International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys.14, No.1 (Jul-23)

http://dx.doi.org/10.12785/ijcds/140109

A Coupled System to Detect Pedestrians Under Various Intricate Scenarios for Design and Implementation of Reliable Autonomous Vehicles

Mohit Kumar[1], Gurram Sahithi Priya[1], Praneeth Gadipudi[1] and V. M. Manikandan[1]

[1]*Department of Computer Science and Engineering, SRM University-AP, Andhra Pradesh, India.*

Received 18 Apr. 2022, Revised 6 May. 2023, Accepted 13 May. 2023, Published 1 Jul. 2023

Abstract: The pedestrian detection algorithm (PDA) is one of the most widely used techniques in modern automated vehicles, surveillance systems, human-machine interfaces, intelligent cameras, robots, etc. Despite considerable work in this field, PDA is still receptive to several scopes of advancements considering some adverse weather conditions like fog, rain, low visibility, etc. Along with this, there are certain intricate scenarios where the accuracy of a given PDA becomes contentious. As we are progressing toward autonomous vehicles, it becomes vital for such vehicles to ensure the safety of both passengers and pedestrians walking around the road. To do so, they require a much more reliable and effective pedestrian detection system capable of working under adverse conditions. This paper considers all such issues to develop certain machine learning (ML) and deep neural network (DNN) methods to solve such issues. YOLOv4 is a deep learning-based object identification method that is currently functioning well yet is not robust. The core premise of YOLOv4 is initially explored and evaluated in this paper to discover its importance in our task. This research devises a coupled system capable of detecting pedestrians under various adverse and intricate scenarios. To do so, we use the YOLOv4 object detection technique coupled with some image denoising, low light enhancement and image dehazing features. We are using the wavelet and YCbCr methods for image denoising and low-light enhancement. To dehaze the video frames, we use airtight estimation and tuning the transmission by deriving the boundary constraints. The paper tries to cover most of the aspects that an autonomous vehicle may face while on the road. Overall, we deliver a reliable model that fosters more accuracy even in complex scenarios and unfavourable weather conditions.

Keywords: Pedestrian Detection, Machine learning, Deep neural network, Image Denoising, Wavelet, YCbCr model, Image Dehazing; YOLOv4;

1. Introduction and Overview

With the ever-evolving situations and endless exertions in technology and science, we are moving towards a high-tech world that provides us with facilities that reduce human efforts and make our life comfortable and straightforward. In the early 1880s, the first car was introduced by Karl Benz, and by the current era, we have seen several expansions in the field of the automobile industry. One such innovation is autonomous vehicles, an advancement in the machinery and automobile industry. Autonomous vehicles or self-driving cars are the future of the automobile industry, which deliver consumers a safe drive to their destination without requiring a human driver. In other words, these cars are powered by AI and ML technologies that assist in finding the safest route and driving themselves to reach their destination. Several enterprise goliaths like Tesla, Apple, Waymo, Kia-Hyundai, Huawei, and Nvidia have turned to autonomous vehicles and related technologies.

Along with these companies, several scholars and engineers function on the required technologies and bring up several innovative ideas to make autonomous vehicles more reliable and efficient. Autonomous vehicles come up with several modes of operation, initiating from level 0 to level 5. Under these several modes, the car has additional capabilities, and while moving on to level 5, the level of automation also increases. At level 5, the vehicle is competent enough to decide without even taking a small input from any driver. The objective is to make the journey effortless, comfortable, and unassailable.

One of the essential aspects of autonomous vehicles is to tackle the objects that come in the way. It's not just the traffic lights, traffic signs, zebra crossings, or congested roads. The road also has pedestrians strolling around. It is vital to detect and distinguish between a pedestrian and other objects present on its way. The recent advancements in DNN and computer vision have brought pedestrian detection into reality [1]. Pedestrian detection uses the input video provided by the car on its journey and then applies

E-mail address: mohitkumar_mahto@srmap.edu.in, sahithipriya_gurram@srmap.edu.in, praneeth_gadipudi@srmap.edu.in, manikandan.v@srmap.edu.in

the proposed scheme to that input video to filter out frames and detect if there is a pedestrian in the path of the vehicle or not. The output of the detection model is then used as input to decide whether to slow down the speed or apply brakes.

The aim is to lessen the risk of accidents and minimize the loss of lives due to vehicles moving on the road. If the model fails to distinguish the pedestrian from other objects on the road, it may lead to accidents. Despite numerous works done in this field, it remains a topic of interest because of its significance in the future. In comparing the several works done in this field, we came across several challenges and limitations that the earlier models have. Considering that, we propose a more reliable and optimized strategy than the ones existing before. Our proposed approach is robust enough to resist several intricate scenarios and conditions described below.

- *Adverse Weather Conditions:* Weather conditions are an indispensable factor affecting vehicle performance, passengers, and pedestrians' safety. Many features are coming up for automated vehicles with sensors and software to overcome various environmental circumstances. Still, most of them are not suitable to encounter adverse weather conditions [2]. Fog, hazy weather, snow, and heavy rain can negatively impact and severely limit the performance of sensors and cameras. Therefore the erroneous information given by the sensors in these conditions can lead to car crashes or severe accidents.

- *Obstacle detection:* Autonomous vehicles have to detect obstacles and various objects to decide whether to stop at a considerable distance or move further to avoid collisions. When a vehicle is set to its destination, one wouldn't know the different obstacles an autonomous vehicle can encounter.

- *A diverse range of pedestrians:* There are thousands of ethnicities, cultures, subcultures, and religions among today's billions of people. Pedestrians might have different body shapes, clothing styles, and skin tones, creating hazards for the current models to detect and distinguish pedestrians.

- *Various illumination conditions:* Uncontrolled illumination circumstances generate undesirable results on sensory data, which is a typical challenge in autonomous vehicles [3]. We cannot expect that Autonomous vehicles' surroundings will have an ample amount of light all the time to capture the snaps. Illumination plays a vital role in capturing images accurately. So, in the nighttime or daytime, or the various lighting conditions, the vehicle should capture the images efficiently and make the proper conclusions.

- *Speed and position of the vehicle:* Speed plays a crucial role in any vehicle. Autonomous vehicles should adjust their speed according to the surrounding road conditions. While making decisions, it should also consider the permitted speed, road signs like school zone, U-turns, etc [4]. If there is a massive crowd of people on the road, the car should determine which speed to reach the destination smoothly without causing any accidents.

- *Synchronization and optimization:* Even though the working of the sensors is exemplary, if it is not synchronized with the vehicle's machinery, it would affect the judgments and lead to mishaps. Let's take a scenario where a person is crossing the road, and the sensors are not appropriately synchronized with the machinery and make a delay in the decision. It may lead to collisions and accidents.

The several unfavourable scenarios and conditions are graphically shown in *Figure 1*.

The proposed approach is not just confined to autonomous vehicles and can be applied widely to other domains, too, as mentioned below.

- *Disaster management and rescue operations*: When there is a flood, a landslide, or any natural calamity, the roads get clogged. It becomes pretty complicated for the rescue team to locate a route to reach the affected area. People stuck there require immediate help, and such a blockage causes a delay which makes people deprived of any assistance. We also see rescue helicopters going over those areas to distribute snacks and meals to the people, but they can't recognize everyone from such a height.
  In that case, we can use several drones that can carry small meal packets and are powered by human and animal detection modules. These drones can then self-go to those areas and pinpoint where there are humans or animals stuck and then provide them meals and transmit the report to the rescue team. The rescue team can then determine where humans or animals are stuck and rescue those areas. So in such cases, our proposed scheme can assist in identifying the humans stuck in those locations.

- *Security surveillance systems*: Public places like railway stations, airports, and private regions like residences and universities require surveillance cameras to ensure higher security. To enhance their safety, they can use pedestrian detection systems coupled with their security cameras to enable the security personnel to monitor and track people's activity around the areas. We comprehend that nobody can sit near the camera 24/7 to scrutinize what is happening, so this system can ensure that everything is alright. The coupled surveillance system can notify the users or security whenever any unusual activity is detected.

Figure 1. Unfavorable scenarios encountered by the pedestrian detection system.

- *Human-Robot interaction*: Robots are getting trained and employed at many places like research centres and hospitals. Soon several other businesses like restaurants, tourist places, offices, etc., will also start using robots to handle their work. For a random scenario, just presume that you visited a restaurant and the robot instantly comes near your table to collect your order and then fetch your meals. For instance, robots can guide humans about tourist places and offer them a whole tour. To accomplish such things, the first step for the robot is to recognize a human and then interact with them.

- *Traffic regulations and safety*: The traffic regulations department has cameras to scrutinize cars and people who break safety rules in many places. If any vehicle breaks a signal, the vehicle number plate is immediately captured and sent to the officials for action. Indeed, the cameras powered by pedestrian detection modules can monitor people breaking the rules or track any harmful road activity.

- *Workplace monitoring*: Whether it's an office, university, or school, everyone has staff to manage the work. There are several security cameras in separate wings of the building or office rooms to keep an eye on them, but nobody can constantly monitor every staff member. The proposed scheme can help detect humans and identify their activity in such cases. For instance, if a staff member sleeps in his cabin or is operating a mobile during work hours, that activity can be immediately tracked and noted in the records.

The proposed schema in our paper resolves the existing issues, overcomes the intricate scenarios, and optimizes the outcome of pedestrian detection, which unlocks and advances the routes for autonomous vehicles. This document's contents are categorized as follows: In 2, the paper highlights the different relevant research already done in the field of pedestrian detection and attempts to outline the limitations of such systems. Section 3 describes the proposed scheme, including the terminologies, algorithms, libraries used, and implementation. Section 4, the paper presents the analyzed outcomes and results obtained via trials through pictorial, tabular, and descriptive formats.

Section 5 summarizes the entire work and deals with this scheme's future scope.

2. Related Work

In search for a practical approach for pedestrian detection, several techniques came into existence, for instance, region-based convolutional neural networks (RCNN) [5], Fast RCNN [6], and Faster RCNN [7]. These algorithms create region proposal networks (RPN) [8], and the region recommendations are then classified. Single-shot detectors (SSD) [9] and YOLO [10], for example, are object detection algorithms that use regression. These techniques similarly generate region proposal networks (RPN); however, these RPNs are sorted into categories during production. All of the techniques discussed above are effective in object recognition and localization.

In the case of the RCNN approach, it finds a collection of boxes for the picture using a selective search and then examines each box to see whether it contains a subject [5]. Each picture splits into 2,000 portions based on a selective search [10]. We must use convolutional neural networks (CNN) to choose features for each area or chunk of the picture. RCNN takes around 40-50 seconds of computation time. In Fast RCNN, each image is run through CNN only once to extract the characteristics. All of the different models are integrated to generate a single model and to obtain results for target recognition, and it uses a selective search strategy using feature maps [10]. This procedure is time-consuming and takes a long time to complete. Faster RCNN is a variation of RCNN that uses an RPN and produces a collection of item proposals as well as a score for each recommendation, taking feature maps of an image as input [8]. In the case of Faster RCNN, numerous passes are required to extract all of the targets in a single image. Diverse systems are functioning in succession; consequently, the success of the forthcoming operation relies on the effectiveness of earlier processes.

SSD is one of the most extensively used techniques for detecting pedestrians and other objects. SSD achieves an outstanding blend of speed and precision in its output. SSD applies a neural network-based model on the input picture for a set instant to compute the feature map [9]. It also uses anchor boxes at varying dimensions, equivalent to faster RCNN, and analyzes the offset rather than identifying the box. A neural network's bottom layers may not have enough high-level information to anticipate tiny objects. As a result, SSD performs poorly for tiny objects.

Table I discusses and illustrates some of the relevant efforts on the subject of pedestrian detection, as well as their conclusions and limitations.

3. Proposed Scheme

Object detection and classification are employed in various areas of human life, including health, automobile, and education. Because all of these industries are fast evolving, one-stage models must likewise evolve to meet their needs. Pedestrian detection is a type of object detection that needs a more reliable and optimized approach. As we know, pedestrians stroll around the road and rarely stay in a single place. The paper aims to detect moving pedestrians with higher accuracy in such circumstances. Considering the limitations of the above methods, we are here with an improved and robust approach to detecting moving pedestrians.

The manuscript proposes a coupled system that consists of three different components. Firstly, we take the input video and break it into a sequence of frames, and then in each frame, we try to remove haze if present. Then the same frame goes through our image denoising module, where the frame is enhanced, and low light conditions are tackled. The above two processes filter the frame and smooth it in such a way that the originality of the frame is intact and the picture is clear of any disturbance or noise.

The frame is then passed over the YOLOv4 model to predict whether there is a pedestrian on the road. Using YOLOv4 not just makes the process simple but also increases efficiency. Comparing YOLOv4 with other existing schemes, we found that YOLOv4 is much more suitable for object detection and classification. *Figure 2* shows the proposed scheme's flow and describes the implementation aspect of our model.

A. The relevance of YOLOv4 in our suggested approach

YOLOv4 is a project developed by Google to teach computers how to detect pedestrians in the vicinity of intersections. It's a scientific approach, similar to the Bayesian probability rule, which allows computers to scale complex theoretical models and determine if an event might occur or not. The current model has YOLO error rates that seem reasonable for self-driving cars. However, it's still too early in its development stages, and there are emerging flaws in its ability that have yet to be solved.

Google used data collected from the University of Washington Computer Science and Engineering Department to run the model [11]. They went on to apply it to a Volkswagen Golf Sportwagen (an actual automobile). In this way, they could test their algorithm in a much more complex environment than parking lots and intersections. Despite this, it's tough to predict how well their algorithm will function in a real-world setting, considering the number and types of obstacles autonomous vehicles could encounter.

YOLO error is a type of error that describes the difference between what a computer or other device predicts and what happened. According to their predictions, if something occurs, YOLOv4 will detect it. However, it's deemed an error if it does not occur. YOLOv4 can identify real-world objects in its vicinity with a relatively high degree of accuracy. However, some issues may arise in the future when cars rely on this algorithm to make decisions that affect their safety and the safety of others on the road.

TABLE I. RELATED WORKS

Scheme	Research Methodology	Findings and Drawbacks
Scheme [5]	RCNN approach finds a collection of boxes for the picture using a selective search and then examines each box to see whether it contains a subject or not.	We must use convolutional neural networks (CNN) to choose features for each area or chunk of the picture. RCNN takes around 40-50 seconds of computation time.
Scheme [6]	In Fast RCNN, each image is run through CNN only once to extract the characteristics. In RCNN, all different models are integrated to develop a single model.	To obtain results for target recognition, it uses a selective search strategy which is time-consuming and takes a long time to complete.
Scheme [7]	Faster RCNN is a variant of RCNN that uses a region proposal network (RPN) and takes image feature maps as input and outputs a set of object suggestions and a score for each recommendation.	In the case of Faster RCNN, numerous passes are required to extract all targets in a single image. Diverse systems are functioning in succession; consequently, the success of the forthcoming operation relies on the effectiveness of earlier processes.
Scheme [9]	SSD implements a CNN-based model to the input image for a fixed moment to compute the feature map.	A neural network's bottom layers may not have enough high-level information to anticipate tiny objects. As a result, SSD performs poorly for tiny objects.

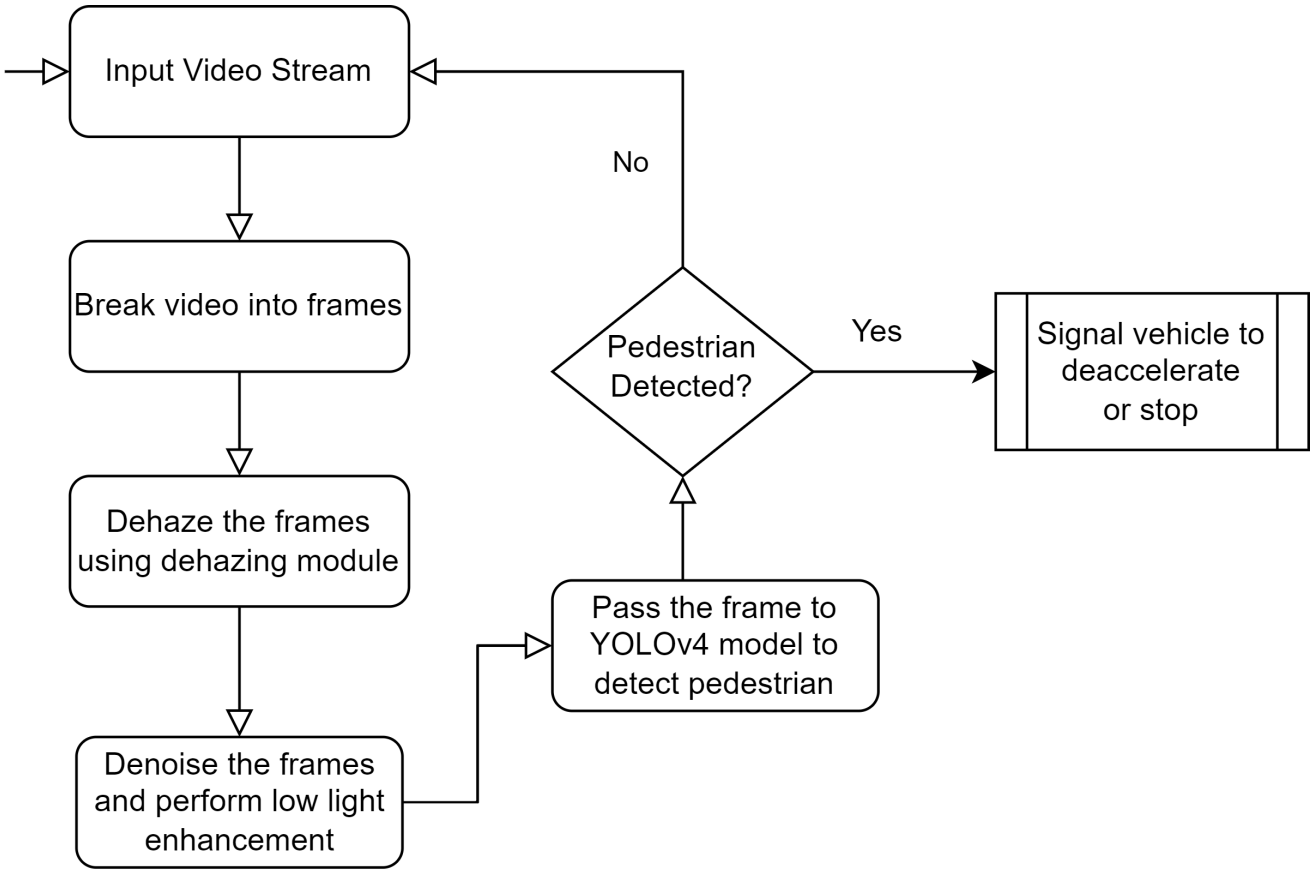


Figure 2. Working of the coupled system

YOLOv4 has been tested on many different automobiles, including vehicles from Toyota, Tesla, and Volkswagen Golfs. In some of these tests, the car experienced a failure and hit a curb or went over a traffic cone. These types of errors can be hazardous and could result in serious injuries or fatalities.

Google has continued to test its algorithm on different automobiles that have encountered problems with steering and braking. These are problems that are essential to any self-driving car. Due to its continuous evolution and experiments carried out by several organizations, it makes YOLOv4 the most suitable algorithm to be used here.

B. Enhancing the visibility by dehazing the input frame

Before people can have autonomous vehicles, it is necessary to have cars that can drive in all types of conditions. These conditions include driving through dense, hazy areas where the visibility rate is meager. To create a working autonomous vehicle for these types of conditions, we must be able to see through hazy areas and recognize objects against the backdrop. For example, if there is dense fog or heavy rain, even a regular driver may feel it difficult to differentiate the objects. Autonomous cars use pre-programmed maps that contain information about the car's surroundings, including road shapes and topography. These maps are typically derived from a combination of high-altitude aerial photography and low-resolution street-view imagery. Unfortunately, these maps have limited detail for areas with fog or heavy rain, both of which can cause poor visibility for autonomous vehicles.

Moreover, such autonomous vehicles cannot discern other objects that might obstruct their paths, such as fallen trees or people on roadsides. Thus, better visibility algorithms are needed to address the challenges raised by poor weather and the limited information found in maps. Dehazing is used to improve the view of the road ahead.

Garbage in garbage out (GIGO) is a popular terminology in the computer science field, implying that if the input provided is inappropriate, even the output wouldn't be suitable. The input is a video feed of the pedestrian or general road area for our proposed schemes. As discussed earlier, we know that several intricate weather conditions like fog or haze can lower the visibility in the video captured by the cameras. When provided with the proposed scheme, the same video will, in turn, result in the wrong or inappropriate identification of pedestrians. To make the proposed system more accurate and reliable, we are coupling the scheme with a visibility enhancement model that takes the hazed or foggy image as input and tries to enhance the image by dehazing it.

In the algorithm, first, we estimate the airlight (A), which is the global atmospheric light (the brightest light in the picture, for example, the bright sky or headlights of a car). Here, we estimate airlight (A) by beginning with filtering each color channel of an input image by a minimum filter with a moving window, so the maximum value of each color channel is taken as an estimate [12]. Then we estimate and tune the transmission function to recover the scene radiance and the observed image. The scene radiance imposes a boundary constraint on the transmission function, so we need to calculate the boundary constraints. These constraints provide a new perspective to the existing algorithm dark channel prior. These boundary constraints still hold even though brighter pixels come from various light sources where the dark channel prior fails to this condition. Now we will perform the dehazing by using the estimated airlight and transmission.

To estimate the airlight(A), we apply a minimum filter to each color channel of the input image, which can be represented as follows:

Let I be an input image, and I_r, I_g, I_b be its red, green, and blue color channels, respectively. Then, the filtered image I_min can be defined as:

$$I\_min(x,y) = min(I\_r(x,y), I\_g(x,y), I\_b(x,y))$$

The maximum value of each color channel is taken as an estimate for the airlight (A), which can be represented as follows:

$$A = max(I_r(x,y)) + max(I_g(x,y)) + max(I_b(x,y))$$

To estimate and tune the transmission function, we need to calculate the boundary constraints. Let J be the observed hazy image, and J_r, J_g, J_b be its red, green, and blue color channels, respectively. The scene radiance L can be estimated by dividing the observed image by the transmission function T, which can be represented as follows:

$$L\_r(x,y) = J\_r(x,y) \,/\, T\_r(x,y)$$

$$L\_g(x,y) = J\_g(x,y) \,/\, T\_g(x,y)$$

$$L\_b(x,y) = J\_b(x,y) \,/\, T\_b(x,y)$$

To calculate the boundary constraints, we use the following formula:

$$B = 1 - w \times min(T\_r(x,y),\ T\_g(x,y),\ T\_b(x,y))$$

where w is a weight parameter, and B represents the boundary constraints.

Finally, the dehazed image can be obtained by applying the estimated airlight (A) and transmission function (T) to the observed image J, which can be represented as follows:

$$J\_dehazed(x,y) = (J\_r(x,y) - A)/max(T\_r(x,y), t\_min) + A$$

$$J\_dehazed(x,y) = (J\_g(x,y) - A)/max(T\_g(x,y), t\_min) + A$$

$$J\_dehazed(x,y) = (J\_b(x,y) - A)/max(T\_b(x,y), t\_min) + A$$

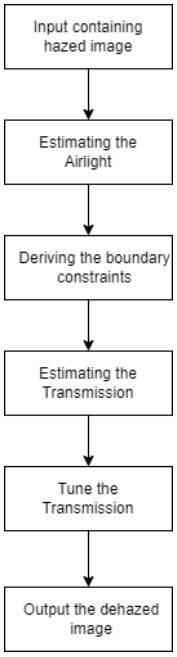


Figure 3. Performing dehazing by using these Airlight and Transmission

where t_min is a small positive constant to avoid division by zero.

This method is applied to a collection of frames having haze in them from a particular video data set to evaluate its effectiveness, yielding better performance results. Image denoising is applied after dehazing to reduce over-smoothing and sharpening artifacts, ensuring a realistic view. *Figure 3* graphically describes the image dehazing process.

C. Image Denoising and low light enhancement

Image denoising is a relevant subject in a variety of image processing and computer vision challenges. The most crucial feature of a good image-denoising model is that it should eliminate noise as thoroughly as feasible while still preserving edges. In general, image sensor data sets are tainted by noise. Imperfect instrumentation, issues with the data-collecting method, and interacting natural events can all skew the results. Transmission faults and compression can also introduce noise [13].

A wavelet is a waveform with an effective duration of zero and an average value of zero. Wavelet algorithms are useful tools for signal processing such as image compression and denoising. There are two main types of wavelet transformations, Continuous Wavelet Transform(CWT) and Discrete Wavelet Transform(DWT). The most commonly used denoising method based on wavelet coefficients is the soft thresholding strategy, which has been theoretically justified by Donoho and Johnstone in their paper [14] [15].

Signal denoising is one of the important aspects of the wavelet transform. The detailed coefficients at each level are thresholded. We can achieve signal denoising or filtering by cutting all these detail coefficients at different levels. Applications of various threshold estimation techniques again find the threshold levels. When a signal is transmitted over some distance, it is frequently contaminated by noise. The simplest model for the acquisition of noise by the signal is additive noise. The below-mentioned formula represents the addictive formation of noise to the original signal.

$$g'(x,y) = g(x,y) + n(x,y)$$
$$g'(x,y) = contaminatedsignal$$
$$g(x,y) = Originalsignal$$
$$n(x,y) = Noisesignal$$
$$x, y \rightarrow frameslocations$$

Assuming that our noise is addictive and random (white Gaussian noise with 'zero' mean). Our objective is to remove noise n(x,y) from the noisy image signal wavelet.

The signal $g(x,y)$ is first decomposed into the wavelet coefficients, approximation coefficients, and detail coefficients, then these coefficients are thresholded, and then these coefficients are selected and by taking inverse discrete wavelet transform, synthesis, or reconstruction we will get our signal back which is a denoised signal. We have various methods for estimating the noise level and setting the threshold differently such as Universal threshold(VishuShirk), Bayes, SURE, MinMaxi etc.

We have done our programming part in Python language, where we used the scikit-image package for performing the image denoising. The method we used is the Universal threshold(VishuShirk) as threshold and noise estimation. For the colour image denoising, the following parameters are included in the wavelet_denoise() method and they are multi-channel, and convert2ycbcr. The multi-channel means if it's true the wavelet denoising is done separately for each channel because the colour images are RGB. Convert2ycbcr is true and multi-channel is true, wavelet denoising is done in the YCbCr colour space instead of the RGB colour space. This typically results in better performance than RGB images. *Figure 4* pictorially represents the process of denoising the image.

D. Detecting Pedestrians using the coupled system

So, the two subsections covered the aspects of how the input frame quality can be improved in various intricate scenarios. Now we will use this improved image for our further process. Pedestrians are one of the many object classes that YOLO can detect, alongside cars, bicycles, traffic lights, and more. The YOLO algorithm is based on a deep neural network that is trained on a large dataset of images with labelled objects. During training, the algorithm learns to recognize patterns and features that are common among different object classes, including pedestrians. When

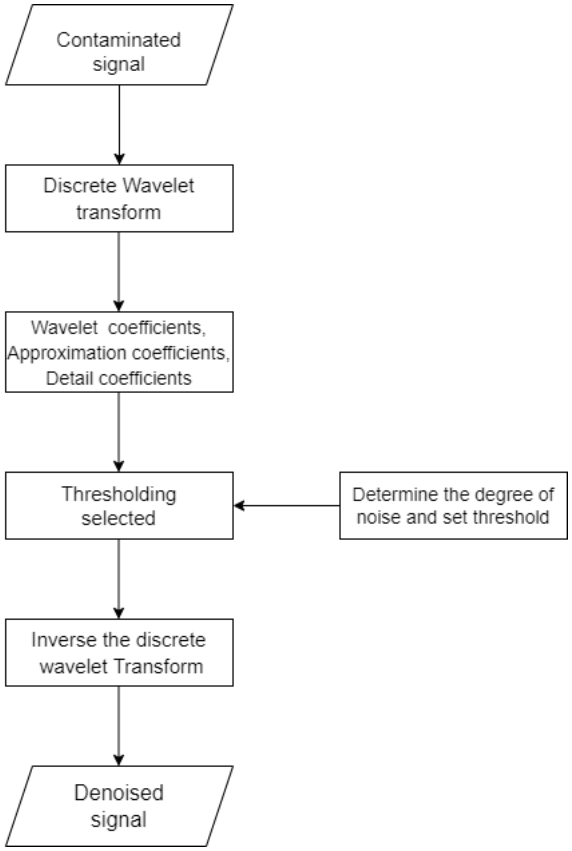


Figure 4. Performing denoising over the obtained frame

YOLO detects a pedestrian, it distinguishes it from other types of objects based on a combination of factors. These include the shape, size, and texture of the object, as well as its relative position and context within the image.

For example, a pedestrian is typically upright and has a roughly human shape, which helps YOLO distinguish it from other objects like cars, bicycles or traffic lights. Additionally, YOLO uses a probabilistic approach to object detection, where it assigns a probability score to each object class that it detects. The score reflects the confidence of the algorithm that the object belongs to a specific class. For example, if YOLO detects an object that has a high probability score for the pedestrian class, it is more likely to be a pedestrian than an object with a lower score. In summary, YOLO distinguishes pedestrians from other types of objects based on a combination of visual features and contextual information, and it assigns a probability score to each detected object class to reflect its confidence in the classification.

In our scenario, the proposed scheme uses an improved version of YOLO known as scaled YOLOv4, which is comparatively much faster and more accurate than the previous versions. So, using YOLO, we are not just trying to detect the pedestrian but also localizing the pedestrian on the given image frame. By localizing, we mean showing the person and their position in the frame, signified by a bounding box.

YOLO divides a given frame into grids of 4 by four and makes a vector for each grid. So, in our case, we use a vector with the following parameters: Pc, Bx, By, Bw, Bh, Pd. Pc denotes the probability of having an object in a given frame. Pc can take on two values, 0 or 1, depending on if an object is present in the frame. For instance, if there is any object in the frame, Pc is set to 1; else, 0. Coming on Bx, By, Bw, and Bh specify the coordinates of the bounding box around the object. Pd is a parameter that takes on two values, 0 or 1, depending upon the object. For instance, if the object is classified as a pedestrian, we will set Pd to 1 else 0. So using this single vector, we can quickly analyze whether or not the video frame has a pedestrian. Ultimately, we try to develop all the bounding boxes around all the pedestrians present in the frame in one forward pass.

While working on YOLO, we found some issues with multiple bounding boxes for the same pedestrian. Hence, we sorted that issue using the IOU concept, known as intersection over the union. So, using IOU, we find an overlapping area of the bounding boxes for a pedestrian and then divide it by the total area these bounding boxes are taking up. If the resultant was high, we discarded those overlapping boxes and kept the box having the highest class probability for the pedestrian. *Figure 5* pictorially demonstrates the pedestrian detection process using scaled YOLOv4.

E. Real-time processing while performing pedestrian detection

Real-time processing is essential for pedestrian detection, as autonomous vehicles need to make decisions quickly to avoid accidents. The processing time can be impacted by factors such as the complexity of the detection model, the image resolution, and the hardware capabilities. Scaled YOLOv4 is faster than the original YOLOv4 model due to several optimizations made to the architecture. First, the backbone network of the model has been modified to use CSP (cross-stage partial) blocks, which allows for better feature representation and reduces the number of parameters. Second, the model uses a spatial pyramid pooling (SPP) module, which extracts features at multiple scales and improves the model's ability to detect objects of different sizes. Third, the model uses a path-aggregation network (PAN) that combines features from different levels of the network and helps the model better handle objects at different scales.

Finally, the model uses a dynamic anchor assignment technique that adapts the size and aspect ratio of the anchor boxes based on the input image size, improving the accuracy of object detection. All these optimizations work together to make scaled YOLOv4 faster and more accurate than the original YOLOv4 model. By improving processing time, autonomous vehicles equipped with pedestrian detection

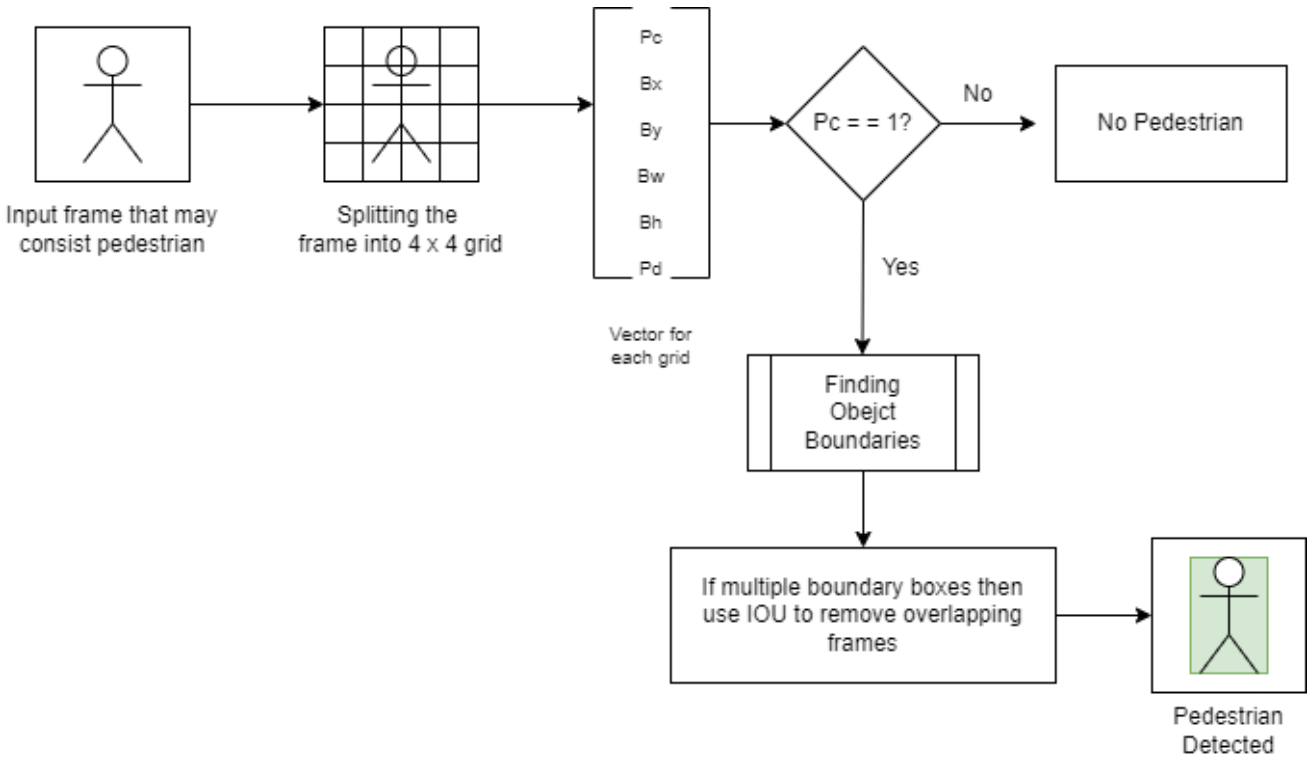


Figure 5. Detecting Pedestrian using Scaled YOLOv4

models can react more quickly and potentially reduce the number of pedestrian accidents on roads.

F. Libraries used in the proposed scheme

The proposed scheme was simulated in Python due to the variety of libraries available there. The first step was to collect the input, a video file. Then in the next step, we read the frames available in the input video. After that, each frame was passed to the visibility enhancement module, which returned the enhanced frame. The enhanced frame is further passed to the scaled YOLOv4 object detection module to detect pedestrians on road. The libraries that assisted us in this testing process were as follows:

- *OpenCV:*
  OpenCV is an open-source library capable of handling data stored in an image or a video. OpenCV assists in doing several AI/ML-based tasks. It also has some rich components that let us scan the elements present in a video or an image and then detect objects, persons, etc., from it. OpenCV played a crucial role in our model by helping us to scan each frame from the video and applying the classifier to those frames to identify the people.

- *imutils:*
  Imutils is also an open-source library capable of doing most image-processing tasks such as image resizing, rotation, translation, etc. Since the input video can contain frames of different dimensions, we use imutils to convert them to a fixed width for the classifier not face issues with the changing dimensions.

- *image_dehazer:*
  Image_dehazer is an open-source library that assists in the visibility enhancement of images that are subject to fog or haze. In the schema proposed, we needed a pre-built model to scan an input frame and enhance the quality and visibility of the frame.

- *numpy:*
  Numpy is a package that is used for statistical computing. We used numpy to perform efficient calculations with large multi-dimensional arrays and many mathematical functions.

- *Scipy:*
  It is an extension of Numpy, which helps manipulate, visualize and analyze data. We used scipy to solve various scientific and statistical problems. The whole imaze_dehazer module is built over numpy and scipy.

G. Implementation of the proposed scheme in autonomous vehicles

For self-driving vehicles to plan accurate and dependable reactions, they must observe and comprehend the behaviours of other road users. So to implement pedestrian detection in autonomous cars, a wide range of sensors and mounted cameras are used—various technologies like

RADAR(radio detection and ranging) and LiDAR(light detection and ranging). A revolving LiDAR sensor on the vehicle's top will collect a 360-degree field of vision at a high rate of speed, providing a comprehensive picture of the vehicle's surroundings and identifying the far objects. But the disadvantage is that it doesn't work well in bad weather. Computer vision, which is primarily based on machine learning techniques, allows autonomous vehicles to carry out tasks, including object detection, recognition of pedestrians, measurement evaluation, such as distance and speed, traffic sign knowledge, etc. [16]. There are some problems involved as well like, the sensors need to detect the surroundings in various weather conditions, and mounted cameras should get a clear image after dehazing. *Algorithm 1* clearly explains the algorithm which we are following in the proposed scheme. *Figure 6* visually depicts the execution of proposed system.

Data: A video file V
Result: A signal of 1 or 0 indicating whether the pedestrian is detected or not detected
$FLAG = 0$;
while V *is not empty* **do**
  Extract one by one the T Frames from the video provided by the camera mounted over the autonomous vehicle. Call the obtained frame as X ;
  Pass the X frame to the image dehazing module and get an enhanced frame by removing the haze or fog(if any). Call the processed frame as Y ;
  Apply the image denoising and low light enhancement over the processed frame Y to get an improved frame ;
  Pass the obtained frame to the YOLOv4 model and detect if any pedestrians in the path ;
  if $PEDESTRIAN == DETECTED$ **then**
    Request the micro-controller to apply brakes and decelerate;
    return $FLAG = 1$;
  else
    return $FLAG = 0$;
  end
end

Algorithm 1: Proposed scheme for implementation of pedestrian detection in autonomous vehicles.

4. Experimental Results

Pedestrian detection is challenging in computer vision, as pedestrians can vary greatly in appearance, pose, and occlusion. There are several places other than roads where we see people walking around. We examined the results of our proposed scheme with videos of several areas and scenarios to test the proposed scheme's likelihood. Combining scaled YOLOv4 with image dehazing and denoising improves the accuracy and robustness of pedestrian detection. Here's a justification for the same. Scaled YOLOv4 has achieved state-of-the-art performance in several object detection benchmarks such as COCO and VOC, indicating that it can effectively detect pedestrians in complex environments. Additionally, image dehazing can improve the visibility and clarity of hazy images, while image denoising can remove noise and artifacts from images. This can improve the accuracy and robustness of pedestrian detection algorithms. To evaluate the effectiveness of this approach, we have used metrics such as precision, recall, and F1-score.

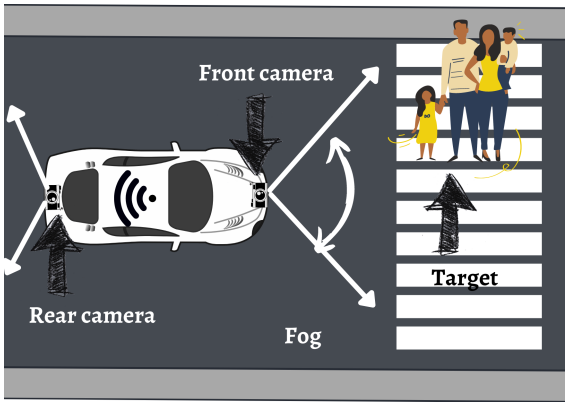


Figure 6. Pictorial representation of the proposed system

$$Precision(P) = T_P/(T_P + F_P)$$

$$Recall(R) = T_P/(T_P + F_N)$$

$$F1 - Score = 2 \times (P \times R)/(P + R)$$

where T_P denotes true positives, F_P denotes false positives, F_N denotes false negatives.

These metrics measure the algorithm's performance in detecting true positives and negatives and provide a balanced measure of both metrics. We explored the number of actual pedestrians in the video and the number of pedestrians the proposed scheme detected. For testing our proposed scheme, we relied on Nvidia 1050 Ti GPU and a variety of videos from both Kaggle and YouTube to test our proposed scheme's effectiveness. We used nearly 3,00,000 frames which we extracted from the above-mentioned video sets. With these many frames, we got a clear idea of how effective the model is under various circumstances. In some videos where the quality was a bit low, we observed that the proposed system classified one or two objects as pedestrians as they resembled like that. The experimental observations with the series of video samples are presented in *Table II*.

As discussed earlier, the proposed scheme consisted of several sub-portions such as a dehazing, denoising and YOLOv4 model for detecting pedestrians. The task of the dehazing model was to collect a frame from the given foggy or low-visibility video and then enhance the frame to increase the visibility rate. Then the frame was further smoothed and enhanced by denoising it. The model was

TABLE II. Key observations with several input videos.

Video	Pedestrians (Actual)	Pedestrians (Detected)	Precision (P)	Recall (R)	F1-Score
1	15	15	1.00	1.00	1.00
2	8	8	1.00	1.00	1.00
3	23	24	0.98	1.00	0.99
4	18	18	0.98	1.00	0.99
5	12	12	0.98	1.00	0.99
6	25	27	0.98	1.00	0.99
7	9	9	0.98	1.00	0.99
8	15	15	0.98	1.00	0.99
9	13	13	0.98	1.00	0.99
10	8	8	0.98	1.00	0.99

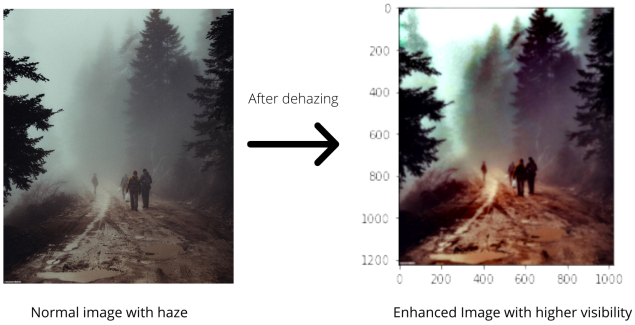


Figure 7. Converting standard image with low visibility to an enhanced one with greater visibility

Figure 8. Plotting pedestrians on the frame using OpenCV

successfully operating and gave us the desired results, as shown in *Figure 7*. Further, the YOLOv4 worked as required, it took the enhanced image as input, and identified the pedestrians in the frame, as seen in *Figure 8*. Overall for autonomous vehicles, the output is processed as a signal. The signal is high when a pedestrian is detected and low when the road doesn't have pedestrians. This signal is then sent to the microcontroller, which controls the linear actuator to move the brake pedal in order to apply brakes or lower the speed of the vehicle.

From the observations, we came across several sample test cases through which we understood that under intricate scenarios, the proposed scheme also shows higher reliability. Out of the 10 series, we only found 2 sets where there was a slight disturbance in the output, which can be overcome by providing high-quality input videos. We compared and analyzed our proposed scheme with the existing ones by using the hardware configurations, and we figured out that our proposed system is reliable and relatively fast. We also applied image denoising and dehazing techniques to other methods to evaluate their performances. One of the main reasons why scaled YOLOv4 is more time-efficient than Fast/Faster RCNN is due to its single-shot detection architecture. In contrast, Fast/Faster RCNN require a two-stage process of region proposal and object detection, which significantly increases computation time. In addition, scaled YOLOv4 uses a more efficient backbone network, called CSPDarknet, which is designed to reduce memory usage and computation time while maintaining high accuracy. The use of CSPDarknet allows for faster training and inference times compared to the backbone networks used in Fast/Faster RCNN. Furthermore, scaled YOLOv4 uses anchor-based object detection, which reduces the number of proposals that need to be processed during object detection. This, in turn, reduces computation time and increases efficiency.

Overall, the combination of single-shot detection architecture, efficient backbone network, and anchor-based object detection make scaled YOLOv4 significantly more time-efficient than Fast/Faster RCNN. In terms of speed, the proposed plan is much more efficient and more accurate. The bar plot shown in *Figure 9* gives a clear picture of the comparison of our proposed scheme with the other ones.

The proposed scheme can work more effectively if the cameras can capture more frames per second considerably 60 frames per second is well-suited. Along with this, we can reduce the decision-making process or the computation time by using high-standard systems powered by some notable GPUs like Nvidia GeForce or Tesla V100.

5. Conclusion and Future Work

The paper's experimental outcomes and discussions reveal that the suggested method has a greater classification accuracy and is significantly more streamlined than pre-

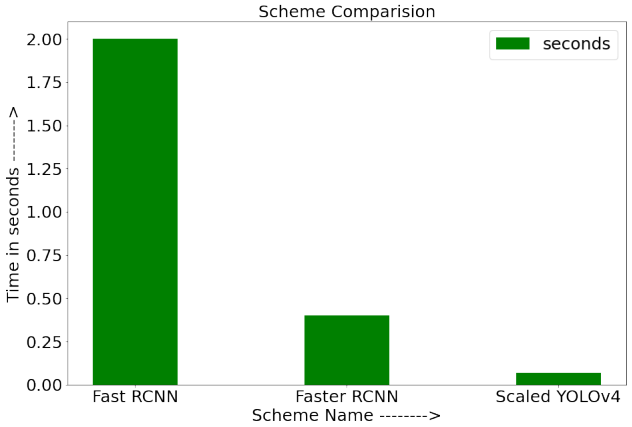


Figure 9. Comparison of the proposed scheme with other existing schemes

vious techniques. On the contrary, it justified identifying pedestrians under adverse conditions. Using the proposed scheme to implement autonomous vehicles is quite feasible and straightforward. Autonomous cars, the future of automobile industries, can use the proposed system to safeguard vehicles and make proper judgments. Considering the several results and the overall work, we concluded that the proposed scheme is reliable and relatively efficient. The proposed scheme can further be equipped with other algorithms or techniques to apply to several domains. In the case of disaster management, the proposed system can easily classify humans stuck in the affected areas. Further, in the future, we are looking forward to extending the proposed scheme to further domains like disaster management, security surveillance, and workspace monitoring.

Acknowledgment

The authors of this paper thank all the kind support received from the parent institute, SRM University-AP, Andhra Pradesh, India.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[2] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar, "The impact of adverse weather conditions on autonomous vehicles: how rain, snow, fog, and hail affect the performance of a self-driving car," *IEEE vehicular technology magazine*, vol. 14, no. 2, pp. 103–111, 2019.

[3] L. Leontaris, N. Dimitriou, D. Ioannidis, K. Votis, D. Tzovaras, and E. Papageorgiou, "An autonomous illumination system for vehicle documentation based on deep reinforcement learning," *IEEE Access*, vol. 9, pp. 75 336–75 348, 2021.

[4] Q. Liu, X. Li, S. Yuan, and Z. Li, "Decision-making technology for autonomous vehicles: Learning-based methods, applications and future outlook," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 30–37.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[6] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, pp. 91–99, 2015.

[8] P. Dong and W. Wang, "Better region proposals for pedestrian detection with r-cnn," in *2016 Visual Communications and Image Processing (VCIP)*. IEEE, 2016, pp. 1–4.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[10] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[11] K. Govil and M. Günaydin, "Deformed twistors and higher spin conformal (super-) algebras in four dimensions," *Journal of High Energy Physics*, vol. 2015, no. 3, pp. 1–41, 2015.

[12] G. Meng, Y. Wang, J. Duan, S. Xiang, and C. Pan, "Efficient image dehazing with boundary constraint and contextual regularization," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 617–624.

[13] R. Rajni and A. Anutam, "Image denoising techniques-an overview," *International Journal of Computer Applications*, vol. 86, no. 16, pp. 13–17, 2014.

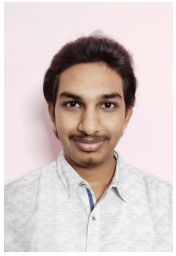
[14] D. L. Donoho and J. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *biometrika*, vol. 81, no. 3, pp. 425–455, 1994.

[15] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the american statistical association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[16] A. Hbaieb, J. Rezgui, and L. Chaari, "Pedestrian detection for autonomous driving within cooperative communication system," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2019, pp. 1–6.

Mohit Kumar Mohit Kumar is currently pursuing his B.Tech in Computer Science and Engineering from SRM University AP, Andhra Pradesh, India. He has an excellent knowledge of full-stack development and data science. He is fond of emerging technologies and tries to innovate and solve existing problems. He has developed several projects and has won various hackathons and contests. He has also secured a gold medal at the research day organized by SRM University AP. He also represented his college in the ACM ICPC Regionals in the year 2020. His constant interests in the field of Machine Learning and autonomous vehicles led him to research several existing problems with autonomous cars.

Gurram Sahithi Priya Sahithi Priya Gurram is currently pursuing her B.Tech in Computer Science and Engineering from SRM University AP, Andhra Pradesh, India. She is keenly interested in the field of Machine Learning and Big Data. She acquired a lot of knowledge in the field of Big Data and Analytics and has worked on several projects. She is also one of the participants who represented her college in the ACM ICPC Regionals 2020. She has worked with computer vision and image processing tools and is quite experienced in building surveillance and object detection systems.



Praneeth Gadipudi Praneeth Gadipudi is currently pursuing a bachelor's degree in computer science engineering, at Sri Ramaswamy Memorial (SRM) University, Andhra Pradesh, India. His research interests include data science, machine learning, and computer vision.

V. M. Manikandan Dr. V. M. Manikandan is currently working as an Asst. Professor in Computer Science and Engineering Department at SRM University-AP, Andhra Pradesh, India. Before that, he was associated with the Indian Institute of Information Technology (IIIT) Kottayam, Kerala. He did his M.Tech in Software Engineering from Cochin University of Science and Engineering (CUSAT), Kerala in 2010 and his Ph.D. in Computer Science and Engineering from the Indian Institute of Information Technology Design and Manufacturing (IIITDM) Kancheepuram, Chennai, Tamilnadu in 2018. He has more than 11 years of experience in teaching and research and published more than 65 research articles having international repute. Dr. Manikandan V. M. is a lifetime member of the Institution of Engineers (India) and a member of IEEE. His recent research work is focused on reversible data hiding, digital image forgery detection, and content-based image retrieval.