University of Diyala - College of Science

Computer Science Department

Drowsiness Detection System

This research was presented to the Council of the College of Science - University of Diyala - Department of Computing as part of the requirements to get a bachelor's degree in Computer science

By:

Aisha Ahmed Ali

Nadya Ali Sultan

Under the Supervision of:

Dr. Bashar Talib AL-Nuaimi

1442 A.H. 2021 A.D.

ABSTRUCT

In the last years, the traffic accidents study is become important because they produce several died and hurt around the world. To help in reducing this fatality, in this project, a Drowsiness Detection System for automatic driver’s drowsiness detection based on visual information and Machine Learning is presented. This system works on several stages to be fully automatic. In addition, the aim of this algorithm is to locate and to track the face and the eyes to compute a drowsiness index. Examples of different driver’s images taken over real vehicle are shown to validate the algorithm that works in real time.

SUPERVISOR CERTIFICATION

I certify that the preparation of this project entitled

Drowsiness Detection System

Prepared By

Aisha Ahmed Ali

Nadya Ali Sultan

was made under my supervision in the Department of Computer Science/College of Science/University of Diyala and it is part of the requirements for obtaining a Bachelor's degree in Computer Science

Signature :

Name :

Date :

الاهداء

إلى النور الذي ينير لي درب النجاح (أبي)

و يا من علمتني الصمود مهما تبدلت الظروف (أمي)

إلى من يضيئون لي الطريق ويساندوني ويتنازلون عن حقوقهم لإرضائي والعيش في هناء (أخوتي)

إلى روح من أنارت لي الطريق وامسكت لي مشعل النور استاذتي وقدوتي (أختي مروة) وإلى اساتذتي الأفاضل

د. بشار وجميع اساتذة قسم علوم الحاسبات،

إلى كل من أضاء بعلمه عقل غيره،

أو هدى بالجواب الصحيح حيره سائليه

فأظهر بسماحته تواضع العلماء

وبرحابته سماحه العارفين.

شكر وتقدير

اشكر الله العلي القدير الذي أنعم عليَّ بنعمة العقل والدين. القائل في محكم التنزيل "وَفَوْقَ كُلِّ ذِي عِلْمٍ عَلِيمٌ" سورة يوسف آية 76.... صدق الله العظيم .

وقال رسول الله (صلي الله عليه وسلم):"من صنع إليكم معروفاً فكافئوه, فإن لم تجدوا ما تكافئونه به فادعوا له حتى تروا أنكم كافأتموه" ..... (رواه أبو داوود) .

فبعد شكر المولى عز وجل ، المتفضل بجليل النعم ، وعظيم الجزاء.. يجدر بي أن أتقدم ببالغ الامتنان ، وجزيل العرفان إلى كل من وجهني ، وعلمني ، وأخذ بيدي في سبيل إنجاز هذا البحث .. وأخص بذلك مشرفي الذي قوم ، وتابع ، وصوب ، بحسن إرشاده لي في كل مراحل البحث ، والذي وجدت في توجيهاته حرص المعلم ، التي تؤتي ثمارها الطيبة بإذن الله ...

كما أحمل الشكر والعرفان لكل من أمدني بالعلم ، والمعرفة ، وأسدى ليَّ النصح ، والتوجيه ، وإلى ذلك الصرح العلمي الشامخ متمثلاً في جامعة ديالى ، وأخص بالذكر كلية العلوم ، قسم الحاسبات، والقائمين عليها ,كما أتوجه بالشكر إلى كل من ساندني بدعواته الصادقة ، أو تمنياته المخلصة

أشكرهم جميعاً وأتمنى من الله عز وجل أن يجعل ذلك في موازين حسناتهم.

Table of Contents

List of Figures

CHPTER 1

INTRODUCTION

1.1 Introduction

Driver drowsiness is one of the leading causes of motor vehicle crashes. This was confirmed by a study [1] conducted by the AAA Foundation for Traffic Safety, which showed that 23.5% of all automobile crashes recorded in the United States in 2015 were sleep-related: 16.5% for fatal crashes and 7% for non-fatal crashes. Essentially, this report implied that over 5,000 Americans lost their lives as a result of sleep-related vehicular crashes.

The development of drowsiness detection technologies is both an industrial and academic challenge. In the automotive industry, Volvo developed the Driver Alert Control which warns drivers suspected of drowsy driving by using a vehicle-mounted camera connected to its lane departure warning system (LDWS). Following a similar vein, an Attention Assist System has been developed and introduced by Mercedes-Benz that collects data drawn from a driver's driving patterns incessantly ascertains if the obtained information correlates with the steering movement and the driving circumstance at hand.

Notably, the use of these safety systems which detect drowsiness is not widespread and is uncommon among drivers because they are usually available in luxury vehicles. An increased embedding and connecting of smart devices equipped with sensors and mobile operating systems like Android, which has the largest installed operating system in cars, was shown by surveys in 2015 [2]. In addition, machine learning has made groundbreaking advances in recent years, especially in the area of deep learning. Thus, the use of these new technologies and methodologies can be an effective way to not only increase the efficiencies of the existing real-time driver drowsiness detection system but also provide a tool that can be widely used by drivers.

1.2 Problem Statement

Road crashes and related forms of accidents are a common cause of injury and death among the human population. According to 2015 data from the World Health Organization, road traffic injuries resulted in approximately 1.25 million deaths worldwide, i.e. approximately every 25 seconds an individual will experience a fatal crash. In this project, a novel approach towards real-time drowsiness detection system is proposed. This approach is based on a machine learning method that can be implemented with python with high accuracy.

1.3 Project Objective

The objective of this project is to bulit of a system capable of detecting drowsiness in the driver while driving and alerting him to prevent traffic accidents that may occur due to drowsiness of the driver.

1.4 Project Benefit

The benefit of this study is to provide a system to detect driver drowsiness to easily reduce catastrophic accidents or something else.

CHPTER 2

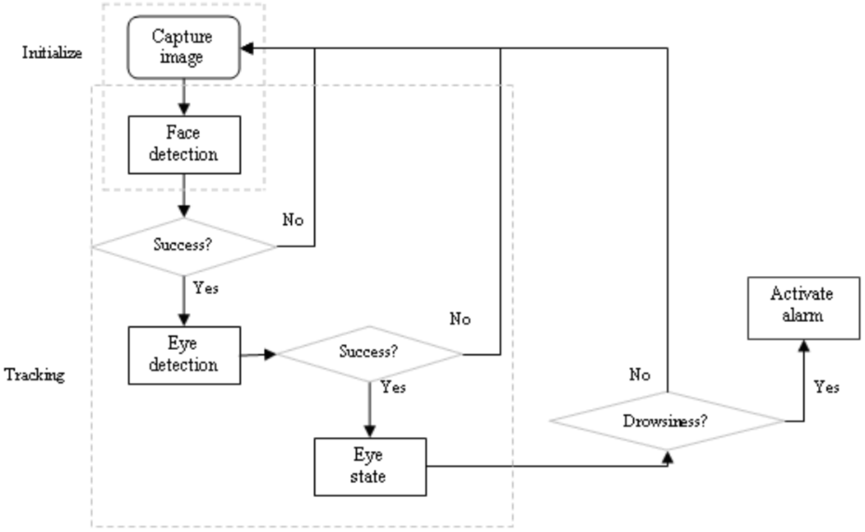
LITERATURE REVIEW

2.1 Review of Related Work

To analyze driver's drowsiness several systems have been built. They usually require simplifying the problem to work partially or under special environments, for example, Ji et al. in [3] and [4] has presented a detection drowsiness system based on infrared light illumination and stereo vision. This system localizes the eye position using image differences based on the bright pupil effect. Afterwards, this system computes the blind eyelid frequency and eye gaze to build two drowsiness indices: PERCLOS and AECS. Bergasa et al. [5] also has developed a non-intrusive system using infrared light illumination, this system computes driver vigilance level using a finite state automata with six eye states that computes several indices, among them, PERCLOS; on the other hand, the system is able to detect inattention through face pose analysis. Another work using infrared illumination is presented by Grace et al. [7] for measuring slow eyelid closure. D'Orazio et al. in [6] has proposed an eye detection algorithm that searches the eyes on the whole image assuming that the iris is always darker than the sclera and based on circle Hough transform and geometrical constrains the eyes candidates are located, next, they are passed to a neural network that classify between eyes and non-eyes. This system is able to classify the eyes between open or closed state. The mains limitations of this algorithm are: it is applicable when the eyes are only visible in the image, and it is not robust at changing illumination.

2.2 Proposed System

In this Python project, we have built a drowsiness detection system that you can implement in numerous ways. We used OpenCV to detect faces and eyes using a hear cascade classifier and then we used a CNN model to predict the status.



\

Figure 2.1 scheme of the system

To localize the face, this system uses OpenCV which is a machine learning approach for visual object detection. It uses three important aspects to make an efficient object detector based on the integral image, AdaBoost technique and cascade classifier [8]. Each one of these elements is important to process the images efficiently and near real-time with 90% of correct detection. A further important aspect of this method is its robustness under changing light conditions. However, in spite of the abovementioned, its principal disadvantage is that it cannot extrapolate and does not work appropriately when the face is not in front of the camera axis. Such would be the case when the driver moves his/her head; however, this shortcoming will be analyzed later on.

2.3 OpenCV

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 18 million. The library is used extensively in companies, research groups and by governmental bodies.

2.4 CNN Model

Convolutional Neural Networks (CNN) are typically used to analyze image data and map images to output variables. However, we decided to build a 1-D CNN and send in numerical features as sequential input data to try and understand the spatial relationship between each feature for the two states. Our CNN model has 5 layers including 1 convolutional layer, 1 flatten later, 2 fully connected dense layers, and 1 dropout layer before the output layer. The flatten layer flattens the output from the convolutional layer and makes it linear before passing it into the first dense layer. The dropout layer randomly drops 20% of the output nodes from the second dense layer in order to prevent our model from overfitting to the training data. The final dense layer has a single output node that outputs 0 for alert and 1 for drowsy.

2.5 Model Architecture

The model we used is built with Keras using Convolutional Neural Networks (CNN). A convolutional neural network is a special type of deep neural network which performs extremely well for image classification purposes. A CNN basically consists of an input layer, an output layer and a hidden layer which can have multiple numbers of layers. A convolution operation is performed on these layers using a filter that performs 2D matrix multiplication on the layer and filter.

The CNN model architecture consists of the following layers:

- Convolutional layer; 32 nodes, kernel size 3

- Convolutional layer; 32 nodes, kernel size 3
- Convolutional layer; 64 nodes, kernel size 3
- Fully connected layer; 128 nodes

The final layer is also a fully connected layer with 2 nodes. In all the layers, a Relu activation function is used except the output layer in which we used SoftMax.

2.6 Prerequisites

The requirement for this Python project is a webcam through which we will capture images. You need to have Python (3.6 version recommended) installed on your system.

2.6.1 Python (programming language)

We will use Python to build the project. Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects[9] .

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library[9] .

then using pip, you can install the necessary packages.

1. **OpenCV –** pip install opencv-python (face and eye detection).
2. **TensorFlow –** pip install tensorflow (keras uses TensorFlow as backend).
3. **Keras –** pip install keras (to build our classification model).
4. **Pygame –** pip install pygame (to play alarm sound).

CHPTER 3

SYSTEM IMPLEMENTATION

3.1 Project Data Set

The dataset used for this model is created by us. To create the dataset, we wrote a script that captures eyes from a camera and stores in our local disk. We separated them into their respective labels ‘Open’ or ‘Closed’. The data was manually cleaned by removing the unwanted images which were not necessary for building the model. The data comprises around 7000 images of people’s eyes under different lighting conditions. After training the model on our dataset, we have attached the final weights and model architecture file “models/cnnCat2.h5”.

Now, you can use this model to classify if a person’s eye is open or closed.

3.2 Project Implementation

In our project, using the Python language, we will import the necessary library in the first step of the project.

```
#Importing the libraries

import cv2

from keras.models import load_model

import numpy as np

from pygame import mixer
```

Let’s now Explain how our algorithm works step by step.

Step 1 – Take Image as Input from a Camera

With a webcam, we will take images as input. So to access the webcam, we made an infinite loop that will capture each frame. We use the method provided by OpenCV, **cv2.VideoCapture(0)** to access the camera and set the capture object (cap). **cap.read()** will read each frame and we store the image in a frame variable.

Step 2 – Detect Face in the Image and Create a Region of Interest (ROI)

To detect the face in the image, we need to first convert the image into grayscale as the OpenCV algorithm for object detection takes gray images in the input. We don't need color information to detect the objects. We will be using haar cascade classifier to detect faces. This line is used to set our classifier **face = cv2.CascadeClassifier(' path to our haar cascade xml file')**. Then we perform the detection using **faces = face.detectMultiScale(gray)**. It returns an array of detections with x,y coordinates, and height, the width of the boundary box of the object. Now we can iterate over the faces and draw boundary boxes for each face.

```
for (x,y,w,h) in faces:

        cv2.rectangle(frame, (x,y), (x+w, y+h), (100,100,100), 1 )
```

Step 3 – Detect the eyes from ROI and feed it to the classifier

The same procedure to detect faces is used to detect eyes. First, we set the cascade classifier for eyes in **leye** and **reye** respectively then detect the eyes using **left_eye = leye.detectMultiScale(gray)**. Now we need to extract only the eyes data from the full image. This can be achieved by extracting the boundary box of the eye and then we can pull out the eye image from the frame with this code.

```
l_eye = frame[ y : y+h, x : x+w ]
```

l_eye only contains the image data of the eye. This will be fed into our CNN classifier which will predict if eyes are open or closed. Similarly, we will be extracting the right eye into r_eye.

Step 4 – Classifier will Categorize whether Eyes are Open or Closed

We are using CNN classifier for predicting the eye status. To feed our image into the model, we need to perform certain operations because the model needs the correct dimensions to start with. First, we convert the color image into grayscale using **r_eye = cv2.cvtColor(r_eye, cv2.COLOR_BGR2GRAY)**. Then, we resize the image to 24*24 pixels as our model was trained on 24*24 pixel images **cv2.resize(r_eye, (24,24))**. We normalize our data for better convergence **r_eye = r_eye/255** (All values will be between 0-1). Expand the dimensions to feed into our classifier. We loaded our model using **model = load_model('models/cnnCat2.h5')** . Now we predict each eye with our model

lpred = model.predict_classes(l_eye). If the value of lpred[0] = 1, it states that eyes are open, if value of lpred[0] = 0 then, it states that eyes are closed.

Step 5 – Calculate Score to Check whether Person is Drowsy

The score is basically a value we will use to determine how long the person has closed his eyes. So if both eyes are closed, we will keep on increasing score and when eyes are open, we decrease the score. We are drawing the result on the screen using cv2.putText() function which will display real time status of the person.

```
cv2.putText(frame, "Open", (10, height-20), font, 1, (255,255,255), 1,
cv2.LINE_AA )
```

A threshold is defined for example if score becomes greater than 10 that means the person's eyes are closed for a long period of time. This is when we beep the alarm using sound.play().

3.3 Project Example

Let's start our project and see the working of our project. It may take a few seconds to open the webcam and start detection. The following figures show the work of the project when opening and closing the eyes and when alerting the driver.

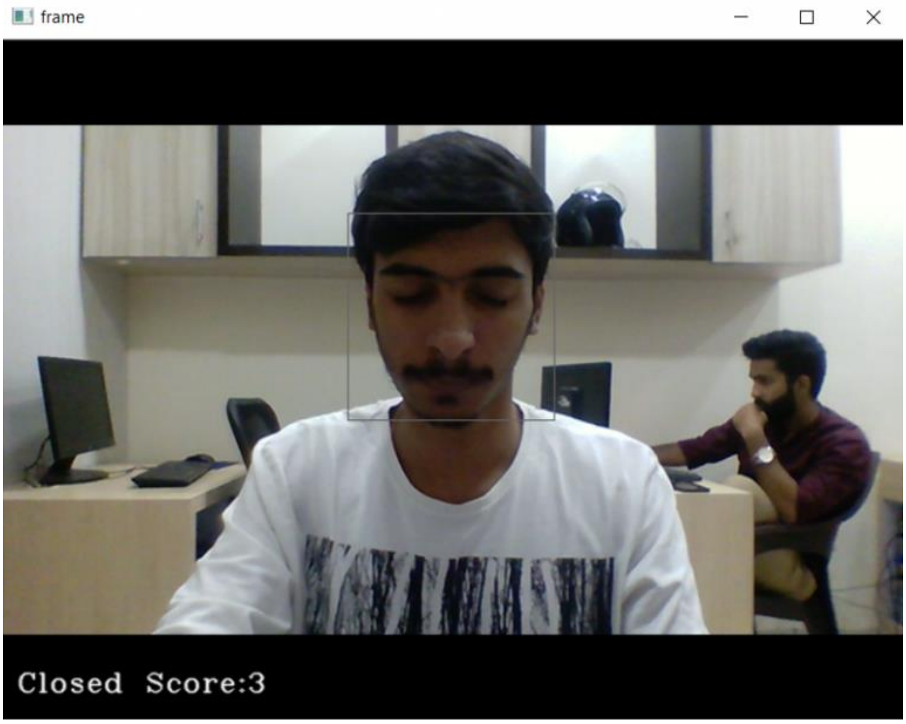


Figure 3.1 Closed Eye Detection

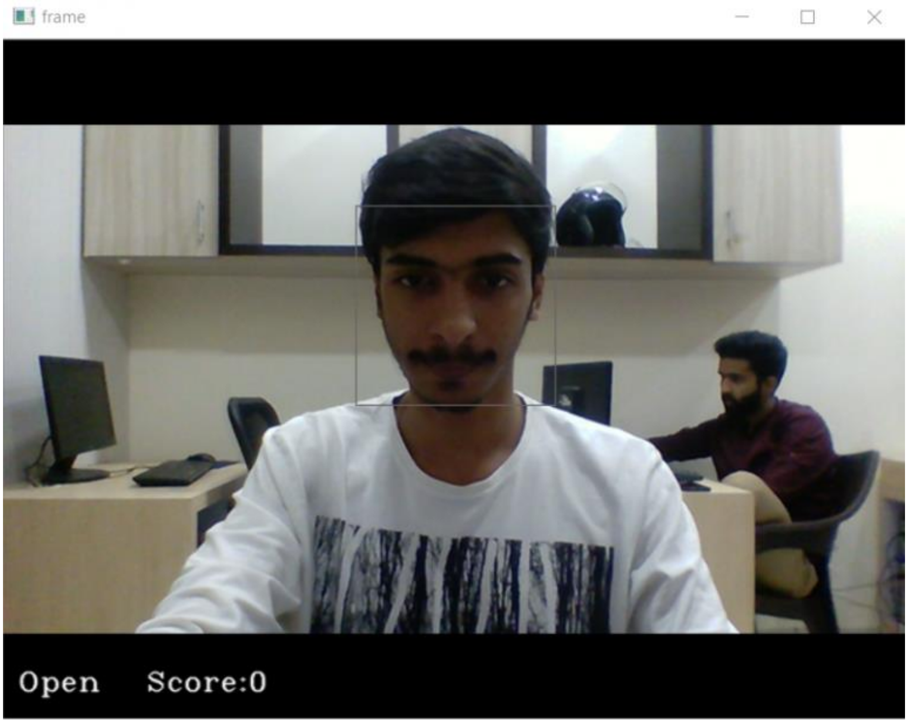
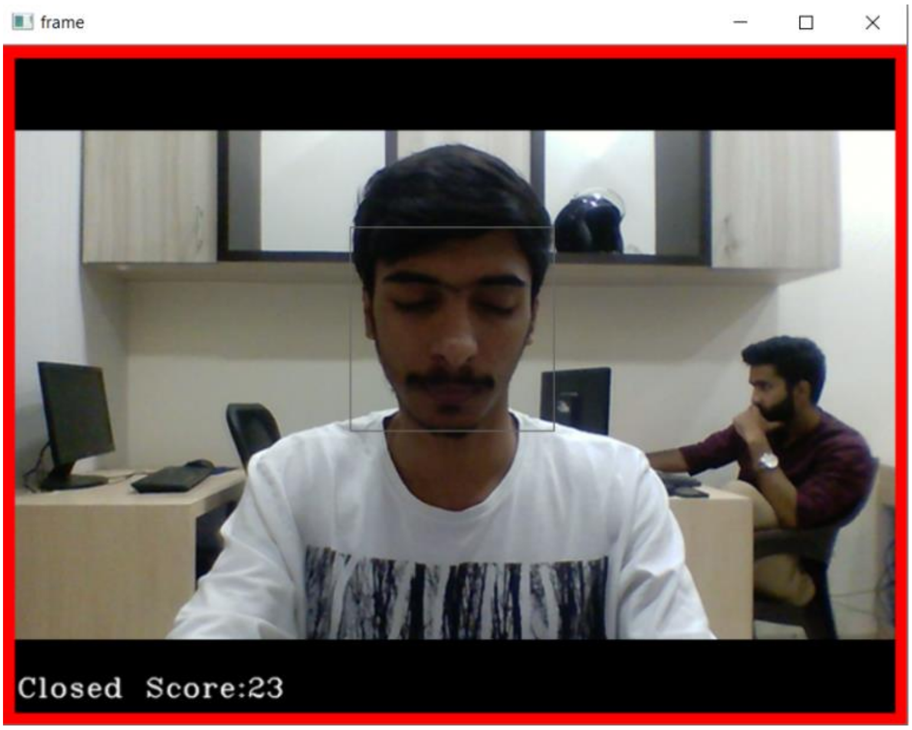


Figure 3.2 Open Eyes Detection



Closed Score:23

Figure 3.3 Sleep Alert

CHPTER 4

CONCLUSION AND SUGGESTIONS

4.1 Conclusions

Drowsiness detection is a safety technology that can prevent accidents that are caused by drivers who fell asleep while driving. So we built of a system capable of detecting drowsiness in the driver while driving and alerting him to prevent traffic accidents that may occur due to drowsiness of the driver.

4.2 Suggestions

Moving forward, there are a few things we can do to further improve our results and fine-tune the models. First, we need to incorporate distance between the facial landmarks to account forany movement by the subject in the video. Realistically the participants will not be static on the screen and we believe sudden movements by the participant may signal drowsiness or waking up from micro-sleep. Second, we want to update parameters with our more complex models (NNs, ensembles, etc.) in order to achieve better results..

References

[1] Drowsy Driving NHTSA reports. (2018, January 08). Retrieved from https://www.nhtsa.gov/risky-driving/drowsy-driving

[2] Cornez T, Cornez R. Android Programming Concepts. Jones & Bartlett Publishers; 2015.

[3] Q. Ji, Z. Zhu y P. Lan. : Real Time Nonintrusive Monitoring and Prediction of Driver Fatigue, IEEE Transactions on Vehicular Technology, Vol. 53, Nº 4 (2004).

[4] Ji Q. and Yang. X.: Real-Time Eye, Gaze, and Face Pose Tracking for Monitoring Driver Vigilance. Real Time Imaging, Nor. 8, Pg. 357- 377, Elsevier Science Ltd, (2002).

[5] Bergasa L., Nuevo J., Sotelo M. and Vazquez M.: Real Time System for Monitoring Driver Vigilance. IEEE Intelligent Vehicles Symposium (2004).

[6] D’Orazio T., Leo M. and Distante A.: Eye Detection in Faces Images Vigilante System. IEEE Intelligent Vehicles Symposium University of Parma, Italy June, 14-17 (2004).

[7] Grace R.: Drowsy Driver Monitor and Warning System. International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design (2001)

[8] Viola P. and Jones M.: Rapid Object Detection using a Boosted Cascade of Simple Features. Accepted Conference on Computer Vision and Pattern

Recognition(2001).

[9] Kuhlman, Dave. "A Python Book: Beginning Python, Advanced Python, and Python Exercises". Section 1.1. Archived from the original (PDF) on 23 June 2012.

((إقرار المشرف))

اشهد بأن اعداد هذا المشروع الموسوم

نظام كشف النعاس

والمعد من قبل الطلاب :-

عائشة أحمد علي

نادية علي سلطان

قد تم تحت أشرافي في قسم علوم الحاسوب / كلية العلوم/جامعة ديالى وهي جزء من متطلبات نيـــــــل شهادة البكالوريوس في اختصاص علوم الحاسوب

التوقيع:

الاسم:

المرتبة العلمية :

التاريخ :

الخلاصة

في السنوات الأخيرة ، أصبحت دراسة حوادث المرور مهمة، لأنها تؤدي إلى موت وإصابة العديد من الأشخاص حول العالم. للمساعدة في الحد من هذه الوفيات ، في هذا المشروع ، يتم تقديم نظام للكشف عن النعاس يكشف بشكل تلقائي عن نعاس السائق بناءً على المعلومات المرئية والتعلم الآلي. يعمل هذا النظام على عدة مراحل ليكون آلياً بالكامل. بالإضافة إلى ذلك ، الهدف من هذه الخوارزمية هو تحديد وتتبع الوجه والعينين لحساب مؤشر النعاس. يتم عرض أمثلة لصور مختلفة للسائقين تم التقاطها على مركبة حقيقية للتحقق من صحة الخوارزمية التي تعمل في الوقت الفعلي.

جامعة ديالى – كلية العلوم

قسم علوم الحاسبات

نظام كشف النعاس

بحث مقدم الى مجلس كلية العلوم – جامعة ديالى – قسم الحاسبات كجزء من متطلبات الحصول على شهادة البكالوريوس في علوم الحاسوب

إعداد الطلاب

عائشة أحمد علي

نادية علي سلطان

اشرف على البحث

ا.م.د بشار طالب حميد

1442هـ

2021م