F-039

Implementing an Automatic and Instant Road Accident Report System Using Knowledge Systems

Helton Agbewonou YAWOVI † Tadachika OZONO † Toramatsu SHINTANI †

1. Introduction

Nowadays, accidents constantly increase in the world. After an accident, the police and insurance companies have to make a report to get circumstances of the event and determine the responsibilities of each actor (who was wrong and who was right). However, this task is time-consuming and, therefore, support systems are requested. Existing researches [1] have been focused on support systems that help actors within non mortal accidents to fulfill the report and draw the sketch by dragging elements on the interface. These approaches still are time consuming and do not help to know automatically the responsibilities of each actor. This paper proposes an automatic and instant road accident report system based on machine learning and knowledge systems to generate accident reports and predict the responsibility of each actor in the accident.

2. Automatic and instant road accident report system

The usage of dash cam (driving recorder), now becomes common for cars owners; therefore, we want to use it as the data source of our system. In case of accident, the dashcam of the car send, via internet, the recorded video of the accident to our system hosted by an online server.

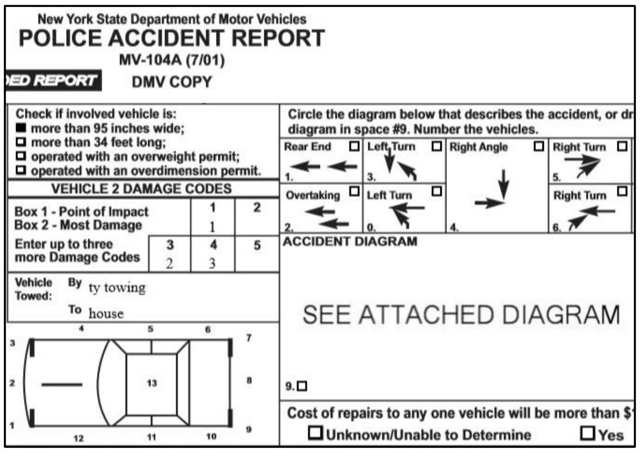
Our system analyzes the video and regarding the situation shown on the video must predict the responsibilities of each actor and establish automatically a printable report for police including a sketch of the accident. This printable report has to respect the police format, therefore, must content some important information such as the date and location of the accident, details on drivers and vehicles (name, address, phone number, driver's license number and date of birth, license plate number…) and details about passengers and witnesses as shown as in Figure 1.

To realize a complete report including these important information mentioned above, our system is composed of two parts. The first part inside the vehicles (an IoT system), that includes speed sensors, GPS chip, accelerometer, gyroscope, shock sensors inside cars and a SIM card to have access to the internet. It detects the shock of the car when an accident happened, get the video from the dashcam and send data to the second part. The second part is located on an online server that processes the information sent by the IoT part and make the automatic report. Figure 2 shows the structure of the system and the interaction between the IoT part and the online server.

In this paper we focus on the second part assuming that the IoT part already exist and efficiently send all required data (such as recorded video of the accidents, sensors information, locations of the accidents, name of the vehicles' owners,…)

3. Design of the system

The system is an expert system that use 1) object detection to



New York State Department of Motor Vehicles
POLICE ACCIDENT REPORT
MV-104A (7/01)
ED REPORT DMV COPY

Check if involved vehicle is:
■ more than 95 inches wide;
☐ more than 34 feet long;
☐ operated with an overweight permit;
☐ operated with an overdimension permit.

VEHICLE 2 DAMAGE CODES

Box 1 - Point of Impact Box 2 - Most Damage		1 1	2
Enter up to three more Damage Codes	3 2	4 3	5

Vehicle Towed: By ty towing
To house

Circle the diagram below that describes the accident, or dr
diagram in space #9. Number the vehicles.

1. Rear End ☐
2. Overtaking ☐
3. Left Turn ☐
0. Left Turn ☐
4. Right Angle ☐
5. Right Turn ☐
6. Right Turn ☐

ACCIDENT DIAGRAM

SEE ATTACHED DIAGRAM

9.☐

Cost of repairs to any one vehicle will be more than $
☐Unknown/Unable to Determine ☐Yes

***Figure 1**: Example of an accident report*

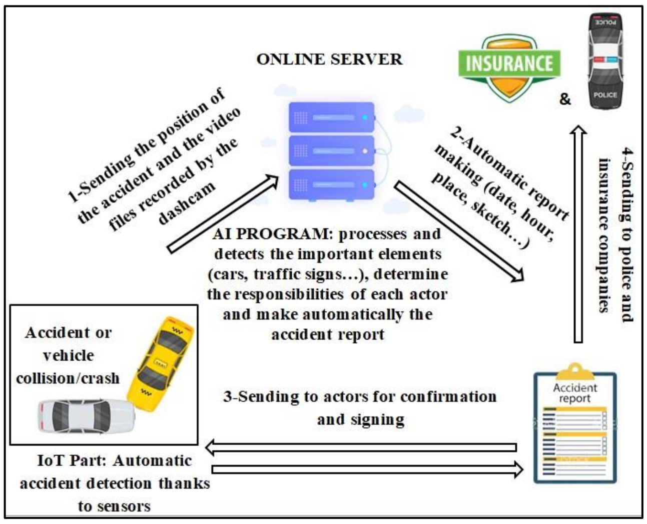

ONLINE SERVER
INSURANCE
&
POLICE
1-Sending the position of the accident and the video files recorded by the dashcam
2-Automatic report making (date, hour, place, sketch…)
4-Sending to police and insurance companies
AI PROGRAM: processes and detects the important elements (cars, traffic signs…), determine the responsibilities of each actor and make automatically the accident report
Accident or vehicle collision/crash
3-Sending to actors for confirmation and signing
Accident report
IoT Part: Automatic accident detection thanks to sensors

***Figure 2:** The System Structure*

detect, in the video important elements for making an accident report, such as traffic signs, cars, road limits, pedestrians…; 2) inference engine and a knowledge base of defined rules to know if the driver is responsible or not and 3) a template of an accident report to generate a report for the corresponding accident.

Generally, there are two primary actors or even one when an accident occurs (the other actors are secondary). So the driver using our system (specifically the IoT part) can have only one of these three statuses: **Responsible, Half Responsible** or **Not Responsible**. The output of the expert algorithm of our system is the degree of negligence (see section 3.2 for more details) and either -1, 0 or 1. -1 set for the driver is not responsible (so the other primary actor is the responsible), 0 set for the driver is half responsible (the responsibilities are shared) and 1 set for the driver is responsible of the accident (so did not respect the traffic rules).

† Department of Computer Science, Graduate School of Engineering - Nagoya Institute of Technology

3.1 Object detection task

This section describes how to implement an object detection system for road information. For the object detection task, we use the object detector YOLOv3 and the library Open CV (The duo is fast and accurate). The program has to detect the traffic signs and their type or current state and not only the class they belong to (Ex: the system must detect a *50 km speed limit sign* and not only a *limit sign* or a *green traffic light* and not only a *traffic light*), road limits, the collided object (it can be a tree, a car…).

Therefore, we use a YOLO custom trained model with a set of thousands of images of green traffic lights, red traffic lights, yellow traffic lights, 20/40/…/130km speed limit signs, crosswalks, road limits and so on. We followed the steps bellow, including data annotation, configuration and training, to train YOLO and get our custom model [2]:

Data annotation: We use the **BBox Label Tool** (a Python library) to annotate the training images (a set of about 300 different images per category: green traffic lights, red traffic lights, crosswalks…). It allows us to easily generate the training data in the correct format YOLO requires.

YOLO configuration files: YOLO needs certain specific files to know how and what to train. We created these three required files: **`cfg/obj.data`**, **`cfg/obj.names`**, **`cfg/yolo-obj.cfg`**. The file **`obj.data`** contains the information about the number of classes we are training, what the train and validation set files are and what file contains the names for the categories we want to detect. The file **`obj.names`** simply contains the name of the categories. Every new category should be on a new line. The file **`yolo-obj.cfg`** is just the duplication of the original **`yolov3-openimages.cfg`** [3] that comes with the code of YOLO.

Training YOLO: To start training, YOLO requires a set of convolutional weights. So we downloaded, from the official YOLO website, a set of convolutional weights that was pre-trained on **ImageNet**. This set of convolutional weights file (**`conv.23`** file) provides an excellent starting point and help us to finally train YOLO.

After the training, we now have a **`.weights`** file that represents our trained model. This trained model file is our detector file that our system uses to detect the custom objects in the videos and these detected objects are the input source for the inference engine and knowledge based task.

3.2 Inference engine and knowledge based task

For the inference engine and knowledge based task we use a set of defined rules to determine either the driver is responsible or not. In road rules there are some basic methods and principles. One of the methods to determine the fault of actors after a road accident occurs is called the **degree of negligence (or percentage of fault)**. Our system determines the responsibility by using this method. According to "**legalmatch.com**", **"*Basically, negligence means that the responsible party acted in a way that disregarded their duty to drive safely on the road, resulting in injury to the plaintiff*** [4]***"***. If the driver has more than 50% of degree of negligence, the program returns the result **Responsible**. If it is less than 50%, the program returns the result **Not Responsible** and if it equals to 50% returns the result **Half Responsible.** These results are output after a couple of derivations within our inference engine using a forward chaining (reasoning) of a set of rules. An example of the rules can be: "***If the pedestrian starts crossing in yellow and the driver enter in red, the degree of negligence for the driver is 90%.***"[5]

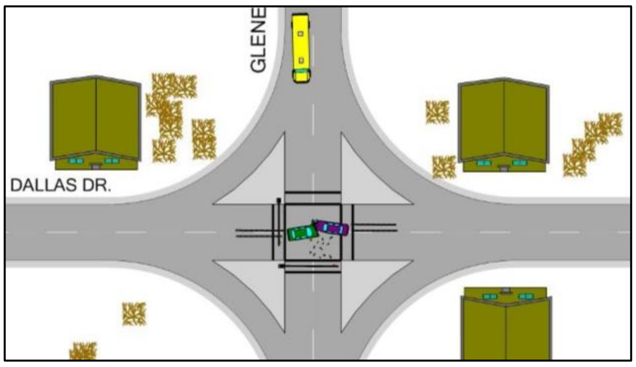


Figure 3*: A sample of a sketch of a road accident*

Finally after detecting objects in video and using the inference engine to predict the responsible of the accident, the system generates a printable report for police and insurance companies. The report must include all important information surrounding the accident including a sketch as shown as in Figure 3 that is a sample of sketch of an accident that occurred on a crossroad. Our system uses information sent by the IoT part such as 1) the accelerometer and gyroscope to circle automatically the checkboxes that describe the circumstances at the moment of the shock (Figure 1) 2) the geolocation information of the GPS chip and a digital map to have a 2D representation of the place of the accident and draws automatically the sketch.

4. Conclusion

After traffic accidents, the police and insurance companies have to make reports to determine who was wrong and who was right. This task being time-consuming and requiring experts' knowledge intervention required support systems. We realize an expert system to help in making a decision on the responsibilities of each actor of the accident and generate an automatic and instant road accident report. The system consists of three functions: (a) detecting objects, e.g. traffic signs, cars, in video taken by driving recorder using YOLO and Open CV, (b) determining who is wrong and who is right based on the road rules using an inference engine and knowledge based of defined rules, and (c) generating the accident report with necessary information including the sketch of the accident.

Acknowledgement

This work was supported in part by JSPS KAKENHI Grant Nu mber JP16K00420, 19K12097, 19K12266.

References

[1] Habib M. Fardoun, Daniyal M. Alghazzawi & Antonio Paules Cipres, "Improving User-Insurance Communication on Accident Reports", Page 1-6 (2014).
[2] https://timebutt.github.io/static/how-to-train-yolov2-to-detect-custom-objects/
[3] https://pjreddie.com/media/files/yolov3-openimages.weights
[4] http://www.jiko-online.com/jiho1.htm
[5] https://www.legalmatch.com/law-library/article/negligence-in-a-car-accident-lawsuit.html