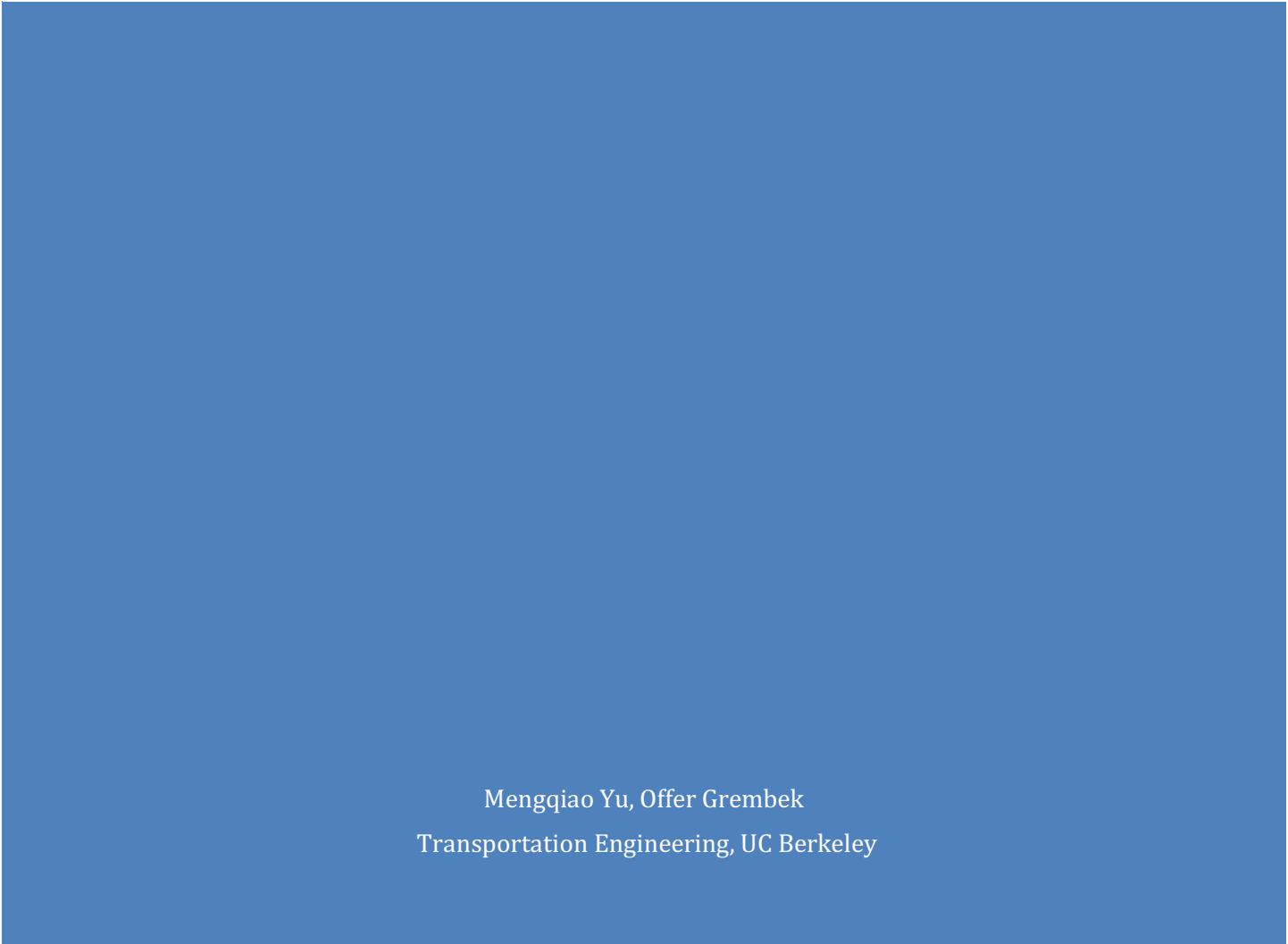


DATA PROCESSING & COLLISION ANALYSIS SYSTEM FOR AV CRASH REPORTS

(CSCRS FELLOWSHIP FINAL REPORT)



Mengqiao Yu, Offer Grembek
Transportation Engineering, UC Berkeley

Table of Contents

1. Introduction	2
2. Dataset	2
3. System Design	3
4. Part I: Data Collection and Processing System	4
4.1 Web crawler.....	4
4.2 Convolution.....	5
4.3 OCR	6
5. Part II: Text Analysis System.....	7
5.1 Reports after April 1, 2018.....	7
5.2 Reports before April 1, 2018.....	8
6. Part III: Collision Analysis System	9
6.1 Summary of crash data	9
6.2 Comparison	10
7. Summary.....	12
Acknowledgement	12
References	13

1. Introduction

Autonomous vehicles (AVs) offer a radically different path to a transport-efficient, crash-free city. Fully autonomous vehicles, relying on precomputed street maps and on-board sensors, is expected to eliminate traffic fatalities by up to 94 percent of crashes involving human error, based on the estimates from US Department of Transportation (DoT) researchers [1]. Such a claim is very attractive, and it naturally raise a critical concern: how safe are autonomous vehicles in current stage?

To date, a large amount of research has focused on safety in automation technology, including the impact of autonomous vehicles on traffic safety and congestion [2], legal and regulatory studies associated with AV implementation and emergency scenarios [3], cyber and communication security [4], etc. However, limited published research tried to address the concern: what's the capabilities of autonomous vehicles under field test? What type of collisions will the AVs be involved? Which part needs further improvement? The objective of this study is to answer these questions and provide a preliminary analysis of real-world crashes involving autonomous vehicles.

The real-world crashes dataset is public on the website of California Department of Motor Vehicles (CA DMV). Based on California regulations, CA DMV allows the manufacturers with testing permit to test their AVs with a safety driver on public road since September 16, 2014; later, it further approves testing without a human driver since April 2, 2018. In addition, it requires manufacturers to provide DMV with a report of traffic collision within 10 business days of the incident. The increasing number of field tests and public crash reports provide us an opportunity of analyzing current performance of AVs and identifying their limitations. This study will build an end-to-end system to analyze AV crashes, and it is anticipated to help the public, the manufacturers, policy makers and traffic planners to better understand the safety issues in autonomous vehicle and make further improvement.

2. Dataset

The CA DMV mandates that all manufacturers testing AVs on public roads file two different types of reports: 1) a Report of Traffic Collision Involving an AV within 10 days after the collision; 2) an annual report summarizing the disengagements (a failure of technology that causes the control of the vehicle to switch from the self-driving mode to the safety driver) [5]. Please note that disengagements are not collisions, and the information provided in these two types of reports are quite different. All latest reports can be found in the links below respectively, https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/autonomousveh_0l316, https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/disengagement_report_2017. The collision reports started from Oct 14, 2014. The engagement annual reports range from 2015 to 2017 while the 2018 report hasn't been released yet.

In this study, we only focus on the collision reports, and will explore disengagement reports in the future. To date (December 11, 2018), the CA DMV has received 129 autonomous vehicle crash reports. Reports before April 1, 2018 contains two pages while those after April 1, 2018 have three pages. The additional contents add more details about the collision from two perspectives:

- (1) Damage of the involving autonomous vehicles. In the new format (reports after April 1, 2018), it's required to identify the severity of the damage and specific damage area in the first page of the report.
- (2) A summary table of the critical information about the collision in the third page of the report, including testing environment condition (weather, lighting, roadway surface, etc.), type of collision (rear-end, side-swipe, etc.), movements of vehicles (stopped, proceeding straight, etc.).

Both the old and new formats of the collision reports are scanned pdf with relatively low resolution, which indicates a challenge of data (texts and information in the tables) extraction. Manual extraction can be a short-term but not sustainable solution. Acrobat embedded conversion function can be another feasible solution but with three drawbacks: (1) Acrobat is not good at converting tables especially tables with special characters; (2) The conversion quality of texts/paragraphs is not as good as other special techniques, and we will compare the quality in the later section; (3) We need to process each report one by one. Therefore, the first part of this study is to build a data processing system that can automatically obtain all essential data from different parts of the report. The second part of the study, AV collision analysis, will rely on the processed data and comparison with the conventional collision dataset.

The conventional collision dataset comes from Transportation Injury Mapping System (TIMS), <https://tims.berkeley.edu/>. TIMS, developed by SafeTREC, provides free access to California crash data, the Statewide Integrated Traffic Records System (SWITRS). We can easily query and summarize key information about the collisions that happened from 2006 to 2017 in California via TIMS, such as the distribution of type of collision in the past three years.

3. System Design

Figure 1 shows the end-to-end pipeline of the whole collision analysis system based on DMV AV collision reports and the TIMS dataset. The rest of the reports will deliberately explain each subsystem and its corresponding functionalities with examples and statistics. The summary is given at the end of the report.

All the relevant codes, examples, and results are public and have been uploaded to <https://github.com/MengqiaoYu/Autonomous-Vehicle-Collision-Analysis-System.git>, please check them based on your need.

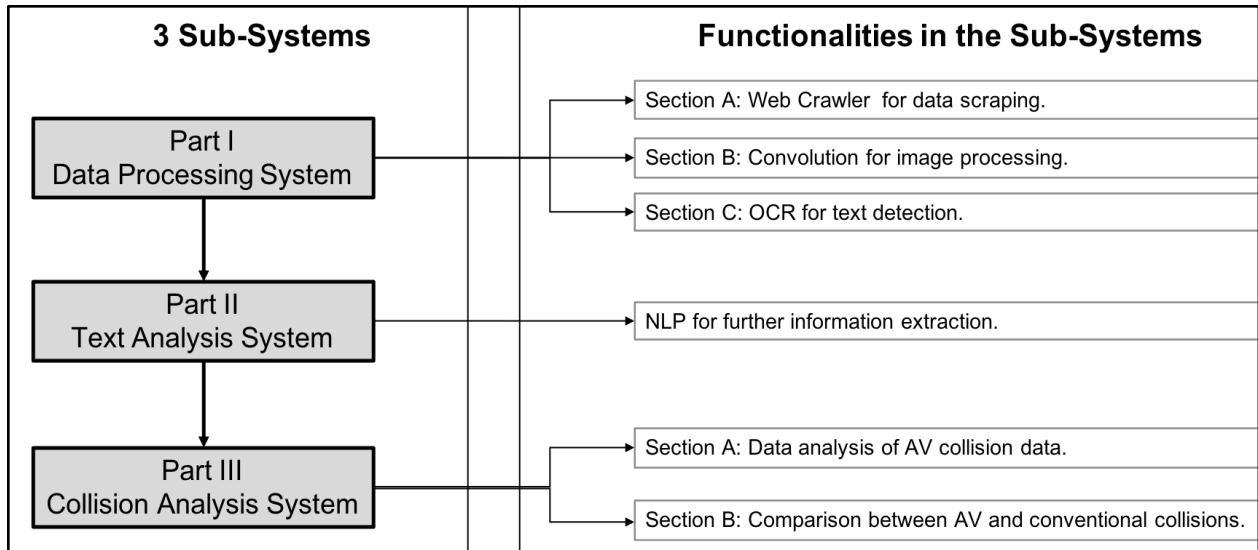


Figure 1. System design overview.

4. Part I: Data Collection and Processing System

The data collection and processing system consists of three functionalities, shown in Figure 1. The first and third ones target all released AV crash reports while the second one is designed only for crash reports after April 1, 2018. Unlike previous analyses for conventional vehicle collisions, which are based solely on the well-formatted state database, AV crash reports are scanned documents with its own format containing both tabulated data and natural language text. Therefore, a challenging but essential task is to build an efficient and sustainable data collection and processing system for the upcoming substantive amounts of AV crash reports. The following subsections will present each functionality of the system and the technique it used.

4.1 Web crawler

Once direct to the DMV web page (https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/autonomousveh_0l316), the common procedure is to download each crash report one by one. It's relatively intuitive and easy when the number of crash reports was limited, like before 2018. However, the number is increasing exponentially in recent three years, and we hence designed a web crawler programming that can scrape data and download all documents from the DMV website in just a few minutes. To summarize, the web crawler consists of three steps as follows. Please check the relevant codes in the webcrawler.py under the Github folder.

- (1) Extract the targeting "Xpath(s)" from the website source code, shown in Figure 2.
- (2) Use "requests" package to execute download button, shown in Figure 2.

(3) Extract title/date information of each document from the source code and save the file.

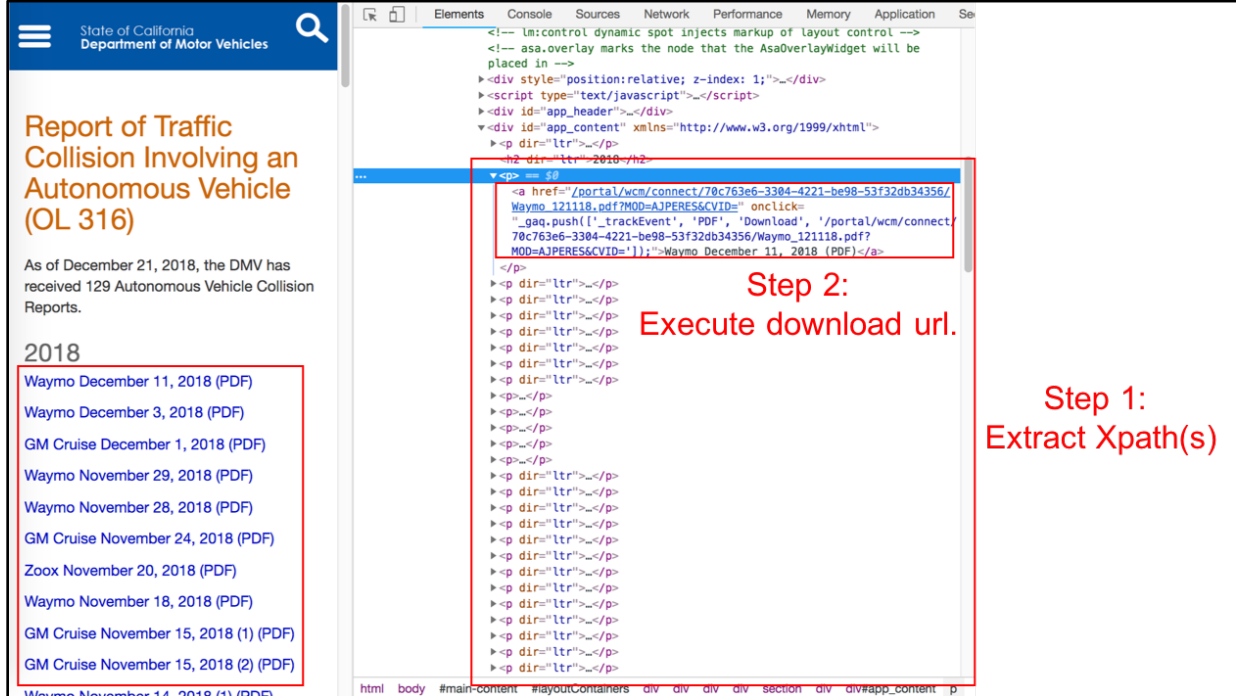


Figure 2. Build “web crawler” to efficiently collect data.

4.2 Convolution

Convolution is an important and popular technique in signal and image processing. For convolution operating on two images, you can think of one as the input image, and the other one as a special “filter” on the input image. Such an operation will produce an output image, which will highlight or blur some desired features of the input image based on the choice of the filter image. Figure 3(a) shows the original scanned table from one crash report. Our objective is clear: identify the key information by locating the checkmarks. The result of convolving standard checkmark image with the input crash report is shown in Figure 3(b). The result image can be converted as a 2-D array, which will be further used to calculate the position of each checkmark. For accuracy consideration, we only focus on the data within the predetermined checkmark area, shown as the red rectangles in Figure 3(b).

During the convolution procedure, several parameters are required to tune for generalization. They are specified as follows:

- (1) The choice of filter image. Due to the low resolution of the scanned pdf, there exists unavoidable noises around each character. The filter image, which will be applied to all crash reports, should be as typical as possible.
- (2) The location of predetermined checkmark area. We consider a larger buffer for the checkmark area in order to generalize the algorithm to all reports; however, still some crash reports are skewed in different degrees, and sometimes it’s hard to accurately confirm the location of each checkmark. For these special cases, we give an warning for

these files, save them to another folder, and process them individually by changing the orientation.

- (3) The width/height of each cell, and the number of “white” pixels in each cell if there is a checkmark in the result image.

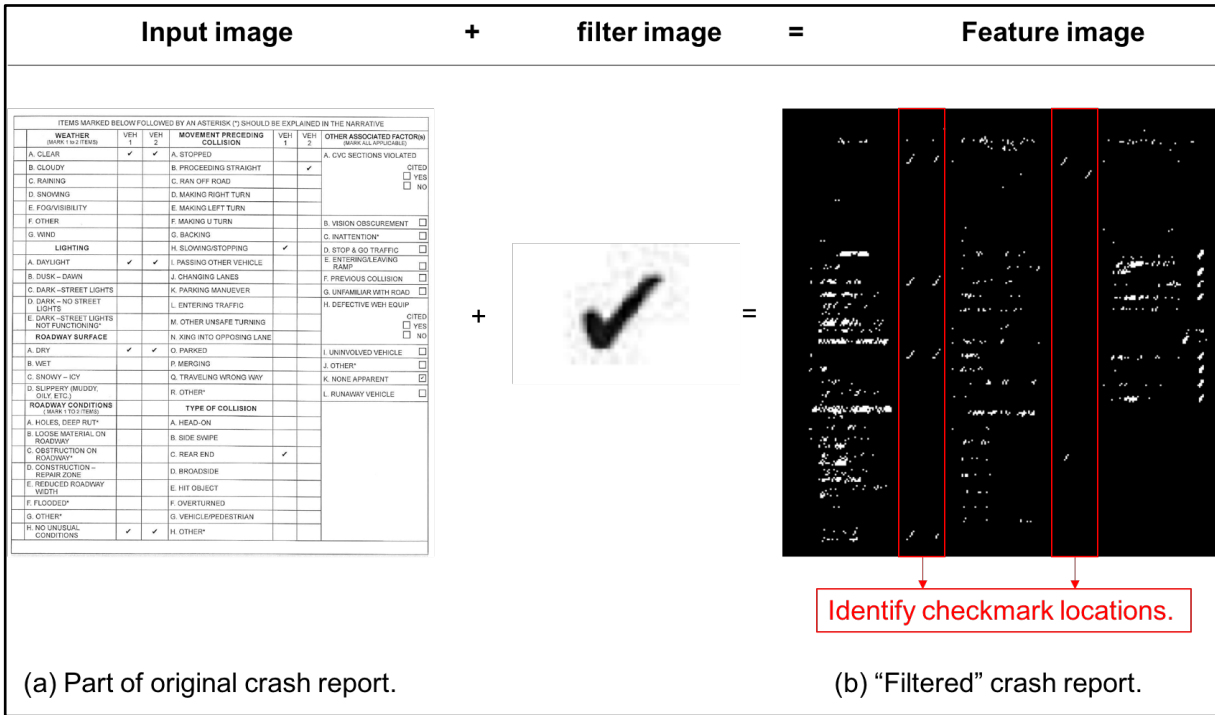


Figure 3. Apply “convolution” to processing images and extracting collision information.

4.3 OCR

In the second page of the crash reports, there is a section “accident details-description” providing additional information about the status before, during, and after the collision. We’d like to extract the text (natural language) in this section for further text analysis. We applied optical character recognition (OCR) technique, which uses Google Tesseract on the scanned documents, to convert either typed, handwritten or printed text into machine-encoded text.

Acrobat also provides a similar conversion function as OCR. The conversion result is shown in Figure 4. It can be concluded that OCR often adds unnecessary line break while Acrobat often split one word with spaces. The line breaking problem can be easily fixed by removing the line break; however, it’s a demanding task to remove ineffective spaces within one word. Also, wrong word splitting undoubtedly increases the difficulty of future text analysis. Therefore, we adopted OCR to process the paragraphs in the crash reports. The relevant codes can be referenced in the OCR.py under the Github folder.

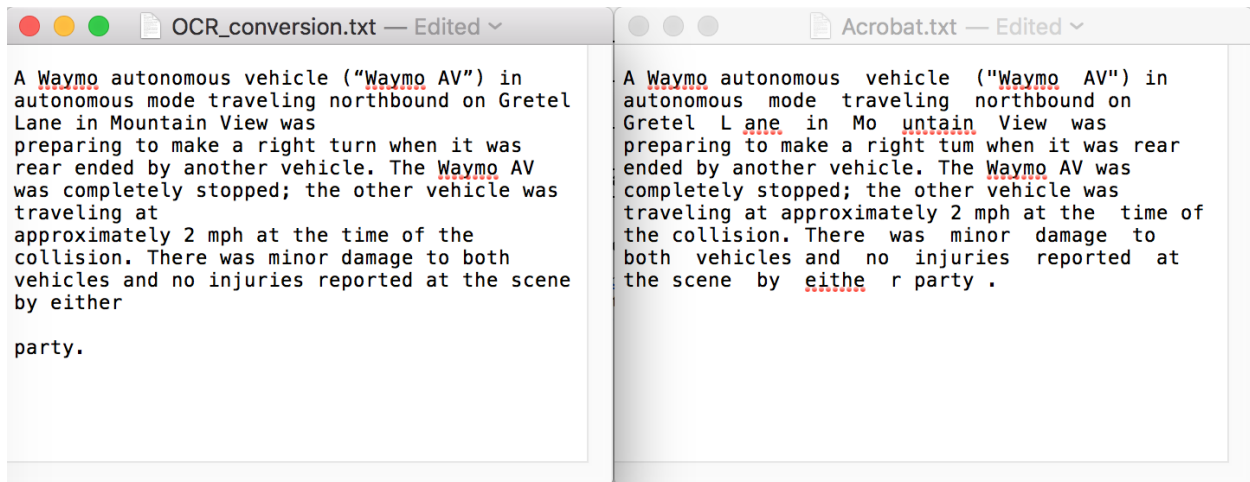


Figure 4. Compare the conversion quality between OCR and Acrobat.
(The left file uses OCR while the right one used Acrobat.)

5. Part II: Text Analysis System

5.1 Reports after April 1, 2018

After the Part I procedure, we'd like to extract more information from the result OCR text file by using natural language processing (NLP). As we discussed before, the reports before and after April 1, 2018 are in different formats, thus we will conduct text analysis for them respectively. For the reports after April 1, 2018, since most of the key elements have been summarized in the table, we will only emphasize one pieces of detailed information about the collision: the speed of each party before the collision happened. Note that the speed might be missing or obscure in some crash reports. The speed information is extracted using two rules:

- (1) Find the sentence with the keyword "mph" by using regular expression.
- (2) Find the subject of the speed by exploring the dependency relationships in the targeting sentence. Dependency relationships is widely known as Dependency Parsing in NLP, and it is the task of recognizing a sentence and assigning a syntactic structure to it. There are many parsing algorithms, and in this study, we implemented it using SpaCy library. Its syntactic dependency scheme is used from the ClearNLP [6].

Take the following sentence as an example:

(Example) "At the time of collision, the Hyundai was traveling at just under 4 MPH and was slightly accelerating towards the Zoox AV prior to impact."

We first located the existence of keyword "mph", and then extracted the sentence before mph and after the nearest punctuation, which is "the Hyundai was traveling at just under 4 MPH ". We conducted dependency parsing on this truncated sentence and yielded the following

structure. We finally confirmed “the Hyundai” as the subject of the speed “4 mph” since its dependency is labeled as nsubj (nominal subject) and it is a noun or pronoun. The rule can be expressed by “token.dep == nsubj and (token.pos == NOUN or token.pos == PROPN)”.

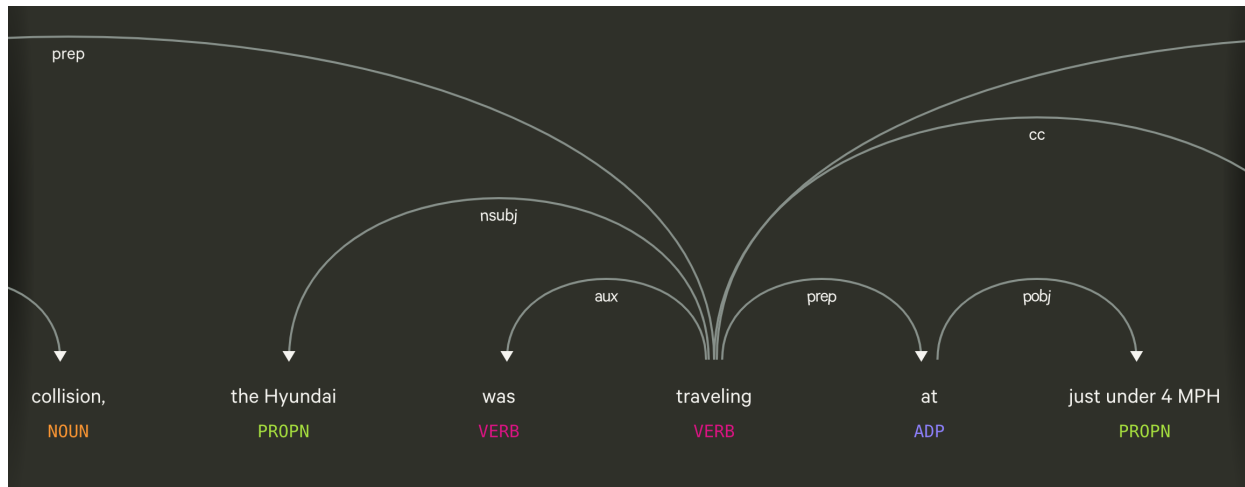


Figure 5. Dependency parsing on one sentence in the crash report.

5.2 Reports before April 1, 2018

For the reports after April 1, 2018, we don't have the summary table, thus we have to find these elements from the text by a more complicated NLP procedure. To clarify the problem, we listed the potential elements we can obtain from the text narrative of the collision as below. We should be aware that environment conditions such as light and weather are not available from the text.

- (1) Type of collision.
- (2) Movement proceeding of each party.
- (3) Speed of each party (same procedure with the reports after April 1, 2018 discussed above).

Let's consider the following sentences:

(Example 1) “A Toyota Camry traveling behind and to the left of the Cruise AV, and gaining on the Cruise AV, did not shift left with its lane and instead crossed over its lane boundary and lightly swiped the side of the Cruise AV.”

(Example 2) “The Cruise AV responded by decelerating, and a car following closely behind rear-ended the Cruise AV.”

(Example 3) “When the light turned green, the Cruise AV began moving forward. Shortly thereafter, with the Cruise AV traveling at <1 mph, a van closely following behind ran into the back of the Cruise AV.”

It can be observed that the type of collision can be implied from the key verb such as “swipe” and “rear-end”. Therefore, the problem can be converted to finding the key verbs. The key verb pool is created based on the seven categories specified in the table after April 1, 2018. Notwithstanding the key word pool, sometimes it’s still ambiguous for the machine to recognize the type of collision like Example 3. For these special cases, we will manually label them instead of relying on the machine.

Now our last challenge is to determine the movement of each party. The difficulty contributes to the complexity and variety of the sentences, especially if we want to generate a rule that can apply to all crash reports. We adopted two general strategies:

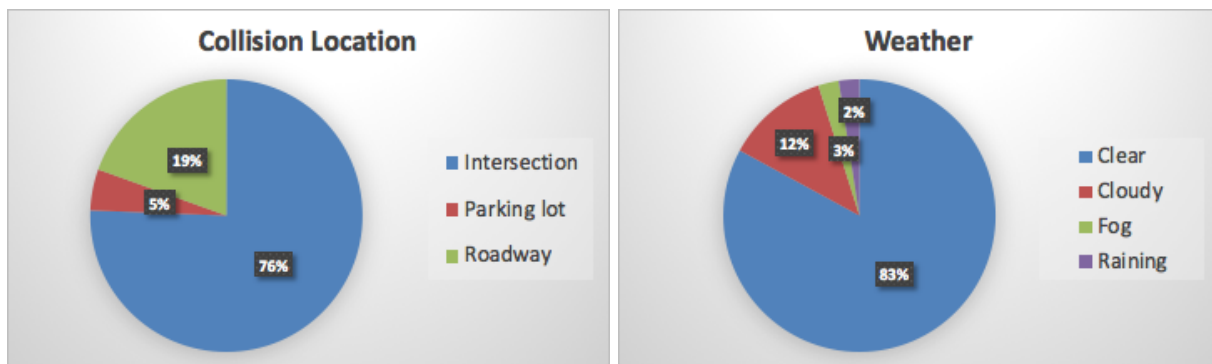
- (1) Keyword-based rule. We still establish a word/phrase pool that indicates movements, such as “turn right”, “make a right turn”, “proceed forward”, “merge”, “pass”, etc.
- (2) Clarify the subject of each movement as much as possible. This is actually an extremely onerous task in practice. It’s misleading if you read the sentences starting with “the other vehicle”, “it was rear-ended”, or the names of one subject are not consistent among several sentences in one crash report.

Compared with the reports after April 1, 2018, the reports before that usually elucidate the collision with more details. It’s encouraged to make more efforts on NLP analysis on these reports as a future research task.

6. Part III: Collision Analysis System

6.1 Summary of crash data

This study only focuses on analyzing the crash reports in 2018. There are 41 effective crash reports in 2018 in total, and here the “effective” represents that the test autonomous vehicle was operating in autonomous mode but not conventional mode. The distribution of each key element is shown in Figure 6. The movement information is summarized in Table 1.



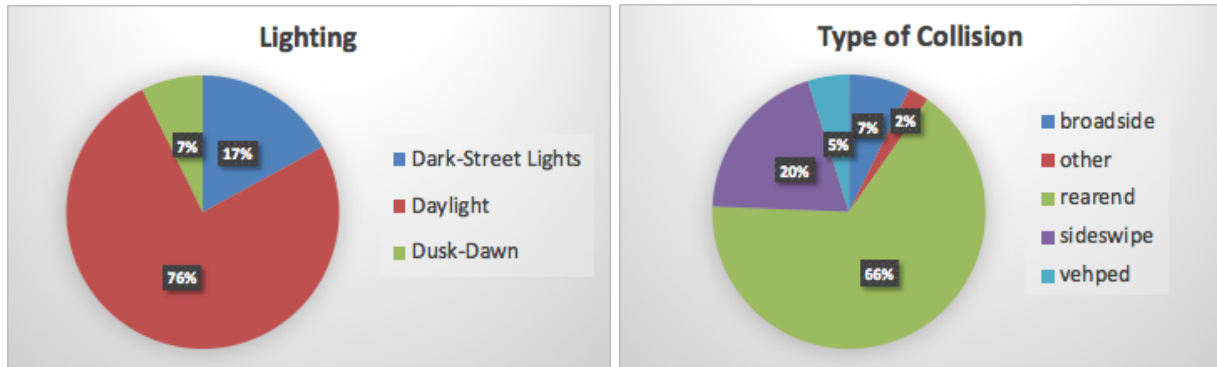


Figure 6. Distribution of each key elements in the crash reports.

As high as 76% AV collisions happened at intersections, which indicates that intersections can be a demanding situation for AV and that current AV tests emphasized more on intersections than other roadway scenarios. The weather and lighting distributions show that most of the tests were still conducted under a good environment condition (clear and daylight), and the performance of AV under worse conditions needs further validation.

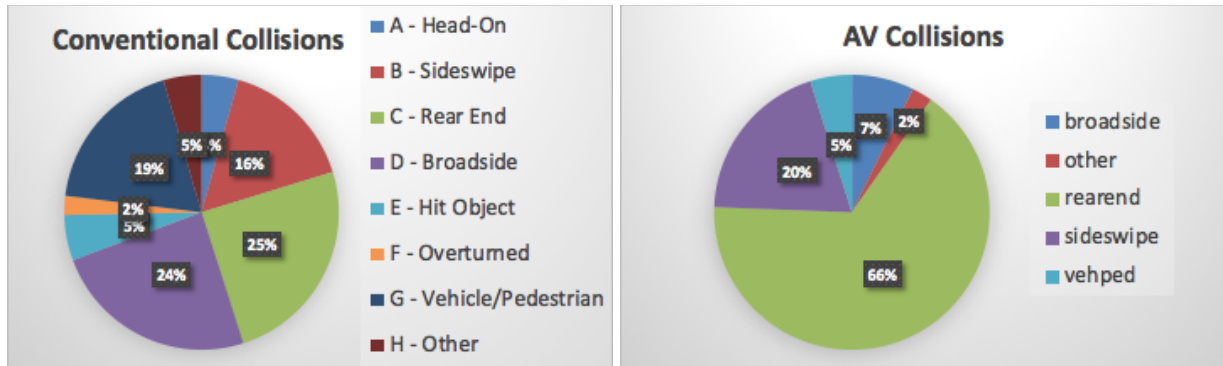
We'd like to point out an interesting finding on the distribution of type of collision. Nearly two thirds of AV collisions belong to rear-end, followed by the sideswipe covering 20%. To better understand this phenomenon, we reorganized the movement of each party shown in Table 1. It can be observed that the autonomous vehicle was usually the party that was rear-ended, and it stopped in most of the time. We deduce that it is because either the AV slowing/braking behavior is abrupt or sometimes the AV is so conservative that arouses impatience or misunderstanding of the following human drivers. Furthermore, when the autonomous vehicle is proceeding straight, changing lanes and passing behavior contribute to most of the collisions under this situation. This is reasonable considering these two behaviors are common and dangerous in our daily life. To make our analysis more rigorous and better understand the AV crashes, we compared the AV statistics with the conventional vehicle's.

6.2 Comparison

Figure 7(a) presents the distribution of type of collision in San Francisco in 2017 since the 2018 data is not available up to now. We can observe that rear-end is also the primary type of collision but is far less than (25% vs 66%) the percentage in AV collisions. Another big difference reflects in the share of vehicle/pedestrian (vehped). It covers nearly 20% of collisions in San Francisco compared to 5% in AV crash reports. Such a gap can be explained by two possible reasons: (1) As we mentioned, the AV control algorithms are very cautious and conservative, thus AVs will yield to pedestrians as much as possible; (2) Current test settings are relatively simple, and usually the test areas avoided CBD area and the test hour avoided peak hours, which reduces the exposure of pedestrians. The share of sideswipe is similar for conventional and AV collisions. Considering the limited number of crashes involving AVs in current stage, all the above analysis and conclusions are preliminary with no statistical significance; however, we still can get some insights from this and provide recommendations for manufacturers and policy makers.

Table 1. Movement summary.
(Veh 1 represents the autonomous vehicle.)

Movement of Each Party	Count
Veh1: Making Left Turn	3
Veh2: Making Left Turn	1
Veh2: Proceeding Straight	2
Veh1: Making Right Turn	1
Veh2: Proceeding Straight	1
Veh1: Proceeding Straight	11
Veh2: Changing Lanes	3
Veh2: Making Left Turn	2
N/A	1
Veh2: Passing other vehicles	3
Veh2: Proceeding Straight	2
Veh1: Proceeding Straight/Slowing/Stopping	1
Veh2: Proceeding Straight	1
Veh1: Slowing/Stopping	1
Veh2: Proceeding Straight	1
Veh1: Stopped	22
Veh2: Changing Lanes	1
Veh2: Making Right Turn	5
N/A	3
Veh2: Passing other vehicles	1
Veh2: Proceeding Straight	7
Veh2: Proceeding Straight/Slowing/Stopping	2
Veh2: Slowing/Stopping	3
Veh1: Stopped/Merging	1
Veh2: Proceeding Straight	1
Veh1: Stopped/Parked	1
Veh2: Backing/Parking Manuever	1
Grand Total	41



(a) (b)
Figure 7. Comparison between conventional and AV collisions.

We propose three viewpoints for reference:

- (1) Manufacturers are encouraged to explore the reasons behind each rear-end collision and adjust their control algorithms to follow common human driving behavior.
- (2) The performance of autonomous vehicle is still uncertain, especially under a little more complicated scenario. More simulation and field test settings need to be considered in the future.
- (3) Although the vehicle/pedestrian crash is relatively rare in current stage, the government needs to contemplate on how to protect pedestrian from injure by AVs which can be avoided by human drivers and how to evaluate the safety of AVs in the future.

7. Summary

This study proposed and implemented an end-to-end Data Processing & Collision Analysis System based on raw DMV crash reports. The input is really simple, and it can be a single url link to the DMV dataset. The system will then automatically extract key information in each crash report and provide safety insights. The system incorporates interdisciplinary knowledge, including data scraping, image processing, natural language processing, transportation safety, etc. To the best of the author's knowledge, such an effort can be seen as a pioneer in analyzing real-world AV crashes and still has a lot of aspects to work on. Some meaningful perspectives can be exploring a complementary strategy in text analysis section, conducting statistical tests once more crash reports are released, etc.

Acknowledgement

Funding for this project was provided by UC Berkeley Safe Transportation and Research Education Center (SafeTREC) and the Collaborative Sciences Center for Road Safety (CSCRS), a U.S. Department of Transportation-funded National University Transportation Center led by the University of North Carolina at Chapel Hill's Highway Safety Research Center.

References

- [1] National Highway Traffic Safety Administration (NHTSA) (2018): National Automated Vehicles for Safety. Available at: <https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety#issue-road-self-driving>. [2] Fagnant, Daniel J., and Kara Kockelman. "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations." *Transportation Research Part A: Policy and Practice* 77 (2015): 167-181.
- [3] Wood, Stephen P., et al. "The potential regulatory challenges of increasingly autonomous motor vehicles." *Santa Clara L. Rev.* 52 (2012): 1423.
- [4] Petit, Jonathan, and Steven E. Shladover. "Potential cyberattacks on automated vehicles." *IEEE Trans. Intelligent Transportation Systems* 16.2 (2015): 546-556.
- [5] State of California (2015). *Autonomous vehicles in California*. Available at: <https://www.dmv.ca.gov/portal/dmv/detail/vr/autonomous/testing>
- [6] Choi, Jinho D., and Martha Palmer. "Getting the most out of transition-based dependency parsing." *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*. Association for Computational Linguistics, 2011.