Design and implementation of a driver drowsiness detection system: a practical approach

Aleksandar Colic[1] , Oge Marques[2] and Borko Furht[3]

***Abstract*— This paper describes the steps involved in designing and implementing a driver drowsiness detection system based on visual input (driver's face and head). It combines off-the-shelf software components for face detection, human skin color detection, and eye state (open vs. closed) classification in a novel way. Preliminary results show that the system is reliable and tolerant to many real-world constraints.**

I. INTRODUCTION

The interest in equipping vehicles with driver drowsiness detection systems has been motivated by alarming statistics, such as the 2013 World Health Organization report [1] stating that: 1.24 million people die on the road every year; approximately 6% of all the accidents are caused by drivers driving in a drowsy state; and most of the accidents of this type result in fatalities.

Drowsiness (also referred to as *sleepiness*) can be defined as "the need to fall asleep". This process is a result of normal human biological rhythm and its sleep-wake cycles. The longer the period of wakefulness, the more pressure builds for sleep and the more difficult it is to resist it [2].

The two most commonly used vehicle-based measures for driver drowsiness detection are: the steering wheel movement (SWM) and the standard deviation of lane position (SDLP).

Steering Wheel Movement (SWM) methods rely on measuring the steering wheel angle using an angle sensor mounted on the steering column, which allows for detection of even the slightest steering wheel position changes. When the driver is drowsy, the number of micro-corrections on the steering wheel is lower than the one found in normal driving conditions [3], [4]. A potential problem with this approach is the high number of false positives. SWM-based systems can function reliably only at particular environments and are too dependent on the geometric characteristics of the road and, to a lesser extent, on the kinetic characteristics of the vehicle [5].

Standard Deviation of Lane Position (SDLP) methods are based on an externally-mounted camera and associated software, which monitor the vehicle's relative position to a lane. SDLP-based systems' limitations are mostly tied to their dependence on external factors such as: road marking, weather, and lighting conditions.

For the work described in this paper, we have adopted a subset of *behavioral methods* for driver drowsiness detection. These methods are based on the detection of behavioral clues, e.g., closing of the eyes, yawning and nodding of the head. They typically use a video camera for image acquisition and rely on a combination of computer vision and machine learning techniques to detect events of interest, measure them, and make a decision on whether the driver may be drowsy or not. If the sequence of captured images and measured parameters (e.g., pattern of nodding or time lapsed in "closed eye state") suggest that the driver is drowsy, an action – such as sounding an audible alarm – might be warranted.

The remainder of the paper is structured as follows: Section II describes the purposes, context, and general architecture of our work, Section III reports results from preliminary experiments. Concluding remarks are presented in Section IV.

II. OUR WORK

This section describes the requirements, constraints, basic architecture, and selected algorithms associated with our driver drowsiness detection system. The hallmarks of the proposed system are its robustness, accuracy, and overall simplicity.

A. Requirements and constraints

The driver drowsiness detection system described in this paper must comply with the following main requirements:

- Algorithmically simple and easy to implement. We chose to rely on off-the-shelf solutions for most stages, based on the popularity and success of the associated algorithms (e.g., Viola-Jones face detector, Support Vector Machine classifier).
- Easily portable to different platforms. The application must run on a mobile device (e.g., Android-based smartphone) mounted on the vehicle's dashboard. Ideally, it should be easily portable to other (e.g., iOS-based) mobile devices of comparable size and computational capabilities.
- Computationally non-intensive. Since (near) real-time performance is required, algorithms must be optimized to ensure continuous monitoring of driver's state without excessive burdening of the device's main processor. As a side benefit, battery consumption is reduced as well.

[1]Aleksandar Colic is with Computer & Electrical Engineering and Computer Science Department, Florida Atlantic University, 777 Glades Road, Florida, 33431, USA acolic@my.fau.edu

[2]Oge Marques is with Computer & Electrical Engineering and Computer Science Department, Florida Atlantic University, 777 Glades Road, Florida, 33431, USA omarques@fau.edu

[3]Borko Furht is with Computer & Electrical Engineering and Computer Science Department, Florida Atlantic University, 777 Glades Road, Florida, 33431, USA bfurht@fau.edu

- Accuracy. One of the main challenges of designing such a system is related to the fact that both type I and type II errors are highly undesirable, for different reasons: type I errors (false positives) will annoy the driver and reduce their willingness to use the system (due to excessive false alarms), whereas type II errors (false negatives) can have literally catastrophic consequences and defeat the purpose of the entire system.
- Robustness. The system must be tolerant to modest amounts of lighting variations, relative camera motion (e.g. due to poor road conditions), changes to the driver's visual appearance (even in the course of a session, e.g., by wearing/removing a hat or sunglasses), camera resolution and frame rates, and different computational capabilities of the device's processors.

Some of the anticipated constraints and limitations faced by the proposed system include:

- Lighting conditions. Frequent and drastic change in darkness or brightness of a scene (or part of it), which may happen even during the shortest driving intervals, have been proven to be a significant challenge for many computer vision algorithms.
- Camera motion. Poor road conditions as well as a more aggressive style of driving can introduce significant amount of vibrations and discomfort to the driving experience. Those vibrations can be passed onto the camera and cause distortion in the images which can significantly skew the results and decrease the overall performance of the system.
- Relative positioning of device. The camera must be positioned within a certain range from the driver and within a certain viewing angle. Every computer vision algorithm has a "comfort zone" in which it performs the best and most reliably. If that comfort zone is left, performance can be dropped significantly.
- Hardware and software limitations. Typical mobile devices have one or two processor cores, reduced working memory and tend to work on lower clock frequencies, compared to their desktop counterparts. The reason for all of this is to reduce the energy consumption but it creates a significant obstacle in designing this type of system.
- Driver cooperation. Last, but certainly not least, all driver drowsiness detection systems assume a cooperative driver, who is willing to assist in the setup steps, keep the monitoring system on at all times, and take proper action when warned by the system of potential risks due to detected drowsiness.

B. System architecture

Our driver drowsiness detection system consists of four main stages (Fig. 1):

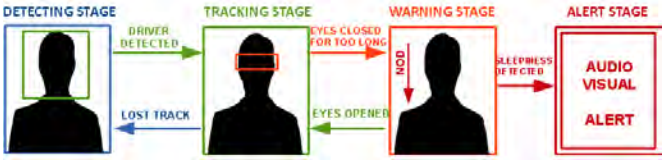


Fig. 1: Four stages of Drowsiness Detection System

1) Detection Stage: This is the initialization stage of the system. Every time the system is started it needs to be set up and optimized for current user and conditions. The key step in this stage is successful head detection (Fig. 2). If the driver's head is correctly located we can proceed to extract the features necessary for setting up the system. Setup steps include: (i) extracting driver's skin color and using that information to create custom skin color model and (ii) collecting a set of open/closed eyes samples, along with driver's normal head position.

To help achieve these goals, user interaction might be required. The driver might be asked to sit comfortably in its normal driving position so that system can determine upper and lower thresholds needed for detecting potential nodding. The driver might also be asked to hold their eyes closed and then open for a matter of few seconds each time. This is enough to get the system started. Over time, the system will expand the dataset of obtained images and will become more error resistant and overall more robust.

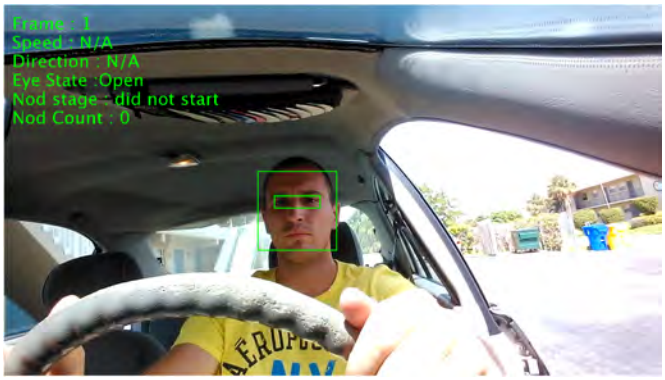


Fig. 2: Successful Initial Face and Eyes Detection

2) Tracking Stage: Once the driver's head and eyes are properly located and all the necessary features are extracted, the system enters the regular tracking (monitoring) stage. A key step in this stage is the continuous monitoring of the driver's eyes within a dynamically allocated tracking area. More specifically, in order to save some processing time, the system will determine the size of the tracking area based on the previous history of eye movements. For example, if the eyes were moving horizontally to the left for a number of frames it is to be expected that that trend with continue in the following frame also. So it is logical to expand the tracking area towards the expected direction of the eyes and shrink the area in other three directions. During this stage, the system must also determine the state

of the eyes. All these tasks must be carried out in real-time; depending on the processor's abilities and current load, it might be necessary to occasionally skip a few frames, without sacrificing algorithmic accuracy.

3) Warning Stage: If the driver keeps his eyes closed for prolonged period of time or starts to nod, alertness has to be raised. The key step within this stage is close monitoring of drivers eyes. The system must determine whether the eyes are still closed, and what is the eyes' position relative to previously established thresholds. We cannot afford to skip frames in this stage. In practice, tracking of eyes is performed much in the same way as in the tracking stage with the addition of the following processes: calculation of velocity and trajectory of the eyes and threshold monitoring. These additional computations are required to improve the system's ability to determine whether the driver is drowsy or not.

4) Alert Stage: Once it has been determined that the driver appears to be in an abnormal driving state, the system has to be proactive and alert the driver of potential dangers that can arise. Combination of audio/visual alerts are used to attract the driver's attention and rase their alertness level. Alerting has to be implemented in such a way as not to cause the opposite effect of intended and startle the driver into causing an accident.

C. Selected algorithms for feature detection and image classification

The proposed system relies on three main features of a driver: face, eyes and skin. We have chosen the popular Viola-Jones algorithm for baseline face detection, due to its wide availability and overall simplicity. We enhanced the face detection process to make use of skin color information. Human skin color has unique features. These features can be best expressed and described by breaking given color into its basic chroma components (red, green, blue), and defining components' ranges. It has been shown that the vast majority of skin color types have their red chroma component in a range between 133 and 173, and their blue chroma component in a range of 77 – 127. One of the contributions of this work is the adoption of a *user-specific* red chroma range. Fig. 3 shows differences between using generalized range and custom, user specific range.

In the initialization stage of the system, when the face of the driver is detected, the area containing the face is used to analyze the specific red chroma range in which current driver's skin color falls in. Chroma values can fall in the range between 0 and 255. Histogram of red chroma component of the area containing driver's face is created to give us that specific range. This specific range is much narrower then the generalized range. We can extract the upper and lower boundary from the histogram and use it in the following stages of the system. So, when the system tracks the eyes or face in the following frames, by analyzing the red chroma component histogram we can confirm that the tracked object really is pair of eyes or a driver's face.

Once our detection algorithm has successfully detected a face and, subsequently, the eyes, it focuses on determining in

(a) Generalized skin color chroma range

(b) User-specific skin color chroma range

Fig. 3: Chroma-based skin detection comparison

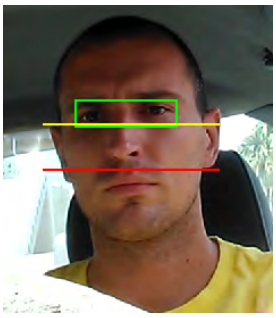
what state the eyes are (closed or open). The proposed system monitors if the driver's eyes are being closed for prolonged period of time. If that is the case, we can conclude that the driver might be experiencing signs of drowsiness. The classification method implemented in our system uses data from the (most recent) setup stage as training data that is applied to a 2-class Support Vector Machine (SVM) classifier whose job is to distinguish the difference between open and closed eyes.

In order to increase the overall robustness of the system, we have adopted a classification approach that relies on both head position and eye state. Our approach is based on the following premises:

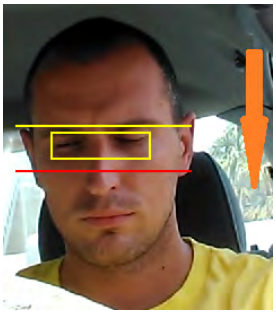
- Driver's head position does not deviate a lot when fully awake.
- When sleepy, head position changes drastically.

Our system implements a dynamic two-threshold approach that is simple and effective way of detecting abnormal head behavior of a driver. The upper threshold is positioned slightly beneath the eyes. As long as the eyes are located above this threshold system considers this to be wide awake, active state of the driver in which he is looking ahead towards the road. When the eyes start going vertically down and cross the upper threshold the system is preparing for potential nodding. If the head continues to nod forward and the eyes eventually cross the lower threshold we know that driver is certainly not focused on the road. Human nodding is very

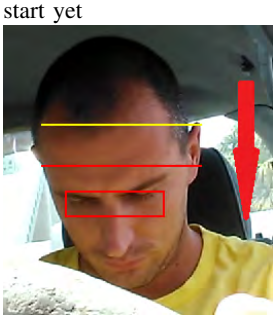
specific and consists of head slowly dropping down followed by rapid recovering right back into original position. Our system takes this into account. The speed of each stage of the nodding is measured in order to filter out potential false positives such as driver nodding in agreement to something. Fig. 4 shows the nodding detection method.



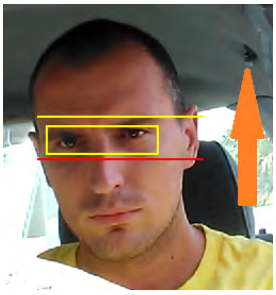
(a) Eyes above the upper threshold - nodding did not start yet



(b) Head tilting down - Eyes beneath the upper threshold



(c) Head tilting down - Eyes beneath the lower threshold



(d) Head recovering up - eyes above lower threshold

(e) Head recovering up - eyes above upper threshold: Nod is complete

Fig. 4: Nodding detection method and its stages.

As long as the eyes are held above the upper threshold, the system is considered to be in a normal state. Once the head start tilting vertically down and crosses the upper threshold, it is considered a beginning of the nodding sequence. From here, if the head recovers back up, the nodding sequence is canceled; but if the head continues down and crosses the lower threshold, we know that the driver is certainly not monitoring the road anymore. Also, while monitoring the position of the eyes relative to the thresholds, the system also monitors the state of the eyes. If at anytime the driver opens his eyes while going down in his nodding sequence, the sequence is canceled. For a "true nod", the driver has to keep his eyes closed while moving his head downwards. Once the lower threshold is crossed, the system can expect rapid vertical recovery of the driver's normal driving position with potential opening of the eyes while moving upward. The system also monitors the velocity of the nod since it is essential in distinguishing a true sleepy nodding from other potentially similar head movements that are *not* caused by nodding.

III. Preliminary Experiments

In order to fulfill all of the challenging prerequisites we have to test out limitations of our chosen off-the-shelf components. Some of our preliminary tests were devised to help us out with that. We started by setting our work environment in MATLAB. For the purposes of testing, we simplified our dynamic solution to its core, into a horizontal, linear solution that can perform basic face/eye detection as well as simple classification task of determining eye state. The goal of the devised set of experiments is to test the basic performance in various ways.

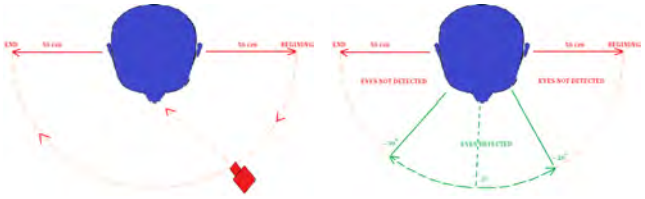
From the very beginning we decided to use Viola-Jones algorithm, Haar-like feature based face/eyes detection algorithm available in MATLAB [6]. It is known for being stable and computationally non-intensive algorithm. Haar-like feature can be described as a set of two or more adjacent rectangular regions organized in a specific way in a given detection window. This algorithm proves to be compute non-intensive since it's core operation is simple summing up of the pixel intensities in a given regions, followed by calculation of differences between them. That difference is then used to categorize subsections of an image to be face/eyes or not.

Also, for differentiating between open and closed eyes we decided to go with a proven two-class Support Vector Machine (SVM) classifier [7]. SVM is a non-probabilistic binary linear classifier. It takes a set of input data and predicts, for each given input, which of two possible classes forms the output. This is perfect since our basic problem can be defined as two class problem: open eyes vs. closed eyes.

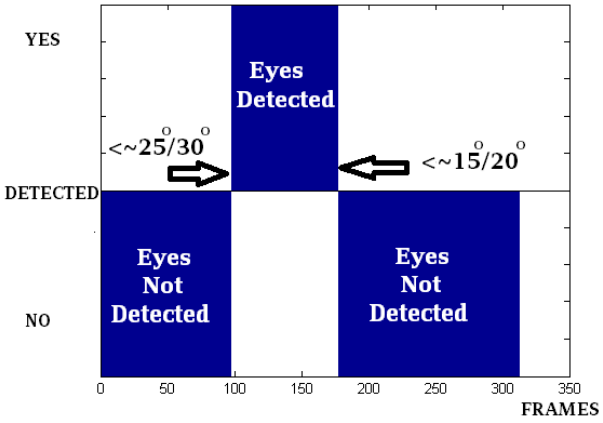
A. Camera rotation test

The aim of this test is to find out how does movement of the camera and change in the viewing angle relative to the driver influences the performance of used face/eye detection algorithm as well as the consistency of its detection. The experiment was setup in following manner: the driver is sitting inside the car, looking towards the road; the camera, approximately 50 centimeters away from the driver's head, will make a semi-circle around the driver's head, keeping the same distance throughout; the camera starts pointed at driver's left profile and ends pointing at driver's right profile. The driver's head stays stationary, only the camera moves around it while keeping constant distance from it. Video used for testing contains 310 frames of driver's head from various angles. Fig. 5 shows the process and the results. We can deduct that the detection algorithm performs consistently. If we define an angle of 0 degrees to be when

the camera is directly facing the driver, we can conclude that the range in which it was reliably detecting drivers eyes falls approximately within 25 degree viewing angle in both direction from a driver. This result is very encouraging since this means that we can position the camera of the system conveniently on car's dashboard (usually has available slot for attaching devices such as phones etc.) which is within driver's reach.



(a) Test setup (b) Angle range results



(c) Consistency results

Fig. 5: Camera rotating: angle change limitations.

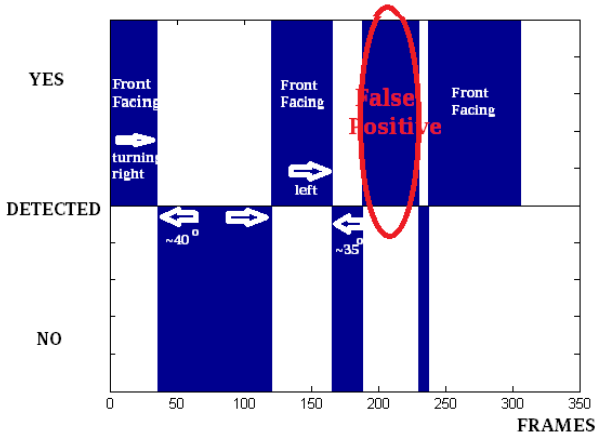
B. Head rotation test

In this test the camera stays directly facing the driver while the driver rotates his head to the right, returns to the starting position and then rotates the head to the left and back to center again. We are again looking to see what is the angle of breaking for our head/eye detection algorithm and what is its consistency across the frames. Fig. 6 is showing the tests setup and results. Difference in detection angles is much smaller than in previous test, only around 5 degrees. It seems that the algorithm is consistently and reliably detecting drivers eyes as long as the driver is facing towards the camera and his gaze does not deviate more then 35 degrees in both directions from the front facing position. Even though the difference in detection angles is much smaller, there is an obvious false positive produced when drivers head turns almost 90 degrees to the left. Though one eye is visible the other eye is completely occluded. Background influenced the algorithm into wrong conclusion.

C. Real-World Test

For pure test of consistency of the detection algorithm we recorded an approximately 900 frame video sequence of a

(a) Test setup (b) Angle range results



(c) Consistency results

Fig. 6: Head rotating: angle change limitations.

driver behaving naturally behind the wheel of the car. He is moving around, adjusting his seat, mirror, turning to face the passengers. We want to see if there are false positives and in what volume and how reliable or basis detection algorithm really is. Fig. 7 is showing quite good consistency: there are no sudden changes except with one spike for the duration of one frame. This spike actually is a true positive.

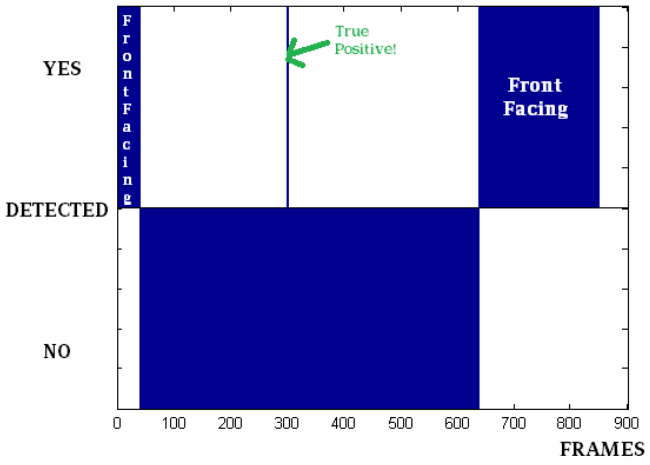
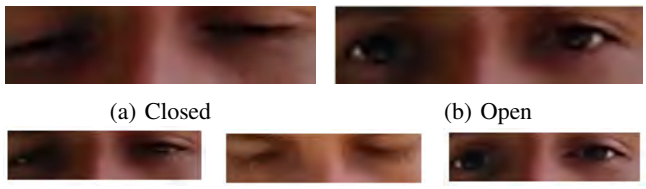


Fig. 7: Real-World test results

D. Open vs. Closed Eyes test

To test the behavior of our chosen classification algorithm we decided to go with the simple solution of using gray-scaled and cropped image regions containing eyes and to

feed that set of pixels as an input to the SVM classification algorithm. A special video containing 571 frames of a driver sitting in a drivers position is used. Driver is moving minimally. Every frame is manually marked as containing open or closed eyes. Open and closed eye samples are roughly the same in number, 304 containing open eyes and 267 containing closed eyes. 70 % of randomly selected samples were used to train the SVM model on which the remaining 30% of the samples were tested. Using available SVM MATLAB version, our preliminary results showed 96 % success rate. We are acknowledging the fact that the sample size is very small and that we only had samples relating to one subject so instead on focusing on high accuracy rate we concluded that building a system in which an SVM eye model is user specific is encouragingly good idea. It can simplify significantly our system as well as provide a dose of robustness and reliability to it. Such an eye model can always be upgraded, thus increase its quality, through frequent use of the system. Some of the classification results are shown on Fig. 8.



(a) Closed (b) Open

(c) Called closed (d) Called open (e) Called closed

Fig. 8: Support Vector Machine result examples: (a) & (b) correct; (c), (d) & (e) incorrect.

IV. Concluding Remarks

In this paper, we have described the process of designing and implementing a driver drowsiness detection system by combining some off-the-shelf algorithms with some of the novel approaches in a clever manner. The system is dynamic, it can update and modify its components like current drivers eye model and skin model throughout its life cycle. Constantly upgrading baseline models used can increase the overall resilience towards errors. Moreover, the system is user specific. All the feature models created are solely based on the current users features instead of using generalized parameters. Such an approach simplifies the system while providing solid performance. Each of the algorithms is performing solidly by itself. But they do have their limitations. To increase the reliability and accuracy of the system, both baseline detection/tracking algorithm and eye-state classification algorithm are complimented with simple, but efficient, custom algorithms.

References

[1] World Health Organization, *Global Status Report on Road Safety 2013: Supporting a Decade of Action : Summary*. World Health Organization, 2013. [Online]. Available: http://books.google.com/books?id=qzK2nQEACAAJ

[2] T. Akertedt, P. Fredlung, M. Gillberg, and B. Jansson, "A prospective study of fatal occupational accidents relationship to sleeping difficulties and occupational factors," *Journal of Sleep Research*, vol. 11, no. 1, pp. 69–71, 2002. [Online]. Available: http://dx.doi.org/10.1046/j.1365-2869.2002.00287.x

[3] S. H. Fairclough and R. Graham, "Impairment of driving performance caused by sleep deprivation or alcohol: A comparative study," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 41, no. 1, pp. 118–128, 1999.

[4] R. Feng, G. Zhang, and B. Cheng, "An on-board system for detecting driver drowsiness based on multi-sensor data fusion using dempster-shafer theory," in *Networking, Sensing and Control, 2009. ICNSC '09. International Conference on Networking*, 2009, pp. 897–902.

[5] E. Vural, "Video based detection of driver fatigue," Ph.D. dissertation, Sabanci University, 2009.

[6] P. Viola and M. Jones, "Robust real-time object detection," in *International Journal of Computer Vision*, 2001.

[7] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. [Online]. Available: http://dx.doi.org/10.1007/BF00994018