Implementing an Automatic Road Accident Report System with an Accident Simulator

Helton Agbewonou YAWOVI † Tadachika OZONO † Toramatsu SHINTANI †

Abstract

The sad side effect of the increasing number of motorized vehicles is the increasing number of road accidents. Road accidents, nowadays, represent a big challenge for all countries in the world. Every year, people conduct lot of researches in AI, Machine Learning and Deep Learning to find efficient solutions to reduce road accidents through predictions thanks to the usage of data from previously occurred road accidents. After an accident occurred, police have to make investigation to know the circumstances of the incident and determine the responsibilities of each actor. To help police to go faster and quicker in these tasks, support systems are requested. Our goal is to create a system that can learn and make work easier and quicker for police and improve the traditional and manual way to determine responsibilities when road accidents occur. In our research, we focused on a system that can automatically build, thanks to a simulator, a 3D simulation (for visualization purpose) of an accident, given as input a manually made accident report. Our simulator, then, automatically generates labeled training data that will be used, later, by the system for image recognition task to predict the responsibilities of each actor in the accident using a custom trained YOLO model and the library Open CV. As a final result, the simulator 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. Researches in AI, Machine learning and IoT are conducted day by day in universities and in vehicles manufacturing companies to find ways to reduce crashes by including more safety technologies in cars and by helping rescuers to save more lives by saving accidents crash data that can be used for future predictions. When an accident occurs, there is for sure a human error. And after an accident, the investigation to know the circumstances around the incident is very important for police and insurance companies 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 accident report and draw the sketch by dragging some graphic 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.

The starting point of our system is, then, a road accident report and the output is the same road accident report but with added values such as the automatically generated sketch of the accident and the prediction of the responsibilities of each actor in the accident.

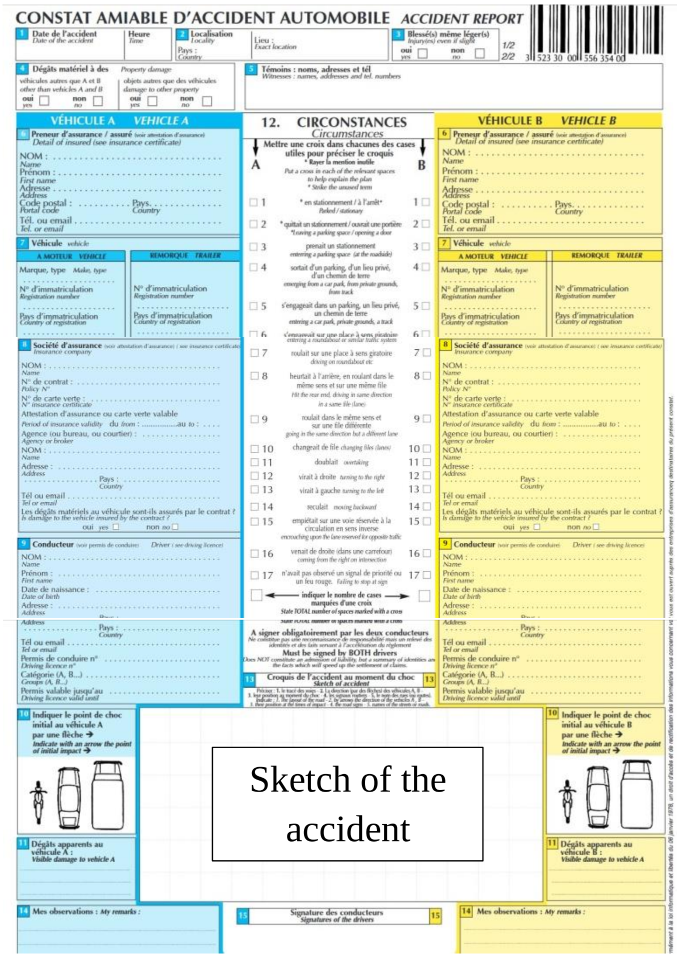

Figure 1: *A sample of the European accident report template*

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 the 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 of the actors.

The road accident report inputted by the user in 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 described in section four (4.1 The inputted accident report) and shown in **Figure 1**.

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 implement the image recognition feature, we need to collect training data, and the simulator automatically generates labeled training data for that purpose. Moreover, the simulator makes the sketch for the accident report the users manually made and inputted in the system. 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 2** shows the previous structure of the system.

In the previous structure of our system, the input data was not an accident report previously made by the user but the information of the sensors that the IoT part sends to the AI part on the online server including the recorded video of the crash recorded by the dashcam (driving recorder inside the vehicle). The difficulty in such architecture was the fact that if one of both of the cars included in the accident does not have the IoT part, it will be difficult to determine all the circumstances of the crash and the prediction of the responsibilities cannot be accurate and reliable.

In the new structure of our system, we replaced the IoT part with the manually made report that the user has to input to start the simulation process and have the needed output.

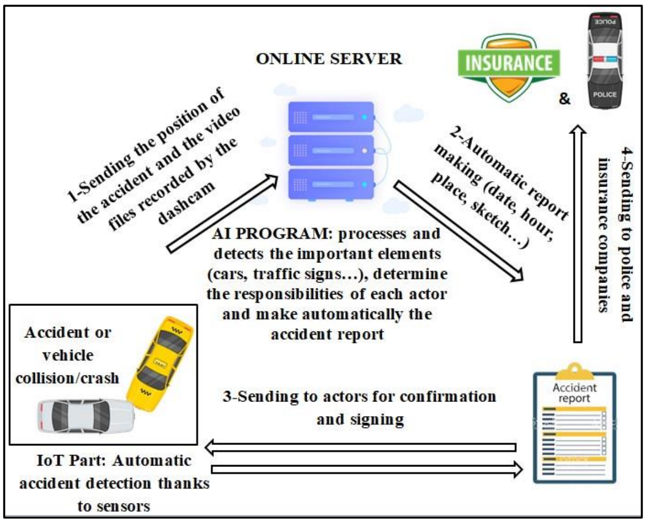

Figure 2: *The previous system structure*

4. Design of the system

Generally, when an accident occurs, there are only two primary actors or even one sometimes (the other implicated actors are secondary). So in an accident report, there are usually two parties. So one party in the accident can have only one of these three statuses: **Responsible**, **Half Responsible** or **Not Responsible**.

To be able to predict the responsibilities of actors in the accident, our system is composed of a knowledge system that helps to output the corresponding status of each actor. 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**". The output "**unknown**" is set for unknown result. In this case, even after multiple deductions from the knowledge system, the responsibilities of the actors cannot be determined by the system. The output "**-1**" set for the first primary driver is not responsible. So here, the other party or actor in the accident is the responsible. The output "**0**" set for the first primary driver is half responsible. Here, the responsibilities are shared between the actors of the accident. And the output "**1**" set for the first primary driver is responsible of the accident and did not respect the traffic rules.

To achieve that, our system is essentially composed of three (3) main parts: a) a form that contents a template of an accident report that respect the European accident statement format; b) a

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

3D view for the simulation of the accident and which will export the crash video as a dashcam recorded video and c) the output panel that will show the newly exported report with the sketch of the accident and the prediction about the responsibilities of the actors of the accident.

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 in the form. Then he has the choice of selecting the vehicles that will be used for the 3D simulation and press the "**Simulate now**" button to start the simulation. The inputted report respecting the European accident statement format is composed of two parts, each part for each actor of the accident. Each part is composed of sections to input the environment of the accident (date, time, location, property damage, injuries…), the vehicle's information (brand, type, registration number…), the driver information (name, address, driving license information,…), witnesses information (names, addresses, phone numbers), the insurance company's information (name, insurance certificate number, policy number, period of insurance validity, address, email, phone number,…), the impact information (visible damage parts,…).

All this information has to be inputted by the user in the accident report form of the system and the circumstances of the accident has to be checked from the check-boxes.

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 analyzes and selects automatically the type of the corresponding road accident in the list of the existing animated accidents in the system. 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.

After the simulation, the system generates the sketch image and put it in the "**Sketch of the accident**" section in the accident report.

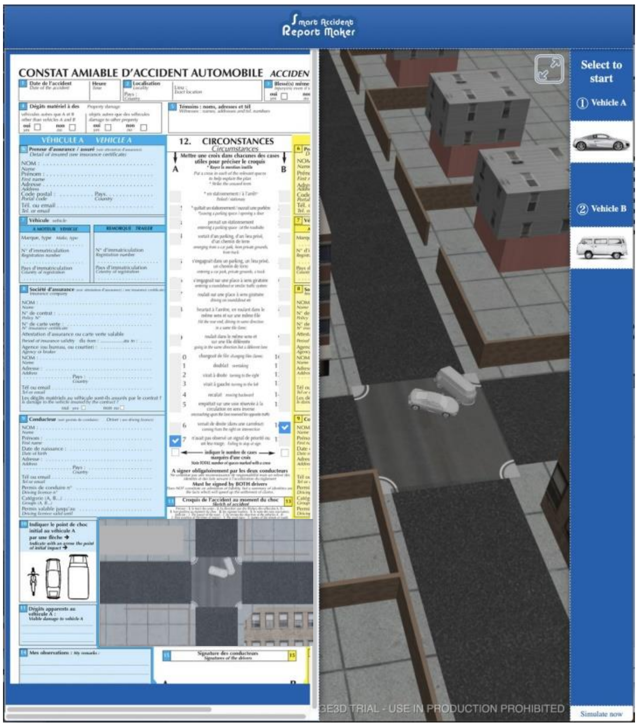

Figure 3: *A screenshot of our simulator taking in input an accident report*

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: The first step to get our custom YOLO model is the data annotation. To realize that, 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 and make possible and easy the recognition.

YOLO configuration files: YOLO needs certain specific files to know how and what to train. We created these three required YOLO configuration 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 for the responsibility prediction.

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]

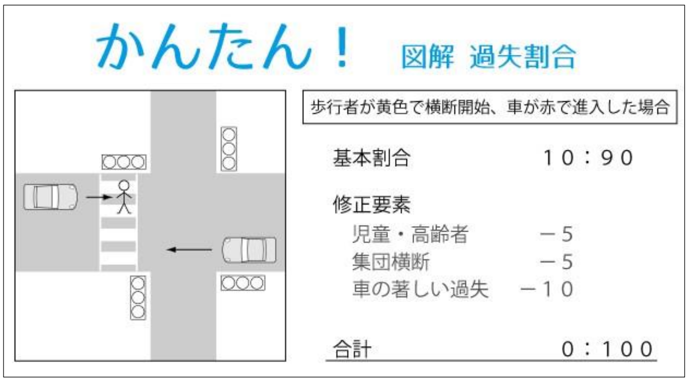

Figure 4: *An example of degree of negligence determination based on the traffic rules in Japan. From jiko-online.com*

Finally after detecting objects in the generated video, the system uses the inference engine to predict the responsible of the accident. Police and insurance companies can, then, use this data of prediction to make necessary documents.

5. Conclusion

After a traffic accident, the police have to make a report to determine the responsibilities of each actor during the incident so that insurance companies can pay the damage to victims. This task being time-consuming and requiring experts' knowledge intervention required support systems. We realized a system composed of an expert system (with image recognition and knowledge system to predict and help in the making a decision on the responsibilities of each actor of the accident) and a 3D simulator to visualize the accident and generate the sketch of the accident giving as input a complete manually made accident report that will be export as an output after the 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

