End-to-End Crash Avoidance Deep IoT-based Solution

Mohammed Abdou[‖], Rawan Mohammed[‖], Zeinab Hosny[‖], Mayada Essam[‖], Mayada Zaki[‖], Maha hassan[‖], Mohammed Eid[‖], Hassan Mostafa[‖*]
[‖]Department of Electronics and Communications Engineering, Cairo University, Egypt [*]University of Science and Technology, Nanotechnology and Nanoelectronics Program, Zewail City of Science and Technology, October Gardens, 6th of October, Giza 12578, Egypt

***Abstract*—Fully Autonomous Driving is considered as one of the difficult problems faced the automotive applications. It is forbidden due to the presence of some restricted Laws that prevent cars from being autonomous for the fear of accidents occurrence. However, researchers try to reach autonomous driving as a new area for research for the aim of having a strong push against these restricted Laws. Crash Avoidance functionality is one of the most important features in Self-Driving Cars that is partially integrated recently. We propose an end-to-end Crash Avoidance Deep IoT solution which is decomposed into two main parts: a) Detection Deep Neural Network which aims to detect accident occurrence in front of the ego-vehicle, and b) Accident Information Spreading IoT which is responsible for informing upcoming vehicles that there is an accident, then these vehicles will be able to take the reasonable actions either changing their routes, or changing their lanes avoiding crash. Due to the lack of Crash benchmarks, we build our own benchmark, depending only on a front camera, using ROS-Gazebo Simulation environment covering various crashes situations. In General, our proposed idea is the first solution that merges Deep Learning with IoT in automotive applications.**

***Index Terms*—Autonomous Driving, Crash Avoidance, Deep Learning, IoT**

I. Introduction

Thousands of people across the world are losing their lives due to car accidents and road disasters every year and the related lost costs due to medical expenses and general maintenance and repairs costs of the road and highway systems are very large. Another big problem because of accidents is the traffic crowd, many people lose their time in roads due to car accidents, and this crowd also makes the ambulance and the rescue reach the accident location very late which may cause losses in lives. In this paper, we aim to help in saving these problems by providing a car accident avoidance system. Deep Learning and Computer Vision, recently, invade the automotive field powerfully. Deep Learning is contributing greatly in many automotive applications. Forward Collision Warning (FCW) and Automatic Emergency Braking (AEB) are considered as the initial trials to integrate Crash Avoidance functionality. These actions are taken by the ego-vehicle based on integrating advanced sensors like: Laser, Camera and Radar, but these Vehicle actions lead to occurrence of many crashes, in addition to being source of congestion because of its poorness. We propose a system in cars to help emergency to reach faster to the accident site and to help saving critical conditions. This would happen by sending an alarm message through V2V from the car that captured the accident, from its front camera, to all nearby cars so the drivers can change their lane or even route to avoid traffic, allowing emergency to arrive faster and help in saving lives. Fig. 1 shows the system flow.

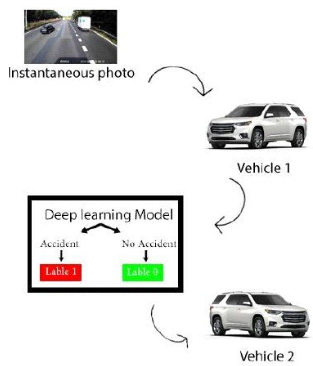

Fig. 1. System Flow

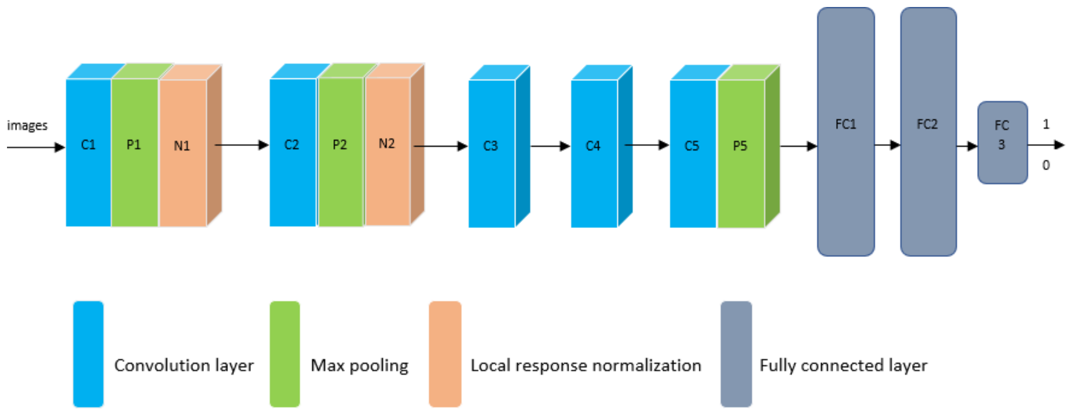

Fig. 2. Network Architecture

A. Related Work

The paper [1] has presented a brief survey on automatic road accident detection techniques. These techniques include smartphones, GSM and GPS technologies, vehicular ad-hoc networks, and mobile applications. Among these four accident detection methods, accident detection using VANET (Vehicular ad-hoc network) is the best method because it not only detects an accident but also provides an optimum route to the ambulance to reach the accident spot as soon as possible. However, the purpose of the thesis [2] is to investigate deep learning and its use for autonomous collision avoidance. A deep neural network was chosen, implemented and used with the robotic car for avoiding collisions in real-time. Also, the paper [3] proposed a new approach that is based on detecting damaged vehicles from footage received from surveillance cameras installed in roads and highways which would indicate the occurrence of a road accident. Detection of damaged cars falls under the category of object detection in the field of machine vision. This paper proposed a new supervised learning method comprising of three different stages that are combined serially into a single framework. The three stages use five support vector machines trained with the features Histogram of gradients (HOG) and Gray level co-occurrence matrix (GLCM). The paper [4] proposed a solution based on IoT accident prediction and detection using a supervised machine learning algorithm. The system collects the necessary information or data from using MEMS and vibration sensor. Through a machine learning algorithm (k-nearest neighbors), the accident will be predicted using data sets. A notification is passed to the users predefine contacts, nearby police station, and hospital.

II. Methodology

The proposed solution aims to build End-to-End Crash Avoidance system in cars as follows:

- Analyzing the video which is taken by the front camera of each car.
- Frames are considered the input of a well-trained neural network [12].
- Classifying the frames whether they have an accident or not.
- Sending an alarm, if an accident occurs, to all nearby cars using V2V Communication system [14] [15].

The first subsystem is the classifier network used to detect crashes. Alexnet network is the used classifier in the model which considered as a convolutional neural network that has had a large impact on the field of machine learning [5]. The architecture used is similar to the design of AlexNet paper [5] but with some modifications to fit the concerned problem as shown in Fig. 2. The proposed classifier is composed of 5 convolution layers followed by 3 fully connected layers as shown in Fig. 2. The input of the network is the images provided from the simulation of size 224*224 JPEG. The output is mapped to be either 1 or 0, which indicates if there is an accident in the image or not. The total number of parameters in the network is 143.47405 million parameters. Local response normalization is applied to the first two convolution layers [5] where the used type in the network architecture is Inter-Channel LRN with the following hyper-parameters:

- n (depth radius) = 5
- k (avoiding singularities parameter) = 2
- α (normalization constant) = 1e-4
- β (contrast constant) = 0.75

The second subsystem is IoT protocol for spreading accident information among all vehicles. Among different ways for implementing IoT protocol, V2V (vehicle-to-vehicle) communication is chosen to be the communication way in the system. V2V is a way for vehicles to send and receive signals to each other to explain their location, speed and direction [6]. The used V2V technology is based on IEEE STD 802.11PTm-2010: Wireless Access in Vehicular Environments

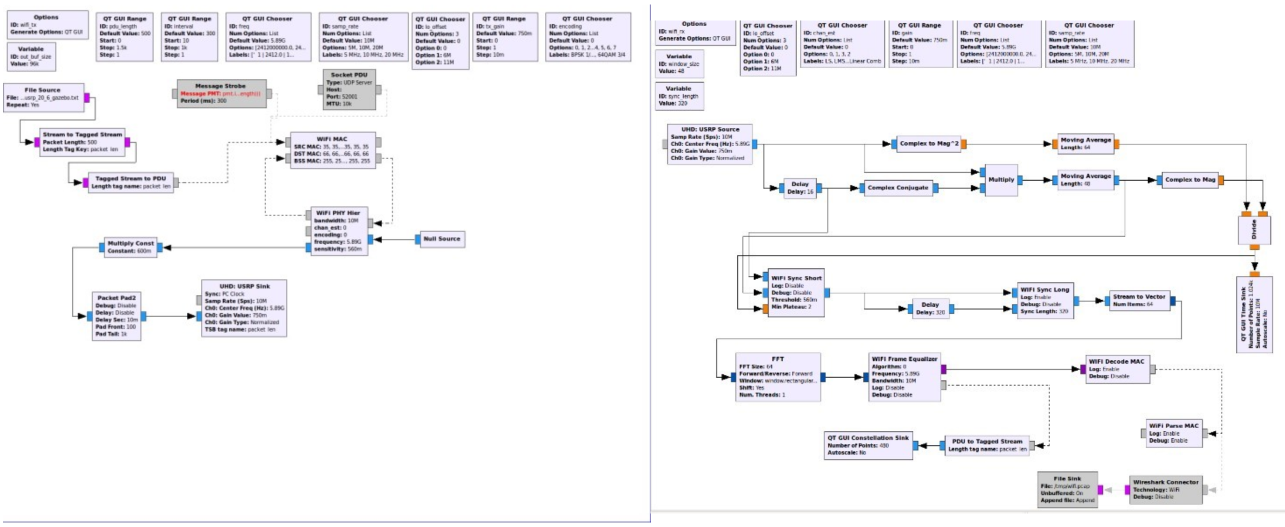

Fig. 3. Wifi Transmitter and Receiver

which explains the physical and MAC (Media Access Control) layers of vehicular transceivers [7]. The operating frequency for this technology is 5.9 GHz [6].

To implement the V2V communication subsystem, a software and a hardware tool are used: GNU Radio is a free software development toolkit that provides signal process- ing blocks to implement software-defined radios and signal-processing systems [8]. Wi-Fi Transmitter and the Wi-Fi Receiver block diagrams [8] are used to transmit data through the USRP (Universal Software Radio Peripheral) channel as shown in Fig. 3. USRP B200 hardware kit [13] is used to transfer the data from one end to the another end [9] after setting the required antenna parameters and bandwidth occupied by the transferred data and this happened through two principle blocks in GNU radio; USRP sink which is responsible for adjusting the parameters at the transmitter side and USRP source which is responsible for adjusting the parameters at the receiver side.

Integrating both subsystems is done by writing the classifier results in a text file by python code then putting this file in the file source of the Wi-Fi transmitter. Two USRPs are used; one at the transmitter side and the other at the receiver side and these USRPs are interfaced with GNU radio by some terminal commands. And by connecting RF antenna to each USRP, data in the file source can be transmitted from one end to another.

III. Experimental Setup

ROS, Robotics Operating System, processes are represented as nodes in a graph structure, connected by edges called topics [10]. ROS nodes can pass messages to one another through topics, make service calls to other nodes, provide a service for other nodes, or set or retrieve shared data from a communal database called the parameter server. Building the environment throughout different static objects like buildings and trees, and dynamic objects like vehicles. Static objects: Entities marked as static, those having the $< static > true < /static >$ element in the SDF file, are objects which only have collision geometry. All objects which are not meant to move should be marked as static. Dynamic objects: Entities marked as dynamic, either missing the "$< static >$" element or setting false in the SDF file, are objects which have both inertia and collision geometry. Accident and no-accident scenarios were created to imitate the real situations that happen in the real life as much as possible considering: Different environments, Types of vehicles, Crashes with different objects and with different sides, Crashes with simulated persons.

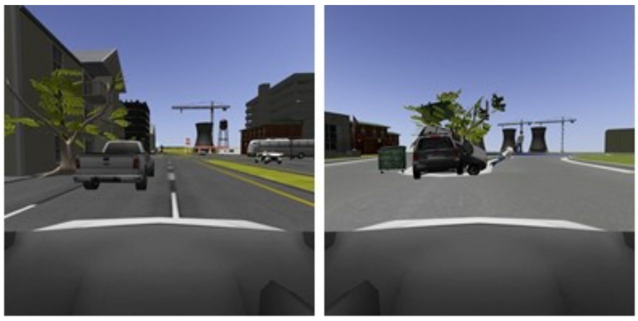

Fig. 4. No Accident and Accident Examples

The simulation of the Prius car in the environment was done using ROS Kinetic and Gazebo8 [11] as shown in Fig. 5. The speed and directions of the car are controlled through a python script. ROS enabled the simulation to be developed faster by using existing software and libraries.

Fig. 4 shows examples of No accident and accident images simulated in Gazebo environment. The output of the Gazebo simulator was around 5000 Labeled images as the training and the validation dataset. Then by using data augmentation, the total number of images was doubled to reach 7560 images for training and 2047 images for validation. For testing the model, another 1032 images were generated from the simulator. Normalization

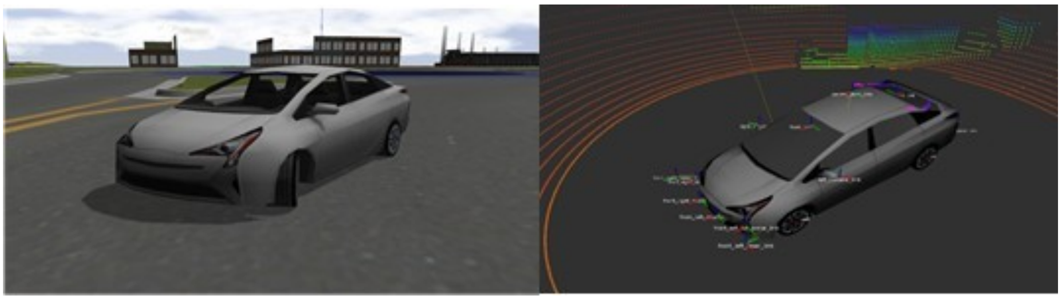

Fig. 5. Prius Car

is done by dividing each value by 255 so the values range becomes between 0 and 1. This allows easier and faster computation process in the convolution layers. The input images are resized to be 720*720 pixels. For better feature extraction the unwanted parts can be cropped from the dataset like the hoods. The final step is resizing all the images in the dataset to the chosen size of AlexNet input which is 224*224 pixels. After modifying the network hyper parameters several times and applying the training and validation set from data preparation, the following hyper parameters is chosen to achieve the best accuracy on the test dataset:

- Number of epochs is equal to 25.
- Fully connected layers consist of 1024 neurons.
- Batch size is equal to 120.
- Keep probability is equal to 0.5.
- Learning rate is equal to 1e-4.

IV. Experimental Results

Testing phase is done through two phases. The first phase is to test the classifier network for different accident and non accident scenarios. Our problem is a supervised learning problem so the accuracy was calculated by comparing the predicted label with the original label of the test dataset. Training process was done on 7560 images and 2047 images for validation. Firstly, testing is done on about 1032 balanced images from accident and non accident scenarios extracted from Gazebo different from the scenarios that are used in training the model. In training the model, some hyper-parameters tuning are done like number of epochs, dropout probability, number of neurons in the three fully connected layers and number of kernels in the convolution layers. The training accuracies of all these models are close to each other. So, only the test accuracy will enable us to choose the best model. The best model gave the highest test accuracy among other models which is 77.1%. Secondly; we tested on real data to ensure that the model can generalize. The used real data is represented in 526 non-accident images found in a paper of "Road-Accident Detection" and 820 balanced accident and non-accident frames captured from online videos. MATLAB code is used that takes the video and captures frames from it and we made a suitable delay between capturing frames so that the non accident frames and the accident frames can be extracted easily from the video. We also tested the 526 non-accident images from the paper and the resulting accuracy

TABLE I
TESTING RESULTS FOR CALIBRATED REAL DATA

Types of test data		No. of images	Test accuracy
Paper dataset	Non accidents	526	61%
Online videos	Accidents	410	55%
	Non accidents	410	44%
	All	820	48%

found to be 48%. And we thought that it is a predictable result as the distribution of these images is different from Gazebo images, so calibration phase is needed for the real images, which means having the same distribution in both images; so we calibrated the real data by the parameters found in the generated code from a tool in MATLAB. TABLE I shows the different results after calibrating the real data. From TABLE I we conclude that:

- Testing the model on real non-accident scenarios from online videos that contain bikes and motorbikes results in an accuracy which is not satisfied enough compared to the accuracy resulted from the paper dataset which is close to the environment exists in Gazebo.
- The sensitivity of detecting accident is more than detecting non-accident which isn't bad.

The second phase is to perform a real complete simulation cycle for the system as follows:

- Taking live video in streets of the college using USB camera connected to Laptop.
- Using Open-CV, we capture frames from this video with a sufficient delay between frames to be the input to the classifier deep learning model.
- Testing these frames using the proposed classifier model and writing the predictions in a text file.
- Transmitting this text file which is in the file source in Tx side to the Rx side by using the USRP channel.

From this real simulation, results are as follows:

- The test accuracy for non-accident video is 90%
- The test accuracy for accident video is 73.3%
- V2V communication range is theoretically up to 300 meters but when using this real simulation, this range is affected by weather conditions like any other wireless channel.

- The time taken for the complete cycle starting from capturing frames till receiving the data in the Rx side depends on number of frames captured in the one cycle, the conditions of the channel between the transmitter and the receiver and running the model either on CPU or GPU (in this real simulation using the classifier model on CPU). For 20 frames it takes about 22 seconds.

V. CONCLUSIONS

End-to-End Crash Avoidance Deep IoT Solution System is proposed solving the poor actions taken nowadays by the already existing solutions like Forwarding Collision Braking (FCW), and Automatic Emergency Braking (AEB). It is considered as the first time merging the Deep Learning field with the V2V communication technology especially in the automotive applications. Crash Avoidance Functionality is one of the most important functionalities in Self-Driving Car Systems. Currently, it is partially integrated into the system, so the proposed architecture aims to extend this functionality wider. Our proposed architecture depends on the detection neural network, and Accident Information Spreading IoT (especially V2V protocol). Detection Neural Network aims to detect that there is a collision in front of the ego-vehicle. This Network is trained on simulated data generated from ROS with various scenarios, and it could be augmented with some Real data in order to ensure having a generalized Deep Learning model. Accident Information Spreading IoT aims to inform the upcoming vehicles with advanced information about the accident.

ACKNOWLEDGMENT

This work was partially funded by ONE Lab at Zewail City of Science and Technology and at Cairo University, NTRA, ITIDA, and ASRT.

REFERENCES

[1] U. Khalil, T. Javid, and A. Nasir, “Automatic road accident detection techniques: A brief survey,” in 2017 International Symposium on Wireless Systems and Networks (ISWSN). IEEE, 2017, pp. 1–6.
[2] O. Str o¨mgren, “Deep learning for autonomous collision avoidance,” 2018.
[3] V. Ravindran, L. Viswanathan, and S. Rangaswamy, “A novel approach to automatic road-accident detection using machine vision techniques,” Int J Adv Comput Sci Appl, vol. 7, no. 11, pp. 235–242, 2016.
[4] A. K. Bharath P, Saravanan M, “Literature survey on smart vehicle accident prediction using machinelearning algorithms,” International Research Journal of Engineering and Technology, Vol. 06, Issue. 02, e-ISSN. 2395-0056, p-ISSN. 2395-0072.
[5] I. Sutskever, G. E. Hinton, and A. Krizhevsky, “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, pp. 1097–1105, 2012.
[6] “vehicle-to-vehicle communications, extreme tech.” 2019. [Online]. Available:https://www.extremetech.com/extreme/176093-v2v-what-are-vehicle-to-vehicle-communications-and-how-does-it-work
[7] “Wireless lan 802.11 wi-fi, engineering and technology history wiki, extreme tech.” 2019. [Online]. Available:https://ethw.org/Wireless LAN 802.11 Wi-Fi
[8] “Gnu radio manual and c++ api reference.” 2019. [Online]. Available: https://www.gnuradio.org/doc/doxygen
[9] “Usrp b200 usb software defined radio (sdr), ettus research.” 2019. [Online]. Available: https://www.ettus.com/all-products/ub200-kit
[10] “Ros wiki-documentation, ros.org.” 2019. [Online]. Available: http://wiki.ros.org/Documentation
[11] “Gazebo tutorial.” 2019. [Online]. Available: http://gazebosim.org/?fbclid=IwAR2bLTUT8AysuDM6DcVunehOO1D4OZLXS6sQYU5YQ1KE6cdxYdUle5VO2U
[12] M. Elgammal, A. N. Mohieldin, and H. Mostafa, “A Comparison of Artificial Neural Network (ANN) and Support Vector Machine(SVM) Classifiers for Neural Seizure Detection”, IEEE International Mid-West Symposium on Circuits and Systems (MWSCAS 2019), Dallas, Texas, USA, In Press.
[13] C. Osama, Heba Magdy, H. Mahmoud, K. Khaled, M. Osama, Yara Hossam, S. Hosny, and H. Mostafa, “Wireless Communication Between ZYNQ 7000 FPGAs using USRP”, IEEE International Conference on Modern Circuits and Systems Technology (MOCAST 2019), Thessaloniki, Greece, pp. 1-4, 2019.
[14] A. Hosny, M. Yousef, W. Gamil, M. ADEL, H. Mostafa, and S. M. Darwish, “Demonstration of Forward Collision Avoidance Algorithm Based on V2V Communication “, IEEE International Conference on Modern Circuits and Systems Technology (MOCAST 2019), Thessaloniki, Greece, pp. 1-4, 2019.
[15] H. Fahmy, G. Baumann, M. AbdelGhany, and H. Mostafa, “V2V-Based Vehicle Risk Assessment and Control for Lane-Keeping and Collision Avoidance”, IEEE International Conference on Microelectronics (ICM 2017), Beirut, Lebanon, pp. 61-65, 2017.