditions will further enhance accuracy. Additionally, techniques for night time detection and transfer learning will be explored, alongside the creation of a real-time video processing pipeline and edge computing architectures to reduce latency. The project will also focus on autonomous vehicle applications, advanced data augmentation, and multi-task learning for simultaneous object detection. Future developments will involve collecting extensive datasets, improving detection speed and accuracy, and creating modules for unauthorized vehicle and animal detection. Finally, collaborations with industry partners, academic institutions, and government initiatives will be pursued to advance intelligent transportation systems and enhance road safety.

REFERENCES

1. S. Sivaraman and M. M. Trivedi. Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behaviour analysis. Intelligent Transportation Systems, IEEE Transactions on, 14(4):1773–1795, 2013.
2. Zaytar MA, El Amrani C (2016) Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. Int J Compute Apply 143.
3. Rather AM, Agarwal A, Sastry VN (2015) Recurrent neural network and a hybrid model for prediction of stock returns. Expert System Apply 42(6):3234–3241.
4. Sak H, Senior A, Rao K, Beaufays F (2015) Fast and accurate recurrent neural network acoustic models for speech recognition. ArXiv preprint arXiv: 1507.06947
5. Liang M, Hu X (2015) recurrent convolutional neural network for object recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3367–3375. https://doi.org/10.1109/CVPR.2015.7298958
6. Zhang XY, Yin F, Zhang YM, Liu CL, Bengio Y (2017) Drawing and recognizing Chinese characters with recurrent neural network. IEEE Trans Pattern Anal Mach Intell 40(4):849–862
7. Kim J, Kim J, Thu HLT, Kim H (2016) Long short-term memory recurrent neural network classifier for intrusion detection. In: 2016 international conference on platform technology and service (PlatCon),https://doi.org/10.1109/PlatCon.2016.7456805

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.771| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

8. Mezaal MR, Pradhan B, Sameen MI, Shafri M, Zulhaidi H, Yusoff ZM (2017) Optimized neural architecture for automatic landslide detection from high resolution airborne laser scanning data. Appl Sci 7(7):730, https://doi.org/10.3390/app7070730
9. Che Z, Purushottam S, Cho K, Sontag D, Liu Y (2018) Recurrent neural networks for multivariate time series with missing values. Sci Rep 8(1):1–12
10. Lai S, Xu L, Liu K, Zhao J (2015) Recurrent convolutional neural networks for text classification. In: Twenty-ninth AAAI conference on artificial intelligence.
11. Wei D, Wang B, Lin G, Liu D, Dong Z, Liu H, Liu Y (2017) Research on unstructured text data mining and fault classification based on RNN-LSTM with malfunction inspection report. Energies 10(3):406. https://doi.org/10.3390/en10030406
12. Xu N, Liu AA, Wong Y, Zhang Y, Nie W, Su Y, Kankan Halli M (2018) Dual-stream recurrent neural network for video captioning. IEEE Trans Circuits System Vid Techno 29(8):2482–2493. https://doi.org/10.1109/TCSVT.2018.2867286
13. Morris T (2004) Computer Vision and Image Processing, Palgrave Macmillan Ltd, 1st edition, pp 1–320
14. Voulodimos A, Doulamis N, Doulamis A, Protopapadakis E (2018) Deep learning for computer vision: a brief review. Compute Intell Neuroscience 2018:1–13
15. Thai LH, Hai TS, Thuy NT (2012) Image classification using support vector machine and artificial neural network. Int J Inform Technical Compute Sci 4(5):32–38
16. Agarwal S, Terrail JO, Jurie F (2018) Recent advances in object detection in the age of deep convolutional neural networks. arXiv preprint arXiv:1809.03193.https://doi.org/10.48550/arXiv.1809.03193
17. Rey J (2017) Object detection with deep learning: the definitive guide
18. https://xailient.com/blog/6-problems-that-you-can-overcome-with-object-detection/
19. Liu L, Ouyang W, Wang X, FieguthP, Chen J, Liu X, Pietikäinen M (2020) Deep learning for generic object detection: a survey. Int J Compute Vis 128(2):261–318.https://doi.org/10.48550/arXiv.1809.02165
20. Detection or localization and segmentation https://www.oreilly.com/ library/view/deep-learning-for/9781788295628/ 4fe36c40-7612-44b8-8846-43c0c4e64157.xhtml
21. Albelwi S, Mahmood A (2017) A framework for designing the architectures of deep convolutional neural networks. Entropy 19(6):242.
22. Chen, T.; Xu, R.; He, Y.; Wang, X. A gloss composition and context clustering based distributed word sense representation model. Entropy **2015**, *17*, 6007–6024.
23. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In proceedings of the IEEE conference on computer vision and pattern recognition, pp 779-788.
24. Object Detection using yolo https://medium.com/analytics-vidhya/object-detection-using-yolo-and-car-detection-implementation-1ec79e882875.
25. Precision and Recallhttps://www.v7labs.com/blog/precision-vs-recall-guide
26. Intersection over Union for Object Detection. https://www.baeldung.com/cs/object-detection- intersection-vs-union
27. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587.
28. Bhattacharya S, Maddikunta PKR, Pham QV, Gadekallu TR, Chowdhary CL, Alazab M, Piran MJ (2021) Deep learning and medical image processing for coronavirus (COVID-19) pandemic: a survey. Sustain Cities Soc 65:102589. https://doi.org/10.1016/j.scs.2020.102589.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details