Drowsiness Detection System

Prof. Mayank Mangal[1] Sana Sharad Choughule[2] Shubham Mahendra Ghag[3] Rupesh Mangesh Mohite[4] Tasleem Ahemad Ansari[5]

[1,2,3,4,5]Department of Information Technology

[1,2,3,4,5]Alamuri Ratnamala Institute of Engineering and Technology (of Mumbai University) Shahapur, Thane, India

Abstract— Driver fatigue is one of the major causes of accidents in the world. Detecting the drowsiness of the driver is one of the surest ways of measuring driver fatigue. In this project, we aim to develop a prototype snooze state detection system. This system works by monitoring the eyes of the driver and sounding an alarm when he/she is drowsy. This system will also prove to be helpful for detecting drowsiness of security guard during night duty. The system to designed is a non-intrusive real-time obtrusive. The system detects the eye blinking of the driver. If the driver's eyes remain closed for more than a certain period of time, the driver is said to be drowsy and an alarm is sounded.

Keywords: Driver fatigue, driver performance, drowsiness detection

I. Introduction (Drowsiness)

Drowsy means sleepy and having low energy. Drowsiness is a position of near to sleep, a strong desire for sleep. Drowsiness refers to being unable to keep your eyes open, or feeling sleepy or tired. drowsiness is also called excess sleepiness. Drowsiness may lead to forgetfulness or falling asleep at inappropriate times. It can be accompanied by weakness lethargy, and lack of mental alertness. Depression, sorrow and stress are also associated with compromised sleep. Now drowsiness of person driving vehicle is very important. It may not be due to some medical disorders but long driving by tired driver. This may cause drowsiness so there is need to detect this to avoid miss happening. [1]

II. Driver Fatigue and Road Accidents

Driver fatigue sometimes results in road accidents every year. It is not easy to estimate the exact Amount of sleep related accidents but research presents that driver fatigue may be a contributing Reason in up to 20% in road accidents. These types of accidents are about 50% more expected to Result in death or serious hurt. They happen mainly at higher speed impacts. And the driver who has fallen asleep cannot brake. Drowsiness reduces response time which is a serious element of secure Driving. It also reduces alertness, vigilance, and concentration so that the capacity to perform Attention-based activities i.e. driving is impaired.

The speed at which information is processed is also reduced by drowsiness. The quality of decision-making may also be affected. It is clear that Drivers are aware when they are feeling sleepy, and so make a conscious decision about whether to continue driving or to stop for a rest. It may be that those who persist in driving underestimate the Risk of actually falling asleep while driving. Or it may be that some drivers choose to ignore the Risks in the way drivers drink. Crashes caused by tired drivers are most likely to happen on long Journeys on monotonous roads, such as motorways, between 2pm and 4pm especially after eating Or taking an alcoholic drink, between 2am and 6am, after having less sleep than normal, after Drinking alcohol, it driver takes medicines that cause drowsiness and after long working hours or on Journeys home after long shifts, especially night shifts.

Tiredness and fatigue can often affect a person's driving ability long before he/she even notices that he/she is getting tired. Fatigue related crashes are often more severe than others because driver's reaction times are delayed or the drivers have failed to make any maneuvers to avoid a crash. The number of hours spent driving has a strong correlation to the number of fatigue related accidents.[3],[7],[9]

III. Motivation of Detection of Problem

Driver drowsiness is a serious hazard in transportation systems. It has been identified as a direct or contributing cause of road accident. Driver drowsiness is one of the major causes of road accident. Drowsiness can seriously slow reaction time, decrease awareness and impair a driver's judgment. It is concluded that driving while drowsy is similar to driving under the influence of alcohol or drugs. In industrialized countries, drowsiness has been estimated to be involved in 2% to 23% of all crashes. Systems that detect when drivers are becoming drowsy and sound a warning promise to be a valuable aid in preventing accidents. Possible techniques for detecting drowsiness in drivers can be generally divided into the following categories:

Sensing of physiological characteristics, sensing of driver operation, sensing of vehicle response, monitoring the response of driver. Driver fatigue is a significant factor in a large number of vehicle accidents. Recent statistics estimate that annually 1,200 deaths and 76,000 injuries can be attributed to fatigue related crashes. The development of technologies for detecting or preventing drowsiness at the wheel is a major challenge in the field of accident avoidance systems. Because of the hazard that drowsiness presents on the road, methods need to be developed for counteracting its affects.

A Snooze State Detection System has been developed, using a non-intrusive machine vision based concepts. The system uses a small monochrome security camera that points directly towards the driver's face and monitors the driver's eyes in order to detect fatigue. In such a case when fatigue is detected, a warning signal is issued to alert the driver.

This paper describes how to find the eyes, and also how to determine if the eyes are open or closed. The algorithm developed is unique to any currently published papers, which was a primary objective of the project. The system deals with using information obtained for the binary version of the image to find the edges of the face, which narrows the area of where the eyes may exist. Once the face area is found, the eyes are found by computing the horizontal averages in the area.

All rights reserved by *www.ijsrd.com*

Taking into account the knowledge that eye regions in the face present great intensity changes, the eyes are located by finding the significant intensity changes in the face. Once the eyes are located, measuring the distances between the intensity changes in the eye area determine whether the eyes are open or closed. A large distance corresponds to eye closure.

If the eyes are found closed for 5 consecutive frames, the system draws the conclusion that the driver is falling asleep and issues a warning signal. The system is also able to detect when the eyes cannot be found, and works under reasonable lighting conditions.[1],[6]

A. Related work

In 2005 Tiesheng Wang et. al., had developed a system based on yawning detection for determining driver drowsiness. A system had an aim to detect driver drowsiness or fatigue on the base of video analysis which was presented. The main object of this study was on how to extract driver yawning. A real face detector was implemented to trace driver's face region. In this study, mouth window was traced. In which face region and degree of mouth openness was extracted to find driver yawning in video. This method was computationally capable because it ran at real-time on average. When the driver moved his head away by lack of concentration, the eyes and mouth might be occluded and might be detected. There was another situation should be reminded of the driver. For this other method must be found to deal with it.[8],[9]

B. Need of the new system

As shown in fig 1, around 60% of accidents are caused because of the drowsy, distracted driving. Unknown other sources are nothing but the potential disturbances as we discussed earlier.

The development of technologies for detecting or preventing drowsiness at the wheel is a major challenge in the field of accident-avoidance systems. Because of the hazard that drowsiness presents on the road, methods need to be developed for counteracting its affects.

The aim of this project is to develop a prototype drowsiness detection system. The focus will be placed on designing a system that will accurately monitor the open or closed state of the driver's eyes in real-time.

By monitoring the eyes, it is believed that the symptoms of driver fatigue can be detected early enough to avoid a car accident. Detection of fatigue involves the observation of eye movements in a sequence of video frames. And on the basis of face detection system detects the attentiveness of driver.

C. Making integrated system which will work as follow:

- Installing a camera in car which will continuously keep track on face and eye of driver.
- Camera should not distract the driver.
- By observing movement of eyes of a driver, system should detect drowsiness and should provide accurate result.
- Alert the driver while he is not paying attention on driving.

D. System design for Drowsiness Detection

For our project face and eye classifiers are required. So, we used the learning objects method to create our own haarclassifier .xml files. The Haarcascade files are one for the face and one for the eyes.

Around 1000 positive and 1500 negative samples were taken. It took a long time to train them. Finally, face.xml and Haarcascade_eye.xml files are created.

These xml files are directly used for object detection using haardetectobjects () function. It detects a sequence of objects (in our case our face and eyes). Haarcascade-eye.xml is designed only for open eyes. So, when eyes are closed the system doesn't detect anything. This is a blink. When a blink lasts for more than 5 frames, the driver is judged to be drowsy and an alarm is sounded.[8]

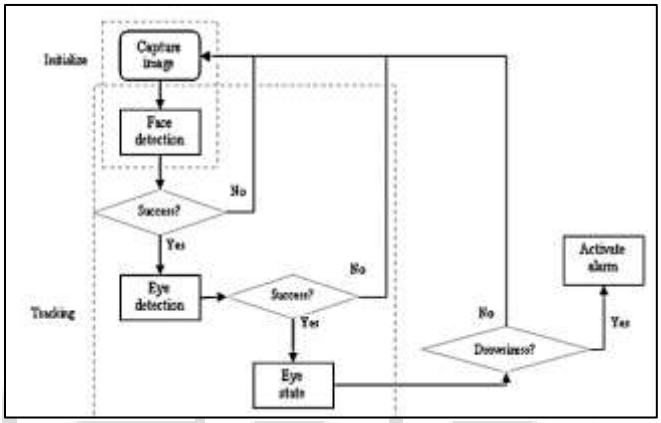

Fig. 1: system design for DDS

IV. ALGORITHM AND IMPLEMENTATION

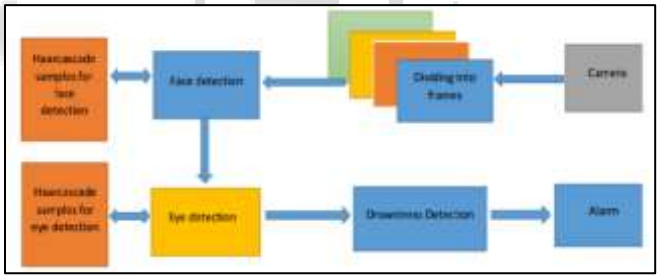

Fig. 2: Block diagram for DDS

Drowsiness of a person can be measured by the extended period of time for which his/her eyes are in closed state. In our system, primary attention is given to the faster detection and processing of data. The number of frames for which eyes are closed is monitored. If the number of frames exceeds a certain value, then a warning message is generated on the display showing that the driver is feeling drowsy.

In our algorithm, first the image is acquired by the webcam for processing. Then we use the Haarcascade file face.xml to search and detect the faces in each individual frame. If no face is detected then another frame is acquired. If a face is detected, then a region of interest in marked within the face. This region of interest contains the eyes. Defining a region of interest significantly reduces the computational requirements of the system. After that the eyes are detected from the region of interest by using Haarcascade_eye.xml[3],.[9]

All rights reserved by *www.ijsrd.com*

A. Algorithm

The algorithm is as follows: -
step1: START
step2: Program initialization
step3: System check
step4: Capture image
step5: Face Top and Width Detection
step6: Find intensity changes
step7: Track Eye position
step8: Check Eye state
step8.1: If white pixels are more > then driver is Active
step8.2: If black pixels are more > then driver is Drowsy
step9: Recognition of whether Eyes are open/closed.
step10: Calculation of criteria for judging Drowsiness.
step11: Driver Drowsy? / Driver distracted?
step12: Initiate Alarm
step13: STOP

B. Image Acquisition

The function cvCaptureFromCAM allocates and initialized the CvCapture structure for reading a video stream from the camera. CvCapture* cvCaptureFromCAM (int index); Index of the camera to be used. If there is only one camera or it does not matter what camera to use, -1 may be passed.
cvSetCaptureProperty Sets camera properties for example
cvSetCaptureProperty (capture, CV_CAP_PROP_FRAME_WIDTH, 280);
cvSetCaptureProperty (capture, CV_CAP_PROP_FRAME_HEIGHT, 220);
The function cvQueryFrame () grabs a frame from a camera or video file, the function cvQueryFrame () grabs a frame from a camera or video file, decompresses it and returns it. This function is just a combination of Grab Frame and Retrieve Frame, but in one call. The returned image should not be released or modified by the user. In the event of an error, the return value may be NULL.

C. Dividing into frames

We are dealing with real time situation where video is recorded and has to be processed. But the processing or application of algorithm can be done only on an image. Hence the captured video has to be divided into frames for analyzing.

D. Face detection

In this stage we detect the region containing the face of the driver. A specified algorithm is for detection of face in every frame. By face detection we mean that locating the face in a frame or in other words finding location of facial characters through a type of technology with the use of computer. The frame may be any random frame. Only facial related structures or features are detected and all others types of objects like buildings, tree, bodies are ignored.

We know that face is also a type of object. So, we can consider detection of face as a particular case of object detection. In this type of object type of class detection, we try to know where the objects in the interest image are located and what is their size which may belongs to a particular class. The work of algorithm that is made for face detection is mostly concentrated on finding the front side of the face. But the algorithms that are developed recently focus on more general cases. For our case it may be face in the tilted position or any other portion of the faces and also it finds the possibility of multiple faces. Which means the rotation axis with respect to the present observer from the reference of face in a particular? Or even if there is vertical rotation plane then also it is able to solve the purpose. In new type of algorithm, it is considered that the picture or video is a variable which means that different condition in them like hue contrast may change its variance. The amount of light may also affect. Also, the position of the input may vary the output. Many calculations actualize the face-detection assignment as a two-way pattern-differentiation task.
Cascade is loaded by:

```
cascade = (CvHaarClassifierCascade*)cvLoad(
cascade_name,
0, 0, 0 );
```

storage is allocated:

```
CvMemStorage* storage = cvCreateMemStorage(0);
CvSeq* cvHaarDetectObjects(
const CvArr* image,
CvHaarClassifierCascade* cascade,
CvMemStorage* storage,
double scale_factor = 1.1,
int min_neighbors = 3,
int flags = 0,
CvSize min_size = cvSize(40,40)
);
cvRectangle(
img,
cvPoint(face->x, face->y),
cvPoint(
face->x + face->width,
face->y + face->height
),
CV_RGB(0, 255, 0),
1, 8, 0
)
```

The above function draws a rectangle in the image for the corresponding corner points. the other parameter is for drawing color and thickness and type of lines in the rectangle.

1) Set Image Region of interest: -

```
cvSetImageROI(
img, /* the source image */
cvRect(
face->x, /* x = start from leftmost */
face->y + (face->height)/5, /* y = a few pixels from the top */
face->width, /* width = same width with the face */
(face->height)/3 /* height = 1/2 of face height */
)
```

It sets the ROI for the corresponding image. We have taken from the left most point in the face to a few pixels from the top to half of face height. Width of the region is same as that of the face. The rectangular region is now used to get eyes.

E. Recognition of face region

To detect the region of face after minimizing the background extra portion from the image, labelling method is used. In the

All rights reserved by www.ijsrd.com

labelling method, components are connected in 2-D binary image. This function will return a matrix of the same size as binary image. It contains labels for the connected objects in binary image. it can have a value of either 4 where 4 specifies 4connected objects or 8 where 8 specifies 8-connected objects. If the argument is omitted, it defaults to 8. The elements of matrix are integer values greater than or equal to 0. The pixels labeled 0 are the background. The pixels labelled 1 make up one object; the pixels labelled 2 make up a second object and so forth.

After that it is required to measure the properties of image regions. The region property's function is used. This function measures a set of properties for each labelled region in the label matrix.

Positive integer elements of matrix correspond to different regions. For example, the set of elements of matrix equal to 1 corresponds to region 1; the set of elements of matrix equal to 2 corresponds to region 2; and so on. The return value is a structure array of length matrix. The fields of the structure array denote different measurements for each region, as specified by properties. Properties can be a comma-separated list of strings, a cell array containing strings, the single string 'all', or the string 'basic'. There are set of valid property strings. Property strings are case insensitive. This rectangle box is used to store different operated images. And these operated images are three kind of. First operated image which is stored as matrix in rectangle box is cropped cb-cr image. Second image is gray image which is converted from colour image. Third image is histogram equalization image which is stored as matrix in rectangle box.[2]

F. Eye Detection: -

In our method eye is the decision parameter for finding the state of driver. Though detection of eye may be easier to locate, but it's really quite complicated. At this point it performs the detection of eye in the required particular region with the use of detection of several features. Generally, Eigen approach is used for this process. It is a time taking process. When eye detection is done then the result is matched with the reference or threshold value for deciding the state of the driver.

Poor contrast of eyes generally creates lots of problems in its detection. After successful detection of face eye needs to be detected for further processing. In our method eye is the decision parameter for finding the state of driver.[2]

G. Binarization: -

The first step to localize the eyes is binarizing the picture. Binarization is converting the image to a binary image.

A binary image is an image in which each pixel assumes the value of only two discrete values. In this case the values are 0 and 1, 0 representing black and 1 representing white. With the binary image it is easy to distinguish objects from the background. The grayscale image is converting to a binary image via thresholding. The output binary image has values of 0 (black) for all pixels in the original image with luminance less than level and 1 (white) for all other pixels. Thresholds are often determined based on surrounding lighting conditions, and the complexion of the driver. After observing many images of different faces under various lighting conditions a threshold value of 150 was found to be effective. The criteria used in choosing the correct threshold was based on the idea that the binary image of the driver's face should be majority white, allowing a few black blobs from the eyes, nose and/or lips. An example of an optimum binary image for the eye detection algorithm in that the background is uniformly black, and the face is primary white. This will allow finding the edges of the face.[8]

H. Face Top and Width Detection

The next step in the eye detection function is determining the top and side of the driver's face. This is important since finding the outline of the face narrows down the region in which the eyes are, which makes it easier (computationally) to localize the position of the eyes. The first step is to find the top of the face. The first step is to find a starting point on the face, followed by decrementing the y-coordinates until the top of the face is detected. Assuming that the person's face is approximately in the center of the image, the initial starting point used is (100,240). The starting x-coordinate of 100 was chosen, to ensure that the starting point is a black pixel (no on the face). The following algorithm describes how to find the actual starting point on the face, which will be used to find the top of the face.

1) Starting at (100,240), increment the x-coordinate until a white pixel is found. This considered the left side of the face.
2) If the initial white pixel is followed by 25 more white pixels, keep incrementing x until a black pixel is found.
3) Count the number of black pixels followed by the pixel found in step2, if a series of 25 black pixels is found, this is the right side.
4) The new starting x-coordinate value (x1) is the middle point of the left side and right side.

Using the new starting point (x1, 240), the top of the head can be found. The following is the algorithm to find the top of the head:

1) Beginning at the starting point, decrement the y-coordinate (i.e.; moving up the face).
2) Continue to decrement y until a black pixel is found. If y becomes 0 (reached the top of the image), set this to the top of the head.
3) Check to see if any white pixels follow the black pixel.
    1) If a significant number of white pixels are found, continue to decrement y.
    2) If no white pixels are found, the top of the head is found at the point of the initial black pixel.

Once the top of the driver's head is found, the sides of the face can also be found.

Below are the steps used to find the left and right sides of the face.

1) Increment the y-coordinate of the top (found above) by 10. Label this y1 = y + top.
2) Find the center of the face using the following steps:
    1) At point (x1, y1), move left until 25 consecutive black pixels are found, this is the left side (lx).
    2) At point (x1, y1), move right until 25 consecutive white pixels are found, this is the right side (rx).
    3) The center of the face (in x-direction) is: (rx – lx)/2. Label this x2.
3) Starting at the point (x2, y1), find the top of the face again. This will result in a new y-coordinate, y2.

All rights reserved by *www.ijsrd.com*

4) Finally, the edges of the face can be found using the point (x2, y2).
   1) Increment y-coordinate.
   2) Move left by decrementing the x-coordinate, when 5 black consecutive pixels are found, this is the left side, add the x-coordinate to an array labeled 'left_x'.
   3) Move right by incrementing the x-coordinate, when 5 black consecutive pixels are found, this is the right side, add the x-coordinate to an array labeled 'right_x'.
   4) Repeat the above steps 200 times (200 different y-coordinates).

The result of the face top and width detection is shown in Figure 7.4, these were marked on the picture as part of the computer simulation the edges of the face are not accurate. Using the edges found in this initial step would never localize the eyes, since the eyes fall outside the determined boundary of the face. This is due to the blobs of black pixels on the face, primarily in the eye area.[2],[7],[8]

I. Removal of Noise

The removal of noise in the binary image is very straightforward. Starting at the top, (x2,y2), move left on pixel by decrementing x2, and set each y value to white (for 200 y values). Repeat the same for the right side of the face. The key to this is to stop at left and right edge of the face; otherwise, the information of where the edges of the face are will be lost.

After removing the black blobs on the face, the edges of the face are found again.

As seen below, the second time of doing this results in accurately finding the edges of the face.[2],[4].[7].[8]

J. Finding Intensity Changes on the Face

The next step in locating the eyes is finding the intensity changes on the face. This is done using the original image, *not* the binary image. The first step is to calculate the average intensity for each y – coordinate. This is called the horizontal average, since the averages are taken among the horizontal values. The valleys (dips) in the plot of the horizontal values indicate intensity changes. When the horizontal values were initially plotted, it was found that there were many small valleys, which do not represent intensity changes, but result from small differences in the averages. To correct this, a smoothing algorithm was implemented. The smoothing algorithm eliminated and small changes, resulting in a more smooth, clean graph. After obtaining the horizontal average data, the next step is to find the most significant valleys, which will indicate the eye area. Assuming that the person has a uniform forehead (i.e.; little hair covering the forehead), this is based on the notion that from the top of the face, moving down, the first intensity change is the eyebrow, and the next change is the upper edge of the eye, as shown below.

The valleys are found by finding the change in slope from negative to positive. And peaks are found by a change in slope from positive to negative. The size of the valley is determined by finding the distance between the peak and the valley. Once all the valleys are found, they are sorted by their size.[2],[4]

K. Detection of Vertical Eye Position

The first largest valley with the lowest y – coordinate is the eyebrow, and the second largest valley with the next lowest y-coordinate is the eye.

This process is done for the left and right side of the face separately, and then the found eye areas of the left and right side are compared to check whether the eyes are found correctly. Calculating the left side means taking the averages from the left edge to the center of the face, and similarly for the right side of the face. The reason for doing the two sides separately is because when the driver's head is tilted the horizontal averages are not accurate. For example, if the head is tilted to the right, the horizontal average of the eyebrow area will be of the left eyebrow, and possibly the right-hand side of the forehead.[2],[8]

L. Determining the State of the Eyes

The state of the eyes (whether it is open or closed) is determined by distance between the first two intensity changes found in the above step. When the eyes are closed, the distance between the y – coordinates of the intensity changes are larger if compared to when the eyes are open.

The limitation to this is if the driver moves their face closer to or further from the camera. If this occurs, the distances will vary, since the number of pixels the face takes up varies, as seen below. Because of this limitation, the system developed assumes that the driver's face stays almost the same distance from the camera at all times.[2]

M. Judging Drowsiness

When there are 5 consecutive frames find the eye closed, then the alarm is activated, and a driver is alerted to wake up. Consecutive number of closed frames is needed to avoid including instances of eye closure due to blinking. Criteria for judging the alertness level on the basis of eye closure count is based on the results found in a previous study.[8]

N. Judging Driver not paying attention:

If user is not paying attention to driving like talking with neighboring person or talking on mobile for consecutive 5 frames then "Please pay attention" alert will be raised.

V. CONCLUSION AND RESULT:

This is the study that uses a large set of spontaneous facial expressions for the detection of drowsiness. Previous approaches to drowsiness detection primarily make pre-assumptions about the relevant behavior, focusing on blink rate, eye closure, and yawning. Here there is implementation of learning methods to determine actual human behavior during drowsiness episodes. This study reveals that facial expressions are very reliable indicators of driver drowsiness and facial expressions can be used to do fine discrimination in the different levels of drowsiness and reliably predict the time to crash. In laboratory conditions computer vision expression recognition systems can be used to reliably detect drowsiness and predict crash with high reliability. Field studies are needed to evaluate the performance of these systems in actual driving environments. Spontaneous facial expressions under drowsiness are very different from posed expressions of drowsiness. A non-invasive system is able to localize the eyes and monitor fatigue was developed. During

All rights reserved by *www.ijsrd.com*

the monitoring, the system is able to decide if the eyes are opened or closed.

Thus, we have successfully designed a drowsiness detection system using OpenCV software and Haar Classifiers. The system so developed was successfully tested

To obtain the result a large number of videos were taken and their accuracy in determining eye blinks and drowsiness was tested.

For this project we used a 0.3megapixel webcam connected to the computer. The webcam required white LEDs attached to it for providing better illumination. In real time scenario, infrared LEDs should be used instead of white LEDs so that the system is non-intrusive.

Buzzer is used to produce alert sound output in order to wake up the driver when drowsiness exceeds a certain threshold.

The system was tested for different people in different ambient lighting conditions (daytime and night-time). When the webcam backlight was turned ON and the face is kept at an optimum distance, then the system is able to detect blinks as well as drowsiness with more than 90% accuracy. This is a good result and can be implemented in real-time systems as well.

Sample outputs for various conditions in various images are given below. Videos were taken; in which both face and eyes were detected. Though the two processes have relatively equal accuracy.

Fig. 1: when eyes are closed alarm generates

Fig. 2: when eyes are open

For future work, a driver's distraction identification system will be developed. Therefore, fig.3 shows an example system which validates the results.

A. Result of eye tracking

Seq. no	Drowsiness Detection System		
	Total frames	Tracking Failure	Correct Rate
1.	500	8	98.40%
2.	960	20	97.91%
3.	330	14	95.75%
4.	900	30	96.60%

ACKNOWLEDGMENT

Success of any work depends upon the dedication, sincerity and hard work. It also requires some ingredients such as motivation, guidance and encouragement. Wholehearted efforts altogether make the project useful and meaningful.

While completion of the project, I came across a number of people whose contributions in various ways helped my field of research and they deserve special thanks. It is a pleasure to convey my gratitude to all of them.

First of all I extend my deepest gratitude to our reverent Principal Prof. S. N. Barai for his support.

I am grateful to Prof. Mayank Mangal, HOD(IT)for providing us such an imminent responsibility as well as for guiding me at every step of my work that helped me to complete my work successfully.

I extend my sincere thanks to Project Coordinator Prof. Mayank Mangal and all the faculties of the department for their valuable guidance during development of my work.

I also would like to thank my colleagues, lab assistants and my family members whose constant support and inspiration led me to complete this thesis.

REFERENCES

[1] [1] B Sangle, R Rathore, A Rathod, A Yadav, ; V Yadav, A Varghese, S Shenoy, K P Ks, ; R Remya, Patra ISCAIE 2018 -2018 IEEE Symposium on Computer Applications and Industrial Electronics, volume 2018, p. 339 – 344 Posted: 2018

[2] R. Fusek, MRL Eye Dataset, May 2019, [online] Available: http://mrl.cs.vsb.cz/eyedataset.

[3] Chen, Y.; Lin, C.; Lin, C.; Chuang, H.; Wang, L. Electronic commerce marketing-based social networks in evaluating competitive advantages using SORM. *Int. J. Soc. Humanist. Comput.* 2017, *2*, 261–277.

[4] Azmat, M.; Kummer, S. Potential applications of unmanned ground and aerial vehicles to mitigate challenges of transport and logistics-related critical success factors in the humanitarian supply chain. *AJSSR* 2020, *5*, 3

[5] Wintersberger, S.; Azmat, M.; Kummer, S. Are We Ready to Ride Autonomous Vehicles? A Pilot Study on Austrian Consumers' Perspective. *Logistics* 2019, *3*, 20.

[6] Azmat, M.; Kummer, S.; Moura, L.T.; Gennaro, F.D.; Moser, R. Future Outlook of Highway Operations with Implementation of Innovative Technologies Like AV, CV, IoT and Big Data. *Logistics* 2019, *3*, 15.

[7] Kyong Hee Lee, Whui Kim, Hyun Kyun Choi, Byung Tae Jan. A Study on Feature Extraction Methods Used to Estimate a Drivers Level of Drowsiness, IEEE, February 2019.

[8] Fouzia, Roopalakshmi R, Jayantkumar A Rathod, Ashwitha S, Supriya K, Driver Drowsiness Detection System Based on Visual Features. , IEEE, April 2018.

[9] Kofi Sarpong Adu-Manu, Nadir Adam, Cristiano Tapparello, Hoda Ayatollahi, and Wendi Heinzelman. 2018. Energy-harvesting wireless sensor networks (EH-WSNs): A review. ACM Trans. Sen. Netw. 6, 2, Article 10 (4 2018), 50 pages.

All rights reserved by *www.ijsrd.com*