Cursor Control with Eye and Head Gestures

COM 402 Graduation Design Project

İ. Yiğit Egemen - 180254036
Dr. Yılmaz Kemal Yüce

Dr. Özge Öztimur Karadağ
ALKU

CONTENTS

01

PROBLEM

Definition of the problem.

COMPUTERS

In the standard ergonomics and designs of a computer, there is a keyboard and a mouse interface to control it.

THE PROBLEM

However, not everyone have the physical ability to use those interfaces.

People with;
- Tetra-amelia Syndrome,
- Amputations,
- Birth defects that cause physical disability,
- Permanent paralysis

do not have the ability to use a keyboard or a mouse.

02

PURPOSE

Our purpose with this project.

PURPOSE

Our purpose with this project is to provide an interface to control the cursor in the operating system, in real-time, that does not require any hand ability or movement below the head.

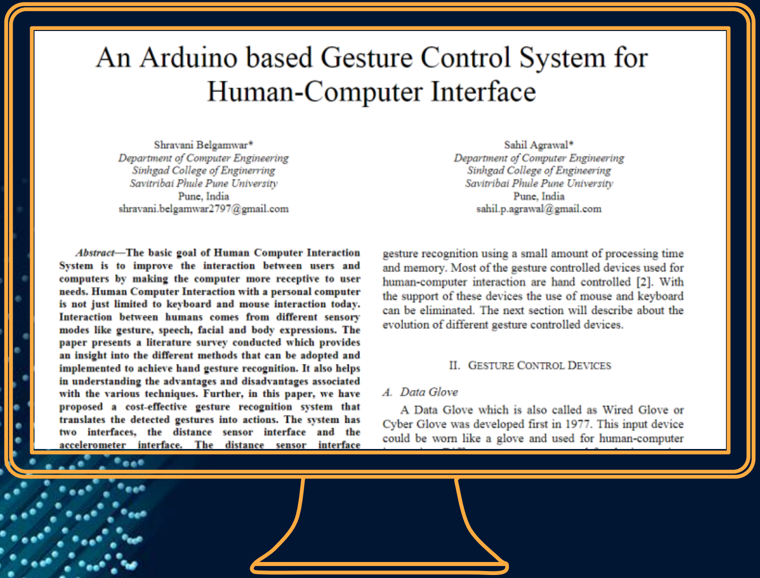
03

LITERATURE

Some background and
literature research.

LITERATURE RESEARCH

An Arduino based Gesture Control System for Human-Computer Interface (2018)



An Arduino based Gesture Control System for Human-Computer Interface

Shravani Belgamwar*
Department of Computer Engineering
Sinhgad College of Engineering
Savitribai Phule Pune University
Pune, India
shravani.belgamwar2797@gmail.com

Sahil Agrawal*
Department of Computer Engineering
Sinhgad College of Engineering
Savitribai Phule Pune University
Pune, India
sahil.p.agrawal@gmail.com

***Abstract*—The basic goal of Human Computer Interaction System is to improve the interaction between users and computers by making the computer more receptive to user needs. Human Computer Interaction with a personal computer is not just limited to keyboard and mouse interaction today. Interaction between humans comes from different sensory modes like gesture, speech, facial and body expressions. The paper presents a literature survey conducted which provides an insight into the different methods that can be adopted and implemented to achieve hand gesture recognition. It also helps in understanding the advantages and disadvantages associated with the various techniques. Further, in this paper, we have proposed a cost-effective gesture recognition system that translates the detected gestures into actions. The system has two interfaces, the distance sensor interface and the accelerometer interface. The distance sensor interface**

gesture recognition using a small amount of processing time and memory. Most of the gesture controlled devices used for human-computer interaction are hand controlled [2]. With the support of these devices the use of mouse and keyboard can be eliminated. The next section will describe about the evolution of different gesture controlled devices.

II. GESTURE CONTROL DEVICES

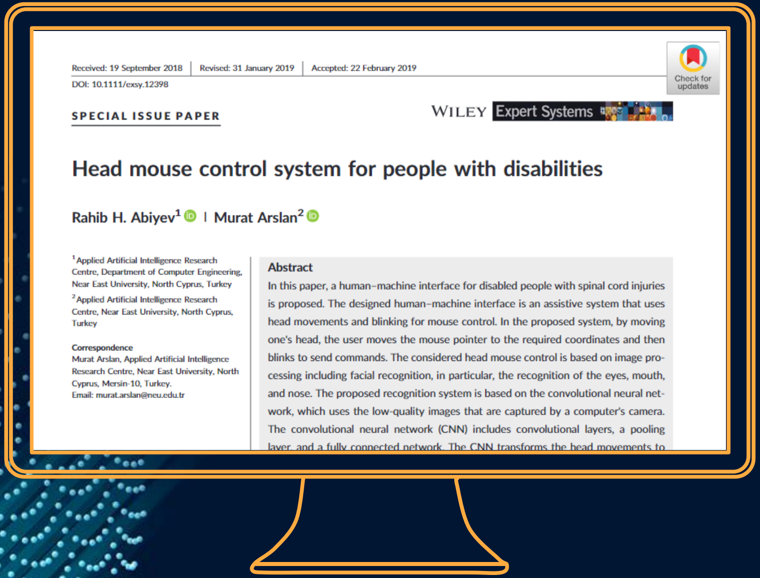
A. Data Glove

A Data Glove which is also called as Wired Glove or Cyber Glove was developed first in 1977. This input device could be worn like a glove and used for human-computer

In this paper, they useD a camera, an accelerometer device, several Arduino microcontrollers and Ultrasonic Distance Sensors to capture body activity. However, this involves the usage of hands.

LITERATURE RESEARCH

Head mouse control system for people with disabilities (2019)



Received: 19 September 2018 | Revised: 31 January 2019 | Accepted: 22 February 2019
DOI: 10.1111/exsy.12398

SPECIAL ISSUE PAPER

WILEY Expert Systems

Head mouse control system for people with disabilities

Rahib H. Abiyev[1] | Murat Arslan[2]

[1]Applied Artificial Intelligence Research Centre, Department of Computer Engineering, Near East University, North Cyprus, Turkey
[2]Applied Artificial Intelligence Research Centre, Near East University, North Cyprus, Turkey

Correspondence
Murat Arslan, Applied Artificial Intelligence Research Centre, Near East University, North Cyprus, Mersin-10, Turkey.
Email: murat.arslan@neu.edu.tr

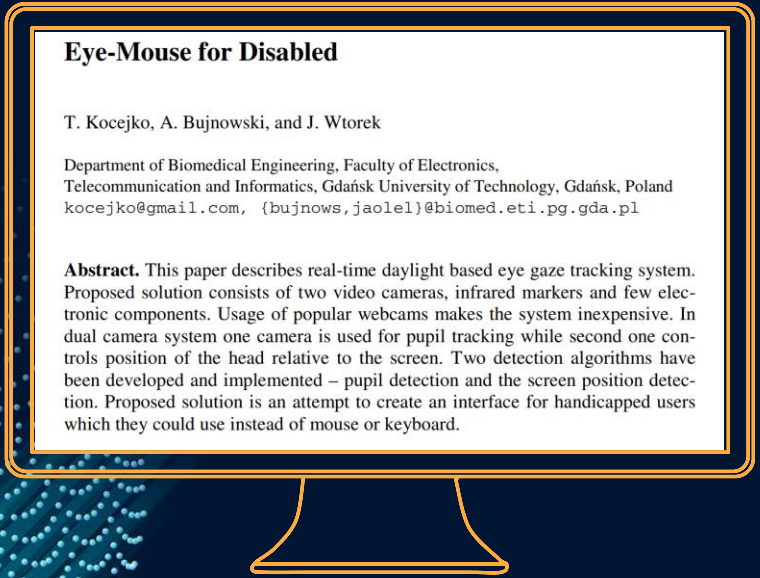
Abstract
In this paper, a human–machine interface for disabled people with spinal cord injuries is proposed. The designed human–machine interface is an assistive system that uses head movements and blinking for mouse control. In the proposed system, by moving one's head, the user moves the mouse pointer to the required coordinates and then blinks to send commands. The considered head mouse control is based on image processing including facial recognition, in particular, the recognition of the eyes, mouth, and nose. The proposed recognition system is based on the convolutional neural network, which uses the low-quality images that are captured by a computer's camera. The convolutional neural network (CNN) includes convolutional layers, a pooling layer, and a fully connected network. The CNN transforms the head movements to

In this paper, they trained a custom model using Convolutional Neural Network (CNN), which finds the face and eyes at the same time. And they used both the head movements and eye blinks from camera to control the mouse.

They did not mention any required processing power or the processing time.

LITERATURE RESEARCH

Eye-Mouse for Disabled (2009)



Eye-Mouse for Disabled

T. Kocejko, A. Bujnowski, and J. Wtorek

Department of Biomedical Engineering, Faculty of Electronics,
Telecommunication and Informatics, Gdańsk University of Technology, Gdańsk, Poland
kocejko@gmail.com, {bujnows,jaolel}@biomed.eti.pg.gda.pl

Abstract. This paper describes real-time daylight based eye gaze tracking system. Proposed solution consists of two video cameras, infrared markers and few electronic components. Usage of popular webcams makes the system inexpensive. In dual camera system one camera is used for pupil tracking while second one controls position of the head relative to the screen. Two detection algorithms have been developed and implemented – pupil detection and the screen position detection. Proposed solution is an attempt to create an interface for handicapped users which they could use instead of mouse or keyboard.

In this paper, they used one camera on eyeglasses to capture the screen (which has IR markers on the corners), and one camera to capture the eye pupil to control the mouse.

04

ANALYSIS

Analysis stage.

ANALYSIS

We aim to have the following;

FEATURES

- Visual interface to show what is captured
- Support the most important mouse functions
  - Left-click
  - Right-click
  - Double left-click
- Precise cursor movement
- Consistent performance among different face/eye types
- Consistent performance in different environments

SPECIFICATIONS

- Low processing time
- Easy to setup
- Low cost

05

DESIGN

The design and the architecture.

DESIGN

With a facing camera attached to the monitor, we captured the eye blinks using methods that involve machine learning methods. This way, we adapted to different type of environments and human ethnics.

Additionally, we attached a small wireless motion sensor to the user’s head, which captured the head movements precisely to move the cursor.

DESIGN

The user is able to move the cursor with small head movements and click using eye blinks.

Combination of capturing eye blinks using camera and head movements using a motion sensor, resulted in a low cost, but accurate and easy to setup product.

MetaMotionRL

MetaMotionRL has an accelerometer inside, using Bosch’s BMI160 motion sensor.

HEAD MOVEMENT

An accelerometer can gauge the orientation of a stationary item in relation to Earth's surface, thus it is viable for our purpose.

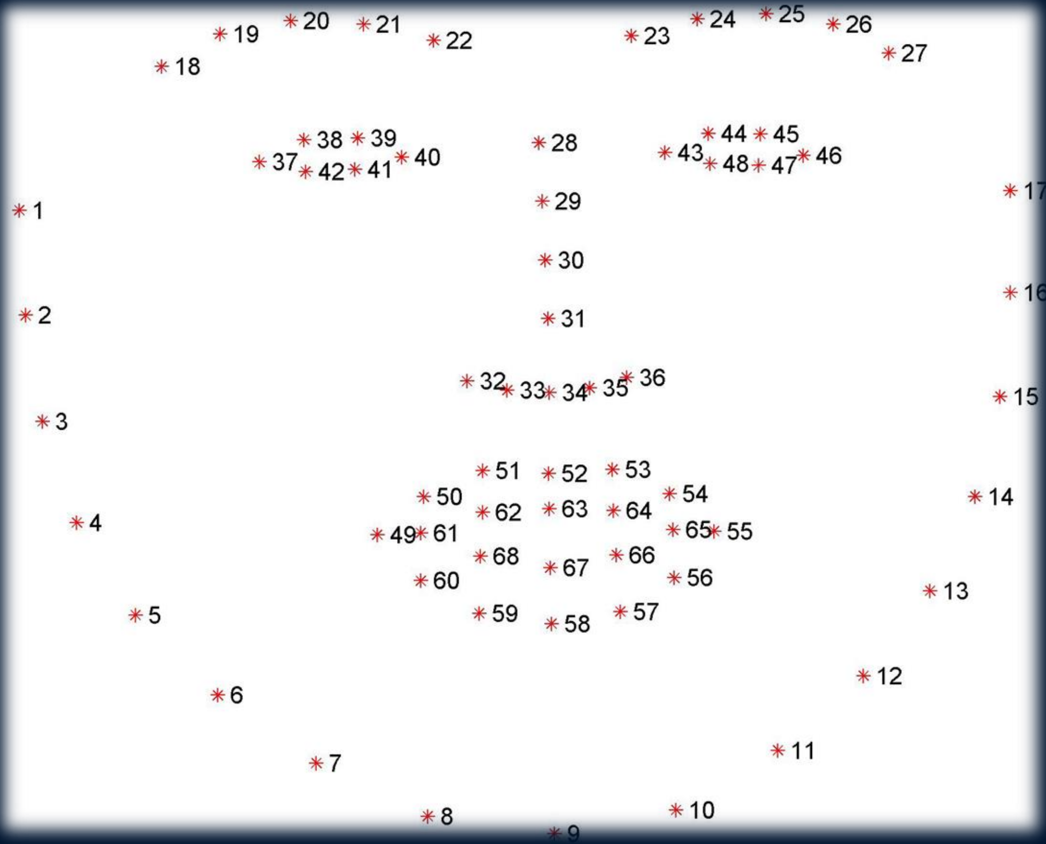
The best location for the motion sensor to be in the face is the forehead, which is directly affected by any head movement that is useful to move the cursor.

HEAD MOVEMENT

Thus, with the accelerometer, we successfully captured the angle of the head. After processing the incoming data, we used the first values of the accelerometer as our starting point at the start of our application. To put it simply, we calibrate the sensor at launch.

HEAD MOVEMENT

Then, we have 2 variables to adjust the movement: sensor deadzone and sensitivity. The deadzone is a zone at the center which is ignored. This is required as it is near impossible for the user to center their head precisely each time they want to stop the movement of the cursor.



Dlib

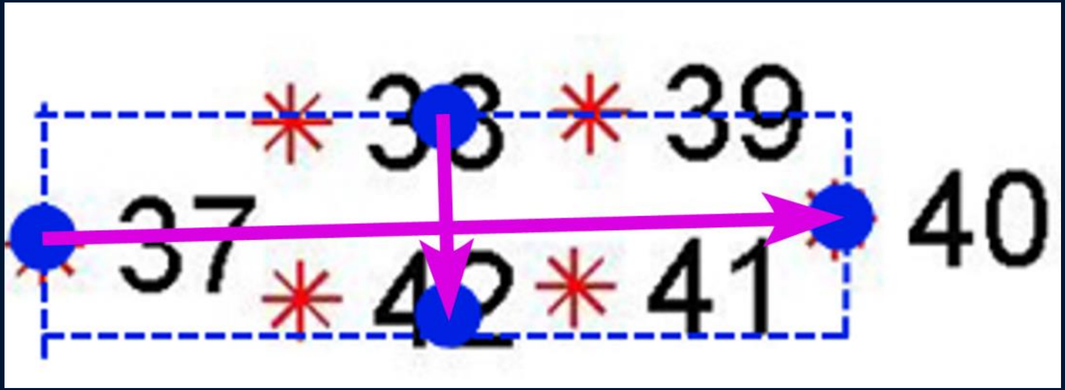
Dlib’s face shape predictor creates 68 face landmarks which gives us the ability to use these for useful purposes.

BLINKS

We can use the landmarks 37, 38, 39, 40, 41, 42 for the right eye, and 43, 44, 45, 46, 47, 58 for the left eye. 37, 40, 43 and 46 are edge points, however, the rest are points that are not centered. So we can calculate and consider their midpoints. The result will give us the horizontal and vertical borders, which can be connected to acquire a rectangle.

BLINKS

Then, if we calculate the Euclidean distance between opposing edges, and compare them, we can achieve the ratio of the closeness of the eye, no matter how small/big or far the eye is.



BLINKS

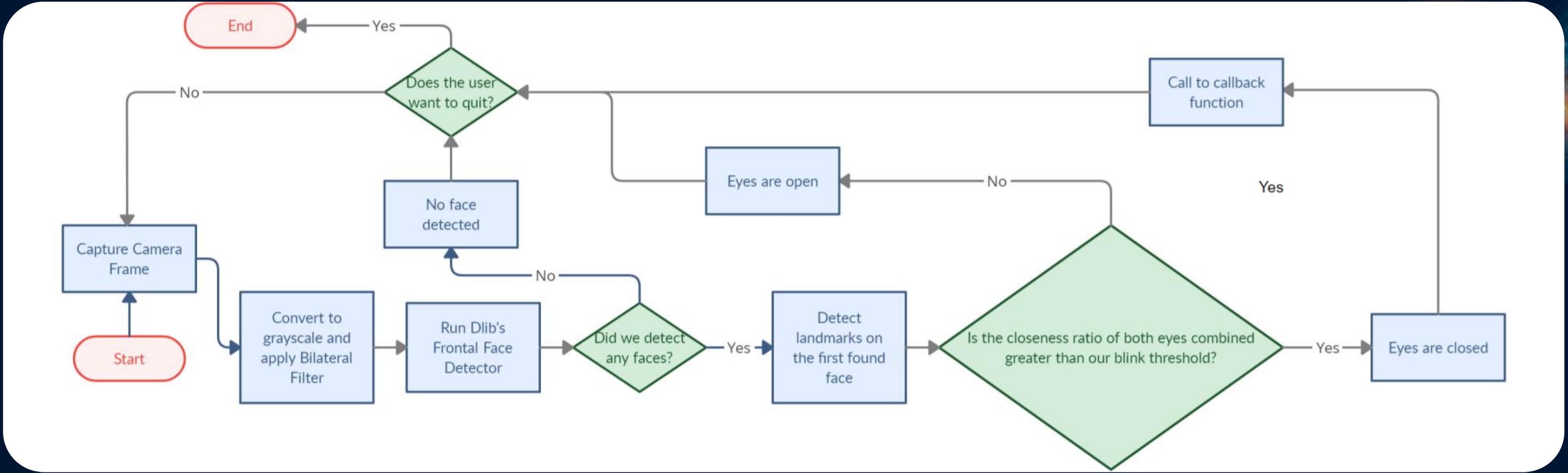
According to scientific papers, a voluntary blink lasts around 134 milliseconds, and the change in blink periods averages only ± 2.23 ms. Thus, we have decided to use the blink threshold as 135 milliseconds in our project.

06

MODEL

Models/diagrams of our model.

FLOWCHART



End
Yes
Does the user want to quit?
No
Call to callback function
Eyes are open
No
Yes
No face detected
Capture Camera Frame
No
Start
Convert to grayscale and apply Bilateral Filter
Run Dlib's Frontal Face Detector
Did we detect any faces?
Yes
Detect landmarks on the first found face
Is the closeness ratio of both eyes combined greater than our blink threshold?
Yes
Eyes are closed

07

MATERIALS

Materials and tools used.

MATERIALS & TOOLS

Software

- Python
- OpenCV
- Dlib
- Matploblib

Hardware

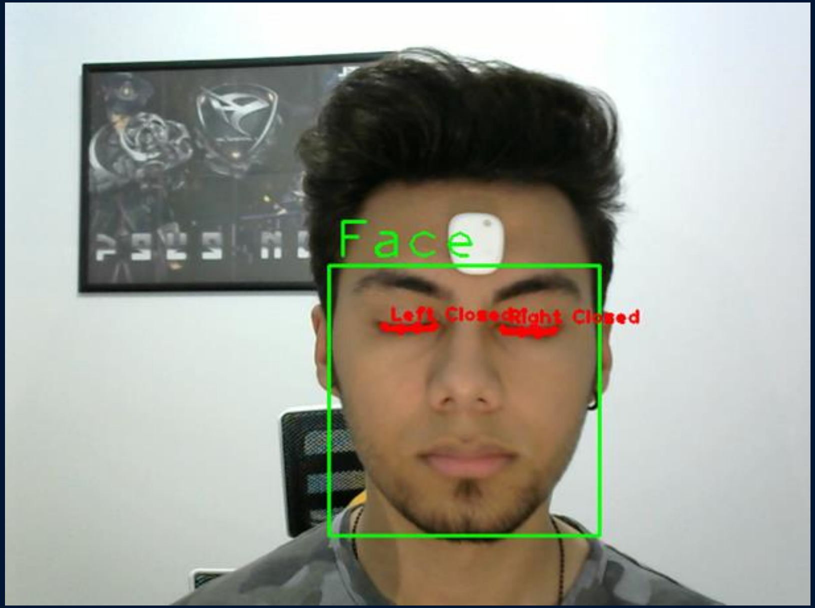
- Computer
  - Intel i7-9750H 4.00 GHz CPU
- Camera
  - Logitech C505 Camera (640x480)
- Accelerometer Sensor
  - MetaMotionRL

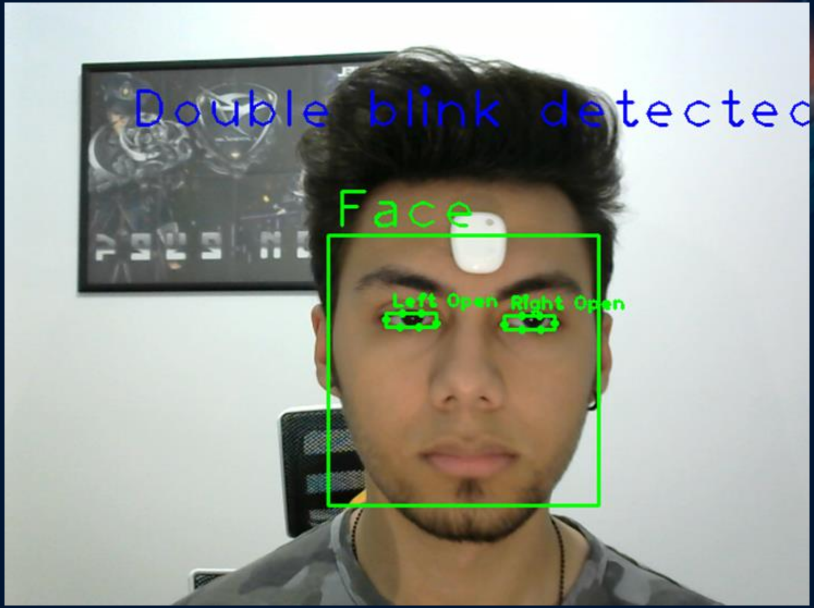
08

DEMO

A demo of the blink detection prototype.

DEMO


Face
Left Closed
Right Closed

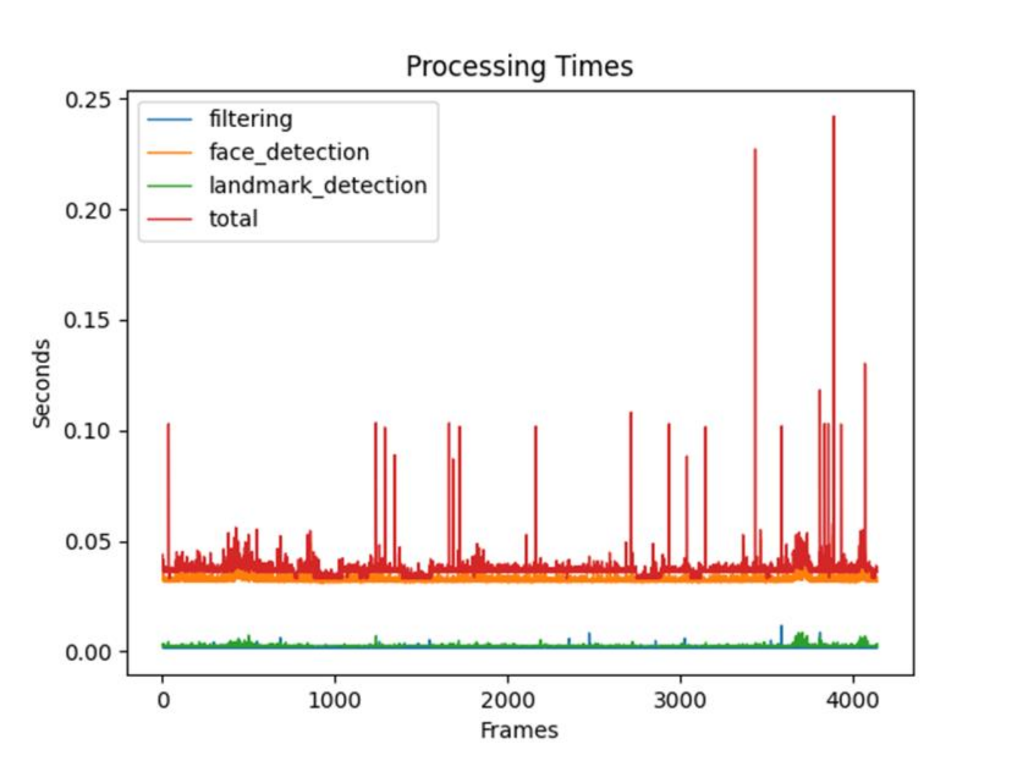

Double blink detecte
Face
Left Open
Right Open

09

STATISTICS

Some statistics.

STATISTICS


Processing Times
filtering
face_detection
landmark_detection
total
0.25
0.20
0.15
0.10
0.05
0.00
Seconds
0
1000
2000
3000
4000
Frames

STATISTICS

	Filtering	Face Detection	Landmarks Detection	Deciding Eye Status and Other Processes	Combined
Avg. Processing Time (seconds)	0.0017	0.032	0.002	0.0013	0.037

Captured for 5 minutes on Intel i7-9750H 4.00 GHz CPU with Logitech C505 camera on resolution 640x480. According to our total processing time, we can support around 27 frames per second.

STATISTICS

	False Positive	False Negative	True Positive	True Negative
Frame Count	37	41	363	559
Total Frame Count	400	600	400	600
Percentage	9,25%	6,84%	90,75%	93,16%

Looking at the contingency table, we have a precision value of 90,75% and an accuracy value of 92,20% while detecting the blinks.

10

REFERENCES

Sources that have been used.

REFERENCES

- M. J. P. R. Dutta, «Tetra Amelia Syndrome - A Case Report,» 2012.
- J. Dormandy, L. Heeck ve S. Vig, «Major amputations: clinical patterns and predictors.,» 1999.
- Carmona ve H. Richard, «The global challenges of birth defects and disabilities.,» 2005.
- W. J. Meggs, «Permanent Paralysis at Sites of Dermal Exposure to Chlorpyrifos,» pp. 883-886, 2003.
- S. Belgamwar ve S. Agrawal, «An Arduino Based Gesture Control System for Human-Computer Interface,» pp. 1-3, 2018.
- M. Arslan ve R. Abiyev, «Head mouse control system for people with disabilities,» Expert Systems, cilt 37, no. 1, pp. 1-14, 2020.
- T. Kocejko, A. Bujnowski ve J. Wtorek, «Eye-Mouse for Disabled,» Advances in Intelligent and Soft Computing, 2009.
- B. Sensortec, «BMI160 DataSheet,» Bosch, 2020.
- Kwon ve Kyung-Ah, «High-speed camera characterization of voluntary eye blinking kinematics.,» Journal of the Royal Society Interface, 2013.
- Kassem, S. Iris ve C. Evinger, «Asymmetry of blinking.,» Investigative ophthalmology & visual science, cilt 47, no. 1, pp. 195-201, 2006.

THANKS.

Yiğit Egemen – 180254036
Dr. Yılmaz Kemal Yüce

Dr. Özge Öztimur Karadağ

Alanya Alaaddin Keykubat University