2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution (CSITSS)

Evaluation of Internal Factors in Driver Distraction

H.S. Srihari
Computer Science Department
R.V. College of Engineering
Bengaluru, India
hssrihari.cs16@rvce.edu.in

Aishwarya Seth
Computer Science Department
R.V. College of Engineering
Bengaluru, India
aishwaryaseth.cs16@rvce.edu.in

Dr. Sowmyarani C.N.
Computer Science Department
R.V. College of Engineering
Bengaluru, India
sowmyaranicn@rvce.edu.in

***Abstract*— Classification and in-depth analysis of these various forms of distraction can allow for intelligent alert systems. Before creating an artificial intelligence system, it is imperative to understand the various parameters that are interplaying in the current environment. Based on how these factors influence the behavior of a driver, the corrective measures should be suggested to the driver. (At later stages, such changes may be incorporated into the vehicle itself.)**

I. INTRODUCTION

Drivers during long journeys or during slow moving traffic, can get very distracted. The sources of these distractions may be either external or internal. Internal distractions are more complex to study, because they are reflective of mental distractions and are hence, harder to quantify. However, these play a major role in common accidents, as the individual may be distracted by something within his vehicle, rather than by an object on the road. Being distracted by an object on the road, means it would be identified in time, and the driver can attempt to take measures to avoid the same. Internal distractions indicate that the driver is mentally occupied, and hence more likely to not take the correct decisions.

The major reasons are eating/drinking, conversations, using handheld devices or something that is dropped. To study or quantify the amount of distraction experienced in each case, the deviation of the person's gaze is studied. More the deviation, higher is the level of distraction.

Classification and in-depth analysis of these various forms of distraction can allow for intelligent alert systems. Before creating an artificial intelligence system, it is imperative to understand the various parameters that are interplaying in the current environment. Based on how these factors influence the behavior of a driver, the corrective measures should be suggested to the driver. (At later stages, such changes may be incorporated into the vehicle itself.)

To decide the required corrective measures, the severity of each distraction needs to be identified. This is done by collecting the data for a pre-decided list of distractions, and designing their interaction by using the technique of categorization.

Earlier projects use posture of the driver to identify and categorize distraction. Since distractions may be purely in the mind of the driver, the gaze is used as an indicator of the amount of distraction the driver is facing at a particular instant. Thus, this project focuses on the collection of deviations in the gaze, followed by the categorization of this deviation.

The architecture diagram is shown in Figure 1.

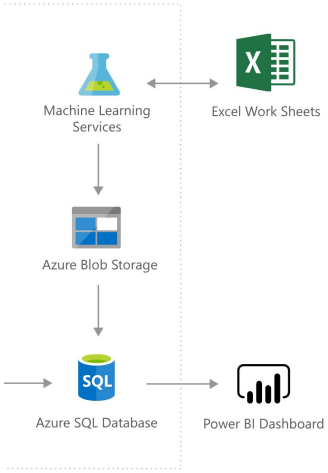


Figure 1: Architecture Diagram

II. COMPLEXITY OF IMPLEMENTATION AND TOOLS USED

A. Complexity of prototype

Azure, in itself, is a complex product to work with. The components have a number of intricacies and dependencies that must be addressed before any software can be developed. Understanding these complexities and implementing the system accordingly was one of the most fundamental challenges found in this project.

Identification of the factors that would lead to distraction of the driver had to be done arbitrarily. Most prior studies do not analyze these factors in a quantified manner, and hence the relevance of each factor had to be guessed, and only the most relevant factors were considered.

Once the parameters were identified, there is no clear way to quantify them. While gaze detection or tracking is assumed to be the most optimal way of identifying the existence of a distraction, it is possible to be unfocused even while starting straight at the road. Thus, the model designed would not be perfect. The model was expected to be sensitive to sudden, short bursts of interaction, causing sudden deviations in the received data. To avoid this, the average of several readings was recorded. The data point is set only when the reading has stabilized. Uploading any form of data to a remote service requires a high level of security and stability. This is accomplished by using the Azure Web Services, which generates a secure API key, and restricts unauthorized access of information. The model designed in this experiment is generated specific to the individual. For it to be used on a wider scale, it would have to be adaptive, and designed for a group of people. Thus, the current approach

978-1-7281-2619-7/19/$31.00 ©2019 IEEE

must be generalized by using a larger number of samples to make the model more robust and less susceptible to changes in the user. The integration of the various components was complex, as they are designed across various platforms, and each has a number of modules that must be interlinked accurately for proper information processing. Thus, using Python as the user input interface, and the Azure Web Services to use the model created in Azure ML Studio, this was accomplished.

B. Tools and Technologies used

The described project required a proper user interface in order to perform the data collection. Once collected, the data had to be processed, and passed through a model to provide the categorization of the various inputs. Python and Qt used to develop a data-gathering base front end. The entire workflow was designed to understand the various conversions and formats, with CSV as the base format for information storage due to compatibility. Thus, the user can utilize the lightweight interface application to record new samples, or to upload existing samples, and obtain an output CSV file. This CSV file is created directly in the format required for usage by the developed module. HTML, CSS, Javascript was used to create a basic web interface. This was later abandoned due to compatibility issues, and the web service was transferred onto an Azure based platform, to enable cloud storage of data. To simplify the machine learning, Azure MLStudio was used which provides a GUI for model creation. This allowed for faster development of the final product. Amazon Web Services were used to allow the user to push data directly onto a remote cloud-based server. This enables remote access of data, and made the entire project portable. Additionally, users are provided with a number of ways to input the information, and the processing is moved from the client's browser, to Azure's systems. Although this transfer of information makes the process more time-consuming, it improves the computational efficiency and security of the entire system. Using this approach also allows for immediate deployment of the developed system.

III. EXPERIMENTAL ANALYSIS

A. Capturing Data

The data is captured using a Python-Qt interface. The radio button provided allows the user to decide whether the current source of distraction is the person looking down, having a conversation, or drinking or eating. The panel shows the current position of the left and right pupil, which can be mapped onto the main window pane on the right by using the show pupil checkbox. The threshold bar shown helps control the level of contrast in the base image used for classification. The sample GUI window for collection of data is shown in Figure 2.

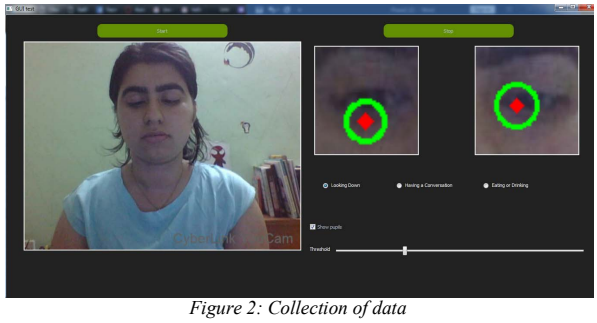


Figure 2: Collection of data

This system uses the Haar classification to identify the presence of the eye. Using this, the direction of the gaze of the individual is estimated. The program takes into account the previous locations, and gives a time-averaged location of the pupil. Thus, it returns data to the user only when the gaze is stable.

B. Haar Classifier

A Haar Classifier is used to identify the location of the images. The image is converted to the greyscale form, and distinctions in the image color are used to create a blob-like object, which indicates the position of the pupil. The Haar classifier used in this case is trained to identify circular objects. A Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region and calculates the difference between these sums. This difference is used to categorize the subsections of an image. The Haar-like features are weak learners or classifiers so large number of Haar-like features are necessary to classify images with higher accuracy. Thus, Haar-like features are cascaded to achieve this. Once a blob is plotted to show the position of the pupil, a circle is drawn to show the location and track the movement of the eye.

C. Data Processing

The data was obtained in the form of a series of .csv files . These files were collated with the respective labels where
1: Looking down
2: Having a conversation
3: Eating or drinking
These parameters were considered to be the most influential indicators of driver distraction, and were hence used to quantify the same. The restriction of influencing factors allowed for the development of an end-to-end workflow, creating a scalable prototype that can be easily extended for various purposes. The CSV format was used since this format allows for easy interpretation of data across the various platforms that were utilized in the design of this system.

D. Machine Learning - Two Class Logistic Regression

A two-class logistic regression model is built. This was designed on MLStudio and deployed as a Web Service, using cloud storage and making it available remotely.This system allows for an efficient prediction of the current distraction based on the input data. Figure 3 shows the training model that was created. The input data was split 70-30 in order to test the output model after its development.

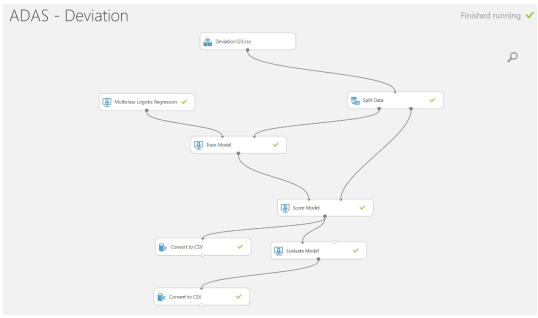


Figure 3: Training Model

The model used in this system is the Two-Class Logistic Regression. To obtain the actual quantified values of the gaze, regression was used. Since these are real world values, the logistic function is the best fit. Softmax is the most commonly used activation function due to its overall efficiency, and was also utilized in this model. The current prediction required needs multiclass approach, but the two class approach is quite stable and can be easily extended to multiple classes.

The probability of a particular class is found using the logistic function:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}.$$

To divide into multiple classes, binary classification is performed multiple times, one for each class. This is because performing the calculation for multiple classes simultaneously leads to unpredictable and unstable results. The algorithm for this approach is described below.

```
Divide the problem into n binary classification problems
For each class:
   Predict the probability that observations are in that single class
Prediction = MAX(Probability of the classes)
```

Softmax calculates a probability for every possible class. It is used to map a non-normalized output to a probability distribution over predicted output classes. It is a generalization of the sigmoid function. While the sigmoid function is used only for binary classification, the softmax function can be used even when the number of classes is greater than 2. Once the model is created, it can be used for prediction by linking the current set up to a web service, which can provide the appropriate input and output. This is shown in figure 4.

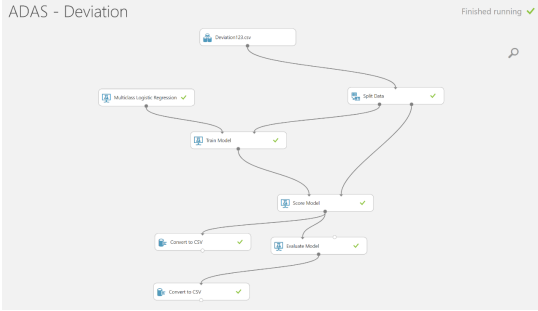


Figure 4: Predictive Model for the System

IV. RESULTS AND EVALUATION

Once the model has been created, the results need to be evaluated. Since the data was labelled, the desired output is known. In order to test the output of the model, a confusion matrix was generated. This matrix contained a test for the accuracy, a visualization of the outputs received, and thus provided a complete view of the model that had been created and its associated results and measures.

The model results are shown in Figure 5.

ADAS - Deviation › Score Model › Scored dataset

rows 60 columns 9

	LeftX	LeftY	RightX	RightY	Dist_Type	Scored Probabilities for Class "1"	Scored Probabilities for Class "2"	Scored Probabilities for Class "3"	Scored Labels
view as									
	1	-1	7	-12	3	0.13676	0.381664	0.481576	3
	12	13	17	5	2	0.185187	0.535417	0.279396	2
	-3	-6	-5	-6	1	0.525898	0.31387	0.160232	1
	-10	-12	14	-5	3	0.128201	0.257689	0.614109	3
	8	9	1	-11	2	0.22748	0.5181	0.25442	2
	7	4	-3	-4	1	0.470371	0.399212	0.130417	1
	12	10	9	-10	3	0.12098	0.496687	0.382334	2
	4	2	6	-13	3	0.135617	0.416335	0.448048	3
	0	-3	0	-1	1	0.497787	0.339666	0.162547	1
	-6	-5	0	-12	2	0.267692	0.376747	0.35556	2
	17	18	5	-7	2	0.190473	0.604047	0.20548	2

Figure 5: Scored Model Results

The actual and the predicted labels can be seen, along with the confidence of the classification. This confidence is a measure of the accuracy of the classification, denoting the probability that the class was assigned to the input accurately.

The result shows a reasonable level of accuracy and can thus be used for further predictions. The level of accuracy is about 88%, but increases to 92% on average. This means that when a higher number of samples is used, the model will still perform well.

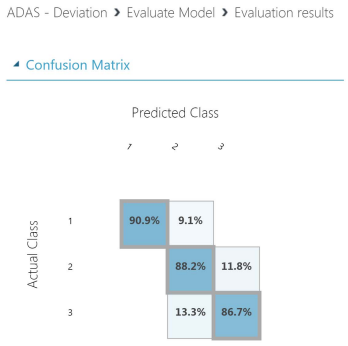
The accuracy and confusion matrix for the data is shown in Figure 6.

ADAS - Deviation › Evaluate Model › Evaluation results

Metrics

Overall accuracy	0.883333
Average accuracy	0.922222
Micro-averaged precision	0.883333
Macro-averaged precision	0.891266
Micro-averaged recall	0.883333
Macro-averaged recall	0.886037

(a)



(b)

Figure 6: (a) Accuracy and (b) Confusion Matrix for the Model

Taking Inputs from the Users

The users can provide data as a single data point, or upload it as a .csv file. The system also accepts data in terms of batches, but due to remote processing, this is a time-consuming process and has not been used widely. A sample of the upload of a single data point onto the web service, and the obtaining of an output is shown in Figure 7. The data uploaded in the form of multiple data points as a .csv is shown in Figure 8.

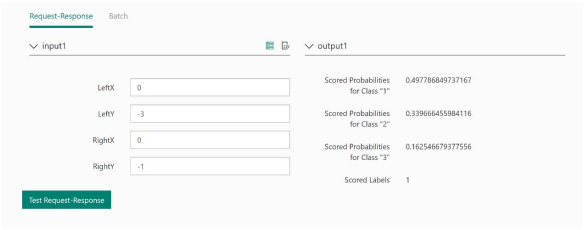


Figure 7: Sending in a Single Data Point with the Request/Response Method

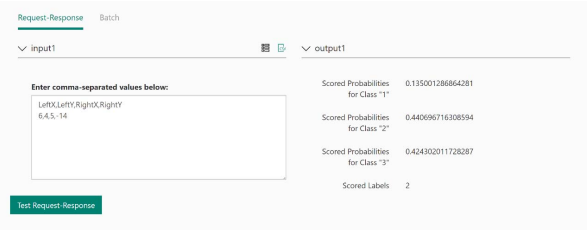


Figure 8: Sending in Multiple Data Points from a CSV File

V. CONCLUSION

A model predicting the source of the distraction is created. This system can be used for intelligent alert systems. A web service is deployed allowing for easy access to the created model. Cloud based system is used to store large amounts of information, with remote access.

The current platform consists of two basic modules. These must be integrated for proper usage. Once that is complete, the two interacting units can be used to build an alert system using these input factors. The current model focuses on niche sources of distraction, but are designed to be scalable. Analysis of multiple samples to study trends would improve prediction design. Creating a complete independent tool which can be utilized on portable devices can be done to make this prototype usable in a real-world scenario.

[1] REFERENCES

[1] Sany R. Zein, Raheem Dilgir, Vivian Law, Quantifying Driver Distractions at Intersections, Published: 2009, Available at: http://ctep.ca/wp-content/uploads/2016/11/Quantifying-Driver-Distractions-at-Intersections-FINAL-REPORT.pdf Accessed: 20-02-2019

[2] Abouelnaga, Yehya & Eraqi, Hesham & Moustafa, Mohamed. (2018). Real-time Distracted Driver Posture Classification.

[3] T. Santini, W. Fuhl, and E. Kasneci, “PuReST,” in Proceedings of the 2018 ACM Symposium on Eye Tracking Research & Applications - ETRA ’18, 2018.

[4] G. B. Shobha, G. S. Gamya, T. Yuvaraja, "Analysis of space vector intonation system", *Electrical Electronics Communication Computer and Optimization Techniques (ICEECCOT) 2017 International Conference on*, pp. 456-458, 2017

[5] H. C. Mat Haris, M A N Mohd Nor, M. H. Fazalul Rahiman, N F A Abdul Rahman, "Development of energy harvesting using speed breaker", *Systems Process and Control (ICSPC) 2016 IEEE Conference on*, pp. 53-57, 2016.

[6] W. Devapriya, C. Nelson Kennedy Babu, T. Srihari, "Real time speed bump detection using Gaussian filtering and connected component approach", *Futuristic Trends in Research and Innovation for Social Welfare (Startup Conclave) World Conference on*, pp. 1-5, 2016.

[7] Kwang Ming Lion, Kae Hsiang Kwong, Weng Kin Lai, "Smart speed bump detection and estimation with kinect", *Control Automation and Robotics (ICCAR) 2018 4th International Conference on*, pp. 465-469, 2018.

[8] James Obuhuma, Henry Okoyo, Sylvester Mcoyowo, "Real-time Driver Advisory Model: Intelligent Transportation Systems", *IST-Africa Week Conference (IST-Africa) 2018*, pp. Page 1 of 11-Page 11 of 11, 2018.

[9] Hyeong-Seok Yun, Tae-Hyeong Kim, Tae-Hyoung Park, "Speed-Bump Detection for Autonomous Vehicles by Lidar and Camera", *Journal of Electrical Engineering & Technology*, 2019.

[10] V S K P Varma, S Adarsh, K I Ramachandran, Binoy B Nair, "Real Time Detection of Speed Hump/Bump and Distance Estimation with Deep Learning using GPU and ZED Stereo Camera", *Procedia Computer Science*, vol. 143, pp. 988, 2018.

[11] "BOSCH", *Chassis systems control| bosch study on driver assistance systems*, 2012, [online] Available: http://www.bosch-presse.de/pressportal/de/media/migrated-download/del.

[12] "McKinsey&Company", *Advanced driver-assistance systems: Challenges and opportunities ahead*, 2018, [online] Available: https://www.mckinsey.com/industries/semiconductors/our-insights/.

[13] P. Tchankue, J. Wesson, D. Vogts, "Using Machine Learning to Predict Driving Context whilst Driving", *SAICSIT 2013 South African Institute for Computer Scientists and Information Technologists East London South Africa*, 2013.

[14] "Mercedes-Benz", *What is mercedes-benz attention assistance?*, 2017, [online] Available: http://mercedesbenz.starmotorcars.comlblog/.

[15] X. LI, X. Xu, L. Zuo, "Reinforcement learning based overtaking decision-making for highway autonomous driving", *6th International Conference on Intelligent Control and Information Processing*, 2015.