Frontiers in Health InformaticsISSN-Online: 2676-7104 www.healthinformaticsjournal.com
2024; Vol 13: Issue 3 Open Access

Hybrid YOLOv5-CNN Framework with Grey Wolf Optimization for Enhanced Accident Prevention in Traffic and Industrial Environments

Ms. Prashanthi Gosika[1], Dr.S. Pratap Singh[2]
M.Tech Scholar, Department of CSE, Marri Laxman Reddy Institute of Technology and Management, Dundigal, Hyderabad – 500043[1]
Professor, CSE Department, Marri Laxman Reddy Institute of Technology and Management, Dundigal, Hyderabad-500043[2]
prashanthigosike@gmail.com[1], pratap.singh.s@gmail.com[2]

Cite this paper as: Prashanthi Gosika, S. Pratap Singh (2024 Hybrid YOLOv5-CNN Framework with Grey Wolf Optimization for Enhanced Accident Prevention in Traffic and Industrial Environments. *Frontiers in Health Informatics,* 13(3), 5223-5244

Abstract
Accidents on roads and in industrial areas pose a serious threat to life and property. This study introduces a novel approach that combines advanced computer vision techniques with optimization techniques to improve the effectiveness of accident prevention strategies. The study uses the highly efficient YOLOv5 object search algorithm to quickly identify potential threats. In addition, Convolutional Neural Networks (CNNs) are used to analyze image data to provide context that enhances detection accuracy. The frameworks dataset, sourced from Kaggle, consists of meticulously curated and annotated traffic camera images from various countries, featuring diverse weather and lighting conditions, with bounding box labels for multiple objects such as cars, people, and traffic signs, making it ideal for real-world object detection applications. Deep learning models with powerful CNNs are particularly well suited for visualization when they self-learn reference sequences Model performance is further optimized by Gray Wolf Optimization (GWO) an implemented over, an algorithm inspired by Gray Wolf hunting techniques implemented on Python, and its effectiveness is evaluated through tests on detailed data sets of real-world images. The hybrid YOLOv5-CNN-GWO model exhibits this accuracy compared to widely used object detection methods such as DSTA, SSD, and embedded computer vision. The results show that the YOLOv5-CNN-GWO model performs significantly better than traditional methods, making it a promising tool for crash prevention programs. With an accuracy of 98%, the proposed method outperforms RCNN, FAYOLO, FESSD, and YOLOV4+NSF by 9.82%. The YOLOv5-CNN-GWO framework contains the potential for use in a variety of applications including robotics, autonomous vehicles, and analytics, where accurate and efficient object detection and classification is required.

Keywords— Object Detection, Object Classification, Convolutional Neural Networks (CNN), YOLOv5, Image Analysis, Deep Learning

Introduction

The number of individuals owning cars has increased together with the expansion of social business sectors and the rise in standard of living Numerous transportation and traffic control challenges, such as heavy traffic jams and crashes, have been brought on by the increase in road traffic as a result of modernization. A study of road accidents in India [1] found that unstructured traffic environment resulted in 35.2% unintentional mortality and that moving objects such as cars (10%), two-wheelers (23.2%), and pedestrians (16.1%) were the most common. According to the World Health Organization report [2], 53% of the world's vehicles are operating in a traffic roads, which results in 74% of traffic accidents. Government regulations as well as (NCAP) requirements have forced OEMs to concentrate on incorporating safety systems like airbags and ABS into the majority of passenger vehicles. A lot of work has been put into developing traffic surveillance systems over the past decade with the goal of increasing safety through on-road scene observation. Regarding the growth of smart

cities and autonomous mobility, one of its most important challenges is road safety. The hottest subjects in traffic safety is accident avoidance, where the objective is to discover a reliable system for foreseeing accidents before they occur.

Numerous applications of object detection depends on Machine and Deep learning exist in industries and academic research, including identifying faces, recognizing pedestrians [3], logo understanding [4], vehicle detection and many more [5]. Automated driving, safety monitoring, traffic surveillance, and robotic vision are a few examples of real-world applications. The identification and identification of objects is now possible using a wide range of modalities, including radar, (LIDAR) [6], and computer vision (CV). The key causes for the quick advancement of CV-based tracking and identification of objects are the deep convolution neural network's quick growth and GPUs' increased computational capacity [7]. Because of the poor visual characteristics and the noisy environment they are hidden in, small object recognition continues to be among the most difficult tasks in computer vision. Traffic signs, signals, and pedestrians are just a few of the complicated visual conflicting variables that can obscure small objectives in transportation scenes.

Monocular cameras can be used to recognize and classify objects through various means, including traffic sign recognition, color recognition, and lanes' distinct shapes and colors [8].Moving things and immobile ones are the two basic types of items that are recognized in the driving area. Obstacles, traffic signals, and road signs are examples of fixed things. Items that move include people, animals, and motorized automobiles [9]. To gather reliable traffic images for the creation of effective traffic management systems, a variety of automobile detection approaches have been developed. The first category's old machine learning algorithms, like a SVM,HOG characteristics with AdaBoost classifiers, and Haar-like features, are taught using manually specified characteristics. The second category's deep learning model-based techniques are constructed from a large amount of labelling data. Although both groups may now meet real-time needs, machine learning models built from manually defined features still have lower recognition and categorization precision than deep learning model-based solutions. Deep learning is more effective than machine learning methods from the past because of its ability to recognize abstraction [10]. Multi-object tracking (MOT), a technique that predicts the motions of numerous objects found in a sequence of videos, has been the subject of a number of recent investigations.

Camera images' depictions of the environment are closely related to how people see things. It makes logical for automated cars to rely on similar depiction because humans primarily experience the driving environment through their sense of sight. However, visibility is decreased in bad weather, such as persistent rain or dense fog, making it possible that driving safely won't always be possible. Additionally, in low-light circumstances, noise has a growing impact on camera sensors. Radar sensors are more resistant to environmental factors like weather, variations in the brightness, and fog than those of cameras [11].Designing an effective object recognition method for this complex circumstance is difficult because object identification serves as one of the essential algorithms in automobile industry. Deep learning-based techniques have made the most progress and have enabled a number of applications, including self-driving vehicles, video surveillance and analysis, and recognizing people and communication [12]. conduct instance segmentation and object recognition on images, and other tasks. 2D object recognition in the conventional picture domain has not been able to satisfy the needs of humans due to the complexity situations and the demands of perfect industry. Various 3D object identification methods rely on picture data collected by monocular cameras to extract the three-dimensional data of important objects, covering details like the object's centre position, dimensions, and height [13]. To reduce traffic accidents, autonomous driving technology is being developed. The self-driving automobile system relies on identifying objects to observe the environment and give digital image data for the vehicle's making choices [14].

Using information from the multipurpose sensors mounted to the self-driving automobiles, the autopilot interprets the conditions on the road ahead and reacts appropriately. A multitude of sensors, including high-resolution cameras, radar devices, GPS, and IMUs [15], are now easily accessible to support the huge advancement of industry and aid in the imprecise mapping of the surroundings by autonomous automobiles [16]. Effective solutions for reducing traffic jams in cities, such as measuring the total length of a vehicle's

queue in the path of traffic and forecasting traffic, are necessary [17]. Numerous machine learning and deep learning-based object detection methods exists. Small object detection, despite the fact that these algorithms have been used for a variety of object identification apps, struggles with low resolution. The creation of a compact and reliable object recognition technology that is capable of accurately detecting small things is crucial [5]. Object detection devices built using deep learning methods rely on already trained networks like Alexnet, Imagenet, and Resnet. Due to their availability to consumers and flexibility for customization, these pre-programmed networks have been a blessing [18]. The combination of two potent technologies YOLOv5 for object detection and Convolutional Neural Networks improved using Grey Wolf Optimization presents a revolutionary method for accident avoidance in traffic environments is the main purpose of this proposed model.

The following are the research study's principal contributions:

- The research employs a multi-stage image pre-processing approach, including noise reduction, low-light enhancement, histogram equalization, and video frame extraction, which enhances the quality of input data for object detection.
- Utilizing YOLOv5, the study effectively locates and identifies objects in both images and videos, providing a robust solution for real-time detection tasks.
- The Grey Wolf algorithm leverages high-level features from the CNN to improve object classification accuracy, enhancing the model's overall robustness and effectiveness.
- By optimizing feature selection and hyperparameters through the Grey Wolf algorithm, the model achieves efficient object detection with minimal latency while maintaining high accuracy.
- The primary contribution of the study lies in its provision of a reliable and accurate solution for real-world object detection, which significantly enhances road safety by reducing the risk of accidents.

The remaining sections of this article are organized as follows: A brief summary of related studies is provided in Section 2. A problem statement of the present framework is provided in Section 3. Section 4 of the study describes the study methodology and architecture proposed YOLOv5 and Gray Wolf enhanced CNN pattern recognition. The results of the evaluation and subsequent discussion are presented in Section 5. Section 6 discusses the findings of the proposed model and its future applications.

Related Works

Djenouri et al. [19] presented a vehicular identification system for smart roads that uses an upgraded region convolution neural network to avoid accidents. The article investigates the issue of identifying vehicles and presents an enhanced regional convolution neural network. The noise is initially eliminated via the SIFT extractor from the vehicle's data. The G-RCNN method is applied for dealing with automobile image data, and by defining efficient techniques for choosing the appropriate a accurate identification object recognition system is employed. Through extensive testing using the difficult BOXY automobile data, IRCNN-VD is accessed. IRCNN-VD works better than the traditional approaches in terms of runtime and reliability. When processing 1.9 identified vehicles and 200,000 photos, the IRCNN-VD's mAP achieved 0.85; by contrast, the mAP for the competing methods is less than 0.75. The disadvantage is regarded as improving the suggested method for managing massive and heavy vehicle information in real time by investigating high-performance computing solutions.

Karim et al.[20] proposed a Dynamic Spatial-Temporal Attention Network for Early Traffic Accident Prediction. The study made use of a DSTA network and the TSAA module utilizing dashcam data for the early accident prediction. The DSTA-network selects discriminative temporal segments of a video stream by means of a Dynamic Temporal Attention unit. It also adjusts to focus on the salient areas of frames by utilizing a Dynamic Spatial Attention module. To forecast the likelihood of a future accident, an attention module and a (GRU) are trained together. When evaluated on the CCD dataset, this algorithm achieves 4.48 seconds mTTA. More challenges arise when trying to apply the DSTA-network to these kinds of robots. The DSTA-network

shares a drawback with deep learning models in that the network's decision-making procedure is not transparently justified.

Liang et al.[21] presented a Category-Assisted Transformer model for Object Recognition in traffic scenarios. A category-assisted object detector transformer for automated cars identified as Detect Former was proposed. The Global Extract Encoder (GEE) and proposal groups, in particular, aid Class Decoder in improving category sensitivity and recognition capabilities. When compared to RetinaNet and FCOS, Liang's method achieved a superior real-time recognition performance in traffic situations. The object detector attained a greater accuracy in compared to the starting point. On the BCTSDB dataset, the technique had a recognition performance of 97.5% in AP50 and 91.5% in AP75, respectively. This model increased detection accuracy, but faced numerous difficulties when used in real-world traffic situations with things like illumination and weather. The deployment of models to cars and object recognition in an open environment is the disadvantage.

Mhalla et al.[22] suggested A Multi-Object Detection System for Embedded Computer Vision for Traffic Surveillance. It offered an embedded traffic surveillance system that can be used in these difficult circumstances. The system keeps track of and investigates traffic, focus in particular upon the problem of locating and classifying traffic objects under different traffic situations. The author concentrated on expanding the algorithm to include more complex visual signals into our MFR-CNN network, like context information or optical flow, as well as to introduce some spatiotemporal data. Numerous tests have performed that the technique has formed a traffic object detector which performs better during the day and at night. It is now dependable. A median score of 57% is gained over the generic MF R-CNN by the customized one. One potential limitation of the system is that it might be improved to incorporate more intricate visual cues, such as optical flow or information about context, in addition to introducing some data that is spatiotemporal.

Tao et al. [24] suggested an Intelligent Railway Traffic Object Detection System with Feature-Enhanced Single-Shot Detector. This method inherits a feature transfer block of FB-Net and the prior identification module of RON. Additionally, a newly created responsive field-enhancement module is used. Combining these three modules greatly enhances feature discrimination and robustness. Using convolution neural networks and a graphics processing unit (GPU), the author created an algorithm for automatically detecting railway objects. The algorithm identifies seven various types of items, including pedestrians, helmets, spanners, and bullet trains as well as straight and curved railroad tracks. When the GPU in the computer is detected, the suggested technology can notify the driver to operate the train correctly. With an input size of 320 320 pixels, the FE-SSD outperformed four additional detection devices, resulting in a mAP of 0.895 and an effective frame rate of 38 FPS. According to the trial, FE-SSD is more effective at detecting railroad items than SSD. The disadvantage of the model is that its cost could be reduced while the overall speed could be enhanced.

Zhang et al. [25] suggested an Accident Prevention Combined Sensation and V2V Communication for Predictive Motion Planning. The study specifically included the perception system and vehicle to vehicle communication module to give the participating AVs real-time traffic and vehicle data. Then, a strategy for learning optimal motion planning using a cutting-edge deep learning technology is offered. Utilizing the CARLA system with accessible vehicle to vehicle communication ensures the robustness of the perception system. Comprehensive simulations were used to show how well the suggested system model worked in various scenarios. Additionally, the validity of the perception system's resilience was confirmed. The intersection dataset used to train the algorithm yielded an average reliability of 86% for the intersection and 98% for the highway scenario. This demonstrates the generalizability of the developed model to additional contexts. The drawback is modelling the uncertainties associated with abrupt accidents.

Due to the poor quality of photos, the lack of details, the low illumination, and the distance, which could result in serious accidents, it was demonstrated in the aforementioned articles that it is very difficult to recognize vehicles. Several traditional methods exist for recognizing automobiles in surveillance footage, but their greatest

drawbacks are their high cost and lack of accuracy. Additionally, it has some difficulties looking for items that are gathered together and have several cut-offs and overlays in images taken with small-angle cameras. The traditional approaches for managing massive and enormous vehicle data in real time to prevent real-time accidents need to be improved, and high-performance computer options should be investigated.

Problem Statement

From the reviewed literature the problem suggests that the identification of objects in images is becoming more crucial as a result of developments in computer vision because it can be used for a various applications, such as human identification, face recognition, lane detection and many more. High-performance computing options need to be looked at in order to improve the conventional methods for managing large and enormous vehicle data in real time to prevent real time accidents. When it comes to detecting little items or objects with low intensity in the image, YOLO v4 may not perform as well as certain other object identification models. A YOLO v4 model's training process can be challenging and time-consuming [26] [23]. To obtain decent performance, it might take a lot of labelled data and hardware capacity. This drawback could be overcome by merging the existing YOLO v5 object recognition model with convolutional neural networks and improving its performance with the Grey Wolf Optimization (GWO) technique, this project seeks to enhance it. The main goal is to contrast the enhanced YOLO v5-CNN-GWO model's performance with that of the original YOLOv5 model in the context of preventing accidents by detecting objects with improved accuracy and performance in traffic scenarios.

Proposed YOLOv5 and GWO-CNN for Object Detection

Object detection for accident prevention in traffic scenarios is a critical application of computer vision and deep learning. YOLOv5 is a popular real-time object detection algorithm, and using Convolutional Neural Networks for classification and feature extraction can enhance the system's capabilities. The sophisticated YOLOv5 algorithm was employed, which improves the training data while also passing each batch through the data loader. Data improvements that the data loader can carry out include scaling, color correction, and mosaic enhancement. Tune hyper parameters, such as anchor sizes, learning rates, and batch sizes, to optimize the model's performance. Monitor and visualize the training process using tools like TensorBoard. Use the CNN as a feature extractor for both the detection and classification branches.

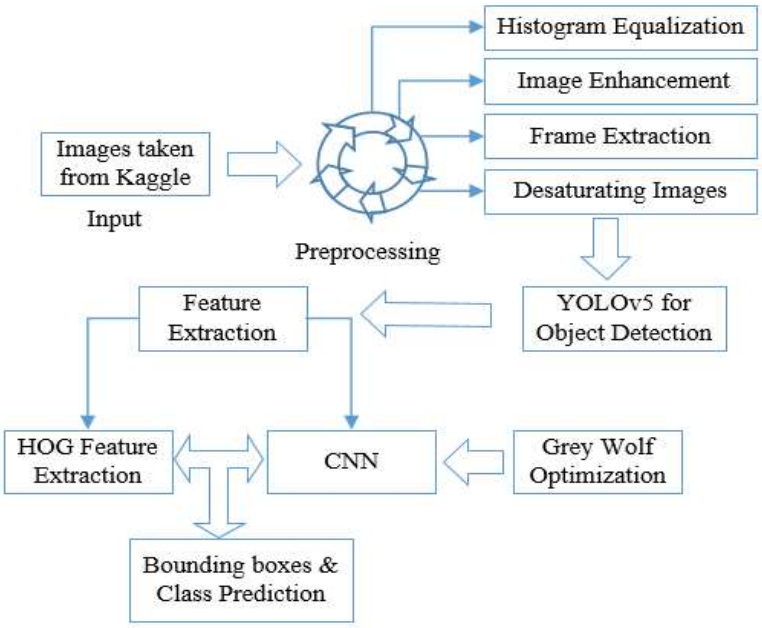

Fig. 1. The Proposed YOLOv5+GWOCNN Overall Block Diagram

Extract features from the final convolutional layers before the classification head. Concatenate or combine these features with the YOLOv5 features (from the final YOLO layer) to create a fused feature representation. Deploy the system on appropriate hardware (e.g., GPUs or edge devices) for real-time or near-real-time inference in traffic scenarios. The suggested methodology's general block diagram is represented in Fig 1, which makes use of CNN and YOLO v5.

Data Collection

This extensive dataset, which has been meticulously curated and annotated, provides a rich collection of traffic camera photos [27] fom different nations across the globe. This set of data is a useful tool for your projects, regardless of whether you're an AI researcher, a machine learning enthusiast, or a developer working on traffic management solutions. Each image is carefully labelled using bounding boxes to identify numerous items, including as cars, people, and traffic signs, making it ideal for object detection applications. The collection comprises pictures taken in various weather, the brightness, and transport situations, making it appropriate for use in practical situations. Table 1 represents the comprehensive synopsis of the dataset

Table 1. A Comprehensive Synopsis of the Kaggle Dataset.

Classes	Total Images(Kaggle)	Training Images	Testing Image
Vehicular Images from surveillance camera	6084	5805	279

Data Pre-Processing

Low Light Image Enhancement Using Retinex Model

Images taken in low light have a significant effect on both daily life and output. For instance, in terms of traffic safety, when vehicle cameras or surveillance equipment cannot pick up on stealthy things at night, they provide a risk because the low light visibility environment can greatly deceive drivers, rendering them unable of understanding road conditions and prone to accidents. The majority of today's imaging equipment' fundamental parts also have poor ambient lighting processing capabilities in low-light conditions and are unable to differentiate images effectively due to noise. For the purpose of studying low light photos, some scientists and researchers have put forth a number of augmentation ideas based on conventional digital image process [28]. Our technique is a member of the Retinex-based group, which aims to improve low-light images by calculating their illumination map Retinex can balance compression, edges improvement, and color consistency with dynamic range, in contrast to traditional linear and non-linear algorithms that exclusively enhance specific image types.

In order to obtain the desired outcome, the Retinex model minimizes both the solution space and the computational cost by only estimating illumination. Comparatively speaking, traditional Retinex-based techniques split an image into its illumination and reflectance components. The illuminating map is first built by determining the highest pixel intensity in the R, G, and B channels. The light map is then improved by taking advantage of the illumination's structure. In order to precisely solve the refinement problem, an Augmented Lagrangian Multiplier (ALM) based technique is provided, and a speeded-up solution is created to significantly reduce the load on the computer. Experimentation carried on a variety of difficult photos to demonstrate the benefits of the approach over other cutting-edge techniques. Equation (1) denotes both the intended recovery and the image captured.

$$P = R \circ I \quad (1)$$

The picture that was captured and the intended recovery are denoted by P and R in the equation above, respectively. Additionally, I stands for the illumination map, and the operator ◦ denotes multiplication of elements. Assume that three channels in a color image have identical illumination map. Use I ($\hat{I}$) to denote one-channel and three-channel illumination maps interchangeably, with a small amount of notation abuse.

Illumination Map Estimation

Max-RGB, one of the first color constancy methods, calculates the maximum value for each one of the three main color channels—R, G, and B—in an effort to ascertain illumination. But this estimate will only result in a rise in global illumination. You should use the alternate initial estimation Eqn. (2) that is explained below for work with non-uniform illuminations.

$$\hat{I}\,(\mathrm{a}) \leftarrow \max_{\mathrm{c}\in\{\mathrm{R,G,B}\}} P^{c}\,(\mathrm{a}), \tag{2}$$

for every single pixel a. The process described above is based on the idea that lighting should be at least as bright as three channels at any given place. Because of the following, the obtained I (a) ensures that the recovery won't be saturated:

$$\mathrm{R(a)} = \frac{P(a)}{\left(\max_{c} P^{c}(\mathrm{a}) + \epsilon\right)}, \tag{3}$$

In the above Eqn. (3) where ϵ is a tiny constant used to get around the numerator of zero. To unevenly improve the illumination of low-light photos rather than get rid of the hue shift brought on by light sources is the main attention.

Noise Removal and Histogram Equalization

The original image is first blurred using a Gaussian filter. A common method for removing noise and smoothing out edges from photos while leaving the majority of the image intact is to blur the image. The region of the blurred image containing the face is taken and utilize DLIB's library [29] to extract it from the overall image. In specifically, we employ its face detector, which determines the face's location's coordinates by employing a linear supporting vector machine (SVM) and histograms of oriented gradients (HOG).

Frame Extraction

Traffic sign segmentation systems are an effective way to carry out frame extraction in the prior processing system and reduce the computational load for artificially intelligent systems. Traffic sign colors and shapes have specific characteristics. In order to streamline the neural network model training process, it is necessary to first extract specific images from the video frames. A video source operating at a frame rate of 30 frames per second could yield about 5000 images. The main source of data for the prediction model consists of these recovered images.

Desaturating the Images

Monochrome color is the format used by the majority of surveillance cameras. Most especially if it employed infrared. Desaturation is made to the RGB photos in order to make the dataset more closely resemble the inferencing data to be anticipated.

YOLO v5 for Object Detection

A You Only Look Once (YOLO) methodology has been used to create some of the most effective models to date. YOLOv5, the most recent iteration of the YOLO series, is a state-of-the-art one-step recognition procedure. The YOLOv5 network is made up of three main parts: the Head, Neck, and Backbone. The Backbone's convolutional neural networks bundle and form representational features at different granularities (Li et al. 2020). In order to improve prediction, the structure's neck is composed of several layers that integrate and combine image representational features. The head is the same as the neck; it has access to the box and class prediction capability and borrows neck features. The CSPDarknet53 backbone of YOLOv5 consists of 29 layers of convolutional neural networks, each all three by three, with a combined total of 27.6 M attributes and

a field that is receptive size of 725 by 725. Additionally, YOLO's CSPDarknet53's SPP block connected over it increases the proportion of receptive fields without affecting how it operates. By utilizing various tiers of backbone, the feature aggregation is carried out through PANet. YOLOv5 leverages characteristics like traverse mini-batch, cross-stage partial connections, weighted residual connections, standards, and self-adversarial training to extend the field's bounds. In the current study, YOLOv5 model trained and implemented using the PyTorch framework. By adjusting the following hyper parameters, the YOLOv5 model is improved to further achieve the objective of vehicle recognition: batch-size 64, optimizer decay of weight value of 0.0005, learning rate of 0.01 and momentum maintenance at 0.937.

Feature Extraction and Classification Using HOG and CNN

Although this method has a low detection efficiency, the HOG features have been utilized for obtaining exact targets by examining the object's dense and overlapping traits with extensive picture scanning and complex calculations (Arróspide, Salgado, and Camplani 2013). Nevertheless, HOG features have the advantage of characterizing objects. This study uses integral images as well as HOG features to detect cars. This is accomplished by calculating the image's gradients in both the horizontal and vertical directions, then dividing it into equal-sized cells and joining adjacent cells to form larger blocks. Then, by superimposing image slides, the HOG features of the recognizing window are constructed.

HOG Feature Extraction

The gradient is calculated using a straightforward first-order template, and features are extracted using R-HOG. One way to apply such descriptors is to first divide the pic into tiny interconnected regions known as "cells," and then for each cell, compute the histogram of the edge orientations. The process involves computing picture gradients for every pixel, which are then utilized to ascertain the gradients' orientations and magnitudes. Secondly, weighed votes for the gradient magnitude are totalled and placed into orientation bins across spatial cells. The weighted votes are interpolated bi-nearly between the neighbouring bins in both orientation and position to avoid aliasing. The weight is divided into eight surrounding bins by tri-linear interpolation and into the two nearest neighbouring bins by linear interpolation; the third step normalizes the contrast of each block. Compiling the histograms for every block that overlaps over the detection window is the last and fourth step.

The gradient chart is used to implement HOG characteristics. First, the front-view image's horizontal dx and vertical gradients dy are determined. Eqns. (4) and (5) allow for the calculation of gradient magnitude and direction.

$$\mathrm{X(a,b)} = \sqrt{(\mathrm{K(a+1,b)} - \mathrm{K(a-1,b)})^2 + (K(a,b+1) - k(a,b-1))^2} \quad (4)$$

$$\theta(\mathrm{a,b}) = \arctan\left(\frac{\mathrm{K(a,b+1)} - \mathrm{K(a,b-1)}}{\mathrm{K(a+1,b)} - \mathrm{K(a-1,b)}}\right) \quad (5)$$

Targets in the photograph varied greatly in their sizes and placements. The identical target appears in various sizes depending on the viewpoint. The scaling factor for this time period is also 1.2. The object can be detected in numerous scales by gradually scaling the minimum 64*64 size. The method of estimating the integral image is similar to the one previously described. There is variation in the nine areas of the HOG features integral picture. Nine integral images with the same dimensions as the initial pictures can be produced by classifying the phase angle within a 180-degree range and accumulating the amplitude. This is carried out in compliance with symmetry and 20 degree intervals.

Convolutional Neural Network (CNN)

Convolutional neural networks (CNNs), a family of deep, feed-forward artificial neural networks, have demonstrated accurate performance in computer vision tasks and picture classification, feature extraction, and detection (Krizhevsky, Sutskever, and Hinton 2017). Transfer learning, in which CNN models that have already been trained are employed as feature extractors, is another method for extracting CNN features. CNNs are especially well-suited for tasks involving grid-like data, like pictures and videos, since their architecture allows them to automatically and adaptably discover spatial feature hierarchies from the input data.

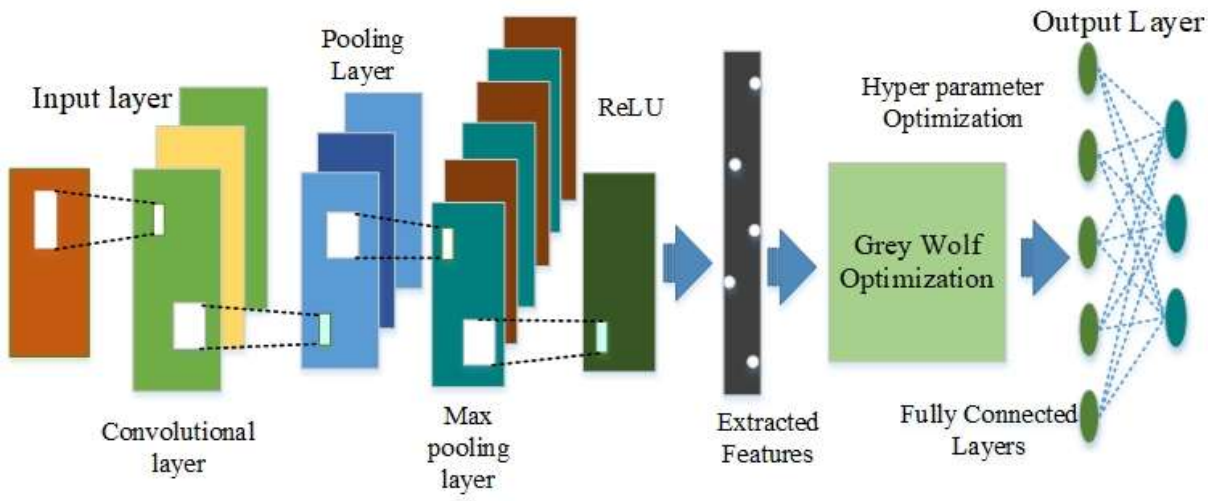

Fig. 2. Architecture of Convolutional Neural NetworkA popular kind of artificial neural network in deep learning for a range of computer vision applications is the CNN. CNNs are particularly well-suited to tasks that require them to autonomously and adaptively learn spatial hierarchies of features from the source data. Examples of such tasks include images and videos. Fig 2 represents the architecture of convolutional neural network.

The key components and concepts of a Convolutional Neural Network include:

Convolutional Layers

Convolutional layers are used to extract features from an image input, such as edges, faces, textures, or complex patterns. They draw filters on the source data, each designed to recognize specific features or shapes. The output of these filters is a feature map, which can be used to create sophisticated images capable of extracting finer details..

The equation. (6) represents the input pixel, filter, and in-place value

$$A(p,q) = \sum x \sum y\, B(p+x, q+y) * E(x,y) + b \quad (6)$$

Where A (p, q) is the value in the future map at position (x, y).

Pooling Layers

Pooling layers are used to reduce the spatial extent of user input and speed up analysis. They come after winding layers in the processing chain, reducing the number of objects or loads in the system. This helps prevent overfitting and speeds up training in the image. Without max pooling, the network would not recognize features irrespective of small shifts or rotations, making the model less robust to variations in object positioning within the image.

Max pooling

Max pooling extracts the maximum value of each feature map, selecting the pixel with the highest value within the pooling window. The sum of each value in the pooling window is determined by average pooling, presenting a picture of soft, soft objects. The equation. (7) represents the max pooling function.

$$F(p,q) = \max\left(F(2r, 2s), F(2r, 2r+1), F(2r+1, 2s), F(2r+1, 2s+1)\right) \quad (7)$$

Activation Functions:

The system is made non-linear by applying activation functions such as ReLU (Rectified Linear Unit), which allow the system to develop intricate maps between the inputs and results. The ReLU is commonly used: ReLU is given in Eqn. (8)

$$A(p,q) = \max(0, A(p,q)) \quad (8)$$

Grey Wolf Optimization for Enhancing CNN

In 2014, Seyedali Mirjalili created the GWO by mimicking the social actions, hierarchical structure, and hunting practices of grey wolves on group property. Where there is wildlife, grey-colored wolves usually live in packs. Five to twelve people take part. They uphold a rigid structure of social domination. The most leading male or female wolves are the alphas, or top dogs in the hierarchy. They decide most of exactly what the pack of wolves eats, sleeps on, hunts, uses as a home base, and engages in other activities. The alpha wolf is followed by all other wolves. The beta wolves, who follow the alpha's instructions and tend to the lower-level wolves, are the next tier of the wolf hierarchy. The delta wolves, who assist the alpha and beta wolves in their hunting and prey-finding, are represented by the next category. They protect the borders of their domain, tending to the weak and injured wolves and alerting the other wolves to any risk that may arise. The lowest-ranking wolves, known as omegas, obey the orders of all other wolves. The social structure of wolves is crucial to their success in hunting. By describing the wolf's position as the best option in the search area and viewing the prey's location as the ideal solution, it is possible to quantitatively describe the social habits of grey wolves. The best option are the alpha wolves because of their closeness to the prey. Alpha, beta, and delta wolves shift about in the search space, and omega wolves adapt their position accordingly. Let the locations of the alpha, beta, delta, and omega wolves be denoted as $Y_\alpha, Y_\beta, Y_\delta, Y_\omega$. The main GWO activities include encircling, hunting, attacking, and searching for prey.

Encircling Prey Mechanism

When wolves hunt, they encircle their victim, a process known as prey encircling that can be illustrated mathematically by equations (9) to (10).

$$\vec{E} = \left|\vec{D} \cdot \overrightarrow{Y_P}(a) - \vec{Y}(a)\right| \quad (9)$$

$$\vec{Y}(a+1) = \overrightarrow{Y_p}(a) - \vec{A}.\vec{E} \quad (10)$$

where 'a' denotes the present iteration,$\overrightarrow{Y_p}$ denotes the position, $\vec{Y}$ denotes the wolfs' position,$\vec{A}$ and $\vec{D}$ denotes the coefficient vectors, and $\vec{a}$ is reduced from 2 to 0 throughout the iteration. It is used to approach the solution range $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$ as displayed below in Eqns. (11) and (13) , respectively.

$$\vec{A} = 2\vec{a}.\vec{r} - b \quad (11)$$

$$\vec{D} = 2\overrightarrow{r_2} \quad (12)$$

$$\vec{a} = 2\left(1 - \frac{y}{G}\right) \quad (13)$$

Hunting Prey Mechanism

α, β and δ wolves performed the hunting activity ; they have a better understanding of position.

$$\overrightarrow{E_\alpha} = \left|\overrightarrow{D_1}.\overrightarrow{Y_\alpha} - \vec{Y}\right| , \overrightarrow{E_\beta} = \left|\overrightarrow{D_2}.\overrightarrow{Y_\beta} - \vec{Y}\right| , \overrightarrow{E_\delta} = \left|\overrightarrow{D_3}.\overrightarrow{Y_\delta} - \vec{Y}\right| \quad (14)$$

where 'a' stands for the amount of iterations currently being performed and A1, A2, and A3 are the random vectors. The wolfs' location is shown with the letter$\vec{Y}$*. The positions of the wolves* α *are shown by the* $\overrightarrow{Y_\alpha}$ *and* $\overrightarrow{E_\alpha}$ *symbols, respectively. The positions of the current and modified wolves* β *are shown by the symbols* $\overrightarrow{Y_\beta}$ *and* $\overrightarrow{E_\beta}$*. The positions of the current and modified* δ *wolves are shown by the symbols* $\overrightarrow{Y_\delta}$ *and* $\overrightarrow{E_\delta}$*. After calculating distances, it is estimated that current wolves* $\overrightarrow{Y_1}, \overrightarrow{Y_2}$*, and* $\overrightarrow{Y_3}$ *are exactly where Eqns. (15) and (16) depict them.*

$$\overrightarrow{Y_1} = \overrightarrow{Y_\alpha} - \overrightarrow{A_1}.\overrightarrow{E_\alpha}, \overrightarrow{Y_2} = \overrightarrow{Y_\beta} - \overrightarrow{A_2}.\overrightarrow{E_\beta}, \overrightarrow{Y_3} = \overrightarrow{Y_\delta} - \overrightarrow{A_3}.\overrightarrow{E_\delta} \quad (15)$$

$$\vec{Y}(a+1) = \frac{\overrightarrow{Y_1}(a+1) + \overrightarrow{Y_2}(a+2) + \overrightarrow{Y_3}(a+3)}{3} \quad (16)$$

where 'a' stands for the number of iterations currently being performed and $\overrightarrow{A_1}$, $\overrightarrow{A_{2,}}$, *and* $\overrightarrow{A_3}$ *are the random vectors.*

Attacking Prey

The attacking step and exploitation step is same and it is denoted by $\vec{a}$. The moving wolves attacks the prey when it stops. $\vec{A}$ is a random value in range $[-2r, 2r]$ and $[-1,1]$ is the range of r2. The next position is any position between prey and its recent position. So $|\vec{A}| < 1$ the attacking condition is suitable. According to the searching behaviour of the wolves searching and exploration of an optimal solution is modelled. When $|\vec{A}| > 1$ wolves diverges for finding better prey. When $|\vec{A}| < 1$ wolves converges for finding the best prey.To avoid local optimum and favouring exploration random $\vec{D}$ is used. It gives random value at both initial stage and final stage of the algorithm.

Fully Connected Layers

Fully connected layers are used at the end of a convolutional neural network (CNN) to use features obtained by pooling the min and max layers for prediction purposes, such as input labeling. This allows the network to combine and integrate all extracted features across the entire image, without considering local features. It helps to understand the global context of the image and maps the combined attributes to expected outcomes like class labels in function classification. The first section examines simple models, and the next section builds on it to examine more complex models. The equation. (17) and (18) as

$$\text{Weighted sum: } Z = B.C + b \quad (17)$$

$$\text{Activation: A= softmax (Z)} \quad (18)$$

The final stage of a CNN is similar to the last layer of a traditional neural network in the fully connected layer. The input is an image with a pixel value, available in three dimensions: depth, width, and RGB modes. The convolutional layer computes the output of neurons associated with specific input local areas, while level parameters are detectable filters that rotate over the input volume and extend its depth to calculate the dot product between each entry of the filter and the visible input volume. An improved linear unit (ReLU) level function is activated element-wise.

Output Layer

For image categorization tasks, the amount of neurons (nodes) in the layer of output is usually equivalent to the amount of classes or groups that the algorithm is projected to classify. For classification tasks, common activation coefficients in the output layer include Sigmoid for classifying binary data and Softmax, which turns the raw class scores into class probabilities. The Eqn. (19) represents

$$Output\ Probability(Z) = Sigmoid(Neuron\ Output\ (Y)) \quad (19)$$

GWO-CNN Algorithm

Input: Objects in image
Output: Image with object detection
Load the source image
Perform pre-processing operation
//Noise removal and histogram equalization
//Object detection using YOLOv5
Extract image features
//Feature extraction by HOG using Eqn.(4)
//CNN Feature Extraction
for each convolutional layer do:
for each layer in X do:

calculate a_{ij}^{m} from X by the convolution layer process
End for
//Dimension of a is (512-Kernal size=1,Filtersize)
If a_{ij}^{m} length<512 do:
Apply zero padding to a_{ij}^{m}
//Dimension of a is (512,FilterSize)
End if
End for
Run feature selection module
//Optimize data from high dimension to low dimension
End for
Run Prediction module
//Put feature vectors of the train set as the predictor input
//Grey Wolf Optimization for Hyper Parameter Tuning
Initialize the wolf populations Y_i(i=1,2…..n)
Initialize A ,C and a
While(t<max no of iterations)
For each agent
Update the current agent's position by using Eqn. (10)
End for
//Update a, A and D
//Calculate search agents using Eqn. (15)
t=t+1
End while
Return search agents
//Classification using CNN
//Through forward propagation, the model output the classification score
Return The highest classification score as the prediction results
End

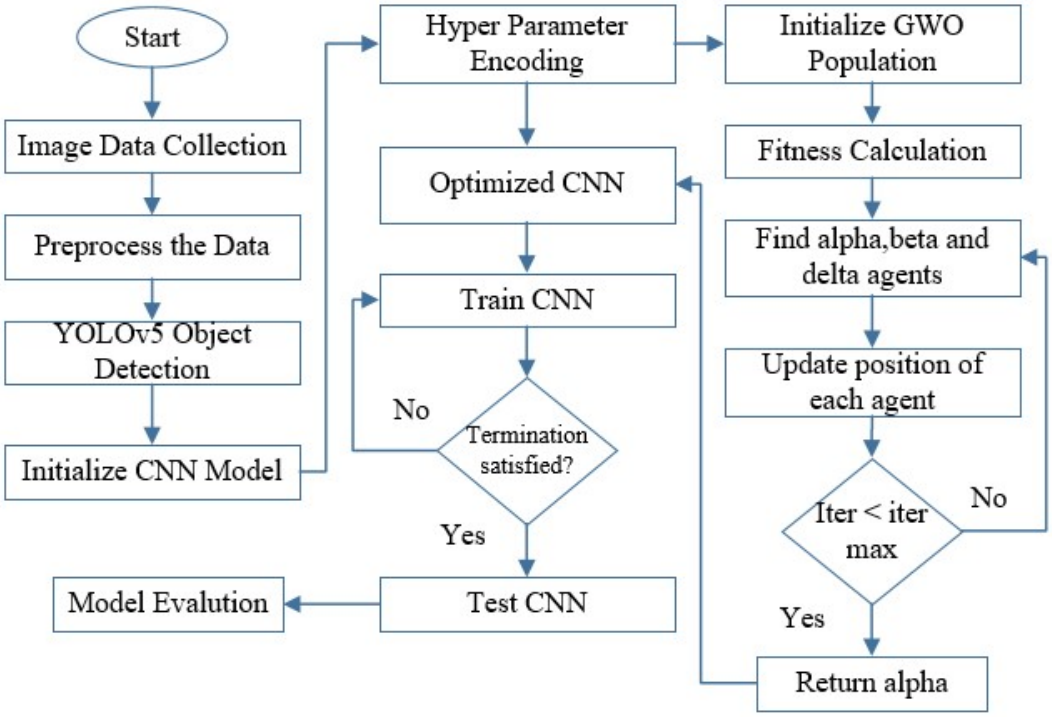

Fig. 3. Flowchart of the Proposed YOLOv5+GWOCNN MethodFig.3 shows the overall workflow of the proposed Gray Wolf optimized convolutional-neural network model. What YOLOv5 has to offer in GWOCNN enables better analysis, more accurate classification, and more reliable detection for accident prevention in traffic accidents.

Result and Discussion

Traffic Detection Project datasets are used to identify the object. The Traffic Detection Project dataset is used as the project's input image source and to find objects in an image, the YOLO v5model is employed. CNNs are utilized for analysing images, classifying objects from CCTV surveillance photos. The CNN's parameters are adjusted via the Grey wolf optimization, which improves its capacity to identify objects from the picture input. A CNN classifier is used to reach the final detection conclusion. A well-liked machine learning approach called Grey Wolf optimization excels at managing complicated, high-dimensional data High-dimensional data and precise predictions are provided by CNN. Gaussian filtering is used to eliminate noise from the input images and feature extraction by HOG.

Evaluation of Data pre-processing

A comparison of images before and after preprocessing shows the effectiveness of the data enhancement methods. Factors such as poor lighting and noise in "pre-processing" images severely hamper visibility, making it difficult to distinguish basic features, especially in the case of traffic control. Retinex images effectively enhance the image by creating a light map, which changes the balance of light throughout the scene as shown in the fig.4.

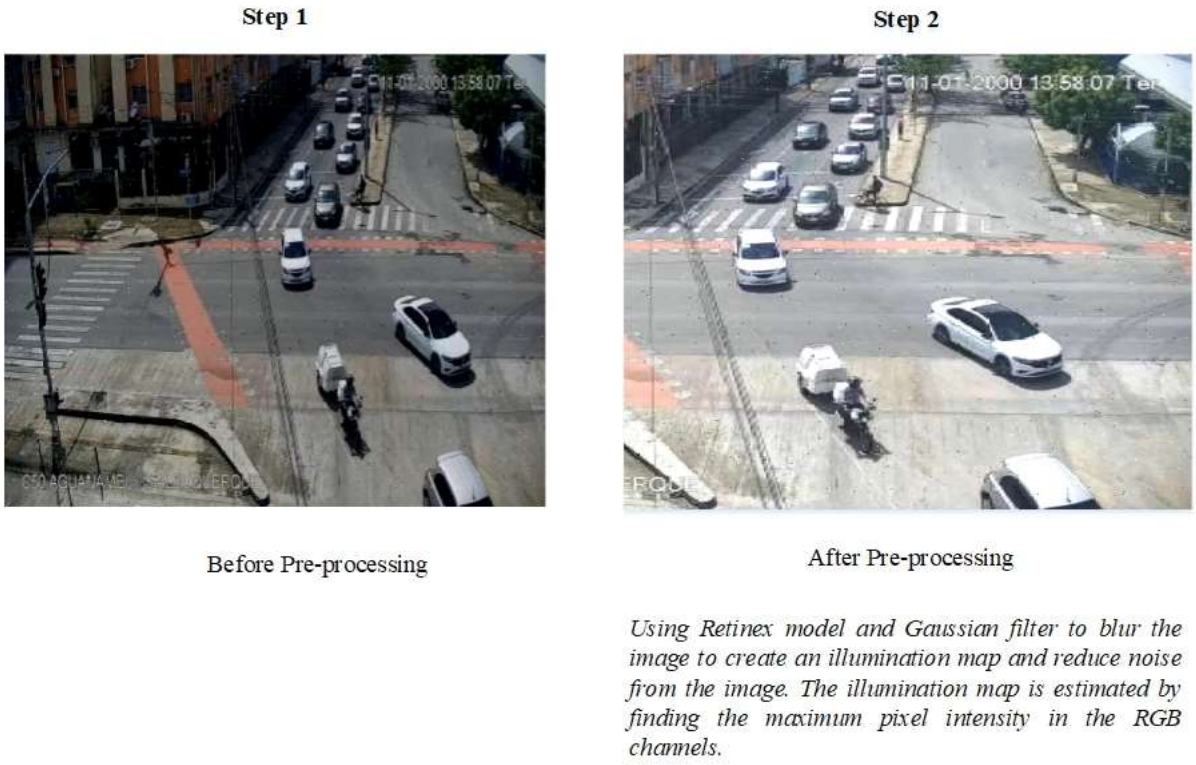

Fig. 4. Pre-processing using Retinex Model and Gaussian FilterThis technique can brighten dark areas while preserving the natural look of well-lit areas, resulting in an evenly illuminated image Besides, the use of a Gaussian filter reduces noise down again, it simplifies unnecessary information without compromising the clarity of important features such as traffic and road signs , as it provides clear, consistent information that makes it easier to control devices -learn to process and interpret patterns The result is a much cleaner image, with better visibility, less noise, which provides any underlying tasks such as object recognition or traffic signals.

Outcome of Object Detection by the Proposed YOLOv5 and GWOCNN Model

The object recognition results using the proposed YOLOv5 and GWOCNN (Grey Wolf Optimization Convolutional Neural Network) models are visually presented in the figure, where vehicles and a bicycle are accurately detected and it is written on it. The boundary boxes around each detected object are well placed, which means that the images are more accurate to distinguish between vehicles The survey system captures objects of different sizes and distances from the camera very well , and shows the robustness of the images in real-world conditions with complex background and lighting conditions They detect, while GWOCNN increases the detection accuracy by fine-tuning the feature extraction process This stated approach together this provides a highly reliable detection system that can be used in traffic detection, surveillance, and autonomous driving applications.

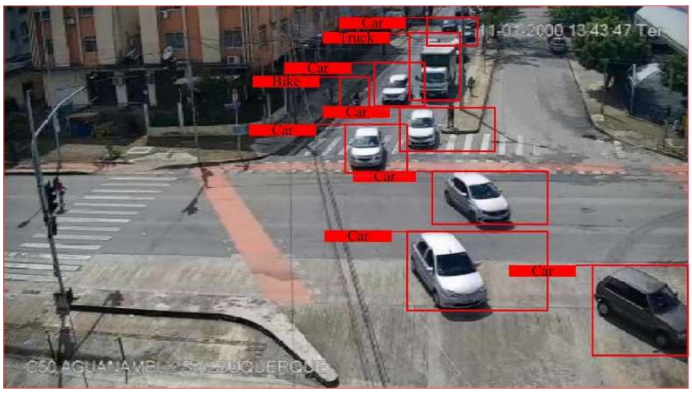

Fig.5. Demonstrations of Detection Results by the Proposed Method

The full visibility of all objects in the frame means that the models are well trained and all objects can be moved to different locations, which greatly improves the object recognition performance A total of 6084 images are used, of which 279 are test images and 5805 are training images. On 6084 images, the proposed model achieved an accuracy rate of 98% on the test data set. Fig. 5, shows the images of objects (cars, trucks, and bicycles) detected by YOLOv5 and the proposed GWOCNN models.

Performance Evaluation of Proposed YOLOv5 with Existing Approaches

Table 3 provides a detailed comparison of the mean absolute error and mean squared error performance parameters of the proposed YOLOv5+GWOCNN model in comparison to the traditional search methods of YOLOv4 , YOLOv3, and YOLOCC among others. In particular, the YOLOv5+GWOCNN model records 7.26% MAE and 8.20% MSE, which perform at 11.36% and 19.90% MAE and MSE values, respectively, than the YOLOv4 method as well as YOLOv3, with 14.13% (MAE); and 28.64% (MSE) higher error rate, YOLOCC , with 10.46% (MAE) and 17.40% (MSE) are also lower compared to the proposed method.

Table II. Evaluation of Proposed YOLOV5's Performance Metrics

Methods	MAE (%)	MSE (%)
YOLOv4 [34]	11.36	19.90
YOLOv3[35]	14.13	28.64
YOLOCC [36]	10.46	17.40
Proposed YOLOV5	7.26	8.20

This significant reduction in both MAE and MSE highlights the effectiveness of the YOLOv5 model hybridized with the Gray Wolf Optimization Convolutional Neural Network (GWOCNN) YOLOv5 algorithm for improved feature extraction and detection capabilities, period a GWOCNN continuously optimizes the detection process by fine tuning model parameters, resulting in improved accuracy Low error rates indicate that the proposed model is accurate and reliable in object recognition tasks , and leads to fewer errors in the prediction of object locations and distributions in images The MSE and MAE values of the proposed method YPLOv5+GWOCNN are lower than the existing methods YOLOv4, YOLOv3 and YOLOCC.

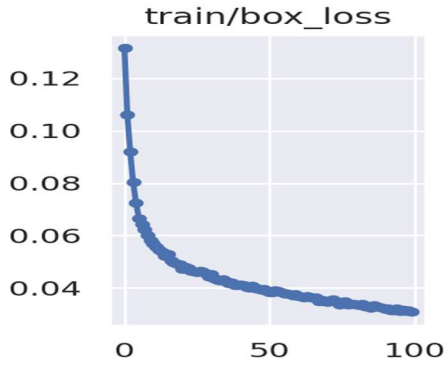

Fig. 6. Absolute Error Graph for Traffic Detection Using YOLOv5

The absolute error bars show the performance of the YOLOv5 model in the training and validation phase. Key metrics such as box missing, object missing, and classification missing are plotted, showing a steady decrease in error as training progresses, indicating that the model is learning better as the model efficiently changes its parameters and reduces error over time as shown in Fig 6.

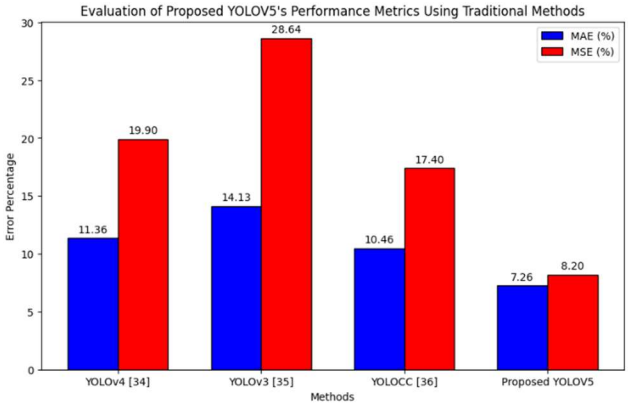

Fig. 7. Graphical Illustration of MSE and MAE Values of YOLOv5 with Existing Approaches

Fig 7 visually reinforces these findings by showing the comparative performance of the models. The graph clearly shows that the proposed YOLOv5+GWOCNN method consistently achieves the lowest MSE and MAE values among all the tested methods. This simulation, together with the numerical results in Table 3 , provides strong evidence of the effectiveness of the model in reducing prediction errors, thereby improving the overall performance of the object recognition system in a complex real-world environment.

Performance Evaluation

The proposed model is compared to Faster R-CNN, FA-YOLO, FESSD, and YOLOv4+NSF, focusing on accuracy, recall, F1-score, and accuracy. True Positive Cases are identified as positive, while False Negatives are classified as negative. Recall is crucial for accurate classification.

Accuracy

Precision is crucial for capturing metrics like accuracy, recall, F1-score, and confusion matrices. Accuracy is determined by comparing ground truth labels to predicted class labels, increasing the "correct prediction rate" if predicted lines match actual images as represented by the Eq (20):

$$Accuracy = \frac{RN+}{RP+AP+RN+AN} \quad (20)$$

Precision

For classification topics, accuracy is a common measure, especially in machine learning and statistics. It examines how well a model predicts the future in a positive way. It is defined as the in Eqn. (21) is as follows:

$$Precision = \frac{True\ Positives}{(True\ Positives+Fals\ \ Positives)} \quad (21)$$

The accuracy level ranges from 0 to 1, where 1 indicates perfect accuracy (all equally good predictions) and 0 indicates that no equally good predictions were made Using this equation requires data a are formulated including predictions of the model and actual ground truth scores. Then calculate the actual positives and negatives using the method described above to determine their accuracy.

Recall

"Recall" typically refers to one of the outcome indicators for assessing the effectiveness of a model in a classification task, especially for bipartite or multiclass classes, and the sensitivity and accuracy rates are nominal others for memory. Effective discovery recall refers to the ability of the model to recognize every valid instance of a given class contained in the data set For all actual positive events for a given class as a whole in , it calculates the percentage of truly positive predictions (correctly identified models in that class) . Residuals can be described mathematically as in Eq. (22).

$$Recall(sensitivity) = \frac{True\ Positives}{True\ Positives+Fa\quad Negatives} \quad (22)$$

F1-Score

F1 scores are a statistic widely used to evaluate the performance of ranking models in classification tasks, especially for improving feature identification and classification F1 scores have particular advantages in imbalanced data, i.e. one class is significantly higher than another. Eq. the following characteristics. (23) is used to determine F1 scores:

$$F1\ Score = 2 * \frac{(Precision*Recall)}{(Precision\quad\quad)} \quad (23)$$

While your assessment will consider both, the F1 score provides a helpful neutral measure of recall and accuracy. It is a useful metric to use to select precision and recall, as is common in classification tasks.

.

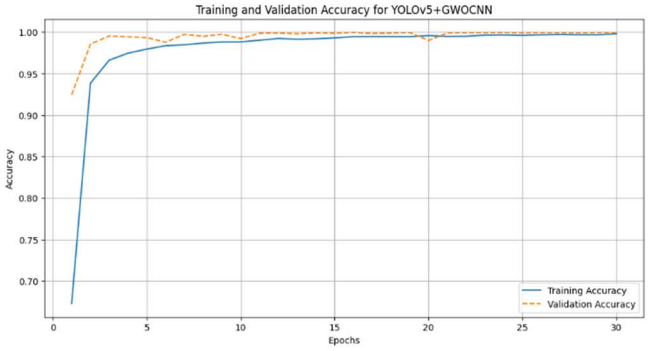

Fig. 8. The Suggested Yolov5+GWOCNN Method's Training and Testing Accuracy

In Fig. 8, the accuracies of testing and training are plotted for comparison with different periods. It is clear that GWOCNN fits the data faster because the accuracy ratio and loss ratio composite graphs remained stable within 100 intervals for GWOCNN collected 1500 samples for the training phase and 1400 samples for the testing phase. During the testing phase, the model continuously predicts the output state as video images are captured from the data set.

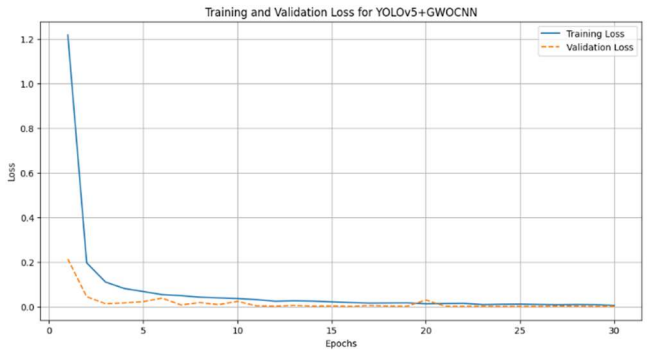

Fig. 9. The Proposed YOLOV5+GWOCNN's Training and Testing Loss

The loss values corresponding to the number of times are shown in Figure 9. This shows the total loss from the proposed YOLOv5+GWOCNN model. A large data set of traffic images from CCTV Surveillance is used to train the GWOCNN model. The hyperparameters of the system are varied using Gray Wolf optimization, which makes the model sensitive to objects.

Performance Comparison with Existing Framework

The accuracy of the proposed model is shown in Table 2. It shows the accuracy (98%), recall (98%), precision (97%) and F1-score (95%) of the proposed method with methods the usual methods are used.

Table III. The PROPOSED Performance Metrics are Compared to Existing Methods.

Method	Accuracy (%)	Precision (%)	Recall (%)	F1Score (%)
Faster R-CNN[30]	87.97	85	90.74	80
FA-YOLO[31]	92.95	92.55	94	88.13
FESSD[32]	83	90.18	85.52	84.63
YOLOV4+NSF[33]	93	94.21	81.06	87.98
YOLOV5+GWOCNN	98	98	97	95

The accuracy of the suggested method YPLOv5+GWOCNN (98%) is greater than the traditional approaches R-CNN (87.97%), FA-YOLO (92.95%), FESSD (83%) and YOLOv4+NSF (93%).

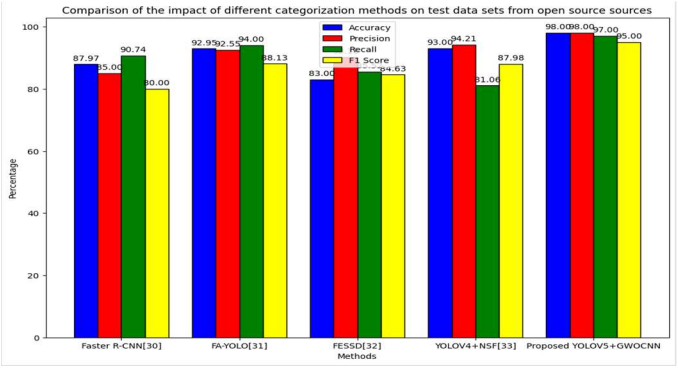

Fig. 10. The Performance Measures of the Suggested Yolov5+GWOCNN with Traditional Methods

Fig 10, shows a graphical representation of the suggested performance metrics in comparison to the current methods. The proposed method YOLOv5+GWOCNN demonstrates the highest accuracy across all five categories RCNN, FAYOLO, FESSD, YOLOV4+NSF, with 98% high accuracy.

Discussion

The proposed YOLOv5+GWOCNN model outperforms traditional detection methods in traffic accidents, exhibiting better performance in terms of MAE, MSE, accuracy, recall, F1-score, and precision YOLOv5 and Gray Wolf Optimization combining them enhances the sample feature extraction and classification capabilities Turns up out pre-processing techniques such as Gaussian filtering and Retinex model further improve image quality, and contribute to model robustness. These improvements make the YOLOv5+GWOCNN model a reliable choice for traffic control and accident prevention applications

CONCLUSION AND FUTURE WORKS

The YOLOv5+GWOCNN framework provides a significantly improved solution for object detection in traffic detection, which outperforms existing methods with more accuracy and reliability. The model's ability to recognize objects in time self-accuracy, coupled with its robustness under various conditions highlights the potential for traffic enhancement safety. Overall, the proposed model is well suited for applications in traffic detection, alert, and autonomous driving, providing improved safety and performance Future research may seek to extend YOLOv5+ The GWOCNN model has extended into other areas, such as pedestrian detection or facial recognition, for measurement and versatility. Moreover, the performance of the model can be further improved by integrating advanced algorithms such as particle swarm optimization (PSO) or genetic algorithm (GA). Expanding the model's validation across diverse datasets and real-world scenarios would also help in generalizing its application and ensuring its effectiveness in various contexts.To improve the precision and effectiveness of identifying objects in traffic scenarios in order to reduce accidents, this novel approach makes use of the strengths of deep learning, gradient boosting, and nature-inspired optimization. Future goals include integrating with existing infrastructure, addressing ethical and legal issues, and advancing technology as part of a comprehensive strategy for preventing traffic accidents.

References

[1] M. Ruikar, 'National statistics of road traffic accidents in India', *J Orthop Traumatol Rehabil*, vol. 6, no. 1, p. 1, 2013, doi: 10.4103/0975-7341.118718.

[2] K. Bhalla and K. Gleason, 'Effects of vehicle safety design on road traffic deaths, injuries, and public health burden in the Latin American region: a modelling study', *The Lancet Global Health*, vol. 8, no. 6, pp. e819–e828, Jun. 2020, doi: 10.1016/S2214-109X(20)30102-9.

[3] X. Chen, P. Wei, W. Ke, Q. Ye, and J. Jiao, 'Pedestrian Detection with Deep Convolutional Neural Network', in *Computer Vision - ACCV 2014 Workshops*, vol. 9008, C. V. Jawahar and S. Shan, Eds., in Lecture Notes in Computer Science, vol. 9008. , Cham: Springer International Publishing, 2015, pp. 354–365. doi: 10.1007/978-3-319-16628-5_26.

[4] S. C. H. Hoi *et al.*, 'LOGO-Net: Large-scale Deep Logo Detection and Brand Recognition with Deep Region-based Convolutional Networks'. arXiv, Nov. 13, 2015. Accessed: Sep. 22, 2023. [Online]. Available: http://arxiv.org/abs/1511.02462

[5] M. J. Akhtar *et al.*, 'A Robust Framework for Object Detection in a Traffic Surveillance System', *Electronics*, vol. 11, no. 21, p. 3425, Oct. 2022, doi: 10.3390/electronics11213425.

[6] L. Jiao *et al.*, 'A Survey of Deep Learning-Based Object Detection', *IEEE Access*, vol. 7, pp. 128837–128868, 2019, doi: 10.1109/ACCESS.2019.2939201.

[7] S. K. Pal, A. Pramanik, J. Maiti, and P. Mitra, 'Deep learning in multi-object detection and tracking: state of the art', *Appl Intell*, vol. 51, no. 9, pp. 6400–6429, Sep. 2021, doi: 10.1007/s10489-021-02293-7.

[8] H.-H. Jebamikyous and R. Kashef, 'Autonomous Vehicles Perception (AVP) Using Deep Learning: Modeling, Assessment, and Challenges', *IEEE Access*, vol. 10, pp. 10523–10535, 2022, doi: 10.1109/ACCESS.2022.3144407.

[9] Y. Zhou, S. Wen, D. Wang, J. Mu, and I. Richard, 'Object Detection in Autonomous Driving Scenarios Based on an Improved Faster-RCNN', *Applied Sciences*, vol. 11, no. 24, p. 11630, Dec. 2021, doi: 10.3390/app112411630.

[10] X.-Z. Chen, C.-M. Chang, C.-W. Yu, and Y.-L. Chen, 'A Real-Time Vehicle Detection System under Various Bad Weather Conditions Based on a Deep Learning Model without Retraining', *Sensors*, vol. 20, no. 20, p. 5731, Oct. 2020, doi: 10.3390/s20205731.

[11] F. Nobis, M. Geisslinger, M. Weber, J. Betz, and M. Lienkamp, 'A Deep Learning-based Radar and Camera Sensor Fusion Architecture for Object Detection'. arXiv, May 15, 2020. Accessed: Sep. 20, 2023. [Online]. Available: http://arxiv.org/abs/2005.07431

[12] Y. Lee, S. Lee, J. Yoo, and S. Kwon, 'Efficient Single-Shot Multi-Object Tracking for Vehicles in Traffic Scenarios', *Sensors*, vol. 21, no. 19, p. 6358, Sep. 2021, doi: 10.3390/s21196358.

[13] M. Yan, Z. Li, X. Yu, and C. Jin, 'An End-to-End Deep Learning Network for 3D Object Detection From RGB-D Data Based on Hough Voting', *IEEE Access*, vol. 8, pp. 138810–138822, 2020, doi: 10.1109/ACCESS.2020.3012695.

[14] Q. Chuai, X. He, and Y. Li, 'Improved Traffic Small Object Detection via Cross-Layer Feature Fusion and Channel Attention', *Electronics*, vol. 12, no. 16, p. 3421, Aug. 2023, doi: 10.3390/electronics12163421.

[15] C. Urmson and W. Whittaker, 'Self-Driving Cars and the Urban Challenge', *IEEE Intell. Syst.*, vol. 23, no. 2, pp. 66–68, Mar. 2008, doi: 10.1109/MIS.2008.34.

[16] M. A. Butt and F. Riaz, 'CARL-D: A vision benchmark suite and large scale dataset for vehicle detection and scene segmentation', *Signal Processing: Image Communication*, vol. 104, p. 116667, May 2022, doi: 10.1016/j.image.2022.116667.

[17] M. Umair, M. U. Farooq, R. H. Raza, Q. Chen, and B. Abdulhai, 'Efficient Video-based Vehicle Queue Length Estimation using Computer Vision and Deep Learning for an Urban Traffic Scenario', *Processes*, vol. 9, no. 10, p. 1786, Oct. 2021, doi: 10.3390/pr9101786.

[18] A. A. Micheal, K. Vani, S. Sanjeevi, and C.-H. Lin, 'Object Detection and Tracking with UAV Data Using Deep Learning', *J Indian Soc Remote Sens*, vol. 49, no. 3, pp. 463–469, Mar. 2021, doi: 10.1007/s12524-020-01229-x.

[19] Y. Djenouri, A. Belhadi, G. Srivastava, D. Djenouri, and J. Chun-Wei Lin, 'Vehicle detection using improved region convolution neural network for accident prevention in smart roads', *Pattern Recognition Letters*, vol. 158, pp. 42–47, Jun. 2022, doi: 10.1016/j.patrec.2022.04.012.

[20] M. M. Karim, Y. Li, R. Qin, and Z. Yin, 'A Dynamic Spatial-temporal Attention Network for Early Anticipation of Traffic Accidents'. arXiv, Dec. 20, 2021. Accessed: Sep. 22, 2023. [Online]. Available: http://arxiv.org/abs/2106.10197

[21] T. Liang, H. Bao, W. Pan, X. Fan, and H. Li, 'DetectFormer: Category-Assisted Transformer for Traffic Scene Object Detection', *Sensors*, vol. 22, no. 13, p. 4833, Jun. 2022, doi: 10.3390/s22134833.

[22] A. Mhalla, T. Chateau, S. Gazzah, and N. E. B. Amara, 'An Embedded Computer-Vision System for Multi-Object Detection in Traffic Surveillance', *IEEE Trans. Intell. Transport. Syst.*, vol. 20, no. 11, pp. 4006–4018, Nov. 2019, doi: 10.1109/TITS.2018.2876614.

[23] Y. Li *et al.*, 'A Deep Learning-Based Hybrid Framework for Object Detection and Recognition in Autonomous Driving', *IEEE Access*, vol. 8, pp. 194228–194239, 2020, doi: 10.1109/ACCESS.2020.3033289.

[24] T. Ye, Z. Zhang, X. Zhang, and F. Zhou, 'Autonomous Railway Traffic Object Detection Using Feature-Enhanced Single-Shot Detector', *IEEE Access*, vol. 8, pp. 145182–145193, 2020, doi: 10.1109/ACCESS.2020.3015251.

[25] S. Zhang, S. Wang, S. Yu, J. J. Q. Yu, and M. Wen, 'Collision Avoidance Predictive Motion Planning Based on Integrated Perception and V2V Communication', *IEEE Trans. Intell. Transport. Syst.*, vol. 23, no. 7, pp. 9640–9653, Jul. 2022, doi: 10.1109/TITS.2022.3173674.

[26] X. Ma, K. Ji, B. Xiong, L. Zhang, S. Feng, and G. Kuang, 'Light-YOLOv4: An Edge-Device Oriented Target Detection Method for Remote Sensing Images', *IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing*, vol. 14, pp. 10808–10820, 2021, doi: 10.1109/JSTARS.2021.3120009.

[27] 'Traffic Detection Project'. Accessed: Sep. 26, 2023. [Online]. Available: https://www.kaggle.com/datasets/yusufberksardoan/traffic-detection-project

[28] Y. Zhang, 'Low Light Image Enhancement and Saliency Object Detection', 2022.

[29] D. E. King, 'Dlib-ml: A Machine Learning Toolkit', 2009.

[30] J. Zhang *et al.*, 'Multi-class object detection using faster R-CNN and estimation of shaking locations for automated shake-and-catch apple harvesting', *Computers and Electronics in Agriculture*, vol. 173, p. 105384, Jun. 2020, doi: 10.1016/j.compag.2020.105384.

[31] S. Du, B. Zhang, P. Zhang, P. Xiang, and H. Xue, 'FA-YOLO: An Improved YOLO Model for Infrared Occlusion Object Detection under Confusing Background', *Wireless Communications and Mobile Computing*, vol. 2021, pp. 1–10, Nov. 2021, doi: 10.1155/2021/1896029.

[32] J. Guo, Z. Wang, and S. Zhang, 'FESSD: Feature Enhancement Single Shot MultiBox Detector Algorithm for Remote Sensing Image Target Detection', *Electronics*, vol. 12, no. 4, p. 946, Feb. 2023, doi: 10.3390/electronics12040946.

[33] Y. Guo *et al.*, 'Improved YOLOV4-CSP Algorithm for Detection of Bamboo Surface Sliver Defects With Extreme Aspect Ratio', *IEEE Access*, vol. 10, pp. 29810–29820, 2022, doi: 10.1109/ACCESS.2022.3152552.

[34] C. Dewi, R.-C. Chen, Y.-T. Liu, X. Jiang, and K. D. Hartomo, 'Yolo V4 for Advanced Traffic Sign Recognition With Synthetic Training Data Generated by Various GAN', *IEEE Access*, vol. 9, pp. 97228–97242, 2021, doi: 10.1109/ACCESS.2021.3094201.

[35] L. Zhao and S. Li, 'Object Detection Algorithm Based on Improved YOLOv3', *Electronics*, vol. 9, no. 3, p. 537, Mar. 2020, doi: 10.3390/electronics9030537.

[36] H. Kong, Z. Chen, W. Yue, and K. Ni, 'Improved YOLOv4 for Pedestrian Detection and Counting in UAV Images', *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–9, Jul. 2022, doi: 10.1155/2022/6106853..