A Smartphone-Based Drowsiness Detection and Warning System for Automotive Drivers

Anirban Dasgupta, *Member, IEEE*, Daleef Rahman, and Aurobinda Routray, *Member, IEEE*

***Abstract*— This paper presents a smartphone-based system for the detection of drowsiness in automotive drivers. The proposed framework uses three-stage drowsiness detection. The first stage uses the percentage of eyelid closure (PERCLOS) obtained through images captured by the front camera with a modified eye state classification method. The system uses near infrared lighting for illuminating the face of the driver during night-driving. The second step uses the voiced to the unvoiced ratio obtained from the speech data from the microphone, in the event PERCLOS crosses the threshold. A final verification stage is used as a touch response within a stipulated time to declare the driver as drowsy and subsequently sound an alarm. The device maintains a log file of the periodic events of the metrics along with the corresponding GPS coordinates. The system has three advantages over existing drowsiness detection systems. First, the three-stage verification process makes the system more reliable. The second advantage is its implementation on an Android smart-phone, which is readily available to most drivers or cab owners as compared to other general purpose embedded platforms. The third advantage is the use of SMS service to inform the control room as well as the passenger regarding the loss of attention of the driver. The framework provides 93.33% drowsiness state classification as compared to a single stage which gives 86.66%.**

***Index Terms*— PERCLOS, voiced-unvoiced ratio, Android, drowsiness.**

I. Introduction

Long distance driving with monotonous driving conditions often leads to drowsiness and mental fatigue in the driver [1]. Sleep deprivation is another cause which leads to drowsiness and fatigue, which may result in road accidents and allied mishaps [2]. Hence, it is necessary to monitor the drowsiness level of the driver and alarm him when required.

A. Motivation of the Work

Most existing solutions address the issue of estimating the drowsiness levels in drivers through a single cue. Some systems require specialized hardware, which limits their use for a general population.

Manuscript received October 12, 2017; revised July 3, 2018; accepted October 24, 2018. Date of publication December 28, 2018; date of current version November 6, 2019. The Associate Editor for this article was Q. Ji. *(Corresponding author: Anirban Dasgupta.)*

A. Dasgupta and A. Routray are with the Department of Electrical Engineering, IIT Kharagpur, Kharagpur 721302, India (e-mail: anirban1828@gmail.com).

D. Rahman is with AppDynamics, Bengaluru 560008, India.

Digital Object Identifier 10.1109/TITS.2018.2879609

In our previous research [3], we have developed an image-based embedded platform for detecting drowsiness in automotive drivers solely based on eye closure rates, addressing the issues such as onboard illumination conditions, driver's head motion, etc. This method [3] provided significant accuracy as compared to similar methods which rely exclusively on PERCLOS. However, the reliability from a single image-based cue may not be extremely robust. This factor motivates us to employ the findings of Dhupati *et al.* [4] and integrate speech signals along with PERCLOS to increase the efficacy of the system.

Moreover, if the embedded platform is a smartphone, the system may reach a broader mass, by just installing the application. The smartphone has added advantages of using cellular data and network for communication, thus enabling the system to send warning messages to control rooms. This communication feature will help cab owners to maintain a record of the drowsiness levels of their drivers and take necessary actions drivers. Such an implementation may assist every single automotive vehicle as a preventive tool for accidents occurring due to drowsiness or fatigue.

B. Objectives of This Work

The objective of this work is to develop a smartphone-based solution to the drowsiness detection problem using a three-stage process. The stages are as follows:

Stage I Compute PERCLOS from the images grabbed using the front camera of the smartphone.

Stage II Compute the voiced to the unvoiced ratio (VUR) in case the PERCLOS reaches a threshold

Stage III Develop a reaction time test as a final stage verifier.

C. Advantages of the Proposed Methodology

The proposed implementation carries a lot of advantages such as

- Smartphones are prevalent these days, and hence a driver or car owner need not purchase additional hardware for the purpose, unlike existing solutions [3], [5], [6].
- The smartphones are equipped with high fidelity sensors for image and speech data acquisition which does not require connection of external add-ons.
- With the advantage of having cellular networks and Internet facilities in a smartphone, the emergency cases may be easily dealt with through SMS and allied online services.

D. Previous Approaches

First, we review out the systems for drowsiness detection based on PERCLOS. Subsequently, we review the approaches which address the driver drowsiness issues using smartphone-based solutions.

1) PERCLOS-Based Drowsiness Detection: The PERCLOS based frameworks follow the pipeline of the face and eye detection followed by the eye state classification. One of the earliest reports on drowsiness detection based on PERCLOS was the work at the Carnegie Mellon Driving Research Center by Grace *et al.* [5]. They have proposed a framework to compute PERCLOS for heavy vehicle drivers, using retinal reflection to classify the eye states. Ji *et al.* [6] have developed a drowsiness detection system using remotely located charge-coupled-device (CCD) cameras, and active infrared illuminators. They have claimed their system to be reasonably robust, reliable, and accurate as the use of active infrared helps in localizing the pupil center effectively through corneal reflections. The primary concern with these two implementations lies in the use of active infrared illumination, which irritates the eyes on prolonged exposure [7]. Hong and Qin [8] have computed PERCLOS after tracking the eyes using the CAMSHIFT algorithm, after face and eye detection using Haar classifier. They have defined a new complexity function based on which they classify the eye state. However, this function is highly dependent on illumination as it considers gray-level intensity only. Lang and Qi [9] have combined PERCLOS with average eyelid closure rates to gauge the drowsiness of the driver. The face and eye detection algorithms use skin color segmentation approach, which sometimes becomes person specific. Qing *et al.* [10] have computed PERCLOS using Haar-like features for face and eye detection, thereby classifying the eye states using an improved template matching.

From the review, it becomes evident that there is still significant scope of research in this area. Combining multiple cues of drowsiness is a feasible solution to increase the robustness of such systems. The implementations can be made for generic by exploiting the sensors present in a smartphone, to obtain an inexpensive solution for mass use. The review implies that most of the PERCLOS based methods use the Haar classifiers for face and eye detection. The differences lie in the manner of eye state classification. Moreover, these systems are built on specific embedded platforms, which limit their use to a larger group of users. With the advent of smartphones, recently the driver drowsiness detection problem has received a new light. There are very few approaches for detecting drowsiness using smartphones.

2) Smartphone-Based Drowsiness Detection: Lee and Chung [11] have developed a driver monitoring system in Android-based smartphones. The smartphone receives sensory data from the camera as well as a photoplethysmograph sensor via a wireless sensor network. They have used a dynamic Bayesian network framework for the final fatigue state evaluation, where a warning alarm is provided if the fatigue level reaches a predefined threshold.

Wan *et al.* [12] have developed a smartphone-based portable attention level monitoring and alarming system based on real-time EEG processing on mobile platforms. A major shortcoming of [12] is that the raw EEG signal will have a high level of motion artifact content during driving, which is hard to filter out online. Additionally, wearing EEG sensors while driving is a comfort issue, that may limit its practical implementation.

You *et al.* [13] have developed an Android application named CarSafe, that detects and alerts drivers to dangerous driving conditions. The drowsiness level is detected using the front camera, whereas the road conditions are tracked using the rear camera. They have achieved this using switching the primary and secondary cameras. The drowsiness detection scheme is based on PERCLOS.

Considering the limitations of EEG-based assessment on a smartphone, we prefer to use the camera-based method to compute PERCLOS and detect the drowsiness of the driver in real-time.

E. Research Issues

The following issues need attention while developing such a system:

- acquired speech data is noisy,
- illumination sensitivity of a smartphone camera is too high,
- placement of near-infrared (NIR) LED's inside the car, considering power and space requirements,
- placement of the smartphone for proper acquisition of speech as well as image,
- synchronizing the critical events with the server, while the software executes in the background.

F. Contributions

The main contribution of this work is the development of a three-stage drowsiness detection solution on an Android smartphone. This is the first attempt reported to fuse voice cues along-with PERCLOS. The significant contributions are:

- a three-stage cascaded framework,
- a new eye-state classification method,
- cross-correlation of estimated drowsiness detection metrics with standard psychometric tests,
- an Android application incorporating the total framework.

Let us have a brief introduction to the system.

G. Overview of the System

The system uses a three-stage approach. The first stage computes the PERCLOS using images captured from the front camera of the smartphone. On PERCLOS being higher than a preset threshold, the system asks the driver to say his full name. As a final stage verification, the driver is asked to touch the screen of the smartphone within 10s, once he is found to be fatigued by the earlier two stages, i.e., the PERCLOS and voice-based measures. Fig. 1 shows the overall framework of the proposed system.

The system maintains a log file, stored in the root directory of the internal memory. It stores the PERCLOS values for each minute, along with the GPS coordinates. Each time the driver is found to be drowsy he is warned with a sound alarm. In case, the driver is found to be drowsy for consecutive five times;

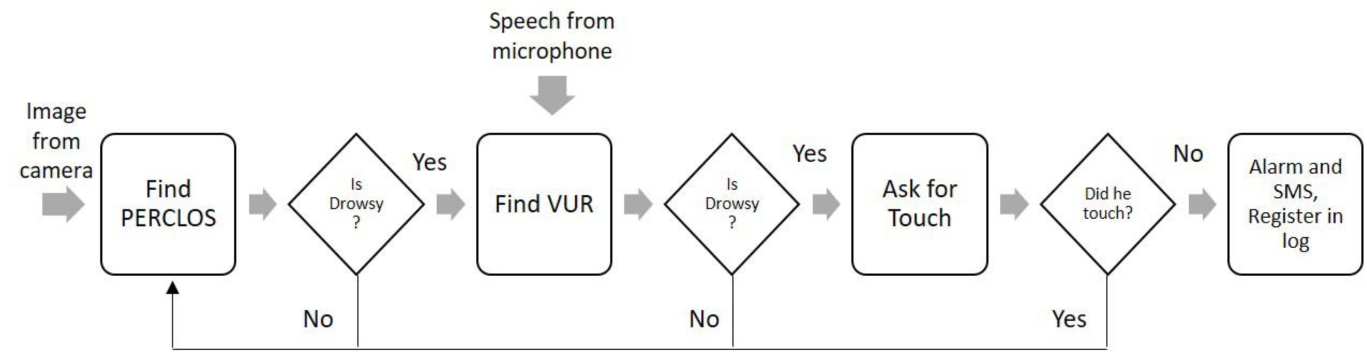


Fig. 1. The three stage framework.

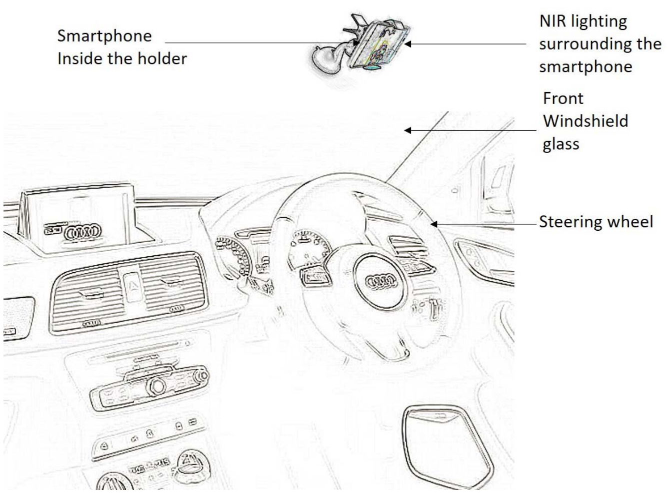


Fig. 2. The concept system.

a repeating loud alarm is sounded via the speakers of the car, connected via the Bluetooth module of the smartphone. The concept system with proper positioning is shown in Fig. 2.

II. Stage I: The PERCLOS Computation Algorithm

PERCLOS is a drowsiness metric, based on eye closure rates. It has been authenticated in [14] as a significant marker of drowsiness. PERCLOS may be defined as the proportion of time in which the eyelids are at least 80% closed over the pupil [14]. Finally, the PERCLOS value is calculated as

$$P = \frac{E_c}{E_o + E_c} \times 100\% \tag{1}$$

Here, E_c and E_o gives the counts of closed and open eyes respectively for a predefined interval. A higher value of P indicates higher drowsiness level and vice versa [1]. The steps involved in the computation of PERCLOS from an image sequence involve face detection followed by eye detection and eye state classification as given in Fig. 3.

We estimate PERCLOS using the following steps.

A. Preprocessing

Real-life driving come up with challenges, where neither the illumination, nor the head pose of the driver is constrained. We address this issue by using preprocessing for geometric and photometric corrections.

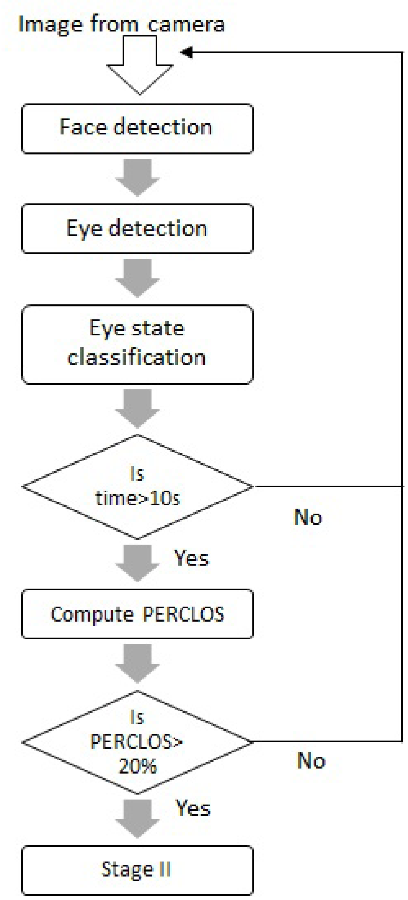


Fig. 3. Stage I framework.

1) Photometric Correction: Let the input image be $\mathbb{I}$ of size $N \times M$. The image is subdivided into blocks of size $N_h \times M_w$. We hence obtain a total number of $h \times w$ boxes, obtained as:

$$h = \frac{N}{N_h} \tag{2}$$

$$w = \frac{M}{M_w} \tag{3}$$

We have selected blocks of size 8×8 following [15]. Say we denote each subimage as $\mathbb{I}_{ij} \forall\ i \in w,\ \ j \in h$. Now each $\mathbb{I}_{ij}$ undergoes a histogram equalization to yield ${}^h\mathbb{I}_{ij}$. A contrast limiting is used to prevent the over-amplification of noise in relatively homogeneous regions of the image.

Fig. 4. Photometric Correction (a) Input Image $\mathbb{I}$, (b) Corrected Image $\mathbb{I}_f(x, y)$.

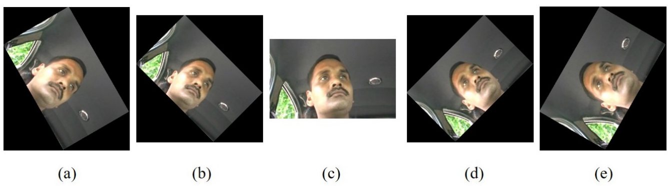


Fig. 5. Successive affine rotations of the input frame for θ = (a)-60^O, (b)-45^O, (c) 0^O, (d) 45^O, (e) 60^O, clockwise rotation is taken as the positive value of θ.

This process improves the local contrast and enhances the details in the image. However, there are some blocking artifacts at the boundaries of the equalized subimages ${}^h\mathbb{I}_{ij}$, when concatenated to form $\mathbb{I}_h$. A smoothing operation by a 5×5 Gaussian filter solves this issue. The size of the filter is optimum in the sense that it smooths out the noise without perturbing the sharp image features. The convolution operation on the image $\mathbb{I}_h$ with the Gaussian mask is given as

$$\mathbb{I}_f = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * \mathbb{I}_h \tag{4}$$

Here (*) is the convolution operation.

2) Geometric Correction: This correction is performed in the event the driver's face has a tilt of more than $\pm 30^O$ from the vertical upright face. An affine rotation of the pixels of the preprocessed image $\mathbb{I}_f(x, y)$ gives rise to a new image $\mathbb{I}_f(x', y')$ which geometrically sets the face in an upright position [3]. The pixel relations are given as:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \tag{5}$$

In the initial frame, the algorithm successively assigns the angles of $\theta = \pm 45^O$ and $\theta = \pm 60^O$ and retains the angle for which the face detection is successful. The starting angle of rotation θ for the next frames is kept the same, until a face detection fails. In such a case, the algorithm assigns the nearest rotation step. For example, if in a frame, the rotation angle is $\theta = 45^O$, and the face detection fails in the next frame. We assign $\theta = 60^O$ instead of 0^O. This theory arises from the law of Physics that the face takes significant time to rotate.

The next step is the localization of the face and eye region.

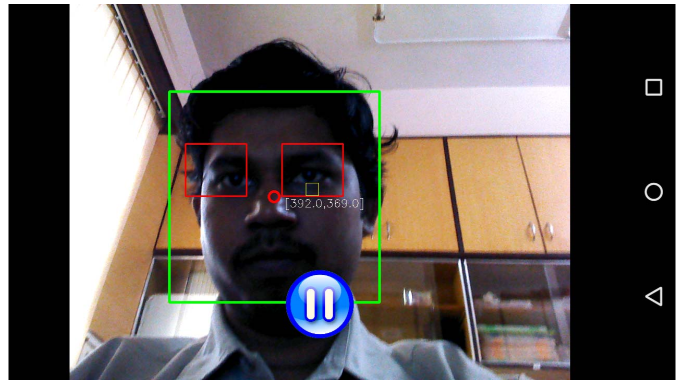


Fig. 6. Sample Face and Eye Detection on the Android device under laboratory conditions.

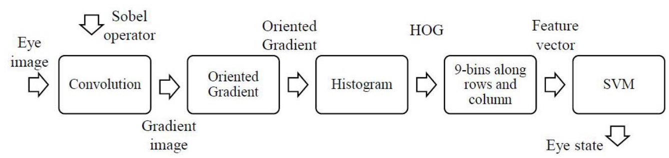


Fig. 7. The eye state classification procedure.

B. Face and Eye Detection

It is apparent that for the accurate estimation of PERCLOS, fast and precise detection of eyes is necessary. For localizing the eyes, we first localize the face region $F(x, y)$ from the preprocessed image $\mathbb{I}_f(x', y')$. This step not only reduces the search space for eye detection but also reduces the false alarms in the eye detection stage. We use a classifier based on Haar-like features for face detection trained with optimal parameters based on the findings of our earlier work [16].

From the detected face region $F(x, y)$, we search for the eyes in the upper half of the face region. We have employed a Haar classifier trained with eye images. The have trained two classifiers - one for visible image during daytime driving and the other for NIR images during nighttime driving. A sample detection on a smartphone using the classifier for daytime is shown in Fig. 6.

C. Eye State Classification

For the accurate estimation of PERCLOS, the localized eye region $E(x, y)$ needs to be accurately classified into opened or closed states. The existing eye state classification methods do not capture the significant discriminating features and hence exhibit limited accuracy. This, in turn, produces a considerable error in the estimate of PERCLOS.

Since the task is to classify the eye image as open or closed, the major difference lies in the orientations of the edgemaps of the images of two classes. The open eye images contain more edge features due to the iris than the closed eye images. The existing classification methods use the edge information, but fail to consider their orientation. Hence, we propose a new set of features based on the fusion of information of edges and their orientations. The first step in this process is to obtain the gradient image as shown in Fig. 7.

1) Gradient Image Computation: We first obtain subimages $E_{i,j}$ of size 8×8 from the eye image E with a 50% overlap

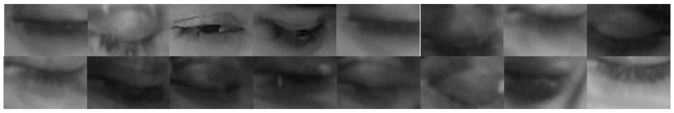


Fig. 8. Sample training images for closed eyes for SVM.

Fig. 9. Sample training images for opened eyes for SVM.

in both the directions. We pass each subimage $E_{i,j}$ through the Sobel operators S_x and S_y given as:

$$S_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{6}$$

$$S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{7}$$

The gradient images are obtained as

$$G_{x(i,j)} = E_{i,j} * S_x \tag{8}$$

$$G_{y(i,j)} = E_{i,j} * S_y \tag{9}$$

2) Orientation Computation: The edge-maps are subjected to gradient operation to find the oriented gradients. The output image magnitude M and orientation D are obtained as

$$M_{i,j} = \sqrt{G^2_{x(i,j)} + G^2_{y(i,j)}} \tag{10}$$

$$D_{i,j} = \arctan \frac{G_{y(i,j)}}{G_{x(i,j)}} \tag{11}$$

In this step, first, we create the cell histograms. A weighted vote is obtained based on the values found in the gradient computation for each pixel. We use an unsigned representation, and hence the histogram channels are evenly spread over 0 to 180 degrees.

3) Feature Computation: Finally, we form 9-bins along the horizontal as well as the vertical axes. An 81-length feature vector is hence obtained using 9-bins of rows and columns respectively. The selected features perform better over other competing features such as edge orientation histograms, scale-invariant feature transform descriptors, etc. because they use overlapping local contrast normalization to obtain improved accuracy.

4) Classification: We train a linear SVM with 1200 images (700 open and 500 closed eyes) taken from the database created using normal and NIR illumination. Some training images are shown in Fig. 9 and Fig. 8.

5) PERCLOS Computation: Once the eyes are classified as open or closed, the algorithm computes the PERCLOS value using (1) over a sliding time window of 10s duration. We use a threshold of 20% PERCLOS value to proceed to the next stage. The threshold is obtained as per the observations from Fig. 13.

The real-time algorithm is deployed into a Coolpad Note 3 mobile having a screen size of 5.5 inch, Octa-core processor, with 1.6 GHz processing speed and 3 GB RAM. The secondary camera is of 5.0 MP having an Operating System of Android Lollipop 5.1.

Fig. 10. Sample images of the dataset.

III. Stage II: Speech Based Framework

PERCLOS is indeed an authentic indicator of drowsiness as validated in [1]. However, in the present implementation, we get an estimate of P which is dependent on the eye state classification algorithm [3]. Hence, a fusion of additional cues such as speech signals can make the drowsiness detection system more reliable. Recently, the voiced-unvoiced ratio (VUR) of speech signals has been validated in [4], as an indicator of drowsiness. In our system, we use the inbuilt microphone of the smartphone to capture speech signals sampled at 20 kHz, since speech information is up to about 7.5 kHz. This speech data is processed frame-wise. The vocal fold vibrations may be assumed periodic if the signal is of short duration (10-30ms). For this reason, the speech data is processed in small frames of size 10-30 ms.

Singular Value Decomposition (SVD) is performed frame-wise to remove noise and redundant information. By constructing the speech signal using the singular vectors with the top five singular values, we can significantly reduce the surrounding noise and other sources of sound. Voiced speech is produced by the vibrations of vocal cords whereas unvoiced sounds are due to the turbulence of air in vocal tract (mouth, tongue, velum, etc.) [4]. Unvoiced speech has lower energy and higher zero crossing rates as compared to the voiced speech. Voiced and unvoiced classification is carried out using a support vector machine (SVM) [17] with the Mel frequency cepstral coefficients (MFCC) as features. MFCC represents the short-term power spectrum of the speech signal. Once the MFCCs are obtained, the SVM returns the voiced speech $\upsilon_s(n)$ and unvoiced speech $u_s(n)$f lengths N_υ and N_u respectively. The VUR is finally obtained as the ratio of the energies as

$$VUR = \frac{\sum_0^{N_\upsilon - 1} \upsilon_s^2(n)}{\sum_0^{N_u - 1} u_s^2(n)} \tag{12}$$

IV. Stage III: Touch Response and Alarm Generation

The third and final verification stage of the framework is the touch-based reaction. The work of Abdullah *et al.* [18] inspires the touch reaction test as a tool of alertness. In this stage, the driver is asked to touch the smartphone screen within a stipulated time of 10s after a voice instruction to

TABLE I

ILLUSTRATION OF THE EFFECTIVENESS OF USING A THREE-STAGE VERIFICATION

Actual	A	A	A	A	A	A	A	A	A	A	A	A	A	A	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
Predicted by P	A	A	A	A	D	A	A	A	D	A	A	A	A	A	D	D	D	D	A	D	D	D	D	D	D	A	D	D	D	D
Predicted by V					D				A						D	D	D	A		D	D	D	D	D	D		D	D	D	D
Predicted by T					A										D	D	D			D	D	D	D	D	D		D	D	D	D

do so. The threshold of 10s is obtained after observing the histogram of reaction times for alert and fatigued subjects as shown in Fig. 12. This stage is invoked when both the voice and vision based classification methods predict the driver to be drowsy. In the event the driver fails to respond within 10s, the final decision is drowsy, and the alarming sound is generated through the speakers. Along with this, an SMS is sent to an emergency number which was obtained at the start of the ride. The event is also marked in the log file stored in the internal memory of the smartphone.

A. Probablistic Model of the Framework Efficacy

Let us illustrate the overall framework in a probabilistic model. Table I shows a sample data of 30 cases sampled from the data of Experiment II. Here, the events A and D denote the user is alert or drowsy respectively. The objective is to show that how the touch verification can improve a false alarming provided by the prior two stages. On visual inspection of the table, we see that the fifth sample point was wrongly classified as D by the first two stages, but finally rectified by the touch-verification process.

Now we statistically observe the effects of the three stages in a probabilistic framework. From the table, we find that

$$\mathbb{P}(A) = \frac{14}{30} = 0.4667 \tag{13}$$
$$\mathbb{P}(D) = 1 - \mathbb{P}(A) = 0.5333 \tag{14}$$

Here, $\mathbb{P}(.)$ denotes the probability operator. Now, the probability of PERCLOS predicting drowsy given the driver is actually drowsy, $\mathbb{P}(P = D|D)$ is obtained as

$$\mathbb{P}(P = D|D) = \frac{14}{16} = 0.8750 \tag{15}$$

The probability of VUR predicting drowsy given the PERCLOS has declared drowsy, $\mathbb{P}(V = D|P = D)$ is

$$\mathbb{P}(V = D|P = D) = \frac{13}{16} = 0.8125 \tag{16}$$

Hence, the probability of VUR predicting drowsy given the driver is drowsy, $\mathbb{P}(V = D|D)$ is

$$\mathbb{P}(V = D|D) = \frac{13}{14} = 0.9286 \tag{17}$$

This indicates that the VUR stage has improved the drowsiness prediction. The final touch shows that the all drowsy cases are classified as drowsy. Similarly we find the classification of alert states may be obtained as $\mathbb{P}(P = A|A) = \frac{12}{14} = 0.8571$, with the false positive probability is $\mathbb{P}(P = D|A) = \frac{2}{14} = 0.1429$. However, these two false classifications are rectified in the subsequent stages. This statistically proves the efficacy of the framework.

TABLE II

STRUCTURE OF THE LOGFILE

Driver Id:				
Vehicle Registration Number:				
PERCLOS	**VUR**	**Reaction Time (s)**	**Alarm (Y/N)**	**GPS location**
12.5 . . . 21.5	0.78	7	N	

B. Alarm Generation and Data Log Management

The device maintains a logfile in the root of the internal storage, for each user separately in a .csv file. The logfile stores the PERCLOS values for every minute, and the VUR values when computed. It also stores the touch response times and alarm generation status. Table II shows its structure.

The system uses three modes of alarm management - the SMS service, the internet, and the sound alarm. At the start of the system, the driver enters an emergency contact number, such as of the passenger or any control room. For each event when PERCLOS crosses the threshold, the emergency number is intimated with an alert SMS, "the driver may be drowsy" along-with the Driver Id and Vehicle Registration Number. It also synchronizes the server with the GPS coordinates. This facility is useful if the cab has a service provider, who wants to track the drowsiness status of their drivers. The third mode is to generate the sound alarm when all the stages declare the driver as drowsy.

V. SYSTEM OVERVIEW

The system comprises of the two portions - the hardware and the software.

A. Hardware Description

The sub-components of the hardware are the smartphone, the NIR module, and the connecting cable. The hardware unit is shown in Fig. 11. The prototype implementation is carried out using a Coolpad Note 3 having 3GB RAM, 16GB internal memory. The smart-phone shown in Fig. 11 is surrounded by the NIR module, while is enabled during the night driving. There is a provision to switch on the NIR, using an OTG power cable. The driver needs to plug in the power cable during the night to turn on the LEDs. The application will provide a warning if the LED's are not powered during the night. The application does so if the intensity falls below a threshold of 10 lux.

B. Description of the Android Application

The developed application can work on any Android device bearing at least a Lollipop kernel i.e. kernel version of 5.0 or greater. The app uses the front camera and the microphone

Fig. 11. Testing during the day.

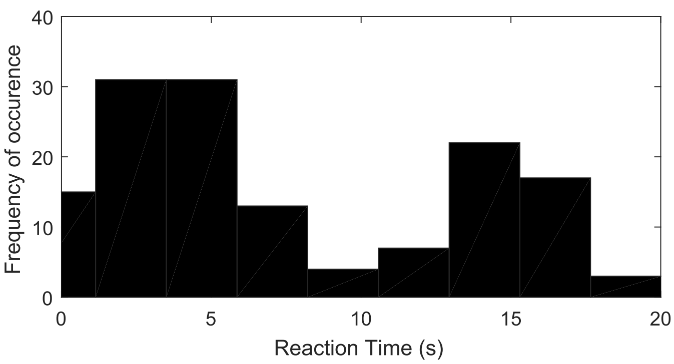


Fig. 12. Histogram of reaction times.

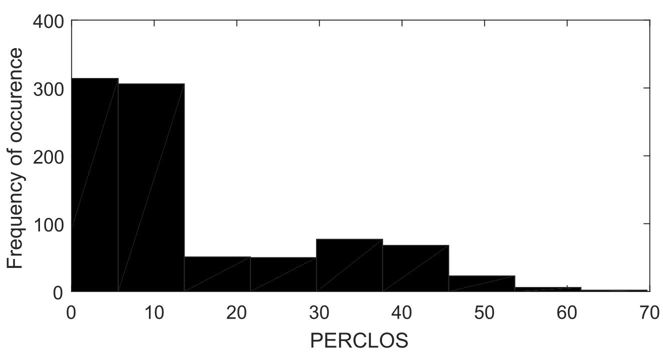


Fig. 13. Histogram of PERCLOS values to select the threshold.

of the Android device, placed on the dashboard of the car. The device sends an alarm SMS to the control room, along with the drivers, name, vehicle registration number and the GPS coordinates while he was found to be lose his attention. In case the PERCLOS is higher than a preset threshold, the system asks the driver to speak and subsequently the voice is recorded. The VUR is computed and if found lower than a preset threshold, the driver is declared as drowsy and the driver along-with the co-passengers is alarmed. A log file is also maintained which keeps the track of drowsiness behavior of the driver. This a decision log contains the registration of the level of drowsiness of the driver, his GPS coordinates, vehicle registration no. as well as his name. The control room server gets a message only when the drivers drowsiness level crosses a threshold. The log file is saved in the smartphone, and after the ride, the data can be retrieved as per the authority's wish. The application has an option to upload the log-file to a cloud server to maintain a record. This will be beneficial to cab service providers, who can maintain their records based on driver performance.

The system was field tested to evaluate the performance. Before describing the testing of the system inside a car, using a smart-phone, let us view the user interface of the application. As per the norms, only a set of registered users can use this app. The emergency number indicates the contact number of the control room or the passenger who is to be warned in the event of inattention of the driver.

VI. Testing of the System

A. Data Description

1) Experiment I: On-Board Experiment: This data was captured to evaluate the performance of face detection, eye detection and eye state classification exclusively. A dataset was prepared under on-board conditions to test the efficacy of the system. The dataset consists of 30 male professional drivers under both daylights as well as night driving conditions. The data was captured on-board to ensure a natural setting concerning illumination, head pose, and distance from the camera. Some sample images of the dataset are shown in Fig. 10. The database is named 'Invedrifac' (In-vehicle Driver's Face) database and is available in the link https://sites.google.com/site/invedrifac/home.

The drivers were asked to drive on rough roads as well as on highways. The recording was conducted in the following successive periods:

- Slot M - 9:00 am to 1:00 pm
- Slot A - 2:00 pm to 5:00 pm
- Slot E - 6:00 pm to 9:00 pm

Each recording lasted for about 15 to 20 minutes. The videos were captured at a resolution of 640 $\times$ 480 at 30 fps. The database was created under normal illumination for the day and with Near Infrared (NIR) illumination during the night. The external lighting system was a matrix of Ga-As passive NIR LEDs and was required to illuminate the face in the absence of sunlight.

2) Experiment II: Off-Board Experiment: We conducted a separate experiment off-board to cross correlate the estimated PERCLOS and VUR against standard psychometric tests as well as subjective scores. This experiment was conducted on 60 participants where 40 were alert and 20 were drowsy. The psychometric tests used are the visual response test (VRT) and an auditory response test (ART). Prior ethical permission was taken from the concerned authority to conduct the experiment. The subjects were provided with verbal instructions, and it was ensured that they have fully understood the tasks.

The protocol of this test is as follows. The subject enters the experiment room and fills up the questionnaire as an online form in a laptop. Then he performs the VRT, followed by ART and text reading task. This whole set comprised of a single stage, while we conducted 12 such stages to introduce fatigue in the subjects due to monotonous tasks.

In VRT, a green ball appears randomly at the right, left, top and bottom of the screen. The subject has to press the correct direction keys accordingly as fast as possible. The reaction

TABLE III

COMPARISON OF EYE STATE CLASSIFICATION WITH RESPECT TO ACCURACY AND SPEED; TOTAL TEST IMAGES = 1200, OPEN EYES = 595, CLOSED EYES = 605

Method	TP	FP	TN	FN	Accuracy (%)	Speed (fps)
Template matching [10]	542	53	549	56	82.40	17.5044
Detecting the iris region [8]	548	47	561	44	92.41	15.6034
Backpropagation network [19]	569	27	577	28	95.66	17.2312
Previous [3]	578	17	585	20	96.91	14.9856
Probabilistic Regression [20]	580	15	586	19	97.16	15.5342
Proposed	582	13	589	16	97.58	14.6752

time and the number of correct responses are used as the performance indicator of the test.

The ART uses an audio stimulus of a random sequence of 1's and 2's. The driver has to press the correct response as fast as possible. The reaction time and the number of correct responses are used as the performance indicator of the test.

3) Experiment III: On-Board Testing: This experiment was conducted on 9 drivers, just to observe the efficacy of the system. The objective is to obtain a table of processing of events as listed in Table IX. The driving was carried out for four hours, with no allowable breaks, except for using the washrooms.

B. Results

We test our system with respect to the performance of our proposed method in eye state classification, drowsiness classification with exclusively PERCLOS, performance of VUR computation, cross-correlation with other measures and the performance of the three-stage framework.

1) Eye State Classification: We compare the proposed method against four popular methods as well as our previous approach concerning their processing speed as well as accuracy. The frame rate is tested on the Android smartphone whose specifications are provided earlier. The accuracy is tested in MATLAB in a standard PC, having Intel i5, 2.2 GHz processor, 8 GB RAM, 64-bit system. In this comparison, we have trained the classifiers using 1200 eye images with 700 open eyes and 500 closed taken from Experiment I data. The testing was carried out on another 1200 images taken from Experiment I data of which 605 were open eyes while 595 were closed. Here, TP, FP, TN, and FN represent the number of true negatives, false positives, true negatives, and false negatives respectively for P positive samples and N negative samples. A TP means the case when an opened eye is correctly classified as opened, FP means a closed eye is wrongly classified as opened. A TN means a closed eye is classified as closed while an FN means an opened eye is classified as closed. The speed for each method is computed as an average of 20 trials. The accuracy is obtained as

$$\text{Accuracy} = \frac{TP + TN}{P + N} \times 100\% \tag{18}$$

Table III and Table IV show the performance comparisons with different algorithms and datasets respectively. The tables show the mean % accuracies and run-time performances. Table III shows that our method shows better classification accuracy, while the run-time performance is comparable with the existing ones. The compromise on the speed is acceptable, as such a critical application demands more accuracy for safety reasons. Table IV shows the testing our eye-state classification framework on the BioId eye-position Database [21], GI4E database [22], CEW [23], Hufapart [24], and our created database - Invidrifac.

TABLE IV

PERFORMANCE OF EYE STATE CLASSIFICATION ON DIFFERENT DATASETS

Database	% Accuracy	Speed (fps)
BioId [21]	95.41	14.7727
GI4E [22]	96.48	14.7727
CEW [23]	98.39	14.3967
Hufapart [24]	96.66	14.1283
Invidrifac	97.58	14.6752

TABLE V

DROWSINESS STATE CLASSIFICATION USING THE ESTIMATED PERCLOS EXCLUSIVELY; MEAN ACCURACY = $\frac{96.25+96.75+96.25+95.5+95.75}{5} = 96.10\%$

Set	Total Drowsy cases (P>20%)	Predicted Drowsy	Total alert cases (P<=20%)	Predicted Alert	Accuracy (%)
I	200	192	200	193	96.25
II	200	194	200	193	96.75
III	200	193	200	192	96.25
IV	200	192	200	190	95.50
V	200	189	200	194	95.75

TABLE VI

TESTING OF THE VUR BASED ALGORITHM; MEAN ACCURACY = $\frac{90+93.33+93.33+90}{4} = 91.67\%$

TP	FP	TN	FN	Accuracy (%)
19	2	8	1	90.00
17	1	11	1	93.33
16	1	12	1	93.33
16	2	11	1	90.00

2) Drowsiness State Classification Using PERCLOS: It can be seen that the eye state classification was reasonably appreciable using the proposed method. Now, we observe how the classification affects the PERCLOS based drowsiness state classification. We created five sets of data, from a random sampling of sampled frames of Experiment II, to compare the drowsiness state classification using the ground truth data and the predicted state. In many situations, the effect of an open eye being classified as closed and vice versa cancels out each other's effect and hence the computed PERCLOS bears no effect even with this pair of misclassifications in the eye state.

3) Testing of the VUR: The VUR was tested by creating four sets using randomly sampled data from Experiment II. Each set was tested with the proposed method and the results are tabulated in Table VI.

4) Cross Correlation With Other Measures: We cross-correlate the different measures of Experiment II to cross-validate the estimated measures. We have used Pearson's correlation coefficient to observe the similarity of measures using the mean measures. The analysis of the accuracy of the system has been carried out offline to evaluate the performance of the system. All the values have been normalized for comparison. VRT and ART indicate the response times taken by the driver to respond in the visual and auditory tasks. The Pearsons correlation coefficients are provided in Table VII.

TABLE VII
CORRELATION MATRIX SHOWING THE PEARSON'S CROSS CORRELATION COEFFICIENTS AMONG THE DIFFERENT MEASURES

Metrics	PERCLOS	VUR	VRT	ART	Subjective
PERCLOS	1	-0.645	0.786	0.751	0.812
VUR	-0.645	1	-0.699	-0.726	-0.684
VRT	0.786	-0.699	1	0.788	0.702
ART	0.751	-0.726	0.788	1	0.639
Subjective	0.812	-0.684	0.702	0.639	1

TABLE VIII
COMPARISON OF USING A SINGLE CUE (PERCLOS) AGAINST THREE CUES IN CASCADE

Method: Exclusively PERCLOS			
Sets	**Ground truth**	**Correct Predictions**	**Accuracy (%)**
Set 1	A=20, D=10	27	90.00
Set 2	A=16, D=14	28	93.33
Set 3	A=21, D=9	25	83.33
Set 4	A=17, D=13	26	86.67
Mean			88.33
Method: With PERCLOS, VUR and touch			
Set 1	A=20, D=10	29	96.67
Set 2	A=16, D=14	27	90.00
Set 3	A=21, D=9	28	93.33
Set 4	A=17, D=13	28	93.33
Mean			93.33

TABLE IX
PROCESSING OF EVENTS AS SHOWN

Hour	**Alarm on Perclos**	**Alarm on VUR**	**Alarm with touch**	**Alarm without touch**
First	5.11	3.88	1.33	0.33
Second	9.33	5.33	2.33	0.33
Third	11.33	7.33	2.33	2.33
Fourth	14.66	8.33	4.00	1.22

Here the correlation magnitude closer to one indicates a strong correspondence among measures. It can be seen that the PERCLOS, VRT, and ART are high for high subjective scores on fatigue while the VUR is low except for a few cases.

5) Performance of the Three-Stage Framework: The three-stage framework is compared against the exclusive PERCLOS-based classification technique. We divide the data of Experiment II into another four sets. The results are provided in Table VIII. We observe how the mean classification is better with our proposed three-stage cascading against the exclusive use of the popular PERCLOS-based method.

VII. CONCLUSION

We have proposed a smartphone-based drowsiness detection solution for automotive drivers. We have proposed a three-stage verification system to address the driver fatigue issue. The measures are PERCLOS, VUR, and a reaction test response of the driver on the smartphone screen. Each stage is triggered based on the decision of the preceding stage. The system maintains a register log which marks the events when the driver was found to be drowsy based on the PERCLOS, VUR and touch response. The application has an option to upload the log file to a cloud server to maintain a record. This option will be beneficial to cab service providers, who can keep their records based on driver performance. We have tested the sub-operations such as the eye state classification, PERCLOS, and VUR-based drowsiness state classification individually as well as with the combined measures and cross-correlated the estimated cues against standard cues. The device may be suitably modified to monitor the loss of attention of any person engaged in a critical safety operation. With improvements in the acquisition frame-rates of the front camera, fast ocular motions such as eye saccades may be captured, which can provide earlier indications of the onset of fatigue. An extension of this work may be tracking the road conditions using the primary camera along with the drivers drowsiness level in parallel. However, such an implementation would require a lot of multi-threading operations.

REFERENCES

[1] D. F. Dinges, "An overview of sleepiness and accidents," *J. Sleep Res.*, vol. 4, no. s2, pp. 4–14, Dec. 1995.

[2] Y.-C. Lee, J. D. Lee, and L. N. Boyle, "Visual attention in driving: The effects of cognitive load and visual disruption," *Hum. Factors, J. Hum. Factors Ergonom. Soc.*, vol. 49, no. 4, pp. 721–733, 2007.

[3] A. Dasgupta, A. George, S. L. Happy, and A. Routray, "A vision-based system for monitoring the loss of attention in automotive drivers," *IEEE Trans. Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1825–1838, Dec. 2013.

[4] L. S. Dhupati, S. Kar, A. Rajaguru, and A. Routray, "A novel drowsiness detection scheme based on speech analysis with validation using simultaneous EEG recordings," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2010, pp. 917–921.

[5] R. Grace *et al.*, "A drowsy driver detection system for heavy vehicles," in *Proc. 17th DASC AIAA/IEEE/SAE Digit. Avionics Syst. Conf.*, vol. 2, Oct./Nov. 1998, pp. I36/1–I36/8.

[6] Q. Ji, Z. Zhu, and P. Lan, "Real-time nonintrusive monitoring and prediction of driver fatigue," *IEEE Trans. Veh. Technol.*, vol. 53, no. 4, pp. 1052–1068, Jul. 2004.

[7] S. M. Michaelson, "Human exposure to nonionizing radiant energy—Potential hazards and safety standards," *Proc. IEEE*, vol. 60, no. 4, pp. 389–421, Apr. 1972.

[8] T. Hong and H. Qin, "Drivers drowsiness detection in embedded system," in *Proc. IEEE Int. Conf. Veh. Electron. Saf. (ICVES)*, Dec. 2007, pp. 1–5.

[9] L. Lang and H. Qi, "The study of driver fatigue monitor algorithm combined PERCLOS and AECS," in *Proc. Int. Conf. Comput. Sci. Softw. Eng.*, vol. 1, Dec. 2008, pp. 349–352.

[10] W. Qing, S. BingXi, X. Bin, and Z. Junjie, "A PERCLOS-based driver fatigue recognition application for smart vehicle space," in *Proc. 3rd Int. Symp. Inf. Process. (ISIP)*, Oct. 2010, pp. 437–441.

[11] B.-G. Lee and W.-Y. Chung, "Driver alertness monitoring using fusion of facial features and bio-signals," *IEEE Sensors J.*, vol. 12, no. 7, pp. 2416–2422, Jul. 2012.

[12] Z. Wan, J. He, and A. Voisine, "An attention level monitoring and alarming system for the driver fatigue in the pervasive environment," in *Brain and Health Informatics*. Springer, 2013, pp. 287–296.

[13] C.-W. You *et al.*, "CarSafe app: Alerting drowsy and distracted drivers using dual cameras on smartphones," in *Proc. 11th Annu. Int. Conf. Mobile Syst., Appl., Services*, Taipei, Taiwan, 2013, pp. 13–26.

[14] D. F. Dinges, M. M. Mallis, G. Maislin, and J. W. Powell, "Evaluation of techniques for ocular measurement as an index of fatigue and the basis for alertness management," Nat. Highway Traffic Saf. Admin., Washington, DC, USA, Tech. Rep. DOT HS 808 762, 1998.

[15] K. Zuiderveld, "Contrast limited adaptive histogram equalization," in *Graphics Gems IV*. New York, NY, USA: Academic, 1994, pp. 474–485.

[16] S. Gupta, A. Dasgupta, and A. Routray, "Analysis of training parameters for classifiers based on Haar-like features to detect human faces," in *Proc. Int. Conf. Image Inf. Process. (ICIIP)*, Nov. 2011, pp. 1–4.

[17] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statist. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.

[18] S. Abdullah *et al.*, "Cognitive rhythms: Unobtrusive and continuous sensing of alertness using a mobile phone," in *Proc. ACM Int. Joint Conf. Pervas. Ubiquitous Comput.*, 2016, pp. 178–189.

[19] W. Dong and P. Qu, "Eye state classification based on multi-feature fusion," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2009, pp. 231–234.

[20] C. Gou, Y. Wu, K. Wang, K. Wang, F.-Y. Wang, and Q. Ji, “A joint cascaded framework for simultaneous eye detection and eye state estimation,” *Pattern Recognit.*, vol. 67, pp. 23–31, Jul. 2017.

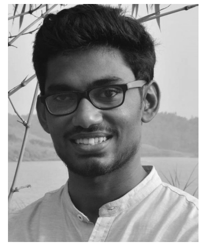
[21] O. Jesorsky, K. J. Kirchberg, and R. W. Frischholz, “Robust face detection using the Hausdorff distance,” in *Proc. Int. Conf. Audio-Video-Based Biometric Person Authentication*. Springer, 2001, pp. 90–95.

[22] A. Villanueva, V. Ponz, L. Sesma-Sanchez, M. Ariz, S. Porta, and R. Cabeza, “Hybrid method based on topography for robust detection of iris center and eye corners,” *ACM Trans. Multimedia Comput., Commun. Appl.*, vol. 9, no. 4, New York, NY, USA, pp. 25-1–25-20, 2013.

[23] F. Song, X. Tan, X. Liu, and S. Chen, “Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients,” *Pattern Recognit.*, vol. 47, no. 9, pp. 2825–2838, 2014.

[24] A. Basu, A. Dasgupta, A. Thyagharajan, A. Routray, P. Mitra, and R. Guha, “A portable personality recognizer based on affective state classification using spectral fusion of features,” *IEEE Trans. Affective Comput.*, vol. 9, no. 3, pp. 330–342, Jul./Sep. 2018.

Anirban Dasgupta received the B.Tech. degree (Hons.) in electrical engineering from the National Institute of Technology at Rourkela, India, in 2010, and the M.S. degree in electrical engineering from IIT Kharagpur, India, where he is currently pursuing the Ph.D. degree. His current research interests include computer vision, embedded systems, and signal and image processing. He became a member of the IEEE EMBS in 2014.



Daleef Rahman received the B.Tech. (Hons.) degree in electrical engineering from IIT Kharagpur, India, in 2016. He is currently a Software Engineer at Appdynamics, Bengaluru. He is an expert in Java and OpenCV-based software development specific to machine learning applications. His current research interests include Android system programming and Android application development.

Aurobinda Routray is currently a Professor with the Department of Electrical Engineering, IIT Kharagpur. His research interests include non-linear and statistical signal processing, signal-based fault detection and diagnosis, real time and embedded signal processing, numerical linear algebra, and data-driven diagnostics. More information is available at http://aroutray.org/