KEC Conference 2025

ISSN: 3021-940X (Print) DOI: https://doi.org/10.3126/injet.v2i2.78582

Reckless Status Calculation by Using Yolo and Model Trained on RNN

Newton Shahi Thakuri[1, *], Bhuma Devi Acharya[2], Ayushma Thapa[3]

[1]*Everest Engineering College, Pokhara University, Sanepa, Lalitpur, Nepal, suton.empire123@gmail.com*
[2]*Everest Engineering College, Pokhara University, Sanepa, Lalitpur, Nepal, shilaacharya2@gmail.com*
[3]*Everest Engineering College, Pokhara University, Sanepa, Lalitpur, Nepal, thapaayushma407@gmail.com*

Abstract

Reckless driving significantly contributes to road accidents and fatalities worldwide, including in Nepal. To tackle this critical issue, we developed reckless driver detection application, an innovative system utilizing advanced machine learning techniques, data analysis to detect and analyze reckless driving behaviors by use of statistical parameters such as standard deviation, mean velocity, variance, kurtosis, skewness, peak-to-peak, comparison mean. The electron app employs the YOLO model to gather information for data analysis for accurate vehicle detection. Once vehicles are detected, Kalman filters track their movements to provide reliable data on speed and trajectory. Pandas is used for saving necessary information in .xlsx files. OpenCV is used for processing the video into its analyzed form with boundaries over vehicles and labels over it, feature extraction such as fps, width, height of the frame. The system further analyzes time-series data using recurrent neural networks trained model to generate a "Reckless Status," quantifying reckless behaviors. During evaluation, the custom RNN model achieved a test loss of 0.2550, accuracy of 94.12%, precision of 1.0, recall of 91.6%, and an F1-score of 0.8979. The ROC curve attained a score of 0.93, indicating strong classification performance. These results demonstrate the effectiveness of the model in identifying reckless driving behaviors with high precision and reliability. This comprehensive solution aims to significantly reduce road accidents and protect vulnerable road users in Nepal.

Keywords: Reckless driving, Machine learning, YOLO, Kalman filters, Recurrent neural networks, Image processing, Deep learning

1. Introduction

Reckless driving is a major contributor to road accidents and fatalities worldwide, including in Nepal. This paper presents the development of a reckless driving detection system utilizing advanced machine learning techniques and data analysis to quantify reckless behaviors. By leveraging a YOLO-based detection model and Kalman filters for tracking, the system ensures accurate vehicle detection and trajectory estimation. Furthermore, recurrent neural networks (RNNs) are used for time-series analysis to determine a "Reckless Status," indicating the likelihood of reckless driving. Reckless term is itself a term that is abstract and hard to define so we have defined it on the basis of velocity and the statistical parameters like peak-to-peak, kurtosis, etc. These thresholds for the values are obtained by heuristic-based approach and only suitable for the experimental videos captured. To make it more dynamic for all situations, further testing and research on this matter specifically will be required which can determine the thresholds for the statistical parameters. With this approach of obtaining the values for threshold, this has helped shorten the period of current research which has focused on training the model in a closed environment to obtain reckless status. The developed app which incorporates such feature is a desktop-based app developed using electron app for UI (html, CSS, JavaScript), python for its backend and Kaggle for training of the model.

2. Literature Review

According to The Himalayan Time (2024), traffic police have issued warnings regarding reckless driving behavior pointing towards the drivers causing the most accidents due to their behavioral pattern of driving. Addressing this issue requires an effective system capable of detecting and analyzing reckless driving behaviors.The system uses YOLO for its object detection due to it being fast comparatively to other tools. Yolo has the best FPS of 60-140 FPS with good enough accuracy to get by of about ~50% (Bochkovskiy, A.,

Corresponding Author

Wang, C.-Y., & Liao, H.-Y. M., 2020). Comparatively Faster R-CNN has less FPS of 7-10 FPS and same accuracy and high complexity (Ren, S., He, K., Girshick, R., & Sun, J., 2015). Whereas Mask R-CNN has FPS of 5-6 FPS and 45-50% and very high complexity (He, K., Gkioxari, G., Dollár, P., & Girshick, R., 2017) and RetinaNet has ~10-12 FPS, ~39-41% accuracy with medium complexity (Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P., 2017) whereas EfficientDet has ~20-30 FPS, ~49% accuracy, medium-high complexity (Tan, M., Pang, R., & Le, Q. V., 2020).

Table 1. Comparative Analysis of Object Detection Models

Model	Speed (FPS)	Accuracy (mAP)	Complexity	Application Domains
YOLOv4	65 FPS	~43.5% mAP	Low	Real-time detection, autonomous driving, CCTV
YOLOv5	70-140 FPS	~50% mAP	Low	Traffic surveillance, embedded systems
Faster R-CNN	7-10 FPS	~42-45% mAP	High	Medical imaging, research
Mask R-CNN	5-6 FPS	~45-50% mAP	Very High	Instance segmentation, biological tasks
RetinaNet	10-12 FPS	~39-41% mAP	Medium	Industrial inspection, robotics vision
EfficientDet	20-30 FPS	~49% mAP	Medium-High	Drone surveillance, mobile vision

Hence, YOLO came out in top among its peers due to its suitability in the context of the desired system prospect of building it as a real time detection system. Currently it's in closed environment where video is given as input and desired output is obtained.

3. Methodology

The system follows a multi-stage approach involving detection, tracking, data processing, and classification. They are:

3.1. Vehicle detection and tracking

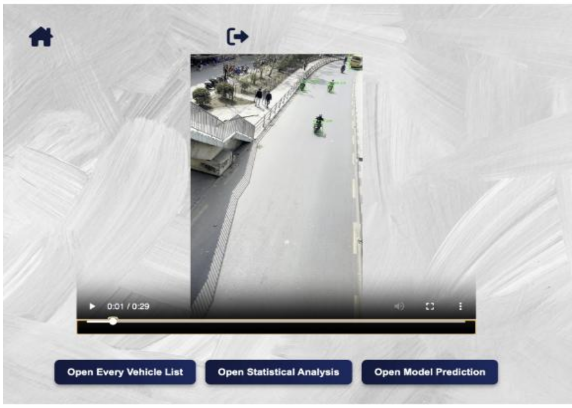

Figure 1. Vehicle Detection and Tracking of the app

A YOLO-based model is employed for detecting vehicles in video footage. This deep learning model, implemented using the sklearn library, has been fine-tuned on a diverse dataset to improve detection accuracy in real-world conditions.After detecting vehicles, the system utilizes Kalman filters to assign unique identifiers and track each vehicle across frames. The Kalman filter ensures smooth and reliable trajectory estimation even under challenging conditions, providing essential data such as velocity and direction.
Kalman Filters are employed to track detected vehicles over time, ensuring accurate predictions of future states and reliable data on speed and trajectory. They are implemented as (Mochnac, J., Marchevsky, S. and Kocan, P., 2009):

State Transition Matrix (F): Defines how the state (position and velocity) evolves over time.

$$Self.F = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(Equation 1)}$$

Process Noise Covariance (Q): Accounts for uncertainty in movement, especially at high speeds.

$$Self.Q = diag([2.0,2.0,0.5,0.5]) \quad \text{(Equation 2)}$$

Measurement Matrix (H): Maps the actual state (position and velocity) to observed position (x, y) from detection.

$$Self.H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \text{(Equation 3)}$$

Measurement Noise Covariance (R): Represents uncertainty in position measurements.

$$Self.R = diag([0,1,0,1]) \quad \text{(Equation 4)}$$

Covariance Matrix (P): *Represents initial uncertainty in tracking fast-moving objects*

$$Self.P = 200.I_4 \quad \text{(Equation 5)}$$

Self is the state of kalm which is array of x,y, velocity in x and velocity of y which datatype is float.

3.2 Velocity Calculation

The data collection was performed using the very Reckless Detection App, which generates a spreadsheet containing essential vehicle information such as velocity and relative velocity. This data is further analyzed to calculate (Lin, T.-Y., Goyal, P., Girshick, R., He, K. and Dollár, P., 2017):

- **Horizontal Scaling Factor:**

$$horizontal\_scale = \frac{horizontal\_road\_length}{\max\_velocity} \quad \text{(Equation 6)}$$

- **Scaled Velocity Calculation by camera position:**

$$scaled\_velocity = \frac{velocity}{\cos\left(radian\left(camera_{angle}\right)\right) * camera_{height}} \quad \text{(Equation 7)}$$

- **Velocity Scaling:**

$$df[\text{'}Velocity\text{'}] = df[\text{'}Velocity\text{'}] * horizontal\_scale \quad \text{(Equation 8)}$$

- **Unit conversion into km/hr:**

$$df[\text{'}Velocity\text{'}] = df[\text{'}Velocity\text{'}] *3.6 \quad \text{(Equation 9)}$$

The order is first velocity is calculated by kalman filter using its state then it undergoes scaling by use of camera position after which it is multiplied by horizontal scaling factor then converted in km/hr (kilometer per hour). The function df represents the column when using pandas in python.

3.3. Data collection and processing

The data collection was performed using the very Reckless Detection App, which generates a spreadsheet containing essential vehicle information such as velocity and relative velocity. This data is further analyzed to calculate statistical parameters necessary for reckless driving assessment. However, challenges such as noise in data, vehicle ID changes due to motion blur or low-quality frames, and incorrect multiple IDs for the same vehicle were encountered. To mitigate these issues, any vehicle with less than 2 seconds of tracked data was removed from the dataset. Data collection was based on analyzing pixel movement per frame. The video data was manually recorded on roads in Lalitpur using overhead bridge cameras. The dataset consists of approximately one hour of video footage, covering around 800 vehicles, with corresponding statistical data stored in Excel. Statistical analysis is performed using Pandas, where parameters such as standard deviation, mean velocity, variance, kurtosis, skewness, peak-to-peak, comparison mean are recorded in .xlsx files.

3.4. Machine learning algorithm

Machine learning algorithm has been applied by leveraging kaggle platform which used python and its various libraries like tensorflow, etc. for effective training and visualizing the evaluations. RNN has been used to train the machine. First classic RNN has been used on one independent variable that is to be determined (Reckless status) and 7 other dependent variables (statistical parameters) which helps obtain the independent variable. This gives a value between 0-1 for reckless status which is then rounded off to make it a binary classification of either reckless or not reckless. This could be tweaked for better and more classification to be slightly reckless and other categories to be added but currently we are observing its effectiveness only. The data and value RNN is trained on is the dataset we collected from YOLO and a video of about 1.5 hours capturing more than 500 vehicles in lalitpur area captured such that only one way of flow of vehicles is taken with limited number of vehicles and certain camera angle and height. This helped model fit the data and properly train on it with about 125 epoch which showed most efficiency and least chance for overfitting and underfitting.

3.5. Reckless status calculation

A recurrent neural network (RNN) is trained to analyze time-series data and assess reckless behavior. (Goodfellow, I., Bengio, Y. & Courville, A., 2016) The model evaluates driving patterns and assigns a "Reckless Status"—1 for reckless and 0 for safe driving. The evaluation metrics include accuracy, precision, recall, and F1-score, with an achieved test loss of 0.02550, accuracy of 94.12%, precision of 1.0, recall of 91.6%, and an F1-score of 0.8979. The ROC curve score of 0.93 further validates the model's strong classification performance.

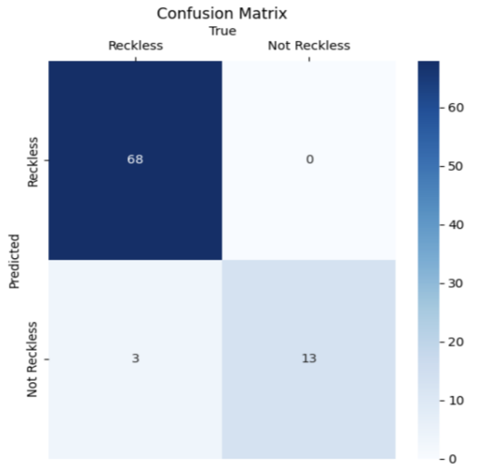

Figure 2. Model Confusion Matrix of Reckless Status

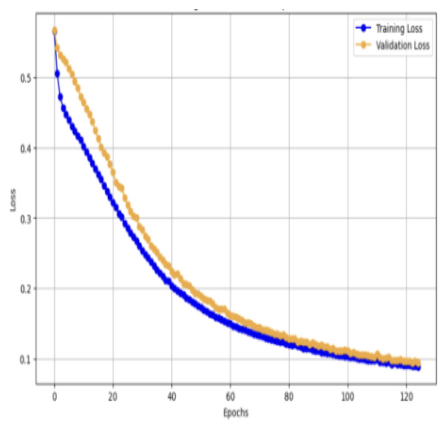

Figure 3. Training And Validation Loss Vs Epochs

4. System workflow

The system follows a structured process to analyze driving behavior from video footage. Initially, the user selects a video for analysis. Once the video is chosen, the system processes it by extracting frames and detecting vehicles using the YOLO model. Kalman filters are then employed to track vehicle movements across frames, ensuring accurate velocity and acceleration measurements. The extracted tracking data is recorded in Excel files, which include parameters such as velocity per frame, standard deviation, mean velocity, variance, kurtosis, skewness, peak-to-peak, comparison mean. This processed data is subsequently passed through a custom-trained recurrent neural network (RNN) model. The model evaluates driving patterns by analyzing speed variations to classify reckless driving tendencies. The system then displays the results, allowing users to review the extracted statistical data or select another video for further analysis. This workflow seamlessly integrates video processing, machine learning, and statistical analysis to ensure accurate detection of reckless driving behaviors. Process video represents the working of various libraries and models like YOLO, kalman filter, OpenCV, etc. YOLO identifies the vehicles to be tracked on the frame and opencv draws a rectangular bounding box around it whereas kalman filter simulatenously instantiates its states, updates it, remembers it, predicts it and in the future finds new state from which velocity is calculated and saved.

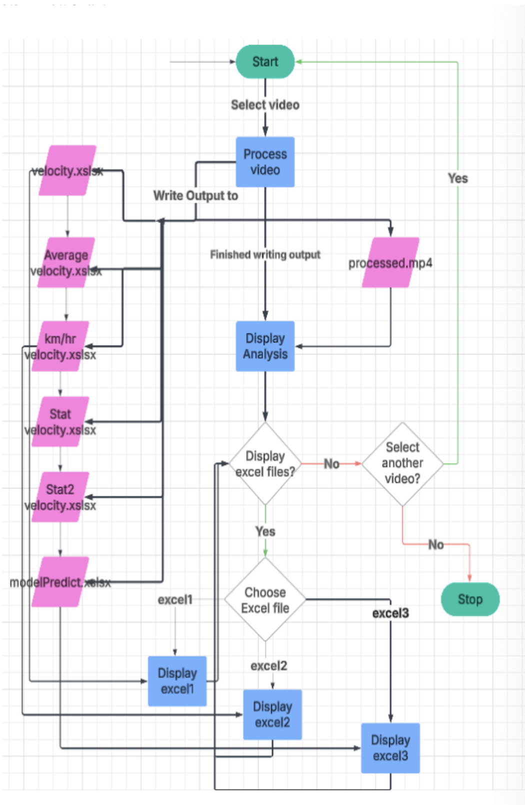

Figure 4. Flowchart of app for reckless status calculation

5. Results and discussion

The proposed system effectively detects reckless driving behaviors with high accuracy. The evaluation results indicate robust classification performance, demonstrating the potential of integrating deep learning and statistical analysis for road safety applications. By automating reckless driving detection, the system provides a scalable and efficient approach to reducing traffic accidents.

Table 2. Table of processed video data result of vehicle id 1 in excel

TimeStamp	Velocity	Relative Velocity
00:00:00	3.3368	15.1811
00:00:00	1.8167	6.5348
00:00:00	1.8167	6.5348

Output of velocity and relative velocity is calculated as seen in Table 1 by using the kalman filter. Time stamp is in format hh:mm:ss where hh is hour, mm is minute and ss is the second. It consists of velocity calculation done pixel per frame. This velocity of pixel per frame is then calculated by scaling it as camera angle.

Table 3. Table of of velocity converted in km/hr of vehicle id 1

TimeStamp	Velocity (km/hr)	Relative Velocity
00:00:00	30.276	8.90344
00:00:01	12.9254	5.33321
00:00:02	17/0836	15.0565

Output of table 1 is processed into table 2 which now has velocity in km/hr. Vehicle id now is in second calculation rather than frame calculation

Table 4. Table of of velocity converted in km/hr of vehicle id 1

Track ID	Reckless Status	Total Flag	Standard Deviation	Flag Standard Deviation	Mean Velocity	Flag Mean Velocity	Variance	Flag Variance
Track ID 35	Yes	4	17.3287	1	20.166	0	299.93	1
Track ID 39	Yes	4	15.8417	1	12.15	0	250.96	1
Track ID 4	No	3	13.2674	0	13.7812	0	176.026	1

Skewness	Flag Skewness	Kurtosis	Flag Kurtosis	Peak-to Peak	Flag Peak-to-Peak	Comparison means	Flag Comparison mean
-0.7069	0	-1.5	1	30.1069	0	19.46	1
0.6485	0	-1.5	1	29.3410	0	11.49	1
0.1038	0	-1.58024	1	29.9276	0	12.82	1

Output of table 2 is then used for calculation of table 3 which consists of many statistical parameters like standard deviation, mean velocity, variance, skewness, kurtosis, peak-to-peak and comparison mean

Table 5. Table predictions by custom model (RNN) in excel

Track ID	Predictions
Track ID 35	1
Track ID 39	1
Track ID 4	0

Outcome of table 4 represents the predictions of reckless status where 1 means the track id (vehicle) is reckless and 0 means it is not reckless. This is done by the model trained under RNN with 64 sets and 125 epochs. Skewness measures the asymmetry of the velocity distribution. In the context of reckless driving, extreme skewness can signal sudden acceleration or deceleration behavior, which may reflect aggressive maneuvers. Kurtosis measures the tailedness or peakedness of the distribution. Reckless drivers often show high kurtosis, as their velocity fluctuates more dramatically than normal driving behavior. Peak-to-peak is the difference between the maximum and minimum velocity in the dataset. This metric is particularly useful for distinguishing between steady drivers and those who frequently accelerate or brake suddenly.

6. Conclusion

This paper presents a comprehensive and innovative solution for reckless driving detection that addresses one of the leading causes of road accidents and fatalities worldwide. By integrating advanced machine learning techniques, YOLO-based vehicle detection, Kalman filter tracking, and RNN-based time-series analysis within an electron app framework, the system provides effective means to identify and quantify reckless driving behaviors. The Reckless Status is calculated using a combination of statistical parameters such as standard deviation, mean velocity, variance, kurtosis, skewness, peak-to-peak, comparison mean with specific flags toggled when predetermined thresholds are exceeded. This approach successfully meets the goal of assessing driver’s reckless status by merging sophisticated computer vision algorithms with predictive analytics techniques. The RNN model is particularly adept at discerning abnormal driving patterns such as sudden acceleration, sharp turns, and irregular braking thereby offering a robust method for identifying drivers who may pose significant risks to road safety. Furthermore, the system shows significant potential for future advancements; real-time implementation could enable immediate identification and response to dangerous driving behaviors, and adaptability to various driving conditions, including urban and rural settings and different weather scenarios, would further enhance its applicability. Ultimately, this system could play a pivotal role in reducing traffic-related injuries and fatalities, contributing to a safer driving environment.

Acknowledgements

We extend our sincere appreciation to all those who have contributed significantly to the development of this paper "Reckless Status Determination by using Reckless Driving Detection". We would like to express our gratitude to the project committee, faculty members of Everest Engineering College for their guidance and unwavering support throughout. Their expertise has played a crucial role in shaping the content and structure of this paper. We are grateful for thorough guidance by Er. Nischal Regmi and Er. Shailesh Pandey. Their assistance has been invaluable in idea generation, writing, and researching. We would like to express our gratitude to our families and friends for their support and understanding throughout the difficulties involved in developing this paper. Their emotional and moral support acts as a catalyst for us to finish this paper.

References

Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y. M., 2020. YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934. Available at: https://arxiv.org/abs/2004.10934 [Accessed 4 Apr. 2025].

Goodfellow, I., Bengio, Y. & Courville, A., 2016. *Deep Learning*. Cambridge, MA: MIT Press. Available at: https://www.deeplearningbook.org/ (Accessed: 2 February 2025).

He, K., Gkioxari, G., Dollár, P. and Girshick, R., 2017. Mask R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969.

Jocher, G., et al., 2023. *YOLOv5 by Ultralytics*. GitHub repository. Available at: https://github.com/ultralytics/yolov5 [Accessed 4 Apr. 2025].

Lin, T.-Y., Goyal, P., Girshick, R., He, K. and Dollár, P., 2017. Focal loss for dense object detection. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988.

Mochnac, J., Marchevsky, S. and Kocan, P., 2009. *Bayesian filtering techniques: Kalman and extended Kalman filter basics*. In: IEEE Radioelektronika. IEEE, pp. 1–22. doi: 10.1109/RADIOELEK.2009.5158765.

Mahony, R. & Murray-Smith, R., 2000. *Optical Flow and Camera Velocity*. In: Proceedings of the 2000 IEEE International Conference on Robotics and Automation. IEEE, pp. 123–128. doi: 10.1109/ROBOT.2000.844015.

Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In: *Advances in Neural Information Processing Systems*, 28.

Tan, M., Pang, R. and Le, Q.V., 2020. EfficientDet: Scalable and efficient object detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10781–10790.

The Himalayan Times, 2024. *Traffic police warn against reckless driving*. Available at: https://thehimalayantimes.com/kathmandu/traffic-police-warn-against-reckless-driving [Accessed 20 September 2024].