Ai approach for road Safety

University of batna 2

Allaoui Mohamed Khalil

Aouina Khansa

mars 8

Abstract

Abstract goes here

Dedication

To mum and dad

Acknowledgements

I want to thank...

Contents

1 Introduction **8**

1.1 Background and Problems . . . 8
1.2 Motivation . . . 9
1.3 Related Work . . . 11
1.4 Thesis Organization . . . 11

2 Theory and Concepts **12**

2.1 Digital Image Processing Basics . . . 12
2.1.1 Sampling . . . 12
2.1.2 Quantization . . . 13
2.2 Color Models . . . 14
2.2.1 RGB to Grayscale . . . 14
2.3 Image Values and Statistics . . . 15
2.4 Classification . . . 15
2.5 Integral Image . . . 17
2.6 Haar Feature-based Classifiers . . . 18
2.6.1 AdaBoost algorithm . . . 20

3 Driver Monitoring **21**

3.1 Introduction . . . 21
3.2 Driver Monitoring Technologies . . . 21
3.2.1 Physiological Signals Sensors . . . 22
3.2.2 Video-based sensors . . . 22
3.2.3 Evaluation . . . 22
3.3 Driver Drowsiness . . . 23
3.4 Driver Inattention . . . 23
3.5 Proposed work . . . 25
3.5.1 method 1 : Haar Cascades . . . 25

List of Figures

1.1 The Evolution of Automated Safety Technologies 9
1.2 Typical types of sensors for ADAS 10

2.1 Sampling and Quantization 13
2.2 Gray scale image of Handwritten digit 13
2.3 Illustration of FP, FN, TP, and TN 16
2.4 Calculation of integral values and integral image 17
2.5 Common types of Haar-like features 18
2.6 Example of Haar-feature matches in eyes 18
2.7 A cascade of weak-classifiers 19

3.1 Illustration of difrent Driver monitoring approches 22
3.2 Distracted Driver Sceenes 24
3.3 Illustration of Bounding Box in object detection using haar-like detector 26
3.4 Initial flowchart for driver monitoring with haarcascades 27
3.5 face/eyes detection with Haar cascades 28
3.6 Image Pyramid and Scale Factor 30
3.7 detections using various MNN values. Dir eyes whdha w face whdha ! 31
3.8 Face/Eyes sizes for diffrent distances of the driver and the camera 31
3.9 face/eyes detection with Haar cascades under diferent lighting conditions 32

List of Tables

3.1 shows the evaluation results of the two above described Driver Monitoring Technologies. The (+) symbol represents the rating level 23
3.2 Table shows basic statistics for gray scale images of Figure 3.9 under difrent lighting conditions 32
3.3 Table shows optimum values for detector parameters under difrent lighting conditions 33

List of Algorithms

2.1 The AdaBoost algorithm for classifier learning. 20
3.2 first algorithm for face/eyes detection using Haar cascades 28

Nomenclature

ACC Accuracy

ADAS Advanced Driving assistance Systems

BSM Blind-Spot Monitoring

DAA Driver Attention Alert

DAS Driving assistance Systems

FN False negative

FP False positive

LAS lane keep assist system

LC Lighting conditions

LIDAR Light Detection and Ranging

MNN Minimum Neighbors

NHTSA National Highway Traffic safety administration

PR Precision rate

RADAR Radio Detection and Ranging

ROI Reagion of Interest

RR Recall rate

SF Scale Factor

TN True negative

TP True positive

Chapter 1

Introduction

Recently, the development of advanced driver assistance systems (ADAS) has facilitated people's daily life from comfort to safety. However, these systems are complex [11], utilizing vehicle parameters, environmental observations, and traffic patterns to assist the driver. These systems are added cost-to-ownership due to the added expense of sensors and computing hardware needed to perceive the environment, especially in real-time monitoring. Thus, further development in this area is needed to improve reliability, performance, and decrease costs.
This work describes a driver assistance system based on computer-vision techniques.

1.1 Background and Problems

According to the World Health Organization (WHO) [9] around 1.3 million people die each year as a result of road traffic accidents, in addition to 50 million serious injuries. This cost most countries 3% of their gross domestic product.In 2016

The report also highlights More than 90% of road traffic deaths occur in low- and middle-income countries. Road traffic injury death rates are highest in the African region. Even within high-income countries, people from lower socioeconomic backgrounds are more likely to be involved in road traffic crashes.

Although current "*passive*"[1] and "*active*"[2] safety systems [31] can reduce the impact of traffic accidents, only a few car accidents are caused by bad weather and unsafe road infrastructure while most by human fault [2], such as: [9]
-Speeding

[1]Passive systems such as air-bags, seat belts, padded dashboards, or physical structure of a vehicle, normally help to reduce the severity or the consequences of an accident.

[2]Active systems like adaptive cruise control (ACC), automatic braking systems (ABS), or lane departure warning systems (LDWS) are designed to prevent or decrease the chance of crash occurrence.

-Driving under the influence of alcohol and other psychoactive substances
-Nonuse of motorcycle helmets, seat-belts, and child restraints
-Inadequate law enforcement of traffic laws

According to [10] the most likely causes of car accidents are: the driver may lose concentration on the road when driving, drivers falling asleep at the wheel, driver fatigue, or driver distraction, no matter the driver is experienced or not. A study in the United States by the National Highway Traffic Safety Administration (NHTSA) [5], confirms that almost 80% of all types of vehicle accidents involve driver fatigue, driver drowsiness, or driver distraction (in general, distracted driving), with the high speed, may cause the driver to have no time to realize the road status, which leads to car accidents.

These shocking statistics highlight the importance of research and development of advanced driver assistance systems (ADAS) focusing on *"Driver Monitoring"* by driver behavior analysis as well as *"Road Monitoring"* by road hazards detection.

1.2 Motivation

Various driving assistant systems have been developed in automotive engineering, the U.S. National Highway Traffic Safety Administration (NHTSA) defined six levels of automation from level 0 to level 5, which describes the relationship from no autonomous driving to fully autonomous driving in automotive engineering, see 1.1 .

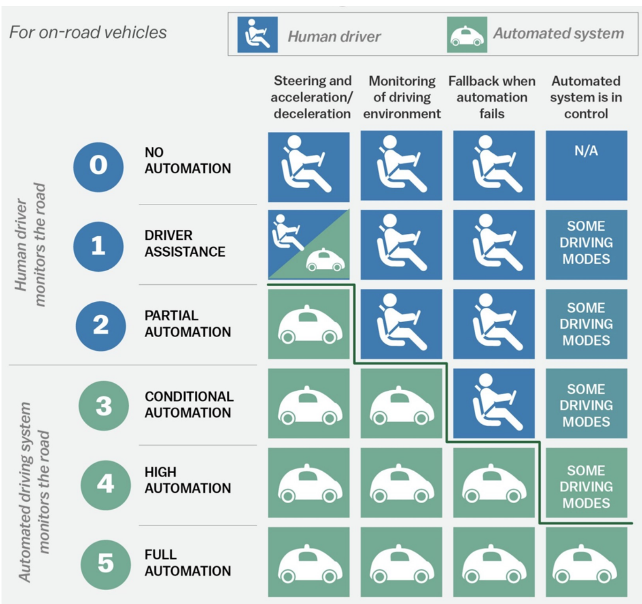


Figure 1.1: The Evolution of Automated Safety Technologies

According to the figure 1.1, in the first level, the driver needs to drive the vehicle and focus on the road to react as soon as possible. In levels 1 and 2, driving automation applies to vehicles with (ADAS) that can take over steering, acceleration, and braking in specific scenarios. But, even though level 1 driver support can control these primary driving tasks. In level 3, the system detects the environment to decide whether the driver needs to drive the vehicle, which is called conditional automation. Level 4 and level 5 indicate high automation and full automation respectively, which means the system will fully control the vehicle.

Among these levels, an (ADAS) is considered to be the basic and important component. Generally, An *ADAS* is an electronic system in a vehicle that uses advanced technologies to assist the driver [22]. It can include many active safety systems, such as [25] lane keep assist system (LAS), blind-spot monitoring (BSM), driver attention alert (DAA), and many other systems that work together to increase the safety of drivers, passengers, pedestrians, and other road users.The objective is to recognize critical driving situations by perception of the vehicle and the divers as *internal parameters*, road as *external parameters*, and the weather and lighting condition as *additional parameters*.

To collect these parameters. ADAS and autonomous driving functions feed off a continuous stream of information about the environment surrounding the vehicle, and it's the sensors' job to provide that [27] see the figure 1.2.

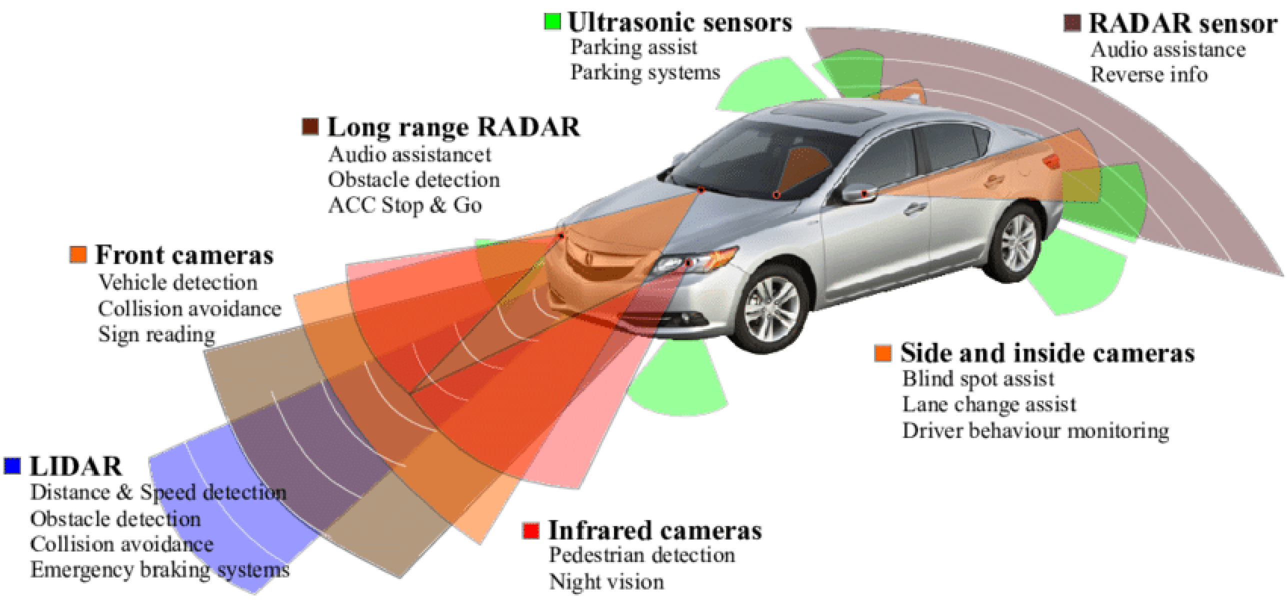


Figure 1.2: Typical types of sensors for ADAS

The three main sensors used by the automotive industry to maintain the perception for autonomous vehicles at various levels of autonomy are [27]:
Ultrasonic sensors [12], (RADAR, LIDAR) [20], Cameras [16].

Ultrasonic sensors operates by transmitting short bursts of sound waves and measuring the time taken for the sound to travel to a target object, be reflected, and return to the receiver, they are usually used for short-distance applications at low speeds, such as park assist, self-parking, and blind-spot detection.

RADAR (Radio Detection and Ranging) sensors emit radio waves and analyse the bounced wave via a receiver. Because RADAR signals can range 300 meters in front of the vehicle, they are particularly important during highway speed driving. Additionally, RADAR can see through bad weather and other visibility occlusions. Because their wavelengths are just a few millimeters long, they can detect objects of several cm or larger . LiDAR (Light Detection and Ranging) systems are used to detect objects and map their distances in real-time. Essentially, LiDAR is a type of RADAR that uses one or more lasers as the energy source. LIDARs can provide a higher resolution result but in a narrower angular field.

Camera sensors are similar to regular consumer cameras, like those that equip most smartphones. They are cheaper than both RADAR and LIDAR sensors. They can be adapted to any vehicle and any user can use them with no difficulty. For many years the fields of computer vision and image processing have used them to solve their problems. On the other hand, camera's performance drops dramatically under bad lighting conditions and they generally need a more complicated post-processing (image processing, image classification, and object detection) in order to convert the raw perceived images into a meaningful information.

Each of the above mentioned sensors have advantages and disadvantages, so that the ideal system would be a combination of all three.

1.3 Related Work

There is a wide range of research topics under the umbrella of road safety and driver assistance systems (DAS) such as *traffic signs recognition* [x], *lane detection* [x] *pedestrian detection* [x], *vehicle detection* [x], and *driver behaviour monitoring* [x] including driver fatigue, drowsiness and distraction detection. However, at a higher level, the research could be classified into two main categories: the research related to "Road monitoring" and the research works that focus on the "Driver monitoring".

1.4 Thesis Organization

bla bla bla

Chapter 2

Theory and Concepts

In this chapter, we will simply introduce some basic concepts, methods, and mathematical background that we use in this thesis. We also provide symbols, image notations, and the equations that will be consistently used in the following chapters.

2.1 Digital Image Processing Basics

Digital Image Processing means processing digital image by means of a digital computer. We can also say that it is a use of computer algorithms, in order to get enhanced image either to extract some useful information [26].

In Digital Image Processing, signals captured from the physical world need to be translated into digital form by Digitization[1] Process. In order to become suitable for digital processing.

An image is defined as a two-dimensional function,I(x,y), where x and y are spatial coordinates, and the amplitude of I at any pair of coordinates (x,y) is called the intensity of that image at that point.An image must be digitized both spatially and in amplitude [26]. This digitization process involves two main processes *Sampling*, and *Quantization* [30].

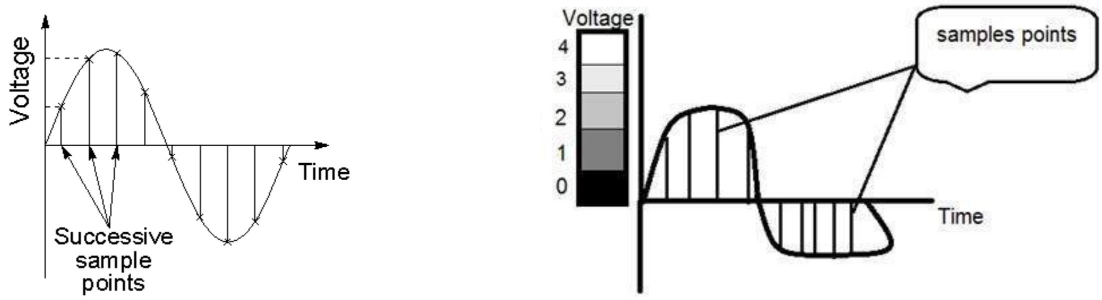
2.1.1 Sampling

In digital image processing, Sampling is the reduction of a continuous-time signal to a discrete-time signal.Since an analogue image is continuous not just in its co-ordinates (x axis), but also in its amplitude (y axis), so the part that deals with the digitizing of co-ordinates is known as sampling [28], see Figure 2.1a .

[1]Digitization: is the process of converting information into a digital (i.e. computer-readable) format.

2.1.2 Quantization

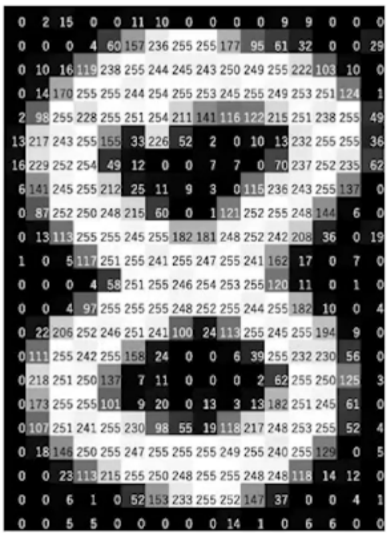
Quantization is the process of mapping input values from a large set to output values in a smaller set, often with a finite number of elements. Quantization is the opposite of sampling, It is done on the y-axis [28], see Figure 2.1b



(a) Sampling (b) Quantization

Figure 2.1: Sampling and Quantization

A digital image is typically composed of picture elements (pixels) located at the intersection of each row "**I**" and column "**J**" in each "**N**" color channels [6]. Digital images are stored in the form of a matrix of numbers where these numbers represent the intensity of each pixel, the range of these numbers (pixel values) is relative to the **Bit depth**[2], in general images are stored in 8 byte that means $2^8 = 256$ possible values.



(a) Matrix of pixels

0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

(b) pixels values

Figure 2.2: Gray scale image of Handwritten digit

[2]The bit depth "**k**" is the number of bits per pixel, the grey scale of an image is equal to 2^k

2.2 Color Models

Generally, images that are captured by camera sensors are color images, by default they use RGB color model [29], and because in the following chapters we mainly work on grayscale images like in Figure 2.2 we need to do a color space conversion. In this section, we will introduce briefly some basic concepts related to color models used in digital image processing and color space conversion

2.2.1 RGB to Grayscale

In the RGB color model, each color in the image is obtained by superimposing three colors, i.e., red, green, blue. In this model, each pixel **P** in the image can be represented by $R(x_p, y_p)$, $G(x_p, y_p)$, $B(x_p, y_p)$. When $R(x_p, y_p) = G(x_p, y_p) = B(x_p, y_p)$ the given image becomes a grayscale image although it still has 3 color channels.

The goal is to convert the 3d [3] RGB image to a 2d[4] Grayscale image because smaller data enables developers to do more complex operations in a shorter time, there are a number of commonly used methods to convert an RGB image to a grayscale image such as average method and weighted method [24].

Average method

The Average method takes the average value of R, G, and B as the grayscale value as follows :

$$Grayscale = \frac{R + G + B}{3}$$

The average method is simple but doesn't work as well as expected. The reason is that human eyeballs react differently to RGB. Eyes are most sensitive to green light, less sensitive to red light, and the least sensitive to blue light. Therefore, the three colors should have different weights in the distribution. That brings us to the weighted method.

The Weighted Method

The weighted method, also called the luminosity method, weighs red, green, and blue according to their wavelengths. The improved formula is as follows:

$$Grayscale = 0.299R + 0.587G + 0.114B$$

[3] 3d means 3 dimensional image (width, height, depth:'number of color channels')

[4] 2d means 2 dimensional (width, height)

2.3 Image Values and Statistics

Considering **I** *a grayscale image, and X, Y are numbers of rows, columns respectively.*

Mean :

$$mean = \frac{1}{XY}\sum_{i=1}^{X}\sum_{j}^{Y} I(i,j)$$

Variance:

$$variance = \frac{1}{XY}\sum_{i=1}^{X}\sum_{j}^{Y} |I(i,j) - mean|^2$$

Energy:

$$energy = \frac{1}{XY}\sum_{i=1}^{X}\sum_{j}^{Y} |I(i,j)|^2$$

2.4 Classification

Classification is a process that uses a set of features or parameters to recognize an object. In this thesis we use supervised classification techniques, which means that an expert defines the classes of objects (e.g., face, eye, vehicles), and also provides a set of sample objects for a given class which is called training set. Regardless of the chosen classification technique (e.g., neural networks, decision trees, or nearest neighbour rule), we have two phases to construct a classifier: a training phase and an application phase.

Based on the provided training dataset, a classifier learns to use which set of features, parameters, and weights to be combined together in order to distinguish objects from non-objects. In the application phase, the classifier applies the already trained features and weights to an unknown query image to detect similar objects, based on what is has previously learned from the training set.

Regardless of the classification technique, the performance of a classifier can be evaluated based on the detection efficiency. Let **TP** denotes the number of true-positives or hit, when a classifiers can correctly detect the objects. Also let **FP** be the number of false-positives or miss, when a classifier wrongly detects a non-object as an object. Similarly, we can define true-negatives **(TN)** and false-negatives **(FN)** to describe correct classification of non-objects and the missing objects, respectively. Although we always can measure (count) the number of **FN**, we can not simply define the number of **TN** for an application such as vehicle detection in a road scene.

That is because the background of an image in basically uncountable. Therefore, for performance evaluation of a classifier we mainly rely on evaluations using TP, FP, and FN.

We define the precision-rate (PR), recall-rate (RR), and accuracy (ACC) as follows [23] :

$$PR = \frac{TP}{TP + FP}$$

$$RR = \frac{TP}{TP + FN}$$

$$ACC = \frac{TP + TN}{TP + FN + TN + FP}$$

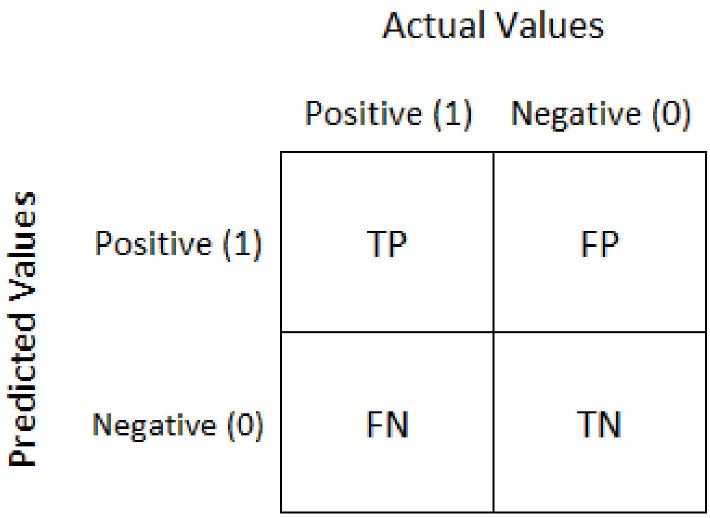


Figure 2.3: Illustration of FP, FN, TP, and TN

Where PR is the ratio of true-positives to all detections by the classifier, RR is the ratio of true-positives to the total number of actual existing objects (i.e. the ground truth objects) in the analysed frames, and ACC is defined as the number of classifications a model correctly predicts divided by the total number of predictions made. It's a way of assessing the performance of a model.

2.5 Integral Image

For a given image I, the integral image I_{int} , which was first used by Viola and Jones in computer vision [21], is the summation of all pixel values in the image, or in a window (sub-image).

Rectangle features can be computed very rapidly using an intermediate representation for the image which we call the integral image. The integral image at location (x, y) contains the sum of the pixels above and to the left of (x, y) , inclusive:

$$I_{int}(x, y) = \sum_{x'<=x, y'<=y} i(x', y')$$

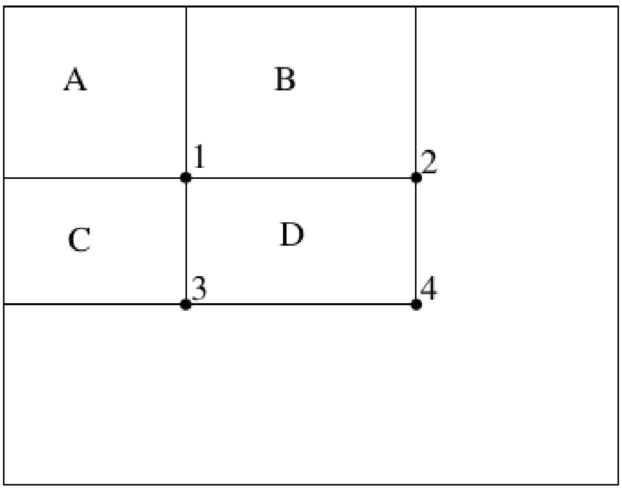


Figure 2.4: Calculation of integral values and integral image

Taking the figure 2.4 as an example, The sum of the pixels within rectangle **D** can be computed with four array references. The value of the integral image at location 1 is the sum of the pixels in rectangle **A**. The value at location 2 is **A + B**, at location 3 is **A + C**,and at location 4 is **A + B + C + D**. The sum within **D** can be computed as $4 + 1 - (2 + 3)$.

Having the integral values of each pixel calculated and saved in a data structure array, we can calculate the integral image of any image or sub-image just by applying one addition and two subtraction operations. This is very fast and cost-efficient for real-time feature-based classification algorithms, with a computational complexity of $O(N_{cols}N_{rows})$.

2.6 Haar Feature-based Classifiers

Inspired by Haar-wavelets, Viola and Jones [21] introduced the idea of Haar-like features with square-shaped adjacent black and white patterns for the first time see Figure 2.5, in the area of face detection.

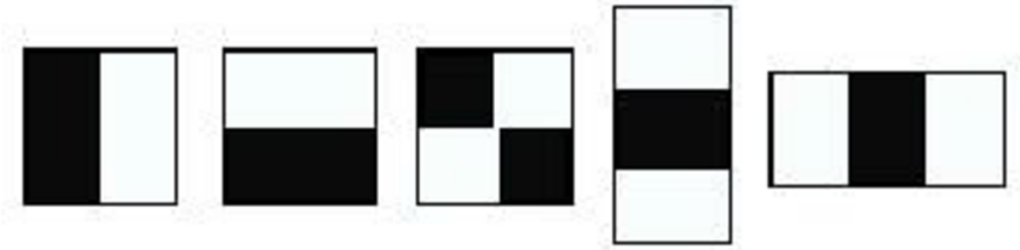


Figure 2.5: Common types of Haar-like features

The common types of Haar-like features are *Line Features*,*Edge features*, and *Diagonal features*. They are just like the convolutional kernel. Each feature is a single value obtained by subtracting the sum of pixels under the white rectangle from the sum of pixels under the black rectangle, to reduce these computations they use the integral image. These features are very important in the context of face detection because they can detect a face based on the face's properties see Figure 2.6.

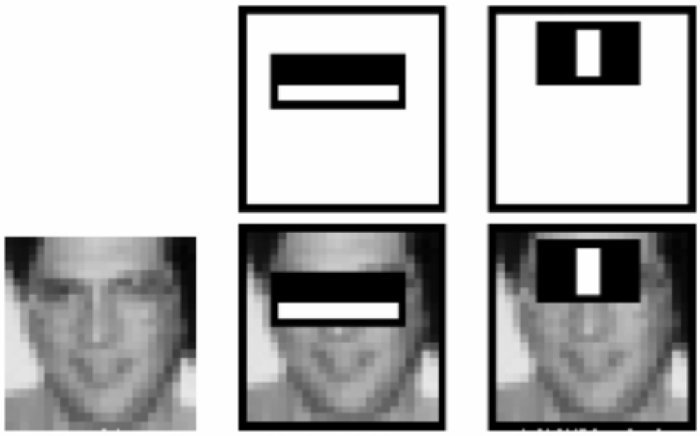


Figure 2.6: Example of Haar-feature matches in eyes

Consider the Figure 2.6. The top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose.

A *strong classifier* comprises of a series of weak classifiers (normally more than 10), and a weak classifier itself includes a set of a few Haar features (normally 2 to 5).

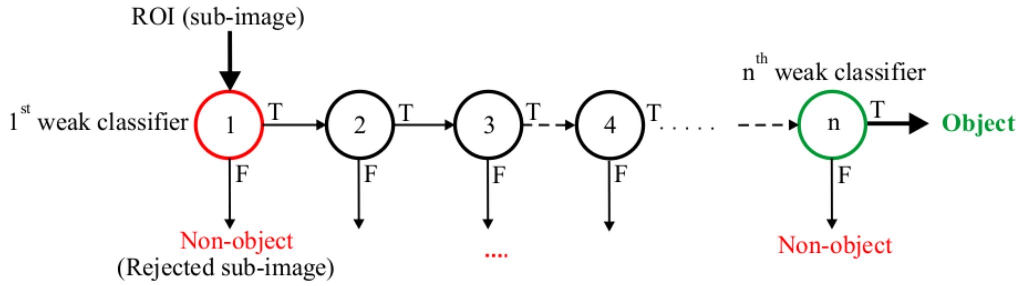


Figure 2.7: A cascade of weak-classifiers

Figure 2.7 visualizes a cascade of weak classifiers, that all together make a strong classifier. The classifier starts with the first weak classifier by evaluating the region of interest (ROI). It proceeds to the second stage (second weak classifier) if all the Haar-features in the first weak classifier match with the ROI, and so on, until reaching the final stage; otherwise the search-region under the sliding window will be rejected as non-object. Then the sliding window moves to the next neighbour ROI. If all the Haar-features within the all weak classifiers, successfully match with the ROI, then the region will be marked with a bonding box as a detected object.

The process of selecting appropriate Haar-features for each weak classifier, and then for the whole strong classifier constitutes the training phase which could be done using a machine learning technique like AdaBoost algorithm [18].

2.6.1 AdaBoost algorithm

is an approach to machine learning based on the idea of creating a highly accurate prediction rule by combining many relatively weak and inaccurate rules. The AdaBoost algorithm of Freund and Schapire was the first practical boosting algorithm, and remains one of the most widely used and studied, with applications in numerous fields [18].

Algorithm 2.1 The AdaBoost algorithm for classifier learning.

- Given example images $(x_1, y_1), ..., (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize the weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1$ to T:

  1. Normalize the weights,
  $$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}$$
  so that w_t is a probability distribution.
  2. For each feature, j, train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to
  $$w_t, \epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$$
  3. Choose the classifier, h_t,with the lowest error ϵ_t.
  4. Update the weights:
  $$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$
  where $e_i = 0$ if example x_i is classified coreectly, $e_i = 1$ otherwise,and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$

- The final strong classifier is:

$$h(x) = \{^{1}_{o} \quad ^{\sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t}_{otherwise} \}$$

where $\alpha_t = \log \frac{1}{\beta_t}$

Chapter 3

Driver Monitoring

To Update ! *In this chapter, we propose methods to assess the driver's state of drowsiness and inattention based on face and eyes-status analysis. The chapter begins with a brief discussion of signs of drowsiness and available methods for detecting them, and it continues with the first proposed method which is based on traditional computer vision techniques followed by the seconde method which is based on deep learning, then we discuss the strengths, weakness, and limitations for each method. The chapter continues with our optimization techniques to improve the performance of these methods in terms of speed, detection rate, and detection accuracy under non-ideal lighting conditions and for noisy images.*

3.1 Introduction

A driver-monitoring system is an advanced safety feature that track driver drowsiness or distraction, and to issue a warning or alert to get the driver's attention back to the task of driving.

Driver-monitoring systems typically use sensors to collect data about the driver and pass these data to a software. The software can then determine whether the driver is blinking more than usual, whether the eyes are narrowing or closing, and whether the head is tilting at an odd angle. It can also determine whether the driver is looking at the road ahead, and whether the driver is actually paying attention or just absent-mindedly staring. [x]

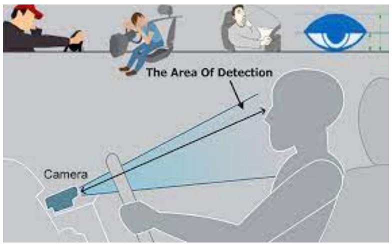
!!!!!!!!!!!!!!!!!!!!!!!!!!!! Into from thesis x about driver monitoring !!!!!!!!!!!!!!!!!!!!!!!!!!!!

3.2 Driver Monitoring Technologies

According to [7] there are many approaches and measurement technologies to predict driver's behaviors. The most commonly used measurement can be categorized upon the monitoring instrument into :

- Video-based sensors

- Physiological signals sensors



(a) Video-based sensors (used in this thesis)

(b) physiological signals sensors

Figure 3.1: Illustration of difrent Driver monitoring approches

3.2.1 Physiological Signals Sensors

Physiological signals of the driver are the most accurate solutions, they can be used to measure his vigilance level since these signals originated from human organs such as brain, eyes, muscles, and heart that can indicate the fatigue and alertness level in real-time as depicted in Figure 3.1b. Physiological measures can be recorded from different organs that show visible correlation with the wakefulness/drowsiness state of a person. This includes [7]:

- **Brain activity**,which can be captured by electroencephalography (EEG) or Near Infrared Spectroscopy (NIRS).
- **Cardiac activity**, monitored through electrocardiography (ECG) and Blood Pressure signals.
- **Ocular activity**, measured by electrooculography (EOG)

3.2.2 Video-based sensors

To determine alertness/drowsiness level of driver as illustrated in Figure 3.1a. The behaviour of the driver is mainly monitored through a camera and thus this approach is known as video-based measure. Visible symptoms of fatigue and sleepiness can be observed when driver becomes drowsy through measuring its abnormal behaviours. Research on fatigue and drowsiness detection using driver behavioural monitoring focused on three main measure: *Eyes state*, *Face expression*, and *Head position*.

3.2.3 Evaluation

The following evaluation and ranking are based on our online search of driver monitoring technologies and we belive that it is not the only evaluation method.

According to [7], physiological sensors make it possible to alert driver at earlier stages of drowsiness and thereby prevent many drastic accidents [17]. Physiological measures have been shown to be reliable and accurate since they are less impacted by environmental and road conditions and thus may have fewer false positives [32].

On the other hand, video sensors technology is user friendly and can be mounted comfortably in various areas inside a vehicle also,it has the lowest coast. The common limitation is lighting conditions.

Technology	Video-Based Sensors	Physiological Signals Sensors
Coast	++	+
Ease of Use	+++	+
Intrusiveness	+	+++
Accuracy	++	+++

Table 3.1: shows the evaluation results of the two above described Driver Monitoring Technologies. The (+) symbol represents the rating level

3.3 Driver Drowsiness

Feeling sleepy or tired during the day is commonly called drowsiness. Drowsiness can lead to additional symptoms, such as forgetting or falling asleep at inappropriate times, especially in the case of driving because it leads to a car accident. Drowsiness in general is accompanied by warning signs that differ from one person to another, such as yawning[1] or blinking frequently, nodding[2], drifting off the track, and the most critical sign of drowsiness is closed eyes.[4]

3.4 Driver Inattention

Driver inattention or distraction occurs when a driver engages in a secondary activity that interferes with the primary task of operating a vehicle. Drivers can be distracted in many ways by things inside or outside of the vehicle. There are three categories in which driver distraction [19]:

1. **Visual** : taking your eyes off the road, such as :
    - Texting
    - Other distractions outside the vehicle

[1]yawning is a response to fatigue, it is characterized by opening up of mouth which is accompanied by a long inspiration, with a brief interruption of ventilation and followed by a short expiration.

[2]nodding also is a response to fatigue, it is characterized by lower or raising the head slowly and briefly

2. **Physical** : taking your hands off the steering wheel, like :
    - Adjusting vehicle controls, such as the air conditioner
    - Eating or drinking
3. **Cognitive** : taking your attention away from the driving task. Cognitive distraction usually accompanies physical and visual distractions.

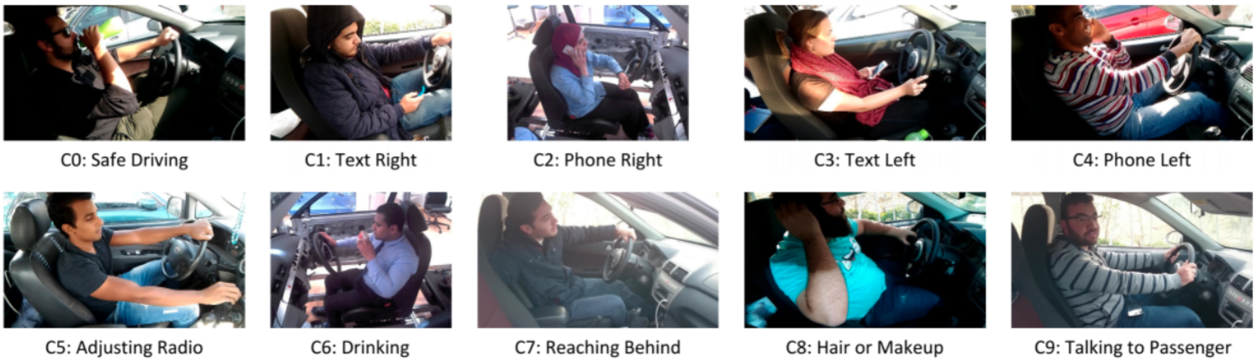


Figure 3.2: Distracted Driver Sceenes

the common sign of driver inattention is the head position, when the driver looks in a direction other than the road it means he is not focusing on the road and this is too dangerous see Figure3.2.

3.5 Proposed work

In this section, we propose methods for detecting driver's drowsiness and distraction using computer vision techniques discussed in chapter 2 with other algorithms that aim for improve performance and accuracy (ACC), then !!!!!!!! gol beli rah ndiro comparaison w rah ndiro analyse l kol methode w nokhorjo b natija ... (!!!! ahki ela system d'alarme)

Before anything, the main goal of this work is to invoke audio-alert when a bad thing happend .. !!!!

For the application phase, the driver's surveillance camera is mounted in front of the driver's face it can be placed in the areas of the steering wheel or it can be hung on the rearview mirror.

3.5.1 method 1 : Haar Cascades

Viola-Jones facial detection technique, commonly known as Haar Cascades. This work was done well before the beginning of the era of deep learning. But it's a great job in comparison with powerful models that can be built with modern deep learning techniques, especially in terms of speed. The algorithm is still used almost everywhere.

Haar Cascades in general is an object detection algorithm uses haar features. Haar Features were not only used to detect faces, but also for eyes, lips, license number plates, etc. [1].

For a good detection rate, we need a strong classifier that is trained in using a large set of *positive*[3] and *negative*[4] samples of a face, however, *OpenCV*[5] can perform face detection out-of-the-box using a pre-trained Haar cascade that is mean OpenCV's Haar cascade has already picked the best haar-like features for face detection, eyes detection, etc.

This ensures that we do not need to provide our own positive and negative samples, train our own classifier, or worry about getting the parameters tuned exactly right. Instead, we need to focus on improving speed, accuracy and finding solutions for challenging conditions [15] [8] [1].

In the application of this method, we use Haar-like detectors provided by *OpenCv*, a haar-like detector takes as argumments [8]:

- **Image**: matrix of the type *CV_8U* [6] containing an gray-scale image where objects are detected.

[3]Positive data points are examples of regions containing a face

[4]Negative data points are examples of regions that do not contain a face

[5]Open Source Computer Vision, is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez.

[6]CV_8U is unsigned 8bit pixel, a pixel can have values 0-255, this is the normal range for most image and video formats.

- **ScaleFactor**: parameter specifying how much the image size is reduced at each image scale.
- **MinNeighbors**: parameter specifying how many neighbors each candidate rectangle should have to retain it.
- **MinSize**: minimum possible object size. Objects smaller than this are ignored.
- **MaxSize**: maximum possible object size. Objects larger than this are ignored.

By default a haar-like detector returns 4 values [8] x-coordinate, y-coordinate, width(w), height(h) of the detected target object, these 4 values represent 2 spatial points (x, y) and $(\quad x + w, \quad y + h)$ of the rectangle that contains the object see Figure 3.3.

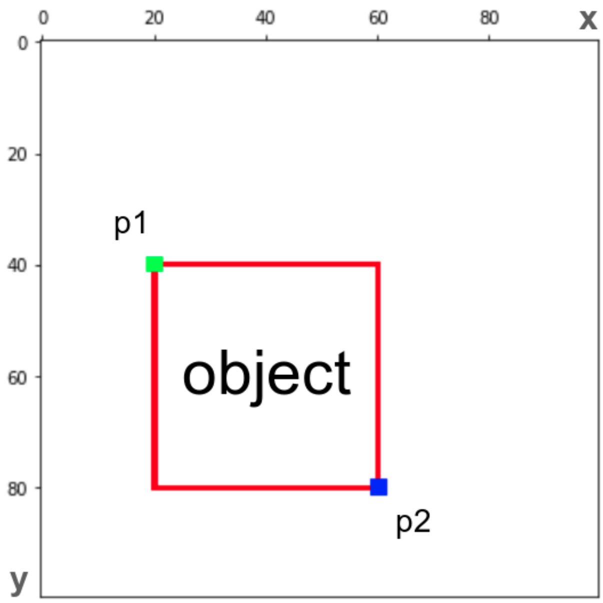


Figure 3.3: Illustration of Bounding Box in object detection using haar-like detector

Taking the Figure 3.3 as an example. After a successful object detection, a haar-like detector returns : $(x, \quad y, \quad w, \quad h) = (20, \quad 40, \quad 40, \quad 40)$ respectively.So, $p1 = (20, \quad 40)$ and $p2 = (20+40, 40+40) = (60, \quad 80)$. However, these bounding boxes are useless in the case of driver monitoring, the most important thing is to alert the driver in real time when he is asleep or distracted.

In the application stage of this method to serve as a driver monitoring technique. we used two Haar-like detectors, the first one is a face detector to check whether a given image contains a face or not, In other words, the face detector can check whether the driver is focusing on the road or distracted. Assuming the camera is in front of the driver's face, if the face detector finds a face on a given image, This means that the driver looks forward and if the detector cannot find a face that means that the driver is distracted by looking in a direction other than the road, which is a dangerous situation, especially when driving fast.

The second Haar-like detector is an eye detector, which can find an "open" eye in a given image. Using this condition, the eye detector can check if the driver's eyes are open or closed, simply by performing an eye detection if the detector finds one or both eyes that means the driver is focusing on the road if not, This means that one or both eyes of the driver are closed and this is the critical sign of driver's drowsiness which requires an alert to wake up the driver.

As this method works with images, and the camera feeds the system with a video stream which is basically a sequence of images we processed the video frame by frame, and for each frame, we performed haar cascades see Figure3.4.

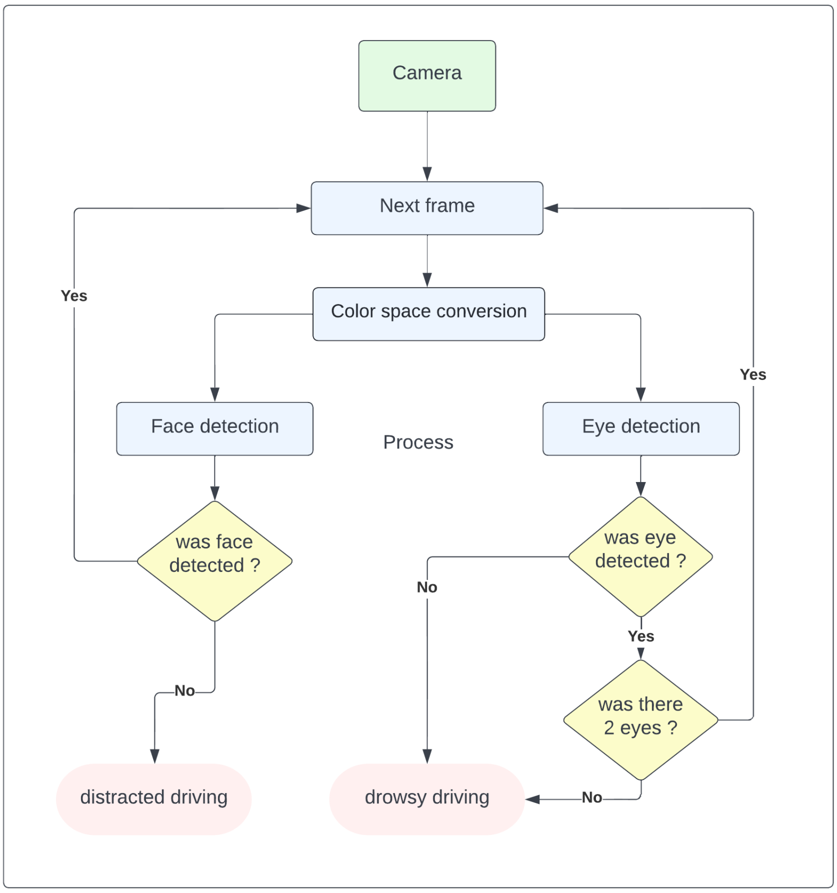


Figure 3.4: Initial flowchart for driver monitoring with haarcascades

The first implementation of this method was done with the following algorithm under difrent lighting conditions :

Algorithm 3.2 first algorithm for face/eyes detection using Haar cascades

- Load haar-like detectors provided by OpenCv
- Foreach frame from video input :
  1. Change color space from RGB to Gray
  2. Face detection with following parameters :
     - **Image** : frame
     - **ScaleFactor** : 1.1
     - **MinNeighbors** : 2
  3. Eye detection with following parameters :
     - **Image** : frame
     - **ScaleFactor** : 1.1
     - **MinNeighbors** : 2
  4. draw bounding boxes on frame
  5. display frame

As expected, the first implementation of face/eyes detection with haar cascades in real-time using a laptop camera was very fast and computation friendly. Despite of the speed, both detectors showed few FPs (Flse positives) see Figure3.5.

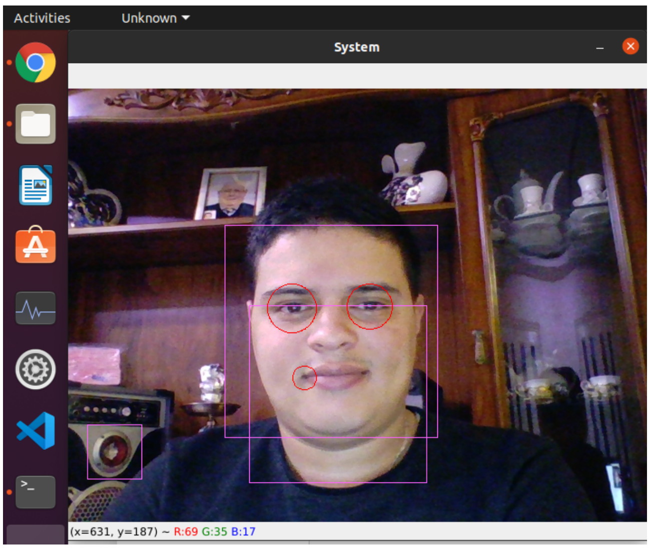


Figure 3.5: face/eyes detection with Haar cascades

So, By applying haar-like detectors for face and eyes detection, we gained time and speed wich allows us to use this method in real-time driver monitoring. However, we also need further improvements, as we still may encounter issues of either missing detections (FNs) or false detections (FPs). In the case of this study FNs does not a serious problem because, in the worst case they only invoke alerts, Unlike FPs. False-positive means the driver is maybe distracted but the system can detect the face/eyes of the driver and this is too dangerous because no alert will be invoked in this case. So, in the optimization section we will focus more on decreasing the (FPs), and improving algorithm robustness under variable lighting conditions.

Problems and limitations

Haar cascades are notoriously prone to false-positives, the Haar-like classifier can easily report a face in an image when no face is present under normal lighting conditions, eye classifier is worse in false detection because there are many parts of face have the same color and shape property of the eyes for example the *Oral commissure*[7]. The situation becomes even more complicated when a part of the driver's face is brighter than the other part (due to light falling in through a side-window), making eye status detection extremely difficult. In addition to that we noticed that sometimes the performance drops dramaticaly after a while due to the big number of computations per frame.

Another important thing to note here , along the testing of this method we found a problem which is sometimes a face or an eye can be detected 2 times in the same instance. this situation apears generraly when trying to approach the camera, sudenly a bounding box apears in the screen including both the object (face / eye), and the other bounding box.

In the next section we propose few methods to tackle the above issues which mainly summarized in :

- False positives for both face and eyes
- double detection, this problem is when an object has been detected two times in the same instance
- detection fails under bad lighting conditions

Optimization

Haar-based detectors use haar features to detect objects. So it is very common to find an object in the image that has the same properties as the face, this problem has also encountered in eye detection. In the first part of this section, we focus to improve the accuracy of the algorithm by reducing Fps detections. In the second part, we define a method to avoid the double detection

[7]The commissure is the corner of the mouth, where the vermillion border of the superior labium (upper lip) meets that of the inferior labium (lower lip).

problem based on spatial cordinates. Then we propose a night vision mode to improove the algorithm robustness under bad lighting conditions.Finally we work on improoving speed of the algorithm.

The main factors that control the object detection process are the detector's arguments (ScaleFactor, MinNeighbors, MinSize, MaxSize). Starting with ScaleFactor, since the classifier was trained with a fixed object size, it can detect the object with the same size that was used during the training phase [14].Usually in the application phase, the size of the object may be smaller or larger than what was used in training, this may be related to the distance between the object and the camera. To ensure the detection of object in different scales[8] there are two terminologies: Image Pyramid, and Scale Factor.

Image pyramid and ScaleFactor specifies how much we reduce the image size each time we scale (i.e., ScaleFactor = 1.5 means reduce the input image size by 5%). Or, how much we increase the sliding window size (i.e., ScaleFactor = 1.2 means increase window size by 20% at each iteration), see Figure3.6.

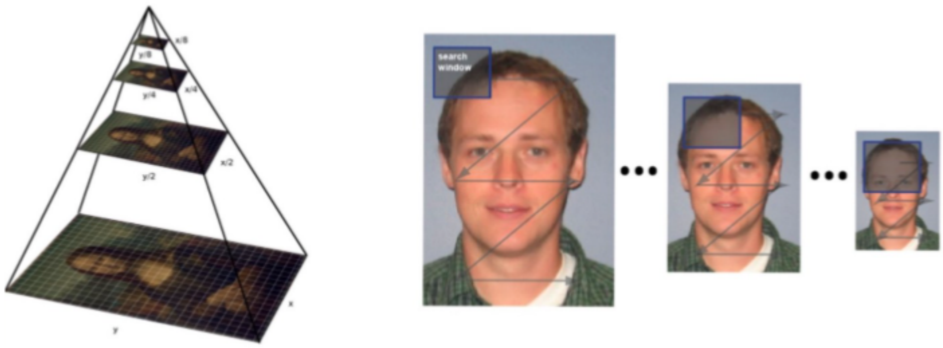


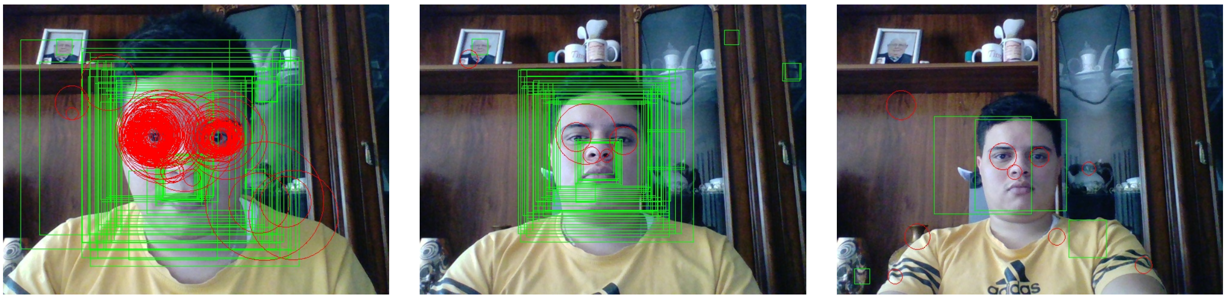
Figure 3.6: Image Pyramid and Scale Factor

Generally, the smaller the ScaleFactor (SF), the more detailed the research, and the higher the calculation costs.

As object detection works within the image pyramid combination (multi-scale) , the detector can find multiple True response for a single region of an object. These responses are object candidates in a given region, here the detector use the Minimum Neighbors (MNN) parameter to confirm the detection by summing the number of rectangles candidates in that region this value must be higher than the MNN to decide whether there is an object or not. So, MNN is a parameter that can enhance algorithm accuracy in case of using high values, thus reduce detection rate. Also, decreasing MNN value increase of detection rate wich means increasing the false-positives (FPs) rate as well See Figure 3.7.

there are also two additional parameters which are Minsize and Maxsize, thier utility is to

[8]Scale is the relative size of different objects in relation to each other or a common standard [3]



(a) Face MNN = 0 , Eye MNN = 0 (b) Face MNN = 0 , Eye MNN = 1 (c) Face MNN = 1 , Eye MNN = 1

Figure 3.7: detections using various MNN values. Dir eyes whdha w face whdha !

control the detection by the size of the detected object (width , height) if the size fall outside the range [minsize ... maxsize] the detection are ignored, only objects which have size in that range are accepted. These arguments are very important for decreasing FPs detection rate. In this point we supposed that the driver will be sitting about 1m far from the camera, and we recorded the width and height of the detected face and eyes, in general the distance between the driver's face and the camera is constant this what insure that we dont need a dynamic values for MinSize and MaxSize only in a specifique cases like changing the vheicle in this case we can simply adjust camera's position. (here put figure illustate the relation between distance and object size !)

(a) distance = 20cm (b) distance = 60cm (c) distance = 100cm

Figure 3.8: Face/Eyes sizes for diffrent distances of the driver and the camera

After finding the good values for (MaxSize, MinSize) relative to the distance between the driver and the camera, these values help us to reduce FPs rate by ignoring False detections based on the face/eye size.

Now we need to find optimum parameters for ScaleFactor and Minimum Neighbors, in this step we recorded SF and MNN values under diffrent conditions and we realized that these parameters have a relation with luminosity level and the distance so they can not be constants. Instead we need to record the best values for diffrent situations esspecially lighting conditions because in general the distance do not change, then make these parameters dynamique with the help of a criterian that will estimate the luminosity level and decide which couple of values for SF and MNN are optimum based on the estimated luminosity level.

For the luminosity level estimation, there are many diffrent methods like deep learning model for *Day-Night Classification* [13]. eventhoug the deep learning approach is the most accurate for this specifique task, it add more computation cost wich will slow down the algorithm. So, we used the image basic statistiques to solve this issue.Figure 3.9 shows face/eyes detection with haarcascades under non-ideal lighting conditions.

(a) (b) (c) (d)

Figure 3.9: face/eyes detection with Haar cascades under diferent lighting conditions

In images (a) and (b), the detection was performed for the face and both eyes, but in the image (c), only the face was detected successfully, both detectors failed in the image (d). For a first test we assumed that the image (b) is the minimum luminosity value for face/eye detection, so we calculated basic statistics (Mean , Variance , Energy) for each image in Figure 3.9 in grayscale color system.

	image (a)	image (b)	image (c)	image (d)
Mean	68.47	13.82	6.71	5.66
Variance	2615.06	428.27	86.71	54.06
Energy	111.55	42.29	31.41	29.59

Table 3.2: Table shows basic statistics for gray scale images of Figure 3.9 under difrent lighting conditions

we need a criteria to classify lighting conditions into (good/bad). According to Table above, the Mean and Energy show a difference between the images, but we cannot use them because the difference between the values of the image (b) (minimum) and the image (c) of the Mean or Energy is too small and can be easily affected by light, however, the Variance shows a large difference between the image (b) and the image (c). So we used the Variance as ctiteria for the classification of lighting conditions.

To find the optimum values in diffrent conditions which was done in terms of TP and FP rates with the help of python programming language wich allows us to see algorithm results with the possibility of changing parameters values in real-time.

	Good LC		Bad LC		All LC	
	SF	MNN	SF	MNN	MinSize	MaxSize
Face	1.3	1	1.1	1	(116,116)	(331,331)
Eye	2.2	3	None	None	(44,44)	(44,44)

Table 3.3: Table shows optimum values for detector parameters under difrent lighting conditions

Due to the fail of eyes detection under bad lighting conditions, we propose a night vision system based on image preprocessing technique called "Power Law Transformations" (Gamma Correction) with the help of the Variance. (AHder ela gamma correction w kifeh takhdem) w dir des photos ela resultats

Bibliography

[1] Girija Shankar Behera. *Face Detection with Haar Cascade*. 2020. URL: https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08.

[2] Hazrat Bilal et al. "Real-Time Lane Detection and Tracking for Advanced Driver Assistance Systems". In: July 2019, pp. 6772–6777. DOI: 10.23919/ChiCC.2019.8866334.

[3] Steven Bradley. *How To Use Size, Scale, And Proportion In Web Design*. 2010. URL: https://vanseodesign.com/web-design/size-scale-proportion/#:~:text=Size%20is%20the%20physical%20dimensions,Proportion%20is%20harmony%20of%20scale..

[4] National Center for Chronic Disease Prevention and Division of Population Health Health Promotion. *Drowsy Driving: Asleep at the Wheel*. 2021. URL: https://www.cdc.gov/sleep/features/drowsy-driving.html.

[5] Charlie Klauer et al. "The impact of driver inattention on near-crash/crash risk: An analysis using the 100-car naturalistic driving study data". In: (2006).

[6] Minakshi Kumar. *DIGITAL IMAGE PROCESSING*. Indian Institute of Remote Sensing, Dehra Dun.

[7] Vero Messaoud Doudou Abdelmadjid Bouabdallah. *A Light on Physiological Sensors for Efficient Driver Drowsiness Detection System*. 2019. URL: https://hal.archives-ouvertes.fr/hal-02162758/document.

[8] OpenCv. *Cascade Classifier*. 2017.

[9] World Health Organization. *Road traffic injuries*. Fact sheet. 21 June 2021.

[10] World Health Organization. *World report on ageing and health*. Publications. 2015, 246 p.

[11] Kunal Patil. *Speeding Up the Development of ADAS Systems with Model-Based Development*. 2016. URL: https://www.dspace.com/fr/fra/home/news/engineers-insights/blog_inc_adas_1607.cfm.

[12] Paul Pickering. "Radar and Ultrasonic Sensors Strengthen ADAS Object Detection". In: (2017). URL: https://www.electronicdesign.com/markets/automotive/article/21805470/radar-and-ultrasonic-sensors-strengthen-adas-object-detection.

[13] Nitin Rai. *ay-Night Classification*. 2020. URL: https://medium.com/@mneonizer/day-night-classification-a01a7d9af695.

[14] Rashmi Ranu. *Terminologies used In Face Detection with Haar Cascade Classifier: Open CV*. 2020. URL: https://ai.plainenglish.io/terminologies-used-in-face-detection-with-haar-cascade-classifier-open-cv-6346c5c926c.

[15] Adrian Rosebrock. *OpenCV Haar Cascades*. 2021. URL: https://pyimagesearch.com/2021/04/12/opencv-haar-cascades/.

[16] rsipvision. *ADAS sensors: 1. RGB cameras*. 2021. URL: https://www.rsipvision.com/adas-sensors-rgb-cameras/.

[17] Arun Sahayadhas, Kenneth Sundaraj, and Murugappan Murugappan. "Detecting Driver Drowsiness Based on Sensors: A Review". In: *Sensors* 12.12 (2012), pp. 16937–16953. ISSN: 1424-8220. DOI: 10.3390/s121216937. URL: https://www.mdpi.com/1424-8220/12/12/16937.

[18] Robert E Schapire. "Explaining adaboost". In: *Empirical inference*. Springer, 2013, pp. 37–52.

[19] lowmanlawfirm web site. *Driver Inattention and Distraction Explained*. 2017. URL: https://www.lowmanlawfirm.com/blog/driver-inattention-and-distraction-explained.

[20] Grant Maloy Smith. *Types of ADAS Sensors in Use Today*. 2021. URL: https://dewesoft.com/daq/types-of-adas-sensors.

[21] P. Viola and M. Jones. "Rapid object detection using a boosted cascade of simple features". In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Ed. by 10.1109/CVPR.2001.990517. 2001.

[22] Aptiv Website. *Mobility inside*. 2020. URL: https://www.aptiv.com/en/insights/article/what-is-adas.

[23] c3.ai website. *machine learning*. 2022. URL: https://c3.ai/glossary/machine-learning/classification/.

[24] Dynamsoft website. *Image Processing 101 Chapter 1.3: Color Space Conversion*. 2019. URL: https://www.dynamsoft.com/blog/insights/image-processing/image-processing-101-color-space-conversion/.

[25] Mazda Website. *Active safety technology helping prevent accidents*. URL: https://www.mazda.com/en/innovation/technology/safety/active_safety.

[26] Nishant Kumar. GeeksforGeeks website. *Digital Image Processing Basics*. 2021. URL: https://www.geeksforgeeks.org/digital-image-processing-basics/.

[27] OXTS Website. *ADAS sensors*. 2020. URL: https://www.oxts.com/adas-sensors/.

[28] Vivadifferences website. *Difference Between Sampling And Quantization In Digital Image Processing*. 2020. URL: https://vivadifferences.com/difference-between-sampling-and-quantization-in-digital-image-processing/.

[29] wikipedia website. *RGB color model*. last edit 2022. URL: https://en.wikipedia.org/wiki/RGB_color_model.

[30] Wimarshika Thamali Meduim website. *Sampling and Quantization in Digital Image Processing*. 2020. URL: https://wimarshikathamali1995.medium.com/sampling-quantization-in-digital-image-processing-8c4490357039.

[31] Wikipedia. *Automotive safety*. 2013. URL: https://en.wikipedia.org/wiki/Automotive_safety.

[32] Eugene Zilberg et al. "Methodology and initial analysis results for development of non-invasive and hybrid driver drowsiness detection systems". In: *The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007)* (2007), pp. 16–16.