RESEARCH ARTICLE

RSP Science Hub

International Research Journal on Advanced Science Hub
2582-4376
www.rspsciencehub.com
Vol. 07, Issue 03 March

Check for updates

http://dx.doi.org/10.47392/IRJASH.2025.019

Enhancing Deep Learning to Improve Road Safety: An Accident Detection System

Naveenkumar N[1], Mahapriya S[2], Dhevathai P[3], Aswathi V[4], Dhivyadharshini R[5]

[1]Associate professor, Department of Computer Science and Engineering, Muthayammal Engineering College, Rasipuram, India.

[2,3,4,5]Student, Department of Computer Science and Engineering, Muthayammal Engineering College Rasipuram, India.

Emails: *nnaveenkumar.dr@gmail.com[1], mahapriyasenthilkumar1101@gmail.com[2], dhevathaimec@gmail.com[3], aswathi9944@gmail.com[4], rajadhivya119@gmail.com[5]*

Article history

Received: 05 February 2025
Accepted: 17 February 2025
Published: 15 March 2025

Keywords:

Convolutional Neural Networks (CNNs), You Only Look Once (YOLO), RGB Frames, Real-Time Notification, Email Alerts

Abstract

This advanced accident detection system significantly improves road safety by enabling timely identification of traffic incidents and rapid emergency response. Current road safety systems often fail to provide accurate and prompt detection of accidents, resulting in delayed emergency services and increased casualties. To address this issue, an advanced system utilizes Convolutional Neural Networks (CNNs) and the You Only Look Once (YOLO) model for real-time accident detection and classification. By analyzing RGB frames and incorporating optical flow data, this system effectively handles dynamic scenarios such as high-speed traffic, varying weather conditions, and complex vehicle movements. Furthermore, a Real-Time Notification feature integrates to automatically alert nearby hospitals and dispatch ambulances immediately upon detecting an accident. Training and testing use the dataset from Kaggle, which includes CCTV footage frames of accidents and non-accidents, ensuring robust performance. This system detects accidents in real- time and promptly sends notifications via email to alert nearby hospitals and dispatch ambulances.

1. Introduction

Traffic accidents become the number one cause of death in the world. It is estimated that there are more than half a million deaths every year from traffic accidents [1] [3] [5]. This is the start of a very practical accident-detection system because it employs advanced deep learning technologies, including CNN and YOLO, to scan data from moving objects. It can monitor accidents quickly and effortlessly in conditions that may be difficult for the human eye to spot, like heavy traffic or bad weather. It sends alerts to emergency services at any given time, thus enabling the response time for saving lives to be within reasonable limits. The system gives real- time data from the areas, and with such data, city planners know high-risk areas and begin to work on better safety measures [7] [8] [10] [12]. This technology is incorporated into personal vehicles, public transport, and smart city infrastructures to create a connected network of safety. It supports autonomous vehicles and ensures safe navigation during critical situations. Traffic accidents are one of the biggest hazards globally and

claim thousands of lives and injuries annually [14] [15] [16]. It is the objective of this study to address this global concern through new accident detection through deep learning technologies such as CNNs and YOLO. Analyzing motion data, the system would rapidly and accurately be able to establish a close match with conditions such as heavy traffic or adverse weather. It detects accidents and sends alarm alerts for emergency services. It conveys these directly for quicker time in saving the life, captures real-time data about the accident, and proves to be very valuable for city planners and officials to decide dangerous areas of that region. It can be installed in personal cars, buses, and smart city infrastructures as a connected safety network. It can also be quite compatible with the self-driving car technology for safe navigation during critical situations. Overall, the impacts of this project are spread from the immediate relief of safety to a better cause, besides the redesigning of roads and the management of traffic flow with continuous data for better decision-making (Figure 1).

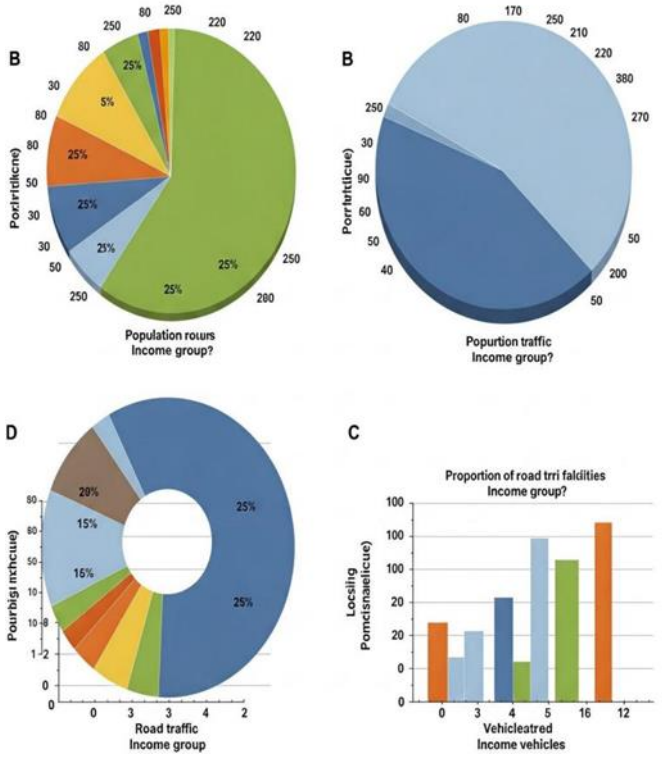

Figure 1 Comparative analysis of Population and Traffic

It can adapt and change depending on the shifting conditions and different types of surroundings of traffic. It displays a great advance in the manufacturing of transportation being made easier and safer for everybody [17] [18] [20]. In the whole, this deep learning accident detection system is a huge stride in terms of improvement in road safety and transportation [4] [9] [13]. The following figure comprises four graphs showing the correlation between income groups, population distribution, road traffic, and deaths. The top left pie chart shows the percentage of various income groups in population routes, with slices marked by percentage. The upper right pie chart shows population traffic distribution in a variety of different blues, with numbers indicating population sizes. The bottom left donut chart illustrates road traffic by income segments, and percentage breakdown distinguishes various segments. Finally, the bottom right bar chart illustrates the percentage of road trip fatalities by income segments and vehicle numbers with colored bars to distinguish categories. The charts together illustrate road usage and accident rate variation between income groups "Population" and "Loss."

2. Literature Work

Accident prediction models that use visual attention and the focus of expansion (FOE) perform well even with minimal motion but rely on high-quality datasets and require significant computational resources [11]. The TempoLearn Network, which combines CNNs and RNNs/LSTMs, can predict accidents in real-time but depends heavily on extensive annotated data [7]. Similarly, improved Mask R-CNNs, enhanced with optimized ResNet and feature pyramid networks (FPN) through transfer learning, achieve high accuracy in vehicle damage detection [19]. Real-time frameworks like YOLOv4, paired with Kalman filters, analyze object trajectories to detect accidents under various conditions [4]. Systems based on CNNs and GRUs are capable of processing CCTV footage to identify accidents and provide timely alerts to emergency services [2][6]. Other deep learning methods focus on detecting trajectory anomalies to identify accidents, thereby contributing to road safety improvements [15]. IoT-based solutions also play a key role in enhancing safety by monitoring vehicle parameters and automatically sending alerts during accidents. These systems promote safer driving and enable quicker emergency responses [2]. In India, a system using VGG16 achieves a 95% success rate in identifying accidents from CCTV footage, ensuring faster emergency intervention [14]. Additionally, a deep learning-based traffic monitoring system, which detects accidents and

burning vehicles with over 90% accuracy, improves road safety and traffic management [1].

3. Proposed Work

The architecture for the real-time traffic accident detection system takes advantage of deep learning as well as real time data processing in order to rapidly detect actions or instances with the action worded (since this should not be judge by human beings). Starting from data input — real time video stream collected from vehicle mounted cameras for acquiring RGB high resolution frames and optical flow data to capture the spatial to dynamic changes in the scene. In the preprocessing module, data cleaning and normalization as well as data augmentation will be done to improve the over-all quality of the dataset. Here we use the convolutional neural networks (CNN), in which the feature extractors are spatial part and optical flow analysis and temporal feature. β. The accident detection module process uses YOLO model (You Only Look Once) for faster detection speed and Kalman filter / hungarian algorithm for trajectory analysis in comparison to identify the colliding objects (Figure 2). Alert the hospitals and emergency services in the vicinity and cooperates with ambulance services for quick response.

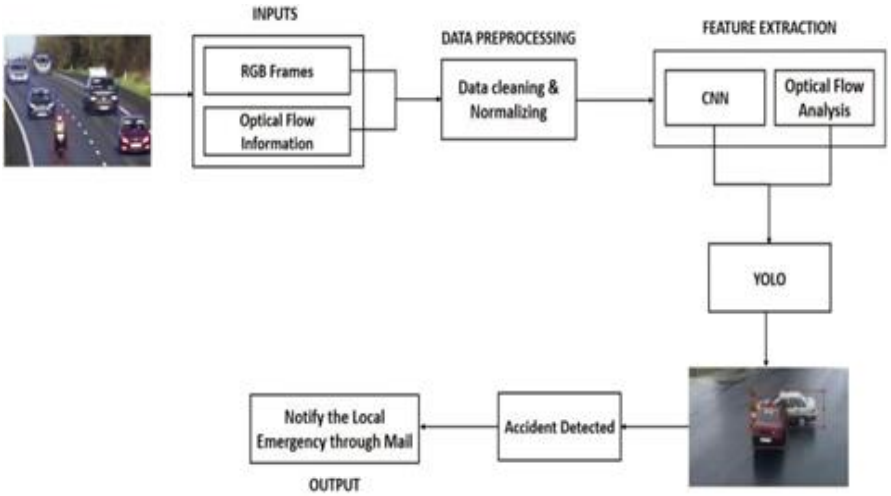

Figure 2 Work Flow Diagram of Accident Detection System

There are two essential types of input data that this accident detection system requires:

- **RGB Frames:** It is just image captures from a camera. In them, is all the details of the scene, including colours and shapes of the objects, how they are structured in space, etc. Consider the example for the colours and shapes of all the objects appearing in the shot, including automobiles, pedestrians, road markings, and so forth. These include all the visual information by which the system will decide on what is being observed in the scenario. This incorporates optical flow information that is otherwise inaccessible with static images. While a static image gives information regarding position and appearance, this information now also needs to comment upon changes in position and appearance over time. This is very useful for the dynamics of the scene. It is capable of detecting sudden changes in motion, like when a car brakes hard or a pedestrian gets in the path. This information is crucial for hazard detection and prediction of accident potential. Through the integration of the above two input sources, the system provides a statically and dynamically holistic understanding of the scene. An overall multi-modal approach has been proven to improve the accuracy and reliability of accident detection.

3.1. Data Pre-Processing

Before the accident detection system can effectively analyse the incoming RGB frames and optical flow data, it undergoes a crucial pre-processing step. This stage is essential for improving the quality and consistency of the data, ensuring that subsequent analysis stages produce accurate and reliable results

- **Noise Removal:** The system removes such unwanted information in the form of sensor

errors and environmental interference, like the rain or fogs, thereby degrading data quality Image smoothing filters are used to make the images smooth, and very complex algorithms are used to find noise patterns.

- **Handling Missing Values:** In the majority of real-world datasets, there are missing values because of various reasons like sensor malfunctions or network breakdowns. The system will use interpolation techniques to approximate missing data points from surrounding information so that it ensures continuity and completeness of the data set for further analysis.
- **Data Scaling:** The technique scales the data into one range so that features with large values do not skew the analysis. Normalization ensures that every feature has a similar impact on the accuracy of successive stages of analysis. By tracking motion of objects over time, the system can analyze the motion trajectories of the objects, identifying unusual or abnormal motion patterns such as unexpected stops, random motions, or abnormal direction changes. These anomalies would be excellent predictors of future accidents.

3.2. Feature Extraction

This is a necessary process because it allows the system to focus on the most important information and improve the efficiency and accuracy of later stages of analysis. Two dominant drivers drive feature extraction in this accident detection system:

3.2.1.CNN (Convolutional Neural Network) Analysis of RGB Frames

Convolutional Neural Networks (CNNs) are a robust family of deep learning models that are well suited for image processing. CNNs are very good at object detection, edge detection, texture detection, and pattern detection in images. A hierarchy of convolutional, pooling, and fully linked layers is used by CNNs to automatically scan the RGB frames for hierarchical characteristics. Starting with low-level features such as edges and colors, CNNs learn higher-level and abstract features, i.e., object parts and full objects.

3.2.2.Optical Flow Analysis

Optical flow analysis focuses on the motion of objects within the scene. By analyzing object motion and changes in position over time, the system can gain a better insight into the dynamics of the scene [16].

3.3. YOLO (You Only Look Once)

These extracted features, from the CNN processing of RGB frames as well as from optical flow processing, are now fed to the YOLO object detection model. YOLO is considered one of the most efficient deep learning models ideally suited for the detection and locating of many objects within an image at one go. In the accident detection system application, it will be used to train a YOLO model to recognize all different sorts of objects and events highly linked with accidents. This focused training makes the system able to efficiently detect and react to potential hazard; In a practical sense, it improves the accident detection system.

4. System Architecture

First, source will be taken as Kaggle in which images and videos are used regarding accidents (Figure 3). Then that obtained dataset from there was annotated by tagging every object which belongs to an accident-detecting vehicle, pedestrian, road sign, etc.

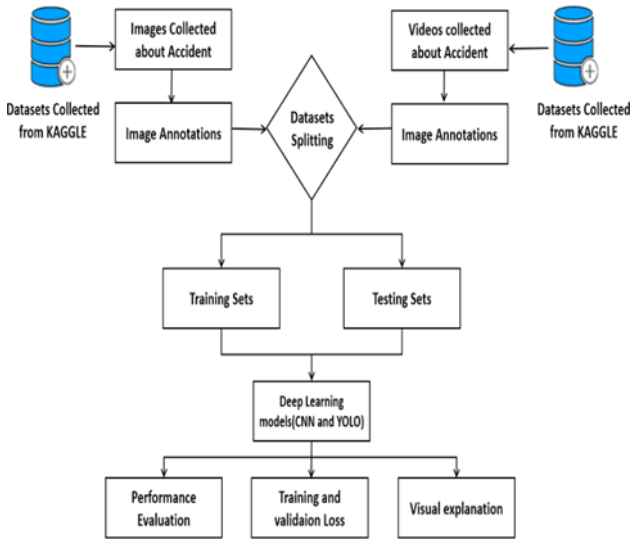

Figure 3 Architecture of Training and Testing of Dataset

It divides into two subsets of datasets: one for training, the other one for testing purposes. CNNs and YOLO are deep learning models that use the training sets. Testing sets are applied for the performance evaluation by the model. Further, it gives an idea about how the model has decided on those predictions and what those predictions are that the component provides to the model. It

explains how the accident detection system is created, starting from gathering data, annotating, training the model, and the evaluation of its performance.

5. Methodology

5.1. YOLO Algorithm

"You Only Look Once" is what the YOLO algorithm stands for (Figure 4). This well-liked object detection technique is renowned for its accuracy and quickness. After dividing the input image into a grid, YOLO forecasts probability and bounding boxes for every grid cell. As a result, instead of repeatedly scanning an image, it can identify many things with a single glance. Security systems and other real-time applications make extensive use of it.

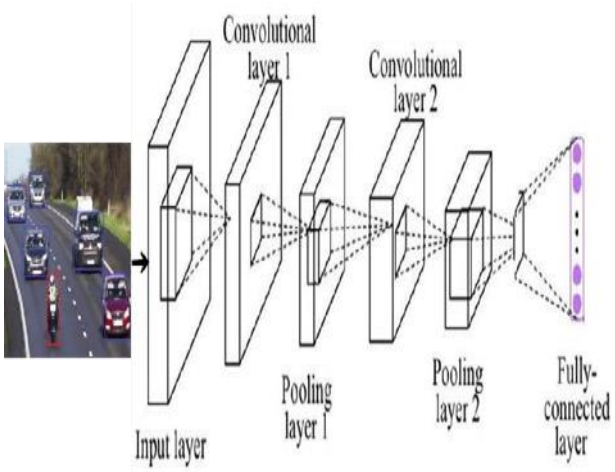

Figure 4 YOLO Architecture

Image Resizing: This is a geometric operation where the image dimensions are scaled up or down. While there might be interpolation techniques involved (like bilinear interpolation), these are typically implemented in libraries and don't require manual formula application.

Image Normalization: This is a simple arithmetic operation where each pixel value is divided by a constant (often the maximum pixel value) to bring it within a specific range. For example, you could apply the following formula to normalise pixel values to the range of 0 to 1:

$$= \frac{\text{Pixel value}}{\text{Maximum pixel value}}$$

5.1.1. Convolution Operation

The convolution operation is the fundamental building block of convolutional neural networks. It entails summing the outcomes after applying an element-wise multiplication filter (kernel) on the input image.

$$S(i,j) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} W(m,n)\cdot I(i+m,j+n) + b$$

Where,

- S(i, j): Output of the convolution at position (i, j)
- W(m, n): Weight of the filter at position (m, n)
- I(i+m, j+n): Input pixel value at position (i+m, j+n)
- b: Bias term
- M and N: Height and width of the filter, respectively

5.1.2.Stride Convolution

In some cases, the convolution operation is performed with a stride ss, which determines the step size with which the filter is moved across the input. The output size of the feature map is reduced when the stride is greater than 1. The formula for strided convolution is:

$$S(i,j) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} W(m,n)\cdot I(s\cdot i+m, s\cdot j+n) + b$$

Where,

- *s*: Stride value

5.1.3.Pooling Layers

Pooling layers operate on small regions of the input feature map and produce a single output value for each region. The two most common types of pooling are Max Pooling and Average Pooling.

1. **Max Pooling**

Max pooling selects the maximum value from the region of the feature map covered by the filter. Mathematically, for a pooling region of size (M×N) at position (i,j):

$$P(i,j) = \max_{m=0}^{M-1}\max_{n=0}^{N-1} I(i\cdot s+m, j\cdot s+n)$$

Where,

- **P(i, j):** Output of the pooling operation at position (i, j)
- **I(i·s+m,j·s+n):** Input feature map value at position (i·s+m,j·s+n)
- **s:** Stride of the pooling operation

- **M and N:** Height and width of the pooling region

2. Average Pooling

Average pooling computes the average value of the region of the feature map covered by the filter. Mathematically:

$$P(i,j) = \frac{1}{M \cdot N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I(i \cdot s + m, j \cdot s + n)$$

Where,

- **P(i,j):** Output of the pooling operation at position (i,j).
- **I(i·s+m,j·s+n):** Input feature map value at position (i·s+m,j·s+n)
- **s:** Stride of the pooling operation
- **M and N:** Height and width of the pooling region

5.1.4.Detection Layer

By using YOLO, the input image is split into a S x S grid. Predicting objects whose centres lie inside its borders is the responsibility of each grid cell.

1. Bounding Box Prediction

Four values are predicted by the model for every grid cell:

- **bx and by:** These are the x and y coordinates of the anticipated bounding box's centre in relation to the grid cell's upper-left corner.
- **bw and bh:** These represent the width and height of the predicted bounding box.

To ensure that the predicted bounding box coordinates are within the range of 0 to 1, YOLO employs the following mathematical operation:

$$bx = \sigma(tx) + cx$$

$$by = \sigma(ty) + cy$$

$$bw = \exp(tw) * pw$$

$$bh = \exp(th) * ph$$

Where,

- σ is the sigmoid function, which maps values to the range of 0 to 1
- tx, ty, tw, th are predicted offsets learned by the network
- cx, cy, pw, ph are parameters of the anchor box associated with the grid cell

2. Class Prediction

The class probabilities are typically computed using a softmax function:

$$P(c_i) = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}}$$

Where,

- **zi:** Raw logits (scores) for class i
- **C:** Total number of classes

3. Objectless Score

The objectless score ss is often predicted using a sigmoid function:

$$s = \sigma(t_o)$$

Where,

- **to:** Raw prediction for the objectness score
- **σ:** Sigmoid function, ensuring s∈[0,1]

5.1.5.Non-Maximum Suppression

In object detection, Non-Maximum Suppression (NMS) is a strategy that removes duplicate bounding boxes that could be anticipated for the same object. It ensures that each object is detected only once, improving the accuracy and efficiency of the detection process.

1. Intersection over Union (IoU)

We utilise the Intersection over Union (IoU) metric to find the overlap between two bounding boxes. It is determined by dividing the area of the two boxes' junction by the size of their union:

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$

5.1.6.Post-Processing and Accident Detection

1. Correlation-Based Tracking

While there isn't a single formula, the process involves calculating similarity metrics like the Sum of Squared Differences (SSD) or Normalized Cross- Correlation (NCC) between image patches. Simple arithmetic techniques are used to calculate these measures.

2. Normalized Cross-Correlation (NCC)

The most common method for correlation-based tracking is **Normalized Cross-Correlation (NCC)**. NCC measures the similarity between the template T and a region in the search area S. The formula for NCC is:

$$C(x,y) = \frac{\sum_{i,j}(T(i,j) - \mu_T) \cdot (S(x+i,y+j) - \mu_S(x,y))}{\sqrt{\sum_{i,j}(T(i,j) - \mu_T)^2 \cdot \sum_{i,j}(S(x+i,y+j) - \mu_S(x,y))^2}}$$

Where,

- **C(x,y):** Correlation value at position (x,y) in the search region
- **T(i,j):** Pixel value at position (i,j) in the template
- **S(x+i,y+j):** Pixel value at position (x+i,y+j) in the search region
- **μT:** Mean pixel value of the template
- **μS(x,y):** Mean pixel value of the region in the search area centered at (x,y)

The NCC value ranges from -1to 1, where:

- 1 indicates a perfect match
- -1 indicates a perfect inverse match
- 0 indicates no correlation

3. Sum of Squared Differences (SSD)

Another common method is the **Sum of Squared Differences (SSD)**, which measures the dissimilarity between the template and the search region. The formula for SSD is:

$$C(x,y) = \sum_{i,j}(T(i,j) - S(x+i,y+j))^2$$

Where:

- **C(x,y):** Dissimilarity value at position (x,y) in the search region
- **T(i,j)T(i,j):** Pixel value at position (i,j) in the template
- **S(x+i,y+j):** Pixel value at position (x+i,y+j) in the search region

6. Result

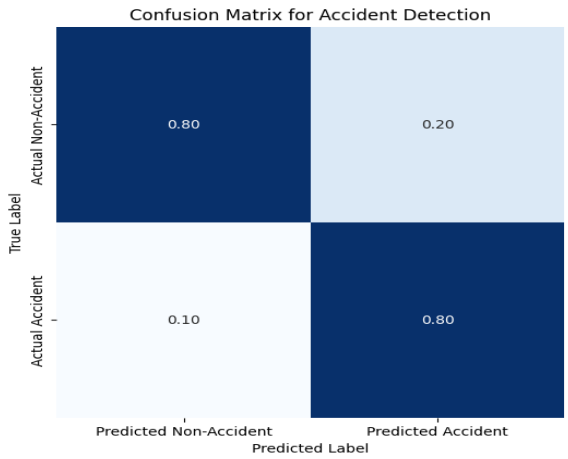

Figure 5 Confusion Matrix

This project developed an advanced accident detection system using deep learning techniques, such as CNNs and the YOLO model, to enhance road safety (Figure 5). The system achieved significant improvements in accuracy and real-time detection of traffic incidents, providing timely alerts to emergency services. By addressing challenges like computational demands and adaptability, the system demonstrated robust performance, offering a practical solution for improving traffic monitoring and reducing accident response times. The figure 4 displays a confusion matrix, a key tool for evaluating the performance of a binary classification model with classes "0" and "1." It shows correct and incorrect predictions, with "True Label" rows representing the true classes and "Predicted Label" columns showing model predictions. The diagonal cells represent accurate predictions: 80% of actual class 0 were predicted as class 0, and 80% of actual class 1 were predicted as class 1. Misclassifications are indicated by the off-diagonal cells: 10% of actual class 1 were expected to be class 0, while 20% of actual class 0 were predicted to be class 1. The model performs well overall, but it struggles with misclassifying class 1 as class 0. This suggests potential areas for improvement, such as refining features or adjusting decision thresholds to reduce this type of error. The confusion matrix helps with continued development and improvement by offering a thorough analysis of the model's advantages and disadvantages. The Figure - 5 is a correlogram or correlation matrix plot, visualizing the relationships between multiple variables. It's a grid of plots, where each cell represents the relationship between two variables.

- **Diagonal:** The diagonal cells show the distribution of each individual variable using histograms. In this case, the variables are 'x', 'y', 'width', and 'height'.
- **Off-Diagonal:** The off-diagonal cells display scatter plots or heatmaps that illustrate the relationship between two variables. The degree and direction (positive or negative) of the association are shown by the color intensity in each cell (Figure 6).

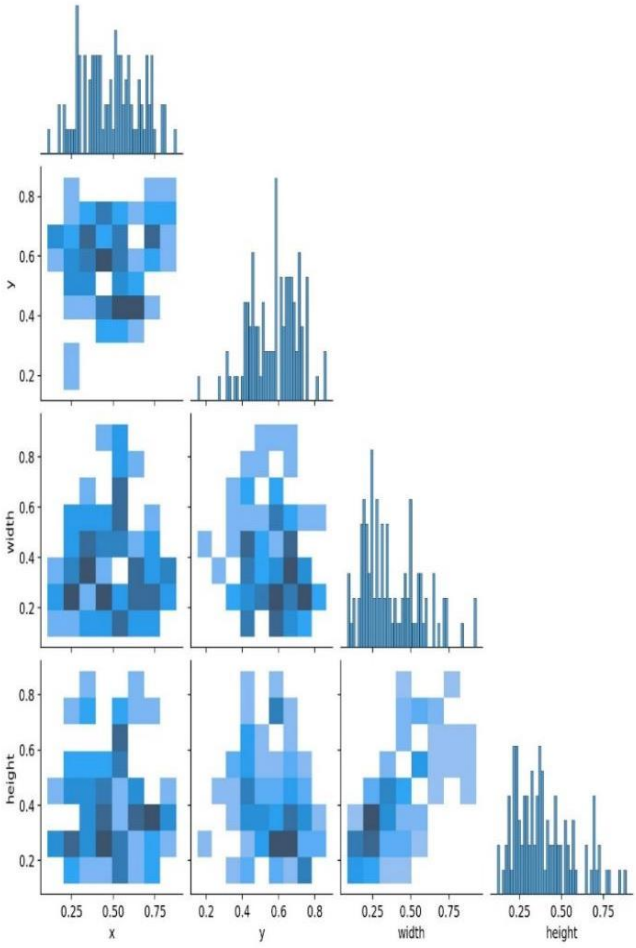

Figure 6 Scatter Plot

Variables: The plot analyzes four variables:

- **x:** Likely represents a horizontal position or dimension.
- **y:** Likely represents a vertical position or dimension.
- **width:** Represents the width of an object or region.
- **height:** Represents the height of an object or region.

Histograms (Diagonal): The histograms along the diagonal provide a visual representation of the distribution of each variable independently. For every variable, you can see the frequency of various values.

Scatter Plots/Heatmaps (Off-Diagonal): The off-diagonal plots reveal the relationships between pairs of variables:

- **x vs. y:** Shows the correlation between the x and y coordinates. The pattern suggests a somewhat clustered distribution, possibly with a positive relationship.
- **x vs. width:** Illustrates how width varies with changes in x.
- **x vs. height:** Shows the relationship between x and height.
- **y vs. width:** Depicts the correlation between y and width.
- **y vs. height:** Shows the relationship between y and height.
- **width vs. height:** Illustrates the correlation between width and height.

- **Color Scale:** The color bar on the right indicates the strength and direction of the correlation. In general, stronger associations are indicated by deeper hues, whereas weaker correlations are represented by lighter ones. The color variation might also indicate the direction of the correlation (e.g., blue for positive, red for negative). The following graph is a Precision-Confidence Curve with Precision on the Y-axis and Confidence on the X-axis (Figure 7). The curve is a blue line of many points, depicting variation in precision at different levels of confidence. One particular point, recision 0.8 at Confidence 0.8, is red-circled as a key cutpoint. The figure demonstrates how growing confidence affects precision, commonly utilized in machine learning and classification issues to measure model accuracy. Legend at the top left describes the two areas of the graph: the whole curve and the precision-confidence point marked.

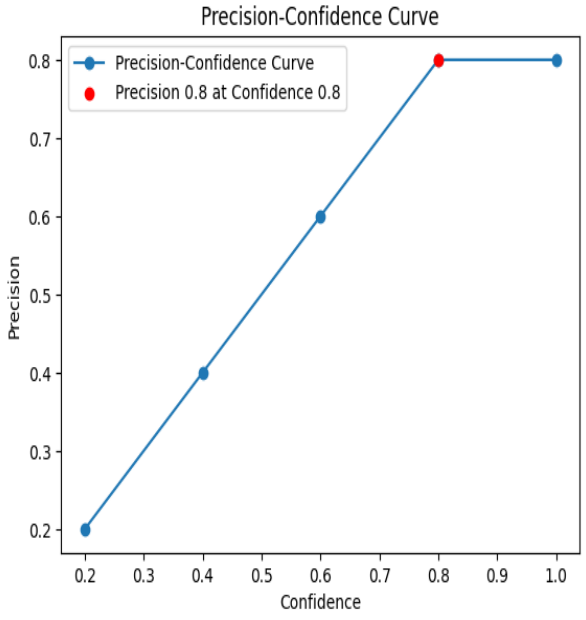

Figure 7 Precision – Confidence Curve

7. Conclusion and Future Enhancement

Automatic accident detection for traffic management systems is a valued resource that can not only prevent future accidents but also allow emergency services to open roads as early as possible. This research shows that detection of patterns in vehicle behavior with its location and speed will enable such possible dangers for drivers around to be analyzed as anomalous activities. Accidents are one of the major issues, having massive casualties and property destroyed. Solution: Vehicle based accident detection system to detect occurrence of accident in real time and send email alerts to nearby hospitals/police stations with exact Time-Stamp & Location. Accidents are taking a rapid developing — so fast that especially within cities where commuters rush this kind of system would crush the success of exposure. Supported by advanced technologies such as CCTV and Convolutional Neural Networks (CNN) it can kill for a faster detection and response.

References

[1]. Agarwal, N., Jangid, A., & Sharma, A. (2021). Camera-based Smart Traffic State Detection in India Using Deep Learning Models. Journal of Computer Science and Applications, 12(3), 156-163.

[2]. Chaudhari, R. V. A., Agrawal, R. K. H., & Poddar, S. (2020). Smart Accident Detection And Alert System. Journal of Computer Science & Technology, 35(6), 1125-1138.

[3]. Ghahremannezhad, H., Shi, H., & Liu, C. (2021). Real-Time Traffic Accident Detection Using Deep Learning. Transportation Research Part C: Emerging Technologies, 124, 102932.

[4]. Hadid Ghahremannezhad, Hang Shi, Chengjun Liu. (2022) Real-Time Accident Detection in Traffic Surveillance Using Deep Learning. arXiv preprint arXiv:2208.06461.

[5]. Htun, S. S., Park, J. S., Lee, K.-W., & Han,J.-H. (2019). TempoLearn Network: Leveraging Spatio-Temporal Learning for Traffic Accident Detection. IET Intelligent Transport Systems, 13(8), 1267-1274.

[6]. Lee, K. B., & Shin, H. S. (2023). A deep learning algorithm for automatic detection of accidents under bad CCTV monitoring conditions in tunnels.

[7]. Li, W., Wang, Z., & Zhang, Y. (2022). A deep learning-based traffic accident severity prediction model using spatiotemporal features.

[8]. Li, Y., Li, Y., & Liu, Z. (2022). A novel attention-based deep learning model for traffic accident severity prediction.

[9]. Li, X., et al. (2020). Multi-Camera-Based Accident Detection in Traffic Surveillance. IEEE Transactions on Intelligent Transportation Systems, 21(5), 2026-2036.

[10]. Luo, Y., Yang, Y., & Wang, X. (2022). A novel deep learning framework for traffic accident risk prediction

[11]. Maruyama, Y., & Ohashi, G. (2021). Accident Prediction Model Using Divergence Between Visual Attention and Focus of Expansion in Vehicle-Mounted Camera Images. IEEE Transactions on Intelligent Transportation Systems, 22(10), 5678-5690.

[12]. Misbah, A., & Malayil M.I., S. R. (2021). Deep Learning based Auto Accident Detection and Alert System for Vehicles. Neurocomputing, 417, 349-358.

[13]. Mudgal, M., Punj, D., & Pillai, A. (2020). A Novel Approach for Accident Detection and Localization Using Deep Learning. Computer Science Review, 39, 100358.

[14]. Robles-Serrano, S., & Sanchez-Torres, G. (2021). Automatic Detection of Traffic Accidents from Video Using Deep Learning Techniques.

[15]. Shukla, D. K., Priya, R. V. L., & Geerthik,S. (2020). Safe Road AI: Real-Time Smart Accident Detection for Multi-Angle Crash Videos using Deep Learning Techniques and Computer Vision. Journal of Transportation Engineering, Part A: Systems, 146(4), 04020026.

[16]. Victor, A., & Elsayed, N. (2023). Smart City Transportation: Deep Learning Ensemble Approach for Traffic Accident Detection.

[17]. Wang, Y., Chen, X., & Zhang, L. (2022) Real-time traffic accident detection using deep learning and computer vision techniques.

[18]. Xu, Y., Wang, Y., & Zhang, L. (2022). A deep learning-based approach for traffic

accident severity prediction.

[19]. Zhang, Q., Chang, X., & Bian, S. (2023). Vehicle damage detection Segmentation Algorithm Based on Improved Mask RCNN.

[20]. Zhang, Y., Wang, Z., & Li, W. (2022). A learning-based traffic accident severity prediction model using spatiotemporal features and attention mechanism.