

SISTEMA DE CONTROL OCULAR PARA UNA SILLA DE RUEDAS  
MOTORIZADA

CARLOS EDUARDO DÍAZ TORRES

UNIVERSIDAD PEDAGÓGICA NACIONAL  
LICENCIATURA EN ELECTRÓNICA  
FACULTAD CIENCIA Y TECNOLOGÍA

BOGOTÁ  
2018


SISTEMA DE CONTROL OCULAR PARA UNA SILLA DE RUEDAS  
MOTORIZADA

Trabajo de grado para optar por el título de Licenciado en Electrónica

CARLOS EDUARDO DÍAZ TORRES  
2013103019

DIRECTOR DE PROYECTO DE GRADO  
DIEGO ACERO


UNIVERSIDAD PEDAGÓGICA NACIONAL  
LICENCIATURA EN ELECTRÓNICA  
FACULTAD CIENCIA Y TECNOLOGÍA  
BOGOTÁ  
2018

 UNIVERSIDAD PEDAGÓGICA NACIONAL <i>Formando al profesional</i>	<b>FORMATO</b>  <b>RESUMEN ANALÍTICO EN EDUCACIÓN - RAE</b>	
Código: FOR020GIB	Versión: 01	
Fecha de Aprobación: 10-10-2012	Página 3 de 63	

1. Información General	
<b>Tipo de documento</b>	Trabajo de grado
<b>Acceso al documento</b>	Biblioteca Central
<b>Título del documento</b>	Sistema de control ocular para una silla de ruedas motorizada
<b>Autor(es)</b>	Díaz Torres, Carlos Eduardo
<b>Director</b>	Acero, Diego Mauricio
<b>Publicación</b>	Bogotá. Universidad Pedagógica Nacional, 2018, 63 p.
<b>Unidad Patrocinante</b>	Universidad Pedagógica Nacional.
<b>Palabras Claves</b>	SISTEMA DE CONTROL, ESTRABISMO, VISIÓN ARTIFICIAL, MOVIMIENTO OCULAR, PREDICCIÓN DE MARCAS FACIALES, GUIÑOS, PARPADEOS

2. Descripción
<p>En este trabajo de grado, se desea elaborar un sistema de control ocular el cual permita tanto a una persona que padezca estrabismo como a una que no, controlar el movimiento de una silla de ruedas motorizada.</p> <p>Se muestra, por lo tanto, el desarrollo de varios algoritmos de visión artificial por medio de los cuales sea posible indicar a una silla de ruedas motorizada las cuatro órdenes básicas (Avanzar, Girar a la derecha, Girar a la izquierda, Frenar).</p> <p>Se presentan algunas pruebas con estos algoritmos, realizando el análisis de su viabilidad, para su posterior utilización o descarte. Con base en ello, se describe todo el desarrollo del sistema de control en su versión final, explicando su estructura, implementación de algoritmos propios y funciones propias, se incluyen descripción y análisis de pruebas de calibración y funcionamiento realizadas por una persona con estrabismo.</p>

3. Fuentes
<p>Silla de ruedas Inteligente – Moviliza a personas Paraplégicas  <a href="https://www.youtube.com/watch?v=Y27_oqvq-ws">https://www.youtube.com/watch?v=Y27_oqvq-ws</a></p> <p>Jóvenes pereiranos inventaron silla de ruedas que se maneja con los ojos - Enero 29 de 2013  <a href="https://www.youtube.com/watch?v=8HkSNpi3nWE">https://www.youtube.com/watch?v=8HkSNpi3nWE</a></p> <p>Silla de Ruedas Controlada por Músculos Oculares - Canal 40  <a href="https://www.youtube.com/watch?v=SwRJv9kPqrl">https://www.youtube.com/watch?v=SwRJv9kPqrl</a></p> <p>En Colombia inventan silla de ruedas que se mueve con los ojos  <a href="https://espanol.upi.com/Curiosidades/2006/09/08/En-Colombia-inventan-silla-de-ruedas-que-se-mueve-con-los-ojos/37511157753940/">https://espanol.upi.com/Curiosidades/2006/09/08/En-Colombia-inventan-silla-de-ruedas-que-se-mueve-con-los-ojos/37511157753940/</a></p>

 UNIVERSIDAD PEDAGÓGICA NACIONAL <i>Construyendo la educación</i>	<b>FORMATO</b>	
<b>Código: FOR020GIB</b>		<b>Versión: 01</b>
<b>Fecha de Aprobación: 10-10-2012</b>		<b>Página 4 de 63</b>

Uso del movimiento de los ojos para desplazarse en silla de ruedas  
<http://ingenierias.utp.edu.co/ingenieria-electronica/trabajos-de-grado.html>

My blog about Making, Coding, Building things  
<https://zerozeroonezeroonezeroonezero.wordpress.com>

Estrabismo  
<http://fon.org.co/estrabismo>

Visión Artificial  
[http://www.academia.edu/6084688/VISI%C3%93N\\_ARTIFICIAL](http://www.academia.edu/6084688/VISI%C3%93N_ARTIFICIAL)

OpenCV-About  
<https://opencv.org/about.html>

Dlib C++ Library  
<http://dlib.net/>

Object Detection in Computer Vision  
<https://www.mathworks.com/discovery/object-detection.html>

An Introduction to CNN based Object Detection  
[https://blog.csdn.net/joshua\\_1988/article/details/73252738](https://blog.csdn.net/joshua_1988/article/details/73252738)

Deep Learning  
<https://www.investopedia.com/terms/d/deep-learning.asp>

What is Deep Learning?  
<https://www.mathworks.com/discovery/deep-learning.html>

Train your own OpenCV Haar classifier  
<http://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html>

Creating your own Haar Cascade OpenCV Python Tutorial  
<https://pythonprogramming.net/haar-cascade-object-detection-python-opencv-tutorial/>

What Is Face Recongition  
<https://www.facefirst.com/face-recognition-glossary/what-is-face-recognition>


Objective C—Image Face Detection  
<https://medium.com/@klin505/objective-c-image-face-detection-1f634215311c>

Real-Time Face Pose Estimation  
<http://blog.dlib.net/2014/08/real-time-face-pose-estimation.html>

Eye blink detection with OpenCV, Python, and dlib  
<https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib>

<https://edoras.sdsu.edu/doc/matlab/toolbox/images/hsvcone.gif>

Assignment of genes coding for brown eye colour and brown hair colour (HCL3) on chromosome 15q.  
<https://www.ncbi.nlm.nih.gov/pubmed/8875191>

 UNIVERSIDAD PEDAGÓGICA NACIONAL <i>Construyendo la educación</i>	<b>FORMATO</b>	
<b>Código: FOR020GIB</b>		<b>Versión: 01</b>
<b>Fecha de Aprobación: 10-10-2012</b>		<b>Página 5 de 63</b>

Estrabismo y Ambliopía

<https://www.pediatruiintegral.es/numeros-anteriores/publicacion-2013-09/estrabismo-y-ambliopia/>

¿Qué es el estrés visual?

[https://www.clarin.com/buena-vida/estres-vista-astenopia-prevencion\\_0\\_S1Jedzjqv7g.html](https://www.clarin.com/buena-vida/estres-vista-astenopia-prevencion_0_S1Jedzjqv7g.html)

Estrabismo

<https://areaoftalmologica.com/enfermedades-oculares/estrabismo/>

Facial Point Annotations

<https://ibug.doc.ic.ac.uk/resources/facial-point-annotations>

What does top, left, right and bottom mean in Android Rect object

<https://stackoverflow.com/questions/22589322/what-does-top-left-right-and-bottom-mean-in-android-rect-object>

Soukupova, T. Cech, Jan. (2016) *Real-Time Eye Blink Detection using Facial Landmarks*. Rimske Toplice, Eslovenia

<http://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf>

Benoit Courti takes black and white to another level

<http://www.artdiscover.com/en/news/benoit-court-takes-black-and-white-to-another-level/290>

[http://1.bp.blogspot.com/-jl\\_oPqMVOyE/VQrjD2SKyHI/AAAAAAAAALE/5kPCBKBxBvk/s1600/as.jpg](http://1.bp.blogspot.com/-jl_oPqMVOyE/VQrjD2SKyHI/AAAAAAAAALE/5kPCBKBxBvk/s1600/as.jpg)

<http://www.abc.es/media/recreo/2016/06/21/beautiful-eyes-9-kl3--620x349@abc.jpg>

<https://www.gettyimages.es/fotos/boca-abierta>

El Inglés también es el idioma de Internet

<https://www.schola.es/blog/el-ingles-tambien-es-el-idioma-de-internet/>

Oostendorp, N. Oliver, A. Scott, K. DeMaagd, K. (2012) *Practical Computer Vision with SimpleCV*, Estados Unidos, O'Reilly Media


Bradski, G. Kaehler, A. (2008) *Learning OpenCV*, Estados Unidos, O'Reilly Media

#### 4. Contenidos

Se dispone de una primera parte la cual contiene la introducción del proyecto, la descripción del problema que dio origen al mismo, justificación, objetivos, limitaciones y alcances y bases teóricas.

El proyecto cuenta también con un apartado de desarrollo, en el cual se explica el funcionamiento e implementación de algoritmos de visión artificial que son pertinentes para el proyecto. Asimismo, se analiza la viabilidad de su implementación en el desarrollo del sistema de control, determinando si se utilizan o se descartan.

En un último apartado de realización e implementación, se describen los algoritmos utilizados para la versión final del sistema; del mismo modo se describe el funcionamiento e implementación de algoritmos propios y funciones propias, una ilustración de la estructura de la que consta el sistema, se realiza el cálculo de tiempos de los procesos, y análisis y descripción de pruebas de calibración y funcionamiento del sistema con una persona sin estrabismo (el autor) y una persona que padece la condición. Por último, se plantean las conclusiones del proyecto.

 UNIVERSIDAD PEDAGÓGICA NACIONAL <i>Formación de líderes</i>	<b>FORMATO</b>	
<b>Código: FOR020GIB</b>		<b>Versión: 01</b>
<b>Fecha de Aprobación: 10-10-2012</b>		<b>Página 6 de 63</b>

5. Metodología
<p>Se desarrolla un sistema de visión artificial por medio del cual se da solución al problema planteado en un principio. El sistema cumple con lo siguiente:</p> <ul style="list-style-type: none"> <li>- Se realiza una descripción de la implementación y un análisis de los algoritmos pertinentes para llevar a cabo el sistema de control.</li> <li>- Después de evaluar la viabilidad de los diferentes algoritmos, se seleccionan los adecuados para realizar el sistema de control; el más importante de ellos, consiste en un sistema de predicción de marcas faciales, mediante el cual se realiza un reconocimiento de ojos y boca, ya que por medio de los ojos la persona brinda las cuatro órdenes básicas de movimiento (Arranca, Girar a la derecha, Girar a la izquierda, Frenar) a una silla de ruedas motorizada, y por medio de la boca finaliza el programa.</li> <li>- Se realizan los algoritmos y funciones que permiten indicar al sistema las cuatro órdenes mencionadas, realizadas por medio de patrones de apertura de ojos, guiños y parpadeos; y del mismo modo se implementa un algoritmo para llevar a cabo la finalización del programa, que se lleva a cabo mediante la ejecución de un patrón determinado de aperturas de boca.</li> </ul>

6. Conclusiones
<p>Las conclusiones obtenidas en el trabajo fueron las siguientes:</p> <ul style="list-style-type: none"> <li>• Si bien el programa podría funcionar para otra persona (Con los mismos valores proporcionados por el proceso de calibración para el autor), es necesario realizar una calibración para cada usuario, ya que como bien se dijo el valor de EAR es diferente para cada persona.</li> <li>• No es conveniente exportar el sistema a una Raspberry Pi® 3 o anterior, ya que se presentó un lag de aproximadamente 4 segundos, lo cual no es útil para sistemas en tiempo real. Sin embargo, debido a las constantes mejoras de la Raspberry Pi en cuanto a velocidad y almacenamiento, no se descarta que una futura generación de este hardware esté en la capacidad de hospedar y correr el programa.</li> <li>• Para aplicaciones posteriores, se podría acoplar el sistema a una silla de ruedas eléctrica, ya que el proyecto se realizó para este fin.</li> <li>• Para la implementación del sistema, es necesario contar con buenas condiciones de luminosidad, ya sea natural o artificial con el fin de evitar fallos de lectura.</li> </ul>

<b>Elaborado por:</b>	Díaz Torres, Carlos Eduardo
<b>Revisado por:</b>	Acero, Diego Mauricio

<b>Fecha de elaboración del Resumen:</b>	07	06	2018
--	----	----	------

## **Tabla de Contenidos**

<b>Introducción.....</b>	<b>9</b>
<b>Planteamiento y formulación del problema.....</b>	<b>10</b>
<b>Justificación.....</b>	<b>10</b>
<b>Antecedentes.....</b>	<b>11</b>
<b>Limitaciones, alcances y proyección.....</b>	<b>12</b>
<b>Objetivos</b>	
<b>Objetivo General.....</b>	<b>13</b>
<b>Objetivos Específicos.....</b>	<b>13</b>
<b>Marco Teórico.....</b>	<b>14</b>
<b>Estrabismo.....</b>	<b>14</b>
<b>Visión Artificial.....</b>	<b>14</b>
<b>OpenCV.....</b>	<b>16</b>
<b>Dlib.....</b>	<b>16</b>
<b>Desarrollo.....</b>	<b>17</b>
<b>Object Detection.....</b>	<b>17</b>
<b>Deep Learning.....</b>	<b>18</b>
<b>HAAR Cascade Classifier.....</b>	<b>20</b>
<b>Facial Recognition y predicción de marcas faciales.....</b>	<b>23</b>
<b>Object tracking.....</b>	<b>27</b>
<b>Object tracking por colores.....</b>	<b>27</b>

Object tracking: Motion detection.....	30
Realización e implementación.....	32
Algoritmo predictor de marcas faciales.....	32
Reconocimiento de ojos.....	36
Eye Aspect Ratio.....	37
Guiños.....	45
Finalización del programa.....	49
Cálculo de tiempos.....	53
Pruebas.....	55
Conclusiones.....	60
Bibliografía.....	61



## Introducción:

Se plantea el diseño de un sistema de control ocular pensando en que este permita, tanto a una persona que padezca estrabismo como a una que no, controlar una silla de ruedas mediante movimientos de sus ojos.

En un principio, el sistema no estaba proyectado para personas con estrabismo; pero con la revisión de la bibliografía, se vio que los controles de este tipo utilizan sistema “eye tracking”, los cuales se encargan de obtener la posición del iris respecto al ojo, para así, de este modo, identificar la dirección hacia la cual mira la persona.

Para los videos de pruebas y demostraciones, las personas tenían sus movimientos oculares coordinados<sup>123</sup>. Por ello se piensa en diseñar un sistema que permita a un paciente con estrabismo, proporcionar órdenes de dirección (Derecha, Izquierda, Adelante, Detener) que puedan proporcionarse a una silla de ruedas motorizada.

---

<sup>1</sup> <[https://www.youtube.com/watch?v=Y27\\_oqvq-ws](https://www.youtube.com/watch?v=Y27_oqvq-ws)>

<sup>2</sup> <<https://www.youtube.com/watch?v=8HkSNpi3nWE>>

<sup>3</sup> <<https://www.youtube.com/watch?v=SwRJv9kPqrl>>

**Problema:**

Diseñar un sistema de control para una silla de ruedas que pueda ser controlada por el movimiento ocular de persona con Estrabismo.

¿Cuáles son los mecanismos para determinar los movimientos oculares de una persona con estrabismo?

**Justificación:**

En la actualidad hay una gran cantidad de personas en condición de discapacidad, que requieren de una silla de ruedas para poder movilizarse, depende del nivel de su discapacidad, la persona podrá o no mover la silla; el sistema de control se ha pensado para este último caso, de modo que la persona podrá controlar el movimiento de la silla usando sus ojos.

Se han realizado varios modelos de sillas de ruedas controladas con el movimiento ocular, para ello utilizan una o dos cámaras y del mismo modo, procesan el movimiento de uno o ambos ojos. Las sillas las han utilizado personas cuyos movimientos oculares están coordinados, pero dicho mecanismo no es efectivo en personas que padecen estrabismo, de modo que se busca diseñar un sistema que se pueda implementar en personas con dicho problema visual.

## Antecedentes:

- *Carlos Alberto Millán, Diego Bernal (2006) Universidad del Bosque, Colombia.*

Se desarrolla un sistema, mediante el cual, se pueda manejar con el movimiento de los ojos. La innovación es sin precedentes. <sup>4</sup>Proyecto de grado para optar por el título de Ingeniero Electrónico.

- *Jhoan Sebastián Ferro Toro (2012). Sistema de seguimiento ocular para el control de un robot móvil. Universidad Tecnológica de Pereira.*

Se desarrolla “el prototipo de un dispositivo que le permite a un individuo desplazarse valiéndose del movimiento de sus ojos para darle órdenes a una silla de ruedas”. “La investigación se presenta 2012, tiene el apoyo de la Universidad Tecnológica de Pereira, el Grupo de Investigación Robótica Aplicada (GIRA) y el TecnoParque Sena”<sup>5</sup>. Proyecto de grado para optar por el título de Ingeniero Electrónico.

- *Myrijam Stoetzer, Paul Foltin (2015).*

Dos estudiantes alemanas en un hackaton (un evento para desarrolladores) usando una placa Arduino, una Raspberry Pi, y servomotores para crear una silla de ruedas que puede ser controlada por los ojos.<sup>6</sup>

---

<sup>4</sup> <<http://espanol.upi.com/2006/09/08/En-Colombia-inventan-silla-de-ruedas-que-se-mueve-con-los-ojos/37511157753940/>>

<sup>5</sup> <<http://ingenierias.utp.edu.co/ingenieria-electronica/trabajos-de-grado.html>>

<sup>6</sup> <<https://zerozeroonezeroonezeroonezero.wordpress.com/>>

### **Limitaciones, alcances y proyección:**

El sistema de control debe ser efectivo para personas con o sin estrabismo. La implementación de dicho sistema se realiza en un pc con cámara de 5 MP y 20 fps, procesador AMD A9-9420 a 3 GHz, 8 GB de RAM, 1TB de almacenamiento y arquitectura x64, sistema operativo Ubuntu, y programado en Python haciendo uso de bibliotecas como Dlib y OpenCV, las cuales son ampliamente utilizadas en aplicaciones que requieran procesamiento de imágenes; al ser Python un lenguaje multiplataforma, el sistema se podría exportar a otros hardware como la Raspberry Pi®. En cuanto a la ejecución del sistema, se deben proporcionar cuatro órdenes diferentes: Derecha, Izquierda, Avanzar y Frenar, visualizadas mediante un texto en pantalla en tiempo real, y un historial de órdenes en la ventana de comandos.

## **Objetivos:**

### **OBJETIVOS:**

#### **Objetivo general:**

- Desarrollar un sistema de control ocular para el movimiento de una silla de ruedas motorizada.

#### **Objetivos específicos:**

- Diseñar el sistema de control para la silla de ruedas.
- Desarrollar los protocolos de comunicación entre las cámaras y el PC. (Este no es necesario, ya que la cámara viene incorporada con el pc, dado el caso que fuera externa, como una con conexión vía usb, se muestra de qué manera se modificaría el código para que se utilizarla.)

## **Marco Teórico:**

### **Estrabismo:**

*“La definición científica del estrabismo es la desviación ocular permanente de uno o de los dos ojos, impidiendo la fijación bifoveolar (Se limita la mirada cuando se quiere enfocar un determinado punto del espacio.) y la visión de profundidad. Es un trastorno en el cual los dos ojos no se alinean en la misma dirección y, por lo tanto, no miran al mismo objeto al mismo tiempo. Esta afección se conoce comúnmente como “ojos bizcos”, impidiendo a una persona ejercer labores como arquitecto, cirujano, chef, y hasta ver una película en tercera dimensión.*

*En una visión normal seis músculos diferentes circundan los ojos y trabajan “como un equipo” de manera que ambos ojos se puedan enfocar en el mismo objeto, en alguien con estrabismo, estos músculos no trabajan juntos. Como resultado, un ojo mira a un objeto, mientras el otro se voltea en una dirección diferente y se enfoca sobre otro objeto, cuando esto ocurre, se envían dos imágenes diferentes al cerebro (una desde cada ojo). Esto confunde al cerebro y puede ignorar la imagen proveniente del más débil.*

*Es importante tratar esta enfermedad en el momento de detectar la anomalía, pues si no se corrige, el ojo que no es tenido en cuenta por el cerebro nunca verá bien, causando ambliopía o como común mente le dicen “ojo perezoso”, aunque en algunos casos esta se presenta primero y es la causa del estrabismo”.<sup>7</sup>*

### **Visión Artificial:**

*“Se puede definir la “Visión Artificial” como un campo de la “Inteligencia Artificial” que, mediante la utilización de las técnicas adecuadas, permite la*

---

<sup>7</sup> <<http://fon.org.co/estrabismo/>>

*obtención, procesamiento y análisis de cualquier tipo de información especial obtenida a través de imágenes digitales.*

*La visión artificial la componen un conjunto de procesos destinados a realizar el análisis de imágenes. Estos procesos son: captación de imágenes, memorización de la información, procesado e interpretación de los resultados.*

*Con la visión artificial se pueden:*

- *Automatizar tareas repetitivas de inspección realizadas por operadores.*
- *Realizar controles de calidad de productos que no era posible verificar por métodos tradicionales.*
- *Realizar inspecciones de objetos sin contacto físico.*
- *Realizar la inspección del 100% de la producción (calidad total) a gran velocidad.*
- *Reducir el tiempo de ciclo en procesos automatizados.*
- *Realizar inspecciones en procesos donde existe diversidad de piezas con cambios frecuentes de producción.*

*Las principales aplicaciones de la visión artificial en la industria actual son:*

- *Identificación e inspección de objetos.*
- *Determinación de la posición de los objetos en el espacio.*
- *Establecimiento de relaciones espaciales entre varios objetos (guiado de robots)*
- *Determinación de las coordenadas importantes de un objeto.*
- *Realización de mediciones angulares.*
- *Mediciones tridimensionales.”<sup>8</sup>*

---

<sup>8</sup> < [http://www.academia.edu/6084688/VISI%C3%93N\\_ARTIFICIAL](http://www.academia.edu/6084688/VISI%C3%93N_ARTIFICIAL)>

## **OpenCV:**

*“OpenCV (Open Source Computer Vision Library) es una biblioteca de software de visión abierta y software de aprendizaje automático. Es una biblioteca de código abierto de software de aprendizaje automático y visión por computadora.*

*OpenCV fue construido para proporcionar una infraestructura común para aplicaciones de visión por computadora y para acelerar el uso de la percepción de la máquina en los productos comerciales. Al ser un producto con licencia de BSD (Berkeley Software Distribution), OpenCV facilita a las empresas la utilización y modificación del código.*

*La biblioteca cuenta con más de 2500 algoritmos optimizados, que incluyen un conjunto completo de algoritmos de visión artificial y de aprendizaje automático tanto clásicos como avanzados. Estos algoritmos se pueden usar para detectar y reconocer rostros, identificar objetos, clasificar acciones humanas en videos, rastrear movimientos de la cámara, rastrear objetos en movimiento, extraer modelos 3D de objetos, producir nubes de puntos 3D desde cámaras estéreo, unir imágenes para producir una alta resolución imagen de una escena completa, encontrar imágenes similares de una base de datos de imágenes, eliminar los ojos rojos de las imágenes tomadas con flash, seguir los movimientos oculares, etc. OpenCV tiene un número estimado de descargas que excede los 14 millones. La biblioteca se usa ampliamente en compañías, grupos de investigación y por organismos gubernamentales.”<sup>9</sup>*

## **Dlib:**

*“Dlib es un moderno kit de herramientas de C ++ que contiene algoritmos de aprendizaje automático y herramientas para crear software complejo en C ++ para resolver problemas del mundo real. Se usa tanto en la industria como en la academia*

---

<sup>9</sup> <<https://opencv.org/about.html>>



*en una amplia gama de dominios, incluidos la robótica, los dispositivos integrados, los teléfonos móviles y los grandes entornos informáticos de alto rendimiento. La licencia de código abierto de Dlib le permite usarla en cualquier aplicación, sin costo alguno.”<sup>10</sup>*

## **Desarrollo:**

Para el desarrollo del sistema se tienen en cuenta los siguientes algoritmos:

### **Object Detection**

Los algoritmos de object detection o de detección de objetos por su traducción al español, identifican la presencia de objetos en una imagen dada; señalándolos mediante un rectángulo o cuadrado y en algunas ocasiones, poniendo etiquetas de la clase a la cual pertenecen (Persona, Carro, Rostro, Bicicleta, etc.). Puede haber objetos que pertenezcan a otros como por ejemplo en el caso de rostros a personas o boca a rostro.

*“Object detection es el proceso de encontrar instancias de objetos del mundo real tales como caras, bicicletas y edificios en imágenes o videos. Los algoritmos de detección de objetos generalmente usan características extraídas y algoritmos de aprendizaje para reconocer instancias de una categoría de objeto. Es usado comúnmente en aplicaciones como recuperación de imágenes, seguridad, vigilancia y sistemas ADAS (Advanced Driver Assistance Systems)”<sup>11</sup>*

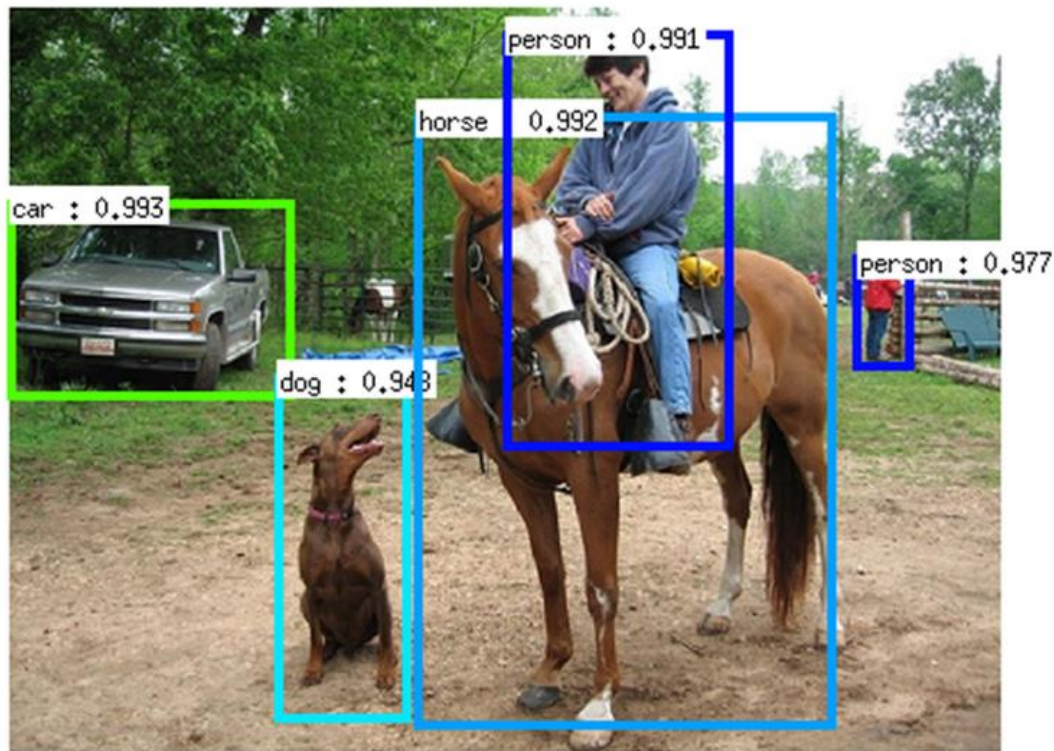
Los sistemas de recuperación de imágenes consisten en, a partir de una imagen dada, encontrar imágenes visualmente similares de una base de datos (De imágenes).

---

<sup>10</sup> <<http://dlib.net/>>

<sup>11</sup> <<https://www.mathworks.com/discovery/object-detection.html>>

En ocasiones, aparte de etiquetarse, se proporciona un número correspondiente al nivel de coincidencia del objeto identificado con la clase a la cual pertenece.



12

*Figura 1. Ejemplo de algoritmo de Object detection*

Los algoritmos de object detection utilizan a su vez modelos pre-entrenados de deep learning por medio de los cuales es posible realizar la clasificación de objetos.

## **Deep Learning**

*“Deep Learning es una función de inteligencia artificial que imita el funcionamiento del cerebro humano en procesamiento de datos y creación de patrones para usarlos en toma de decisiones: El deep learning es un subconjunto*

---

<sup>12</sup> <[https://blog.csdn.net/joshua\\_1988/article/details/73252738](https://blog.csdn.net/joshua_1988/article/details/73252738)>

*del aprendizaje automático en Inteligencia Artificial que tiene redes capaces de aprender sin supervisión a partir de datos sin estructurar ni etiquetar.”<sup>13</sup>*

Como bien se dijo, para el object detection se utilizan modelos pre-entrenados de deep learning; estos modelos se entrenan haciendo uso de redes neuronales (Neural Training)

Debido a que la mayoría de los métodos de Deep learning utilizan redes neuronales, los modelos de deep learning se suelen denominar Deep neural networks.

*“En aprendizaje profundo, un modelo de computadora aprende a realizar tareas de clasificación directamente desde imágenes, texto o sonido. Los modelos de aprendizaje profundo pueden lograr una precisión de vanguardia, que a veces excede el rendimiento a nivel humano. Los modelos se entrenan mediante el uso de un gran conjunto de arquitecturas de redes neuronales y datos etiquetados.*

*La mayoría de los métodos de aprendizaje profundo utilizan arquitecturas de redes neuronales, por lo que los modelos de aprendizaje profundo a menudo se denominan redes neuronales profundas.”<sup>14</sup>*

En cuanto al modo en que se realiza el proceso de entrenamiento o neural training, se hace referencia a un tema bastante extenso que amerita de por sí un estudio aparte, sin embargo, desde una perspectiva general, el proceso se basa en tener una red neuronal compuesto por consiguiente de neuronas artificiales, las cuales son resultado de un modelamiento matemático y en donde el equivalente de las dendritas son señales de entrada.

Si bien existen varias formas de implementar el deep learning, y la más común es utilizar redes neuronales.

---

<sup>13</sup> <<https://www.investopedia.com/terms/d/deep-learning.asp>>

<sup>14</sup> <<https://www.mathworks.com/discovery/deep-learning.html>>

Se puede hacer por tanto la analogía de cerebro-deep learning, neuronas-neuronas artificiales.

## HAAR Cascade Classifier

Es un algoritmo por medio del cual se puede realizar object detection. Este algoritmo involucra un archivo xml. Como resultado de un neural training.

Desde alguna base de datos, o bien de forma manual, (por medio de descarga), se seleccionan varias imágenes las cuales deben contener el objeto a encontrar o detectar. Las mismas reciben el nombre de “imágenes positivas”. De este modo, la precisión del sistema es proporcional al número de imágenes seleccionadas.

Del mismo modo que se realiza con las imágenes positivas, se deben seleccionar varias imágenes negativas, las cuales, por consiguiente, no deben contener el objeto a detectar.

Con base en estas imágenes se realiza un “Cascade Classifier Training” el cual genera los correspondientes archivos .xml. Las imágenes tanto positivas como negativas se convierten a un archivo .txt de la siguiente manera:

```
find ./negative_images -iname "*.jpg" > negatives.txt
```

```
find ./positive_images -iname "*.jpg" > positives.txt
```

15

Donde positive\_images y negative\_images son las carpetas en las que se encuentran las imágenes postivas y negativas respectivamente. Cabe resaltar que se deben recortar las imágenes positivas de acuerdo con el área de interés, que debe corresponder a la ubicación del objeto a detectar.

---

<sup>15</sup> <<http://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html>>

En el caso de que las imágenes no se encuentren descargadas, y se haya hecho su búsqueda por medio de una base de datos, se realizaría un programa similar a este:

```
import urllib.request
import cv2
import numpy as np
import os

def store_raw_images():
    neg_images_link = '//image-net.org/api/text/imagenet.synset.geturls?wni
    neg_image_urls = urllib.request.urlopen(neg_images_link).read().decode(
    pic_num = 1

    if not os.path.exists('neg'):
        os.makedirs('neg')

    for i in neg_image_urls.split('\n'):
        try:
            print(i)
            urllib.request.urlretrieve(i, "neg/"+str(pic_num)+".jpg")
            img = cv2.imread("neg/"+str(pic_num)+".jpg",cv2.IMREAD_GRAYSCALE)
            # should be larger than samples / pos pic (so we can place our
            resized_image = cv2.resize(img, (100, 100))
            cv2.imwrite("neg/"+str(pic_num)+".jpg",resized_image)
            pic_num += 1

        except Exception as e:
            print(str(e))
```

*Figura 2. Ejemplo de Algoritmo HAAR Cascade Classifier (Primera parte)*

```
def create_pos_n_neg():
    for file_type in ['neg']:

        for img in os.listdir(file_type):

            if file_type == 'pos':
                line = file_type+'/'+img+' 1 0 0 50 50\n'
                with open('info.dat','a') as f:
                    f.write(line)
            elif file_type == 'neg':
                line = file_type+'/'+img+'\n'
                with open('bg.txt','a') as f:
                    f.write(line)
```

16

*Figura 3. Ejemplo de Algoritmo HAAR Cascade Classifier (Continuación)*

Para este caso, se recurre a la base de datos image-net.org.

Una vez se tienen ambos archivos con extensión .txt, se genera el archivo .xml. Este es el punto en el que se realiza el “cascade classifier training”. Se indica además un número de muestras positivas, las cuales son resultado de aplicar transformaciones y distorsiones a las imágenes positivas; así como de la combinación de estas con cada una de las imágenes negativas. Para el ejemplo, se indica realizar 1500 muestras. El número de muestras, de igual forma permite mayor precisión, pero el tiempo de compilación será mucho mayor.

```
perl bin/createsamples.pl positives.txt negatives.txt samples 1500\
```

Una vez se tiene el archivo xml, se realiza el algoritmo de detección del objeto. El siguiente código, corresponde a un detector de reloj. La sencillez del código es debido a que no se deben encontrar más objetos y es un reloj en específico.

<sup>16</sup> <<https://pythonprogramming.net/haar-cascade-object-detection-python-opencv-tutorial/>>

```

import numpy as np
import cv2
watch_cascade = cv2.CascadeClassifier('watchcascade10stage.xml')
cap = cv2.VideoCapture(0)

while 1:
    ret, img = cap.read()
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    watches = watch_cascade.detectMultiScale(gray, 50, 50)

    for (x,y,w,h) in watches:
        cv2.rectangle(img,(x,y),(x+w,y+h),(255,255,0),2)

    cv2.imshow('img',img)
    k = cv2.waitKey(30) & 0xff
    if k == 27:
        break

cap.release()
cv2.destroyAllWindows()

```

17

*Figura 4. Ejemplo de Algoritmo HAAR Cascade Classifier con modelos preentrenados*

Los sistemas de object detection se basan en modelos pre-entrenados y el HAAR Cascade Training es un método por medio del cual se pueden entrenar estos modelos.

## **Facial Recognition y predicción de marcas faciales**

“Los sistemas de reconocimiento facial emplean un algoritmo que puede predecir si hay una coincidencia basada en numerosos puntos en la cara del individuo. Mientras que el ojo humano es fácilmente confundido con pelo facial, sombreros u

---

<sup>17</sup> <<https://pythonprogramming.net/haar-cascade-object-detection-python-opencv-tutorial/>>

otros factores, la tecnología de reconocimiento facial es mucho más precisa al encontrar las coincidencias. Esta tecnología es por lo tanto usada frecuentemente para propósitos de seguridad y vigilancia. Por ejemplo, ayudando a las tiendas minoristas a identificar criminales y a los aeropuertos a identificar a las personas en listas de vigilancia de terroristas”<sup>18</sup>

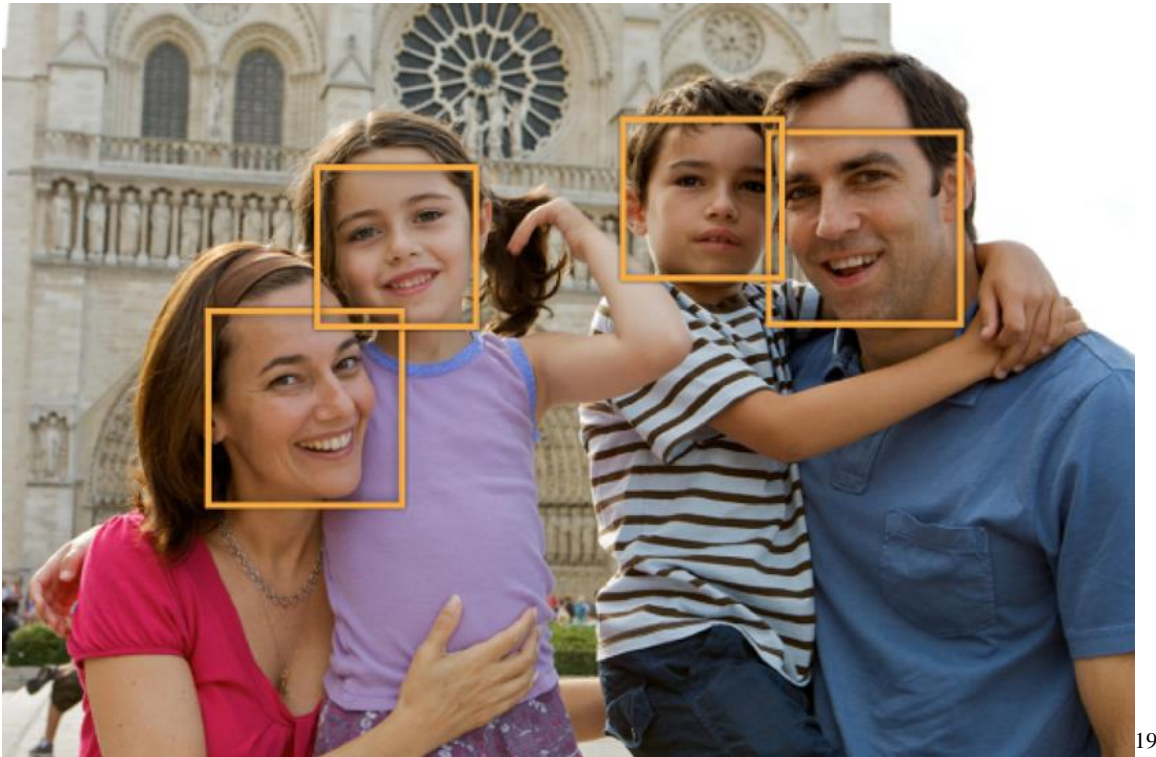
Un sistema de face detection no brinda la información necesaria que se requiere para la realización del sistema, ya que solo indicaría la localización del rostro. Se opta por lo tanto por realizar un sistema que involucre algún algoritmo de predicción de marcas faciales. Estos algoritmos, presentes en los sistemas de reconocimiento facial, describen por medio de una serie de puntos, la ubicación de ojos, boca, y demás rasgos faciales, pudiéndose así obtener información adicional como la apertura de ojos y boca, inclinación del rostro, etc.

Para la implementación del sistema, se necesita obtener los puntos que describen los ojos y no solo saber si hay o no ojos en el frame. Por lo que se deja de lado la realización de los algoritmos de object detection.

---

<sup>18</sup> <<https://www.facefirst.com/face-recognition-glossary/what-is-face-recognition/>>





*Figura 5. Ejemplo de un sistema de detección facial*

---

<sup>19</sup> <<https://medium.com/@klin505/objective-c-image-face-detection-1f634215311c>>



*Figura 6. Ejemplo de un sistema de reconocimiento facial con algoritmo de predicción de marcas faciales (De la biblioteca Dlib c++)*

Se trabaja con base en un algoritmo contador de parpadeos, introducido por Adrian Rosebrock (2017) (Eye blink detection with OpenCV, Python and dlib<sup>20</sup>) el cual utiliza un algoritmo de predicción de marcas faciales de la biblioteca dlib. Este código se modifica y adecúa para que cumpla los requerimientos del sistema de control deseado, haciendo en primer lugar, que no cuente parpadeos involuntarios.

En un principio se piensa realizar un algoritmo de object tracking para realizar el seguimiento de iris y añadirlo al contador de parpadeos.

---

<sup>20</sup> <<http://blog.dlib.net/2014/08/real-time-face-pose-estimation.html>>

<sup>21</sup> <<https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib/>>

Por consiguiente, se implementaron algoritmos object detection del iris para determinar la dirección de la mirada, con base en la localización del iris dentro del ojo y mediante el cálculo de la distancia entre este y los puntos correspondientes a los ojos, los cuales son proporcionados por el algoritmo predictor de marcas faciales.

Cabe aclarar que un algoritmo no se implementó en su totalidad, y otro se descartó, ya que aparte de los inconvenientes propios, una búsqueda bibliográfica indicó lo inconveniente que resultan los movimientos oculares forzados para personas con estrabismo, los cuales pueden generar fatiga visual y empeorar la condición.

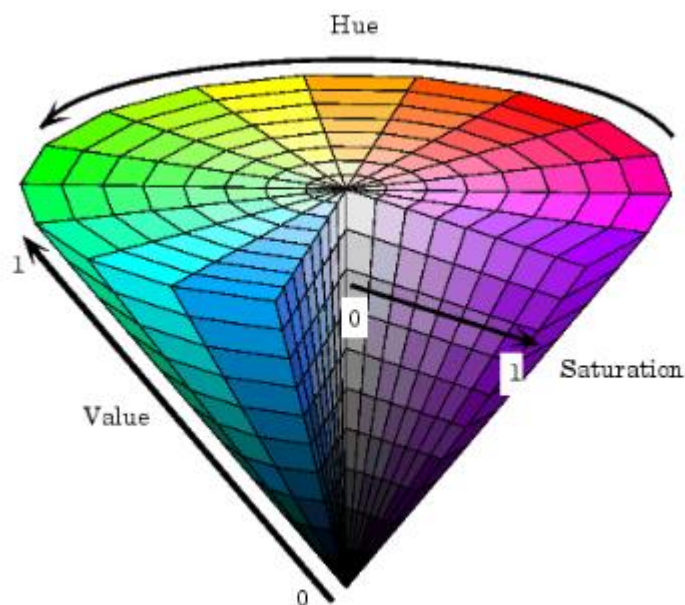
### **Object tracking:**

Se tuvo varias alternativas para realizar un sistema de object tracking:

#### **Object tracking por colores:**

Se implementó un sistema de object tracking de colores, el cual realiza el seguimiento de un objeto de color determinado, este proceso se realiza indicando al sistema el rango de color correspondiente; para este caso, se indica el tipo de color en HSV.

El modelo de color HSV también conocido como HSB, define los colores con base en sus tres propiedades, las cuales son matiz, saturación y luminosidad (Hue, Saturation, Brightness/Value). Para este modelo, se suele utilizar un cono, cuya base representa la los valores de matiz, los cuales varían entre  $0^\circ$  y  $359^\circ$ , correspondiendo un color a cada valor; la saturación corresponde a la distancia al eje oscuridad-luz (A menor saturación, más tonalidad grisácea), y se da en porcentaje, donde 0% corresponde al centro del círculo del cono y 100% a su borde; finalmente, la luminosidad corresponde a la altura del cono, y se da también en porcentaje, donde en 0% se sitúa la punta del cono y se tiene menor luminosidad, y 100% se sitúa en la base del cono, donde habrá mayor luminosidad.



22

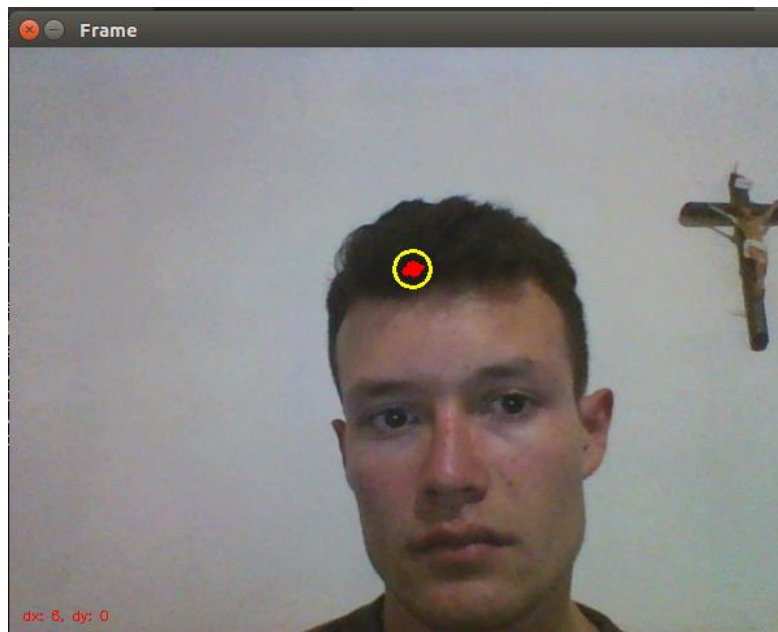
*Figura 7. Modelo de color HSV*

El sistema identifica el objeto del color proporcionado y brinda las coordenadas rectangulares de su posición. Este sistema tuvo dos inconvenientes para su implementación:

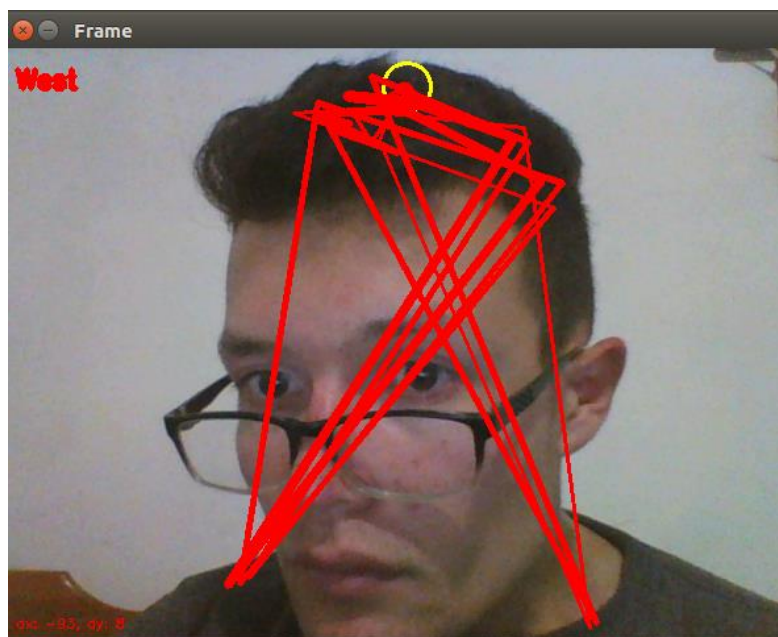
En primer lugar, que el color de iris varía en cada persona, por lo que habría que hacer sistemas para cada color de iris (Negro, Café claro y oscuro, verde, azul, gris, etc). En segundo lugar, si se realiza una generalización realizando el sistema para el color de iris más común<sup>23</sup>, es decir marrón, se confunde con el cabello, cejas e incluso ropa de colores cercanos.

<sup>22</sup> <<https://edoras.sdsu.edu/doc/matlab/toolbox/images/hsvcone.gif>>

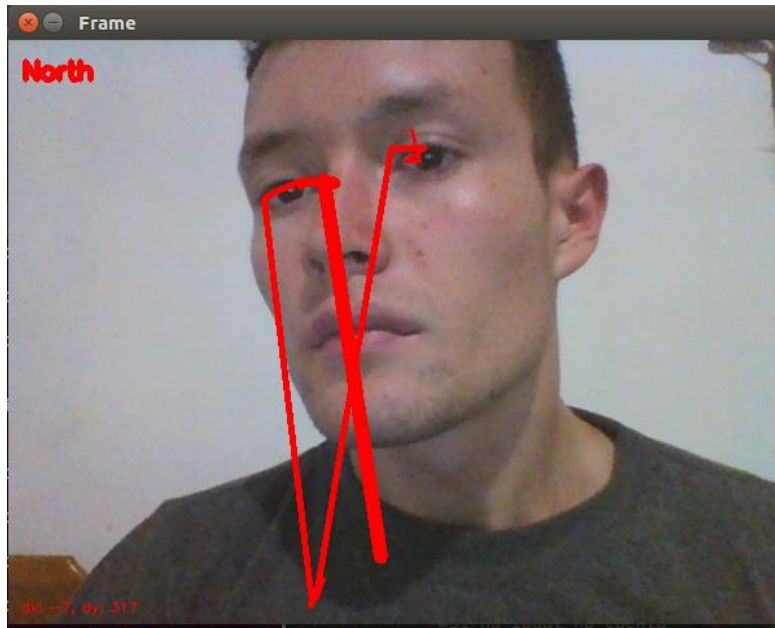
<sup>23</sup> <<https://www.ncbi.nlm.nih.gov/pubmed/8875191>>



*Figura 8. Sistema object tracking para el color marrón*



*Figura 9. Sistema object tracking para el color marrón*



*Figura 10. Sistema object tracking para el color marrón*

Se puede apreciar en la figura 8, que el sistema señala el cabello, en la figura 9, señala el cabello y partes de la ropa y en la figura 10, aunque señala los iris, señala también regiones de la ropa.

Debido a los inconvenientes ya mencionados, se descarta la aplicación de este algoritmo.

### **Object tracking: Motion detection:**

Este sistema es ampliamente utilizado en aplicaciones de vigilancia. Consiste en enfocar una imagen estática, e identificar cambios en el fondo de esta imagen, identificando así el objeto que se está moviendo. El gran inconveniente de este método consiste en que, para la aplicación, la persona debería estar totalmente quieta, y no podría mover nada más que los ojos, no podría mover cejas, ni boca ni la cabeza, ni parpadear para no modificar el fondo, por lo que no se implementa este algoritmo.



Se opta por implementar un algoritmo de object tracking utilizando un modelo pre-entrenado HAAR Cascade Classifier que localice el iris del ojo en el frame para poderlo relacionar con la posición del ojo proporcionada por el sistema de predicción. Este sistema no se termina de implementar, ya que una revisión de la bibliografía indicó lo siguiente:

En un principio se pensó que el sistema, podría determinar la dirección de la mirada del paciente con base en la posición del iris de ambos ojos e indicando cuál es el ojo dominante, para que, de este modo, solo se realicen los cálculos con el movimiento del ojo dominante. Sin embargo, existe una condición conocida como estrabismo alternante, en el cual *“se emplea indistintamente uno u otro ojo para fijar, mientras que el otro se desvía, por lo que no se puede hablar de establecer un ojo dominante.”*<sup>24</sup>

Adicionalmente, se encontró que los movimientos oculares forzados suponen fatiga ocular:

*“La fatiga ocular o astenopia es la respuesta del ojo frente al esfuerzo muscular excesivo durante un largo período de tiempo. En la mayoría de los casos, la fatiga ocular está relacionada con hábitos inadecuados y exceso de “trabajo ocular”. Los factores relacionados con la aparición de cansancio incluyen la concentración visual en un objeto durante largo tiempo [...].”*<sup>25</sup>

*“Factores como el stress, el cansancio o la fatiga visual pueden empeorar de forma transitoria el estrabismo [...].”*<sup>26</sup>

Por ello, se opta por implementar un sistema basado en guiños, el cual no implica los movimientos oculares prolongados que suponen ese esfuerzo muscular

---

<sup>24</sup> <<https://www.pediatrintegral.es/numeros-antiores/publicacion-2013-09/estrabismo-y-ambliopia/>>

<sup>25</sup> <[https://www.clarin.com/buena-vida/estres-vista-astenopia-prevencion\\_0\\_S1Jedzjqv7g.html](https://www.clarin.com/buena-vida/estres-vista-astenopia-prevencion_0_S1Jedzjqv7g.html)>

<sup>26</sup> <<https://areaoftalmologica.com/enfermedades-oculares/estrabismo/>>

excesivo. De este modo se podrá dirigir la mirada en cualquier dirección sin indicar ninguna orden al sistema.

Teniendo en cuenta lo anterior, el sistema se realiza de la siguiente manera:

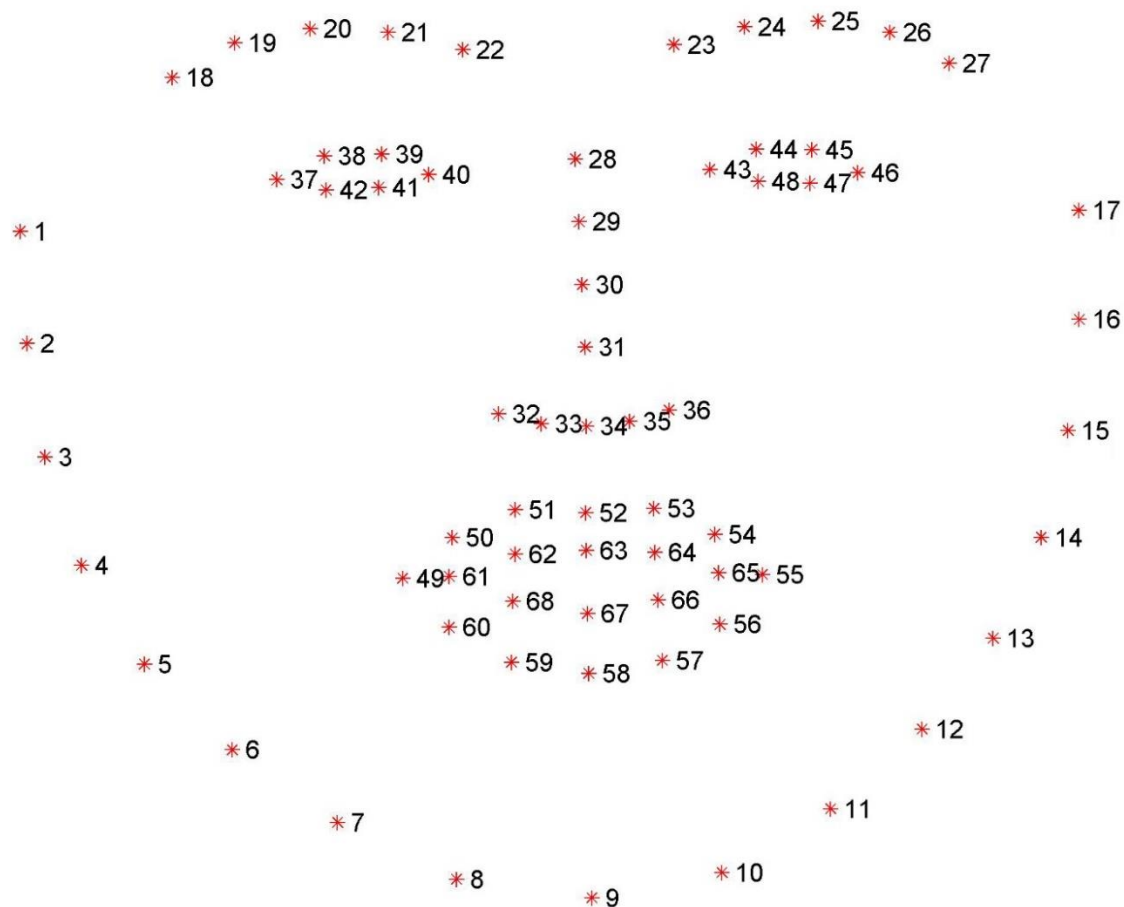
## **Realización final e implementación**

### **Algoritmo predictor de marcas faciales**

Como bien se dijo, existen sistemas tanto de reconocimiento facial como de detección facial. Para el proyecto se utiliza un sistema de predicción de marcas faciales. La ventaja de este por sobre el de detección, consiste en que identifica patrones de puntos que conforman el rostro, de modo que se puede obtener la posición de cada uno de estos, y relacionarlos entre sí. Un sistema de detección facial, debida su sencillez, se limita a indicar si hay o no un rostro en pantalla, señalándolo generalmente mediante un cuadrado, que son los sistemas que vienen implementados en gran variedad de cámaras como las de pc o celular.

El algoritmo de predicción de marcas faciales utilizado detecta una serie de puntos que conforman el patrón de rostro humano, este proceso se realiza mediante un modelo de predicción neuronal, como bien se dijo, el modo en que este proceso se realiza amerita por sí mismo un trabajo de investigación aparte [Modelado de sistemas mediante redes neuronales, Tesis de Maestría, Efraín Alcorta García, Monterrey-México 1992 y Modelos Predictivos Basados En Redes Neuronales Recurrentes De Tiempo Discreto, Tesis Doctoral, Juan Antonio Pérez Ortiz, Alicante-España 2002]. Dentro del algoritmo contador de parpadeos con el que se trabaja como base, se recurre a un algoritmo de predicción de 68 puntos de la librería dlib, el cual es ampliamente utilizado en aplicaciones que requieran reconocimiento facial debido a su alta precisión y eficacia. Este algoritmo predictor involucra reconocimiento de ojos, cejas, nariz, boca, y contorno de rostro.





27

*Figura 11. Distribución de los 68 puntos del algoritmo predictor de marcas faciales de Dlib.*

Dado que se trata de un sistema de predicción, se pueden presentar ciertos errores de lectura en el uso de gafas, por lo que es recomendable tener buenas condiciones de luminosidad y ausencia de gafas.

<sup>27</sup> <<https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/>>



*Figura 12. Algoritmo predictor de marcas faciales utilizando gafas*



*Figura 13. Algoritmo predictor de marcas faciales utilizando gafas*

En la imagen de la figura 12 se puede notar como el sistema predictor sitúa los ojos un poco más arriba del lugar en el que se sitúan, mientras en la figura 13, la ubicación de los ojos coincide.



*Figura 14. Implementación del sistema con condiciones de luminosidad óptimas*



*Figura 15. Implementación del sistema en un ambiente oscuro*

En la figura 14 se puede apreciar la ejecución del sistema en su versión final, donde la predicción de marcas faciales se realiza correctamente; en la figura 15, el programa se ejecuta en un ambiente oscuro, y el sistema de predicción describe el ojo izquierdo como si este se encontrara abierto.

## Reconocimiento de ojos:

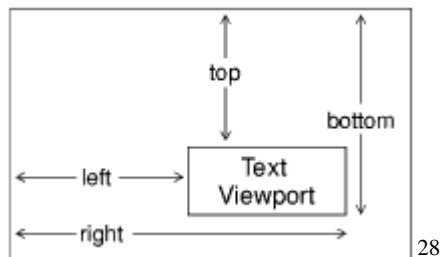
El reconocimiento de los ojos se encuentra dentro del reconocimiento facial, ya que hace parte de este, por lo que solo hace falta identificar los puntos correspondientes que describen ambos ojos: puntos 37 al 42 para el ojo izquierdo y puntos 43 al 48 para el derecho. Los puntos se distribuyen de la siguiente forma:



Una vez se tienen estos puntos, se convierten a un array, al tiempo que se obtienen las coordenadas de cada uno:

```
x=rect.left()  
y=rect.top()
```

En el anterior fragmento de código, se declaran las variables: “x” para obtener la posición en “x” mediante la función `rect.left()`; y “y” para obtener la posición en “y” mediante la función `rect.top()`.



*Figura 16. Descripción de la posición de un objeto mediante top, left, right y bottom*

<sup>28</sup> <<https://stackoverflow.com/questions/22589322/what-does-top-left-right-and-bottom-mean-in-android-rect-object>>

```
landmarks = np.matrix([[p.x,p.y] for p in predictor(frame, rect).parts()])
```

```
ojo_der=landmarks[puntos_derecho]
```

En el fragmento de código anterior, se declara la función landmarks, la cual se encarga de convertir en array los puntos ingresados. La variable p hace referencia a cada uno de los puntos del frame actual; p.x y p.y dado que cada punto tiene coordenadas horizontales y verticales.

En este caso, el array se almacena en la variable ojo\_der.

### **Eye Aspect Ratio:**

Es un término fue introducido por Tereza Soukupová y Jan Čech en su trabajo Real-Time Eye Blink Detection Using Facial Landmarks en 2016<sup>29</sup>

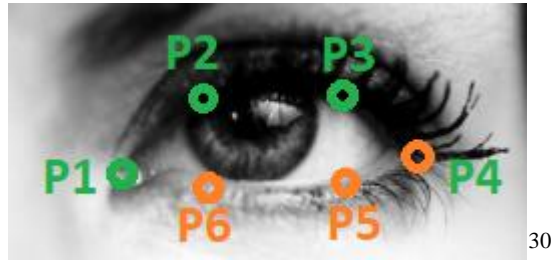
Este término hace referencia a un valor escalar, que varía de manera directamente proporcional a la apertura del ojo. Dado que se tienen seis puntos para describir cada ojo, se calcula el EAR (Eye Aspect Ratio) de los ojos del siguiente modo:

$$EAR = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

La fórmula proporcionada por el trabajo que se acaba de mencionar. Los puntos se distribuyen en el ojo de la siguiente manera:

---

<sup>29</sup> <<http://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf>>



30

*Figura 17. Distribución de los 6 puntos en el ojo*

Se escribe entonces una función la cual recibe el objeto ojo, conformado por seis elementos, que corresponden a seis puntos, y cada uno de ellos descrito por dos valores, correspondientes a su coordenada rectangular, el fragmento de código es el siguiente:

```
def eye_aspect_ratio(eye):
    # (distancia(p2-p6)+distancia(p3-p5))/(2*distancia(p1-p4))

    A = dist.euclidean(eye[1], eye[5])
    B = dist.euclidean(eye[2], eye[4])

    C = dist.euclidean(eye[0], eye[3])

    ear = (A + B) / (2.0 * C)

    return ear
```

La subfunción `dist.euclidean`, halla la distancia euclidiana entre dos puntos; la distancia euclidiana es el valor simple que se obtendría al medir con una regla la distancia entre dos puntos.

Siguiendo la fórmula proporcionada por Tereza Soukupová y Jan Čech el EAR vendría descrito por la suma de la distancia entre los puntos 1-5 y 2-4, dividido entre dos veces la distancia entre los puntos 0-3.

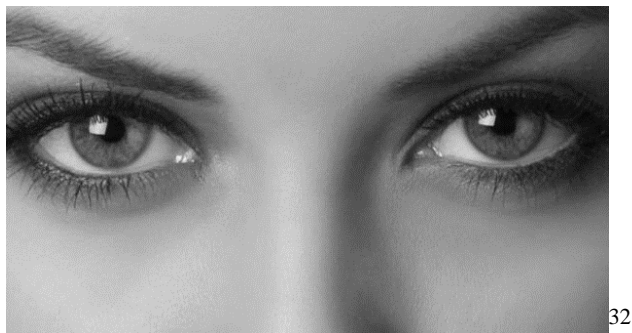
<sup>30</sup> <<http://www.artdiscover.com/en/news/benoit-court-takes-black-and-white-to-another-level/290>>

El sistema contador de parpadeos que se tomó como base tenía ciertos errores en la toma de los puntos para cada ojo, si bien contaba los parpadeos de manera correcta, cuando se individualizó el EAR, la lectura decía que la diferencia de este valor entre ambos ojos siempre era cero, es decir, se encontraban a la misma apertura en la totalidad de las situaciones.

El valor EAR es diferente en cada persona, esto se puede evidenciar muy claramente al observar la población oriental, cuya apertura de ojos es menor comparada con la nuestra.



*Figura 18. Ejemplo de ojos orientales.*



*Figura 19. Ejemplo de ojos occidentales.*

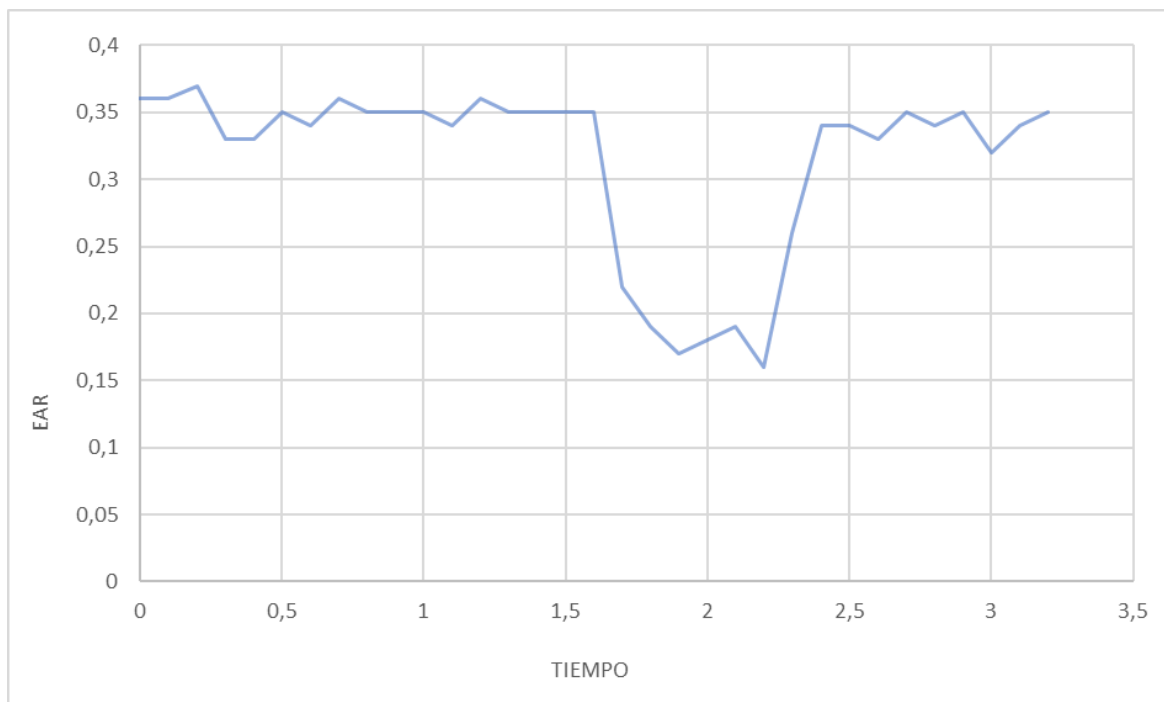
---

<sup>31</sup> <[http://1.bp.blogspot.com/-jI\\_oPqMVOyE/VQrjD2SKyHI/AAAAAAAAALE/5kPCBKbxBvk/s1600/as.jpg](http://1.bp.blogspot.com/-jI_oPqMVOyE/VQrjD2SKyHI/AAAAAAAAALE/5kPCBKbxBvk/s1600/as.jpg)>

<sup>32</sup> <<http://www.abc.es/media/recreo/2016/06/21/beautiful-eyes-9-kL3--620x349@abc.jpg>>

Es por ello por lo que el sistema requiere de una calibración inicial, ya que el funcionamiento total del sistema depende del valor de EAR proporcionado. El desarrollo del sistema se implementa a partir de las características de EAR del autor; por lo que los valores establecidos en el programa son producto de la información obtenida a partir de pruebas de calibración, lo cual se explica a continuación.

En la siguiente imagen, se muestra una gráfica, la cual se obtuvo al realizar el proceso de calibración del sistema para el uso del autor. Para este fin, se implementa un algoritmo que obtiene el EAR de ambos ojos e imprime este valor en pantalla. Se obtienen 10 muestras por segundo.



*Figura 20. Gráfica de variación de EAR en el tiempo: Calibración realizada para el autor*

El valor EAR se sitúa en el eje ‘y’ mientras el tiempo se toma como la variable independiente.

Se puede apreciar que el valor característico de EAR para este caso, se encuentra alrededor de 0,35. El tiempo transcurrido entre 1,6 y 2,4 segundos

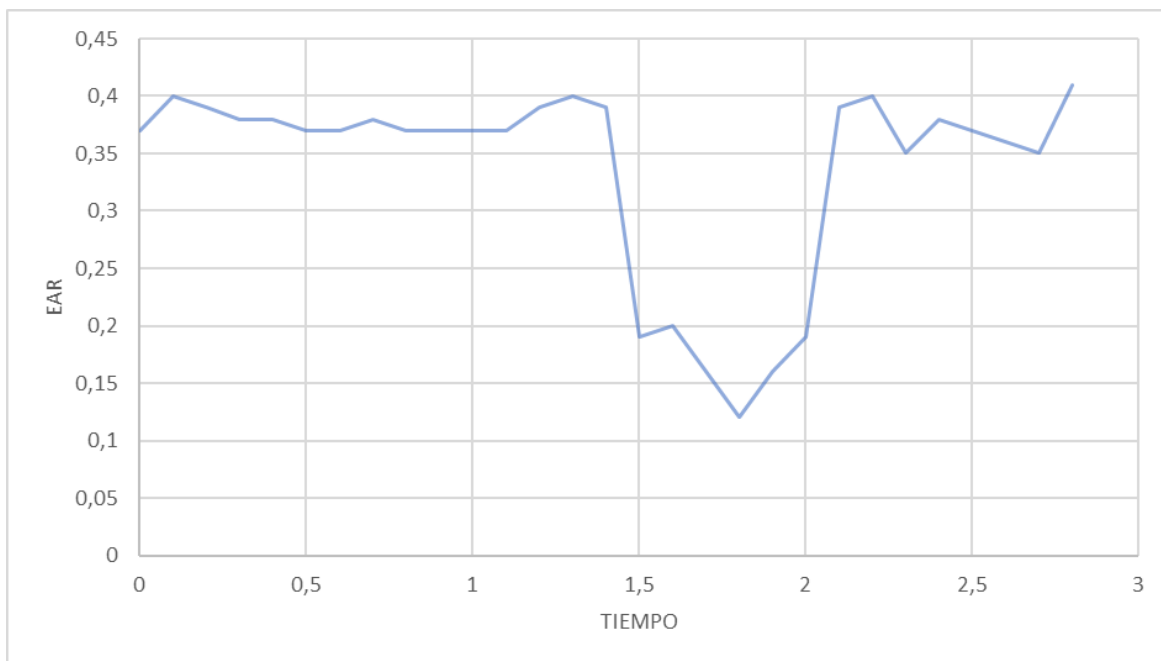


corresponde a un parpadeo, donde el EAR disminuye y aumenta críticamente y vuelve a estabilizarse, es decir, vuelve al valor promedio. El programa se ejecutó durante 3,2 segundos. El valor mínimo de EAR para este caso, fue de 0.16 y el máximo, de 0.37. Los datos obtenidos fueron los siguientes:

Tiempo	EAR
0	0,36
0,1	0,36
0,2	0,37
0,3	0,33
0,4	0,33
0,5	0,35
0,6	0,34
0,7	0,36
0,8	0,35
0,9	0,35
1	0,35
1,1	0,34
1,2	0,36
1,3	0,35
1,4	0,35
1,5	0,35
1,6	0,35
1,7	0,22
1,8	0,19
1,9	0,17
2	0,18
2,1	0,19
2,2	0,16
2,3	0,26
2,4	0,34
2,5	0,34
2,6	0,33
2,7	0,35
2,8	0,34
2,9	0,35
3	0,32
3,1	0,34
3,2	0,35

*Figura 21. Datos obtenidos al realizar el proceso de calibración graficado en la figura 20*

A simple vista, puede haber dos personas con el mismo valor de EAR, sin embargo, esta diferencia puede ser imperceptible. En el siguiente ejemplo, se grafica el mismo proceso anterior de determinación de valor característico de EAR; se puede evidenciar que este valor es mayor al característico del autor:



*Figura 22. Gráfica de variación de EAR en el tiempo: Calibración realizada para otra persona*

Se puede evidenciar que el EAR característico para este caso, se sitúa alrededor de 0,37, con valor máximo de 0,4 y mínimo de 0,12. El parpadeo se realiza entre los 1,4 y 2,1 segundos. Se utilizó el mismo tiempo de muestreo, pero el programa se ejecutó durante 2,8 segundos. Los datos obtenidos fueron los siguientes:

Tiempo	EAR
0	0,37
0,1	0,4
0,2	0,39
0,3	0,38
0,4	0,38
0,5	0,37
0,6	0,37
0,7	0,38
0,8	0,37
0,9	0,37
1	0,37
1,1	0,37
1,2	0,39
1,3	0,4
1,4	0,39
1,5	0,19
1,6	0,2
1,7	0,16
1,8	0,12
1,9	0,16
2	0,19
2,1	0,39
2,2	0,4
2,3	0,35
2,4	0,38
2,5	0,37
2,6	0,36
2,7	0,35
2,8	0,41

*Figura 23. Datos obtenidos al realizar el proceso de calibración graficado en la figura 22*

Se evidenció que hay una característica que puede modificar el valor de EAR de una persona, consiste en la inclinación de la cabeza, de este modo:



*Figura 24. Valor de EAR para una inclinación de cabeza normal*



*Figura 25. Valor de EAR para una inclinación de cabeza hacia atrás*

Con base en las imágenes anteriores, se puede observar, que, al realizar una inclinación de la cabeza hacia atrás, la apertura de los ojos disminuye y por consiguiente el EAR. En la figura 24, este valor es de 0.31, mientras que en figura 25, desciende a 0.19.

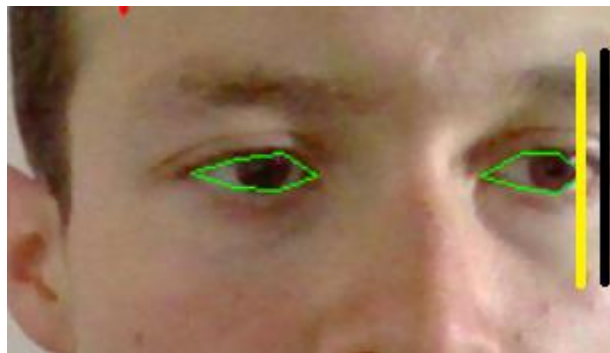
También se evidenció que sitios con poca iluminación ocasionan fallos de lectura, por lo que el rostro de quien controla el sistema debe estar óptimamente iluminado, ya sea de forma natural o artificial.

### **Guiños:**

#### *Sistema de predicción*

El sistema contador de parpadeos que se tomó como base tenía ciertos errores en la toma de los puntos para cada ojo, si bien contaba los parpadeos de manera correcta, cuando se individualizó el EAR, la lectura decía que la diferencia de este valor entre ambos ojos siempre era cero, es decir, se encontraban a la misma apertura en la totalidad de las situaciones, por lo que, dentro de todas las adecuaciones realizadas al código, se incluyó la corrección respectiva de este inconveniente.

Sin embargo, se presentaron errores de lectura, en donde a pesar de mantener los ojos en una apertura similar, se contaba un guiño. Esto debido a que el sistema situaba una o ambas esquinas de alguno de los ojos, en el interior de este:



*Figura 26. Error de lectura en el cálculo del EAR*

En la figura 26 se puede apreciar un error de lectura de la esquina externa del ojo derecho. Se señala con una línea vertical amarilla el punto en el cual el sistema de predicción ubica esta esquina externa y con una línea negra, se señala la posición real

de la misma. Cuando el sistema de predicción sitúa una de las esquinas dentro del ojo, el valor de EAR obtenido aumenta para este mismo ojo, dado que disminuye la distancia horizontal entre esquinas, y cuando el valor de EAR en un ojo aumenta con respecto al otro, se da la orden de guiño.

Por este motivo, se implementó una segunda función de EAR, calculando de forma independiente la apertura de cada ojo.

En vez de relacionar esta apertura con la longitud horizontal del ojo, se relacionó con la apertura del ojo contrario, de modo que este error de lectura (Figura 26) proveniente del algoritmo de predicción, no afecta al sistema cuando se proporcionan las órdenes de guiños.

#### *Diferencia de apertura entre ojos*

Idealmente, ambos ojos se encuentran abiertos a la misma medida, es decir, cada ojo tiene el mismo EAR. Sin embargo, uno de los dos puede estar más abierto que el otro, aunque la esta diferencia es casi imperceptible.

Por esta razón, se establece un valor de tolerancia, el cual se obtiene mediante pruebas de calibración, que corresponde al valor máximo de diferencia de EAR entre ambos ojos. Se establece entonces que, si la diferencia de EAR entre un ojo y otro se encuentra dentro de esta tolerancia, no se cuenta guiño. Si la diferencia entre el EAR de ambos ojos es un poco mayor a esta tolerancia, se cuenta el guiño correspondiente, ya sea derecho o izquierdo.



*Figura 27. Visualización de la diferencia de EAR entre ambos ojos*

En la figura 27, se puede apreciar que el valor de la diferencia entre el EAR de ambos ojos es de 0.03, (Este fue el mayor valor registrado durante la realización de varias pruebas) y se visualiza mediante la variable resta. Así, para contar guiño, la diferencia entre el EAR de ambos ojos debe ser mayor a 0.04, valor para el cual se establece la tolerancia.

Las órdenes de derecha e izquierda se implementan mediante la realización de estos guiños.

Para la orden de arrancar, se debe realizar un parpadeo prolongado (Ojos cerrados aproximadamente 2 segundos), para este caso, el EAR de ambos ojos debe ser menor a los valores de EAR característicos (Gráfica 1: Figura 20), de modo que se cuente el parpadeo y se descarten ciertos valores que corresponden a entrecerrar los ojos.

El motivo de hacer el parpadeo prolongado se debe a que el parpadeo es un movimiento involuntario que se realiza constantemente; por lo tanto, se implementa

un ciclo mediante condicionales, de modo que si durante determinado número de ciclos, el EAR fue siempre menor al valor establecido, se cuenta la orden correspondiente.

El mismo proceso se realiza para la orden de frenar, solo que el condicional, revisa que el EAR sea mayor a un valor establecido, y que, de igual forma, suceda durante un número determinado de ciclos. Es decir, los ojos deben abrirse a una amplitud mucho mayor que la característica.

A pesar de que se tienen los valores de tolerancia y los establecidos para contar órdenes de arrancar y frenar, y como ya se mencionó, el sistema requiere una calibración manual para cada nuevo paciente, que consiste en identificar el valor EAR para el cual los ojos se encuentran normalmente abiertos la mayoría de tiempo. Y con base en este valor, proporcionar el EAR de amplitud máxima (Frenar), amplitud mínima (Parpadeo) y el valor de tolerancia para distinguir los guiños.

Para dar la orden de derecha, se guiña el ojo derecho por un poco más de un segundo; del mismo modo para dar la orden de izquierda. Para arrancar, se cierra ambos ojos por más de dos segundos y para frenar, se abre los ojos en una medida superior a lo que habitualmente permanecen abiertos.



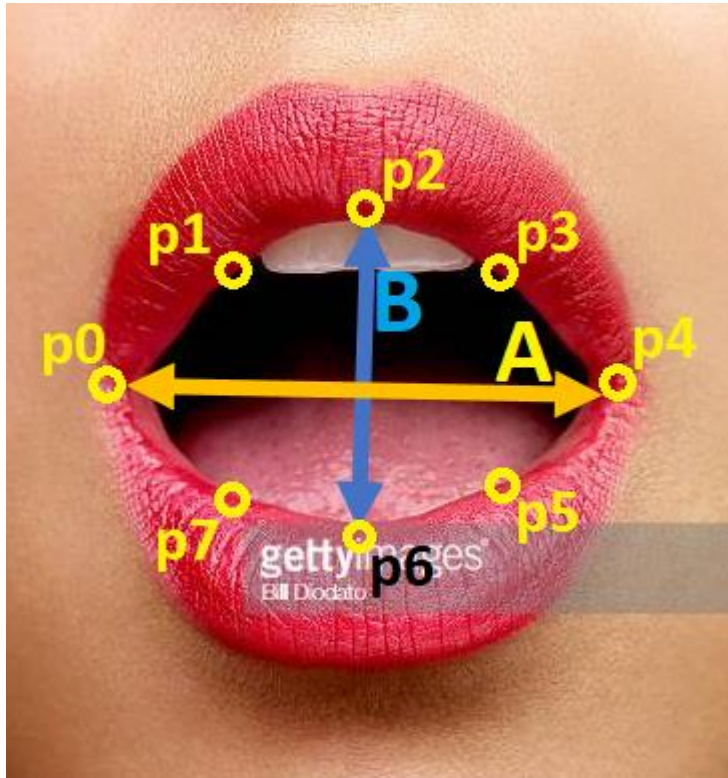
### Finalización del programa:

Ya que el programa se ejecuta dentro de un bucle, se agregó un algoritmo que cierra el programa cuando se abre la boca tres veces seguidas

Se toman los valores que describen el contorno interior de la boca:



Los cuales corresponden a los puntos 61 a 68, para la implementación de la función, se calcula la distancia A: entre los puntos 61 y 65 y B, entre los puntos 63 y 67. Los puntos 61 y 65 corresponden a los extremos del contorno interior de la boca y los puntos 63 y 67 corresponden a la longitud media de la boca, en donde se abre con mayor longitud:



33

*Figura 28. Visualización de la distribución de puntos en el contorno interior de la boca, y la ubicación de las distancias A y B*

La apertura de la boca vendría dada entonces por el valor de B, respecto a la longitud de la boca (A), con lo cual se tendría:

$$Apertura = \frac{|p6 - p2|}{|p4 - p0|} = \frac{B}{A}$$

<sup>33</sup> <<https://www.gettyimages.es/fotos/boca-abierta>>

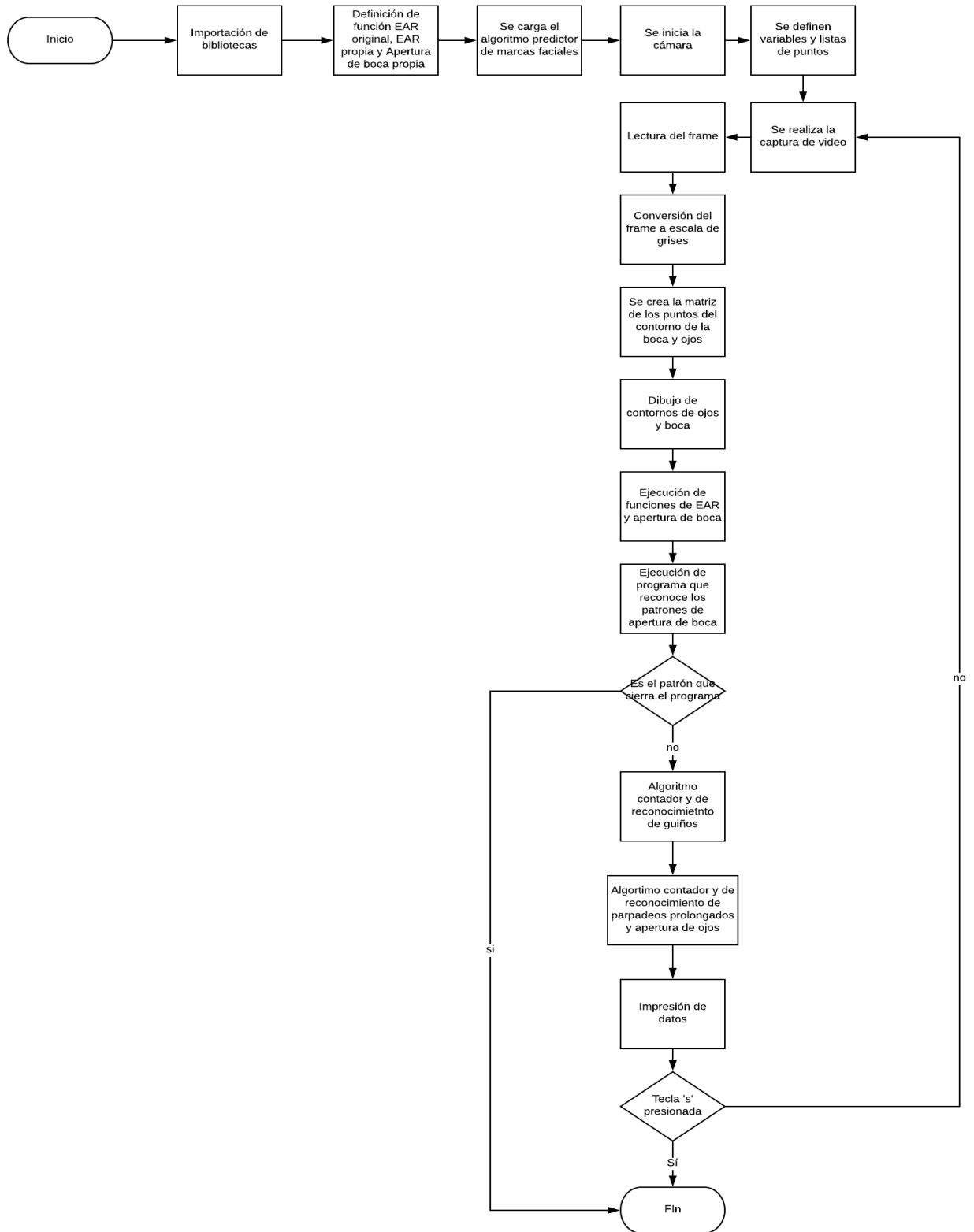
Donde p0 corresponde al punto 61, p2 corresponde al punto 63, p4 corresponde al punto 65 y p6 corresponde al punto 67. Por lo que la función que se implementa es la siguiente:

```
def apertura(b):  
    A = dist.euclidean(b[0], b[4])  
    B = dist.euclidean(b[2], b[6])  
  
    ab = (B/A)  
  
    return ab
```

En el programa, si el valor de apertura es mayor a 0.43, el programa se cierra. Este valor se establece de modo que se deba abrir la boca de manera notable, pero sin que llegue a ser exagerado.

En caso de que se presenten errores o lag en el programa, este también se puede finalizar presionando la tecla 's' o bien mediante la combinación de teclas Ctrl+C, ya que el programa se ejecuta en terminal

## Diagrama de flujo



En el diagrama de flujo anterior (Imagen adjunta), se presenta a modo de resumen y en orden, la estructura general del sistema, la mayoría de los bloques corresponden a algoritmos y funciones con varios subprocesos y ciclos.

### **Cálculo de tiempos:**

Mediante el uso de la función *time()* en el programa, se realizó el cálculo del tiempo de los procesos que se llevan a cabo. Los datos proporcionados, son resultado del promedio de 10 datos obtenidos al realizar varias pruebas:

*Importación de librerías, definición de las funciones EAR y la función de apertura de boca, declaración de variables y listas, carga del algoritmo predictor de marcas faciales (\*.dat) e inicio de cámara:*

```
(cv) carlos@carlos-HP-Laptop-14-bw0xx:~$ python /home/carlos/Desktop/control_0.p  
y --shape-predictor /home/carlos/Desktop/shape_predictor_68_face_landmarks.dat  
Cargando archivo de reconocimiento facial...  
Iniciando camara  
2.05881929397583
```

```
(cv) carlos@carlos-HP-Laptop-14-bw0xx:~$ python /home/carlos/Desktop/control_0.p  
y --shape-predictor /home/carlos/Desktop/shape_predictor_68_face_landmarks.dat  
Cargando archivo de reconocimiento facial...  
Iniciando camara  
1.9559688568115234
```

Para el tiempo en el cual se realizaron todos estos procesos, se obtuvieron los siguientes datos:

2.058819293, 1.945445644, 1.998567525, 1.95596885, 1.926475811, 2.004624598,  
2.007645729, 1.995373293, 2.012585812, 2.03039858, 1.97045127

Por lo que este proceso, tarda en promedio 1.993590513 segundos  $\approx$  2 segundos

*Tiempo transcurrido entre la apertura de ojos correspondiente a la orden de frenar y la impresión en pantalla de esta misma orden*

Es importante conocer el tiempo que transcurre entre el momento en que se indica al sistema la orden de frenado, y el momento en que el sistema recibe la orden (Cuando la imprime en pantalla). Este tiempo debe ser bastante pequeño, ya que al pensar el sistema para una silla de ruedas motorizada, el frenado debe suceder de manera casi instantánea.

0.010242223739624023

0.04276585578918457

Se obtuvieron los siguientes datos:

0.010242223, 0.024546457, 0.032348658, 0.018353594, 0.042765855, 0.044234895, 0.032388110, 0.037273482, 0.017490302, 0.021758941.

Por lo tanto, este proceso tarda en promedio 0.0281402517 *segundos*  $\approx 28mS$

*Tiempo ejecutando las funciones de EAR y de apertura de boca, dibujo de contornos de ojos y boca, creación de matrices, cálculo de posiciones de puntos, lectura del frame y conversión del frame a escala de grises.*

Esta cantidad de procesos se ejecutan durante cada ciclo, es decir, cada vez que se toma un frame de la cámara. Los tiempos obtenidos durante la ejecución de estos procesos, fueron los siguientes:

0.2783172130584717

0.25723910331726074

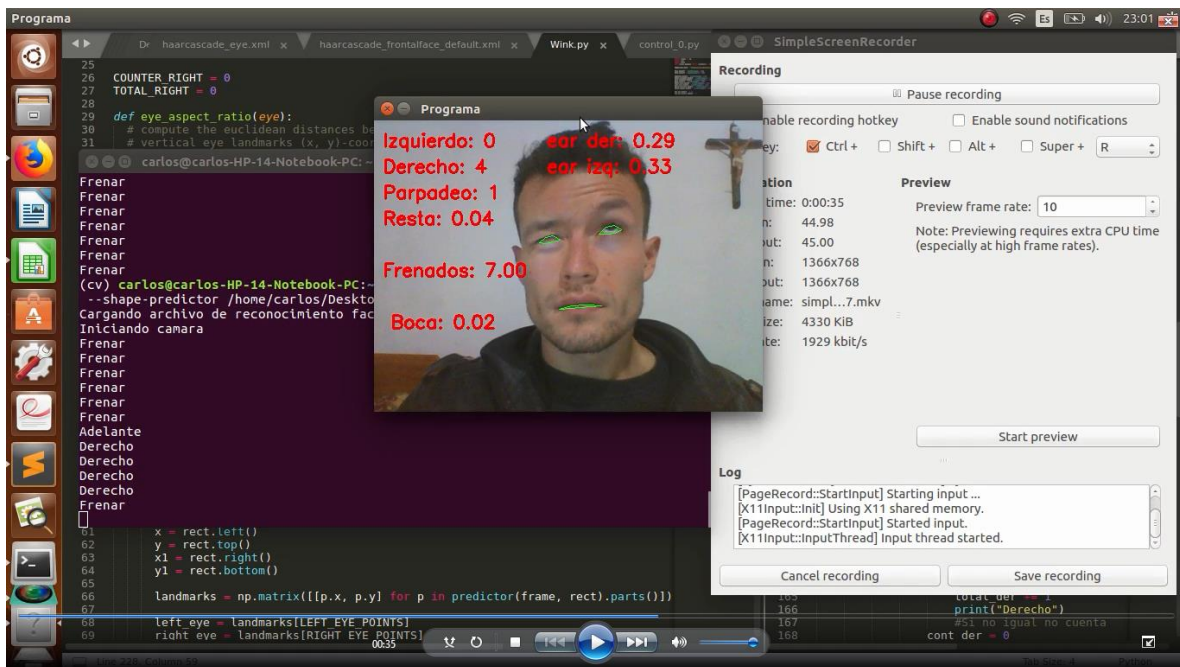
0.278317213, 0.217696964, 0.222348758, 0.257239103, 0.257584932, 0.238593403, 0.213248345, 0.254054385, 0.214774325, 0.227567956

Por lo tanto, este conjunto de procesos se ejecutan en promedio durante 0.2381425384 *segundos*  $\approx 234mS$

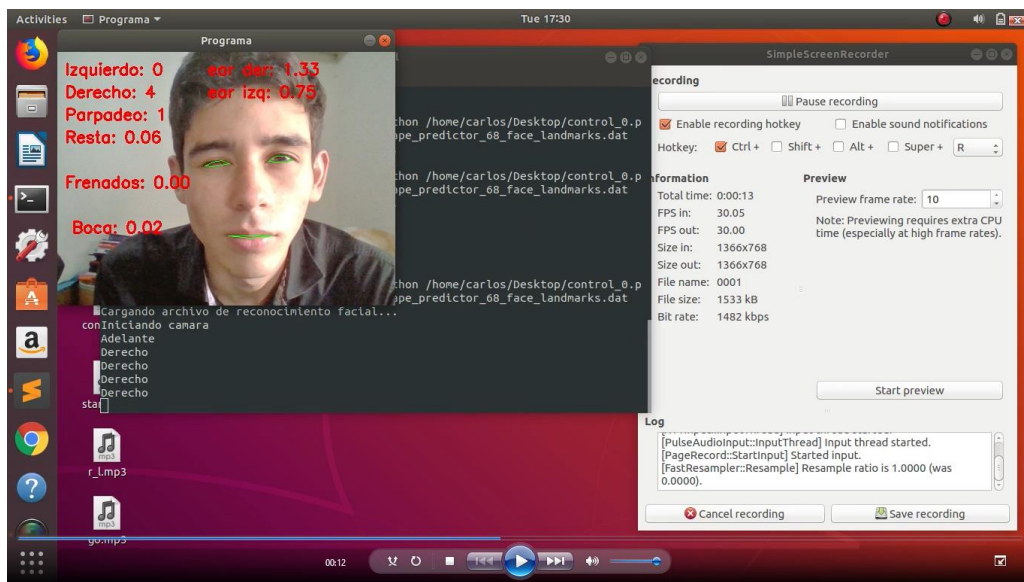
## Pruebas

Se realizaron diversas pruebas durante la realización del sistema, las cuales permitieron la identificación de errores de cálculos que condujeron al mejoramiento del sistema de visión artificial desarrollado presentado como versión final.

Las pruebas fueron realizadas por el autor y por una persona con estrabismo (convergente intermitente)



*Figura 29. Captura de pantalla de una prueba de la última versión del sistema, ejecutada desde el sistema operativo Ubuntu*





La persona con estrabismo que realizó la prueba del sistema es de género masculino, de 18 años de edad, y no posee enfermedades o problemas oculares o de visión adicionales a su estrabismo convergente intermitente.

El autor, quien también realizó la prueba, es de género masculino, de 20 años de edad, padece de miopía en el ojo derecho, y no posee enfermedades o problemas oculares o de visión adicionales a esta condición.

La realización de las pruebas tuvo los siguientes momentos:

1. Ubicación de la persona frente a la cámara del computador, disponiendo de un entorno con condiciones de luminosidad óptimas.
2. Comienzo de la grabación de pantalla mediante el software Simple Screen Recorder.
3. Puesta en marcha del programa de calibración.
4. Toma de datos de los valores de calibración, graficación y análisis.
5. Finalización de la grabación de pantalla.
6. Modificación del sistema de control con base en los datos obtenidos mediante la calibración.
7. Comienzo de la grabación de pantalla mediante el software Simple Screen Recorder.
8. Puesta en marcha del sistema de control.
9. Proporcionar al sistema las órdenes de Adelante, Girar (Derecha e Izquierda), Frenar, Finalizar programa.
10. Finalización de la grabación de pantalla.
11. Análisis del video obtenido, para detectar errores de lectura y órdenes incorrectas.

La secuencia de las ordenes descritas en el momento número 9, se realiza pensando en la aplicación para la cual se realizó el sistema: Una silla de ruedas motorizada, donde la primera orden sería arrancar, seguida de movimientos de derecha e izquierda para luego dar una orden de frenado, y una posible finalización del programa.

#### *Persona con estrabismo.*

Para la persona con estrabismo se realizó un total de 18 pruebas, los errores que se presentaron se muestran a continuación:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Guiño leído como orden de Adelante					X													
Error de lectura en la apertura de boca												X						

En el 11.11% de las pruebas, se obtuvieron errores de lectura, uno de los errores correspondió a la lectura de un guiño como orden de adelante, y para el error restante, el sistema señalaba una apertura de boca menor a la real, de modo que se canceló la finalización del programa. Por consiguiente, para la persona con estrabismo, el sistema funcionó satisfactoriamente el 88.89% de las veces.

#### *Autor*

Para el caso del autor, se realizó de igual modo un total de 18 pruebas, y los errores presentados se presentan a continuación:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Guiño no leído		X		X														
Error de lectura en la apertura de boca												X						

Para este caso, en el 16.66% de las pruebas, se obtuvieron errores de lectura, en dos ocasiones, el sistema no realizó la lectura de un guiño; l otro error consistió en que, el sistema señalaba una apertura de boca menor a la real, de igual modo que ocurrió a la persona con estrabismo. Por lo tanto, para el caso del autor, el sistema funcionó satisfactoriamente en el 83.34% de las veces.

El funcionamiento satisfactorio del programa fue del 86.115%.

Se puede afirmar, por lo tanto, que el sistema es efectivo en personas que padecen estrabismo, (Como se puede apreciar en la figura 30 y video adjunto como anexo) y de igual modo en personas que no poseen la condición (Como el autor en la figura 29 y video adjunto como anexo), dado que las órdenes no dependen de la posición del iris, sino de guiños, y patrones de parpadeos y apertura de ojos.

Nota: Las personas de las figuras 24-25, y 30-31 autorizaron al autor para incluir su imagen en este documento.

## Conclusiones

- Si bien el programa podría funcionar para otra persona (Con los mismos valores proporcionados por el proceso de calibración para el autor), es necesario realizar una calibración para cada usuario, ya que como bien se dijo el valor de EAR es diferente para cada persona.
- No es conveniente exportar el sistema a una Raspberry Pi® 3 o anterior, ya que se presentó un lag de aproximadamente 4 segundos, lo cual no es útil para sistemas en tiempo real. Sin embargo, debido a las constantes mejoras de la Raspberry Pi en cuanto a velocidad y almacenamiento<sup>34</sup>, no se descarta que una futura generación de este hardware esté en la capacidad de hospedar y correr el programa.
- Para aplicaciones posteriores, se podría acoplar el sistema a una silla de ruedas eléctrica, ya que el proyecto se realizó para este fin.
- Para la implementación del sistema, es necesario contar con buenas condiciones de luminosidad, ya sea natural o artificial con el fin de evitar fallos de lectura.

---

<sup>34</sup> < <http://www.trustedreviews.com/opinion/raspberry-pi-3-vs-pi-2-2936374> >

## **Bibliografía:**

Silla de ruedas Inteligente – Moviliza a personas Paraplégicas

[https://www.youtube.com/watch?v=Y27\\_oqvq-ws](https://www.youtube.com/watch?v=Y27_oqvq-ws)

Jóvenes pereiranos inventaron silla de ruedas que se maneja con los ojos - Enero 29 de 2013

<https://www.youtube.com/watch?v=8HkSNpi3nWE>

Silla de Ruedas Controlada por Músculos Oculares - Canal 40

<https://www.youtube.com/watch?v=SwRJv9kPqrI>

En Colombia inventan silla de ruedas que se mueve con los ojos

<https://espanol.upi.com/Curiosidades/2006/09/08/En-Colombia-inventan-silla-de-ruedas-que-se-mueve-con-los-ojos/37511157753940/>

Uso del movimiento de los ojos para desplazarse en silla de ruedas

<http://ingenierias.utp.edu.co/ingenieria-electronica/trabajos-de-grado.html>

My blog about Making, Coding, Building things

<https://zerozeroonezeroonezeroonezero.wordpress.com>

Estrabismo

<http://fon.org.co/estrabismo>

Visión Artificial

[http://www.academia.edu/6084688/VISI%C3%93N\\_ARTIFICIAL](http://www.academia.edu/6084688/VISI%C3%93N_ARTIFICIAL)

OpenCV-About

<https://opencv.org/about.html>

Dlib C++ Library

<http://dlib.net/>

Object Detection in Computer Vision

<https://www.mathworks.com/discovery/object-detection.html>

An Introduction to CNN based Object Detection

[https://blog.csdn.net/joshua\\_1988/article/details/73252738](https://blog.csdn.net/joshua_1988/article/details/73252738)

Deep Learning

<https://www.investopedia.com/terms/d/deep-learning.asp>

What is Deep Learning?

<https://www.mathworks.com/discovery/deep-learning.html>

Train your own OpenCV Haar classifier

<http://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html>

Creating your own Haar Cascade OpenCV Python Tutorial

<https://pythonprogramming.net/haar-cascade-object-detection-python-opencv-tutorial/>

What Is Face Recongition

<https://www.facefirst.com/face-recognition-glossary/what-is-face-recognition>

Objective C—Image Face Detection

<https://medium.com/@klin505/objective-c-image-face-detection-1f634215311c>

Real-Time Face Pose Estimation

<http://blog.dlib.net/2014/08/real-time-face-pose-estimation.html>

Eye blink detection with OpenCV, Python, and dlib

<https://www.pyimagesearch.com/2017/04/24/eye-blink-detection-opencv-python-dlib>

<https://edoras.sdsu.edu/doc/matlab/toolbox/images/hsvcone.gif>

Assignment of genes coding for brown eye colour (BEY2) and brown hair colour (HCL3) on chromosome 15q.

<https://www.ncbi.nlm.nih.gov/pubmed/8875191>

Estrabismo y Ambliopía

<https://www.pediatrintegral.es/numeros-anteriores/publicacion-2013->

09/estrabismo-y-ambliopia/

¿Qué es el estrés visual?

[https://www.clarin.com/buena-vida/estres-vista-astenopia-prevencion\\_0\\_S1Jedzjqv7g.html](https://www.clarin.com/buena-vida/estres-vista-astenopia-prevencion_0_S1Jedzjqv7g.html)

Estrabismo

<https://areaoftalmologica.com/enfermedades-oculares/estrabismo/>

Facial Point Annotations

<https://ibug.doc.ic.ac.uk/resources/facial-point-annotations>

What does top, left, right and bottom mean in Android Rect object

<https://stackoverflow.com/questions/22589322/what-does-top-left-right-and-bottom-mean-in-android-rect-object>

Soukupova, T. Cech, Jan. (2016) *Real-Time Eye Blink Detection using Facial Landmarks*. Rimske Toplice, Eslovenia

<http://vision.fe.uni-lj.si/cvww2016/proceedings/papers/05.pdf>

Benoit Courti takes black and white to another level

<http://www.artdiscover.com/en/news/benoit-court-takes-black-and-white-to-another-level/290>

[http://1.bp.blogspot.com/-](http://1.bp.blogspot.com/-jl_oPqMVOyE/VQrjD2SKyHI/AAAAAAAAALE/5kPCBKBxBvk/s1600/as.jpg)

[jl\\_oPqMVOyE/VQrjD2SKyHI/AAAAAAAAALE/5kPCBKBxBvk/s1600/as.jpg](http://1.bp.blogspot.com/-jl_oPqMVOyE/VQrjD2SKyHI/AAAAAAAAALE/5kPCBKBxBvk/s1600/as.jpg)

<http://www.abc.es/media/recreo/2016/06/21/beautiful-eyes-9-kL3--620x349@abc.jpg>

<https://www.gettyimages.es/fotos/boca-abierta>

El Inglés también es el idioma de Internet

<https://www.schola.es/blog/el-ingles-tambien-es-el-idioma-de-internet/>

Oostendorp, N. Oliver, A. Scott, K. DeMaagd, K. (2012) *Practical Computer Vision with SimpleCV*, Estados Unidos, O'Reilly Media

Bradski, G. Kaehler, A. (2008) *Learning OpenCV*, Estados Unidos, O'Reilly Media