

Developing a Road Accident Simulator for Automatic Road Accident Report System

Helton Agbewonou YAWOVI † Tadachika OZONO † Toramatsu SHINTANI †

Abstract: Nowadays, road accidents represent a big challenge for every country in the world. Lot of researches in AI, Machine Learning and Deep Learning are conducting every year to find some solutions to reduce road accidents through predictions by using previous data of vehicles crashes. In Our research, we focused on a system that can automatically build a 3D simulation (for visualization purpose) of an accident given in input the manually made accident report. Our simulator, then, automatically generates labeled training data that will be used by the system for image recognition task to predict the responsibilities of each actor in the accident. The simulator, also, generates a sketch of the accident to append to the manually made accident report inputted into the system by the user.

1. Introduction

Since the apparition of vehicles, road accidents are a serious issue that all countries in the world are facing and challenging. After an accident, the investigation to know the circumstances around the incident is very important to determine the responsibilities of each actor (who was wrong and who was right). This task sometimes can be 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. This paper proposes an approach of simulating in 3D a given accident taking in input its complete accident report. The system then, after enough training, learns to make an automatic and instant road accident report based on machine learning and knowledge systems and predict the responsibility of each actor in an accident.

2. Automatic road accidents' 3D simulation system

Our system is a web based application composed of a 3D simulation system for road accidents that generates the simulation of an accident based on the inputted accident report in the system. It takes a full completed accident report, analyzes it to know which kind of road accident it is and simulate that accident in his 3D view. After the simulation, the system generates the sketch of the accident that is attached to the accident report and train a model that will be used to predict the responsibilities of each actor and export again a new accident report that includes the generated sketch of the accident. Also, after the exportation of the new accident report with the sketch included, the system exports the video of the crash as a recorded video by a dash cam (driving recorder). That video is then stored as the crash video of the accident to be used later during the training process of the model for the AI part to make the decision about the responsibilities.

The inputted report of the system has to respect a general police format, therefore, must contain 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 2.

Our road accident simulator has two purposes: 1) to generate training data for an image recognition system and 2) to make sketches of road accidents. To build the image recognition, we need to collect training data, and the simulator automatically

generates labeled training data. Moreover, the simulator makes a sketch for an accident report when users manually make the report. Furthermore, they can refer to the simulation or its movies to consider accidents.

In this paper we describe a) the process of inputting these important information in the system via the submission of the completed accident report with a possible selection of the circumstances of the accident via check-boxes; b) the simulation of the accident in the 3D view; c) the exporting of the sketch of the accident with exportation of the video of the simulation as the dash cam crash video; d) The model training and the inference engine for predicting the responsibilities of each actors of the accident.

3. Previous work

In a previous work [2], we based our system on a) the driver recorder of the vehicles (that is the source of the video to train the model for the image recognition task), b) an IoT part inside the vehicles that includes speed sensors, GPS chip, accelerometer, gyroscope, shock sensors and detects the shock of the car when an accident happened, then gets the video from the driver recorder and send all the data to c) the AI part that processes the information sent by the IoT part and generates automatically the associated report and predict the responsibilities of each actors using a knowledge system. Figure 1 shows the previous structure of the system.

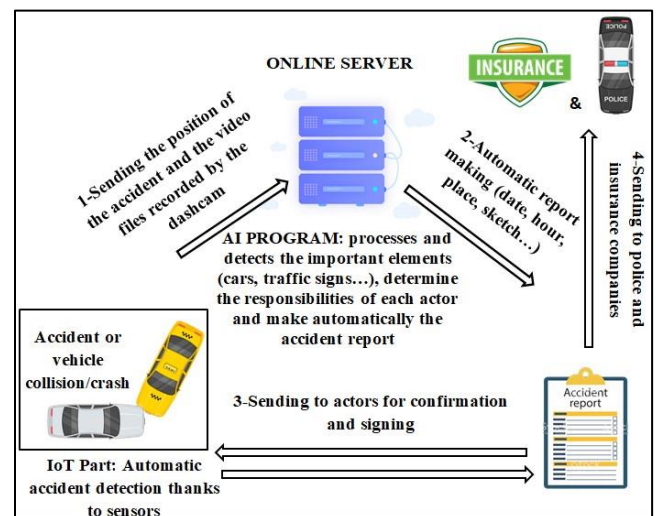


Figure 1: The previous system structure

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

4. Design of the system

Generally, when an accident occurs, there are two primary actors or even one sometimes (the other actors are secondary). So in an accident report, there are usually 2 parties. So one party in the accident 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 3.4 for more details) and either “**unknown**”, “**-1**”, “**0**” or “**1**”. “**unknown**” is set for unknown result (the responsibilities cannot be determined by the system), “**-1**” set for the first primary driver is not responsible (so the other party is the responsible), “**0**” set for the first primary driver is half responsible (the responsibilities are shared between the actors) and “**1**” set for the first primary driver is responsible of the accident (so did not respect the traffic rules).

To achieve that, our system is essentially composed of 3 main parts: a) A form that contents a template of the European accident statement; b) a 3D view for the simulation that will export the crash video and c) the output panel that will show the new exported report with the sketch of the accident and the prediction about the responsibilities of the actors.

4.1 The inputted accident report

Accident reports vary from region to region. So a standard version of an accident report does not exist. But in this paper we will use the European accident statement format to input data in our system. To use the system, the user, first needs to fill the form of the accident and describe the circumstances of the accident by checking some check-boxes and press the “**Simulate now**” button.

4.2 The simulation by the 3D simulator

To make our 3D simulator available via browser, we implemented it using Blender 3D (a free and open-source 3D computer graphics software) and Verge 3D (a real-time renderer of 3D on websites). With Blender 3D, we created (and in some cases imported free models from internet) and animated all objects (vehicles, buildings, roads, traffic signs...) in the 3D world. We animated different kinds of accidents with different scenario and export all these objects into web readable files using Verge 3D.

To make a simulation with our system, the user has to fill the form of the accident report located at the left side of the simulator (as shown in Figure 3) by inputting some important information of the accident 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. Then, he has to select the circumstances of the accident by checking the check-boxes located in the middle of the form. By checking these check-boxes, the system knows and selects automatically the type of the corresponding road accident in the list of the accidents located in the right side of the simulator (as shown in Figure 3). We also make available the possibility for the user to select the vehicles he wants to be involved in the simulation. After the selection of those elements, the user simply presses the button “**Simulate now**” to simulate the accident and visualize it in the 3D view. While simulating the accident, the system generates the video of the crash and the labeled training data for the image recognition task to predict the responsibility of each actor in the accident. Also, after the simulation, it generates the sketch image and put it in the “**Sketch of the accident**” section in the accident report.

The image shows a sample of the European accident report template, titled "CONSTAT AMIABLE D'ACCIDENT AUTOMOBILE ACCIDENT REPORT". The form is divided into several sections:

- 1. Date de l'accident / Date of the accident:** Includes fields for date, time, location, and locality.
- 2. Dégâts matériels à des véhicules autres que A et B / Property damage:** Fields for damage to other vehicles, with checkboxes for "yes" or "no".
- 3. Véhicule A / VEHICLE A:** Fields for name, address, postal code, and country.
- 4. Véhicule B / VEHICLE B:** Similar fields to section 3.
- 12. CIRCONSTANCES / CIRCUMSTANCES:** A central section with multiple checkboxes for describing the accident circumstances, such as "stationnement à l'arrêt", "sorti d'un parking", "emerging from a parking", etc.
- 13. Croquis de l'accident au moment du choc / SKETCH OF ACCIDENT:** A section for drawing the accident scene, with a "Sketch of the accident" label overlaid on the form.
- 14. Dégâts apparents au véhicule A / Visible damage to vehicle A:** Fields for describing damage to vehicle A.
- 15. Dégâts apparents au véhicule B / Visible damage to vehicle B:** Fields for describing damage to vehicle B.
- 16. Mes observations / My remarks:** A text area for additional notes.
- 17. Signature des conducteurs / Signatures of the drivers:** Fields for driver signatures.

Figure 2: A sample of the European accident report template

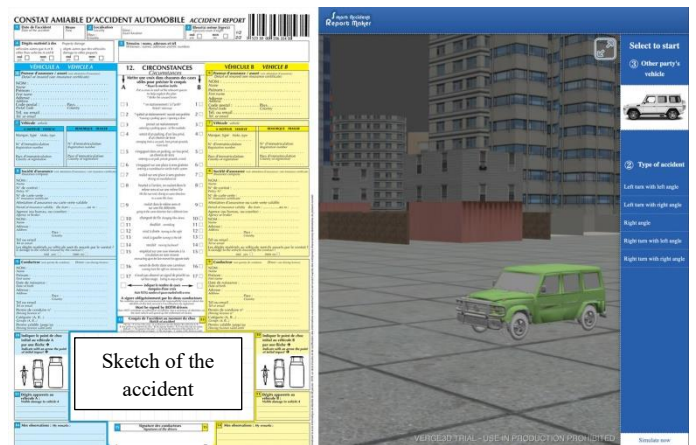


Figure 3: A screenshot of our simulator taking in input an accident report

4.3 The data training

In this section, we describe how to implement the object detection system for road information in our system. 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 images of green/red/yellow traffic lights, 20/130km speed limit signs, crosswalks, road limits,... generated by the 3D simulator after a simulation. We followed the steps bellow, including data annotation, configuration and training, to train YOLO and get our custom model [3]:

Data annotation: We use the **BBox Label Tool** (a Python library) to annotate the training images (a set of 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` [4] 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 generated videos and these detected objects are the input source for the inference engine and knowledge based task.

4.4 Inference engine and knowledge based task

For the inference engine and knowledge based task of our system, we use a set of defined rules to determine either the first party of the accident 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*" [5].

If the first party 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 outputted after a couple of derivations within our inference engine using a forward chaining (reasoning) of a set of rules. An example (as shown as in Figure 4 bellow) of the rules can be: "*If the pedestrian (the other party in the accident) starts crossing in yellow and the driver (the first party in the accident) enter in red, the degree of negligence for the driver is 90%.*" [6]

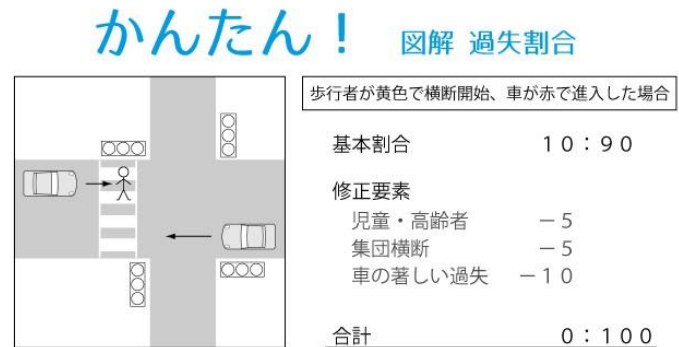


Figure 4: An example of degree of negligence determination based on the traffic rules in Japan. From *jiko-online.com*

Finally after detecting objects in the generated video, the system uses the inference engine to predict the responsible of the accident.

5. Conclusion

After traffic accidents, the police and insurance companies have to make reports to determine the responsibilities of each actor. This task being time-consuming and requiring experts' knowledge intervention required support systems. We realized an expert system to help in making a decision on the responsibilities of each actor of the accident and generate a sketch of the accident giving as input a completed accident report after simulating the crash in the 3D simulator of the system.

Acknowledgement

This work was supported in part by JSPS KAKENHI Grant Number 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] YAWOVI Agbewonou Helton, OZONO Tadachika, SHINTANI Toramatsu: Implementing an automatic and instant road accident report system using Knowledge System, FIT2019, Vol. 2, F-039, pp. 353--354.
- [3] <https://timebutt.github.io/static/how-to-train-yolov2-to-detect-custom-objects/>
- [4] <https://pjreddie.com/media/files/yolov3-openimages.weights>
- [5] <http://www.jiko-online.com/jiho1.htm>
- [6] <https://www.legalmatch.com/law-library/article/negligence-in-a-car-accident-lawsuit.html>