Driver Distraction Detection

Weicong Feng
wfeng@gradcenter.cuny.edu

Nancy Sea
csea@gradcenter.cuny.edu

May 15, 2022

Abstract

Advanced Driver Assistance System (ADAS) has been introduced to efficiently mitigate risks of distracted driving, which is one of he leading causes of death. We believe that a great addition to ADAS could be a smartphone app which can help detect manual signs of distracted driving. This project explores the application of computer vision algorithms for distraction detection. Our algorithm achieves an accuracy of 91.6% compared to advanced machine learning algorithms. This report introduces the experimental setting, data collection, the architecture and implementation of the algorithms. This project lays the foundation for future ADAS app development.
Keywords: ADAS, driver distraction, computer vision, machine learning

1 Introduction

Rapid development of technology and economy has allowed the usage of automobiles in developing countries to grow in the same manner in the past few decades.According to the WHO, every year the lives of approximately 1.3 million people are cut short because of a road traffic crash. Between 20 and 50 million more people suffer non-fatal injuries, with many incurring a disability as a result of their injury [1]. The driver's distractions accounts for somewhere between 25% to 75% of all crashes and near-crashes, and is becoming an increasing serious social concern.

Newer car models that are out in the market today includes a system called Advanced Driving Assistance System (ADAS). The function of the system varies amongst manufacturers and its definitions is still being formed and enriched. However most of the common functions include includes pedestrian detection and avoidance, lane departure warning and correction, traffic sign recognition, automatic emergency braking, adaptive cruise control, driver monitoring system, and so on. ADAS can reduce the workload and risks for both drivers and pedestrians.

Researchers have identified numerous behaviors that compromise driving safety, for example, texting, talking, eating/drinking, operating the radio, drowsiness, vision off-road, reaching behind, making up, and so on. This project only considers 3 common behaviors, drowsiness, talking, and vision off road [2].

2 Related Works

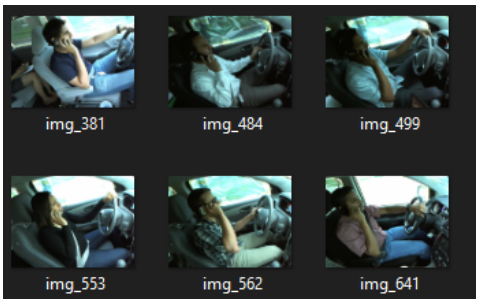
Currently, there are three major approaches to identifying driving distraction: computer vision, machine learning, and combining use of sensors and radars [2].

Naik et. al introduced algorithm for drowsiness detection with OpenCV-base algorithms and sensor-based algorithms [3]. Lee et. al discusses the ellipsoidal models to determine the driver's yaw for gaze estimation [4]. Kumar et. al presented an ensemble of six deep learning models, namely, AlexNet, VGG-16, EfficientNet B0, Vanilla CNN, Modified DenseNet and InceptionV3+BiLSTM [5] tested on the AUC Distracted Driver Dataset and State Farm Driver Dataset achieving a high accuracy.

Deep learning has shown great progress in recent years that has attracted more and more researchers in exploring its applications, however the approach requires a great amount of training and validation data than what is currently available for distracted driving. On the other hand, computer vision experts have had a extensive research on the field of facial and human state recognition. Unlike deep learning, computer vision approaches are considerably more flexible to the limited existing data.

Viewing angle	Dataset	Number of classes	Number of drivers	Number of frames/duration	Year	Publicly available
Side	SEU-DP	6	20	80 frames	2011	No
	State farm	10	26	22 K frames	2016	Yes
	AUC-V1		31	17 K frames	2017	
	AUC-V2		44	14 K frames	2019	
Front	EBDD	5	13	40 min videos	2016	
	SHRP2	Unlabeled	2600	_	2013	
	VIVA Face		_	Images from 39 videos	2015	

Figure 1: Common open-source dataset



(a) Viewing angles from State Farm data set

(b) Front-facing viewing angles of our dataset

Figure 2: Data sets in this experiment

3 Dataset

To this day, the open data sets in driver distraction are still limited. Moslemi et. al [6] lists available open-source data sets shown in Figure 1. Although widely popular, the angles of the videos in the data set suggests that its most practical in recognition in deep learning models but not to computer vision approaches (see Figure 2a). We show this with testing the State Farm data set on our algorithm as well.

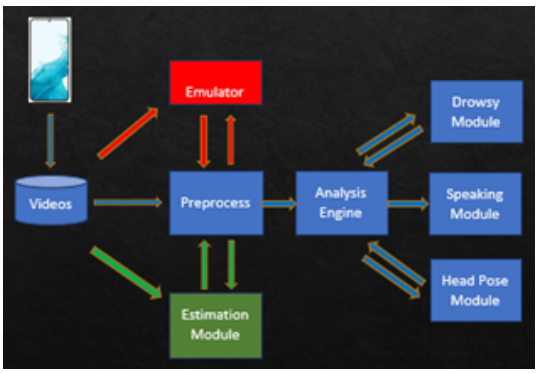
Thus, to test our algorithm, we collected the data ourselves by recording 13 videos using an iPhone front camera that is mounted on the dashboard in front of the driver (see Figure 2b). The set contains a total of 9,890 frames. Another reasoning for using self-collected data is that it is very rare that ADAS will use the angles in the data set mentioned in [6].

4 Architecture

New technologies and its applications are rolled out on a phased basis. In the long run, every vehicle will be equipped with a feature-rich ADAS. This project however aims to help existing vehicles without ADAS. We hope to develop a smartphone app that captures the driver's video in real-time and sends the stream to the analysis engine to determine whether a dangerous behavior is detected.

The scope of this project is to only focus on the analysis engine (Figure 3 b). In our experiment, we recorded videos by smartphone and stored them on a hard drive. Three major modules are developed, the emulator is used to imitate the real application, and the estimation module is used to estimate the performance of the algorithms. The most important module is the core module, the analysis engine, the emulator and the estimation will call the analysis engine. The analysis engine has three sub-modules, drowsy, speaking, and head pose, respectively.

(a) Application architecture



(b) Experiment architecture

Figure 3

5 Implementation

5.1 Drowsiness and talking Detection

Generally, when we are falling asleep, the distance between the upper and lower eyelids goes smaller than normal. When we are silent, the distance between the top and bottom lips is almost zero, but if taking, the distance will change. The combination of distances between eyelids and lips are the key to identifying drowsiness and talking.

Because the distance between the drivers' face and camera is relatively constant, the depth variances of the points on the eyelids and lips are very small. A weak-perspective projection model is suitable for this environment.

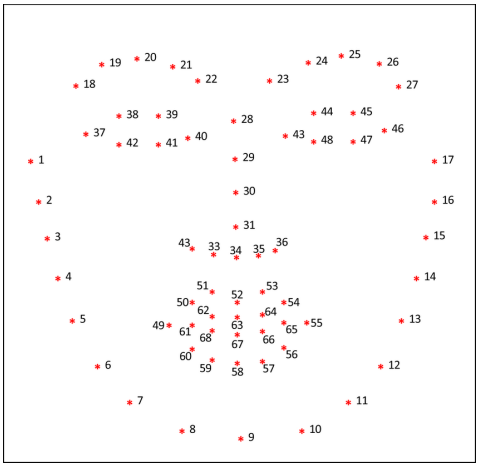
$$x = f\frac{X}{\overline{Z}}, y = f\frac{Y}{\overline{Z}} \tag{1}$$

Facial landmarks are detected using `DLIB`'s implementation of [7]. The aspect rations and Euclidean distance between lisp are used to identify drowsiness and talking. These are not the only factors needed to be considered in the judgment of drowsiness and talking. Another crucial factor is the time of duration, we call it the temporal factor. For example, closing our eyes is normal even during driving because we need to blink. So, we can't judge drowsy by a single frame with closing eyes. We only can say drowsiness happens when the closing eyes are in 20 consecutive frames or more. When people speak, their mouths open and close quickly. 20 frames or more is not a good threshold to identify speaking. However, using a single frame to judge speaking will cause a number of error detection. In our experiment, we set 3 consecutive detected frames to identify the speaking. There are several parameters need to be tuned so that improve the computer vision model better performance, such as threshold (how many eyes aspect ratio is less than consider drowsiness, how many pixels the distance between lips is bigger than consider a speaking), number of consecutive frames (how many consecutive frames can be considered identified as a distraction)

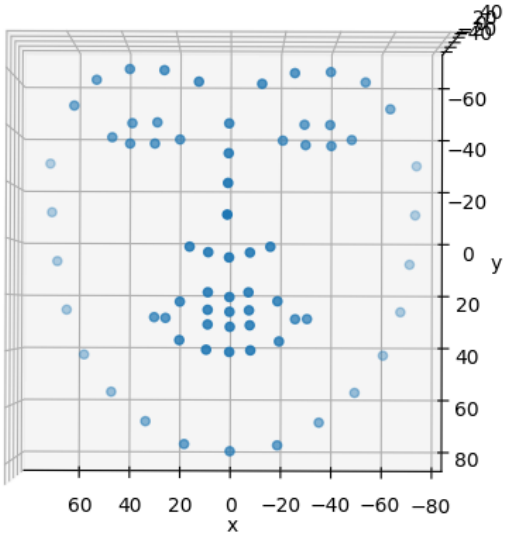
5.2 Head Pose Module

For the third component of our analysis engine of determining out of road detection, the pose of the head from each frame of the input video is analyzed. Like in the drowsiness and talking detection, the driver's face and facial features are detected using the face land-marking model from the paper 'One Millisecond Facial Alignment with Ensemble of Regression Trees' [7]. This model allows us to obtain 68 points of interest in figure 4a in total for the next steps of our module.

Out of road detection is framed as a head pose estimation problem. Given the facial features on the image plane from the step above, we want to determine the pitch, roll, and yaw of the driver in the real world coordinates. Then using a method called Effective Perspective-n-Point, we can obtain the pose of the head in relation to the camera as desired.



(a) Facial landmarks detected from captured frame using [7]



(b) Generic 3D face model used as reference object in world coordinates

Figure 4: Facial models used in experiment

5.2.1 Perspective Projection Model

Recall the equation representing the perspective projection model

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -f_x & 0 & o_x \\ 0 & -f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & T_x \\ r_{21} & r_{22} & r_{23} & T_y \\ r_{31} & r_{32} & r_{33} & T_z \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} \tag{2}$$

From this equation and the definition of the problem, observe we only have one component, the projection points on the image plane, $\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix}^T$ whereas the intrinsic and extrinsic parameters as well as the world coordinates are still unknown. The subsections below will discuss the assumptions we have made to approximate these unknowns.

Camera Intrinsic Parameters There are various different procedures that can be implemented to estimate the camera intrinsic parameters. In our work, we have tried Zhang's method [8] and continued our experiment successfully. However, we realized that we want a system that is dynamic and flexible where the driver do not need to calibrate their camera first for our module to be able to analyze their pose.

Hence we decided to disregard the need for camera calibration and decided to estimate the parameters directly from the video input itself instead. The effective focal lengths f_x and f_y are assumed to be equal to the height h of the input. The center of projection $(o_x, o_y) = (width/2, height/2)$ is the center of the image. This estimation still yields appropriate results but with different scale as the real focal length is not considered. It should be noted that since the camera is stationary in our dataset, we could also use the approximate measurement from where the camera is placed on the dashboard to the head of the driver in as the focal length as well.

World Coordinates Just as with estimation of camera calibration parameters, it is very difficult to map the projections from the image plane to the object in world coordinate system. We opted in using a generic 3D model of a face as the reference for the driver's head in the 'world' coordinate system shown in figure 4b.

5.3 Perspective-n-Point

With the assumptions above, our camera is now 'calibrated' and we have 68 corresponding 3D-to-2D points, we can solve this Perspective-n-Point problem for the known coordinates in the camera

coordinate system from the camera's pose and orientation. We wanted our analysis engine to work with inputs of captured videos and real-time videos, it is important to employ a solution with lower computational complexity which is proposed by Lepetit et. al [9] and implemented on the `Open-CV` library.

The function `cv.SOLVEPNP_EPNP` returns the rotation and translation vector which represented the pose of the camera. Utilizing more of the library, we are able to obtain a $3x3$ rotation matrix, stacking it together with the translation vector, resulting in a projection matrix which is decomposed to obtain the Euler angles needed for determining Out of Road Distraction detection.

5.3.1 Out of Road Distraction

After estimating the angles, Out of Road Distraction detection is simple. For each of the three axes, we set a threshold of allowed movement. These are the following thresholds for our experiment $pitch = 5^o, yaw = 5^o, roll = 20^o$ (radian degrees) with the addition of a temporal threshold of 50 frames. The frame threshold is an important addition since we want to allow certain movement such as head turn to check for oncoming vehicles rather than classifying it as a distraction. When the pose of the driver exceeds the thresholds, an alert will be shown to the driver. This simple algorithm wraps up the Out of Road Distractions.

5.3.2 Challenges

In this section, we want to discuss the challenges of estimating the driver's pose. First, we assumed the intrinsic parameters - which affects the scaling thus the threshold might need to be adjusted for videos with the camera placed closer or further away from the driver. For this module to work, we also assume that the driver's face is directly facing the camera with all points and landmarks in view. The algorithm does not when parts of the face is occluded or if there are extreme lighting changes between the frames.

6 Results and Discussions

Thirteen videos recorded by us are used for estimating the algorithms' performance, while the State Farm data set is used to train a VGG16 deep learning model. The experiment result is shown in table 2. The data set used for estimation is recorded in real driving environments, including light changing, shadow, dazzle, shake, and with glasses. Hence, the experiment result would close the real-world driving environment.

Results			
	CV (our approach)	CV with temporal factor (our approach)	DL - VGG16
Accuracy	69.5%	91.6%	1.66%

Table 1: Results of the experiment

The computer vision method operated with temporal information (use consecutive frames) and without temporal information (use single frame). From the above result, it shows that the temporal information improves the accuracy from 69.5% to 91.6%. The parameter of the number of consecutive frames and the threshold compact on the accuracy significantly.

At first, we planned to use ensemble, combining deep learning and computer vision. Eventually, we abandoned this idea because of the poor accuracy of the deep learning model. When we trained a modified VGG16 transfer model[1], the testing accuracy is 80%, as shown in figure 5. However, its performance in our data set is terrible. We suppose it is because the training data set has a totally different shooting direction from our testing data set, leading to the training data set distribution being different from the test data set.

[1] https://github.com/Apoorvajasti/Distracted-Driver-Detection

Figure 5: DL Training Results

7 Future Works

Although decent results have been obtained, there is still room for improvement. First, single driver data set will cause a bias issue. Collecting more videos, especially of various drivers, with glasses and without glasses, male and female, is the first task. Second, continue tuning the parameter, especially using data sets from various drivers. Third, other distracted driving behaviors should be added to the detector. Forth, retrain the deep learning model with our data set, and use the ensemble model. Finally, the development of a mobile app for this algorithm would be the most meaningful addition to our tasks.

8 Conclusion

At a time when distracted driving is increasingly becoming a threat to people's safety, ADAS cannot be popularized in a short time, and the demand for a smartphone-based ADAS becomes growing urgent. As an exploration of the mobile version of ADAS, our team created a data set, labeled the data set, developed an analysis engine, and obtained an accuracy of 91.6%. The emulator provides a simulation of real application environments. This project lays the foundation for future ADAS app development.

References

[1] World Health Organization, "Road traffic injuries." https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries, 2021. Last accessed 15 May 2022.

[2] A. Kashevnik, R. Shchedrin, C. Kaiser, and A. Stocker, "Driver distraction detection methods: A literature review and framework," *IEEE Access*, vol. 9, pp. 60063–60076, 2021.

[3] D. G. Naik, Suha, and S. G. Bhagwath, "Driver distraction monitering alert system by using opencv algorithm," *International Journal of Research in Engineering and Science*, vol. 10, 2022.

[4] S. Lee, J. Jo, H. Jung, K. Park, and J. Kim, "Real-time gaze estimator based on driver's head orientation for forward collision warning system," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 12, pp. 254 – 267, 04 2011.

[5] A. Kumar, K. S. Sangwan, and Dhiraj, "A computer vision based approach for driver distraction recognition using deep learning and genetic algorithm based ensemble," in *Artificial Intelligence and Soft Computing: 20th International Conference, ICAISC 2021, Virtual Event, June 21–23, 2021, Proceedings, Part II*, (Berlin, Heidelberg), p. 44–56, Springer-Verlag, 2021.

[6] N. Moslemi, M. Soryani, and R. Azmi, "Computer vision-based recognition of driver distraction: A review," *Concurrency and Computation: Practice and Experience*, vol. 33, 2021.

[7] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1867–1874, 2014.

[8] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1, pp. 666–673 vol.1, 1999.

[9] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Epnp: An accurate o(n) solution to the pnp problem," 2008.