Real time image recognition of facial features for detecting a true breath

Mälardalens Högskola

Akademin för Innovation, Design och Teknik

Adam Eriksson och Jonas Holmberg

Bachelor's Degree

Datum: 2015-05-27

Examiner: Mikael Ekström

Supervisor: Magnus Otterskog

Abstract

The effects of driving under the influence of alcohol can be severe. In an effort to reduce the number of drunk drivers, alcolocks have been deployed. However, if a driver tries to cheat the system and succeeds, the effects may cause severe damage to both humans and property. One way of beating the system is by fooling the breathalyzer that the breath produced doesn't contain alcohol. This is usually done using a device which either filters a breath, or by having a device that blows clean air into the alcolock. In this paper the problem of if it's possible to determine whether a true breath is made by using facial and feature detection, is answered. A true breath is an unobstructed, real breath from a person. True breath identification can be used in order to, for example, determine if a mouthpiece-less alcolock device is being used correctly. To solve this problem facial and feature detection was used. In order to find the best methods for facial and feature detection, research result in this area was used. The method used to solve this problem works by first tracking the face. The mouth is then searched for in the lower part of the face region. After that the mouth is found, the lips are to be extracted from the mouth area using various methods. The lip tracking is then used to determine the mouth state, it does this by calculating the distance between the different points produced by the lip tracking. The mouth state, mouth and face tracking are then used during the identification process to determine if a real person was present and if he/she made a true breath. The test fails if the mouth is closed during the time of breath or if there is no mouth or face present. Attempts to manipulate, result in a failed alcohol test. In the majority of the test cases the tracking and mouth state scored acceptable or better results. The performance of the system concluded that it's possible to complete the analysis in real time. The result of this thesis is a true breath identification system. It still has some issues in cases where the illumination and lighting is uneven or bad. Facial hair may also interfere with the tracking in some cases. Although there are still some areas that needs improving, this work was more of a proof of concept. This thesis shows that it is possible, under certain circumstances, to determine a true breath using facial and feature detection.

Acknowledgements

We would like to thank Hök Instrument for all the help and workspace they've provided. We would also like to thank Magnus Otterskog for helping and providing us with feedback.

Table of contents

1. Introduction

This report will cover the progress of the work that has been made during this project. The purpose has been to find out if it is possible to develop a method or technique to distinguish a real breath from a false breath when blown into an alcolock. A false breath is either produced by a device or through filtering a breath. Another important part of the work is to determine if the chosen solution actually worked as intended from the start, and if the result it produced was satisfactory compared to the initial assessment.

Alcolocks are used in cars to prevent a person with too high blood alcohol concentration level (BAC) from starting the vehicle and thus driving it. Alcolocks are devices that can be used by a private person with the intention of preventing them from driving drunk. It can also be forced upon people convicted of drunk driving, as the condition to allow them to drive a vehicle. Further use of the alcolocks can be for example in buses and trucks. This is so that the individuals who are driving these big vehicles are not doing so under the influence of alcohol. For these kind of vehicles, that could do a lot of damage if they were to be involved in an accident, an alcolock could be a good solution to prevent alcohol related accidents from happening. Since drunk driving is one of the bigger causes of death in traffic [1], it is something that is important to reduce. An alcolock is a good solution that can help dealing with this dangerous problem.

Even though alcolocks are good at reducing the number of drunk driving accidents the solution itself is not entirely immune to cheating. There exists ways to fool a breath test so that it can yield a result saying that there is no alcohol in the blood, when it is in fact the contrary. This means that alcolocks may not be able to protect themselves against all possible methods of manipulation. Therefore some alcolocks may be vulnerable to certain methods or techniques of manipulation and in that case it makes way for a potentially drunk driver to venture out on the streets.

With this basis, the foundation of the definition is formed. There exists methods of manipulation that does not work on all alcolocks, but what about the ones undetected? Is there another way to be able to detect if a true breath has been made by the alcolock user? This is what this report intends to answer. It presents a solution that uses facial and feature detection to make an assessment if there was a real breath made by the user, or if it was an attempt to manipulate.

This report will cover what similar work has been performed in this field so far. What conclusions have they reached, and were the results they achieved satisfactory. How does this kind of method work in a real environment, and from that try to form an understanding if this kind of method is applicable and able to do what it is intended to do? The report will cover what methods were used to solve the problem and how they were implemented. It will go into the difficulties that exist and how they can impact the solution. The performance of the method will be judged to see if it does what it is intended to do and if it actually helps with the problem. Lastly it will address the results from the work and draw conclusions regarding the initial problem definition.

1.1 Problem Definition

The purpose will be to answer the question, is it possible to develop a method that makes it possible to detect if a person has performed a true breath into an alcolock in real time. By real time is meant the time frame defined by a breath test, meaning the response time should not exceed 100 milliseconds. Because the basis of the method that will be used was predetermined to be facial detection, this report will also attempt to assess if this solution actually works in a good way. Is this really a good method for identification and will it actually add anything to the current security attributes of alcolocks? Or will it just be an annoying feature that does not really add anything useful?

Because the project will use facial detection there are several problems to assess. For the facial detection it will be important to find and correctly implement the methods that would work best for this kind of task. To detect a breath, the key aspects would be to correctly be able to detect the persons face, and then some of the different features of the face. In this case most likely the mouth. It is important that the identification process of the software is accurate and able to form a good assessment, if the breath that was made was a true breath or not. If it's not accurate enough then there would be no point of the system. On the other hand if it's too strict it may become too much of a problematic and time consuming process for the user, by proclaiming some breaths that were real as false. This may happen if the identification process was too particular about the judgment. Another aspect of this software is that it needs to be optimized. Since it is supposed to run in real time, it needs to be efficient and not too time consuming. If the number of images it can process each second is too low, it may become problematic for the system to estimate if it was a real or false breath.

1.1.1 Project boundary

These are the main points that need to be solved. For this kind of implementation to work in the end it needs to be used together with a mouthpiece-less alcolock device. This will also ensure there is a clear image of the face with nothing blocking it. Otherwise the proposed facial detection approach may not be able to do this in a good way. To solve the facial detection, existing methods will be used. The purpose is not to invent an entirely new facial detection technique, but to use the existing ones in a way that solves the problem at hand.

2. Background

Facial detection is something that is used in different applications and devices now nowadays. The most common facial detection that can be found, is probably the simple one that can be found in most cameras. When a camera is pointed at a person it will often display a rectangle around the persons face. This is facial detection although in a simple manner. But it's just one use of facial detection. The facial detection will be used, together with other facial feature extraction techniques, to determine if a person who blows into an alcolock is trying to fool the alcolock. The system that is going to be developed is supposed to work as an extra line of defense for the alcolock. Most alcolocks have security measures built into them for detecting if it was truly a real breath, but it may not be able to detect every method of manipulation. Therefore this system will try to detect these other forms of manipulation.

Facial detection is an area where there exists quite a few different techniques on how to best detect a person's face. In extension to just detecting the face there are other methods for detecting different parts of a face, like eyes or mouth. This is something that will probably be relevant, especially the detection of a person's mouth. This report will take a closer look on other works that have used facial detection to solve various problems. This is done to get an understanding of where it stands today and what is possible to achieve with it. From there it will be determined if this is a solution that will be suited to solve the problem presented. This report will also look on some of the different methods for facial detection and feature recognition. That is done to get an idea what would work best and if it will be applicable.

Another thing that is important to do is to differentiate facial detection from facial recognition. Detection is the act of just being able to recognize if there is a face in the given image frame. Facial recognition is being able to tell who the face belongs to, in other words being able to identify a person. This project will focus on the facial detection, being able to identify a face and other facial features to try and make an informed decision if the person made a real breath.

There is a possibility to extend this software with facial recognition. This would be useful to for example detect if the person that is blowing into the alcolock is actually a person authorized to maneuver the vehicle it is installed to. The basis for facial recognition is facial detection. Therefore to extend the software and add a facial identification part would not be impossible. Nevertheless the main focus is the facial detection part and that is what the work will be focused on.

In this report the current alcolocks on the market will be studied. This will be done in order to find the strengths and weaknesses that these systems possess. The purpose of doing this is to be able to counter the weaknesses and also utilize the strengths.

2.2 Current Alcolock systems

The alcolocks on the market are available both for private and corporate use. None of these locks utilize facial recognition or detection however. The systems that have been looked at fulfill the European standard. Most of the systems are mouthpiece variants, which require the user to blow into the device using a mouthpiece. Most of the systems have safety measures in place to protect the lock from tampering. However, some scenarios are not covered by these defenses.

2.3 Do alcolocks help?

Yes, alcolocks help with reducing the number of drunk drivers and they also help with reducing the amount of relapsing drunk drivers [2]. Since many convicted drunk drivers choose to have an alcolock installed [3] it's important that the device can't be tampered with. Furthermore the alcolocks also act like a reminder or mental block to make the potential drunk driver rethink [4].

2.4 Related work

In this section related work and patents will be studied. The purpose of this is to see if there are any patents that might be like this one and if that might cause any issues. Another reason for this is too see how others have chosen to apply the technology. Also if they've done anything wrong or right that might be helpful.

2.4.1 Related work

On the subject of identifying the person driving and matching that with the person performing the test there has been some work [5], [6]. By using facial recognition they match the person who does the test with the person who is driving. In case there's not a match, the car will either not start or requires a retest. In [5] a picture is taken of the driver when the car is started and when the alcolock is being used. If the test is accepted the system will take another picture when the car is stopping. This picture will be matched to the previous one. If it is not a match, a retest has to be made. The problem with these is that the system cannot tell whether the driver is somehow manipulating the system by not providing a true breath.

Facial recognition is not only used to identify drunk drivers [7], [8]. Since facial recognition can be applied in many fields, studies will also be made in other fields where it has been applied. By doing this, the strength and weaknesses will be studied in order to see if any of that can be used or avoided in this work. Facial recognition is used by the British police in order to identify persons of interest [8]. This project however has not been very successful, since not a lot of individuals have been found. The reason for this is that the cameras are often badly placed and the camera quality is often too low. This is an important weakness that needs to be circumvented if the facial analysis is to be correct and useful. In [8] there's also a mention of "skin texture analysis", here a person's facial patterns and lines are analyzed in order to provide a more accurate recognition rate [9] [10].

Another way to monitor drivers is through transdermal methods. This is done by placing a sensor on the driver and measure the driver's alcohol level through the skin. This can be combined with a regular breathalyzer [11]. Since transdermal methods constantly monitor the driver, he/she has to continuously beat the system in order to be able to drive.

A related work was done with a camera and a breathalyzer in order to check if a driver was tired or under the influence of alcohol [12]. To see if the driver was about to fall asleep facial recognition was used and checked if the eyes were opened or closed. The alcolock used in this system was a MQ-3 alcohol gas sensor. In the case of a failed test, an alarm was triggered and the car couldn't be started. This work gives some indications that it would be possible for these kind of systems to be implemented in vehicle environment.

There have been studies on the subject of driver drowsiness [13]-[19]. These studies use facial detection, specifically eye detection to check the driver's status. Many of these studies try to check if the eyes are opened or closed. This is something most interesting because it gives a good indication to that it's possible to utilize facial detection in a vehicle environment. However, the state of the mouth will be the area of interest, since an open or partially open mouth may indicate a true breath.

In [20] eye tracking is used to monitor the driver. The Nystagmus effect is used as an indicator to see if a driver is under the influence of alcohol. Even if this project is not much like the one described in this paper, it still shows the strength of facial detection and that it can be used in real time.

Some work that is close to this paper, is the work done on yawn detection [21]-[25]. Yawn detection is used to stop drivers from falling asleep while driving. But there are more fields of application. Such as in [23], here the mouth of the driver is tracked to determine if the driver is talking, to prevent drivers from talking on the phone while driving.

The paper [21] brings up the Support Vector Machine (SVM). The SVM is trained to recognize how an open mouth as well as a resting mouth looks. In [24] they've succeeded in creating an algorithm which increases the detection rate by 20%, putting it at 95% to be used in yawn detection. In [24] moods and mouth movements can also be read, however with too many limitations to be useful. These are all quite close to the work done in this report. What separates them is that none have used facial detection in the way that this work will use it.

2.4.2 Related Patents

In this section patents related to this work will be analyzed. This is done to see if there's any overlap with any other works out there.

Impairment detection and interlock system with tester identification [26] uses a camera built into the alcolock device. The camera records the person making the test. The recording is used to verify the identity of the one doing the test. This is not a real time solution.

Device and method for recognising the correct usage of an alcohol measuring device by a driver in a vehicle [27] uses a camera to verify correct use of a breathalyzer. The camera is placed to identify the driver, thus it is not part of the device itself. The camera is used to document the driver and not only for tampering or manipulation prevention.

Breath alcohol ignition interlock device with biometric facial recognition with real-time verification of the user [28] uses facial biometrics features to identify the driver. Identification is done by comparing this scan with stored authorized drivers. If the scan doesn't match a verified and approved driver, the test is failed and the vehicle cannot be started. This is used with an alcolock, after the initial scan the system decides whether to make an alcohol test or not.

Monitored ignition lock [29] aims to make the alcolock regulated, so that only certain people can use it. This is to be decided via identification code. It communicates wirelessly with a management system, which controls the access to the car. Uses the camera only to photograph the driver when using the substance detector.

Ignition interlock breathalyzer [30] monitors the driver with two cameras. One camera to identify the driver and a second to identify the one making the alcohol test. Uses a relay box to record tampering attempts as well as an accelerometer to record attempts to destroy the device.

Device and process for recognizing correct use of an alcohol-measuring device by a driver in a vehicle [31] uses a control unit, analysis unit and an alcolock device unit to decide if access should be allowed. A camera is used to detect the driver's face and stored by the control and analysis units. When a test has been made these units decide if access should be allowed.

Automated recognition algorithm for detecting facial expressions [7] is, even though this patent isn't related to alcolocks, its functionality is interesting. Here the scanning of micro-expressions, expressions lasting only fractions of a second, is done. This is in order to decide the mood an expression symbolizes.

Automatic face detection and identity masking in images, and applications thereof [32] is used to identify faces and their orientation in order to blur and/or replace them. The blurring is based on the orientation of the face.

2.4.2.1 Conclusions Regarding Related Patents

None of the patents above use the camera the way this project does. Even though many of them used cameras in combination with alcolocks, they didn't use them in the same way as this work intends to. Also none of the works presented above are for the purpose of direct detection of manipulation attempts. The conclusion is that this work is unique and applies existing technologies on a new area.

2.6 Facial detection

In this section the background of the subject of facial detection will be provided. The issue of determining if a true breath is made will be largely dependent on facial detection. By going through the different facial detection techniques as well as the libraries that implement them. Here the difficulties and requirements will be presented to further help with building a background.

2.6.1 Difficulties with facial detection

There are certain aspects of facial detection that needs to be avoided or in other ways solved for the technology to work correctly.
These main points of interest are [33]:

- **Camera angle.** The detection of a face depends on the camera angle. If the angle is wrong no face will be detected.
- **Presence of facial features.** Facial features such as beards, mustaches or glasses can create issues during detection. Due to these features differing shapes, colors and sizes, it is difficult to create a good method for countering this issue.
- **Facial expressions.** The face is affected by a person's expressions. This can make detecting a face harder, at least tracking it.
- **Blockage.** Faces being partly obstructed by other objects makes detection difficult.
- **Image rotation.** The angle of the image itself will affect the outcome of the detection.
- **Image quality.** This is a most important point. The higher the quality of the image the higher the quality of the detection.
- **Light.** The way that light hits and reflects on a face may cause issues when trying to detect it.

2.6.2 Techniques

The different facial detection methods can be put into a few main categories [33], [34], [35].

- **Knowledge-based methods.** These methods are based on rules that describe a face and its features. These methods are primarily used to detect faces.
- **Feature invariant approaches.** These methods tries to find the information it needs from things like face shape, color and facial features, such as nose and eyes. These methods are primarily used to detect faces.
- **Template matching methods.** Works by using several standard facial patterns for faces and facial features. These kind of methods are used both for detection and recognition.
- **Appearance-based methods.** Models are created by using sets of images. Once the models have learned they're then used for detection. These are mainly used for recognition.

The detection of faces will be most important. Because of this the Feature-invariant method will be most relevant, since it's primarily for detecting faces.

In this section some of the algorithms for detecting faces will be researched. This for finding the strengths and weaknesses of them. Different ways for optimizing face and feature detection will also be brought up. This for the purposes of fulfilling the real time aspect.

2.6.3 Viola-Jones Object Detection

Viola-Jones [36] is a face detection framework with fast face detection. This method is implemented in environments such as OpenCV and MATLAB. The primary reasons for using this algorithm is:

- High rate of positive detection and low detection of false positives.
- Good for real time, since it's fast.
- The ability to detect not only faces but also facial features.

Viola-Jones is however only for face detection and not for facial recognition/identification.

However there are some weaknesses of Viola-Jones:

- Only effective in detecting faces from the front
- It has an issue with detecting faces at an angle
- Very light sensitive
- Issues with detection of the same face because of overlapping sub detections

Since Viola-Jones works well in real time. It will be useful in applications where the number of analyzed frames per second is important. This can be worth it if it's possible to sacrifice a little precision for faster calculations.

What the Viola-Jones method uses to detect features is Haar-like features [37] [38]. Haar-like features are a way to detect faces without a high performance cost.

By using several neighboring rectangular regions and then calculating the brightness level of the region, the image is split into sections. This is done for the entire image. Since the human face has a certain pattern according to Haar-like features, where for example the eyes are darker than the forehead or cheeks, it is possible to match a region in the image. In Figure 1 it is shown how the reference regions work.

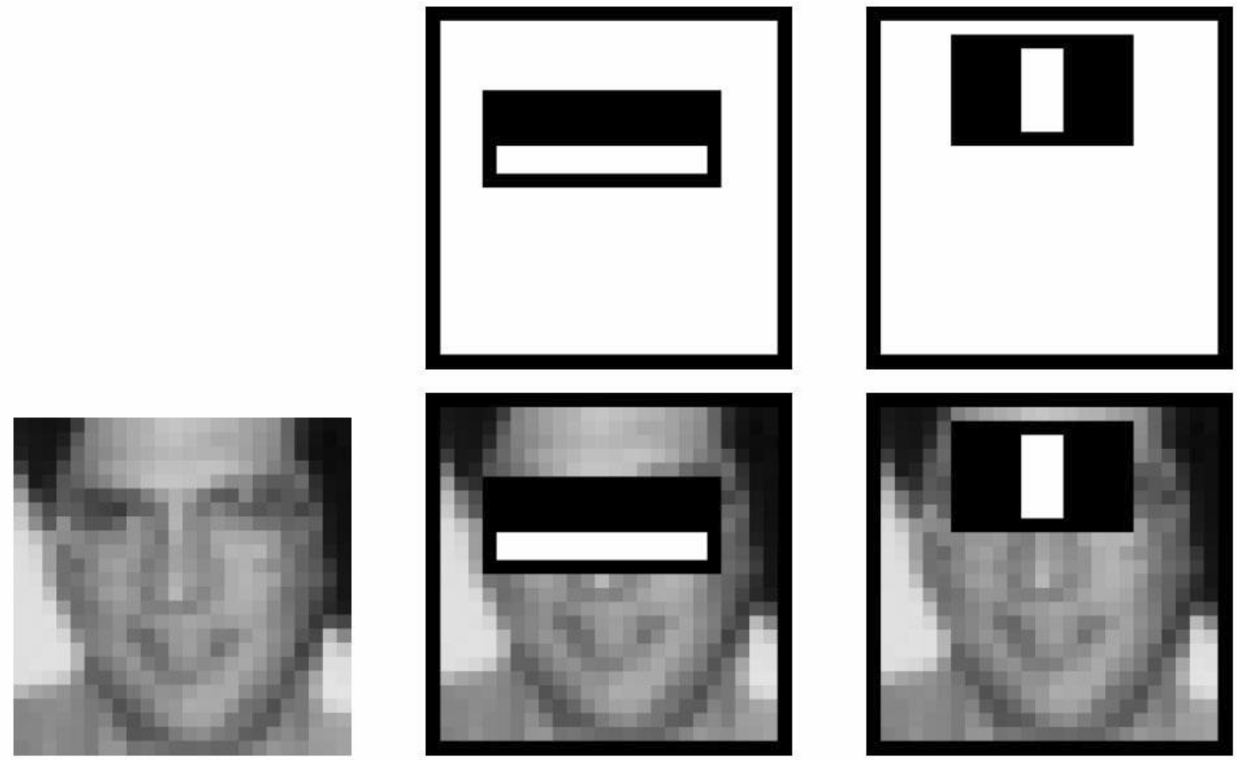


Fig. 1. In this image the Haar-like features (rectangular reference regions) used in face detection are shown. Image from [36].

One of the biggest strengths of Haar-like features is the fact that it's fast and efficient. Besides detecting faces, Haar-like can also be used to detect other objects. It works the same way, but with different reference regions. The object detection of Haar-like is most important, since facial features will be easy to detect. Because Haar-like features is based on light levels the detection is sensitive to light and angle. This requires the images to be gathered in a way that provides good quality.

2.6.4 Neural Networks

Another method for face detection is neural networks. Some work has been done in the area [39], [40], however these method is not adapted for real time solutions. Though there are some newer implementations that could work in real time [35]. In [35] skin is detected in order to decide if there is a face. This solution can achieve 6-20 frames per second. This solution is however light sensitive. If the background light is close to the skin color there could be problem with the skin detection and if the lighting is bad there can also be some issues in detecting the skin.

2.7 Libraries

In this section different face detection and face recognition libraries will be researched and evaluated. The strength and weaknesses of each library will be examined in order to see which ones are the most relevant.

2.7.1 OpenCV
Built in C++ and used both commercially and in academic work. There are many features (such as facial detection) available in this library. There is also a lot of information and documentation on this library.

2.7.2 libface
Built using OpenCV, doesn't require OpenCV though. libface has both facial detection and recognition. Unfortunately it is not recommended for real time, since its calculations are too slow. This library will likely not be used.

2.7.3 Aforge .NET
Computer Vision library built in C# for .NET framework. Much like OpenCV, however not as many features. If Accord.NET, machine learning framework, is added to this library it achieves a lot of the functionality that OpenCV does, at least on the facial detection and recognition part.

2.7.4 VXL(Vision X Library)
A collection of C++ based libraries. For VXL there are a lot of scientific libraries, VXL can be used for object detection and biometrics. This can be interesting since there are libraries within the image processing area.

2.7.5 CCV
A lightweight open source library that can analyze, track and detect faces. Could be adapted for this work. However since there already are more developed libraries this one will probably not be used.

2.7.6 Conclusions regarding libraries
OpenCV will most likely be used. If necessary VXL might be utilized. The reason for using OpenCV is because it's the most complete and used library as well as it's easy to use. OpenCV is also largely platform independent. Since there is a lot of information and documentation it will be easy to modify and use this library. This is needed, since the implementations need to be customized for a specific purpose.

2.8 Techniques for protecting and manipulating alcolocks
In this section the different methods of manipulating alcolocks will be brought up as well as the countermeasures in place. Many of the ways to manipulate are already fixed in the current market alcolocks.

2.8.1 Methods for manipulation
Many of the manipulation techniques that will be mentioned in this section have already been countered. However there are some that have yet to be solved.

2.8.1.1 Hot-wiring
By trying to hot-wire, or in other ways tamper with the alcolock systems wires and cables, the driver can gain access to the vehicle. This is however usually monitored and when cables and wires are cut or tampered with the data recorder will log this.

2.8.1.2 Drivers avoiding the breath test
A way to circumvent the system is to have a person not driving (a passenger for example), use the alcolock to give a drunk driver access to the vehicle. There's currently no countermeasure to this in place for most systems.

2.8.1.3 Carbon filter
By using active carbon, a manipulation device can be made. This device will work as a reversed gas mask, filtrating the air from alcohol positive to negative. This is one of the methods this work will attempt to prevent.

2.8.2 Methods for countering manipulation
The security of the alcolock system is important. Here a look is taken at how the current alcolocks are protected.

2.8.2.1 Data recorder
The alcolocks system utilizes sensors which report back to a data recorder. If the driver attempts to tamper with the system, this will be reported to the data recorder. When the alcolock is serviced, this information can be retrieved.

2.8.2.2 Photographics
The systems takes photographs of the driver before, during and after the breath test. It can then save the images to the data recorder and thus deciding if the driver is in fact the person who did the test. There are also versions where a failed breath test will capture a picture of the driver.

2.8.2.3 Continuous testing
By prompting the driver to make breath test throughout the operating of the vehicle. If the breath test is failed while the vehicle is active it will however not shut down, since this would be a traffic hazard. Instead the sound and light signals are used, forcing the driver to stop the vehicle. This information can also be logged by the data recorder.

2.8.2.4 Breath techniques
By forcing the driver to make the breath test in a certain way, security is ensured. This can be the blow-and-suck technique, where the driver after blowing into mouthpiece needs to suck back in order to complete the test. It can also be by hum-tone. This forces the driver to "hum" while making the test in order to succeed. These techniques require some training and attempts to make sure that only qualified people can use the vehicle.

2.8.3 Conclusions regarding security and manipulations
The ways to secure the system are many and provides most systems with good security. However the ways to fool the systems are also many, most of them are countered in the modern day systems. Using the carbon filter there are no real time counters to. This work will attempt to solve this problem. Also this work is unique in that it uses the facial detection to detect a true breath.

3. Method
To answer the question if it's possible to develop a method to detect if a real breath is made into an alcolock, the predefined criteria of facial detection will be used. Different methods of facial

detection will be researched. This will be done to find what's needed to track the face and possible features of the face. They will also be researched to find out what parts of the face that are actually needed to be tracked, in order to determine if a real breath has been made.

This will result in a very early stage software that in a simple manner can be able to detect if a real breath is made. This will be tested with a few techniques that could possibly fool an alcolock, to see if it is possible to detect the attempted manipulation. From the test results it will be determined if is feasible to use face detection in this manner, together with an alcolock to increase the difficulty of tricking the system.

The system will be implemented with the help of OpenCV, as concluded from the background study. OpenCV houses all the necessary tools needed and it is easy to use together with several different program languages. This project will utilize C++ for the implementation because C++ is one of the most common languages for OpenCV, along with Python.

4. Technical Descriptions

This part of the text will cover the developed method in regards to its design and some of the functionality. The initial plan will be covered and the thoughts behind it will be explained. Then some overview on how the tasks were solved and how the result, if it is a real breath or not, is obtained.

4.1 Proposed algorithm flow

This section will cover the proposed functionality of the system, what is needed to be done and at what point, in order to form an accurate decision of if a real breath was made. An overview of the method can be seen in Figure 2. When the process is started the camera will start to record. The program will fetch images from this live stream of images one at a time. Considering the system will operate in real time it needs to have an acceptable execution time on the feature detection. Otherwise it may not be able to obtain enough images for the breath identification. When an image is fetched from the video stream it will be processed, meaning the system will try to extract facial features needed for the identification. This part of the flow will be discussed in the next section. The processed images will be stored for a set time frame, no longer than a few seconds. They will be used when a breath is registered by the system. For the identification to be made accurately several images need to be analyzed. When a breath is detected the image capturing will cease and the currently stored images will be further processed for identification. The identification process inspects the image sequence and will look for certain conditions that needs to be fulfilled for the breath to be considered a real breath. If one of the set conditions is not met it will not accept it as a real breath.

Even if the detection concluded that it was not a real breath, the program must await the BrAC (Breath Alcohol Content) analysis from the alcolock before it can get the permission to try again. This is needed if the breath actually contained alcohol. The alcolock standard [41] states that a user cannot make another attempt at starting the vehicle until a few minutes have passed. Therefore there is no need for the system to keep running, thus the detection will stop. If the BrAC analysis comes back negative of any alcohol levels then two possibilities exist. An attempt at manipulation and the identification process concluding that it was not a real breath. The other case would be that the identification method just happened to get a false positive, it made a wrong decision regarding if it was a real breath or not. In this case the process flow would return to the beginning and a new breath could be made into the alcolock.

If the detection concluded that the supposed breath actually was a real breath, and not a manipulation, it will hand over all the responsibility to the alcolock. Then one of two things will happen. The BrAC analysis will say there was no alcohol detected and the car will start. The other possibility is that the breath contained traces of alcohol and the car will not start. This part does not need the detection algorithm so it can just stop right when the decision was made that it was in fact a real breath.

These are all the tasks the system should perform, described in a simple manner. The heaviest computational part will be the feature detection. This will take up most execution time and therefore what needs to most efficient for the solution to work properly. Would it happen that this part become a bottleneck, then the entire process could fail. The breath identification will be the most important part of the algorithm. Fortunately this part of the algorithm is not needed to be all too fast. The result from the BrAC analysis will take some time so it won't make a difference if the detection would take some time to calculate.

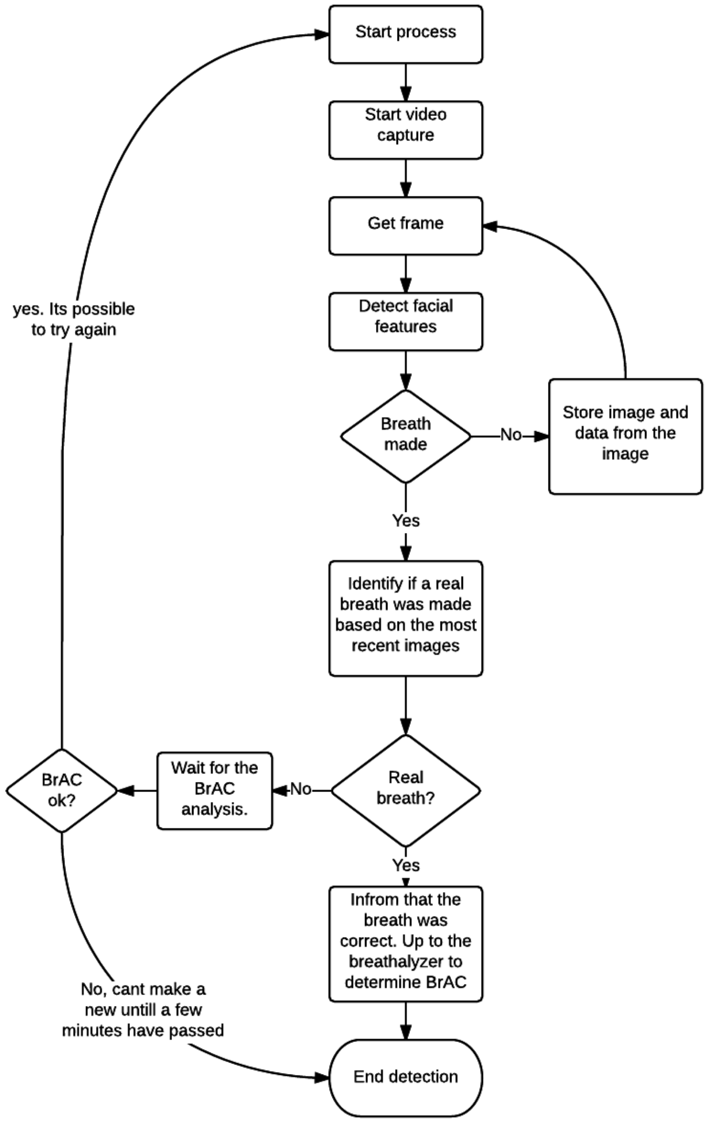


Fig. 2. Proposed flow of the algorithm to detect if a real breath was made into the alcolock.

4.2 Facial detection

The detection of the face and other features of the face is the first step in the process of breath identification. This will be done in different parts as illustrated by Figure 3. This image shows the necessary steps to get the needed facial features for the breath identification process. This section will present the chosen methods to solve these requirements but may also mention some of the considered solutions that were discarded and why.

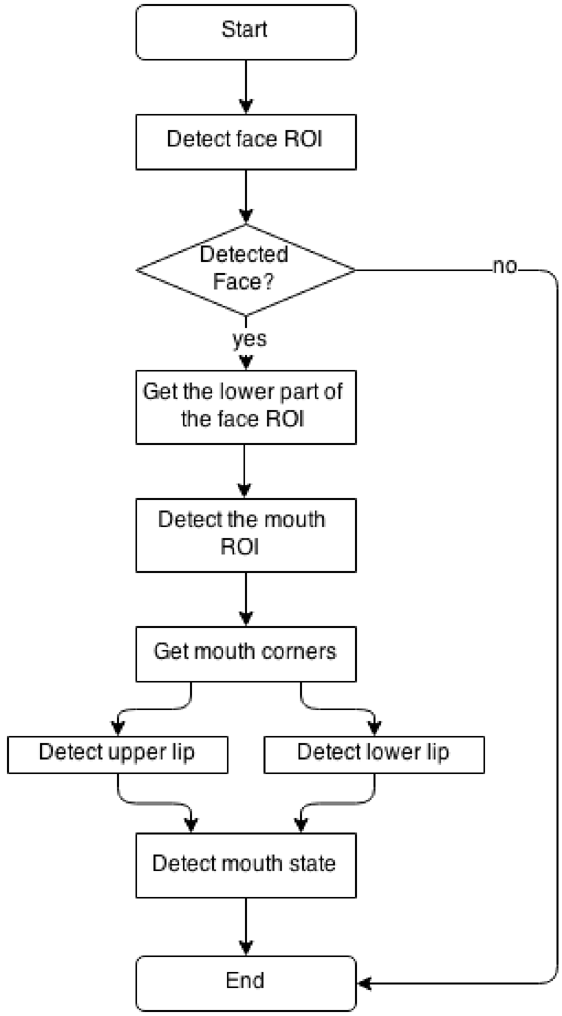


Fig. 3. The flow of the facial feature detection for each frame.

4.2.1 Detection of face and ROIs

The initial step of the feature detection is to detect the regions of interest (ROI). In this case these regions will be the face and the mouth of said face. The mouth ROI will later be used for the detection of the lips. Since this is the first step of the detection and it makes calculations on the whole original image when trying to detect the initial face, it will also be the most computationally expensive calculation. There are some different methods available but the most appropriate solution was to use Viola-Jones [36] detection. One of the reasons why it is good is that it works well in real time. That is one of the essential parts of this project and Viola-Jones handles it in a good way. Another good thing about this method is that it can be used not only for detection of faces, but also other facial features like mouth, eyes and so on. Some of the drawbacks with this method is that it can only detect faces from a frontal view and it can be pretty light sensitive. That it can only detect faces from a frontal view is not a problem however, because that is what is intended. The lighting will become difficult in almost every solution [33] and it's one of facial detections greatest difficulties.

When the initial face has been detected the next step is to try and find the location of the mouth region. To optimize the calculations and the speed of detecting the mouth area, the initial face ROI will be divided into different parts [37], [42]. Instead of searching the whole image again to find a possible mouth it is enough to just search in the face ROI. In this case, the face ROI itself is divided into smaller parts where the different features likely are positioned. The image space that needs to be checked for these features is severely reduced. Doing this greatly improves the speed of the execution, which again is very important when doing the calculations in real time. By doing this, the number of false positives will also be reduced. This is because the area where the mouth is searched for is now in the lower part of the face region. Therefore, if a mouth is found, it is likely that it's a real mouth and that it belongs to the face. It is important to try eliminating false positives. If there are any, it will impact the identification process in the later stages. Therefore checking if a found feature is where it most likely should be is a good thing to do.

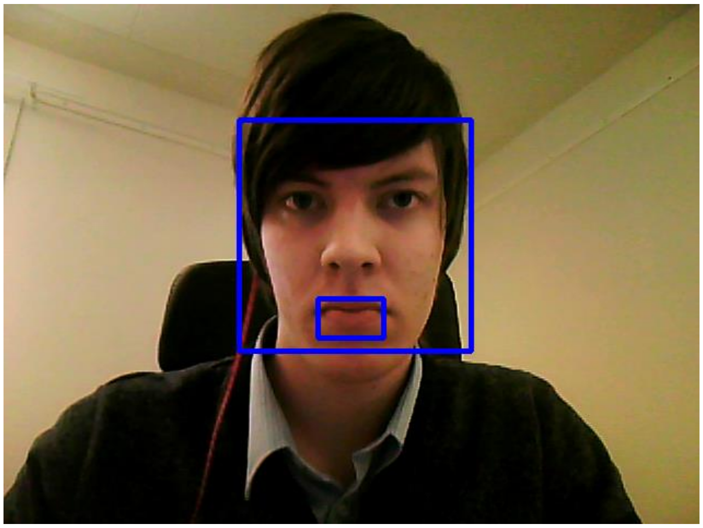


Fig. 4. Image showing the detected face and mouth ROI (located in the lower part of the face ROI).

When the face ROI and mouth ROI have been detected this part of the algorithm is done, the result is visible in Figure 4. The following step of detecting the mouth elements can now be done in the mouth ROI alone. That will also improve the speed and the actual detection rate of the features. If this ROI would not be used it would not be possible to do all this in real time. Because then all the calculations will be done on the full image and that would not be advisable.

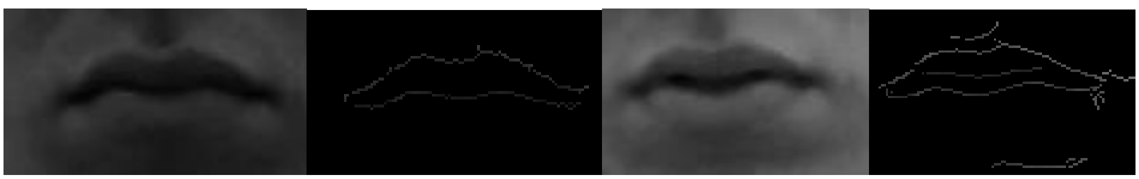
4.2.2 Detection of mouth features

The detection of the mouth features consists of a few different parts. Firstly the corners of the mouth are calculated. After that the algorithm tries to detect the upper and lower parts of the lip. This is data that can be used later to try and determine the state of the mouth, if it's opened or closed. The reason for having different methods for detecting the upper and lower lips is because some of the methods are more suited than the other. If there are perfect lighting conditions and image quality, the methods perform well on both the upper and lower parts of the lip. This may not be the case for this work. The images that will be analyzed will not always be in good lighting and good conditions. This is because the images are supposed to be captured in a car where the conditions can vary severely from time to time.

Utilizing different methods for different tasks will hopefully result in good performance for the feature detection. This approach could increase the computational time needed for each image, but not enough to make any real difference. Furthermore better detection results outweigh small improvements to the execution time, since sufficient speed should still be obtainable.

4.2.2.1 Mouth corner detection

The detection of the mouth corners is computed in a few steps. The first thing done is tracking the outline of the lip with the help of canny edge detection [43]. This will attempt tracking the edges of features in an image. When tracking the edges, a threshold value will be used to decide what to count as an edge. Using a fixed value for the threshold will make it difficult to adapt to different amounts of light. For the purpose of improving the edge detection in varying lighting conditions, the threshold will be calculated from the mean value of the image histogram. Doing this makes it more adaptable to different amounts of light than it would be using a fixed value. Figure 5(b) shows the result from the canny edge detection.

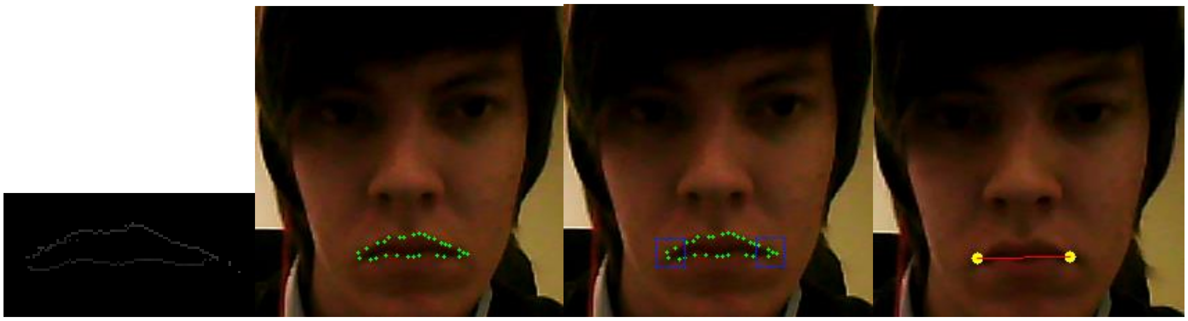


(a) (b) (c) (d)

Fig. 5. The image show the transition from grayscale to the canny edge tracked image. (a) is the original image of the mouth and (b) is the image with a detected edge. (c) is the original image and (d) is the image with tracked edges.

Because of these varying lighting conditions the tracking is often only good on the upper and middle part of the lips, as can be seen in Figure 5(b). In different cases the tracking of the edges will result in images that may not be totally optimal, as seen in Figure 5(d). There the lighting on the original image makes it hard to detect only the lips as edges so small artifacts can appear. Because of these kind of situations, doing this is best suited to detect only the corners of the mouth. Otherwise this could have possibly been used to detect at least the upper part of the lips. But because of these problems another method had to be used for that purpose.

When the edges have been found, points will be tracked on that image. The points will be tracked using a method called Shi–Tomasi corner detection [44]. What it does is that it tries to detect the most prominent corners in an image. The reason for detecting the edges of the mouth before tracking the points is because it will generate a more accurate point distribution. Since the points will only have the possibility to be placed on a thin line, they will be placed on said line and close to each other, as shown by Figure 6. Subsequently when the points have been calculated and placed on the mouth, its corners can be estimated based on those points. This is accomplished by selecting the areas of the image where the mouth corners should be located, as seen in Figure 6(c). From the selected regions all intersecting points will be detected and from them an average position will be obtained, this will be the corner of the mouth. The same thing will be done for both sides of the mouth, the result can be seen in Figure 6(d).



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

Fig. 6. (a) is the canny edge image. (b) is the points tracked from the previous. (c) show the approximate regions where the mouth corners will be detected. (d) is the corners of the mouth from the average of the points inside the blue region.

When that is done the detection of the mouth corners is complete. This approach will generate pretty accurate detection of the mouth corners. Although the results may differ a bit depending on the lighting conditions and from which direction the light is coming from.

4.2.2.2 Lower lip detection

To achieve point tracking on the lower lip the area needs to be segmented first. What that means is that the area of the lips needs to be masked. In this case the end product will be a binary image, a black and white image. The white area in this binary image will represent the detected lip area. When the area of the lip is known, it's possible from this binary image to extract points that indicates the bottom part of the lower lip as well as the upper part of the lower lip.

Segmentation of the lip is achieved with the creation of a lip map [45]-[47]. To do this the image is converted to the YCbCr color space. The Cb value is the blue-difference chroma and the Cr value is the red-difference chroma components. This can be utilized to map the area of the lips. Because the lip area will contain a stronger red component and a weaker blue component than other parts of the face would do [45]. From this it is possible to calculate the probability that the given pixel is a part of the lip area. The following equation is used to generate the lip area:

$$LipMap = Cr^2 \cdot \left(Cr^2 - \eta \cdot \frac{Cr}{Cb} \right)^2$$

$$\eta = 0.95 \cdot \left(\frac{\frac{1}{n} \sum Cr(x,y)^2}{\frac{1}{n} \sum \frac{Cr(x,y)}{Cb(x,y)}} \right)$$

This equation will generate a pixel value that will represent the possibility that the processed pixel would be part of the lip area. The higher the value is, the more likely it is that the pixel is part of the lip. These values are mapped to a grayscale image in which the lip area should be lighter than the surrounding area.

Depending on the lighting, there may be other parts of the face that could be considered a lip color. That means there is a possibility for some weird artifacts to be created when the image is converted to a black and white binary image. To try and lessen the amount of artifacts, various image processing techniques are used to try and clean up the image and create a better end result.

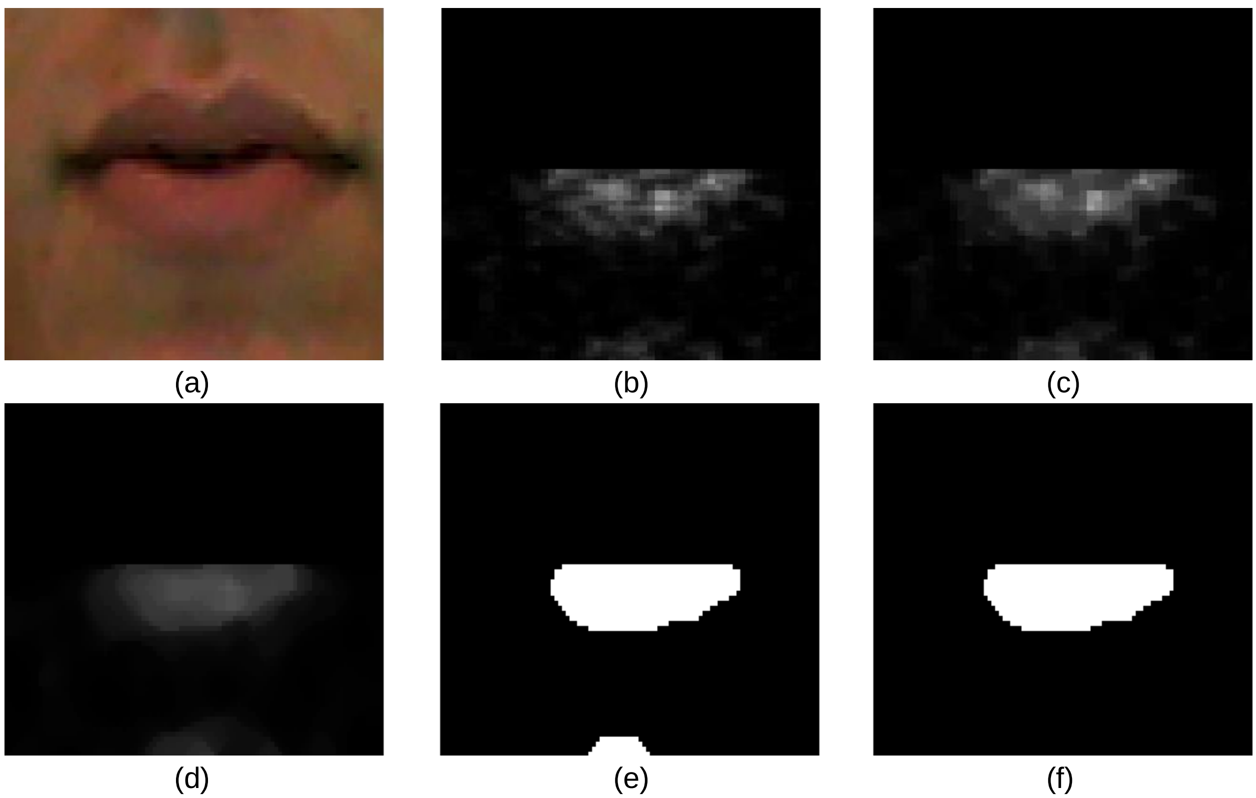


Fig. 7. Image showing the stages of the lip detection. (a) is the original image. The rest of the images only work on the lower half of the original (a) because they are supposed to only get a mask for the lower lip. (b) is the image after the lip map equation has been calculated. (c) is the image after morphology transformations. (d) is the image after it has been blurred with a median filter. (e) is the image after thresholding with Otsu. The final image (f) is the image after all the small objects, that should not be the lip, have been removed.

How the image is processed can be seen in Figure 7. After the lip map has been calculated, the image is first morphed and blurred to try and remove some of the small artifacts that exist. When that is done the image is ready to be converted to a binary image. This is accomplished by using the Otsu thresholding [48]. This may still create some smaller objects that should not be part of the lips. This leads to the final step, which to remove all but the largest single geometry from the image. The assumption is made that the largest object would be the lip area. After that the final binary image is complete. This can now be used to try and extract points that would correspond with the top and bottom of the lower lip.

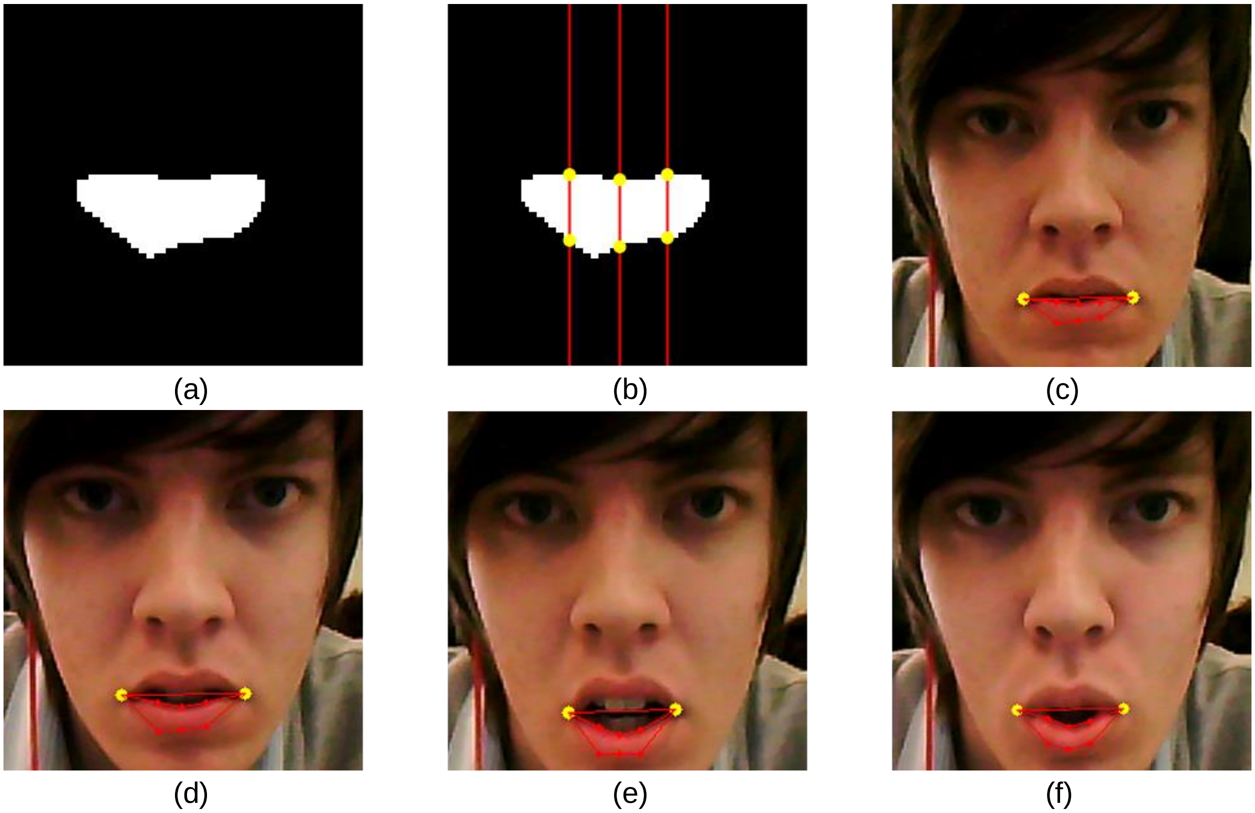


Fig. 8. (a) is the binary image that will be used to track the points on the lip. (b) show how the points will be found, simply scan from the top of the image and the points will be the lowest and the highest in which the pixel is white. Image (c) shows the final result of the lower lip tracking. Lines are drawn between the corners and the lip points for visual purposes. (d)(e)(f) shows the result in different mouth states.

When the binary image is complete the points for the top and bottom part of the lip will be extracted. This process is shown in Figure 8. Firstly the points that will be extracted are the lowest points in the binary image. This is done by scanning the image from the top and down. The goal is to find the extreme points of the white area in the binary image. These extreme points should represent the bottom position of the lower lip. The same method is also used to find the upper part of the lower lip.

When this is done the points of the upper and lower part of the lower lips should have been extracted correctly. These points can be of use to help determine the state of the mouth, if it is open or closed.

4.2.2.3 Upper lip detection

To track the upper lip a different method was used than for the lower lip. To get a point from the upper lip, a RGB image of the mouth area is split into three single channel images, one for each color. The upper lip is thus found through the color cues and variations [49]. Each channel is then thresholded to highlight the upper lip area. This area is then used to track the center of the upper lip, in order to use this in for example tracking the mouth state.

To get the value for which to threshold the image, the average brightness of the image is calculated. This average value is computed by converting the RGB image to HSV format. The brightness value for each pixel is then used when calculating the average value. The center of the upper part of the upper lip is most likely the brightest of the artifacts in that area, this makes the usage of average brightness value an appropriate threshold value. Anything below the average brightness is not the searched area, the thresholding helps remove irrelevant artifacts.

The brightness average value is then used to threshold each of the RGB color channels separately. As can be seen in Figure 9, the center of the upper lip has a distinct color compared to the rest of the area.

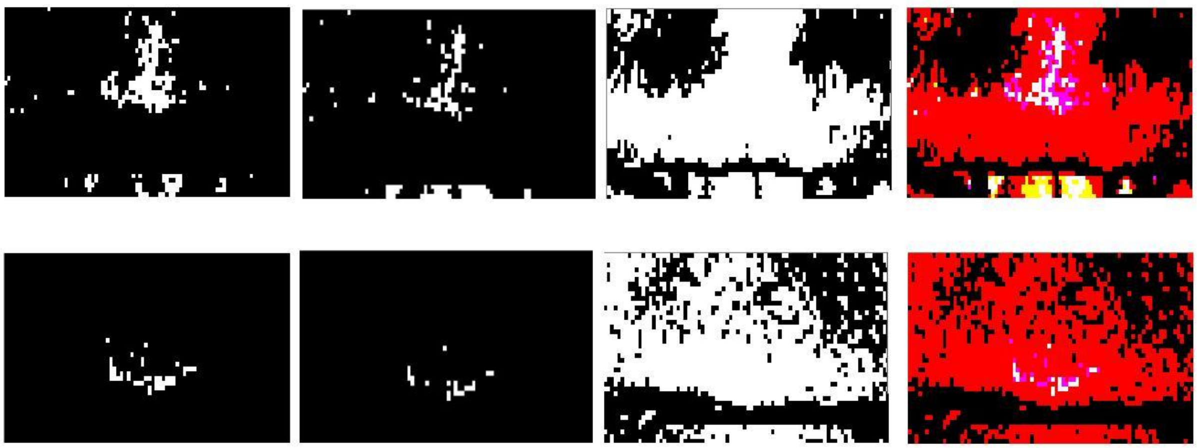


Fig. 9. In the image each separate channel is shown (from left in both rows Blue, Green, Red) after the binary thresholding. In the fourth image the result of the merge is shown. In the top row the result of using this method with facial hair is shown. The center of the upper lip is visible in the center of all the images. In the bottom row facial hair was not present.

The channels are merged back together, result can be seen in the rightmost column of Figure 9. To further increase precision (and speed up performance) another region of interest is created. This in the area where the center of the upper lip will most likely be.

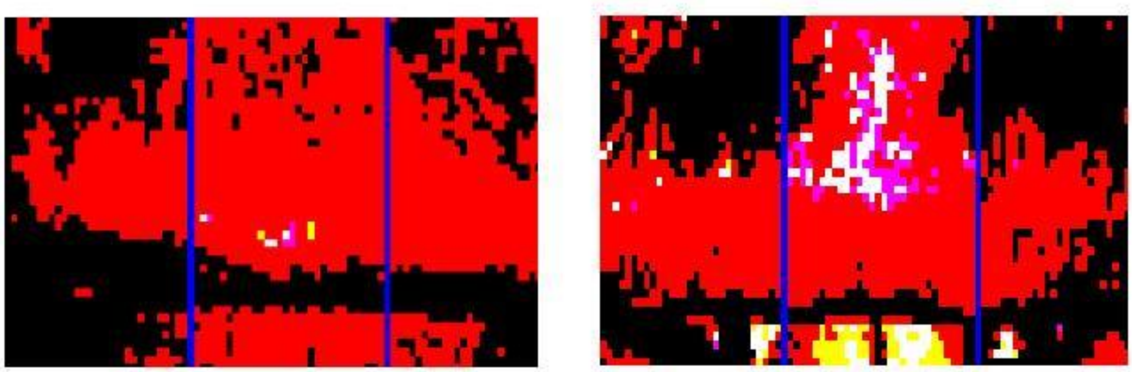


Fig. 10. Here the region of interest within the upper lip area is shown. It is within the blue lines that the analysis will be made. Here the filtering so far is also visible. The image to the right has greater black areas in the upper right and left. This is facial hair, which will be ignored due to being too dark and outside of the ROI.

The pixel colors within the center of the upper lip are more mixed and will therefore give a good indicator to where a reference point should be, as can be seen in Figure 10. By then checking all the pixels fulfilling the requirement of non-black color, a set of pixels can be produced. This is done in order to fix the issues created by the facial hair. By not counting the black pixels, the facial hair and mouth opening is ignored. The set of pixels is then used to create an average point.

All pixels below this average point are then to be checked for a specific pattern, the point to search below is visible in Figure 11(b) and (e). The pattern to search for is when a non-black pixel have a black pixel neighboring below, which signifies the edge of the mouth opening. This produces a set of pixels located just above the mouth opening on the upper lip. The average value of this set is then used to describe the upper lip center point.

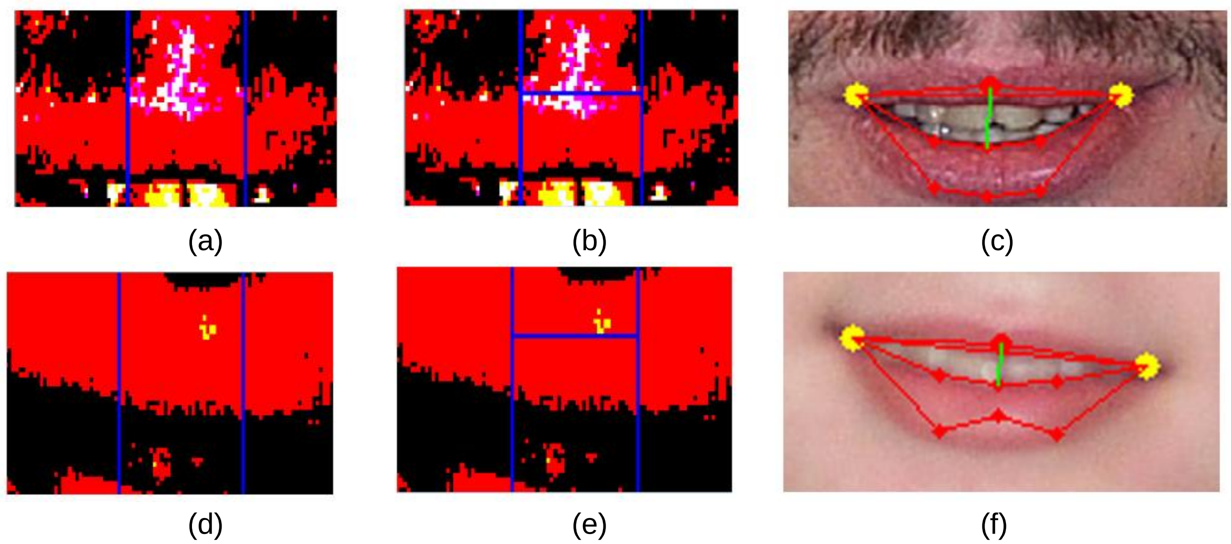


Fig.11. From left to right (both rows), the first image shows the first stage where the average non-black pixels are used to create a border. All values below this border (as seen in the second image) are then scanned for the pattern where a colored pixel with a black pixel bordering below is accepted. The rightmost pictures show the result of the full tracking. In the top image the teeth are due to the lighting conditions visible, however they are ignored since they do not fulfill the requirements of the pattern.

There are some issues with this method. It is based on that the corner detection is correct. If the corner detection is not correct, the top half of the mouth cannot be created in a correct and accurate way. The method also requires even illumination. If the illumination in the upper lip area is uneven or heavily focused on an area close to the center, the tracking will be inaccurate.

4.2.2.4 Mouth state detection

In this section two methods of finding the mouth state will be presented. To know the state of the mouth will be important for the breath identification.

4.2.2.4.1 Lower lip and midpoint method

The detection of the possible states the mouth can have, is something that is of great help when trying to make a decision if a real breath was made. These states are simply if the mouth is closed or open. When doing the breath identification it would be good to be able to tell if the mouth is open or not. Because often when a person blows out air, the mouth will most likely be open or at least partially open.

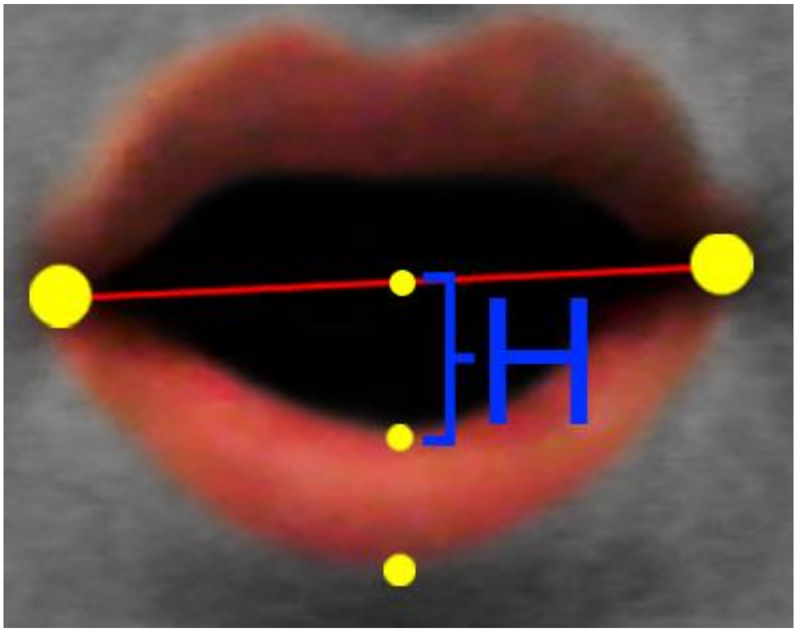


Fig.12. The image shows how the state is checked. The value H is the length between the upper point on the lower lip and the midpoint of the mouth, taken from the average value of the two mouth corners.

The detection of the mouth state is done in a simple manner as can be seen in Figure 12. The distance between the upper point on the lower lip and the midpoint of the mouth, taken from the average of the two mouth corners, is used to check how much the mouth is opened. The calculation to check if the mouth are open can be seen below.

$$OpenRatio = \frac{UpperLowerLip(x,y)}{MidMouth(x,y)}$$

$$MouthState = \begin{cases} Open & if(OpenRatio) > 1.30 \\ SlightlyOpen & elseif(OpenRatio) > 1.17 \\ Closed & else \end{cases}$$

The closer the two points are to each other the closer the open ratio will be to one. The further away they are the bigger the number will become. Therefore, when the ratio is larger than a certain value the mouth state will change. The values have been obtained by testing and trying to figure out what works best. The reason the upper lower lip and the middle of the mouth is used is because they are the most stable points. Nevertheless doing this may sometimes, with certain facial expressions, yield a wrong result. But in this case, when the check should be done when the person is blowing air into the breathalyzer, it should do the job fairly well.

4.2.2.4.1 Lower lip upper point and upper lip point method

This is a second method which utilizes the upper lip when deciding the mouth state. This works by using the upper point of the lower lip and the upper lip point. These two points are then used in a formula to get a value, which will be used to indicate if the mouth is open or closed.

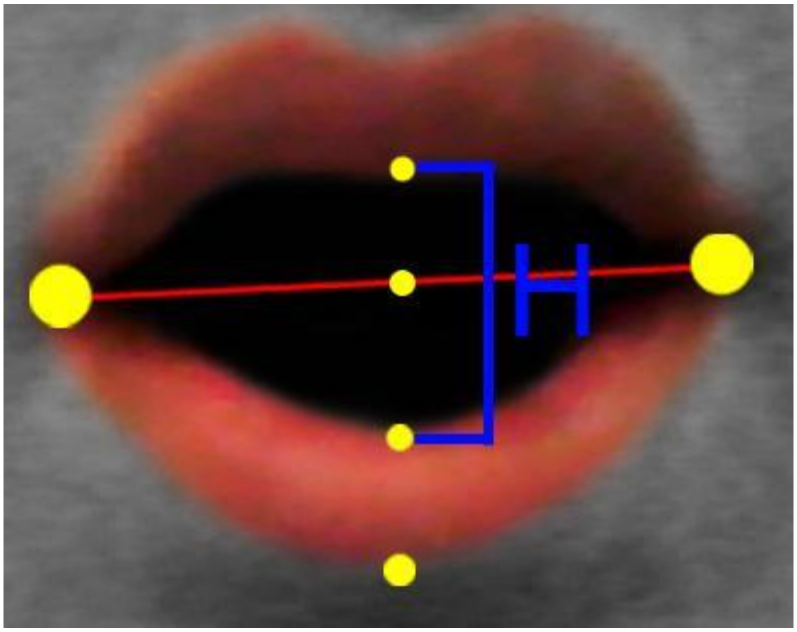


Fig. 13. This image show the two points used to determine the mouth state. The distance H is the distance between the top lip center and the bottom lip upper center.

The closer the value gets to one the more closed it is. The y-values of the two points are the values used in the formula, the points used are visible in Figure 13. The formula used is displayed below.

$$OpenRatio = \frac{UpperLipCentre(y)}{LowerLipUpperCentre(y)}$$

$$MouthState = \begin{cases} Open & if(OpenRatio) < 0.965 \\ Closed & else \end{cases}$$

The closer the ratio is to one the more likely it is that the mouth is closed. The border value for open and closed was found through testing. This method works well if the tracking is true. Also it considers the entire mouth area which should cover cases where the mouth top half is opened.

4.3 Breath identification and Manipulation prevention

Other than just being able to detect if a person has actually made a real breath into the alcolock, it's also important to note that it should reject if a person is trying to manipulate. Therefore to make a decision if it's manipulation or not, there should be some points that needs to be fulfilled for a breath to be considered real.

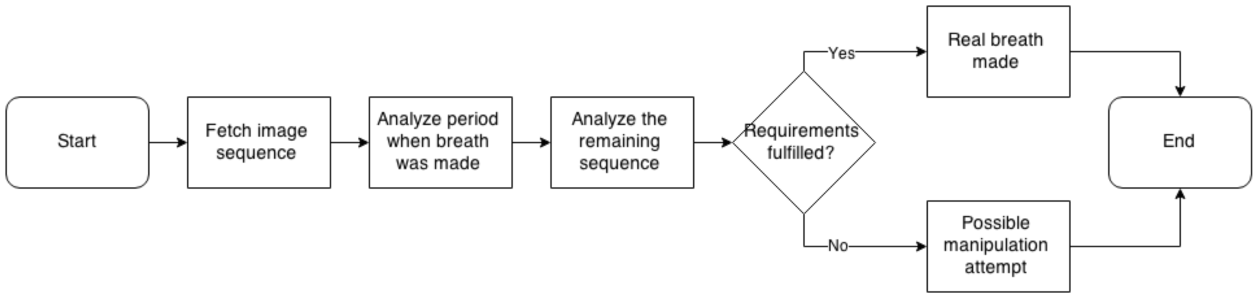


Fig. 14. Image describing the sequence for the breath identification. First the image sequence is fetched, split into two parts and analyzed. One part is from the point the breath was made to a second or two before it was registered. The restoring images forms the other sequence, which represents the frames right before the breath was made. If all the requirements are fulfilled then it will be regarded as a real breath.

For the analysis to produce a good result, it will not be enough to just consider the frame at the moment where the supposed breath was made. Still the part where the breath was made is important, it's vital to investigate if nothing is interfering or anything strange is happening at the moment of the breath. Nevertheless to get the full picture and a more accurate reasoning on the decision, more frames needs to be analyzed, see Figure 14. By doing this it's possible to observe if the breath sequence seems to be in order and nothing seems out of place. This also helps compensating for possible feature extraction errors that can occur for a particular image. The feature extraction is not always free from detection errors depending on various factors. This helps reduce the amount of real breaths that could be decided as false by the system because of faulty feature extraction.

For a breath to be judged as a real one there are a few points that needs to be fulfilled. If they are, then the breath will be seen as a real one. If any of these points are left out, then the breath will be labeled false by the system. If it's deemed false, it will not matter if the result from the alcolock says the breath was clean of alcohol. Another test will be required to be done. This makes the decision sequence difficult to apply. It needs to be strict enough that most of the possible manipulation attempts are blocked. Furthermore if it's too strict real breaths may be considered faulty ones by the system. This will hurt the user experience and may cause confusion for the user. It is a thin line between the two.

The identification process will consist of a few parts that will be checked, from the point where the breath was registered by the alcolock to a few seconds before it was made. If it's possible to iterate through the sequence and all the points are valid in the end, it will be decided as a real breath.

For example if the user is using some tool to blow through to try and fool the alcolock, it will probably not work. Because the system needs to detect a mouth at the point the breath was made. If it can't find any mouths when analyzing the sequence, it will simply be labeled as a false breath. If another device would be used to blow air through the alcolock, that is not a human, it would not pass. Because it needs to detect a face and mouth somewhere in the sequence. A face and a mouth needs to be found in the sequence for it to be able to give a positive result at all. If something at any point would block the mouth or face so a detection can't be made it will not be able to pass the detection. Therefore it’s more about detecting if anything is interfering or a detection can’t be made, rather than just be able to tell if a breath was made.

The main thing that the identification will check for is an open or slightly open mouth at the time when the breath was received by the device. A set amount of images in the sequence must fulfill this criteria otherwise the breath attempt will be rejected. In this case, if the mouth would have something blocking, even for only a short time, it probably would not pass the test. Because it needs to detect the mouth in every frame, minus a few to counter the natural errors that might occur in the feature extraction.

The rest of the frames will also be checked. The goal is to detect a mouth in most of these frames. Furthermore, a set amount of these frames must also include a full view of the face. This is to determine that there was in fact a person in the frame so that the mouth detection itself was not a trick.

If all these criteria are fulfilled the result will be that it was a real breath that was made. Otherwise if any of the criteria are not fulfilled, the result will be that it was not a real breath but an attempt at manipulation.

5. Results

To judge the performance of the system a few tests have been made for the purpose of testing different aspects of the system. Testing has been made on the feature detection to get an understanding on the correctness of the feature extraction for the mouth corners, upper and lower lip. Furthermore the execution time of the system have also undergone testing. Testing has also been made to check if the system can detect when someone is trying to manipulate. Finally, some tests have been made to see if the system can also detect when a real breath is made.

5.1 Feature detection

To conclude the accuracy of the algorithms facial feature detection, it was tested against several images from the Caltech face database [50]. The images from this data set have different lighting and background conditions. The images are taken from a group of around 20 people and there are several images of each person, and a few different facial expressions. Some of the images were omitted if they were too low quality or if the lighting conditions were far too dark. Furthermore some images where the person was too far from the camera were also not included.

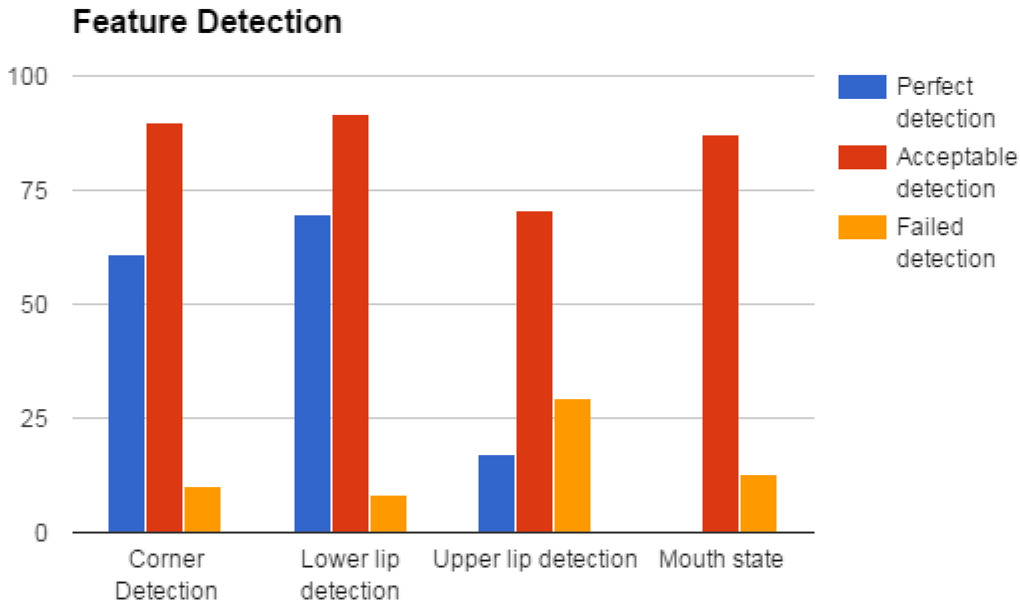


Chart. 1. Chart visualizing the detection rates of the feature extraction from the values in Table 1.

	Perfect detection	Acceptable detection	Failed detection
Corner Detection	61%	89.8%	10.2%
Lower lip detection	69.9%	91.8%	8.2%
Upper lip detection	17%	70.8%	29.2%
Mouth state	-	87.3%	12.7%

Table. 1. Showing the results of the feature tracking on the Caltech face database.

The results can be seen in Table 1 above. For this test 392 images were analyzed from the data set. Acceptable detection represents detections good enough to generate the correct result. The perfect detection are the percent of images that more or less made a perfect detection of the specific feature.

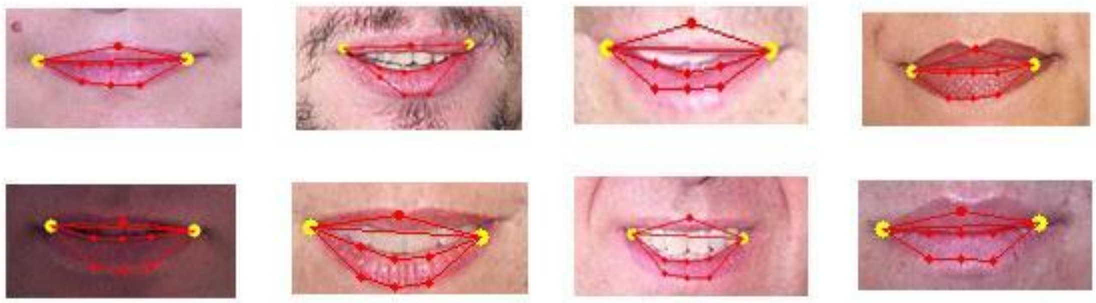


Fig. 15. The image shows the result on a few of the images. The yellow dot is the mouth corner. The red dots mark the upper and lower part of the lower lip as well as the upper lip. Red lines are drawn between the points for visual purposes.

Both corner detection and lower lip detection achieve a detection rate of around 90 %, some of the results can be seen in Figure 15. The images that the corner detection had the most problem with was images where a person was smiling or a beard was present. This could cause the corner points to be placed somewhat off target. But overall the corner detection performed well enough. Since a lot of the errors were due to people smiling it should not be of much concern. Because when a person breathes into an alcolock there is a high probability that a smile would not be present. The bigger issue in this case is that a beard in some cases could cause some problem for the detection.

The lower lip detection performs well overall and also have a high amount of images that get perfect detection. The cases where the lip detection had most problem was when the lip were exposed directly to a strong light source. If the lip would appear to specular the detection could have problems properly differentiate the lips from the skin.

The upper lip detection did not perform as well as the lower lip. The method used for the upper lip had decent accuracy but certain issues are still present. If the illumination in the upper lip area is uneven the tracking may be inaccurate. This is due to the method's way of finding the center of the upper part of the upper lip. The method works as long as the brightness and illumination is not uneven. It also has issues with people smiling. If a person is smiling the teeth will be visible in the upper lip area. This may cause the mean values to be skewed, due to picking up the high color values of the teeth, however due to the use of the search of neighboring black pixels this problem can be avoided in most cases. The upper lip tracking is also based off of the corner detection. Thus if corner detection is inaccurate so may the upper lip tracking be, however this is not necessarily always the case but the risk is still there. On the test set it only got 70% acceptable (or better), compared to the lower lip which got over 90%. Examples of acceptable (or better) results of the upper lip tracking can be seen in Figure.16.

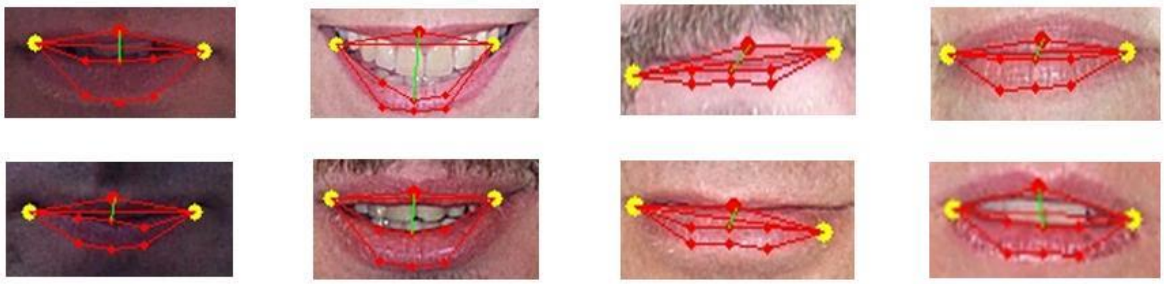


Fig. 16. Here the result of the full tracking is visible. The images are from different people and under different light conditions.

To decide mouth state the method involving the lower lip and midpoint was used. Due to the upper lip tracking being only 70% accurate, the method using upper and lower lip points was tested but in the end not used. The upper lip method did however produce good results, but would fail more than the other method. In some cases the lower lip and midpoint method would not detect that the mouth was open, because only the top half would be opened. In these cases the lower lip upper point and upper lip point method performed better. But overall the lower lip and midpoint performed better since the tracking was better.

Fig. 17. Show a few images where the mouth state detection would generate the wrong answer because the upper lip is more open than the lower lip compared to the middle line of the mouth.

The mouth state detection performed decently, although a lot of the images in this data set depicted people with closed mouths it was able to detect if a mouth was open as well. Still in some cases it would fail to detect if the mouth was open. For instance, if the upper lip would be more open than the lower lip, in relation to the middle of the mouth as seen in Figure 17. The reason for this is the way the mouth state is determined. It uses the upper point on the lower lip in relation to the middle of the mouth to assess if the mouth is open or not. Another thing that could cause the mouth state detection to generate the wrong result is incorrectly detected mouth corners. The middle point that is used to decide if the mouth is open is generated from the mouth corners. In case of a corner miscalculation there is a possibility that the same would happen to the mouth state.

Despite some of these errors, the feature detection works well enough to get a correct detection in most of the images. It showed that images where the person was smiling was the most difficult to extract features from correctly. Fortunately this should not cause too many problems, for a person would rarely smile when doing an alcohol test in a correct way.

5.2 Processing time

Since the proposed method should run in real time, processing time must be kept as low as possible. Enough images must be processed each second to enable accurate decision making on the breath identification analysis. If there are not enough images then there is a possibility that manipulation attempts could pass. There is also a chance that a real breath would not be considered one, because the low amount of images were not enough to compensate for the possible feature extraction errors.

The test of the method was done on a laptop computer with a 2.4 GHz dual core processor. The size of the images were (640*480). It's hard to present a result that would represent all the situations that can occur. Depending on the camera used the processing time could change to some degree, due to resolution differences. Also because the time it takes to fetch the original image, before any processing has been made, could vary. Another thing that can affect the processing speed is the light. Depending on the conditions and the camera it could be possible to change the speed. The background is also something that in some cases could affect the capture speed of the camera negatively.

Therefore, it may be more important to pay attention to the execution time of the feature extraction. Since the other parts of the execution could vary. Although it could vary, it should not be too big of a variation to affect the results in this test. Another thing to note is that the amount of processing power used for this test might not be available in a real environment. This means that the results presented here could be slightly better than they would be if implemented into a vehicle.

	Face ROI	Mouth ROI	Mouth corners	lower lip	Upper Lip	Other
Time (ms)	11.6	11.5	2.13	3.18	1.71	4.48

Table. 2. The table shows the average execution time for the feature detection part of the software. The total average time for the execution is 34.6 ms.

	Feature detect	Other operations	Total time
Time(ms)	34.6	19.9	54.5

Table. 3. The table shows the total time it take to get a finished frame. Feature detection is the calculations described in Table 2, and Other are the restoring operations needed to be done.

The average execution time for the whole process of one image is 54.5 ms, as shown by Table 3 above. For the total calculation, 34.6 ms are spent detecting facial features. How the execution time is split for the feature detection can be seen in Table 2 as well as Chart 2. The majority of the time is spent detecting the regions for the mouth and the face, this amounts to 23.1 ms of the total 34.6 ms. The calculation of the corner points and the upper and lower lips amount to a total of 7.02 ms. When all that is calculated it still remains 4.48 ms unaccounted for. These are all the remaining calculations done inside the feature detection that is not directly calculations of the feature points.

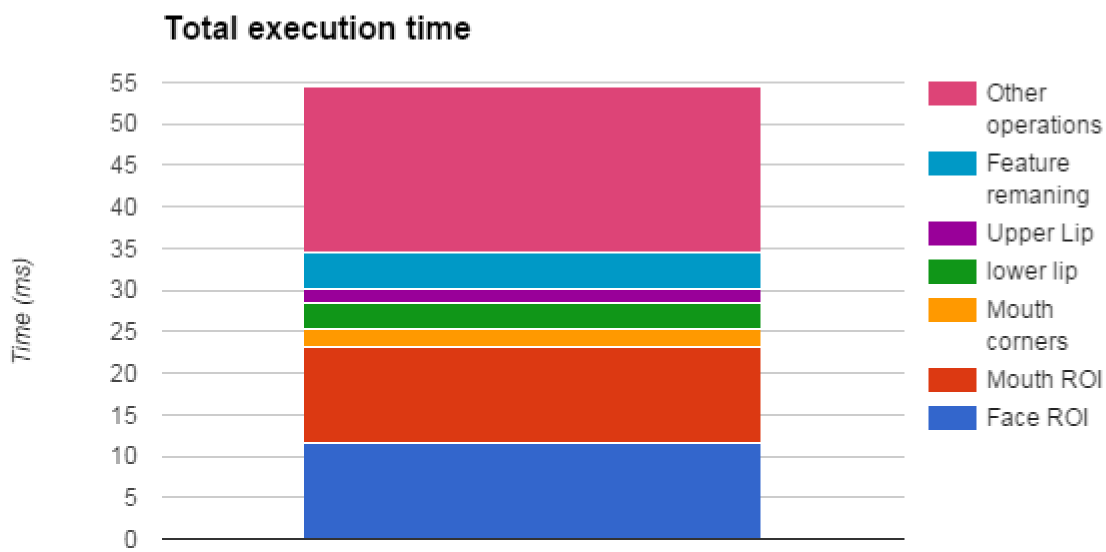


Chart. 2. Chart to give a visual example of the average execution time. All values are part of the feature detection except for the other operations value.

When the feature detection is done another 19.9 ms are still unaccounted for in the total calculation, as seen in Table 3. These are a few operations made before and after the feature detection is made. The majority of that time is spent fetching the image from the video stream. This will result in an average frame rate of 18.3 frames per second. This is more than enough to make an accurate identification of the breath. Especially considering the amount of room that exist for improvements.

5.3 Breath identification

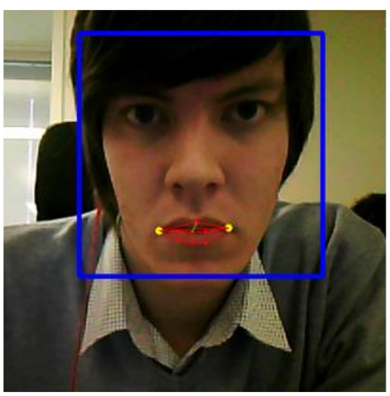
To evaluate if the system could detect a manipulation attempt or a real breath, testing was done in an experimental environment. A laptop webcam acted as the image capture source. It was imagined that the person were blowing into an alcolock device positioned in front of the camera so the whole face would be visible. To simulate the signal from the alcolock that a breath was made, a simple button press on the keyboard acted as the signal. When that button was pressed the system was told that a breath was made and started to analyze the captured images.

5.3.1 Manipulation attempt

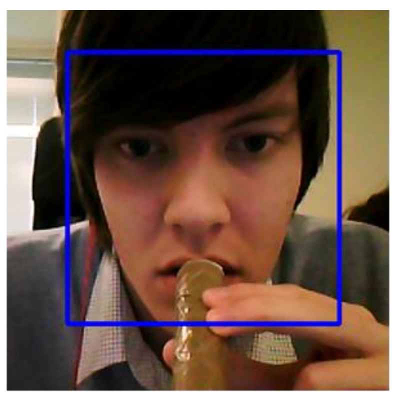
For the test of the manipulation detection, a breath tried to be made when using a form of manipulation. The system should respond with a false breath detection in the test for it to be deemed successful. The purpose of this test is to investigate if the proposed solution actually is able to reject the manipulation attempt. By doing that strengthen the claim that this is a method that has potential to work in a real environment.

Fig. 18. The image shows a carbon filter that could be used to potentially be able to fool the breath test of an alcolock device.

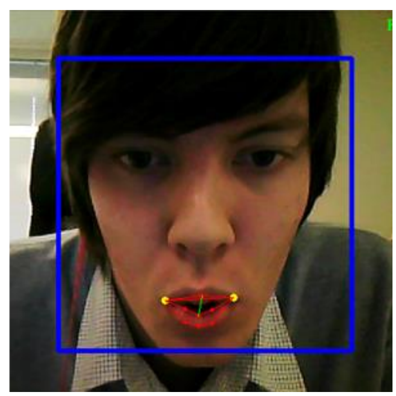
The test was done together with a carbon filter, can be seen in Figure 18 above. This is a potential method that can be used to try to fool an alcolock system to generate a negative result that would enable the user to start the car. The filter is approximately 10 cm long and 2.5 cm wide. The idea is that the when trying to use this filter the system will be able to detect that a manipulation attempt could be happening. Another way to view it is that it cannot detect a real breath, therefore the attempt is seen as a false breath.



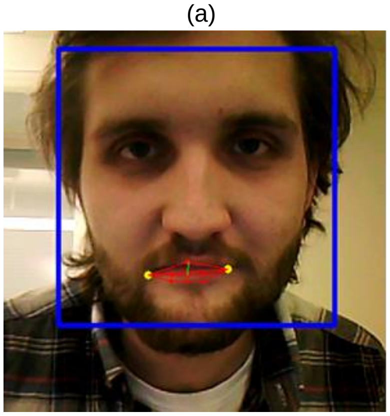
(a)



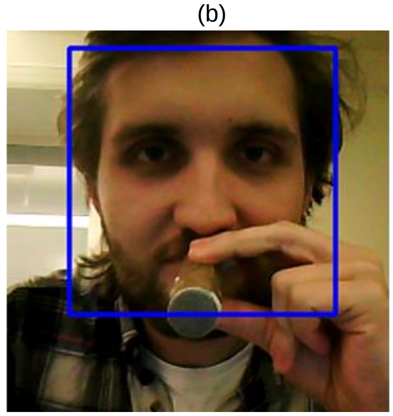
(b)



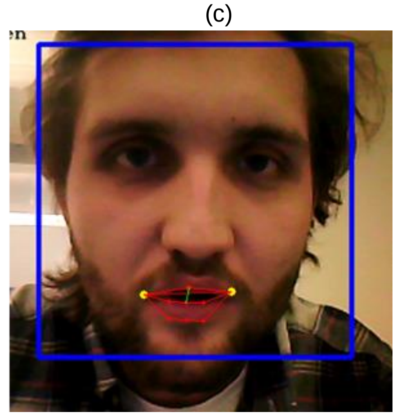
(c)



(d)



(e)



(f)

Fig. 19. The images show how it looks when trying to blow into the possible alcolock. (a) and (d) shows the state before the breath attempt is made. The mouth is clearly visible and features are tracked. The mouth state is closed. (b) and (e) shows when a breath is trying to be made with the help of the carbon filter. The mouth tracking is not working because it is blocked. This would result in the system classifying it as a false breath. (c) and (f) shows how it could look when actually making a real breath. The mouth is tracked correctly with all its features and the mouth state is open. This would result in a true breath.

The test was done by blowing air through the filter into the pretended alcolock device, this can be seen in Figure 19. As seen in Figure 19(a) and (e). When the carbon filter is placed in front of the mouth the tracking is not possible. Sometimes the tracking might pick up the mouth when the filter is present but that only happens for a few frames. To get a true breath from the system all images, minus possible misses in the feature detection, must have a visible mouth tracked. Furthermore, when the breath is made, the mouth must be deemed open by the system around the time when the breath was made, as seen in Figure 19(c) and (f).

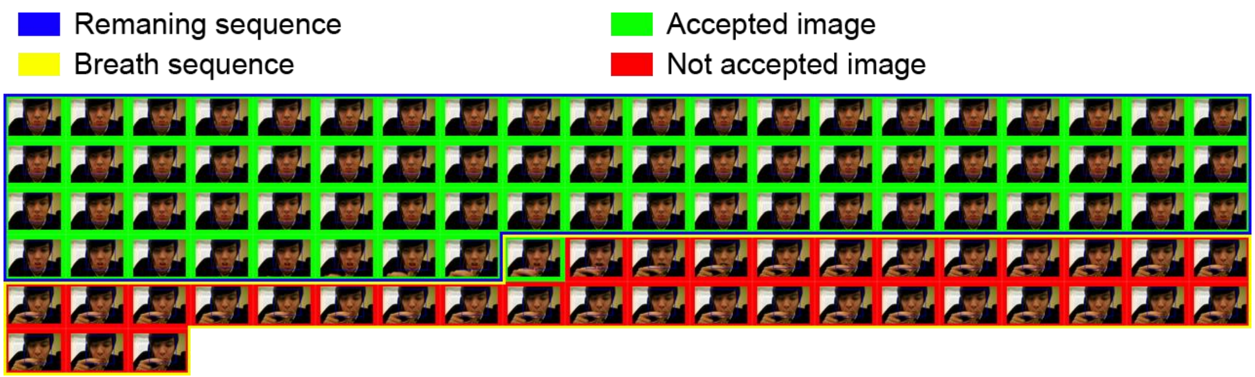


Fig. 20. The image shows a sequence of frames, when blowing through a carbon filter, analyzed. All but one frame in the breath sequence were rejected. The carbon filter was not brought up to the mouth until the breath sequence starts, that is why the frames before are accepted. The result from this sequence would be a false breath.

An example of an image sequence when blowing through the filter can be seen in Figure 20. The figure shows that before the breath sequence the images were accepted and all was fine. But when the carbon filter was brought up to the mouth in the breath sequence the mouth would be blocked causing the frames to not be accepted. When this many frames get rejected it will result in a false breath and not be accepted.

This was tested several times. There was no case where it was able to get a real breath indication from the system. Because the dimensions of the filter are pretty big it was easy for the system to recognize that it was not a real breath attempt. Among the frames that were analyzed for each attempt, the detection rate did not fulfill the requirements to generate an answer indicating a real breath. It would be enough if the feature extraction failed to extract the mouth features for just a few images, for the result to be decided as false.

5.3.2 Real breath attempt

For the breath to have the possibility to be labeled a true breath, all the requirements must be fulfilled in the breath identification. Basically in the images around the time where the breath was made a mouth must be detected, as shown by Figure 21. Furthermore, most of these images must have an open mouth detected. The rest of the images must have a positive detection of a face and mouth, in any state and in most of the images, for it to give a result indicating a real breath. If this is not the case the breath would be deemed false.

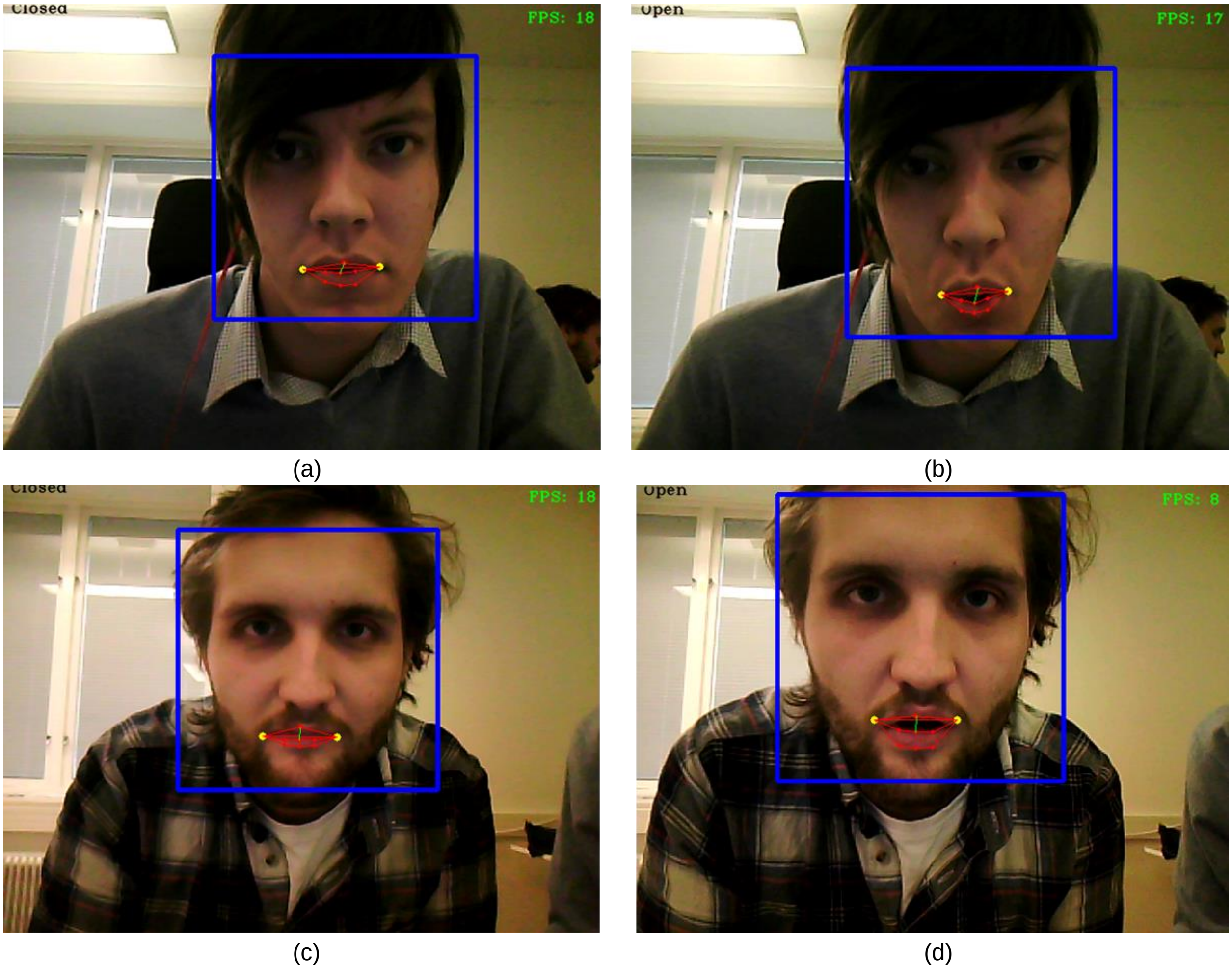


Fig. 21. The image shows images that could be in the beginning and the end of the image sequence. (a) and (c) shows how the first image could look like in the sequence, right before the breath attempt starts. The features of the mouth are correctly tracked and the mouth state is closed. (b) and (d) shows how the last image could look like, at the moment the breath was registered. It fully detects all features and determines that the mouth is open. If there would be nothing out of the ordinary with the rest of the images, this would yield a true breath.

Figure 22 shows a visual representation of a possible image sequence that would result in a true breath. The original sequence is divided into two separate ones. One will represent the moment when the breath was made, and the remaining the moment right before the breath. The image shows that during the breath sequence some of the images did not get accepted by the identification. The reason for this was that the mouth detection failed for those particular frames. However for the result to be a false breath, more frames would need to be rejected. As long as the number of accepted frames are above the threshold it will be classified as a real breath.

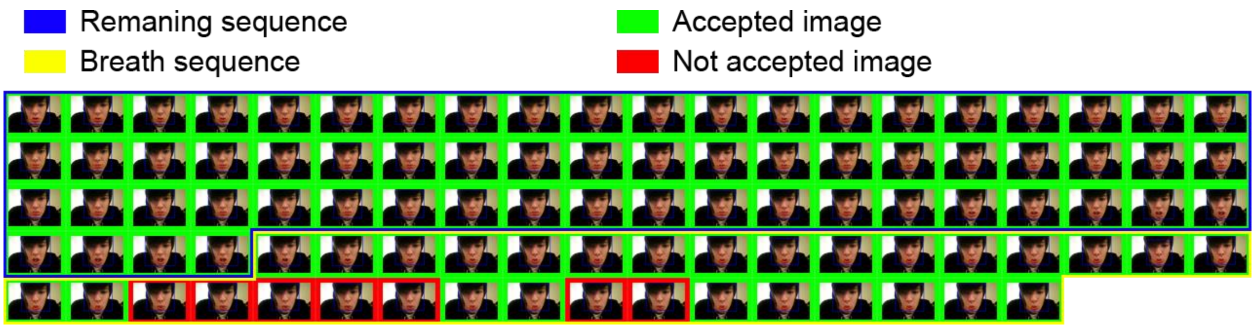
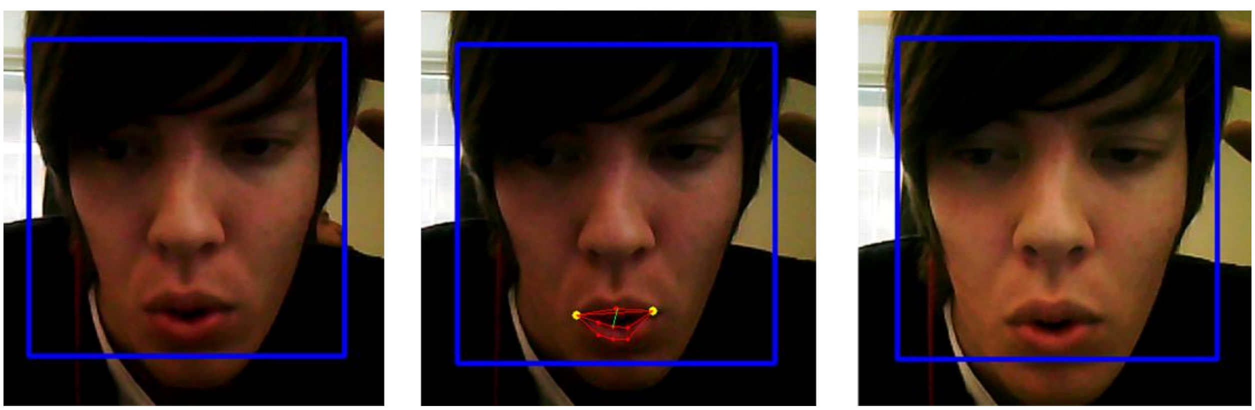


Fig. 22. The image shows an image sequence that was judged as a real breath. Yellow and blue represents the two separated parts of the sequence. Yellow would represent the moment the breath was made and blue the time right before the breath was made. Green indicates that the image was marked as ok by the identification, and red the opposite. Enough images needs to be marked ok or the result will be a false breath.

The testing was also done experimentally by blowing into an imagined alcolock, to get good testing of the breath identification. As long as the mouth and face are detectable at all times and the mouth is open when the breath is made the result will be positive. However in some cases, depending on the shape of the mouth when the breath is made, there may be some problems to detect the mouth ROI. This can result in a false negative in other words generate a false breath when it really should not.

This was tested a few times to get an understanding how this part of the system performs. It will get a correct detection most of the times as long as the features are being detected correctly.



(a) (b) (c)

Fig. 23. The image shows that the system sometimes may have problem extracting the facial features. (b) show when the features are extracted correctly. But in (a) and (c) the feature extraction fails because the mouth could not be detected. If this would happen to many times in the images for the breath detection sequence it would result in a false negative result, meaning that a real breath was believed to be a false one.

However sometimes the feature extraction might fail depending on the facial expression, lighting conditions or some other condition. If too many images in the sequence fail to get accurate feature detection then the breath will be detected as false, even though it was not. An example of a failed feature extraction can be seen in Figure 23. This is something that might happen when the mouth area is not detected. If this would be the case in too many images analyzed in the breath identification, it would yield a result saying the breath was false. It would be under the assumption that something was blocking the mouth and that is why it could not be detected. Another issue is if the user appears quickly in front of the camera and makes the breath quickly. This means that the breath might actually be true, but the mouth has not been in the sequence long enough thus labeling it as false breath.

The initial testing shows that the current system is more accurate than it is inaccurate. It will only miss the detections in some cases when the mouth or the face cannot be detected correctly. It could also fail if the mouth state would be calculated incorrectly.

6. Analysis, Evaluation and Discussion

In this section the result and methods used will be discussed. Here the different difficulties will be brought up in order to see where further progress and research needs to be made.

6.1 Method Analysis and Discussion

In this section the methods described and used will be analyzed and evaluated to determine if they were beneficial or not. They will also be looked at to decide if they are the proper methods to fulfill the goals and solve the problems this work set out to solve.

The usage of OpenCV greatly helped with this work. It provided all the tools necessary to implement the detections needed. Also since performance is a bit of an issue in a work like this, it is important to note that OpenCV performed well and didn't steal too much performance away from the methods used to detect the different mouth features. However one of the drawbacks with OpenCV is the lighting problem. The nature of Haar-like features make certain lighting conditions impossible for OpenCV to work with.

By using a face ROI to then find the mouth in the face ROIs lower region, proved to work. This helped by not only greatly increasing the performance but also reduce the number of false positives. Thus fulfilling both the criteria of time and accuracy.

The mouth corner and lower lip tracking are very important parts. The methods used provided good results, and the methods worked well to fulfill the goals. The upper lip is going to be important if this work is to be further developed, in its current state it's not accurate enough to be used for the mouth state detection.

The detection methods used for the mouth features suffered from the same issues that OpenCV does, namely the lightning and illumination issue. However the mouth feature detection also has their own problems. The corner detection is usually acceptable or better. In some cases facial hair proved to disturb the tracking, but usually not beyond acceptable parameters. The upper lip method dealt with the facial hair problems in a good way. The lower lip however outperforms the method used for the upper lip regarding accuracy.

The tracking is the method that was used to solve another problem, the mouth state detection. The mouth state in turn will be used to decide if the mouth is open or not, an important step to determine a true breath. The mouth state is thus highly dependent on accurate tracking.

Meaning that if the tracking is unpredictable or unreliable the result of the mouth state detection will be as well.

The identification process utilizes a sequence, if the mouth is open in most of the pictures the breath is considered true. However this method suffers from some drawbacks. If the person making the test were to quickly appear in the frame and make the breath, even if it's a true breath it may not be accepted since the mouth was present in the majority of the pictures from the buffered sequence. Also if there is tracking issues there might be some false positives or negatives.

Overall the most common weakness of the methods used is the lighting and illumination. In too dark, light or unevenly lit environments the methods used will behave in a bad way. For the lower lip and corner tracking there is some issues with facial hair. The upper lip has issues with accuracy at times, which causes it to be not reliable enough.

6.2 Result Analysis and Discussion

This section will discuss the results and evaluate them to see if the problems defined at the start were fulfilled. Here the consequences of the result will be brought up and an evaluation and analysis of the results will be made.

The result is more of a proof of concept than an actual product. This work shows that it's possible to use image processing in order to determine a true breath. However there are still some issues that are to be overcome for this to work properly in the varying conditions that it may be applied in. There are questions and ideas that have come up during this work and will be important for future development.

Good tracking is one of the cornerstones of a properly working true breath detector. The tracking of mouth corners and lower lip achieved good performance and could therefore be used for the mouth state detection. However the performance of the upper lip was not as good and therefore it was not used in the mouth state detection. It was however used in testing and proved that it could perform well if only the tracking is made more stable. The upper lip is an area for future work improvements. The reason for setting a high standard on the performance of the tracking is because if it fails all else will be unreliable. Having an unreliable system being used in a safety purpose can result in dire consequences. Even though this work is more of a proof of concept, it is important to keep the safety concern in mind.

The result of the mouth state detection is very dependent on the tracking. The results of mouth state detection in general were good enough to satisfy the question whether or not the mouth was open. This is an important step for the detection of true breath to work. The detection right now does not use the upper lip tracking due to not being stable enough. This means that if a person opens their mouth using mostly the upper lip that may not count as the mouth being open. Although this might be a little too strict, the argument that is made that this device should prioritize safety and stable analysis. Also if the face and mouth is really close to the camera the extraction of the mouth state can fluctuate between open and closed when the mouth actually is closed. But mostly the state of the mouth will be extracted correctly.

The execution time of the system in the experimental environment produced more than enough frames per second needed to make an informed decision on the breath. The more frames the system can analyze the better. Although in a real environment the amount of processing power used for the tests may not be available. Because of this it is important that the system is optimized to calculate as many frames as possible per second. The system still receives pretty good speed at the moment even though it can still be optimized more. For example at the moment multiprocessing is not used, this is something that could help improve the performance. As noted in the results on the execution time a lot of time is spent fetching the image from the camera. If this could be reduced, speed could possibly be gained. It accounted for almost a third of the execution time for each frame. Nevertheless the achieved speed in the test environment was not bad and with room for even more improvement it should be possible to use it on an inferior system as well.

The testing on the breath identification showed that it could successfully reject attempts when the carbon filter was used. Although this is only one method it still shows that the system is capable to reject the attempt of manipulation. This was one of the main points, that it would be able to reject a manipulation attempt. However if the system would not accept a real breath, it would not serve its intended purpose. The testing done for detecting a real breath demonstrated that the system is able to do so pretty well. As long as the tracking is working as it should for most of the images in the detection sequence and the mouth state is calculated correctly. Nonetheless it may not detect a real breath at all times but manages it most of the time. Depending on the shape of the mouth when blowing air the method used for detecting the mouth ROI may fail. How the mouth is shaped when blowing air might also differ from person to person. Because of this that is an area that might need some improvement in the future. Even so, this falls within the boundaries of the project goals. To investigate if it works to use facial detection for the purpose of identifying a true breath. Although improvements to the functionality can be made to allow an even more accurate detection, the system serves its intent to demonstrate that it is feasible.

The issue of a person appearing too quickly in the frame can be solved by camera placement also in order to make a true breath the quickly appearing issue is not something that should be a problem if the user is operating the device properly. In this case being too strict is in line with the safety aspect, it’s better to discard a true breath and allow for a retest than to allow the possibility of a false breath being accepted.

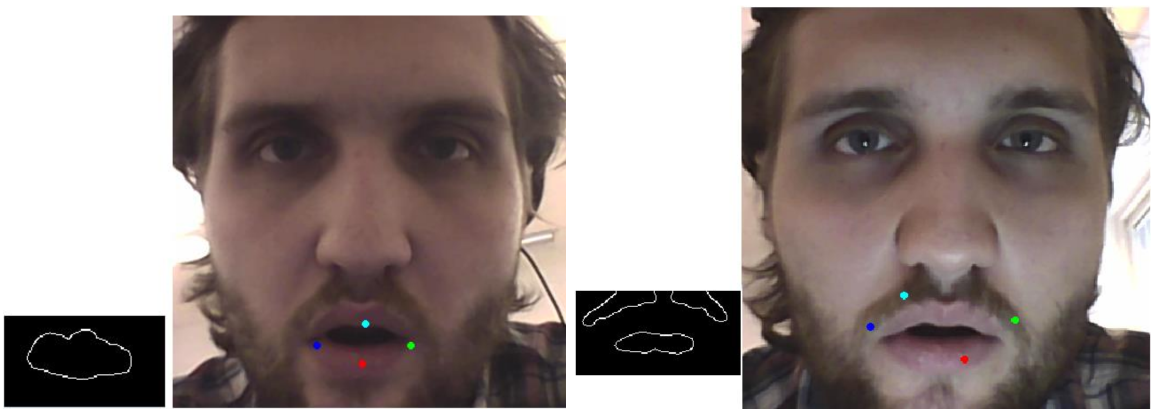
Earlier in this report the mention of manipulation techniques was made. This work will primarily only work for deciding if there’s any tampering being done by a human to simulate a true breath through some sort of device. Either blowing through some sort of filter or having a device capable of blowing air. The issue of having someone doing the test instead of the driver is not covered by this work. One could argue that a true breath should be the breath of the operator of the vehicle, thus implying that facial recognition is needed, so driver can be identified as the one doing the test. In future work that issue could be addressed, in order to provide a better safety standard. However that argument is not made in this work, here the argument is made that a breath made in a correct way is a true breath, regardless of who's doing the breath.

6.3 Difficulties

In this section the difficulties will be brought up. This will be done in order to discuss and analyze how some of them were solved. Some are brought up as future problems that may become issues if this work is to be worked with further. In that case, suggestions are made.

6.3.1 Lighting

The issue of illumination and lighting was an issue that proved to be one of the biggest difficulties, this is displayed in Figure 24 using a canny edge detector. Since this work uses for example edge detectors and Haar-like features, which is highly dependent on even illumination, it is important to keep this in mind.



(a) (b)

Fig. 24. This is picture is an example of the lighting issue and how it affects edge detectors. In (a) and (b) the lighting is the only difference, both use a canny edge detection with the same parameters. In (b) a light source was placed below the camera, in order to cause uneven illumination. Because of this not only the mouth is detected but other facial features, in this case facial hair and the edge detection is not very accurate. (a) is an image taken with optimal lighting, which is an evenly lit room.

The main issue with the illumination is that it can be unpredictable. The angle and strength of the light can vary ad infinitum. Because of this the solution needs to generic and work in many cases. One way to solve this is to calibrate the detection for a certain illumination environment, and then making sure that every analysis will be done under the same circumstances. Due to the nature of Haar-like features and the methods used for tracking, the light can't be coming in with an angle. In order to solve this problem there are some suggestions. One way to solve this may be to adapt the environment it's to be used in. For example, installing lighting or a light source that will illuminate the face in a proper way, thus creating a proper environment. The problem may also be solved using for example illumination equalization algorithms [49]. However when using algorithms it's important to remember that they will increase the time of the total analysis.

6.3.2 Camera angle and placement

Due to the nature of Haar-like features, the camera angle and placement needs to show the face from the front. If the angle is too high or low, false or no detection might be an issue. The methods presented are dependent on that the face and mouth detection is correct, so this is an important difficulty to overcome. One way to solve this is by making the device and the camera have a constant relation to each other. Thus the images from the camera will be captured in a controlled and predictable manner. By guaranteeing that every image taken fulfills the required angle and distance from the face, the quality of the analysis is increased. The distance between the face and the camera depends on the camera's field of view. The angle is important since if the angle is off, the illumination may cause issues. Also due to the nature of Haar-like features, the angle needs to be in a certain way for Haar-like to identify certain areas as features.

6.3.3 Facial Hair

Facial hair is a factor which may cause problems of varying severity, this can be seen in Figure 25. If there isn't any or very little facial hair, then it's usually no issue. However in other cases, where there is a lot of facial hair some issues may arise, the facial hair might cause tracking issues. Edge detection suffers because of this. Because edge detection is based on color differences, facial hair around the mouth area can create interference when analyzing. The upper lip area in particular is therefore exposed to facial hair issues. In this work it was solved by looking at the lip and deciding from both intensities and color cues where to find the upper lip.

(a) (b)

Fig. 25. This image shows the effects of facial hair on the tracking. In (a) the facial hair below the mouth causes the contour extraction for the lower lip tracking to make a faulty decision, the color difference of the lower lip and the facial hair may be too similar. In (b) the edge detecting of the corner detection mistakenly tracks the facial hair in the corner area.

6. Related Research

This section will further reference important sources that have influenced this work. Here related research will be compared in order to see if any common problems exist. The paper [44] helped greatly with the mouth corner detection, since it's a paper on the subject of feature tracking it's related to the problem that this work is faced with, in this case the mouth corners detection. The result of that report is used in this work. More research on lip tracking that relates to the work in this report can be seen in [49]. Here a localized color active color model (LCACM) is used in order to extract the contours of the lips. In that paper an illumination equalization algorithm is also presented, and it's to be used for preprocessing purposes. This could be interesting in the future, depending on the performance requirements. However [49] has no solution to the facial hair issue, instead it is put as future work. However, a possible solution is offered. Detect the mustache area in preprocessing and mask it out, this should get rid of the "mustache effect". The issue with facial hair is also brought up in [47] where the presence of beards or mustaches could cause issues.

In [51], a lip localization method is proposed. This method is meant to be used in conjunction with a lip reading method. Its main focus is to accurately find the mouth in varying illumination circumstances. Parts of [51] may be interesting for future work, the shadow consideration in particular. On the subject of detecting faces, lips and mouths, the method proposed in [52] is related this paper. The result of [52] is a real time faces, lips and mouths tracking algorithm. The illumination is once again the issue. The most common issue overall seen in the related research appears to be the lighting and illumination as well as the facial hair.

There has been quite a lot of work done in the area of lip detection. These are some works [53]-[57] that lay close to this work, in regards to the lip point extraction. In comparison to the results presented in this paper their performance is slightly better regarding the correctness of the detection. For example in [53] they achieve an accepted detection of 96.2% and 97.5% for two different sets of images. These numbers represent the detection of both corners, lower and upper lip. In the results for this paper the corner detection and lower lip detection is not that far off with detection in the range of 89.8% and 91.8% respectively. But it falls short when comparing to the 70.8% detection for the upper lip. This is another indicator that there are improvements that can be made to further increase the accuracy of the feature detection. Although since it's not the intent of this work to necessarily outperform other methods of lip detection these results are not too worrying. The goal was to achieve a detection accurate enough to perform the breath identification, and that is achieved. Nevertheless improving the detection is of course something that will be important in the future, because that will help with the correctness of the mouth state detection and in turn the breath identification.

It's important to note that this system runs in real time, in that regard it performs decently well. Compared to [57] that uses a computer with comparable processing power for their testing and images of similar size, where [57] uses (320*240) as to (640*480) used in this thesis, it holds up quite well. They take a little longer to detect the ROIs albeit it also includes the eye region. In regards to the lip feature extraction they achieve an average execution time of 3.35 ms as compared to 7.02 ms for the detection of the corners, upper and lower lip. Even though this is slower it still won't hurt the performance of the system much since most of the execution time is spent calculating the ROI which this system does faster.

In regards to the other works performance, it's evident that improvements can be made to this work to get a more rigorous feature detection. The execution time performs decently well but even here there is room for improvement. Although this is the case, the system still functions well enough to produce the results needed to answer the question of this thesis.

7. Conclusions

The goal has been to investigate if it is possible to use facial detection to detect if a person has made a real breath into an alcolock device, or if it was an attempt at manipulation. To examine if this was something that was possible a prototype has been developed to serve as a proof of concept.

The main problem was to determine a true breath. This work can solve the problem at a good rate using mouth state detection and lip tracking. The lip tracking is done in three steps: corner, lower lip and upper lip detection. The reason for different methods is because each methods solves a problem specific that part of the mouth/lip area. However upper lip proved to be too unreliable so it's not currently used for mouth state. The lower lip and corner tracking achieved good rates of success, and could therefore contribute in a reliable way to the mouth state detection. The mouth state is calculated using the distance between the upper part of the lower lip and the midpoint (calculated from the mouth corners). On the test set the mouth state achieved an 87% success rate. This is acceptable for a prototype, however in a finished product this should be closer to 100%. The biggest issues is the facial hair, illumination and lighting. These problems are still present, mostly the illumination and lighting issues.

The breath identification system manages to detect when an attempt to manipulate is made using a carbon filter. During the testing it could not beat the system once. If a regular breath is made the system will recognize that most of the time. In some cases the detection of the mouth ROI might fail, and that could cause the breath identification to fail. A failed mouth state identification could also be the cause of failure but that is much rarer. Nevertheless the identification will work as it should most of the time.

Based on this the conclusion is made that this approach is indeed a feasible way of being able to detect if a manipulation attempt or a real breath is made. Of course there exists areas that can be improved. But that is to be expected when the system developed is in the early stages and more of a prototype and a proof of concept than a final product. Although this is the case the given method really works and that is what this work was supposed to answer.

7.1 Future work

Since the solution developed serves as an early stage prototype used as a proof of concept there are many things that could be done to further develop. There are several different aspects that can be improved and added in the future. The first step would most likely be to implement the system together with an alcolock device as well as improving the upper lip tracking. Doing this would make it possible to receive a signal from the alcolock that a breath as been made and then inform the system that it should start the breath identification. This would also make it possible to make a better calibration of the breath identification process. Furthermore this would be the logical next step to take since its real intended use would be together with the alcolock device.

Some work can be made with improving the feature detection part of the system. Both making the execution time better as well as improving on the methods used to extract the features and try to make them more accurate. This is especially important for the upper lip detection. The more precise the features are, the more precise the breath identification process will become. It would also help with improving the detection of the mouth state, which is necessary for the breath identification. If the detected feature points are more stable and trustworthy, the mouth state detection could be improved to give a more accurate answer. Furthermore another method for detecting the mouth ROI could be considered. One that more accurately can detect the mouth when blowing air. Another area of future work is to explore possible ways of improving the feature detection in more difficult lighting conditions. This due to its intended environment being a vehicle where the light may not be optimal.

Improve the breath detection part of the system. This is connected to the feature extraction. The more features that can be extracted and the more precise they are, makes it possible to make the identification phase better. If there are more requirements that needs to be fulfilled for the attempted breath to be considered a true breath, then it would make the system more challenging to tamper with. Additionally if the breath identification is more rigorous it will also reduce the number of false negatives, meaning the number of genuine breaths that the system labels false.

In conclusion regarding the future work, make improvements to the functionality of the system on many points. After all, since the result is at an early stage there are of course many improvements that can be made on many fronts. The first step would however be to connect the system with the alcolock and continue to make other improvements from there.

References

[1] US Dept of Transportation. "Alcohol-Impaired Driving." Internet: http://www-nrd.nhtsa.dot.gov/Pubs/812102.pdf, Dec. 2014 [May.15, 2015].

[2] Näringsdepartementet, (2010, Okt) "Alkolås efter rattfylleri." [Online]. Available: http://www.regeringen.se/content/1/c6/15/44/27/027378de.pdf [May. 15, 2015]

[3] Transportstyrelsen. "Fyra av tio rattfyllerister väljer alkolås." Internet: http://www.transportstyrelsen.se/sv/Press/Pressmeddelanden/Fyra-av-tio-rattfyllerister-valjer-alkolas/, Sep. 2, 2013 [May.15, 2015].

[4] E. Pröckl. "Enkelt att blåsa Volvos alkolås." Internet: http://www.nyteknik.se/nyheter/fordon_motor/bilar/article263056.ece, Jan. 23, 2008 [May. 15, 2015].

[5] Y.S. Chen, J.X. Liu, C.C. Tsai, C.T. Chen. "Anti-Counterfeiting System of Drunk Driving Using Driver's Facial image Identification." in *SAE 2011 World Congress & Exhibition*, 2011.

[6] Y. C. Wu, Y. Q. Xia, P. Xie, X. W. Ji, "The design of an automotive anti-drunk driving system to guarantee the uniqueness of driver," in *International Conference on Information Engineering and Computer Science*, 2009, pp. 1-4.

[7] T. Pfister, M. Pietikäinen, X. Li, and G. Zhao "Automated recognition algorithm for detecting facial expressions." U.S. Patent 8 848 068, Sep. 20, 2014.

[8] K. Rawlinson. "Facial recognition technology: How well does it work?." Internet: http://www.bbc.com/news/technology-31112604, Feb. 3, 2015 [May.15, 2015].

[9] K. Bonsor and R. Johnson, (2001, September) "How facial recognition systems work." [Online]. Available: http://electronics.howstuffworks.com/gadgets/high-tech-gadgets/facial-recognition.htm [April. 1, 2015].

[10] M. Williams, (2007, May) "Better Face-Recognition Software." [Online]. Available: http://www.technologyreview.com/news/407976/better-face-recognition-software [April. 1, 2015]

[11] L.J. Mobley, B. McMillin, and J.R. Lewis "Vehicle ignition interlock systems having transdermal alcohol sensor." U.S. Patent 7 299 890, Nov. 27, 2007.

[12] S.V. Chokkalingam. "A Real Time Embedded System Application for Driver Drowsiness and Alcoholic Intoxication Detection." *International Journal Of Innovative Science And Applied Engineering Research (IJISAER)*, vol. 13, no. 44, pp. 7-15, Mar. 2015.

[13] M.V. Ramesh, A.K. Nair, A.T. Kunnath, " Real-Time Automated Multiplexed Sensor System for Driver Drowsiness Detection," in *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, 2011, pp. 1-4.

[14] R.O. Mbouna, S.G. Kong and M.-G. Chun, "Visual analysis of eye state and head pose for driver alertness monitoring." *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 3, pp.1462 -1468, May. 2013

[15] T. Hong, H. Qin, and Q. Sun, “An improved real time eye state identification system in driver drowsiness detection,” in *Control and Automation, 2007. ICCA 2007. IEEE International Conference on*, 2007, pp.1449- 1453.

[16] S. Vitabile, A. Paola and F. Sorbello, "Bright Pupil Detection in an Embedded, Real-time Drowsiness Monitoring System," in *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, 2010, pp. 661 - 668.

[17] T. Hong and H. Qin, "Drivers drowsiness detection in embedded system," in *Vehicular Electronics and Safety, 2007. ICVES. IEEE International Conference on*, 2007, pp. 1-5.

[18] Z. Zhang , J. Zhang, "Driver fatigue detection based intelligent vehicle control," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, 2006, Vol. 2, pp. 1262-1265.

[19] M.S. Devi and P.R. Bajaj, "Driver fatigue detection based on eye tracking," in *Emerging Trends in Engineering and Technology, 2008. ICETET '08. First International Conference on*, 2008, pp.649 -652.

[20] N.H. Thien, P. Venkatesan, T. Muntsinger. "Horizontal Gaze Nystagmus Detection in Automotive Vehicles." unpublished.

[21] M. Saradadevi, P. Bajaj. "Driver Fatigue Detection Using Mouth and Yawning Analysis." *IJCSNS International Journal of Computer Science and Network Security*, vol. 8, no. 6, pp.183-188, June 2008.

[22] X. Fan, B. Yin, Y. Fun. "Yawning Detection For Monitoring Driver Fatigue," in *Machine Learning and Cybernetics, 2007 International Conference on*, 2007, pp. 664-668.

[23] W. Rongben, G. Lie, T. Bingliang and J. Lisheng. "Monitoring mouth movement for driver fatigue or distraction with one camera," in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, 2004, pp. 314-319.

[24] M. Pantic, M. Tomc, LJ.M. Rothkrantz, "A Hybrid approach to mouth features detection," in *Proceeding of the 2001 Systems, Man and Cybernetics Conference*, 2001, pp. 1188-1193.

[25] T.Wang, P. Shi. "Yawning Detection For Determining Driver Drowsiness." in *VLSI Design and Video Technology, 2005. Proceedings of 2005 IEEE International Workshop on*, 2005, pp. 373-376.

[26] R.C. Freund, T.E. Knowles, and S. Couch. "Impairment detection and interlock system with tester identification." U.S. Patent 6 748 792, June. 15, 2004.

[27] Dr. S. Morley, A. Huth, I. Neumann, and Dr. J. Lagois. "Device and method for recognising the correct usage of an alcohol measuring device by a driver in a vehicle." E.P. Patent 2 390 129, Nov. 30, 2011.

[28] P.M. Crespo, and A. Chan. "Breath alcohol ignition interlock device with biometric facial recognition with real-time verification of the user." U.S. Patent 7 823 681, Nov. 2, 2010.

[29] P.J. Igel, and J.M. Igel. "Monitored ignition lock." U.S. Patent 2012 026 8259, Okt. 25, 2012.

[30] M.W. Walter, and D.E. DeVries "Ignition interlock breathalyzer." U.S. Patent 7 934 577, May. 3, 2011.

[31] S. Morley, J. Lagois, I. Neumann, and A. Huth. "Device and process for recognizing correct use of an alcohol-measuring device by a driver in a vehicle." U.S. Patent 2011 029 2209, Dec. 1, 2011.

[32] S. Ioffe, L. Williams, D. Strelow, A. Frome, and L. Vincent "Automatic face detection and identity masking in images, and applications thereof." U.S. Patent 8 509 499, Aug. 13, 2013.

[33] M.H. Yang, D. Kriegman, and N. Ahuja. "Detecting faces in images: A survey." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 1, pp. 34-58, Aug. 2002.

[34] E. Hjelmås, B.K. Low. "Face detection: A survey." *Computer vision and image understanding*, vol. 83, no. 3, pp. 236-274, Sep. 2001.

[35] A.N. Martinez-Gonzalez and V. Ayala-Ramirez. "Real Time Face Detection Using Neural Networks." in *Artificial Intelligence (MICAI), 2011 10th Mexican International Conference on*, 2011, pp. 144-149.

[36] P. Viola and M. Jones, "Robust real-time face detection." *International journal of computer vision*, vol. 57, no. 2, pp.137-154, May. 2004.

[37] P.I. Wilson, and J. Fernandez. “Facial feature detection using Haar classifiers.” *Journal of Computing Sciences in Colleges*, vol. 21, no. 4, pp. 127-133, Apr. 2006.

[38] M. Castrillón, O. Déniz, L. Antón-Canalís, and J. Lorenzo. "Face and Facial Feature Detection Evaluation," in *VISAPP*, Funchal, Portugal, 2008.

[39] H.A. Rowley, S. Baluja, and T. Kanade. "Rotation invariant neural network-based face detection." in *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on*, 1998, pp. 38-44.

[40] R. Feraund, O. Bernier, J. Viallet and M. Collobert, "A Fast and Accurate Face Detector Based on Neural Networks." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 1, pp. 42-53, Jan. 2002.

[41] *Alcohol interlocks - Test methods and performance requirements - Part 1: Instruments for drink-driving-offender programs,* EN 50436-1:2014, 2014

[42] A. Majumder, L. Behera, and V.K. Subramanian. "Automatic and robust detection of facial features in frontal face images," in *Computer Modelling and Simulation (UKSim), 2011 UkSim 13th International Conference on*, 2011, pp. 331-336.

[43] J. Canny, "A Computational Approach to Edge Detection." *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679-698, 1986.

[44] J. Shi and C. Tomasi. "Good features to track", in *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 1994, pp. 593-600.

[45] R.-L. Hsu, M. Abdel-Mottaleb and A.K. Jain, "Face Detection in Color Images." *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 696-707, May 2002.

[46] H. Kalbkhani, and M. Chehel Amirani. "An efficient algorithm for lip segmentation in colour face images based on local information." *Journal of World's Electrical Engineering and Technology*, vol. 1, pp. 12-16, January 2012.

[47] M. Oravec, B. Kristof, M. Kolarik, and J. Pavlovicova. "Extraction of facial features from color images." *Radioengineering*, vol. 17, no. 3, pp. 115-120, 2008.

[48] N. Otsu. " A threshold selection method from gray-level histograms." *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 9, no. 1, pp. 62-66, 1979.

[49] Y. Cheung, X. Liu, and X. You. "A local region based approach to lip tracking," *Pattern Recognition*, vol. 45, no. 9, pp. 3047-3582, Sep. 2012.

[50] http://www.vision.caltech.edu/html-files/archive.html

[51] M. Li, Y. Cheung, 1st Initial. , "Automatic lip localization under face illumination with shadow consideration," Signal Processing, Vol. 89, no. 12, pp. 2425–2434, December 2009.

[52] C.C. Chiang, W.K. Tai, M.T. Yang, Y.T. Huang, C.J. Huang. "A novel method for detecting lips, eyes and faces in real time." *Real-time imaging*, vol. 9, no. 4, pp. 277-287, Aug. 2003.

[53] E. Skodras, N. Fakotakis. "An unconstrained method for lip detection in color images." in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, 2011, pp. 1013-1016.

[54] Y. Tian, T. Kanade and J. Cohn. "Robust Lip Tracking by Combining Shape, Color, and Motion." in *Proc. Asian Conf. Computer Vision*, 2000, pp. 1040-1045.

[55] N. Eveno, A. Caplier, P.Y. Coulon. "Accurate and quasi-automatic lip tracking." *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 14, no. 5, pp. 706-715, May. 2004.

[56] C. Bouvier, P.Y. Coulon "Unsupervised Lips Segmentation Based on ROI Optimisation and Parametric Model." in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, 2007, pp. 301-304.

[57] Y. Jang, W. Woo. "Adaptive Lip Feature Point Detection Algorithm for Real-Time Computer Vision-Based Smile Training System" in *4th International Conference on E-Learning and Games, Edutainment 2009, Banff, Canada, August 9-11, 2009. Proceedings*, 2009, pp. 379-389.