Vehicle Accident Detection using YOLOv3

A PROJECT REPORT

Submitted by

AYUSHI TIWARI - 18MIS0203
B. ANIKETH - 18MIS0094
C. HARSHAVARDHAN - 18MIS0259

Under the supervision of
LAKSHMIPRIYA G G Assistant
Professor
Department of Information Technology

VELLORE INSTITUTE OF TECHNOLOGY, VELLORE

Abstract

In this paper, a system is developed to automatically detecting accidents related to vehicles in image and videos. We divide these accidents in three cases, 1. Car collided with other cars or any object, 2. Car gets flipped, this case is considered as it is easy to recognise flipped cars and
3. Car on fire after collision or due to other internal reasons. We trained our system to detect such events Automatic Car crash detector can saves lives if it gets installed by CCTV surveillance organisation, as an accident would be immediately reported by the system and help will reach the reported sites. Overall this paper can serve as foundation of such system. This paper addresses the pros and cons of other such systems used. It also contains literature search, practical applications and results of our approach. Finally, we concluded and discuss limitations and future scope.

Contents

List of Figures

Chapter 1

INTRODUCTION

1.1 Problem Definition

Problem statement is: ”Detecting Car Crash through images or videos using convolutional neural network”.

1.2 Motivation

Everyday, hundreds of people die due to road accidents in India. Out of these fatalities, a large percentage is of unreported cases.

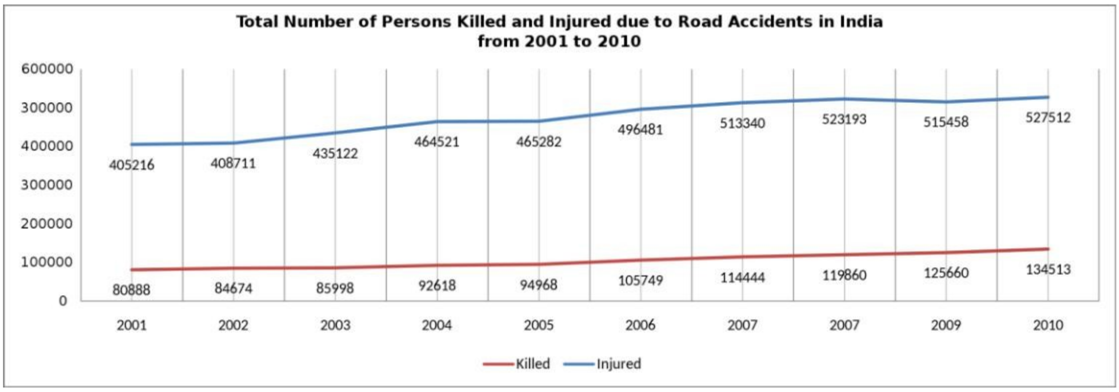

Figure 1.1: Fatalities due to road accidents 2000-2010 [Source: Wikipedia[1]] .

Accidents can be caused due to several reasons like over-speeding, drink and drive, not following traffic rules, car malfunction, etc. In 2015, there were about five lakhs road accidents in India, which killed about 1.5 lakhs people[2]. This motivated us to develop a system to reduce the traffic fatalities.

1.3 Goals

So, our goal is to develop a system which can detect such unfortunate events from CCTV videos or images and report it to respective organisations to take appropriate steps. So that injured persons do not die due to lack of timely treatment. This system is also a foundation for CCTV surveillance system with no human involvement

1.4 Approaches

Applying model based on vehicle detection and tracking leads to various inaccuracies. So, we switched to a new approach in which we train our system to directly recognize various scenarios of crashes using YOLOv3 model which is computationally less expensive and more accurate.

1.5 Problem Faced

We faced lot of problems before implementing our project successfully. Some are:

1. We had to switch from vehicle detection and tracking model to training and testing model as we experienced major inaccuracies.
2. Lack of internet available (as we were home quarantined) due to which we were not able to download dataset of car crashes instead we had to manually download images and label them for training.
3. Training session were very long because of lack of good GPUs, which overheats the laptop, so we switched to Google Colab which is GPU enabled.

Chapter 2

Literature Review

Means for detecting a vehicle accident can be seen in two ways. The one way is to detect by sensor and the other way is analyzing the camera images.

In the past twenty years, researchers have conducted many studies on vision-based traffic crash detection, which can be classified into three categories: (1) modeling of traffic flow patterns (2) modeling of vehicle interactions (3) analysis of vehicle activities .

The first method is to compare vehicle trajectories to typical vehicle motion patterns that can be learned from large data samples. In this framework, if a trajectory is not consistent with typical trajectory patterns, it can be considered as a traffic incident . However, it is not easy to identify whether this incident is a crash due to limited crash trajectory data that can be collected in the real world.

The second method determines crash occurrence based on speed change information, which applies social force model and intelligent driver model to model interactions among vehicles. This method requires a larger number of training samples.

The third method largely depends on trackers because it needs to continuously calculate vehicle motion features (e.g., distance, acceleration, direction etc.). As such, aberrant behaviors related to traffic incidents could be detected. However, it is often difficult to be utilized in practice, limited by high computational costs and unsatisfactory tracking performance in congested traffic environments .

In Another method, A GSM module is used to send the alert message and a GPS module is used to detect the location of the accident. The GPS and GSM module are interfaced to the control unit using serial communication .The accident is detected using two sensors- Micro Electro Mechanical System (MEMS) sensor and vibration sensor. MEMS sensor also helps in measuring the angle of roll over of the car. A 32-bit ARM controller is used as the main high speed data-processing unit. The vibrations are sent from the vibrating sensor to the controller after passing through an amplifying circuit. Similarly the roll over angle is sent from the MEMS sensor to the controller

Chapter 3

Proposed Methodology

We divide our dataset in three classes: 1. Collision, 2. Car-flipped,and 3. Car on fire. We used Darknet and YOLO in training and testing images. Our Project is divided into following steps:
1. Downloading the image using ”Download All Image” extension provided by Chrome web Store .
2. Selecting an image and label it according to its class using labelImg. Done for almost 300 images and prepare a dataset.
3. Training the dataset using weights for the convolutional layers of the YOLOv3 network(Darknet53.conv.74.weights). By using these weights it helps object detector to be way more accurate and not have to train as long.
4. Testing random images using darknet and note predictions for each image and find the overall efficiency.
Following are the descriptions of used mechanics to implement our model:

3.1 YOLOv3

YOLO (“You Only Look Once”) is an effective real-time object recognition algorithm. From the original paper: YOLO is refreshingly simple: see Figure 2.1. A single convolutional network simultaneously predicts multiple bounding boxes and class probabilities for those boxes. YOLO trains on full images and directly optimizes detection performance.[4]

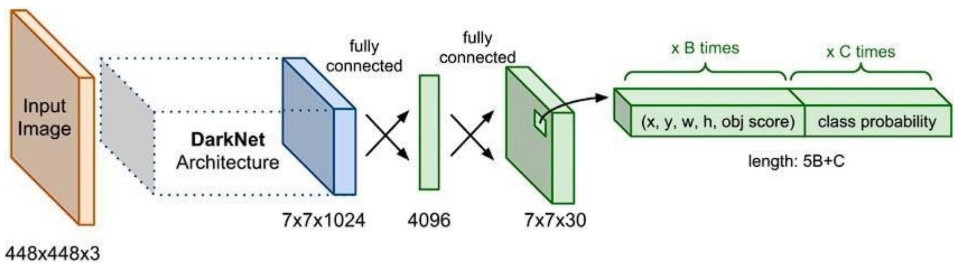

Figure 3.1: Working of YOLO

.

YOLOv3 came about April 2018, it included the fact that bounding boxes get predicted at different scales. The underlying meaty part of the network, Darknet, is expanded in this version to have 53 convolutions layers.

3.2 Darknet

Darknet is a framework to train neural networks, it is open source and written in C/CUDA and serves as the basis for YOLO. Darknet53.conv.74.weights contains the weights for the DarkNet network originally trained for classification on the ImageNet dataset, which is used as the pre-trained feature extractor (backbone) for YOLOv3.[5]

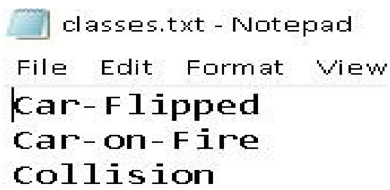

Figure 3.2: Classes

.

3.3 Dataset

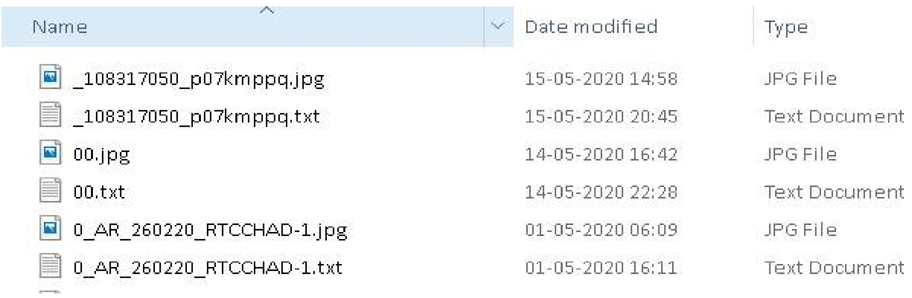

Figure 3.3: Dataset (image + text file)

.

Dataset is divided into three classes, 1. Car collided with other cars or any object, 2. Car gets flipped, this case is considered as it is easy to recognise flipped cars and 3. Car on fire after collision or due to other internal reasons as shown in figure 2.2. Almost 300 images are labeled with respect to their classes and written in a text file with its coordinates using labelImg as shown in figure 2.3. LabelImg is a graphical image annotation tool. It is written in Python and uses Qt for its graphical interface.

Chapter 4

Hardware and Software Used

We used Google colab to develop the system. Google Colab has following features:

1. CUDA version-10010
2. CUDA : 7.6.5
3. GPU : Tesla P100-PCIE-16 GB Other

Software requirements:

1. LabelImg
2. Open CV
3. Darknet53.conv weight
4. compiled Darknet 5. Python 3.7

Chapter 5

RESULTS AND DISCUSSION

A system is developed to automatically detect accidents related to vehicles in image and videos. Proposed method is implemented successfully. We trained our dataset for 4000 iterations for better results. Final outcomes are boundary boxes around susceptible region in image covering colliding cars, flipped cars and car on fire. Following are the results:

Figure 5.1: Crash site

.

Figure 5.2: Detecting flipped car with 90% prediction

.

Figure 5.3: Crash site

.

Figure 5.4: Detecting car-on-fire with 99% prediction

.

Figure 5.5: Crash site

.

Figure 5.6: Detecting Collision with 100% prediction

.

As shown above our model successfully detect the three types of crashes. But the accuracy drops significantly when the image is not of better quality or time is of night. Weather also influence output accuracy. It drops in rainy and winter season when image is not clear to detect such scenarios. It also does not deal with cases other than above listed cases.

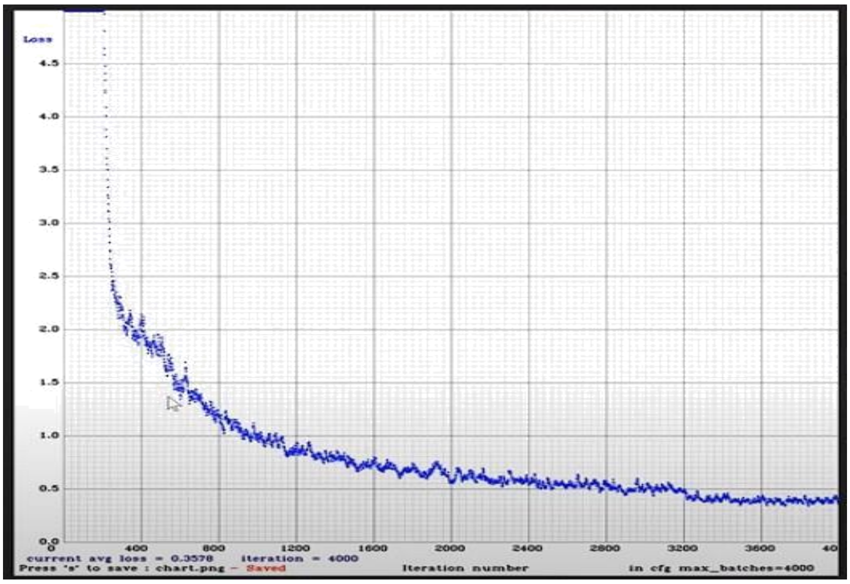

Figure 5.7: Train loss vs Iteration curve

.

After 3600 iterations, the model became convergent. Figure 4.7 provides the train loss vs iteration curve. Loss should be less than 0.5 for better accuracy.

Chapter 6

Conclusion

A deep learning model (i.e., Yolo v3) is trained to detect accident or for crash detection, considering various crash scenarios (collision, car-on-fir, car-flipped).

Overall, the results are encouraging and the framework is promising. Admittedly, there are still some issues that can be further addressed. First, different image enhancement methods could be tried to improve the overall performance. Second, other deep learning method can be used and compared to original Yolo v3 model. Third, other more complex deep learning structure can be examined and compared to the current framework, in terms of accuracy and computational speed.

Bibliography

[1] Traffic Accidents in India, Wikipedia , https://en.wikipedia.org/wiki/Traffic acciden- ts in India

[2] Overview of Road Accidents in India, prsindia https://www.prsindia.org/policy/vital-stats/overview-road-accidents-india

[3] You Only Look Once (CVPR 2016. OpenCV People's Choice Award) https://arxiv.org/pdf/1506.02640v5.pdf

[4] Martina Pugliese, A very shallow overview of YOLO and Darknet https://martinapugliese.github.io/recognise-objects-yolo/

[5] https://github.com/eriklindernoren/PyTorch-YOLOv3/issues/201