In-Vehicle Occupancy Detection And Classification Using Machine Learning

Malneedi Vamsi
Electronics and Communication Engineering
Amrita Vishwa Vidyapeetham
Coimbatore, India
malneedi.vamsi@gmail.com

K.P. Soman
Centre for Computational Engineering and Networking
Amrita Vishwa Vidyapeetham
Coimbatore, India
kp_soman@amrita.edu

Abstract—Occupancy detection is a difficult problem. There are several mechanisms exists for occupancy detection in vehicles, particularly in Automobiles. Now, safety has become an important and necessary aspect of the automobile industry. Airbag became a basic and important safety measure in cars. Even though airbag is a vehicle safety device, it can kill children below 12 years due to its rapid action by the exerting lot of force. This project explains about detecting the number of passengers sitting in the car and then classifying each person whether he/she is a child or an adult by processing the image taken from the camera. So that the deployment of airbags can be avoided near children. Each time car speeds from 0 Kmph to 20 Kmph, occupancy of the car is determined and each one is classified again. We are using widely used technique Haar Cascades, for detection. First, we detect faces and then classify each occupant adult or child.

Keywords—occupancy, human detection, adult/child classification, Haar cascade, face detection

I. Introduction

Automotive safety has become a high priority in the automobile industry today as vehicle population is increasing rapidly which is also increasing the risk of road casualties in both passenger and commercial vehicles which are reinforcing the demand for safety systems[1][2]. Every stakeholder is acknowledging the importance of occupant safety and upgrading their technologies to provide fail-safe systems that can protect both occupants and pedestrians. As everyone is moving towards Autonomous car, safety became a major concern. Safety systems are classified into Active and Passive safety systems. Active safety systems try to prevent the crashes, where the passive safety systems try to protect occupants when a crash happens. Night Vision System (NVS), Tire Pressure Monitoring System (TPMS), Adaptive Cruise Control (ACC), Head-Up Display (HUD), Blind Spot Detection (BSD), Electronic Stability Control (ESC), Automatic Emergency Braking (AEB), Anti-Lock Braking Systems (ABS), Driver Monitoring System (DMS), Road Sign Recognition (RSR) and Lane Departure Warning System (LDWS) are common Active Safety Systems. Whiplash Protection System (WPS), Seat belt (SB), Airbag (AB), Occupant Sensing System (OSS), Child Safety Systems (CSS), Pedestrian Safety Systems (PSS) are common Passive Safety Systems. For years, the seat belt is solely used as passive restraint in cars. Even though seat belt is having providing partial safety, especially for children, but over a while, almost all of the countries made seat-belt laws mandatory. Statistics have shown that thousands of lives were saved by the use of seat belts which might have been lost in crashes. Later airbags have been introduced and reduced 30 per cent risk of dying in a direct frontal crash[3]. Soon after, seat-mounted and door-mounted side airbags are introduced. Presently, some cars are going far off having two airbags to having till eight airbags. Airbag tries to slow down the occupant's speed to zero either with little or no damage. It tries to slow down the occupant's forward motion as much faster as possible like the blink of an eye. It is made of nylon fabric that is folded into the dashboard or steering wheel or the seat or door. The airbag inflates when a collision occurs and force equal to running into a brick wall at 16 to 24 kilometres per hour (10 - 15). Airbag bursts at a speed of up to 200 kilometres per hour (322 miles per hour)[4]. The airbag can severely injure or kill a child if he/she is sitting close to it. Children of age 12 and under, sitting in the front passenger seat of a car can get injured severely or even die if an airbag gets deployed during a crash. Infants under the age of 1 year and weight less than 10kg (20 pounds) should never sit in the front seat. Due to concerns about children, dealers are installing on/off switches for one or both airbags in the car that controls the activation of airbags during the crash.

Only a subgroup of the population, that is children, that have experienced a total rise in the risk of death due to the airbags[5]. Eventually, The car-makers are developing "smart airbags" which can distinguish among a child, an adult and luggage bags and expand in an appropriate manner. For smart airbags, automobile companies are equipping certain cars with Occupant Classification Systems (OCS). OCS contains sensors that are able to differentiate between an adult and a child sitting in a passenger seat. There is no need of on/off switch to deploy airbags because of sophisticated computer technology of OCS can differentiate adult/child sitting in the seat. The purpose of this project is to develop an Occupant Classification System (OCS) that is able to detect each passenger sitting in the car and then classify everyone either as an adult or a child as part of ADAS system.

II. Literature Survey

Farzan, Wassim and Robert have implemented occupancy detection in a vehicle with convolutional networks on thermal images[6]. This paper explains about detection and counting of passengers in a vehicle only but not the classification of the occupants as an adult or a child and this process is costly as it involves thermal sensors.

K Jost has implemented an occupant detection system for advanced airbag deployment[7] which consists of a pressure sensor fitted to the seat, a bladder, and an ECU. But the classification of adult and child becomes faulty because sometimes children may hold some luggage, tighten their seat belt and sit. Then it senses the child as an adult and deploys airbag during a crash.

Omer, Park, J. Song, and Yoon has developed an adult and child classification system[8]. They are detecting both face and body. Instead of using any classification algorithm for adult-child classification, they used a mathematical approach. They calculated the heights of head and body of a person and then found the ratio of them. If the ratio is around 0.2, it is a child and if the ratio is around 0.15, then it is an adult. But we can't apply this in the car because people sitting in the front seats can be viewed fully but people sitting in the back seat can't be viewed fully. And also, the size of the face varies from the front seat to back seat for the same person with the camera fixed at one position.

The earlier occupancy detection is done either using seat pressure sensors or using weight sensors which are not reliable because even luggage in the seat can be treated as a person which leads to false deployment of airbag related to that position. Proper monitoring of passengers boarding a car can increase effectiveness, reliability and safety with proper control of airbags during a crash. Achieving a reliable and low-cost system using machine learning techniques is the main goal of this project. Occupants are captured using an infrared camera which works well during both day and night. This project contains two phases. The first stage is the detection of occupants by using machine learning techniques which are discussed in the upcoming sections. The second stage is the classification of each occupant either as an adult or a child by using techniques of machine learning that are also discussed in further sections.

III. METHODOLOGY

A night vision camera connected to the raspberry pi board for capturing the image. It is processed in the pi board and detects the number occupants sitting in the car. Once detection is completed, then it classifies each person whether he/she is adult or child. The software and hardware modules used for this project has given in section 3. Section 4 explains about collection of Dataset, Section 5 explains about occupant face detection, section 6 about occupant classification technique, section 7 about pipe lining detection and classification, section 8 Hardware implementation and results are discussed in section 9.

Using Haar-cascade and SVM:

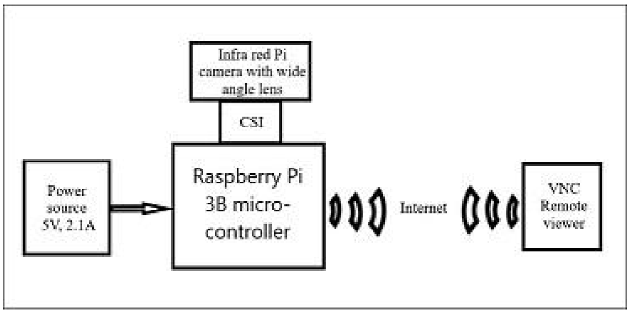

Figure 1: System block diagram

The flow of occupant detection and classification system is shown in the below figure. It starts with capturing the image inside a car from a fixed place. Then the image will be applied for HAAR-cascade detection. If any face has been detected then it will be sent for classification. Classification is done with SVM. If more than a face is found, then each face will be sent for classification individually. The output of classification is sent to VNC viewer through internet from micro-controller. Even if no face is found, data will be sent to VNC viewer. Data will be displayed on the user's side.

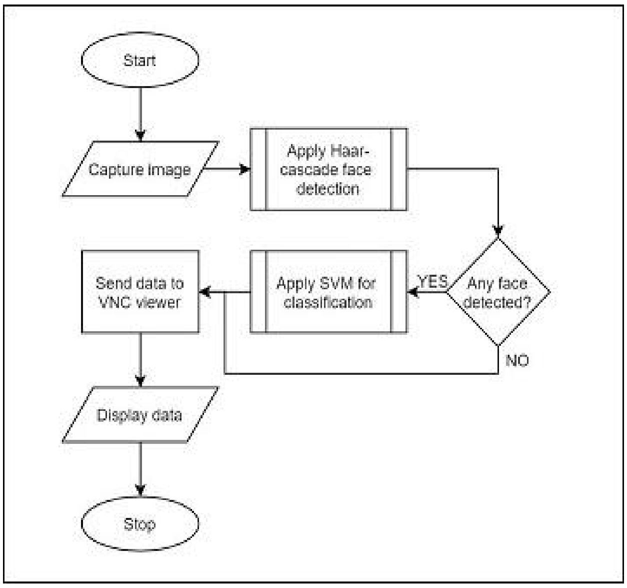

Figure 2: Flow chart

IV. MODULES USED

A. Software modules:

- Spyder IDE [11]
- Python 3.7 [9, 10]
- VNC Viewer

  VNC Viewer is an application that can turn your device into a remote desktop. It can be installed on a desktop or laptop or phone. It is available for all operating systems like Windows, Android and IOS. By installing the application, we can turn our phone as a remote desktop. We can access Mac, Linux and Windows computers instantly. We can view as well as control computer remotely

- Cascade Trainer GUI

  Cascade Trainer GUI is a tool used to train, test and improve cascade classifier models. It can be used to train machine learning classifiers using OpenCV tools [12].

B. Hardware modules:

- Raspberry pi 3

Raspberry Pi is developed by the Raspberry Pi Foundation. It consists of a series of computers. It is developed to teach basic computer science for young students[13]. Raspberry Pi is also called as RPi. RPi 3B is a computer having a single board with Bluetooth and WLAN connectivity. It needs 5V and 2.1A Micro USB power supply. It can be accessed remotely using VNC viewer software from desktop or mobile. Keyboard and mouse can be connected through USB slots available on the board. It can connect to display through DSI port and camera through CSI port. It has 1GB RAM and 40 GPIO pins with 64 bit CPU.

Figure 3: Raspberry pi 3B board

- Night vision pi camera

  It is an infra-red camera having two infra-red LED that supports adjustable focusing. It can be connected to the Raspberry pi board.

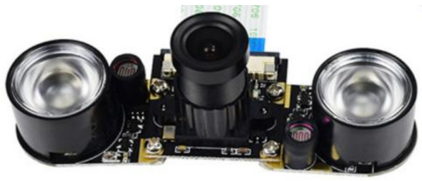

Figure 4: Raspberry pi camera with IR LED

V. Collection of Dataset

The process of collecting information in an equipped arrangement on targeted variables, which then empowers one to evaluate outcomes and answer relevant questions is called data collection. Data collection is very costly, more time-consuming in a machine learning project. To maintain the integrity of research, data collection accuracy is essential. Data has collected using an infrared camera connected to Raspberry Pi. Raspberry Pi is powered using a power bank and placed in the car on the rear viewing mirror which is near the windshield inside the car. Raspberry Pi is connected to the internet using Wi-Fi from a mobile phone through a hot spot. Raspberry Pi is viewed and controlled from mobile using a VNC viewer. The python file is written in such a way that, running it activates the camera and captures the image and stores it in memory. Hence whenever python file is run from VNC viewer, image is captured from the camera and saved. Total 300 images collected.

Figure 5: Sample images

VI. Occupant Face detection

Face detection is a technique used for detecting or finding a person from digital images. Images are taken by a pi camera connected to the Raspberry pi 3 board [14]. It works by searching some facial features. It is used in many domains such as mobile platforms, robotics, security systems, surveillance systems, automobiles, marketing, tracking systems etc. Basically face detection is used for face recognition. It has two steps, one is feature extraction and selection, next is the classification of objects. Some face detection algorithms extract features like relative position, size/shape of eyes, nose, jaw, and cheekbones. Among are several machine learning algorithms Haar-Cascades is the easiest feature extraction with high accuracy and less computation time comparatively.

Paul Viola and Michael Jones proposed an object Detection using HAAR wavelet-based cascade classifiers in 2001 [15]. It is a machine learning-based approach which is an effective method for object detection. From a set of negative and positive images, a cascade function is trained and used for detecting objects in other images. We are using HAAR feature extraction for face detection. A classifier is trained with a lot of negative images (images without faces) and positive images (images of faces) using this algorithm. Then features are extracted from it. Each feature is a single value. Sum of pixels under white and black rectangle is calculated. Each feature is obtained by subtracting the white sum from the black sum. Different features used are shown in the below image.

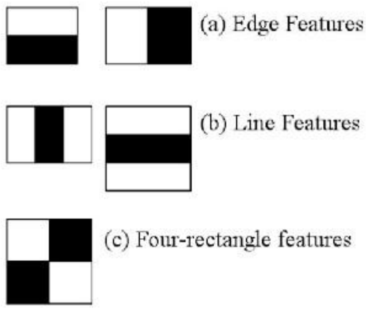

Figure 6: Different Haar-features

For each feature, a classifier is formed, which is called weak classifier because single feature alone not enough to classify the image. The weighted combination of all weak classifiers is the final classifier. It is applied as one by one called a cascade of classifiers instead of applying all classifiers at a time on an image. If one classifier shows negative, then no need to check the remaining classifiers. Hence the time reduces. The part of the image which passes all classifiers is a face region. Haar-Cascade method requires several negative and positive images to train the classifier. Positive images are those images that comprise the images which we need our classifier to recognize. Negative images are those images which do not include the thing we need to recognize. Negative images must NEVER hold any portion of positive images. Classifier makes false-negative detection if it holds some portion of positive images mistakenly. In practical, negative images should be relevant to the positive images somehow.

Figure 7: Negative images from internet

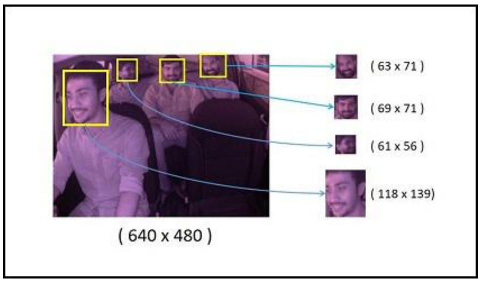

Figure 8: Creating positive images

Figure 9: Creating negative images

'p' is the folder for positive images and 'n' is the folder for negative images. For more accuracy, faces from the collected occupant images need to be cropped and saved. For cropping the images, we are using the cropper utility tool of Cascade Trainer GUI. Each image might be of different size. So we are training a cascade classifier with images of different sizes which should be able to detect faces of all sizes in occupant images.

A total of 395 faces have been cropped from 150 images out of 300 collected occupant images. Remaining 150 images are used for testing. Total 1000 negative images collected. Different parameters have been set to required values like Number of stages are 20, Pre-calculated values buffer size as 2048 MB, Pre-calculated indices buffer size as 2048 MB, Number of threads are 5 and Acceptance ratio break value as -1.00 in Common tab of GUI. Sample width and height as 24, Feature type is HAAR and HAAR feature type as Basic from Cascade tab. Remaining parameters are kept constant in Boost tab like Boost type GAB, Minimum hit rate 0.995, Maximum false alarm rate 0.5, Weight trim rate 0.95, Maximum depth weak tree 1.0 and Maximal weak trees 100 [16]. After setting the parameters, click on the start button which starts training the classifier in GUI. 'cascade.xml' is the cascade classifier created after the successful finish of the training in that folder.

Now the classifier can detect faces. It has to detect even if multiple faces exist in an image. To tests the classifier, I have written a Python script in Spyder IDE which works on the trained classifier and makes rectangles on the image at the locations of the faces. The classifier is tested with 150 test images. As a whole, read the test images from a folder, apply face detection and write them to another folder. The writing folder contains all test images with bounding boxes on the faces. It may contain true positives (TP), false positives (FP) and false negatives (FN) among the detected faces.

- True Positive: Detecting a face as a face.
- False Positive: Detecting as a face which is not at all a face.
- False Negative: Not detecting a face as a face
- True Negative (TN) is not applicable in this scenario.

Figure 10: Tested images

VII. Occupant Classification

Support Vector Machine is shortened as SVM. It is extremely used by several as it provides vital accuracy with more limited computation power. The goal of the SVM algorithm is to obtain a hyper-plane that classifies the data points in multidimensional space.

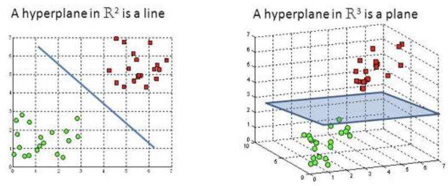

Figure 11:Hyper-planes in 2D and 3D feature space

The data points which are close to the hyper-plane are called Support vectors. They determine the orientation and position of the hyper-plane. The margin of the classifier is maximized using those support vectors. The hyper-plane position will change upon deleting the support vectors. SVM classifier is built with those support vectors.

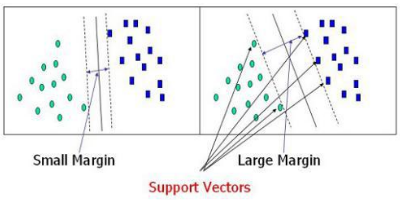

Figure 12: Support vectors of hyper-plane

Now, need to prepare data set for face classification of occupants in an image. Already, 'p' folder contains 395 face images which are extracted from occupant images as explained in section 5 and figure 8. We can reuse the same images to train our SVM classifier. In these 395 images, 300 images are of adult faces and 95 are of children faces. So we need to increase the children face count to train SVM classifier properly without over-fitting or under-fitting. We applied image augmentation techniques to increase the children face count like image rotation. We rotated children face images by 30 degrees and generated new 95 images. Now a total of 490 images with 190 children faces and 300 adult faces. We are converting an image into a vector and giving it directly to the classifier. It means each pixel is a feature. Each image is of different size but the input feature set of SVM should be constant.

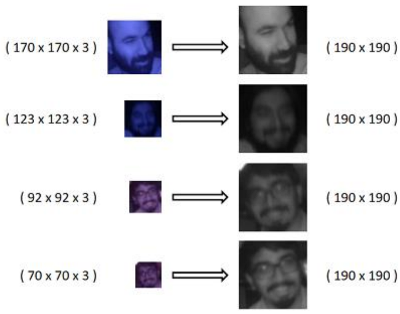

Figure 13: Re-sizing face images

Each re-sized image contains 36100 pixels. As we are converting image and giving it to SVM directly, feature vector shape would be (36100,). We need to label each feature vector. Let's take two classes like 0 and 1, which represents a child and an adult respectively. Just append the class label at the end of each feature vector to specify it as a child or an adult. After adding class labels, convert each image into a vector which will be of shape (36101,). Save all the feature vectors in a CSV file. The shape of the data set prepared is (490, 36101).

For training SVM classifier, first, we need to load the data set which is in the form of CSV file. It can be loaded with the NumPy module. Now load the features into one array 'X' and classes of each feature in another array 'Y'. The shape of 'X' is (490, 36100) and the shape of 'Y' is (490,). Now load the Support Vector Classifier (SVC). We are using 'train test split' to split the training and testing data with test size 0.33 and random state 42 [17]. It generates four variables called X train, X val, Y train, Y val where X train and Y train are used for training the classifier and X val and Y val are used for validating the classifier after successful completion of training. Now train the using method 'fit(X train, Y train)' and then use the method 'predict(X val)' predict the classes of validation data

VIII. Pipe Lining Detection And Classification

Occupant detection and classification are combined such that after capturing an image, it is displayed with bounding boxes of different colors after whole processing. Processing takes place sequentially with detection and classification. After detecting multiple faces, each face is classified individually and then drawn on the original image. If a detected face is of an adult, then box around it would be in Green colour otherwise Blue. The entire process is shown in below block diagram.

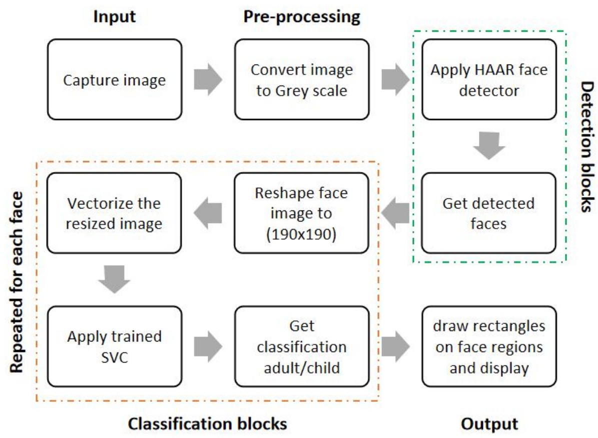

Figure 14: Combination of detection and classification

IX. Hardware Implementation

The above-said methods for occupant detection and classification have been implemented in a vehicle (hatchback car) for both collecting data set as well as for testing. Images of implementation of hardware are shown below for capturing the image and RPi for processing with a power source.

Figure 15:Raspberry Pi mounting on rear view mirror

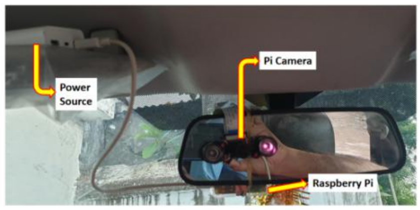

Figure 16: Hardware setup

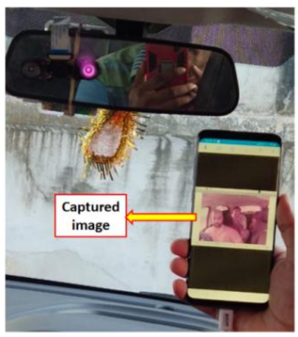

Figure 17: RPi screen remote view

X. Results

A. Face Detector

In a classification problem, a Confusion matrix is used to show the performance of the classifier on test data. Following is the confusion matrix for this problem statement [18].

	PREDICTED AS A FACE	PREDICTED AS NOT A FACE
FACE	555 (TP)	57 (FN)
NOT A FACE	90 (FP)	0 (TN)

Figure 18: Confusion matrix

Accuracy is the ratio of the whole true predictions to the whole input samples. Accuracy of model is 79.05%. Precision tells how accurate the trained classifier is. It tells how many predictions of the model is correct when compared with the total prediction of the model. Precision of this model is 86%. Recall tells how many have been classified as positives over total actual positives. Recall of this model is 90%. F1 Score is the Harmonic mean of Precision and Recall. It is used to seek balance balance between Precision and Recall. F1 Score of this face detector is 87.95.

B. Face Classifier

After predicting 'predict(X val)', calculate accuracy score by comparing the predictions of X val with Y val. After validation, it gave an accuracy score of 99.36%. Below images show the classifier output with adults and children. The green box indicates adult and Blue box indicates child.

Figure 19: Classifier output

XI. Conclusion

In-Vehicle occupants image is captured using an infrared Pi camera connected to Raspberry Pi from a Hatchback car. The camera is controlled by VNC Viewer, an Android application in the mobile. After capturing the image, it is processed in RPi micro-controller, detected faces and classified each as either adult or child and displayed in the mobile again through VNC Viewer. This project is limited to hatchback cars and occupants faces are visible properly. It is not working properly when faces are exposed to bright light.

Future scope includes improvement of accuracy of the face detector either by retraining with different images and parameters or changing the algorithm. Different classifiers can be compared to know the best one for adult-child classification. It can implemented with Deep learning models easily. This project can be extended to recognition of postures of occupants and checking change in vehicle occupancy when a car stops every time. It can also be extended to Sedan cars, SUVs and other automobiles.

References

[1] Shelden, C. Hunter. "Prevention, the only cure for head injuries resulting from automobile accidents." Journal of the American Medical Association 159, no. 10 (1955): 981-986.

[2] Atul Kumar. "Active and passive automotive safety systems - ElectronicSpecifier.com" https://www.electronicspecifier.com/products/sensors/active-and-passive-automotive-safety-systems¿

[3] Hetrick, John W. "Safety cushion assembly for automotive vehicles." U.S. Patent 2,649,311, issued August 18, 1953.

[4] Marshall Brain. "How Airbags Work - HowStuffWorks.com" ¡https://auto.howstuffworks.com/car-driving-safety/safety-regulatory-devices/airbag.htm¿ 28 May 2020

[5] Graham, John D., Sue J. Goldie, Maria Segui-Gomez, Kimberly M. Thompson, Toben Nelson, Roberta Glass, Ashley Simpson, and Leo G. Woerner. "Reducing risks to children in vehicles with passenger airbags." Pediatrics 102, no. 1 (1998): e3-e3.

[6] Erlik Nowruzi, Farzan, Wassim A. El Ahmar, Robert Laganiere, and Amir H. Ghods. "InVehicle Occupancy Detection with Convolutional Networks on Thermal Images." IEEE Conference on Computer Vision and Pattern Recognition, pp. 0-0. 2019.

[7] Jost, K. "DELPHI OCCUPANT DETECTION FOR ADVANCED AIRBAGS. THE PASSIVE OCCUPANT DETECTION SYSTEM (PODS) FROM DELPHI AUTOMOTIVE SYSTEMS." Automotive Engineering International 108, no. 12 (2000).

[8] Ince, Omer F., J. S. Park, J. Song, and B. W. Yoon. "Child and adult classification using ratio of head and body heights in images." International Journal of Computer and Communication Engineering 3, no. 2 (2014): 120.

[9] Venners, Bill. "The making of Python." Artima Developer 1 (2003).

[10] DeBill, Erik. "Module counts." WWW],¡ http://www.modulecounts. com/.[Haettu 1.11.2016.]¿ (2010).

[11] Rick Grehan. "Spyder review – Techworld.com" ¡https://www.techworld.com/review/programmingsoftware/spyder-review-3238833/¿ Sep 08, 2010

[12] Amin Ahmadi. "CASCADE TRAINER GUI - Amin-Ahmadi.com" ¡https://amin-ahmadi.com/cascade-trainer-gui/¿ October 2016

[13] Cellan-Jones, Rory. "A£ 15 computer to inspire young programmers." BBC News 5 (2011).

[14] Pasumarti, Priya, and P. Purna Sekhar. "Classroom Attendance Using Face Detection and Raspberry-Pi." International Research Journal of Engineering and Technology (IRJET) 5, no.03 (2018): 167-171.

[15] Viola, Paul, and Michael Jones. "Rapid object detection using a boosted cascade of simple features." IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, vol. 1, pp. I-I. IEEE, 2001.

[16] Tejas R. Phase , Dr. Suhas S. Patil, 2019, Building Custom HAAR-Cascade Classifier for face Detection, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH TECHNOLOGY (IJERT) Volume 08, Issue 12 (December 2019).

[17] Pedregosa, Fabian, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.

[18] Koo Ping Shung. "Accuracy, Precision, Recall or F1? - TowardsDataScience.com" ¡https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9¿