# Computer Vision, Deep Learning and IoT Based Enhanced Early Warning System for the Safety of Rail Transportation


Sapni De Soysa
*Department of Computing and Statistics*
*Northshore College of Bussiness and Technology*
Colombo, Sri Lanka
sapnidesoysa@gmail.com

Sisuru Sendanayake
*Department of Computing and Statistics*
*Northshore College of Bussiness and Technology*
Colombo, Sri Lanka
dvc@northshore.edu.lk

Shakthi Manawadu
*Department of Computing and Statistics*
*Northshore College of Bussiness and Technology*
Colombo, Sri Lanka
manawadu.shakthi@gmail.com

Upendra Athipola
*Department of Computing and Statistics*
*Northshore College of Bussiness and Technology*
Colombo, Sri Lanka
upendra@northshore.edu.lk



***Abstract* — There is a growing demand for the improvement of the safety of transportation, particularly in the light of escalating frequency of collisions, where casualties and the associated costs are high. This is more so considering the frequent collisions of wild elephants on rail tracks in Sri Lanka. A warning system, if developed based on the concepts of Computer Vision, Deep Learning and IoT(Internet of Things) with a bleeding edge technology, would yield a solution for the improvement of the safety of different transportation modes with the entire process depending on data collection and sharing in real-time in order to achieve the required functionalities. In this study using LoRa(Long Range) advanced technology that best fits the requirements with its' simple but highly stable and effective architecture for communication between IoT units, a camera module having PoE(Power over ethernet) and night vision feature for capturing live footage in daytime as well as in night time and using the IoE(Internet of Everything) concept, the overall process successfully shares real-time data thereby providing an efficient outcome. Computer vision base works well with IoT and object detection through visualization and analysis of digital content, where it is 'videos' in the said context. Python was used as the primary implementation platform and C for the purpose of implementing the IoT functions. PyCharm and Arduino was used as the implementation environments which significantly support the core of implementation platforms selected. The 'dnn'(Deep Neural Networks) module of OpenCV highly supports the Python platform and is used for the purpose of object detection with a deep learning backbone. The dnn module is used over darknet considering the valuable benefits provided by it and mainly its' fast performance capacity. Related formulas are used for measuring the speed of the vehicle, distances, time slots, etc., and the deep learning algorithm is customized to support the scope of the object detection process. By analyzing the collected/shared data the risk levels and warning signal emittance can be derived accordingly. The deep learning approach used has vastly facilitated in gaining an outstanding scalability by multi-object and multiscale detection like humans, elephants, vehicles, etc., ensuring the improvement of prevention of many possible collisions. Though the system was successfully tested for rail transportation, the algorithm could be successfully used in any other transportation mode as well as to detect more object classes with the scalability of the deep learning algorithm.**



***Keywords* — *IoT, IoE, Deep Learning, bleeding edge technology, LoRa, PoE, OpenCV, dnn, Darknet***


## I. Introduction

Train accidents have become a major issue in the context of South Asia, due to lack of safety measures/systems that help to mitigate accidents successfully. The massive number of fatal road accidents that happen around the world has become a considerable issue causing dreadful harm to humans, animals as well as public/private property. The braking capability and the braking distance of these large vehicles are often a factor for road accidents. Many of the readily available systems use complex architectures and resources ending up with a costly solution. Some of them lacks the scalability and interoperability when it comes to safety solutions.

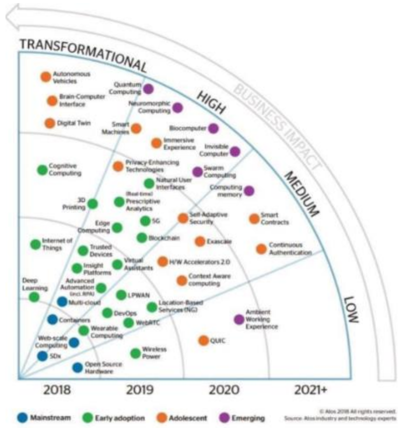


Fig. 1. Updating Technologies [7]

## II. Objectives

The objective of the system is to improve the safety of transportation systems, particularly rail transport, by reducing collision with wild elephants and other related consequences. The focus is given to adopting modern technological trends for building a simple yet effective early warning system, where it can be used to address many safety related scenarios in transportation worldwide. Achieving a high scalability was a priority, so that the system is not limited to a single scenario. The use of new technological trends to cost effectively build a warning system to address an impactful problem in the modern society.

Therefore, the primary goal of the study is to build an efficient and an innovative warning system with bleeding edge technologies, which contribute to modern advanced technological trends including data science and accelerated Machine Learning approach with IoE concept, leading to a highly expandable and cost-effective systematic solution, for the ultimate purpose of enhancing the safety of transportation worldwide.

## III. METHODOLOGY

The entire process of the system can be broken down into 4 main steps; object/barrier detection and recognition, speed and distance calculation, derive risk levels, display warnings. The camera unit will be responsible for capturing live video feed for the detection purpose while LoRa is used for communication between the units for data sharing.

The developed system consists of three IoT units for performing individual tasks and data collection purposes and the data is shared in real-time among the separate units. These three units will be installed at three separate locations; Unit 02 at accident prone locations which will primarily detect objects in real-time, Unit 01 at a considerable distance from Unit 02 for the purpose of on-coming vehicle detection and Unit 03 at the dashboard of the vehicle to display warning signals. The system uses a deep learning approach for object detection and recognition in real-time and the power of the chosen algorithm vastly helps to expand the functionality of the system. Moreover, the system is built considering the braking distances and capability for the efficiency and reliability of the system, where it can be used as an effective solution for reducing fatal road accidents other than railway transportation and is a cost-effective solution compared to other safety features like auto-pilot, Forward-collision warning (FCW), etc with the underlying simple architecture.

Unit 01 will keep checking for approaching vehicles and once a vehicle is detected, the system will calculate the speed of the vehicle and the signal is sent to Unit 02, where object detection starts on the live video feed from the camera and the Unit keeps checking for detections until a given time is reached. The risk level is derived depending on the distance between the approaching vehicle and unit 02 and the warnings are sent and displayed in Unit 03 based on the derived risk level and movements of the object.

Table 1. Risk levels and color gradings

| Risk Level | Indicative Color Grading |
|---|---|
| Low | **Green** |
| Medium | **Yellow** |
| High | **Red** |

LoRa technology is selected for communication of the system for it being a low-cost technology with a wide-area network protocol working on low-power. It offers a long coverage range of op to 10km and secure data transmission for M2M (Machine-to-Machine) and IoT applications.

Table 2. Comparison of Communication technologies [1]

| | Maximum Speed | Range | Cost | Power consumption | Drawback |
|---|---|---|---|---|---|
| **Bluetooth** | Up to 2Mbps | 20m (line of sight) | Free | Average | Not suitable for applications that need a high communication data rate. |
| **Wi-Fi** | Up to 450Mbps (depends on the standards) | < 50m | Somewhat high | High | Only works at where it is setup and has a limited range. |
| **ZigBee** | 250kbps | < 100m | Average | Somewhat low | Suffers low data speed, short range and high maintenance cost. |
| **GSM** | Depends on the network (> 40kbps) | Up to 35km (depends on the network) | High | High | High energy consumption |
| **RFID** | - | < 10m | Low | Low | Radio waves can be affected by metals and liquids. |
| **LoRa** | 32Kbps | Up to 10km | Low | Low | Low bandwidth and limitations on the used frequencies can affect message delivery. |

YOLOv3 algorithm based on deep learning was selected for the implementation of object detection for it works almost in real-time. Its' high confidence scores and FPS rates outperforms many other deep learning algorithms, and it is highly customizable and scalable due its' multi-scale detection, strong feature extraction network and multilabel classification. It has the capability of detecting 80 object classes by default and can can also be used for detecting any other object class(s) by training the network with simple modifications.

Table 3. Comparison between the reviewed algorithms (as per the approximate results on COCO dataset)

| | Confidence Rate | FPS | Accuracy | mAP | Inference Time (ms) |
|---|---|---|---|---|---|
| **YOLOv3 320x320** | High | 45 | Somewhat high | 51.5 | 22 |
| **Tiny YOLOv3** | Low | 220 | Low | 33.1 | - |
| **R-FCN** | High | 6 | Somewhat high | 51.9 | 85 |
| **SSD321** | High | 22 | Somewhat high | 45.4 | 61 |

OpenCV 'dnn' module was used as the backbone for running the algorithm and it provides easier configuration for

YOLO models to run on CPU. 'Darknet' which is the default backbone of YOLOv3 highly performs with a CUDA supported GPU, but its' implementation on a CPU lacks performance as well as accuracy. Darknet works best with C programming language over python, whereas OpenCV supports both C and Python with many supporting functions for object detection.

A trial was carried out to a railway scenario and several formulas were used for implementation purposes. The safety distance for which a train will be dragged by applying emergency brake, was calculated for the worst case scenario by applying conservation of energy theorem, the work done by *Frictional Forces(E)* at braking must equal the *Kinetic Energy($E_{kinetic}$)* dissipated (where maximum braking capacity is applied):

The energy present in an object in motion is given by the following equation [2]:

(m = mass of the object in kilograms, v = the velocity of the object in meters per second (m/s), $E_{kinetic}$ is the kinetic energy in joules (J))

$$E_{kinetic} = \frac{(mv^2)}{2d} \qquad (1)$$

($F_f$ = the force of friction in newtons (N), d = the stopping distance in meters (m), E = the energy produced by the brakes in Joules)

$$E = F_f \times d \qquad (2)$$

Applying conservation of energy (speed in $ms^{-1}$ = 22.2, safety distance = 350m);

$$E_{kinetic} = E;$$

$$\frac{(mv^2)}{2} = F_f \times d;$$

$$F_f = \frac{(mv^2)}{2d} \qquad (3)$$

$$d_{safe} = \frac{v^2}{22.2} \times 350 \qquad (4)$$

For the trial scenario, the speed was taken as $22.2ms^{-1}$, where it is the maximum speed of a train in Sri Lanka and the maximum drag distance of a train which comes in the maximum speed is 350m. Same formulas can be used by changing the values according to the scenario and the vehicle type.

## IV. Results and Discussion

The system was tested for the worst-case scenario of a train arriving at the maximum speed, and the results were taken to conform the required functionalities of the developed system. Fig. 2. shows an overview of the system with the IoT units.

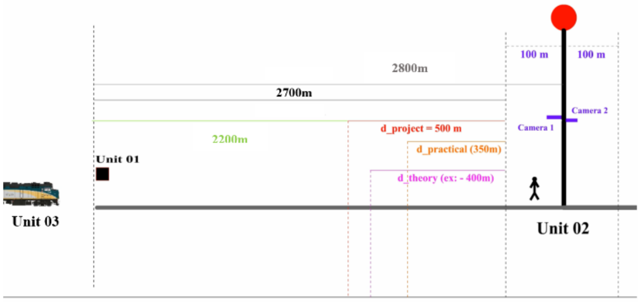


Fig. 2. System Overview

Table 4. Test results for a train

| Description | Input | Expected Results | Actual Results |
|---|---|---|---|
| Check if the calculations are made on vehicle detection | Two distance readings taken for a given time | Speed and safety distance are printed to the serial monitor | As expected |
| Check if the train detection signal and calculated values are sent to unit 02 | Train detection signal | Train detection message and values are printed to the python console | As expected |
| Check if multi-class detection is successful | Bounding boxes, class id | Humans are identified as 'person', animals identified as with relevant animals class name, vehicles identified with relevant vehicle class name | As expected |
| Check if multi-scale detection is successful | Objects with varying size scales | Bounding boxes are correctly derived for varying scales | As expected |
| Check if the time slots are properly derived and the risk levels are derived according to the time slots | Total time, elapsed time, number of risk levels(3) | Three time slots are derived according to the total time taken for the train to reach and the risk level is displayed according to the time slot in which the object(s) is detected. | As expected |
| Check if the warnings are sent to unit 03(vehicle) whenever a detection happens | Detection, risk level | Warnings are printed to the PyCharm console whenever a detection is done | As expected |
| Check if the respective light blinks according to the warning received and whenever a detection happens | Warning signal through pySerial | Green light blinks when the warning message is 'L'.<br><br>Orange light blinks when the warning message is 'M'<br><br>Red light blinks when the warning message is 'H'<br><br>Lights blink on object detection | As expected |

Following figures depict the test results.

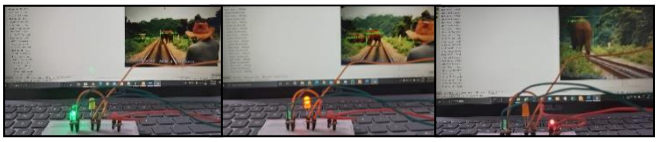

Fig. 3. Object classes are properly identified and Green LED, Yellow LED and Red LED are blinked respectively, when detections happen within 'low', 'medium' and 'high' risk levels

Moreover, the system was accepted by Sri Lanka Railways in order to mitigate the current issues in railway transportation of Sri Lanka.

## V. CONCLUSION

With the unaccountable number of vehicle accidents that happen regularly, many countries tend to go for solutions like auto-pilot vehicles, Forward-collision warning (FCW), vehicles with advanced security features, etc. However, these alternatives can most of the times be costly. Rather than going for a new vehicle with advanced safety technologies, it would always be a valuable alternative to go for a unit that can be easily installed.

As the system is developed and tested for railways using advanced computer technologies including computer vision, deep learning and IoT to achieve the core objectives of the study, and as the system is being accepted by Sri Lanka Railways, it conforms that the system serves the problem domain as well as the field of computer technologies for the ultimate goal of improving the safety of transportation systems. However, the system has the advantage of being more enhanced with integration of several trendiest modern technologies, to gain more connectivity of devices and people through collaborative AI approaches, supporting to move from a cutting edge to bleeding edge system with accelerated machine learning together with IoE concept used among the IoT units. Use of vast computer technology fields like data science gives the advantage of coming up with a more extensible and scalable cost-effective product, while fine tuning the deep learning algorithm to detect more object classes to be used in more relevant scenarios. Therefore, the focus is to further explore on other major transportation related issues and take the developed system to next level, where it can be launched as a single unit with an appropriate camera module to be integrated and the algorithm to be developed to do custom detections, and finally to be installed in major transportation mediums (trucks and buses) worldwide, as a cost-effective handy product, that will efficiently integrate with current high-end technological trends and blend with demanding IT fields like data science, AI, accelerated ML, etc.

## REFERENCES


[1] 6 Leading Types of IoT Wireless Technologies: BehrTech Blog. (2020, August 07). Retrieved April 06, 2020, from https://behrtech.com/blog/6-leading-types-of-iot-wireless-tech-and-their-best-use-cases/.

[2] Cunningham, T., Jenden, J., K., Lloyd, E., Stenhouse, K., Tierney, M., & Donev, J. (2018, May 18). Braking. Retrieved May 28, 2020, from https://energyeducation.ca/encyclopedia/Braking.

[3] Chowdhury, J. D. (2014, December 31). Cctv presentation. Retrieved May 23, 2020, from https://www.slideshare.net/JimuthDutta/cctv-presentation-43115622.

[4] Sonawane, A. (2018, March 29). YOLOv3: A Huge Improvement. Retrieved June 14, 2020, from https://medium.com/@anand_sonawane/yolo3-a-huge-improvement-2bc4e6fc44c5.

[5] Kathuria, A. (2018, April 23). What's new in YOLO v3? Retrieved June 22, 2020, from https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b

[6] Rosebrock, A. (2018, November 12). YOLO object detection with OpenCV. Retrieved October 12, 2020, from https://www.pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/.

[7] Fisher, M. (2018, October 10). Updated Technology Timeline to 2020 {Infographic}#AI #blockchain #deeplearning #Robotics #sdx #IoT #biotech #fintech #martech #ioe #wearables #tech pic.twitter.com/aR1NfcPtHT. Retrieved October 09, 2020, from https://twitter.com/fisher85m/status/1050025927374237696?lang=da.