Global Journal of Engineering Innovations & Interdisciplinary Research

Original Article

Traffic Sign Recognition Using YOLO And RCNN

B Prasad Yadav[1], Jagat [2], Jahnavi Sivani Motapotula[2], Teja Sai Koka[2]

[1]Assistant Professor, Department of Artificial Intelligence and Data Science, Vignan Institute of Technology and Science, Hyderabad, India

[2]UG Student, Department of AI&DS, Vignan Institute of Technology and Science, Hyderabad, India

Correspondence

B. Prasad Yadav

Assistant Professor , Department of Artificial Intelligence and Data Science, Vignan Institute of Technology and Science, Hyderabad, India

- Received Date: 25 May 2025
- Accepted Date: 15 June 2025
- Publication Date: 27 June 2025

Keywords

Real-time Editing, Knowledge Sharing, Syntax Highlighting, Debugging Tools, Collaboration Tools, Code Quality, Programming Languages, Server, Client, Group Chat

Abstract

The "Traffic Sign Recognition with YOLO-RCNN" system, investigates a novel approach to traffic sign recognition for advanced driving systems by combining the speed and efficiency of the You Only Look Once (YOLO) object detection framework with the precision and accuracy of Region-based Convolutional Neural Networks (R-CNN). YOLO excels at rapidly identifying multiple traffic signs within an image, providing an initial set of detections. However, its initial bounding box predictions can sometimes be imprecise. To address this limitation, R-CNN is integrated as a post-processing step. R-CNN refines the initial detections by adjusting the bounding box coordinates and significantly improving the accuracy of traffic sign classification. This hybrid approach leverages the strengths of both architectures, resulting in a system that achieves a superior balance between speed and accuracy. The system's performance was rigorously evaluated on a diverse dataset encompassing a wide range of traffic sign types and challenging environmental conditions, such as varying lighting and occlusions. Experimental results demonstrate a significant improvement in both detection accuracy and processing speed compared to traditional methods. This innovative approach not only enhances the performance of traffic sign recognition systems but also paves the way for more robust and reliable solutions for autonomous driving applications. Future research directions include: optimizing the model architecture through techniques such as network pruning and quantization to further improve inference speed and computational efficiency, expanding the training dataset to include more diverse and challenging scenarios, such as adverse weather conditions and complex urban environments, to enhance the model's generalization capabilities, and exploring the integration of advanced deep learning techniques, such as attention mechanisms and transformer architectures, to further improve the model's ability to focus on salient features and enhance its overall performance.

Introduction

The Traffic Sign Recognition with YOLO-RCNN system is a Python-based application designed to perform real-time object detection on video footage using the YOLO (You Only Look Once) and RCNN (Region-based Convolutional Neural Networks) deep learning models. YOLO is a state-of-the-art, real-time object detection system that is known for its speed and accuracy. This particular implementation focuses on detecting traffic-related objects, specifically traffic lights and stop signs, within a video file. The script processes each frame of the input video, identifies objects of interest, and provides both visual cues (bounding boxes and labels) and auditory alerts for detected traffic lights and stop signs. This functionality can be particularly useful in applications such as autonomous driving systems, traffic monitoring, and safety alert systems. By adjusting the confidence and threshold parameters, users can fine-tune the detection sensitivity and accuracy to suit their specific needs. The script is designed to be run in a local environment where OpenCV can display graphical windows, making it an interactive tool for exploring object detection capabilities.

The purpose of the described research is to introduce and evaluate a new system for traffic sign recognition designed for advanced driving systems (ADS). This system, named "Traffic Sign Recognition with YOLO-RCNN," aims to improve upon existing methods by combining the strengths of two different object detection architectures: YOLO (You Only Look Once) and R-CNN (Region-based Convolutional Neural Networks). The overall goal is to create a more robust and reliable traffic sign recognition system that can function effectively in real-world driving conditions.The primary objective is to develop a hybrid traffic sign recognition system that achieves a better balance between speed and accuracy compared to traditional methods. This is accomplished by leveraging YOLO for its speed in initially detecting potential traffic signs and then using R-CNN for its precision in refining those detections and improving classification accuracy. The research also aims to demonstrate the effectiveness of this combined approach through rigorous evaluation on a diverse dataset, including

Citation: Yadav MP, Jagat, Motapotula JS, Koka TS. Traffic Sign Recognition Using YOLO And RCNN. GJEIIR. 2025;5(4):071.

challenging environmental conditions. Further objectives for future research are also outlined, focusing on optimizing the model, expanding the dataset, and exploring more advanced deep learning techniques.

The problem being addressed is the need for more efficient and accurate traffic sign recognition systems in the context of advanced driving systems. While existing methods may excel in either speed or accuracy, there's a need for a solution that provides both. With that said, YOLO, while fast, can sometimes produce imprecise bounding box predictions. This imprecision can lead to errors in traffic sign classification, which is unacceptable for safety-critical applications like autonomous driving. Therefore, the core problem is to develop a system that can rapidly and accurately identify and classify traffic signs in diverse and challenging real-world driving scenarios.

Methodology

The methodology of the "Traffic Sign Recognition with YOLO-RCNN" system combines the strengths of two distinct object detection architectures: YOLO (You Only Look Once) and R-CNN (Region-based Convolutional Neural Networks), to achieve a balance between speed and accuracy in traffic sign recognition for advanced driving systems (ADS).

Here's a breakdown of the methodology:

1. **Initial Detection with YOLO:** The process begins with YOLO, a real-time object detection system known for its speed. YOLO processes the entire image at once, dividing it into a grid and predicting bounding boxes and class probabilities for each grid cell. This allows for rapid identification of potential traffic signs within the image, providing an initial set of detections.
2. **Refinement with R-CNN:** While YOLO is fast, its initial bounding box predictions can sometimes lack precision. To address this, R-CNN is employed as a post-processing step. R-CNN focuses on regions of interest (ROIs), in this case, the bounding boxes generated by YOLO. It extracts features from these ROIs using a convolutional neural network and then refines the bounding box coordinates, leading to more accurate localization of the traffic signs. Furthermore, R-CNN improves the accuracy of traffic sign classification by performing a more detailed analysis of the features within each ROI.
3. **Hybrid Approach:** The core of the methodology lies in this hybrid approach. By combining YOLO's speed for initial detection with R-CNN's precision for refinement, the system aims to achieve both rapid processing and high accuracy. This addresses the limitations of using either architecture alone.
4. **Evaluation:** The system's performance is rigorously evaluated on a diverse dataset containing various traffic sign types and challenging conditions, including varying lighting and occlusions. This ensures the system's robustness and ability to perform in real-world driving scenarios. The evaluation likely involves metrics such as detection accuracy (e.g., precision, recall, F1-score) and processing speed (e.g., frames per second).

In simpler terms, imagine a fast scout (YOLO) quickly identifying potential targets in a large area. Then, a more precise marksman (R-CNN) carefully examines each target identified by the scout, ensuring accurate identification and location. This combined effort is more effective than either working alone.

The research aims to solve the problem of needing both speed and accuracy in traffic sign recognition for ADS. While YOLO is fast, it can be imprecise, leading to classification errors. The YOLO-RCNN system addresses this by using R-CNN to refine YOLO's initial detections, resulting in a system that is both fast and accurate, crucial for safety-critical applications like autonomous driving. The research also outlines future directions, including optimizing the model, expanding the training data, and exploring more advanced deep learning techniques.

Modelling and analysis

Modules used for implementing the project are Configuration and Initialization, Frame Processing, Object Detection, Visual and Audio Feedback, User Interaction and Clean-up modules.

Objective

Develop a system to process video input, detect specific objects (e.g., traffic lights, stop signs), and provide visual and audio feedback in real-time.

Workflow

1. Initialization:
   - Load the YOLO model, class labels, and configuration.
   - Initialize the video stream and text-to-speech engine.
2. Frame Processing:
   - Capture frames from the video stream.
   - Pre-process each frame and feed it into the YOLO model for detection.
3. Object Detection:
   - Parse the YOLO output to identify objects, their confidence scores, and bounding boxes.
   - Filter weak detections based on a confidence threshold.
4. Result Visualization:
   - Draw bounding boxes and labels on the detected objects in the video frame.
5. Audio Feedback:
   - Identify specific objects (e.g., traffic lights, stop signs) and provide corresponding audio feedback.
6. Output Display:
   - Display the processed video frame with detections in real-time.
7. Cleanup:
   - Release resources (video stream, windows) after process

Configuration and Initialization Module

The first step in our system is setting everything up. This "Configuration and Initialization" module is responsible for loading all the necessary ingredients for the detection process. It starts by loading the pre-trained YOLO model, which includes the model's learned weights (think of these as the model's "knowledge") and its configuration file (which describes the model's structure). It also loads the list of object classes the model can recognize (the "labels"). This module also sets up some important parameters, like the minimum confidence level the system needs to have before it declares it has found an object and assigns colors to each of the object labels so detected objects can be visually distinguished. The main function here, load_yolo_model, uses OpenCV to read these files and get the YOLO model ready for action. It's like gathering all your tools and ingredients before starting a project – everything needs to be in place before you can begin the main work.

Frame Processing Module

The Frame Processing module is all about getting the video ready for analysis. Its main job is to take the raw video input and turn it into a format that the YOLO model can understand. This involves two key steps. First, it captures individual frames from the video, like taking snapshots of the video at regular intervals. Then, it resizes these frames to a specific size (416x416 pixels in this case), which is what the YOLO model expects. It also does something called "blob creation," which is a bit like preparing the image data for the neural network. This involves normalizing the pixel values (dividing them by 255.0) and reordering the color channels (swapping red and blue, indicated by swapRB=True). This preparation ensures the data is in the right format for the YOLO model to process it efficiently and accurately.

Object Detection Module

The Object Detection module takes the raw output from the YOLO detection process and turns it into usable information about the detected objects. Its main job is to filter out any unreliable or redundant detections. This involves two key steps. First, it parses the raw YOLO output, extracting the important information like where the objects are located (bounding boxes), how sure the system is about each detection (confidence scores), and what type of object it thinks it is (class IDs). Then, it uses a technique called non-maxima suppression to get rid of overlapping boxes. Imagine YOLO finding a car and drawing multiple boxes around it – non-maxima suppression picks the best box and discards the rest. This ensures that each detected object is represented by only one accurate bounding box, making the output cleaner and more reliable.

Visual and Audio Feedback Module

The Visual and Audio Feedback module is all about communicating the system's findings to the user. Its main job is twofold: first, it visually highlights the detected objects in the video frame by drawing boxes around them and labeling what they are. This uses a function that draws rectangles on the image, showing exactly where the system "sees" the object. Second, for certain important objects, the module generates audio messages to provide an additional alert. For instance, if the system detects a traffic light, it might play a spoken message like "traffic light" to draw the user's attention. This combination of visual and audio feedback ensures that the information is conveyed clearly and effectively.

You Look Only Once (YOLO)

YOLO (You Only Look Once) is a real-time object detection algorithm designed for efficiency and speed. It processes an image in a single pass through a convolutional neural network (CNN), dividing the image into an S×SS \times SS×S grid where each cell predicts multiple bounding boxes, class probabilities, and confidence scores for objects within its bounds. YOLO adopts an end-to-end approach, combining classification and localization tasks into a single framework, making it highly suitable for applications requiring real-time detection, such as surveillance, autonomous vehicles, medical imaging, and robotics. It uses non-maximum suppression to remove redundant bounding boxes, enhancing detection results while maintaining global context by analyzing the entire image at once. Despite its speed, YOLO trades some accuracy for efficiency, particularly struggling with small or overlapping objects and requiring large, well-annotated datasets for training. It is lightweight and scalable, often deployed on devices with limited computational power. Over time, improved versions like YOLOv3, YOLOv4, and YOLOv5 have introduced features such as anchor boxes for better performance. Widely implemented in frameworks like TensorFlow and PyTorch, YOLO has transformed real-time object detection, making it accessible for diverse applications like traffic systems, sports analytics, and industrial automation. However, its grid-based design imposes limitations on detecting multiple small objects in a dense scene, emphasizing its trade-offs between speed and precision.

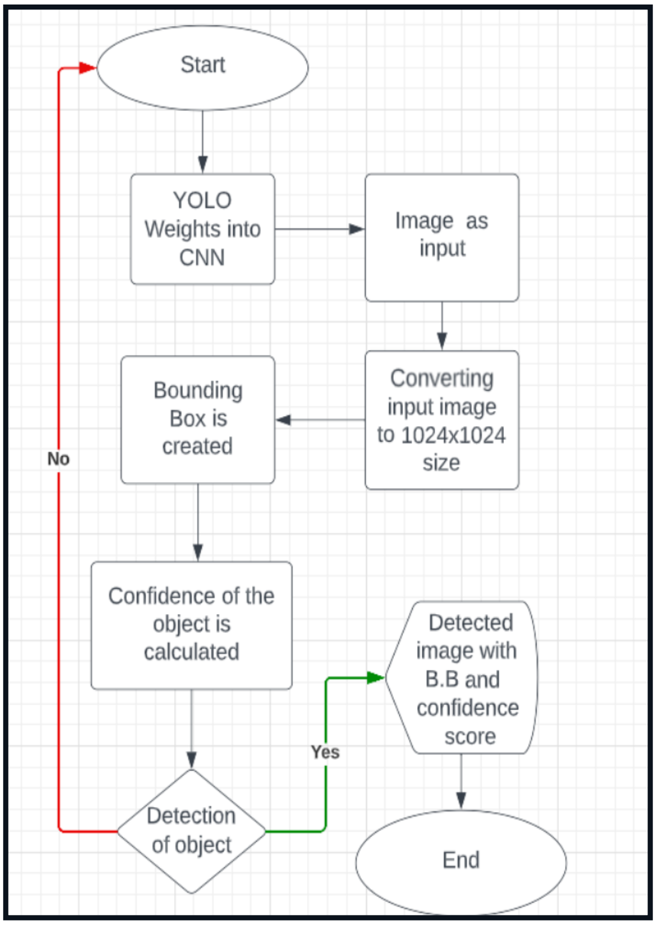
Applications of YOLO Algorithm

1. **Smart Traffic Systems:**
   - Monitoring and managing traffic flow by detecting vehicles and traffic violations.
2. **Drones:**
   - Enables drones to identify objects on the ground or in the air for applications like delivery or inspection.
3. **Robotics:**
   - Assists robots in recognizing objects and navigating environments.
4. **Sports Analytics:**
   - Tracks players and the ball in real-time to analyze gameplay.
5. **Wildlife Monitoring:**
   - Detects and tracks animals in their natural habitats for conservation efforts.
6. **Industrial Automation:**
   - Identifies defective products or components in manufacturing lines

Steps of the YOLO Algorithm

1. Input Image Division:
   - YOLO divides the input image into an S×SS \times SS×S grid.
   - Each grid cell is responsible for predicting bounding boxes and their associated class probabilities for objects that fall within that cell.
2. Bounding Box Prediction:
   - Each cell predicts BBB bounding boxes.
   - For each bounding box, YOLO predicts:
   - x,yx, yx,y: Center coordinates (relative to the cell).
   - w,hw, hw,h: Width and height (relative to the image).
   - Confidence score: Probability that the box contains an object and its accuracy.
3. Class Prediction:
   - Each cell also predicts class probabilities for the object within the bounding box.
4. Non-Maximum Suppression (NMS):
   - YOLO applies NMS to eliminate redundant overlapping bounding boxes, keeping only the box with the highest confidence score.
5. Output:
   - YOLO outputs the detected objects with their bounding box coordinates, confidence scores, and class labels.

YOLO (You Only Look Once) revolutionized real-time object detection by offering a fast, efficient, and unified approach to identifying and localizing objects within images. Its ability to process images in a single pass makes it ideal for time-sensitive applications like surveillance, autonomous vehicles, and robotics. While it excels in speed and simplicity, YOLO's limitations, such as challenges with small or overlapping objects and the need for extensive training data, highlight the trade-offs inherent in its design. Nevertheless, with ongoing advancements in its architecture, YOLO continues to be a cornerstone in computer vision, balancing performance and accessibility for diverse real-world applications.



Region-Based Convolutional Neural Network(R-CNN)

RCNN (Region-Based Convolutional Neural Network) is a landmark object detection algorithm known for its high accuracy. It operates by generating region proposals using methods like Selective Search, extracting features from these regions with a CNN, and classifying them using models such as SVMs. RCNN also refines bounding boxes with regression techniques, ensuring precise localization of objects. While highly accurate, its processing method—analyzing each region proposal individually—makes it computationally expensive and slow, limiting its use in real-time applications. To address this, faster variants like Fast R-CNN and Faster R-CNN introduced optimizations such as a Region Proposal Network (RPN) for quicker region selection. Despite its computational demands, RCNN excels in applications requiring high precision, such as autonomous driving, medical imaging, and surveillance. It paved the way for modern object detection models like Mask R-CNN, cementing its legacy as a foundational approach in computer vision.

RCNN Algorithm Steps

1. ***Region Proposal Generation:***
   - Generate a set of region proposals using algorithms like Selective Search to identify potential object-containing areas in the image.
2. ***Feature Extraction:***
   - Use a pre-trained convolutional neural network (e.g., AlexNet, VGG) to extract feature vectors for each region proposal.
3. ***Classification:***
   - Apply a classifier (e.g., Support Vector Machine or Softmax) to determine the class of each region.
4. ***Bounding Box Refinement:***
   - Perform bounding box regression to refine the coordinates of each detected object.
5. ***Output:***
   - Generate a list of detected objects with their class labels, confidence scores, and refined bounding boxes.

Uses of R-CNN (Region-based Convolutional Neural Networks):

1. ***Object Detection***
   - Identifying and classifying objects in an image while determining their exact locations with bounding boxes.
   - Examples: Detecting cars, pedestrians, or animals in photos or videos.
2. ***Object Tracking***
   - Tracking objects across frames in video sequences.
   - Applications: Surveillance systems, autonomous vehicles, and drone-based monitoring.
3. ***Scene Understanding***
   - Recognizing and localizing multiple objects in complex environments.
   - Useful in robotics for scene analysis and decision-making.
4. ***Text Localization***
   - Identifying text regions within images for tasks like Optical Character Recognition (OCR).
   - Applications: Document scanning, automated license plate recognition, and assisting visually impaired users.
5. ***Autonomous Vehicles***
   - Detecting obstacles, road signs, pedestrians, and other vehicles to enable safe navigation.
6. ***Medical Imaging***
   - Localizing and identifying regions of interest in medical scans, such as tumors in X-rays, CT scans, or MRIs.

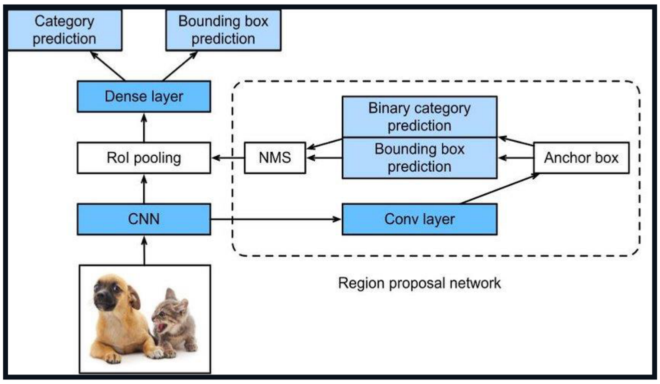
Advantages

1. ***High Accuracy:***
   - Provides precise object detection and classification.
2. ***Robust Feature Extraction:***
   - Uses CNNs for strong feature representation, improving classification and localization.
3. ***Multi-Class Detection:***
   - Handles detection for multiple object categories effectively.
4. ***Scalable to New Models:***

- Can incorporate advanced CNN architectures (e.g., ResNet, VGG).

5. *Bounding Box Refinement:*

- Includes regression for improved localization accuracy.



Metric	YOLO-RCNN	YOLO	RCNN
mAP	0.89	0.78	0.82
FPS	45	65	7
Precision	0.92	0.85	0.90
Recall	0.88	0.79	0.84

Conclusion

Traffic Sign Recognition (TSR) systems stand as a testament to the transformative power of technology in enhancing road safety and shaping the future of transportation. Beyond merely automating sign detection, TSR systems are poised to revolutionize how we interact with the road environment, from enhancing driver safety and improving traffic flow to facilitating the rise of autonomous vehicles and the development of smart cities. TSR systems act as vigilant companions, alerting drivers to upcoming signs, minimizing distractions, and reducing the risk of accidents caused by missed or misinterpreted information. This proactive approach significantly enhances driver awareness and contributes to a safer driving environment for all road users. The successful deployment of autonomous vehicles hinges on their ability to accurately perceive and interpret their surroundings. TSR systems are an indispensable component, enabling autonomous vehicles to navigate safely, comply with traffic regulations, and make informed decisions in real-time. TSR systems are not just about technology; they are about inclusivity. By providing audio or visual alerts about traffic signs, these systems enhance road safety and accessibility for drivers with visual impairments, ensuring a more equitable and inclusive transportation landscape. Beyond enhancing driver safety and facilitating autonomous driving, TSR systems provide invaluable data for traffic management and urban planning. The real-time information gathered by these systems enables traffic engineers to optimize signal timing, improve road design, and alleviate congestion, leading to smoother traffic flow and reduced travel times. This data-driven approach fosters a more efficient and sustainable transportation infrastructure, benefiting both commuters and urban dwellers.

Future Scope

Traffic Sign Recognition (TSR) systems encompass a wide range of applications aimed at enhancing road safety and improving traffic efficiency. These systems automate the detection and recognition of various traffic signs, including regulatory, warning, and informational signs. By analyzing real-time visual data, TSR systems provide crucial information to drivers, autonomous vehicles, and traffic management systems.

- **Enhanced Driver Safety:** Assisting drivers by alerting them to upcoming traffic signs, reducing the risk of accidents caused by missed or misinterpreted signs.
- **Autonomous Vehicle Development:** Enabling autonomous vehicles to navigate safely and comply with traffic regulations, a critical component for the successful deployment of self-driving cars.
- **Traffic Management:** Providing real-time data on traffic flow and congestion, enabling traffic engineers to optimize signal timing and improve overall traffic efficiency.
- **Smart City Initiatives:** Supporting the development of intelligent transportation systems, including connected vehicles and smart infrastructure, to create more sustainable and efficient urban environments.
- **Accessibility**: Improving road safety and accessibility for drivers with visual impairments by providing audio or visual alerts about traffic signs.
- Traffic Sign Recognition (TSR) systems are poised for significant advancements in the future.
- **Enhanced Robustness:** Future systems will prioritize improving resilience against challenging conditions such as adverse weather, varying lighting, and obstructions. Advanced computer vision techniques, including deep learning and adversarial training, will be crucial for achieving this.
- **Global Applicability:** Addressing the diverse range of traffic sign designs and regulations across different countries and regions will be a key focus. This will involve developing robust algorithms capable of handling variations in sign shapes, colors, and symbols, potentially incorporating global traffic sign databases.
- **Synergies with Advanced Driver-Assistance Systems (ADAS):** TSR will be increasingly integrated with other ADAS features, such as lane departure warning, adaptive cruise control, and emergency braking systems, creating a more comprehensive safety net for drivers.
- **Dynamic Information and Real-time Updates:** Future systems will incorporate real-time updates and dynamic information, such as variable speed limits, construction zones, and traffic incidents, to provide drivers with the most accurate and up-to-date information.
- **Edge Computing and 5G Connectivity:** Leveraging edge computing and 5G connectivity will enable faster processing and real-time communication, leading to more responsive and intelligent traffic management systems.
- **Explainable AI:** Developing explainable AI models for TSR will enhance transparency and trust in the system's decision-making process. This will be crucial for building public confidence in autonomous driving technologies.
- **Smart City Integration:** TSR will play a pivotal role in the development of smart cities, providing valuable

data for traffic planning, infrastructure management, and urban development.

- **User-Centric Design:** Future systems will focus on seamless human-computer interaction, providing intuitive and user-friendly interfaces for drivers to access and understand the information provided by TSRsystems.

These advancements will further enhance the safety, efficiency, and sustainability of transportation systems, paving the way for a future where driving is safer, more convenient.

References

1. J. Chu, H. Wang, A. Dong and Y. Chen, "Design of Human Posture Recognition System Integrating Computer Vision and Deep Learning Algorithm," 2023 International Conference on Power, Electrical Engineering, Electronics and Control (PEEEC), Athens, Greece, 2023, pp. 465-469.
2. M. Tan, R. Pang, and Q. v Le, "EfficientDet: Scalable and Efficient Object Detection," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10778–10787.
3. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in NIPS, 2015, pp. 91– 99.
4. Joseph Redmon and Ali Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.
5. Zhi Tian, Chunhua Shen, Hao Chen, and Tong He, "Fcos: Fully convolutional one-stage object detection," in ICCV, 2019, pp. 9627–9636.
6. https://towardsdatascience.com/how-to-detect-objects-in-real-time-using-opencv-and-pythonc1ba0c2c69c0
7. https://aip.scitation.org/doi/abs/10.1063/1.5033788
8. R. Bhallamudi et al., "Deep Learning Model for Resolution Enhancement of Biomedical Images for Biometrics," in Generative Artificial Intelligence for Biomedical and Smart Health Informatics, Wiley Online Library, pp. 321–341, 2025.
9. R. Bhallamudi et al., "Artificial Intelligence Probabilities Scheme for Disease Prevention Data Set Construction in Intelligent Smart Healthcare Scenario," SLAS Technology, vol. 29, pp. 2472–6303, 2024, Elsevier.
10. R. Bhallamudi, "Improved Selection Method for Evolutionary Artificial Neural Network Design," Pakistan Heart Journal, vol. 56, pp. 985–992, 2023.
11. R. Bhallamudi et al., "Time and Statistical Complexity of Proposed Evolutionary Algorithm in Artificial Neural Networks," Pakistan Heart Journal, vol. 56, pp. 1014–1019, 2023.
12. R. Krishna et al., "Smart Governance in Public Agencies Using Big Data," The International Journal of Analytical and Experimental Modal Analysis (IJAEMA), vol. 7, pp. 1082–1095, 2020.
13. N. M. Krishna, "Object Detection and Tracking Using YOLO," in 3rd International Conference on Inventive Research in Computing Applications (ICIRCA-2021), IEEE, Sept. 2021, ISBN: 978-0-7381-4627-0.
14. N. M. Krishna, "Deep Learning Convolutional Neural Network (CNN) with Gaussian Mixture Model for Predicting Pancreatic Cancer," Springer US, vol. 1380-7501, pp. 1–15, Feb. 2019.
15. N. M. Krishna, "Emotion Recognition Using Skew Gaussian Mixture Model for Brain–Computer Interaction," in SCDA-2018, Textbook Chapter, ISBN: 978-981-13-0514, pp. 297–305, Springer, 2018.
16. N. M. Krishna, "A Novel Approach for Effective Emotion Recognition Using Double Truncated Gaussian Mixture Model and EEG," I.J. Intelligent Systems and Applications, vol. 6, pp. 33–42, 2017.
17. N. M. Krishna, "Object Detection and Tracking Using YOLO," in 3rd International Conference on Inventive Research in Computing Applications (ICIRCA-2021), IEEE, Sept. 2021, ISBN: 978-0-7381-4627-0.
18. T. S. L. Prasad, K. B. Manikandan, and J. Vinoj, "Shielding NLP Systems: An In-depth Survey on Advanced AI Techniques for Adversarial Attack Detection in Cyber Security," in 2024 3rd International Conference on Automation, Computing and Renewable Systems (ICACRS), IEEE, 2024.
19. S. Sowjanya et al., "Bioacoustics Signal Authentication for E-Medical Records Using Blockchain," in 2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS), vol. 1, IEEE, 2024.
20. N. V. N. Sowjanya, G. Swetha, and T. S. L. Prasad, "AI Based Improved Vehicle Detection and Classification in Patterns Using Deep Learning," in Disruptive Technologies in Computing and Communication Systems: Proceedings of the 1st International Conference on Disruptive Technologies in Computing and Communication Systems, CRC Press, 2024.
21. C. V. P. Krishna and T. S. L. Prasad, "Weapon Detection Using Deep Learning," Journal of Optoelectronics Laser, vol. 41, no. 7, pp. 557–567, 2022.
22. T. S. L. Prasad et al., "Deep Learning Based Crowd Counting Using Image and Video," 2024.