Accident Detection using Convolutional Neural Networks

Sreyan Ghosh
Department of Computer Science and Engineering,
Christ (Deemed to be University),
Bangalore, India
Email: gsreyan@gmail.com

Sherwin Joseph Sunny
Department of Computer Science and Engineering,
Christ (Deemed to be University),
Bangalore, India
Email: sherwin.sunny@gmail.com

Rohan Roney
Department of Computer Science and Engineering,
Christ (Deemed to be University),
Bangalore, India
Email: rohanroney98@gmail.com

***Abstract*— Accidents have been a major cause of deaths in India. More than 80% of accident-related deaths occur not due to the accident itself but the lack of timely help reaching the accident victims. In highways where the traffic is really light and fast-paced an accident victim could be left unattended for a long time. The intent is to create a system which would detect an accident based on the live feed of video from a CCTV camera installed on a highway. The idea is to take each frame of a video and run it through a deep learning convolution neural network model which has been trained to classify frames of a video into accident or non-accident. Convolutional Neural Networks has proven to be a fast and accurate approach to classify images. CNN based image classifiers have given accuracy's of more than 95% for comparatively smaller datasets and require less pre-processing as compared to other image classifying algorithms.**

Keywords—Convolutional Neural Network; Accident Detection; Deep Learning; Video Classification; Recurrent Neural Network

I. INTRODUCTION

Over 1.3 million deaths happen each year from road accidents, with a further of about 25 to 65 million people suffering from mild injuries as a result of road accidents. In a survey conducted by the World Health Organisation (WHO) on road accidents based on the income status of the country, it is seen that low and middle-income or developing countries have the highest number of road accident related deaths. Developing countries have road accident death rate of about 23.5 per 100,000 population, which is much higher when compared to the 11.3 per 100,000 population for high-income or developed countries [1]. Over 90% of road traffic related deaths happen in developing countries, even though these countries have only half of the world's vehicles. In India, a reported 13 people are killed every hour as victims to road accidents across the country. However, the real case scenario could be much worse as many accident cases are left unreported. With the present data, India is on the way to the number one country in deaths from road accidents due to the poor average record of 13 deaths every hour, which is about 140,000 per year [2]. An accident usually has three phases in which a victim can be found.

First phase of an accident is when the death of the accident victim occurs within a few minutes or seconds of the accident, about 10% of accident deaths happen in this phase.

Second phase of an accident is the time after an hour of the accident which has the highest mortality rate (75% of all deaths). This can be avoided by timely help reaching the victims. The objective is to help accident victims in this critical hour of need.

Third phase of an accident occurs days or weeks after the accident, this phase has a death rate of about 15% and takes medical care and resources to avoid.

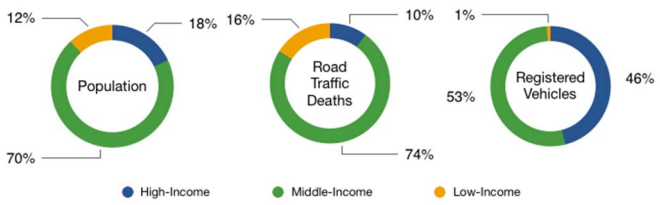

Fig. 1. Comparative analysis of population, income and road accidents

The main objective is to incorporate a system which is able to detect an accident form video footage provided to it using a camera. The system is designed as a tool to help out accident victims in need by timely detecting an accident and henceforth informing the authorities of the same. The focus is to detect an accident within seconds of it happening using advanced Deep Learning Algorithms which use Convolutional Neural Networks (CNN's or ConvNet) to analyze frames taken form the video generated by the camera. We have focused on setting up this system on highways where the traffic is less dense and timely help reaching the accident victims is rare. On highways we can setup CCTV camera's placed at distance of about 500 meters which act as a medium for surveillance, on this camera we can set up the proposed system which takes the footage from the CCTV camera's and runs it on the propsed accident detection model in order to detect accidents.

In this system, we have a Raspberry Pi 3 B+ Model which acts as a portable and remote computer to be set up on a CCTV camera. For demonstration purposes, we will be using a Pi Camera which can be directly set up on a Raspberry Pi. We have pre-trained an Inception v3 model to be able to detect accidents by training it on two different sets of images and sequence of video frames. The images and video frames are 10,000 severe accident frames and 10,000 non-accident frames. The Inception v3 algorithm can now detect an image or frames of a video to be an accident frame by up to 98.5% accuracy. This model was then implemented on a Raspberry Pi using TensorFlow, OpenCV and Keras. When a video is shown to the Raspberry Pi through the Pi camera, it runs each frame of the video through the model created and then predicts whether the given frame is an accident frame or not. If the prediction exceeds a threshold of 60% or 0.6 the Raspberry Pi

978-1-5386-9319-3/19/$31.00 ©

then initiates the GSM module setup with it to send a message to the nearest hospital and police station, informing them about the accident which has been detected with the timestamp of when it occurred, the location of where it occurred, and the frame at which the accident was detected for further analyses. Also, an emergency light lights up.

The system we have made can detect accidents to an accuracy of about 95.0%. It can be done on a Raspberry Pi which is a card-sized computer, which makes it easily portable and remote. The system developed can act as a reliable source of information in detecting accidents which can be done automatically. This project would help us in reducing the ginormous number of road accident related deaths that occur in our country.

II. Related work

In this section, we have tried to compare our work with other accident detection techniques. Most of the studies in this field revolve around the enhancement of tangible infrastructure rather than on Intelligent Transportation Systems (ITS) which include traffic congestion detection, accident detection, detecting the occurrence of an event etc. Even the few existing studies in the domain lack implementation details and are terrain specific i.e. there are constraints both in the geographical as well as demographic aspects. These techniques have been discussed below:

Lexus vehicles [3] introduced in 2014 came with a feature called the "Lexus Enform" wherein an impact sensor was placed at the rear end of the vehicle. In the occurrence of an accident, the sensors would react and thus notify the user via the application. However, the disadvantages of this system were plenty. Sensors were to be placed in every individual vehicle rendering the concept expensive. Also, it requires physical entities like smartphones.

An ancillary company of General Motors called OnStar Corporation introduced an accident notification application called Chevy star. It offered options like on-field assistance to victims as well as a self-regulated crash response [4]. However, this service was based on a subscription model rendering the service expensive. Also, reviews suggested that the service lacked quality because of which the system itself was ineffective.

SoSmart SpA came up with a smartphone application called SOSmart [5] which provided free assistance to the victim of the accident at the time of occurrence. This facility was easy to use and you could avail help at the click of a button. But the obvious flaw is that it is a manual reporting system.

There are certain systems known as ad-hoc systems which are widely used for the collection of traffic data. However, the limitation of these systems revolves around maintenance of communication and data transmission for different terrains and conditions [6][7].

We have come to notice that most accident detection systems make use of expensive sensors placed on the body of the vehicles or it makes use of existing sensors on a smartphone. This dependency of sensors makes this method expensive and less effective as compared to the proposed accident detection system.

III. Components used

A. *Raspberry Pi 3 Model B+*

Raspberry Pi 3 Model B+ is a very small-sized portable computer created by the Raspberry Pi foundation in the United Kingdom to provide a low-cost experience for programming enthusiasts and helping them to understand the basics of computer science. The Raspberry Pi comprises of a 1.4 GHz 64-bit quad-core processor, a dual-band wireless LAN, 4 USB ports along with other features like Bluetooth 4.2, faster Ethernet and is originally shipped with 1GB of RAM. It consists of 40 GPIO (General Purpose Input Output) pins which helps us to easily interface the required hardware with the pi. It does not consist of massive storage like the built-in hard drive, but it does consist of microSD card used for light storage and booting applications. The operating system that the Pi works on is known as Raspbian installed NOOBS. The various programming languages supported in Pi are Python, Wolfram etc. The two key application areas are:

- Interfacing of various hardware components
- Understanding of basic programming concepts

B. *GSM Module SIM800L*

Global System for Mobile Communication is one of the most widely used mobile telephony systems. SIM800L helps to connect onto any global GSM network using a 2G SIM. GSM uses the concept of Time Division Multiple Access. Data is digitized and compressed which is later sent down a communicating channel along with two streams of the user's data each of which has its own unique time frame. SIM800L is quad-band i.e., operation lies in the 850/900/1800/1900 MHz frequency band.

The SIM800L module is capable of supporting a quad-band GSM network, which can be utilized for GPRS as well as remote SMS transmission. Thus, we can send and receive messages using cellular network along with an arbitrary location based on the tower location to which the SIM is synched at that particular moment in time. The board features compact size and low current consumption.

Communication is achieved using the UART port using the various AT commands. Additional features include scanning and reception of FM radio broadcasts.

C. *Pi Camera*

The main task of a Pi camera is to take still photographs and high-definition videos. The module consists of a five-megapixel camera capable of capturing stills as well as support for the 1080p30, 720p60, and VGA90 video modes. It is attached to the CSI port of Pi using the 15cm ribbon cable. The images obtained are generally in JPEG or JPG format. The camera module can be altered to implement various additional effects like time-lapse or slow-motion. Additional libraries can be used to create effects as well.

It can be accessed using the Pi camera Python library. The camera module is used for a wide range of home-security applications as well as in wildlife camera traps.

IV. Software Used

A. Keras

Keras is an open source minimalist library which was developed in Python in order to experiment extensively in the field of deep neural networks. It has the capability to run on top of other Python libraries like Theano and Tensorflow. It was created and maintained by Francois Chollet as part of project ONEIROS. Its primary advantage is that it creates a high-level set of abstractions which are highly intuitive which aids in the creation and development of deep-learning models without any added dependency on the computational backend available. It was developed to work as an interface rather than an independent framework for machine learning. Keras makes working on text data and images easier by providing implementations on blocks of neural networks like activation functions, layers, optimizers and a range of other tools. Apart from supporting convolutional and recurrent neural networks, it's capable of supporting additional utility layers like dropout and pooling. The key advantages of Keras include modularity, minimalism and extensibility.

B. OpenCV

OpenCV (Open Source Computer Vision Library) is an image processing library containing programming functions in order to facilitate research in the domain of computer vision and to provide support for advanced CPU-intensive projects. Since it is released under the BSD license, it is free for both academic as well as commercial use. It supports programming languages like C++, Java and Python as well as the common operating systems like Windows, Mac OS, Linux, iOS and Android. OpenCV was primarily designed in order to provide computational efficiency along with the creation of real time applications. It uses the concept of multi-core processing along with the advantages provided by hardware acceleration provided by an OpenCL based GPU interface. OpenCV also supports a wide range of deep learning libraries like Tensorflow. The main application areas involve ego motion estimation, facial recognition systems, gesture recognition, mobile robotics and augmented reality.

For the scope of the proposed system, there were several challenges we had to face with the images. Initially, the model had difficulties in predicting the right class since the only difference between cars and broken cars were dents, broken headlights etc. Thus, random noise was being accounted during the prediction since not all the accident images were of high clarity.

$$opencvcv.fastNlMeansDenoisingColoredMulti()$$

The above-mentioned function has been used to de-noise all the image sequences prior to feeding them to model.

V. Implementation

CNNs are used in modelling spatial data like images. CNNs have been successful in tasks like image classification, object detection, etc. LSTMs are used to model sequential data and make predictions based on them. LSTMs are widely used in areas of text classification, making language models, sequence generation, etc. Standard LSTMs can be used directly on sequential data where the input is spatial. Thus, to perform tasks which involve sequences of images or videos, a CNN-LSTM architecture needs to be used.

The proposed model is a fusion of CNN and LSTM layers for continuous video classification taken from a camera. The CNN part of the proposed model was mainly inspired by the Inception v3, but with certain tweaks it has fitted well to our training images. The LSTM layers were added to the existing Convolution Network to take into account temporal features along with spatial features. This is further divided into the convolution and recurrent parts of the model.

In a CNN-LSTM network, the CNN is primarily used for feature extraction from the images, which is passed on to the LSTM for sequence prediction. They are widely used in tasks similar to Activity Recognition, Image Description, Video Description etc.

A. Convolution Layer

The convolution layer is the first step toward extracting useful information from images. Convolution helps in preserving the relationship between the pixels of the image by learning image features using small squares of input data. This is done as a mathematical operation that takes two inputs, which are a part of the image as a matrix, and a kernel or filter.

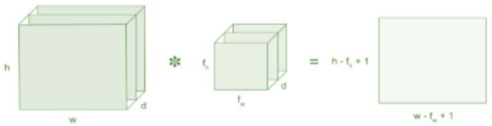

Fig. 2. Image matrix multiplies kernel or filter matrix

Fig. 2. shows an image matrix of dimension $(h \times w \times d)$, a filter $(f_h \times f_w \times d)$ which outputs the dimension$(h - f_h + 1) \times (w - f_w + 1) \times 1$

It is a picture of a 5×5 image matrix convoluted by a 3×3 filter matrix. The resultant matrix is a 3×3matrix, called the feature map. Different filters applied to the same image can extract different information from the image. It may be used to extract spatial information such as edge detection, blur, etc.

B. Pooling Layer

The main job of pooling layers is to reduce the number of parameters when the given image is too large. Spatial pooling, also known as subsampling or down sampling, diminishes the dimensionality of each Feature Map but maintains the most relevant information. Pooling layers are generally of 3 major types:

- Average Pooling
- Max Pooling
- Sum Pooling

C. Fully Connected Layer

In this layer, we smoothen our grid into a vector and give it into a layer like a primitive neural network.

D. LSTM

LSTM units are units of an RNN. LSTM units are made up of a cell, an input gate, an output gate and a forget gate. The cell remembers values over time intervals and the three gates help in regulation of the flow of information into and out of the cell.

VI. Inception v3 Analysis

For several years now, Inception v3 has served as a benchmark for image classification. A brief comparison as shown in table 1 shows several benchmark CNN models for image classification shows that Inception v3 gets lowest error rate among all others [8][9].

TABLE 1: Error comparisons of various CNN models on Imagenet dataset

Model	Error Rate
AlexNet	15.3%
Inception (Google Net)	6.67%
Inception v2	4.9%
Inception v3	3.46%

The inception v3, unlike normal neural networks, works on a heterogeneous set of convolutions. This allows the model to dive deeper into the image and extract more features. The advent of Inception models came from the very question "Why not do it all?" One is generally confused as to which layers to out in a convolutional neural network. Sometimes different filter sizes also seem to work considerably well. Thus, through the Inception architecture, we aim at putting several convolutional filters of different dimensions and also pooling layers, all into the same layer of the network and allow the model to choose the best. This makes the model far more complicated than a primitive CNN but also improves classification margin considerably. The Fig. 4. architecture does two functions. First, it uses the smaller convolutions for the successful recovery of the basic local features. Secondly, it utilises the larger convolutions in order to recover the more complicated abstracted features. The filter sizes in the model were restricted to 1×1, 3×3 and 5×5. Since convolutions are quite expensive, a 1×1 convolution before the actual 3×3 or 5×5 convolutions help in dimensionality reduction and reducing the number of operations performed, thus helping in saving memory.

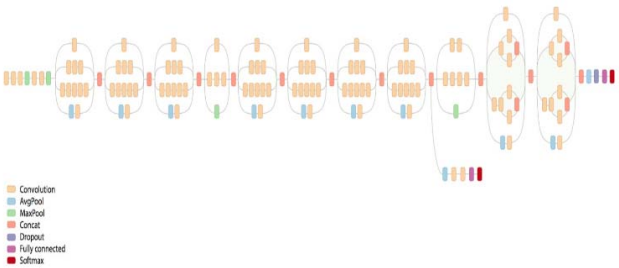

Fig. 3. Schematic Representation of Inception v3

Deep CNNs often suffer from vanishing/exploding gradients. Thus, the auxiliary classifiers in Inception v3 is used as a regularizer. The final loss calculated is an average of all losses from all the softmax layers in the model.

Throughout the model, "same convolutions" have been used to keep the dimensions of the output same for filter concatenation after each inception step.

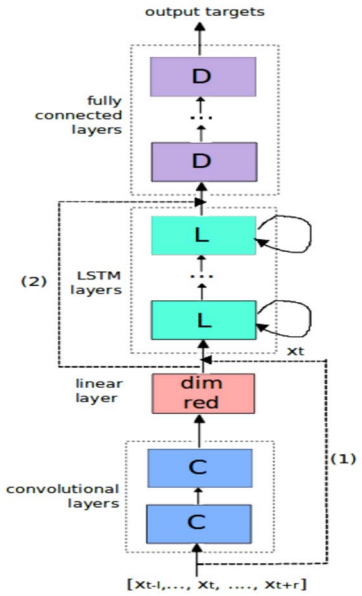

Fig. 4. Architecture of proposed model

As shown in Fig. 4. the LSTM part of the model consists of 2 layers, each with 256 hidden LSTM units followed by dense layers. This shallow network outperformed all other deeper networks. Both these layers were accompanied by a dropout layer in between to prevent overfitting. The data was also transformed to a 4D tensor before it was input to the LSTM.

As mentioned earlier, we have primarily used the Inception v3 for building the proposed model. Extra layers including the LSTM and dense layers were added to the end of the network after the convolution and pooling layers by dropping the final bottleneck layers.

In our approach, we have used the output of the final pool layer and not our softmax predictions to be feed into the LSTM layers, because it gives us a higher accuracy every time, we train it. Our final pool layer gives us a feature representation of the image rather than actual probabilities.

First, we run every frame from our video through Inception v3 and save the output from the final pool layer of the network. We effectively drop the top classification part of the network so that we end up with a 2,048-d vector of features from the final pool layer that we can pass to our RNN.

Second, we convert those extracted features into sequences of extracted features. We stitch the sampled frames from our video together, save that to the disk, and now we train different RNN models without needing to continuously pass our images through the CNN every time we read the same sample or train a new network architecture. In order to do that, we loop through each frame in chronological order, add it to a queue of size N, and pop off the first frame we added. For our experiment we used a queue of size 5, which is 5 time-steps for our LSTM layers. 5 was used so that we could get a prediction each second, since the Pi Camera records videos at 5 frames per second. The working of this has been shown in Fig. 5.

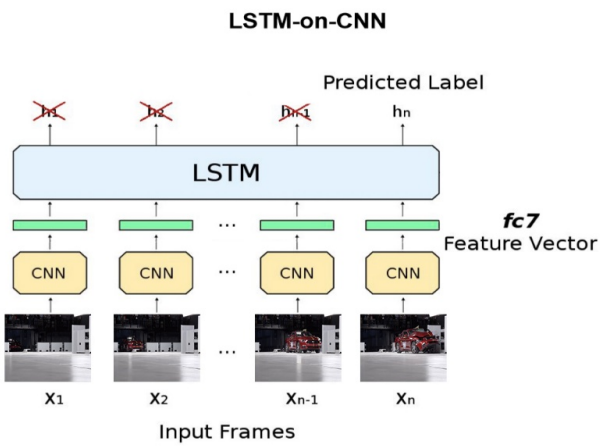

Fig. 5. Feeding Sequences of Frames to the LSTM Layer

The weights of the convolution layers of the Inception v3 were frozen before training and just the LSTM and fully connected layers were trained with our images. As mentioned earlier, these layers were followed by a sigmoid layer for the final classification.

VII. TRAINING THE MODEL

All the layers as shown in table 2 were trained using the Image Data Generator in keras which allows image augmentation. Image Augmentation is a process by which a set of pictures can be used to artificially expand the dataset. The most common features of image augmentation include zoom, pre-processing function, shear etc. These parameters basically apply random sheer, rotation, and zoom on the existing pictures thus giving the model more data to learn on. The values used for the proposed model are as follows:

Rescale=1/255,
shear_range=0.2,
zoom_range=0.2,
horizontal_flip=True

All training images were trained for 500 epochs. The results obtained were as follows:

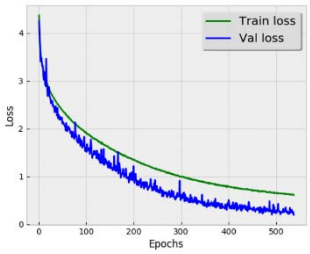

Fig. 6. Training loss and validation loss

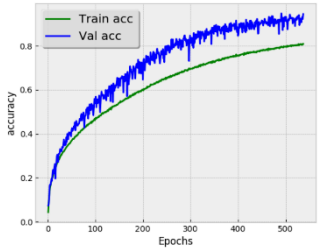

Fig. 7. Training accuracy and validation accuracy

Training Accuracy-0.95
Validation Accuracy-0.85

Training Loss-0.2568
Validation Loss-0.2894

TABLE 2: Layers of the propsed model

Type	**depth**	**#1x1**	**#3X3 reduce**	**#3X3**	**#5X5 reduce**	**#5X5**
convolution	1					
max pool	0					
convolution	2		64	192		
Max pool	0					
inception(3a)	2	64	96	128	16	32
inception (3b)	2	128	128	192	32	96
max pool	0					
inception(4a)	2	192	96	208	16	48
inception(4b)	2	160	112	224	24	64
inception(4c)	2	128	128	256	24	64
inception(4d)	2	112	144	288	32	64
inception(4e)	2	256	160	320	32	128
Max pool	0					
inception(5a)	2	256	160	320	32	128
inception(5b)	2	384	192	384	48	128
Avg pool	0					
Dropout (40%)	0					
Dense Layer (Linear)	1					
Bidirectional LSTM	256					
Dropout (20%)	0					
Bidirectional LSTM	256					
Dropout (20%)	0					
Dense Layer (Linear)	64					
Dense Layer (Linear)	32					

Sigmoid	1					

VIII. RESULTS

Once the system starts running it takes into account each frame of the video that it is capturing from the Pi-camera and runs it through the proposed model and when it detects an accident the system immediately sends a message using the GSM module. It also sends the frame at which it detected an accident and what percentage of accident it is. It also shows the time stamp as to when the accident was detected. Fig. 8. shows an accident frame along with the details.

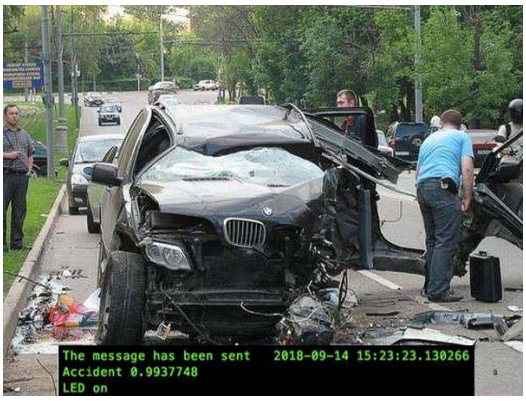

Fig. 8. Accident Detection on a Real Time Video

The below frame is of a Non-Accident situation where only the percent of No Accident is shown.

Fig. 9. Non-Accident Detection on an image

The performance of the model was evaluated using "Accuracy" metric as follows:

$$Accuracy = \frac{NumberofCorrectPrediction}{Totalofallcasestobepredicted}$$

The running model gave us an accuracy ranging from 82% to a maximum accuracy of 98.76%. On an average the model gave us an accuracy of more than 92.38%.

IX. CONCLUSION

Accidents are one of the most common problems that humanity faces on a daily basis, leading to loss of both life as well as property. The proposed system provides a very viable and effective solution to this problem. The proposed vehicle accident detection system can track an accident at its moment of occurrence and sends an instantaneous alert SMS regarding the accident to the nearby hospitals and police stations which includes details like timestamp and the geographical location. Unlike other systems in use, which consists of expensive sensors and unwanted hardware, the proposed system is much more cost effective and foolproof with a much-improved accuracy rate than its counterparts mainly due to a model-based approach. The experimentation, testing and validation has been carried out using images and the results show that higher sensitivity and accuracy is indeed achieved using this method, henceforth, making it a viable option for implementing this system in most of the state and national highways of the country. Thus, the project works towards a social cause and helps create a system which guarantees that no individual is left unattended or helpless in an unforeseen event of an accident, in turn, securing and maintaining the quality of life to the highest standards.

REFERENCES

[1] "Global status report on road safety 2015", World Health Organization, 2019. [Online]. Available: http://www.who.int/violence_injury_prevention/road_safety_status/2015/en/. [Accessed: 07- Mar- 2019].

[2] Prabakar, S., et al. "An enhanced accident detection and victim status indicating system: Prototype." India Conference (INDICON), 2012 Annual IEEE. IEEE, 2012.

[3] "Lexus Enform", *Lexus*, 2019. [Online]. Available: https://www.lexus.com/enform. [Accessed: 07- Mar- 2019].

[4] "OnStar Safety and Security Services", *Onstar.com*, 2019. [Online]. Available: https://www.onstar.com/us/en/services/safety-security/. [Accessed: 07- Mar- 2019].

[5] "SOSmart automatic car crash detection and notification app", *SOSmart automatic car crash detection app*, 2019. [Online]. Available: http://www.sosmartapp.com. [Accessed: 07- Mar- 2019].

[6] C. Kockan, "Communication between vehicles" PhD thesis, Istanbul Technical University, 2008

[7] Zeng, Yuanyuan, Deshi Li, and Athanasios V. Vasilakos. "Opportunistic fleets for road event detection in vehicular sensor networks." *Wireless Networks* 22.2 (2016): 503-521.

[8] Szegedy, Christian, et al. "Rethinking the inception architecture for computer vision." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[9] Szegedy, Christian, et al. "Going deeper with convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.