Driver Assistant in Automotive Socio-cyberphysical System

Reference Model and Case Study

Alexander Smirnov[1,2], Alexey Kashevnik[1,2], Nikolay Shilov[1,2] and Igor Lashkov[2]
[1]*SPIIRAS, 39, 14 Line, St. Petersburg, Russia*
[2]*ITMO University, 49 Kronverksky Pr., St. Petersburg, Russia*

Keywords: Socio-cyberphysical System, ADAS Systems, In-vehicle System, Ford Sync.

Abstract: The paper presents an automotive socio-cyberphysical system for assisting a vehicle driver. The system allows to notify people if they drive while being tired or drowsy. The reference model consist of the driver, the vehicle, driver's personal smartphone, vehicle infotainment system and cloud. Interaction of these components is implemented in a cyber space. Using smartphone cameras, the system determines the driver state using the computer vision algorithms and dangerous events identification diagram proposed in the paper. Presented approach has been implemented for Android-based mobile device and case study has been described in the paper.

1 INTRODUCTION

Vehicle driver assistance is an important modern research and development topic. In the last decades the amount of accidents on the road has remained high. Advanced driver assistance systems are aimed to help drivers in the driving process and prevent dangerous events by alerting the driver about unsafe driving conditions and behavior (Biondi et al., 2014). Such systems use computer vision and machine learning algorithms to monitor and detect whether the driver is tired or distracted using available sensors and cameras. There are two major types of driver assistant solutions available: solutions integrated in the vehicles and application for smartphones or tablet PCs that detect dangerous situations and makes alerts for drivers. However, only a tiny percentage of cars on the road today have these systems. These technologies are quite new and accessible only in business and luxury vehicle segments. At the same time a lot of car manufactures develop vehicle infotainment systems that transformed from simple audio players to complex solutions that allow to communicate with popular smartphones, share information from different vehicle sensors and provide possibilities to deliver information through in-vehicle screen or stereo system (like Ford SYNC, GM OnStar MyLink^{TM}, Chrysler UConnect$^{®}$, Honda HomeLink, Kia UVO, Hyundai Blue Link, MINI Connected, Totyota Entune, BMW ConnectedDrive systems, and other).

There are a lot of mobile applications that aimed to implement driver assistant while driving. The analysis of these applications is presented in (Smirnov and Lashkov, 2015). The following systems can be highlighted: CarSafe (You et al., 2013), DriverSafe (Bergasa, 2014), WalkSafe (Wang et al., 2012). In the paper (Aurichta and Stark, 2014) authors formalise user experience and study how it can be integrated in the validation process of Advanced Driver Assistance Systems.

Driver, vehicle, smartphone, and software services partly integrated in the mobile application and partly accessible in the cloud are considered as a socio-cyberphysical system that integrates physical space (driver and vehicle), social space (driver), and information space (smartphone with mobile application, software services, and vehicle infotainment system).

This paper extends research work presented in (Smirnov et al., 2014) that aims at context-driven on-board information support and providing the driver services needed for him/her at the moment.

The rest of the paper is structured as follows. Section 2 presents reference model of socio-cyberphysical system for driver assistance. Section 3 considers a case study that contains driver assistant system scenario, dangerous events identification diagram that is used to determine dangerous events based on information from smartphone cameras and

Smirnov, A., Kashevnik, A., Shilov, N. and Lashkov, I.
Driver Assistant in Automotive Socio-cyberphysical System - Reference Model and Case Study.
In *Proceedings of the International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS 2016)*, pages 104-111
ISBN: 978-989-758-185-4

available vehicle sensors, and implementation. Main results and findings are summarized in the conclusion.

2 REFERENCE MODEL

The reference model of the proposed automotive social-cyberphysical system is presented in Figure 1. It consists of:

- driver (belongs to social space and physical space);
- vehicle (belongs to physical space);
- smartphone (belongs to information space);
- cloud (belongs to information space);
- vehicle infotainment system (belongs to information space)

The driver interacts with the smartphone and with the vehicle while the vehicle interacts with the vehicle infotainment system. The smartphone interacts with the cloud to store generic information about the driver's behaviour and shares it with other driver assistant systems.

Information for analysing the driver behaviour is collected by the mobile application component from the front-facing and rear-facing cameras. Also, this component acquires information from vehicle sensors using vehicle infotainment system (such as the speed, location, and road signs). Internal components of the mobile application are context-aware camera switcher, application business logic, user interface, computer vision, analysis module, and computation planner. To acquire information from driver face and from the road, a context-aware algorithm is used that switches between the two cameras while processing the data in real-time. The image processing unit is responsible for extracting the visual features from the images taken by the rear and front cameras. The computation planner aims to effectively leverage the multi-core architecture of modern smartphones to perform heavy calculations. Local database is responsible for storing data collected from the smartphone. This data is synchronized with the cloud to be shared with other driver assistant systems.

Such information as smartphone characteristics, application usage statistics, and dangerous events occurred during trip is stored for using in the future. Smartphone characteristics are GPU, sensors (GPS, Accelerometer, Gyroscope, Magnetometer), cameras (front-facing / rear-facing), memory & battery capacity, and version of operation system. In addition, the cloud storage is used for keeping behaviour patterns and driving style patterns. Operations that can be carried out in the cloud storage are:

- recognition of true and false drivers estimations of dangerous events recognition;
- behavior and driving style patterns matching;
- analysis and classification of driver behavior and driving style for generating recommendations for safe driving;

When the dangerous state is determined, the driver is notified using the possibilities provided by vehicle infotainment system. The system allows to display information on vehicle screen, use audio system for sound notification, use text to speech function to provide the driver audio message, and use the steering wheel vibration to notify the driver.

The system is focused on the behavioural and physiological signals acquired from the driver to assess his/her mental state in real-time. In the presented approach, the driver is considered as a set of mental states. Each of these states has its own particular control behaviour and interstate transition probabilities. The canonical example of this type of model would be a bank of standard linear controllers (e.g., Kalman Filters plus a simple control law). Each controller has different dynamics and measurements, sequenced together with a Markov network of probabilistic transitions. The states of the model can be hierarchically organized to describe the short and long-term behaviours by using the driver ontology that includes visual cues and visual behaviours and determines relationships between them.

The vehicle drivers are faced with a multitude of road hazards and an increasing number of distractions (e.g. music, phone calls, smartphone texting and browsing, advertising information on the road, and etc.) that is described by the vehicle ontology. The driver and vehicle ontologies are described in (Lashkov et al., 2015).

3 CASE STUDY

3.1 Driver Assistance Scenario

The driver assistance scenario is shown in Figure 2. The driver assistant system uses smartphone front and rear cameras to recognize the driver's emotional state.

Vehicle infotainment system provides with the driver assistant system information gathered from the vehicle sensors (location, speed, fuel level, road signs, etc.). Driver's state (emotion, fatigue) and information from vehicle sensors together represent the current situation in the car.

This current situation is analyzed by the vehicle infotainment system and based on this analysis alerts

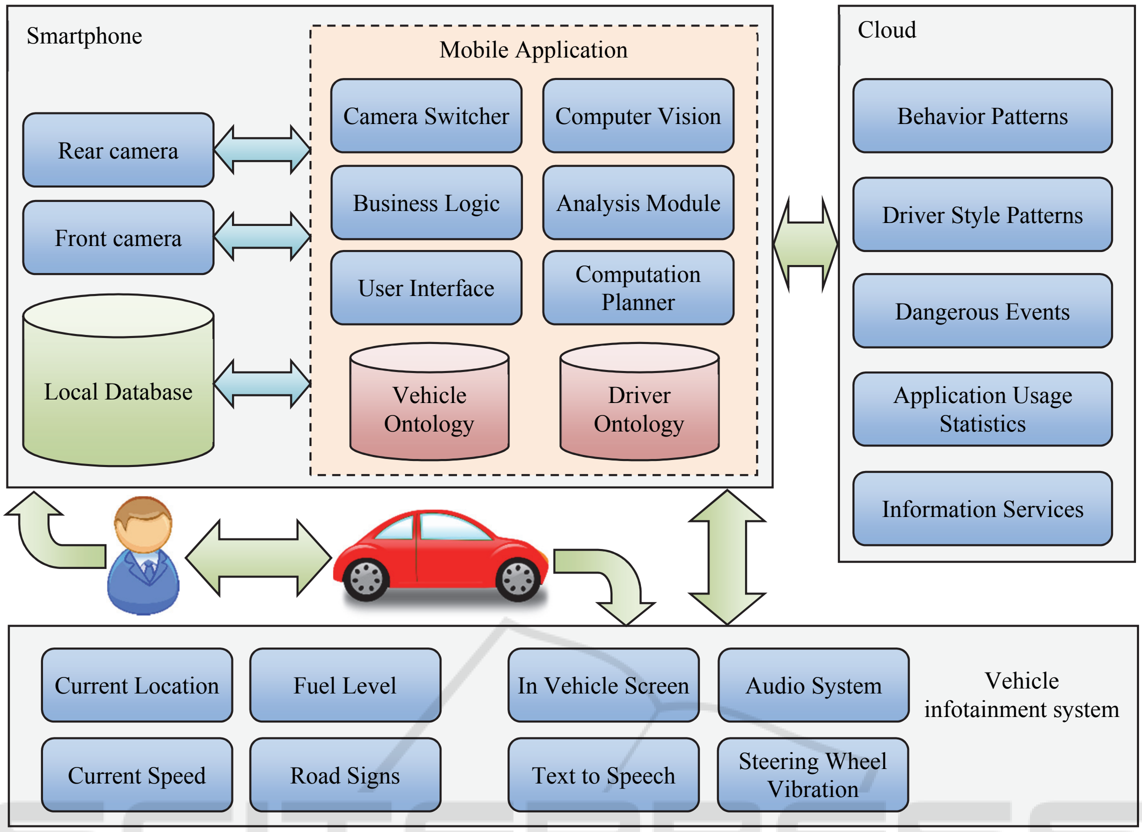

Figure 1: Reference model of the proposed automotive social-cyberphysical system.

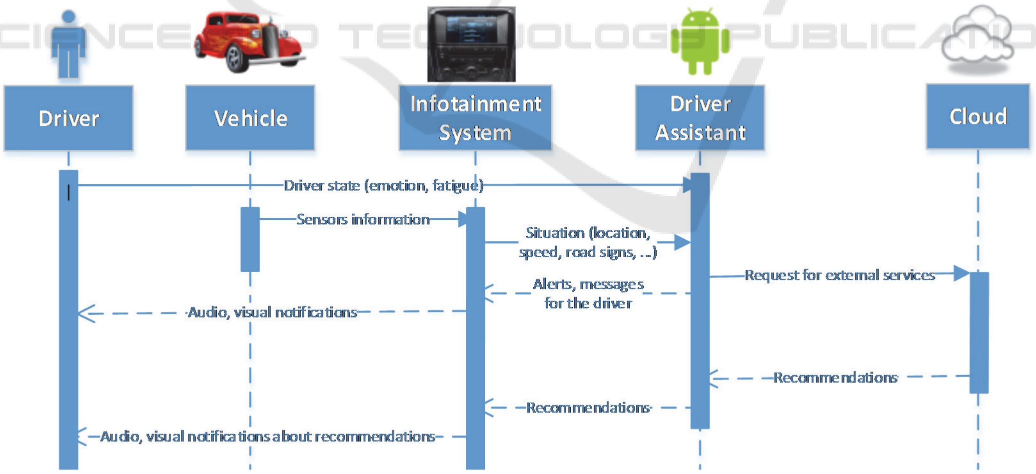

Figure 2: Driver assistance scenario.

are formed for the driver or request to the external recommendation services is created.

The driver assistant system can be considered as a sensor for activation of recommendation services in the cloud that analyses the current situation and provides the driver with recommendations to enhance the driving process (see, Smirnov A. et al., 2014).

If the driver assistant system recognizes that the driver is in a dangerous state, it tries to make alerts or recommendations to prevent an accident. E.g., lets imagine that the driver is tired and the system has determined that he/she is in a drowsiness dangerous situation. First of all, the business logic of the mobile application makes alerts for the driver through the vehicle infotainment system. If the dangerous state appears few times, the system decides to use a recommendation service in the cloud that makes context-based recommendations. E.g., service can propose to stop and drink a cup of coffee at a gas station in 5 km or stay in a hotel in 10 km.

3.2 Dangerous Events Identification

We focus at five most commonly occurring dangerous driving events: drowsiness, distraction (e.g., when the driver is distracted and takes his/her eyes off the road), tailgating (i.e., getting too close to the followed car in front), lane weaving or drifting and careless lane change. Figure 3 presents dangerous events identification diagram.

Drowsiness. As a starting point, the smartphone's front camera should be able to monitor the head position, facial expressions and the prolonged and frequent blinks indicating micro sleep. We adopt a standard metric for measuring alertness, PERcentage of CLOSure of the eyelid – called PERCLOS, which more formally represents the proportion of time within one minute that eyes are at least 80% closed. This driver state information such as PERCLOS and eye-blink speed is provided by the front-facing camera. We continuously compute PERCLOS and declare the driver as "drowsy" if PERCLOS exceeds a threshold (e.g., 28%). Another parameter is the speed of blinking giving the permissible range of 0.5-0.8 seconds per blink. One more indicator of drowsiness is yawning. If the driver makes more than 3 yawns in 30 minutes we consider the driver is in this dangerous state. Finally, the fourth indicator of this dangerous event is the head nodding. If the number of head nods exceeds a threshold (e.g., 4 in 2 minutes), the drowsiness is inferred.

Distraction. Two types of inattentive driving are monitored. In the first case, the output of the face direction classifier based on head movements and head position is tracked. If the driver's face is not facing forward for longer than 3 seconds while the car is moving forward (i.e., while aт acceleration is reported by the accelerometer) and not turning as reported by the turn detector (which is based on the gyroscope readings) then a dangerous driving event is inferred. In the second case, we monitor the turn detector. We recognize four face related categories that include: (1) no face is present; or the driver's face is either (2) facing forwards, towards the road; (3) facing to the left events (i.e., a 15° or more rotation relative to facing directly forwards); and, (4) facing to the right events (another 15° or more rotation, but this time to the right). Each time a turn is detected, the historical output of the face direction classifier is checked. If there is no a head turn corresponding to the car turning event then the driver did not check that the road is clear before turning – as a result, a dangerous event is inferred.

Tailgating. We adopt the calculation for a minimum following distance, which relies on the speed estimation and the recognition of the followed vehicles. A "safe following distance" is a distance, when the driver stays at least 2 seconds behind any vehicle that is directly in front of the driver's vehicle. If we determine the safe following distance is not respected for a period longer than 3 seconds a dangerous driving event is inferred.

Lane Weaving and Drifting. Detecting lane weaving or drifting relies on the trajectory classifier based on the lane markers and lane position of the driver's car. If the classifier infers either lane weaving or lane drifting continuously for longer than 2 seconds, which would be significantly longer than the typical duration of a lane change manoeuvre, we report a dangerous driving event.

Careless Lane Change. Each time the trajectory classifier determines a lane change event the recent inferences made by face direction classification are examined. If there is no a head turn corresponding to the lane change event and the duration of mirror checks is less than 3 seconds, occurring prior to the lane change event detection, than a dangerous driving event is inferred.

3.3 Implementation

3.3.1 Driver Face Recognition

The proposed ADAS system has been implemented for Android-based mobile devices. The driver and vehicle classification pipelines, which represent the most computationally demanding modules, are written in C and C++ based on the open source computer vision library (OpenCV library) (see, Figure 4) and interfaced with Java using JNI wrappers. Currently, using the front-facing camera, our mobile application is able to recognize two dangerous events, drowsiness and distraction. Other architectural components (dangerous driving event engine, context-driven camera switching, and multi-core computation planner) are implemented using pure Java. For the image recognition based on OpenCV library the Haar cascade classifier is called to find faces or eyes at each frame received in video sequence. It returns a vector of rectangles where each rectangle contains the detected object. Each rectangle is presented by OpenCV Rect structure (see Table 1).

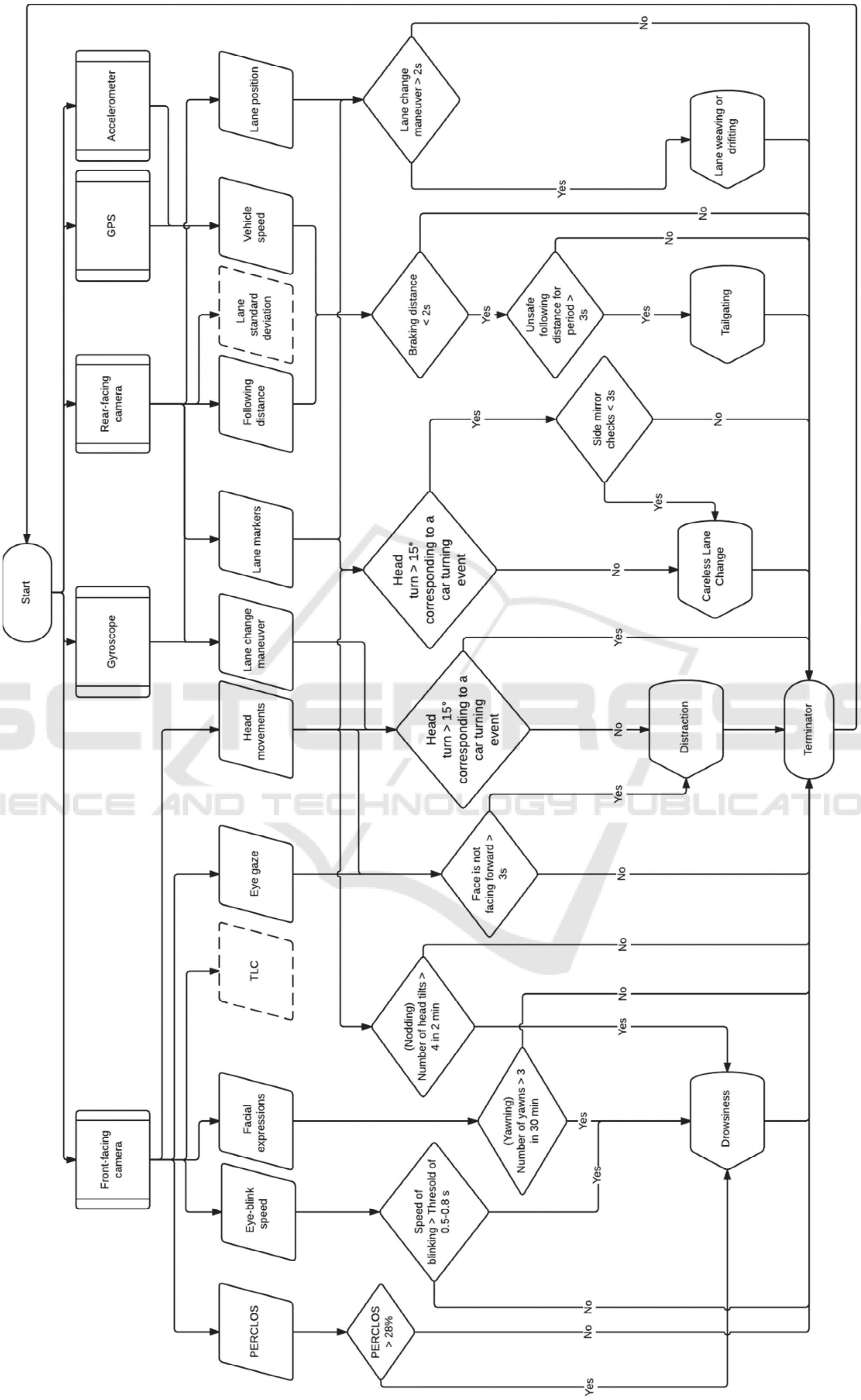

Figure 3: Dangerous Events Identification Diagram.

Table 1: OpenCV output data of face and eyes recognition.

Face				Left eye				Right eye				Time
X	Y	W	H	X	Y	W	H	X	Y	W	H	
243	108	372	372	429	190	163	124	266	190	163	124	119
243	108	372	372	429	190	163	124	266	190	163	124	60
243	108	372	372	429	190	163	124	266	190	163	124	59

The image processing module has input/output signals that allows to transmit the image taken with the camera, receive appropriate driver recommendations. Both driver and vehicle ontologies are involved in the work of the computer vision module and the analysis module. At first, camera image (Bitmap) is scanned to find and identify objects and their position that are relevant to the situation. In the next step, vectors of rectangles that contain the detected objects are passed to the Analysis module for further processing. With the help of predefined rules that cover unsafe situations and scenarios, it runs search-match calculation. If the Analysis module returns true, the system makes an appropriate alert, otherwise this event is ignored. For example, if the system determines that the driver is likely to enter a dangerous drowsy state, it makes an audible alert for a driver. Quantitative parameters helping to identify dangerous driving events are presented as follows.

- *PERCLOS.* Regarding PERCLOS, we consider, for each driver, the proportion of time within one minute that eyes are at least 80% closed. The permissible limit is equal to 30%.
- *Eye-blink Rate.* We compute for each driver the eye-blink rate as the average number of blinks per minute (the range from 8 to 21 blinks per minute).
- *Face Direction.* We classify the direction of the driver's face and recognize four face related categories that include: (1) no face is present; or the driver's face is either; (2) facing forwards, towards the road; (3) facing to the left, if the angle is greater than 15° relative to facing directly forwards; (4) facing to the right, if the angle is greater that 15°, but this time to the right.

To find a face in the image the built-in "FaceDetector" class is used. We create a face detector, which is optimized for tracking a single, relatively large face. Additionally, a "LargestFaceFocusingProcessor" face processor is applied that focuses on tracking a single "prominent face" in conjunction with the associated FaceDetector. A prominent face is defined as a face, which was initially the largest, most central face when tracking began. This face will continue to be tracked as the prominent face for as long as it is visible, even when it is not the largest face. As an optimization, once the prominent face has been identified, the associated detector is instructed to track only that face. This makes face tracking faster.

Determined visual parameters of the face is used to infer drowsiness, distraction and fatigue states. We use a rule-based approach to evaluate presented reference model. We compose our dangerous events as a collection of simpler events based on IF / THEN rules that provide an outcome. We have defined five rules that provide an output score. They are defined as follows:

- If PECLOS parameter exceeds the threshold of 30%, the system declares the driver "drowsy".
- If the driver is not facing forward for longer than three seconds while the car is moving forward, a dangerous driving event is inferred.
- If there is no a head turn corresponding to a car turning event, than the driver did not check that the road is clear before turning – as a result, a dangerous event is inferred.
- If appropriate mirror checks are not performed before lane change event, than the driver did not check blind spots before proceeding – as a result, a dangerous event is inferred.
- If the eye-blink rate of the driver doesn't correspond to the above-mentioned range, then dangerous event is determined.

There exist other rules that have not been considered in this paper. Built rules are the result of combining quantitative parameters and ontologies.

Detection Classification. Inferring the direction of the driver's face is divided into two steps: (1) detection classification and (2) direction classification. Images from the front camera are scanned to find the relative position of the driver's face. The overall image is provided to the classifier that determines if a face is present, and if so the face direction is classified.

Facial Features. Facial analysis includes a number of processing steps that attempt to detect or track the face, to locate characteristics of facial regions such as eyes, pupils if speaking more precisely, and nose, to extract and follow the movement of facial features, such as characteristic points in these regions or model facial gestures using anatomic information.

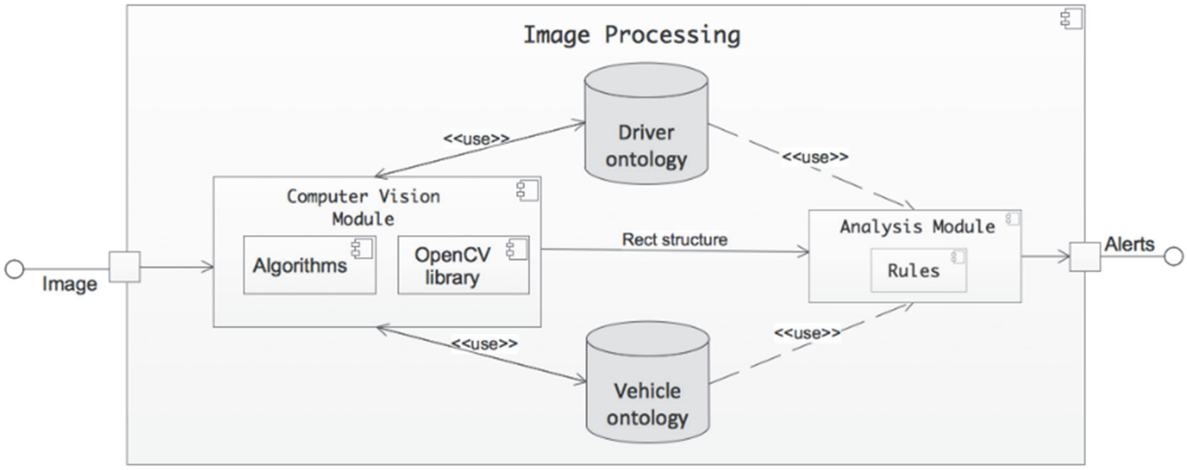

Figure 4: Image processing module implementation.

Drowsy State. Drowsy driving, a combination of sleepiness and driving, has become a worldwide problem that often leads to tragic accidents and outcomes. The smartphone's front camera should be able to monitor the prolonged and frequent blinks indicative of micro sleep. Existing research findings have shown that the percentage of closure of eyelid (a.k.a PERCLOS) is an effective indicator to evaluate the driver's drowsiness. A measure of drowsiness, PERCLOS, was generated and associated with degradation in driving performance in a simulated roadway environment.

Although PERCLOS is considered the best among other indicators in drowsiness detection, the original PERCLOS is not suitable for smart phones, since smart phones could not analyse every frame accurately or effectively. However, in an analogy to PERCLOS, we can detect fatigue by analysing a series of states of eyes, classified by a neural network to either open or closed. The states of eyes are necessary to simulate not only PERCLOS, but also the frequency of wink and the time of continuous closing eyes, which are all used in the driver alert mechanism. The faster we can acquire image of eyes and do the analysis, the less error between simulation and reality of the indicators will be.

Distraction State. By using the front camera of the phone, the head position of the driver can be used to determine an inattentive driving behaviour when the driver is not looking at the road ahead.

Two types of inattentive driving are monitored by our approach. In the first case, the output of the face direction classifier is tracked. If the driver's face is not facing forward for longer than three seconds (Breakthrough Research) while the vehicle is moving forward (i.e., while a positive speed is reported by the available smartphone sensors) and not turning as reported by the turn detector (also reported by car classification pipeline) then a dangerous driving event is inferred. However, that it's not a constant value and this parameter depends on various factors (e.g. vehicle speed, acceleration). In the second case, we monitor the turn detector. Each time a turn is detected the historical output of the face direction classifier is checked. If there is no a head turn corresponding to a car turning event then the driver did not check that the road is clear before turning – as a result, a dangerous event is inferred.

Persons's Gaze. The direction of a person's gaze is determined by two factors: face orientation (face pose) and eye orientation (eye gaze). Face pose determines the global direction of the gaze, while eye gaze determines the local direction of the gaze. Global gaze and local gaze together determine the final gaze of the person. According to these two aspects of gaze information, video-based gaze estimation approaches can be divided into a head-based approach, an ocular-based approach, and a combined head- and eye-based approach.

3.3.2 Integration with Ford Sync

To develop a vehicle module, the Android AppLink emulator v2.3 (AppLink emulator, 2015) and SyncProxy SDK (SyncProxy SDK, 2015) v.1.6.1 have been used. Figure 5 shows an example of alert visualisation on Ford SYNC system. For the alerts visualisation the following interfaces have been used: show information in the car screen (function proxy.show()) and text to speech function for providing audio information of recommended actions (function proxy.speak()).

Using in-vehicle system through Bluetooth network increases the mobile device power consumption but it does not matter because car drivers usually have mobile device chargers.

ADAS System Alert 12:57

Reduce the Speed

Limit is 50 km/h

Figure 5: Example of Smartphone ADAS system integration with Ford Sync.

4 CONCLUSIONS AND FUTURE WORK

The paper presents a reference model of socio-cyberphysical system for driver assistance in the vehicle. The reference model is based on the driver, smartphone, vehicle infotainment system, and cloud. For identification of dangerous events based on information from smartphone cameras and vehicle sensors, the special diagram has been proposed. The case study shows the process of driver face analysis results visualisation in the vehicle screen. For future work, we are planning to use Amazon AWS IoT platform for the cloud service implementation.

ACKNOWLEDGEMENTS

The presented results are part of the research carried out within the project funded by grants # 16-07-00462, # 14-07-00345, # 15-07-08092, # 15-07-08391 of the Russian Foundation for Basic Research, and programs # I.5 and # III.3 of the Russian Academy of Sciences. The work has been partially financially supported by Government of Russian Federation, Grant 074-U01.

REFERENCES

AppLink emulator, 2015, https://developer.ford.com/content/restricted/ALE_Win7_2.3_20140414.zip.

Aurichta, M., Stark, R., 2014, How to Consider Emotional Reactions of the Driver within the Development of Advanced Driver Assistance Systems (ADAS)?, Procedia CIRP, Vol. 21, pp. 70–74.

Bergasa, L., Almería, D., Almazán, J., Yebes, J., Arroyo, R., 2014, DriveSafe: an App for Alerting Inattentive Drivers and Scoring Driving Behaviors", *In Proceedings of IEEE Intelligent Vehicles Symposium*, Dearborn, MI, USA.

Biondi, F., Rossi, R., Gastaldi, M., Mulatti, C., 2014, Beeping ADAS: Reflexive Effect on Drivers' Behavior, *Transportation Research Part F*, Vol. 25, pp. 27–33.

Breakthrough Research on Real-World Driver Behavior, 2015, http://www.nhtsa.gov/Driving+Safety/Dis tracted+Driving+at+Distraction.gov/Breakthrough+Research+on+Real-World+Driver+Behavior+Released.

Lashkov, I. Smirnov, A., Kashevnik, A., Parfenov, V., 2015, Ontology-Based Approach and Implementation of ADAS System for Mobile Device Use While Driving, *6^{th} International Conference on Knowledge Engineering and Semantic Web (KESW 2015)*, Moskov, Russia, pp. 117-131.

OpenCV library, http://opencv.org, 2015.

Smirnov, A., Kashevnik, A., Shilov, N., Gusikhin, O., 2014, Context-Driven On-Board Information Support: Smart Space-Based Architecture, *The 6th International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT 2014)*, IEEE, St.Petersburg, Russia, pp. 195-200.

Smirnov, A. and Lashkov, I., 2015, State-of-the-Art Analysis of Available Advanced Driver Assistance Systems, *Proceedings of the 17th Conference of the Open Innovations Association FRUCT*, Yaroslavl, Russia, pp. 345-349.

Wang, T., Cardone, G., Corradi, A., Torresani, L., Campbell, A., 2012, WalkSafe: A Pedestrian Safety App for Mobile Phone Users Who Walk and Talk While Crossing Roads, in *Proceedings of HotMobile'13*, Article No. 5.

You, C., Lane, N., Chen, F., Wang, R., Chen, Z., Bao, T., Montes-de-Oca, M., Cheng, Y., Lin, M., Torresani, L., Campbell, A., 2013, CarSafe App: Alerting Drowsy and Distracted Drivers using Dual Cameras on Smartphones, in *Proceedings of MobiSys'13*, pp. 461-462.

SyncProxy SDK, 2015, https://developer.ford.com/content/ restricted/SyncProxy_1.6.1_Android.zip.