

Máster en Ingeniería de Sistemas y Control



Trabajo Fin de Máster

Probador virtual de gafas con reconocimiento de caras mediante técnicas de visión por computador

Alumno: Miguel Ángel García Gómez

Director: Gonzalo Pajares Martinsanz

Curso Académico 2013/2014 – Convocatoria de Septiembre

Autorización

Se autoriza a la Universidad Complutense de Madrid (UCM) y a la Universidad Nacional de Educación a Distancia (UNED) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firmado:

Resumen

En este proyecto se describe la metodología utilizada y el diseño realizado para el desarrollo de una aplicación web, con el fin de crear un probador virtual de gafas, en el cual el usuario, situado frente a una cámara de un dispositivo con acceso a internet y enfocando su cara, selecciona las gafas a probarse, de suerte que el sistema superpone las gafas elegidas sobre la cara, previamente reconocida, de la persona creando la sensación de que las gafas están realmente posicionadas en la cara de la persona, como ocurre en un probador real.

La imagen capturada por la cámara del dispositivo (ordenador personal, teléfono móvil o tableta) se envía a un servidor donde se procesa convenientemente, mediante técnicas de visión por computador basadas en el reconocimiento de caras y ojos en las mismas, devolviendo el resultado de la superposición de las gafas sobre la cara reconocida al dispositivo para su visualización por el usuario.

Palabras clave: Visión por computador. Reconocimiento de objetos a través de patrones. *Haar Feature-based Cascade Classifiers*. Algoritmo de Viola y Jones.

ÍNDICE GENERAL

1. INTRODUCCIÓN	9
1.1. MOTIVACIÓN	9
1.2. OBJETIVOS	10
1.3. PLANTEAMIENTO DEL PROBLEMA	10
1.5. ORGANIZACIÓN DE LA MEMORIA	11
2. ESTADO DEL ARTE	13
2.1. VISIÓN GENERAL	13
2.1. ADQUISICIÓN DIGITAL DE IMÁGENES	14
2.1.1. <i>Cámara digital</i>	15
2.1.2. <i>Cámara web</i>	15
2.1.3. <i>Cámara de un teléfono móvil</i>	15
2.1.4. <i>Propiedades de las imágenes en el proceso de adquisición</i>	16
2.2. ENVÍO DE LAS IMÁGENES	20
2.3. PROCESAMIENTO DE LA IMAGEN: RECONOCIMIENTO FACIAL	20
2.4. DEVOLUCIÓN DE LA IMAGEN	22
3. DESCRIPCIÓN DEL MÉTODO PROPUESTO	25
3.1. PROPUESTA GENERAL	25
3.2. ADQUISICIÓN DE IMÁGENES Y ARQUITECTURA CLIENTE-SERVIDOR EN TIEMPO REAL	26
3.2.1. <i>Resolución</i>	26
3.2.2. <i>Distancia de la cara a la cámara</i>	26
3.2.3. <i>Modelo de color</i>	27
3.2.4. <i>Iluminación</i>	27
3.2.5. <i>Entorno</i>	27
3.2.6. <i>Envío de imágenes a través de Internet</i>	28
3.2.7. <i>Tasa de transferencia</i>	28
3.2.8. <i>Arquitectura cliente-servidor para tiempo real</i>	28
3.3. RECONOCIMIENTO FACIAL	29
3.3.1. <i>Transformación a escala de grises de la imagen RGB</i>	29
3.3.2. <i>Algoritmo de Viola y Jones para la detección facial</i>	29
3.3. FUSIÓN DE IMÁGENES	39
3.3.1. <i>Re-escalado de la imagen de las gafas</i>	39
3.3.2. <i>Superposición de la imagen de las gafas</i>	39
4. RESULTADOS OBTENIDOS	41
4.1. DATOS UTILIZADOS	41
4.1.1. <i>Descripción de las imágenes</i>	41
4.1.2. <i>Ficheros clasificadores de Haar</i>	42
4.1.3. <i>Parámetros</i>	43
4.2. DETALLES DEL SERVIDOR Y MÁQUINAS UTILIZADAS	43
4.3. MÉTODOS UTILIZADOS	43
4.3.1. <i>Parámetros ajustables del algoritmo de Viola y Jones</i>	43
4.3.2. <i>Métodos implementados</i>	44
4.4. RESULTADOS	47
4.4.1. <i>Pruebas para el método 1</i>	49

Trabajo Fin de Máster	Página
Curso Académico 2013/2014	v

4.4.2. Pruebas para el método 2.....	54
4.4.3. Pruebas para el método 3.....	59
4.4.4. Comparación de métodos.....	63
5. CONCLUSIONES Y TRABAJO FUTURO.....	69
5.1. CONCLUSIONES	69
5.2. TRABAJO FUTURO	69
APÉNDICE I. BIBLIOGRAFÍA	71
APÉNDICE II. GLOSARIO DE TÉRMINOS Y DEFINICIONES	73

ÍNDICE DE FIGURAS

Figura 1. Ejemplo de pose válida	17
Figura 2. Ejemplo de pose no válida	17
Figura 3. Ejemplo de imagen con entorno despejado	19
Figura 4. Ejemplo de imagen con entorno complejo.....	19
Figura 5. Diseño general del probador virtual	25
Figura 6. Región de la imagen integral	31
Figura 7. La suma de los píxeles dentro del rectángulo D puede ser calculada con cuatro puntos de referencia de la matriz. El valor de la imagen integral en la posición 1 es la suma de los píxeles en el rectángulo A. El valor de la posición 2 es A+B, en la posición 3 es A+C y en la posición 4 es A+B+C+D. La suma dentro de D puede ser calculada como $4 + 1 - (2 + 3)$	32
Figura 8. Ejemplo de creación de la imagen integral a través de la imagen original. El primer caso es la creación de la imagen integral en base a la frontera y la segunda en cualquier otra posición interior.....	32
Figura 9. Ejemplo de características rectangulares mostradas relativas a la ventana principal de detección. La suma de los píxeles que se encuentran dentro de los rectángulos blancos se resta de la suma de los píxeles en los rectángulos grises. Características de dos rectángulos (<i>two-rectangle feature</i>) se muestran en (A) y (B). La figura (C) muestra una característica de tres rectángulos (<i>three-rectangle feature</i>), y (D) una característica de cuatro rectángulos (<i>four-rectangle feature</i>).....	33
Figura 10. Ejemplo de aplicación de algunas características para el reconocimiento facial sobre una sub-ventana.....	34
Figura 11. Fases de reconocimiento	34
Figura 12. Representación esquemática de clasificación en cascada.....	37
Figura 13. Ejemplo simplificado de aplicación de cascada de clasificadores a una sub-ventana que no contiene una cara	38
Figura 14. Ejemplo simplificado de aplicación de cascada de clasificadores a una sub-ventana que sí contiene una cara	38
Figura 15. Imagen de ejemplo de gafas.....	39
Figura 16. Representación esquemática de la posición aproximada de los ojos	40
Figura 17. testUnaPersona01.png	42
Figura 18. testUnaPersona02.png	42
Figura 19. testUnaPersona03.png	42
Figura 20. testUnaPersona04.png	42

Figura 21. testUnaPersona05.png	42
Figura 22. testUnaPersona06.png	42
Figura 23. testUnaPersona07.png	42
Figura 24. testUnaPersona08.png	42
Figura 25. testNoHumano01.png	42
Figura 26. testNoHumano02.png	42
Figura 27. testDosPersona.png.....	42
Figura 28. testUnaPersona-NoHumano.png	42
Figura 29. Posición estimada de los ojos basada en las medidas del ancho y alto de la cara.	47
Figura 30. Selección de parámetros en la herramienta de test.....	48
Figura 31. Resultado de un test en la herramienta de test.	49
Figura 32. Gráfica comparativa del Tiempo de Envío.....	64
Figura 33. Gráfica comparativa del Tiempo de Procesamiento.	64
Figura 34. Gráfica comparativa del Tiempo de Recepción.	65
Figura 35. Gráfica comparativa de la Tasa de Acierto.	65
Figura 36. Superposición de gafas 1.	66
Figura 37. Superposición de gafas 2.	66
Figura 38. Superposición de gafas 3.	66
Figura 39. Superposición de gafas 4.	66

ÍNDICE DE TABLAS

Tabla 1. Resultados de las pruebas de ejecución del método 1.....	54
Tabla 2. Resultados de las pruebas de ejecución del método 2.....	59
Tabla 3. Resultados de las pruebas de ejecución del método 3.....	63
Tabla 4. Resumen de los resultados obtenidos en la ejecución del método 1, 2 y 3.	64

1. INTRODUCCIÓN

1.1. MOTIVACIÓN

En este proyecto se describe la metodología y diseño utilizado, mediante una aplicación web, para crear un probador virtual de gafas, en el cual el usuario, situado frente a una cámara de un dispositivo con acceso a internet y enfocando su cara, selecciona las gafas a probarse, de suerte que el sistema superpone las gafas elegidas sobre la cara, previamente reconocida, de la persona creando la sensación de que las gafas están posicionadas realmente en la cara de la persona, como ocurre en un probador real.

La imagen capturada por la cámara del dispositivo (ordenador personal, teléfono móvil o tableta) se envía a un servidor donde se procesa convenientemente, mediante técnicas de visión por computador basadas en el reconocimiento de caras y ojos en las mismas, devolviendo el resultado de la superposición de las gafas sobre la cara reconocida al dispositivo para su visualización por el usuario.

Las motivaciones del proyecto se resumen en los siguientes puntos:

1. Implementación de una aplicación web basada en un probador virtual de gafas remoto para usuarios que posean un dispositivo conectado a Internet equipado con cámara.
2. Estudio de las tecnologías de acceso remoto y de técnicas de reconocimiento facial para el procesamiento de imágenes en el servidor.
3. Facilitar las compras on-line como sistema de futuro en cualquier parte del mundo con acceso a internet.

La idea del proyecto es aprovechar la tecnología actual para facilitar la vida de las personas. Un gran número de personas posee, al menos, un dispositivo con cámara, ya sea una cámara web en el ordenador o las cámaras delantera o trasera de un teléfono móvil inteligente o una tableta con su correspondiente cámara de captura. Además, estas tecnologías se encuentran en continuo auge con un alto grado de desarrollo y capacidades en la actualidad, con previsión de crecimiento en el futuro, a la vez que aumenta el número de potenciales usuarios. Esta circunstancia ofrece la posibilidad de aplicar las técnicas de visión por computador a actividades tan cotidianas como es el hecho de probarse unas gafas, siendo en este caso la acción concreta por la que se motiva este trabajo, sentando así las bases para futuras ampliaciones para otro tipo de objetos y probablemente en otras partes del cuerpo diferentes a la cara, siempre con la orientación de facilitar pruebas virtuales, simulando a las reales.

El creciente número de tiendas on-line y de venta por Internet tiene actualmente un gran problema, siendo éste el motivo por el cual mucha gente es reacia a realizar compras por Internet. Exactamente el problema surge del hecho de no poder probarse físicamente los productos antes de ser comprados, como se hace en una tienda convencional, lo que retrae a muchas personas a la hora de realizar tal operación.

Con el presente trabajo se pretende abrir el camino a la posibilidad de probarse virtualmente dichos productos sin tenerlos físicamente. El proyecto se enmarca en un nuevo paradigma surgido al amparo de las nuevas tecnologías en lo que se conoce como Internet de las cosas, cuya terminología inglesa es *Internet of Things* con un amplio auge a nivel tanto de la comunidad científica como de las industrias relacionadas con el sector de las tiendas on-line que ofrecen productos cuya adquisición en tiendas convencionales requiere de una prueba previa.

1.2. OBJETIVOS

Los objetivos específicos que se plantean en el presente proyecto son:

1. Diseñar la tecnología necesaria para el acceso web remoto a través de internet con fines de procesamiento de imágenes.
2. Detección de caras humanas en tiempo real en imágenes captadas por la cámara de un dispositivo con acceso a internet.
3. Encontrar el método más óptimo para realizar dicha detección.
4. Superposición de las imágenes con la cara detectada y el objeto a probar.
5. Conseguir que el proceso completo se realice en el menor tiempo posible para crear la sensación de tiempo real.

Así pues, se diseña una aplicación web de acceso remoto a través de internet para el procesamiento de imágenes en el servidor, donde se plantea el reconocimiento de caras en las imágenes captadas por los dispositivos dotados de cámara y equipados con cualquier sistema operativo, de forma que el único requisito adicional de tales dispositivos es que posean conexión a Internet.

Al planteamiento anterior, que constituye el objetivo principal hay que sumarle objetivos secundarios, pero no por ello menos importantes, como es el de encontrar el método más óptimo para la detección de las caras y la superposición de imágenes, ya que hemos de pensar que al utilizar Internet como medio de intercambio de datos y mensajes entre el dispositivo y el servidor, que realizará las labores de detección y transformación de la imagen, el rendimiento es una parte muy importante de cara a la solución del problema propuesto.

Dadas las características del proyecto, éste se plantea de forma global desde el punto de vista que constituye un todo, comprendiendo desde la captura de la imagen, su envío a un servidor para su procesamiento, la superposición de las gafas a probar sobre las caras detectadas hasta la devolución del producto fusionado al dispositivo del usuario para la visualización del resultado de la prueba y proceder en consecuencia.

1.3. PLANTEAMIENTO DEL PROBLEMA

Considerando lo anteriormente expuesto, a nivel global, el problema a resolver presenta el siguiente planteamiento:

- Obtener la imagen de la cara del usuario mediante el dispositivo dotado de cámara.
- Enviar la imagen captada al servidor.
- Procesar dicha imagen en el servidor con el fin de detectar la cara en la misma, superponiendo sobre ella las gafas seleccionadas.
- Devolver la imagen transformada al dispositivo de usuario para su visualización en pantalla.

Una duda importante que surge en este planteamiento es determinar la mejor técnica para la detección facial, que como se ha mencionado previamente constituye una parte relevante del proceso. Entre otras consideraciones es necesario utilizar un procedimiento que en conjunto con los accesos a través de internet cumplan con los requisitos de eficiencia y tiempo real para que el usuario consiga la máxima sensación de proximidad a las pruebas reales.

En este caso se ha optado por utilizar tecnologías bien desarrolladas de acceso a internet, así como una técnica eficiente de detección de objetos basada en las características de *Haar* propuesta por Viola y Jones [1]. En este caso, se ha seleccionado este método ya que cumple con los anteriores requisitos a

juzgar por los trabajos que la utilizan, ofreciendo por tanto, una capacidad de detección suficientemente rápida y con una media de falsos positivos relativamente baja.

En el conjunto del proyecto es necesario determinar las características de los componentes y procesos involucrados en el diseño del probador virtual que se propone, esto es: captura de imágenes, envío al servidor, reconocimiento de caras en la imagen, fusión de imágenes y reenvío de las imágenes procesadas al dispositivo para su visualización. En todos ellos rigen los mencionados requisitos de eficiencia y tiempo real, para que en conjunto el sistema ofrezca las prestaciones suficientes para la aplicación que se pretende.

1.5. ORGANIZACIÓN DE LA MEMORIA

El presente trabajo se estructura en cinco capítulos, que se explican a continuación.

El primer capítulo resume la motivación, los objetivos y el planteamiento a nivel global del problema presentado, tal y como se han expuesto previamente.

En el segundo capítulo se hace una revisión del estado del arte, explicando con mayor profundidad el planteamiento del problema desde la perspectiva de los métodos y estrategias relacionados con el objeto del presente trabajo.

Una vez introducido el procedimiento empleado y el planteamiento del problema, en el tercer capítulo se describe la propuesta del sistema para cumplir con el objetivo de servir de probador virtual de gafas. Se analizan las alternativas de diseño, desarrollo e implementación y una descripción de los métodos y algoritmos utilizados.

El cuarto capítulo muestra y analiza los resultados obtenidos mediante la ejecución de las pruebas mediante el procedimiento propuesto. Dichas pruebas, junto con sus resultados, permiten extraer las conclusiones pertinentes sobre el rendimiento de la estrategia propuesta en base a los métodos utilizados.

Por último, en el quinto capítulo se concretan las conclusiones finales y los posibles trabajos futuros.

También se incluye un apéndice con el glosario de términos y definiciones habitualmente utilizados en el contexto del procedimiento que se plantea y las metodologías utilizadas.

2. ESTADO DEL ARTE

2.1. VISIÓN GENERAL

El escenario propuesto en el probador virtual plantea varios retos, según las fases mencionadas previamente en relación al diseño propuesto: adquisición de la imagen por parte de una cámara web o la cámara frontal o trasera de un teléfono móvil inteligente, envío de imágenes a través de Internet, procesamiento de las mismas en el servidor, incluyendo el procedimiento de detección de la cara o caras y la fusión con el modelo de gafas seleccionado y devolución al dispositivo.

En todas las etapas hay que tener presente que el sistema propuesto es un sistema en tiempo real, por lo que es necesario considerar los métodos y algoritmos que nos proporcionen un alto rendimiento con el menor tiempo de cálculo posible para que el resultado global sea aceptable. Sobre todo hay que tener en cuenta la fase de procesamiento de las imágenes ya que es la parte más crítica del proceso de forma que se realice una correcta detección facial para poder colocar las gafas en el sitio correcto.

Existen algunas soluciones similares ya implementadas, de las que se pueden distinguir varios tipos:

1. **Probador virtual on-line basado en la captación en vivo de una secuencia de imágenes:** estas soluciones son muy similares a las planteadas en este trabajo:
 - a. Probador virtual de gafas de Ray-Ban [2].
 - b. Probador virtual de gafas de Direct-Optic [3].
 - c. Probador virtual desarrollado por Castrillón y col. [4].
2. **Probador virtual on-line basado en la captación de una única imagen:** en este caso se captura una imagen desde una cámara, o se envía una imagen que el usuario tenga almacenada previamente, de forma que el sistema realiza el reconocimiento y posicionamiento sobre esa imagen [5].
3. **Probador virtual on-line basado en modelos:** la solución propuesta por este tipo de probadores es la de probar el producto en un modelo predefinido. El usuario elige el modelo que más se parezca a él y se prueban las gafas en dicho modelo. En este caso no es necesaria la utilización de la cámara.

Existen también soluciones similares en cuanto al concepto de probador. Éstas se sitúan dentro de lo que pueden considerarse como los probadores virtuales de ropa "off-line". Algunas de estas implementaciones requieren la utilización de cámaras y equipos que ya incluyan reconocimiento y detección de gestos, como el sistema Kinect de Microsoft [6]. En este caso es "off-line" porque no necesitan que el equipo esté conectado a Internet puesto que es el propio sistema (la cámara de Kinect) la que proporciona el reconocimiento y detección de los gestos.

Según la categorización de probadores existentes, el proyecto que se describe en la presente memoria se enmarca dentro del tipo de "**Probador virtual on-line basado en la captación en vivo de imágenes**" y, a pesar de que ahora ya existen implementaciones similares, el presente trabajo pretende mejorar las soluciones ya existentes haciendo que el probador sea accesible por mucha más gente utilizando no sólo una cámara web, sino aprovechar la cámara de cualquiera de los dispositivos móviles (ordenadores, teléfonos o tabletas) sin tener en cuenta la plataforma en la que se ejecuten, ya que las soluciones propuestas en (1) son muy dependientes de la plataforma donde se ejecute. De ahí surge la idea de hacerlo a través de la web. Internet es la herramienta que nos proporciona la independencia de la plataforma, porque utiliza estándares que han de ser reconocidos por todas las plataformas. De esta manera, el probador virtual puede ser accedido desde sistemas Windows, Android, iOS u otros sistemas operativos.

Por otra parte, una mejora adicional del planteamiento propuesto consiste en garantizar una eficiencia suficiente independientemente de la cámara de captura que se utilice. Es bien sabido en procesamiento de imágenes, que diferentes dispositivos generan imágenes distintas, tanto en lo que se refiere a matices y tonalidades de color como en niveles de intensidad o resolución espacial. Muchas veces un mismo algoritmo de procesamiento puede tener éxito con una imagen o tipo de imágenes determinadas y fracasar con imágenes de otro tipo, siendo por tanto altamente dependientes del dispositivo de captura según las características intrínsecas de las imágenes. El procedimiento que se propone se plantea desde la perspectiva razonable de independizarlo del tipo de imágenes utilizadas, y por tanto de los dispositivos que las obtienen, orientándose así hacia la consecución del mayor nivel posible de autonomía en este sentido, de cara a su validez universal. Esta circunstancia nos lleva a la consideración de elegir métodos y procedimientos suficientemente robustos en este sentido.

A continuación se revisan los distintos métodos y procedimientos que conforman el proceso completo desde el punto de vista de su situación actual, que constituyen el punto de partida para la propuesta que se formula. En este sentido se siguen los pasos de ésta.

2.1. ADQUISICIÓN DIGITAL DE IMÁGENES

La adquisición de la imagen se hace a través de una cámara web o cámara frontal incorporada a un dispositivo con capacidad de conexión a internet. Se han seleccionado estos dispositivos ya que en la actualidad el mercado en relación a estos dispositivos ofrece una amplia gama de posibilidades, habiendo crecido enormemente y por consiguiente el número de usuarios [7]. En este sentido, el hecho de ofrecer el uso del probador virtual a una gran cantidad de usuarios queda suficientemente cubierto, ya que una gran cantidad de usuarios poseen al menos un ordenador con cámara web o un teléfono móvil inteligente con cámara y cada vez más algunas tabletas con su correspondiente cámara [8].

Existen dos problemas con este tipo de dispositivos:

1. Calidad de las imágenes adquiridas: al ser dispositivos de bajo coste, la calidad que ofrecen no es la misma que una cámara específicamente diseñada para la detección de objetos.
2. No llevan incorporado ningún sistema de detección o reconocimiento de caras, por lo que esta funcionalidad nos la debe ofrecer el sistema al que estén conectados.

Se ha considerado la utilización de cámaras como Kinect [6], desestimándose finalmente la idea porque no es consecuente con la hecho de independizar el probador virtual de la plataforma, ya que una cámara Kinect es dependiente de plataformas Microsoft para su funcionamiento, y también nos aleja del objetivo de ser un sistema de bajo coste, ya que este tipo de cámaras tiene un coste relativamente elevado.

En la sección siguiente se describe brevemente el funcionamiento general de una cámara digital, para explicar su funcionamiento en cámaras web y cámaras de teléfonos móviles inteligentes, que son los dispositivos que se utilizan en nuestro problema, para comprender cómo obtener la imagen digital del usuario que se pose frente a la cámara en relación al estado del arte que se plantea en el presente capítulo.

En general, estos dispositivos ofrecen menores resoluciones espaciales y espectrales que una cámara comercial digital al uso, si bien con el avance de la tecnología y el uso en más campos hacen que cada vez mejores y que incluso lleguen a ofrecer resultados parecidos a una cámara digital comercial [9].

2.1.1. CÁMARA DIGITAL

Una cámara digital es un dispositivo que codifica los niveles de intensidad provenientes de la escena en función de las radiaciones espectrales recibidas por el sensor de color de la cámara.

Tanto las cámaras digitales en dispositivos con conexión a internet como las tradicionales poseen un sistema óptico, basado típicamente en una lente con un diafragma variable para enfocar la escena variando la entrada de luz, y por tanto de radiación, que incide en el sensor [10].

Los dos tipos principales de sensor de imagen digital son CCD y CMOS. Un sensor CCD tiene un amplificador para todos los píxeles, mientras que cada píxel en un sensor de pixel activo CMOS tiene su propio amplificador. En comparación con los CCD, los sensores CMOS consumen menos energía a la vez que por lo general precisan menores niveles de intensidad [11].

La resolución espacial de una cámara digital viene limitada por el tamaño del sensor de la cámara [12] que convierte las radiaciones espectrales en señales discretas. Por otra parte, en lo que respecta a los niveles de intensidad de la imagen, que en el caso de color se corresponde con los tres canales espectrales, normalmente RGB, cuanto mayor sea la magnitud de radiación proveniente de un punto dado de la escena, mayor será el valor de intensidad obtenido en el píxel correspondiente de la imagen.

Dependiendo de la estructura física del sensor, se utilizará una matriz de filtro de color, lo que requiere interpolación cromática para recrear una imagen a todo color, uno de tales filtros es el famoso de Bayer, de amplio uso en este tipo de dispositivos. El número de elementos en el sensor determina el "número de píxeles" de la cámara. En un sensor típico, el número de píxeles es el producto del número de filas por columnas. Por ejemplo, un sensor de 1000 por 1000 píxeles tendría 1.000.000 píxeles, o 1 megapíxeles.

2.1.2. CÁMARA WEB

Una cámara web es una cámara de vídeo que se alimenta de imagen en tiempo real a través de un ordenador o de una red informática.

Cuando se envía a un lugar remoto, el flujo de video se puede guardar, ver o enviarlo a otro lugar. Generalmente está conectada mediante un cable USB o similar. También puede estar integrada en el hardware del equipo, como ocurre en los ordenadores portátiles.

Las cámaras web incluyen una lente, un sensor de imagen, la electrónica de apoyo, y también puede complementarse con un micrófono para sonido.

Los sensores de imagen en cámaras web pueden ser también de tipo CMOS o CCD. La mayoría de las cámaras web son capaces de proporcionar una resolución VGA de vídeo a una velocidad de 30 fotogramas por segundo.

La electrónica de apoyo lee la imagen del sensor y la transmite al ordenador al cual está conectada.

2.1.3. CÁMARA DE UN TELÉFONO MÓVIL

Un teléfono móvil con cámara es capaz de capturar imágenes estáticas y vídeo. La mayoría de los teléfonos móviles actuales poseen una o dos cámaras (frontal y trasera).

Las cámaras de los teléfonos suelen ser más simples que las cámaras digitales independientes. Sus habituales lentes de foco fijo y sensores más pequeños limitan su rendimiento en relación a estas características, así como en lo que se refiere a la intensidad lumínica cuando ésta es escasa. .

Algunos teléfonos con cámara utilizan sensores de imagen CMOS, debido en gran medida a la reducción del consumo de energía en comparación con las cámaras de tipo CCD, que también se utilizan.

Las cámaras de los teléfonos inteligentes se utilizan como dispositivos de entrada en numerosos proyectos de investigación y aplicaciones comerciales. Un ejemplo de éxito comercial es el uso de los códigos QR adjuntos a los objetos físicos. Los códigos QR pueden ser detectados por el teléfono utilizando su cámara y proporcionar un enlace de acuerdo a los contenidos digitales relacionados, por lo general un URL. En esta línea otro enfoque puede ser el reconocimiento de objetos, a través de aplicaciones que se ejecuten desde el teléfono móvil que hayan sido construidas para la detección y reconocimiento de objetos. El análisis de imágenes basada en el contenido se utiliza para reconocer objetos físicos tales como carteles publicitarios para proporcionar información sobre el objeto [13]. La aplicación que se propone en el presente trabajo se plantea como el procesamiento de imágenes en remoto mediante accesos a través de internet donde residen los objetos de prueba, en nuestro caso, las gafas. No obstante, a la vista de los desarrollos sobre aplicaciones móviles, nada impediría que en el futuro los procesamientos de imágenes se pudieran realizar en el propio terminal, accediendo a internet para la descarga de los objetos a probar u otras necesidades, invirtiendo los roles remoto y terminal, siempre y cuando éste tuviese suficiente capacidad de proceso. Por consiguiente, el planteamiento que se formula bien pudiera adaptarse bajo esta perspectiva.

Además de la cámara trasera de costumbre, la mayoría de teléfonos actuales tienen una cámara frontal hacia el usuario para diversos propósitos tales como la videoconferencia y el autorretrato. La resolución de la calidad de una cámara frontal suele ser inferior a la cámara trasera o cámara principal. El avance de la tecnología está haciendo que las cámaras frontales de los móviles, sobre todo con el auge de los autorretratos, sean cada vez mejores.

2.1.4. PROPIEDADES DE LAS IMÁGENES EN EL PROCESO DE ADQUISICIÓN

En cuanto a la adquisición de la imagen se presentan distintos problemas como la resolución de la imagen, la posición del usuario frente a la cámara, la iluminación o el entorno.

2.1.4.1. RESOLUCIÓN

Debido a que las imágenes pueden ser captadas por distintos tipos de cámaras con distintas resoluciones se plantea el problema de qué resolución es la idónea para el probador que se plantea en este proyecto. En este caso es importante tener una buena resolución para que el usuario no perciba pérdida en la calidad de la imagen, si bien también hay que tener en cuenta un tamaño en bytes apropiado para ser transportado por la red sin que afecte al rendimiento del proceso desde el punto de vista de tiempo real. Por lo tanto, desde el punto de vista del probador virtual es necesario buscar una relación equilibrada y óptima entre calidad de imagen y tamaño de la misma.

2.1.4.2. TAMAÑO Y POSICIÓN DE LA CARA

En este sentido hay que tener en cuenta que el usuario puede colocarse a distintas distancias de la cámara, por lo tanto es necesario considerar qué distancias son válidas y cuáles no desde el punto de vista del probador virtual y considerando sus características de calidad y resolución. De esta manera, la

detección de la cara debe hacerse desde tamaños pequeños a tamaños de cara que ocupen toda la pantalla.

La pose del usuario frente a la cara es muy importante. La cara ha de estar recta y centrada. Las variaciones en los ángulos de la cara con respecto a la cámara hacen que la detección sea incorrecta. En las imágenes que se muestran en la Figura 1 y en la Figura 2 las posiciones que aparecen son consideradas como correctas e incorrectas respectivamente.



Figura 1. Ejemplo de pose válida

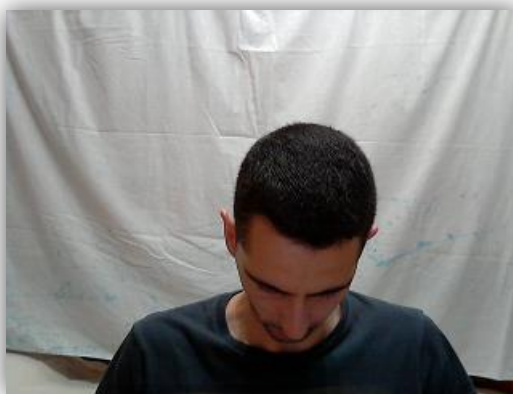


Figura 2. Ejemplo de pose no válida

En lo que respecta al probador virtual se recomienda al usuario seguir unas instrucciones básicas antes de proceder al envío de la imagen para su procesamiento y posterior superposición de las gafas. Estas recomendaciones se orientan a proporcionar al usuario instrucciones sobre cómo debe posicionarse correctamente. La detección automática de posiciones incorrectas o baja calidad de la imagen se plantea como propuesta de futuro a considerar para mejorar la eficiencia del probador virtual.

2.1.4.3. MODELO DE COLOR

Las imágenes se pueden capturar en diversos modelos de colores. Tal y como se especifica en el capítulo 5 de [14] existen diversos modelos de color:

- **Modelo RGB:** es el modelo más común orientado al hardware para monitores de color y una amplia gama de video cámaras.

- Modelo CMY: es un modelo utilizado para impresoras en color.
- Modelo YIQ: se utiliza sobre todo en TV en color.
- Modelo HSI: es una herramienta ideal para desarrollar algoritmos de procesamiento de imágenes basados en alguna de las sensaciones del sistema visual humano.

El más habitual en los dispositivos considerados es el RGB, siendo los demás útiles dependiendo del procesamiento posterior, existiendo funciones de transformación directas e inversas entre ellos.

2.1.4.4. ILUMINACIÓN

La iluminación en la imagen captada es un elemento a tener en cuenta, ya que una correcta iluminación proporciona mejores resultados en la fase de procesamiento. La iluminación determina el nivel de intensidad a la que se ven sometidos los dispositivos de captura (CCD, CMOS) y por tanto el nivel de impacto de las radiaciones correspondientes al espectro visible en las longitudes de onda pertenecientes al rojo, verde y azul.

Se pueden aplicar diferentes operaciones radiométricas sobre el histograma de una imagen para corregir su brillo y contraste:

- *Desplazamiento de histograma*: es utilizado para aclarar u oscurecer una imagen, pero manteniendo la relación entre los valores de niveles de gris. Esta operación puede llevarse a cabo por la simple adición o sustracción de un número fijo a todos los niveles de gris.
- *Contracción del histograma*: esta técnica no produce realmente un realzado de la imagen, por el contrario, el resultado es una disminución del contraste de la imagen.
- *Expansión del histograma*: es la operación opuesta a la contracción del histograma. Tiene el efecto de incrementar el contraste de una imagen de bajo contraste.
- *Igualación o ecualización del histograma*: es una de las técnicas más utilizadas para la mejora del contraste de la imagen original [14,15]. Realza la imagen original mediante una determinada transformación o modificación del histograma tendiendo a distribuir los niveles de intensidad a lo largo de todo el rango posible de valores de intensidad, de ahí su nombre de igualación o ecualización.
- *Realzado adaptativo del contraste*: en lugar de usar histogramas se puede realizar el realzado basado en otras propiedades de los píxeles incluidos en un entorno de la vecindad. Dos de estas propiedades que se usan frecuentemente por su importancia sobre el aspecto de la imagen son la media de intensidad y la varianza (o desviación estándar). Este hecho hace que el realzado dependa de la región donde se aplica y por tanto varíe en las diferentes partes de la imagen, de ahí el concepto de adaptativo.
- *Realzado en el dominio de la frecuencia*: en esta categoría destaca el conocido como filtrado homomórfico, que realiza un realzado mediante la utilización de un filtro paso alto para eliminar las componentes de baja frecuencia asociadas con las imágenes, sobre la base de que éstas están asociadas con la iluminación mientras que las altas frecuencias, que se mantienen, corresponden a las componentes de reflectancia. De esta forma, al eliminar las componentes de iluminación hace que los dispositivos no estén tan afectados por dicha componente, independizándolos de esta forma lo más posible del entorno respecto de la iluminación. .

- *Combinación de técnicas*: se trata de un algoritmo que combina varias de las operaciones definidas anteriormente, incluyendo filtrado paso alto y modificación de histograma.
- *Énfasis de alta frecuencia*: el filtrado de paso alto de forma aislada acentúa los bordes en la imagen pero pierde una gran parte de información visual al eliminar las componentes de baja frecuencia. Este problema se resuelve con el filtrado de énfasis de alta frecuencia, que conserva parte de la información de baja frecuencia.

2.1.4.5. ENTORNO

El entorno desde el cual se estén capturando las imágenes puede presentar problemas si no es el adecuado.

Un entorno adecuado es aquel que no contiene elementos que puedan dar lugar a confusión al procedimiento de detección de objetos, como es el caso de una cara cuando realmente no lo es, que es caso fundamental que nos ocupa.

En la Figura 3 se puede observar un entorno despejado, sin elementos en el fondo de la imagen que puedan interferir en la fase del proceso encargada de la detección de las caras, mientras que en la Figura 4 se comprueba cómo el entorno contiene elementos que producen falsos positivos. Por lo tanto, en esta fase de desarrollo del proyecto, si se ejecuta en un entorno cuyo fondo de imagen contenga diversos elementos resulta más problemática la detección que un entorno libre de elementos. Razón por la cual la captura de las imágenes debe tender hacia el primer tipo de escenarios para garantizar una mayor eficiencia en los resultados.



Figura 3. Ejemplo de imagen con entorno despejado



Figura 4. Ejemplo de imagen con entorno complejo

2.2. ENVÍO DE LAS IMÁGENES

El envío de las imágenes desde el dispositivo captura hacia la unidad de procesamiento, en nuestro caso, el servidor se hace a través de Internet con el protocolo HTTP. Este protocolo, aunque no fue diseñado para enviar y recibir peticiones en tiempo real, la aparición de estándares en las transmisiones entre los dispositivos de captura y el servidor, como por ejemplo el protocolo RTP (Real-time Transport Protocol), así como la mejora de las interconexiones hace que este problema se encuentre en la actualidad relativamente resuelto.

En cuanto al envío de imágenes hay que tener en cuenta que un factor crítico es el tiempo de respuesta, por lo tanto hay que encontrar la mejor relación entre tamaño de las peticiones enviadas y la resolución de la imagen a enviar. Si la resolución de la imagen es demasiado grande, generará gran tráfico y ralentizará el tiempo de respuesta. Las imágenes proporcionadas por los dispositivos de captura poseen una resolución relativamente baja y las razones de compresión de las mismas con estándares JPEG o JPEG2000 determinan que este hecho tampoco plantee problemas de cara a la consecución del requisito de tiempo real del probador virtual que se propone.

2.3. PROCESAMIENTO DE LA IMAGEN: RECONOCIMIENTO FACIAL

El servidor será el encargado de realizar el reconocimiento facial y superponer la imagen de las gafas en la imagen recibida.

La reducción en los costes en la relación precio/rendimiento, junto con los descensos en los costes de adquisición de imágenes de video, implica que los sistemas de visión por computador puedan ser implementados en sistemas de escritorio y sistemas embebidos [16, 17].

Llegados a este punto el problema principal que se plantea en relación al diseño del probador virtual es el reconocimiento facial para identificar sobre la imagen la cara o caras si existen varias y no sólo eso sino que también es necesario identificar la posición de los ojos sobre la cara identificada para determinar dónde posicionar correctamente las gafas.

El primer paso para cualquier sistema de procesamiento facial es detectar las localizaciones de las caras dentro de la imagen. Sin embargo, la detección facial a partir de una única imagen es una tarea difícil debido a la variabilidad en la escala según la distancia existente entre la cara y el dispositivo de captura, la ubicación de la cara en la imagen, la orientación (mirando hacia arriba/abajo, girada), y la posición de la cara (frontal, de perfil). La expresión facial, oclusión, y condiciones de iluminación también pueden cambiar considerablemente la apariencia general de las caras.

En este sentido, una definición formal de detección facial podría ser la siguiente: dada una imagen arbitraria, la meta de la detección facial es determinar si existe o no alguna cara o caras en la imagen, en cuyo caso se devuelve la ubicación de cada cara en la imagen con referencia a un punto fijo de la imagen, normalmente la esquina superior izquierda de la misma.

Los retos asociados con la detección facial, y que nos encontramos en el planteamiento del probador virtual, se pueden atribuir a los siguientes factores:

- *Pose*: las imágenes de una cara varían debido a la posición de la misma respecto a la cámara (frontal, posterior, girada hasta llegar al perfil o con mayor visibilidad de la parte posterior). Las variaciones pueden deberse también a determinadas circunstancias que hacen que características faciales tales como que un ojo o la nariz pueden ser parcial o totalmente ocultadas, por ejemplo por el pelo cuando éste cae sobre la cara.

- *Presencia o ausencia de componentes estructurales:* los rasgos faciales u otros elementos tales como barbas, bigotes y gafas ya superpuestas pueden a veces estar presentes en la imagen, existiendo además una gran variabilidad entre estos componentes, incluyendo la forma, color o tamaño.
- *La expresión facial:* la aparición de las caras se ven directamente afectadas por la expresión facial de una persona en función de sus características propias o estado anímico.
- *Oclusión:* las caras pueden estar parcialmente ocultas por otros objetos. En una imagen con un grupo de personas, algunas caras pueden ocultar parcialmente otras caras.
- *Orientación de la imagen:* las imágenes de rostros varían directamente para diferentes rotaciones alrededor del eje óptico de la cámara.
- *Condiciones de obtención de imágenes:* cuando la imagen se forma, factores como la iluminación (espectros, distribución de código fuente e intensidad) y las características de la cámara (respuesta del sensor, lentes) pueden afectar a la apariencia de una cara.

En esta primera fase de desarrollo del probador virtual el objetivo de la detección se centra en la identificación de una única cara de suerte que sobre ella es necesario identificar características faciales relevantes, más concretamente se trata de identificar la presencia y la ubicación de características esenciales, tales como los ojos, la nariz, fosas nasales, las cejas, la boca, los labios, las orejas [18, 19]. Si bien desde la perspectiva del probador virtual interesa no sólo detectar la cara, sino determinar la posición de los ojos sobre la misma.

Un estudio extenso sobre métodos de reconocimiento facial fue presentado por Samal e Iyengar [20]. Chellapa y col. [21] más recientemente realizaron un análisis comparativo sobre diferentes métodos de reconocimiento facial y algunos métodos de detección de características faciales [20]. Merece la pena constatar en este sentido, que se han utilizado diferentes métricas adoptado para evaluar la eficiencia de los algoritmos, tales como el tiempo de aprendizaje que algunos métodos requieren cuando se utilizan estrategias de este tipo, el número de muestras requeridas para el aprendizaje, tiempo de ejecución, o tasas de detección entre verdaderas y falsas detecciones.

Existen diferentes métodos para detectar una cara en una escena, algunas de ellas más complejas que otras. A continuación se proporciona un listado de los métodos y procedimientos más comunes utilizados en el proceso de detección de caras, según [22]:

- *Encontrar caras en imágenes con el fondo controlado:* bajo esta situación la detección resulta relativamente sencilla, ya que se utilizan imágenes con un fondo monocolor plano o con un fondo predefinido, ya que al eliminar el fondo los límites de la cara aparecen con cierta nitidez. El problema es que el fondo ha de ser siempre el mismo, por lo que este sistema no encaja con el objetivo del probador virtual universal, debido a que se obliga al usuario a utilizarlo siempre bajo el mismo escenario.
- *Encontrar caras por el color:* cuando las imágenes son en color, como son en la mayoría de los dispositivos considerados en el presente trabajo, se puede utilizar el color típico de la piel para encontrar segmentos faciales. No obstante, este método presenta una gran desventaja ya que no funciona adecuadamente con todo tipo de colores de la piel, y además no presenta un grado de robustez suficiente bajo diferentes condiciones de iluminación en la escena [23,24]. Como el objetivo es poder probarse las gafas en cualquier condición de luz y para todo tipo de personas, sin importar el color de piel, estas técnicas basadas exclusivamente en el color tampoco encajan en el probador virtual universal que se pretende.

- *Encontrar caras por el movimiento*: si se poseen dispositivos con capturas de imágenes de video en tiempo real se puede utilizar el hecho de que en la escena una cara puede encontrarse en movimiento. Bastaría con determinar el área en movimiento para identificar la cara. La gran desventaja de estas aproximaciones surge por la existencia de otros objetos que se mueven en la escena. Dado que en nuestro escenario nos podemos encontrar otros objetos en movimiento mientras se está realizando el hecho de probarse las gafas, este método no es el más idóneo para el probador virtual desde esta perspectiva. Pero aún más, dado que en general durante la prueba de las gafas, la cara permanecerá de alguna manera más o menos estática, el planteamiento basado en detección de movimiento pierde su esencia.
- *Usando una mezcla de encontrar caras por color y por movimiento*: la combinación de varios enfoques normalmente generan mejores resultados que cuando se utilizan los métodos de forma independiente, aunque bajo este planteamiento las desventajas individuales señaladas previamente se siguen manteniendo [25].
- *Encontrar caras en escenas sin restricciones*: en esta categoría se encuentran los métodos con mayor amplitud de miras y por tanto más complejos. El planteamiento general consiste en dada una imagen en escala de gris o color determinar la ubicación de la cara mediante mecanismos de detección basados en procedimientos próximos al razonamiento humano, sobre la suposición de que cuando se muestra una imagen conteniendo caras a una persona, ésta no tiene grandes dificultades en su identificación. Es en este punto donde se establece la conexión entre los procesos de percepción computacional (visión artificial) y aprendizaje automático bajo el paradigma de la inteligencia artificial. Algunas de las técnicas más usadas bajo esta consideración son:
 - Redes Neuronales del tipo retro-propagación.
 - Redes Neuronales utilizando información estadística para la identificación por agrupamiento.
 - Identificación facial basada en modelos.
 - Detección facial en tiempo real de detección utilizando coincidencias de orientación de borde [26].
 - Detección facial robusta utilizando la distancia de Hausdorff [27].
 - Cascadas de clasificadores "débiles": un avance importante en la detección facial surgió con la propuesta del algoritmo de Viola y Jones [1]. El uso de una cascada de clasificadores "débiles", usando características simples de Haar [28], puede, después de un entrenamiento exhaustivo, proporcionar unos resultados aceptables en cuanto a rendimiento relativo a tasa de éxito y tiempo de ejecución. El enfoque que se plantea en este algoritmo es ampliamente utilizado para la detección facial. Dentro de la categoría relativa a los métodos sin restricciones éste enfoque resulta adecuado para nuestro cometido, cuyo rendimiento desde el punto de vista del tiempo computacional resulta a su vez apropiado.

2.4. DEVOLUCIÓN DE LA IMAGEN

Una vez procesada la imagen mediante las técnicas de reconocimiento facial y la superposición de las gafas en la posición correcta, el resultado debe devolverse al dispositivo desde el cual se ha capturado la imagen original. Una vez transferido el resultado, éste debe mostrarse en la pantalla del mismo para su

visualización. Este último paso no entraña dificultad alguna, ya que cualquiera de los dispositivos utilizados en el probador virtual poseen esta capacidad suficientemente desarrollada.

Para crear la sensación de imagen en tiempo real hay que tener en cuenta el fenómeno *phi* [29], que se trata de una ilusión óptica de nuestro cerebro que permite percibir movimiento continuo sobre una sucesión de imágenes. El servidor nos devolverá imágenes cada cierto tiempo y eso creará la sensación del pretendido tiempo real. Cuanto menor sea el tiempo entre cada envío de imágenes, mayor será la sensación de movimiento continuo.

Es interesante determinar qué cantidad de imágenes por segundo (*frame rate*) se deben enviar al servidor y recibir desde éste para crear la sensación de movimiento, al igual que se hace en la generación de una película. Aunque es complicado determinar cuántas imágenes por segundo puede percibir el ojo humano [30], utilizaremos el valor de 24 imágenes por segundo (*frames per second, fps*) que se utiliza como estándar en la industria del cine.

En las comunicaciones cliente-servidor la tasa de transferencia (*bit rate*) es un término utilizado para medir la velocidad de transmisión de datos entre el cliente y el servidor. Es importante considerar que la tasa de transferencia en nuestra arquitectura debe ser soportada por la red y por el servidor donde se encuentre alojada la aplicación. Por lo tanto, el ancho de banda debe ser suficiente para enviar y recibir las 24 imágenes por segundo.

Una vez realizado el ciclo completo de captura, envío, procesamiento, recepción y visualización del resultado, el proceso ha de iniciarse de nuevo para realizar un nuevo ciclo completo. Es importante que el inicio de un nuevo ciclo no se demore en exceso para no perder la perspectiva de tiempo real desde el punto de vista del usuario.

3. DESCRIPCIÓN DEL MÉTODO PROPUESTO

3.1. PROPUESTA GENERAL

Una vez establecidos los criterios y requisitos necesarios para el desarrollo del probador virtual de gafas y teniendo en cuenta la revisión sobre el estado del arte en los diferentes aspectos involucrados, el diseño propuesto consta de las etapas mencionadas previamente que se sintetizan en las fases representadas en la Figura 5 e identificadas mediante la correspondiente etiqueta numérica, según el ciclo de proceso global establecido al efecto.

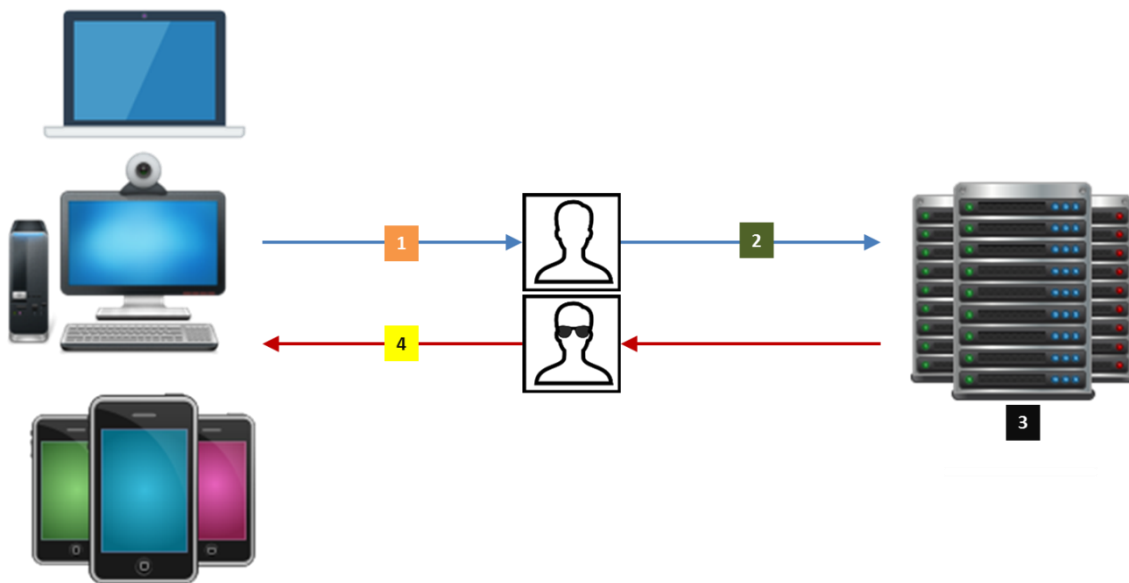


Figura 5. Diseño general del probador virtual

1. Adquisición de imágenes del dispositivo con cámara.

Mediante un navegador web se obtiene la imagen que en ese mismo momento está enfocando la cámara del dispositivo. También se selecciona la imagen de las gafas que se quiere solapar sobre la imagen captada. La imagen de las gafas se encuentra alojada en el servidor, junto con una serie de parámetros que la caracterizan.

2. Envío de la imagen al servidor.

La imagen captada, junto con la información de las gafas seleccionadas, se envía al servidor a través de Internet.

3. Procesamiento de la imagen recibida y fusión del modelo de gafas seleccionado.

Se detectan las caras que contienen la imagen recibida. Esta detección se realiza mediante el algoritmo de Viola y Jones [1]. Una vez detectadas las caras se procede a solapar la imagen de las gafas en la posición correcta. Se supone que en cada momento es un único usuario el que está realizando la prueba virtual por lo que sólo debe existir una única cara, si bien se contempla la posibilidad de que en el escenario puedan aparecer más de una cara por diferentes razones, tales como la presencia de otras personas o muñecos en las proximidades. En este sentido, durante la

fase de pruebas se analizan estas posibilidades, al objeto de establecer distintas posibilidades de aplicación del método seleccionado.

4. Devolución al dispositivo con cámara las imágenes transformadas.

La imagen transformada se devuelve al navegador web, que está escuchando peticiones del servidor referidas a este evento, momento en el que la imagen procesada se muestra en la pantalla del dispositivo del usuario.

3.2. ADQUISICIÓN DE IMÁGENES Y ARQUITECTURA CLIENTE-SERVIDOR EN TIEMPO REAL

La adquisición junto al envío y devolución de las imágenes constituye también una parte importante del probador virtual.

Es muy importante que este proceso se realice lo más rápido posible para que se cree la apariencia de tiempo real para el usuario. Por lo tanto hay que buscar siempre las soluciones más óptimas tanto para la captura como para el envío. Para ello, desde el punto de vista del diseño, a continuación se describen los aspectos relacionados con la resolución de la imagen, la distancia de la cara a la cámara, el modelo de color, la iluminación, el entorno o escenario de captura, además del envío de imágenes a través de internet, la tasa de transferencia o la arquitectura cliente-servidor.

3.2.1. RESOLUCIÓN

Las imágenes se capturan en una resolución de 320 píxeles de ancho por 240 píxeles de alto. Esta resolución nos permite obtener dos ventajas:

- La correcta visualización en dispositivos con un tamaño de pantalla pequeño.
- El tamaño en bytes de la imagen es el adecuado para no sobrecargar la red en las transferencias de datos.

3.2.2. DISTANCIA DE LA CARA A LA CÁMARA

El algoritmo utilizado se ha configurado con la finalidad de poder detectar cualquier objeto siempre y cuando haya sido entrenado previamente para dicho objeto. El algoritmo proporciona dos posibilidades de detección: facial o detección de ojos. Según el caso, los objetos de interés en la detección deben aparecer en la imagen según se indica a continuación:

- Si se utiliza la detección facial, la cara debe presentarse completa y de forma frontal a la cámara.
- Si se utiliza la detección de ojos, éstos deben aparecer en la imagen, también de forma frontal, existiendo la posibilidad de detectarlos tanto con gafas como sin ellas. Por lo tanto, en este caso sólo hace falta que aparezcan los ojos, pudiendo aparecer la cara de forma incompleta en la imagen.

Para ambas detecciones existe un tamaño mínimo, que es de 20 píxeles de ancho por 20 píxeles de alto, a partir del cual no se detectará la cara o los ojos, en cualquier caso el tamaño máximo vendrá determinado por el máximo tamaño de la imagen.

3.2.3. MODELO DE COLOR

En este caso las imágenes capturadas por las cámaras web o cámaras de móviles se obtienen en el modelo de color RGB. La imagen ha de adquirirse en color para poder devolverla más tarde con la misma gama de colores con la que se ha capturado.

En este sentido podría pensarse en capturar la imagen directamente en escala de grises, ya que el método que se utiliza para su procesamiento tiene como entrada una imagen en dicha escala. Si bien, esta solución no parece la más apropiada porque al devolver la imagen ésta se visualizaría en escala de grises, lo cual haría que el usuario no apreciara los colores reales, distanciándose del objetivo que se pretende con el probador virtual que es su aproximación a la realidad en el sentido de que el usuario pueda recibir una sensación lo más próxima a la percibida en una tienda convencional cuando se realiza una prueba real con las gafas, tanto en tamaño con respecto a su cara como en lo que se refiere a la combinación de colores de los distintos modelos de gafas.

3.2.4. ILUMINACIÓN

El método de detección propuesto se basa en un proceso de aprendizaje, basado en clasificadores, que utiliza una base de muestras de imágenes de entrenamiento para determinar los parámetros de aprendizaje pertinentes.

Los clasificadores utilizados para la detección facial y detección de ojos están entrenados con imágenes con distintos niveles de iluminación (desde muy baja hasta muy alta). En este sentido, el entrenamiento se ha realizado mediante imágenes captadas bajo diferentes niveles de iluminación en un entorno de interior con iluminación natural y artificial. Además, se han aplicado correcciones de brillo y contraste mediante las técnicas de modificación del histograma, mediante las técnicas propuestas en la sección 2.1.4.4. En este sentido, sobre una misma imagen original se aplican correcciones de histograma generando al menos cuatro variantes de la misma. Una oscura y otra clara aplicando técnicas de desplazamiento del histograma hacia los niveles bajo y alto del mismo respectivamente. La tercera variante se obtiene mediante realzado de contraste, aplicando técnicas propias de esta operación. Finalmente, la cuarta variante se obtiene mediante la aplicación de un filtrado homomórfico con el fin de eliminar la componente de iluminación respecto de la de reflectancia. Para determinadas imágenes, seleccionadas de forma aleatoria, se aplican varios métodos de realzado con el fin de obtener un mayor número de variantes para el entrenamiento.

Por consiguiente, durante la fase de procesamiento de la imagen en el servidor no se realiza ninguna corrección de brillo y/o contraste.

3.2.5. ENTORNO

En principio, el entorno de la imagen puede ser cualquiera, ya que sólo nos centraremos en las caras u ojos que aparecen en la escena. Aunque hay que tener en cuenta que para una mayor precisión en la detección sólo debería estar presente el usuario o usuarios que se quieran probar las gafas. Cuanta menor complejidad posea el escenario mayores posibilidades de éxito existen. Esto es, se debe evitar la existencia de objetos que puedan dar lugar a confusión de ser candidatos a ser reconocidos como caras u ojos, tales como fotos de caras, o personas ajenas al usuario o usuarios que pretenden realizar la prueba.

3.2.6. ENVÍO DE IMÁGENES A TRAVÉS DE INTERNET

Una vez adquirida la imagen procedente del escenario en cuestión, junto con el modelo de gafas que el usuario desea "probarse", se envía dicha información al servidor a través de Internet, ya que el probador se fundamenta en un navegador web y la imagen se procesa en un servidor remoto, por lo que la red de Internet permite realizar esa transferencia.

En este paso es crítico el tamaño en bytes de la información enviada. Cuanto menor sea éste, más peticiones se podrán hacer por segundo y mayor sensación de tiempo real percibirá el usuario.

3.2.7. TASA DE TRANSFERENCIA

Las imágenes se transmiten a través de la red, por lo que el tamaño en bytes de las mismas debe ser el menor posible, pero sin perder calidad en la transferencia.

Para ello, las imágenes se comprimen con el estándar JPEG (Joint Photographic Experts Group), que es un método de compresión con pérdida estandarizado por la ISO. La cantidad de compresión se puede ajustar para lograr el equilibrio deseado entre el tamaño de archivo y la calidad visual. [31]. En el diseño que se propone se utiliza este estándar, pero sin pérdida, para que la imagen recibida por el servidor sea lo más fiel posible a la captada por la cámara, lo que permite minimizar los errores en la detección por pérdida de información, a la vez que el tamaño permite una razón de transferencia razonable

En efecto, el modelo de color que se utiliza para la representación de las imágenes captadas es RGB, con una representación de 24 bit (3 bytes por píxel).

Teniendo en cuenta la resolución de las imágenes captadas (320x240) y la información del color representada por 3 bytes, se puede hacer un cálculo sencillo del tamaño total de la imagen en bytes (sin compresión):

$$320 \text{ píxeles} \times 240 \text{ píxeles} \times 3 \text{ bytes por píxel} = 230400 \text{ bytes} = 28,125 \text{ KBytes}$$

Por lo tanto, el tamaño máximo de las imágenes utilizadas en las pruebas será de 28,125 KB.

Como se ha indicado previamente, para mantener la sensación de percepción continua, se ha fijado en 24 la cantidad de imágenes por segundo a mostrar, por lo tanto:

$$28,125 \text{ KB} \times 24 \text{ imágenes por segundo} = 675 \text{ KB por segundo}$$

Por consiguiente, la tasa de transferencia mínima requerida resulta ser de 675 KB/s.

3.2.8. ARQUITECTURA CLIENTE-SERVIDOR PARA TIEMPO REAL

La arquitectura seleccionada para el tratamiento de las imágenes captadas por la cámara es una arquitectura cliente-servidor. Sin embargo, no es una estructura cliente servidor común, ya que debe ser en tiempo real para poder hacer *streaming* de video, esto es, que se envíen imágenes desde el cliente al servidor y se reciban desde éste con tasas de transferencia con sensación de continuidad, tal y como se ha especificado previamente. A diferencia de la arquitectura clásica cliente servidor, donde el tiempo de respuesta del servidor no es crucial, en las arquitecturas en tiempo real este tiempo es crítico.

En nuestro caso nos apoyaremos en la arquitectura cliente servidor en la web, más específicamente en el concepto de web en tiempo real. La web en tiempo real comprende un conjunto de tecnologías y prácticas que permiten a los usuarios recibir la información tan pronto como se publique en el servidor,

en lugar de requerir que éstos o su software comprueben periódicamente que la fuente ha actualizado su contenido. De esta manera, en el instante que el servidor recibe una imagen, ésta se envía procesada al cliente de forma inmediata para su visualización por el usuario.

El protocolo utilizado es HTTP, ya que, aunque inicialmente no fue un protocolo orientado a tiempo real, con el paso del tiempo y las mejoras tecnológicas y de conexión se ha ido mejorando para poder realizar estas peticiones bajo los requisitos de tiempo real. Aunque exige algunas modificaciones y consideraciones, como es el uso de la tecnología *WebSocket* para proporcionar un canal de comunicación bidireccional sobre un único socket TCP, siendo éste el protocolo que mejor se ajusta a nuestros requerimientos.

3.3. RECONOCIMIENTO FACIAL

3.3.1. TRANSFORMACIÓN A ESCALA DE GRISES DE LA IMAGEN RGB

El primer paso al recibir la imagen en el servidor es aplicar una transformación desde el modelo de color RGB a escala de grises de la imagen recibida. Este paso es necesario para, posteriormente, aplicar la detección facial mediante el algoritmo de Viola y Jones [1].

Exactamente se aplica la transformación desde RGB a YIQ [14], para obtener la imagen de reflectancia según la siguiente expresión donde R, G y B son los valores de las respectivas componentes de color e Y es el resultado de la transformación a gris.

$$Y = R \times 0.299 + G \times 0.587 + B \times 0.114$$

3.3.2. ALGORITMO DE VIOLA Y JONES PARA LA DETECCIÓN FACIAL

Este algoritmo se enmarca exactamente dentro de los métodos de reconocimiento de patrones basados en la apariencia [14]. Estos métodos utilizan las imágenes de los objetos mismas en lugar de propiedades. Por consiguiente, para ello se dispone de una base de datos donde se tienen almacenadas las imágenes de los objetos. Para el reconocimiento de un objeto concreto se dispone de un conjunto de imágenes sobre las posibles apariciones de dicho objeto. El conjunto de imágenes se genera idealmente, a partir del mayor número posible de puntos de vista y bajo el máximo número de direcciones posibles de iluminación y correcciones de intensidad por modificación de histograma según los planteamientos formulados en las secciones previas.

El *framework* que proponen Viola y Jones [1] se construyó para proporcionar una detección de objetos robusta y eficiente desde el punto de vista computacional. Esta última capacidad le distingue de otros enfoques de detección de caras existentes en la literatura, tal y como se ha expresado en el capítulo dos.

El sistema propuesto explora la imagen de entrada mediante una sub-ventana capacitada para detectar características. Estas características son las que permitirán distinguir si la imagen contiene o no una cara o los ojos de la cara según el tipo de detección que se pretenda, ya que esas características definen las propiedades similares o diferentes propias de las caras u ojos a detectar.

El *framework* de detección de objetos consta de tres etapas fundamentales, que se explican posteriormente más en detalle:

- La primera es una nueva forma de representación de la imagen llamada **imagen integral** que permite la evaluación de características de manera eficiente y rápida.

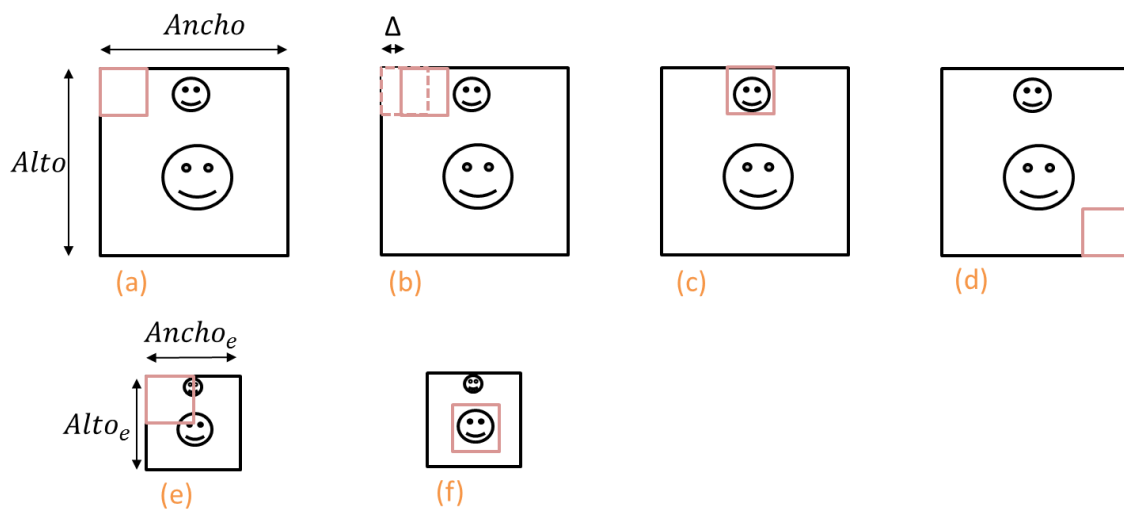
- La segunda se trata de un método para construir un **clasificador** seleccionando un pequeño número de características importantes utilizando AdaBoost [32,33,34].
- La tercera es un método para combinar sucesivamente **clasificadores** más complejos en una estructura **en cascada**.

De forma resumida, el funcionamiento general del algoritmo es el siguiente:

1. Se calcula la **imagen integral** de la imagen original. Este paso simplifica el cálculo de la suma de píxeles utilizado en el siguiente paso.
2. Se crea una **sub-ventana** y se analiza su contenido utilizando las **características** definidas que se calculan sobre ella. Esta sub-ventana posee siempre las mismas dimensiones y nos servirá para determinar si dentro de ella existen caras u ojos de diferentes tamaños, ya que será la imagen la que cambie de tamaño, no la sub-ventana.
3. Se utiliza AdaBoost para seleccionar las características más representativas para la sub-ventana. En este paso se utiliza el concepto de **cascada de clasificadores**, ya que primero se aplica un conjunto pequeño de características sobre la imagen integral. Si pasa esa primera etapa, se considera un segundo conjunto más grande de características y así sucesivamente hasta completar todas las etapas de uso de las diferentes características. Si completa todas las etapas se puede concluir que la sub-ventana explorada contiene el objeto que se intenta detectar. A cada uno de estos clasificadores por etapas se les da el nombre de clasificadores “débiles” y la suma de todos ellos conforman el clasificador “fuerte”.

Este proceso se repite desplazando la sub-ventana por toda la imagen. Una vez completado el recorrido total de la imagen, se repite el proceso mediante un re-escalado de la imagen para detectar caras (u objetos) más pequeños. El proceso termina cuando el tamaño de la imagen es igual al tamaño de la sub-ventana.

En la siguiente imagen se muestra un ejemplo ilustrativo de funcionamiento del desplazamiento de la sub-ventana y el re-escalado de la imagen.



De la fase (a) a la fase (b) se produce un movimiento de la sub-ventana con valor Δ . En la fase (c) se encuentra una cara que se sitúa dentro de los límites de la sub-ventana. En la fase (d) se ha recorrido toda la imagen. La fase (e) muestra el re-escalado de la imagen con los siguientes valores:

$$Ancho_e = \frac{Ancho}{factor\ de\ escala}$$

$$Alto_e = \frac{Alto}{factor\ de\ escala}$$

Este **factor de escala** es uno de los parámetros a tener muy en consideración ya que será uno de los que determina la precisión y velocidad del algoritmo.

La fase (f) muestra cómo al re-escalar la imagen, la otra cara contenida en la imagen será detectada sin haber cambiado el tamaño de la sub-ventana.

3.3.2.1. IMAGEN INTEGRAL

Las características pueden ser calculadas rápidamente usando una representación intermedia de la imagen llamada imagen integral. La imagen integral en la localización espacial con coordenadas de píxel (x, y) contiene la suma de los píxeles desde el origen $(0,0)$ hasta la posición (x, y) , Figura 6:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y'),$$

donde $ii(x, y)$ representa el valor del píxel (x, y) en la imagen integral e $i(x, y)$ es el valor de gris en la imagen original y en la misma posición de píxel.

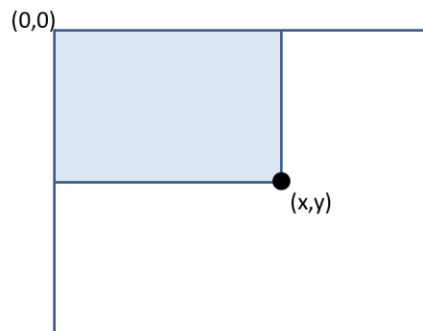


Figura 6. Región de la imagen integral

Usando el siguiente par de ecuaciones se obtiene la imagen integral en una única pasada sobre la imagen original:

$$s(x, y) = s(x, y - 1) + i(x, y) \tag{1}$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \tag{2}$$

donde $s(x, y)$ representa la suma acumulativa de fila, $s(x, -1) = 0$ con $ii(-1, y) = 0$.

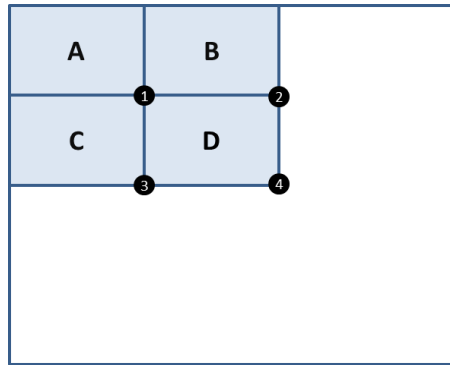


Figura 7. La suma de los píxeles dentro del rectángulo D puede ser calculada con cuatro puntos de referencia de la matriz. El valor de la imagen integral en la posición 1 es la suma de los píxeles en el rectángulo A. El valor de la posición 2 es A+B, en la posición 3 es A+C y en la posición 4 es A+B+C+D. La suma dentro de D puede ser calculada como $4 + 1 - (2 + 3)$

En una imagen integral, el área para cualquier región rectangular en la imagen se puede calcular mediante el uso de sólo cuatro accesos a la matriz. El siguiente ejemplo ayuda a comprender este mecanismo [35]:

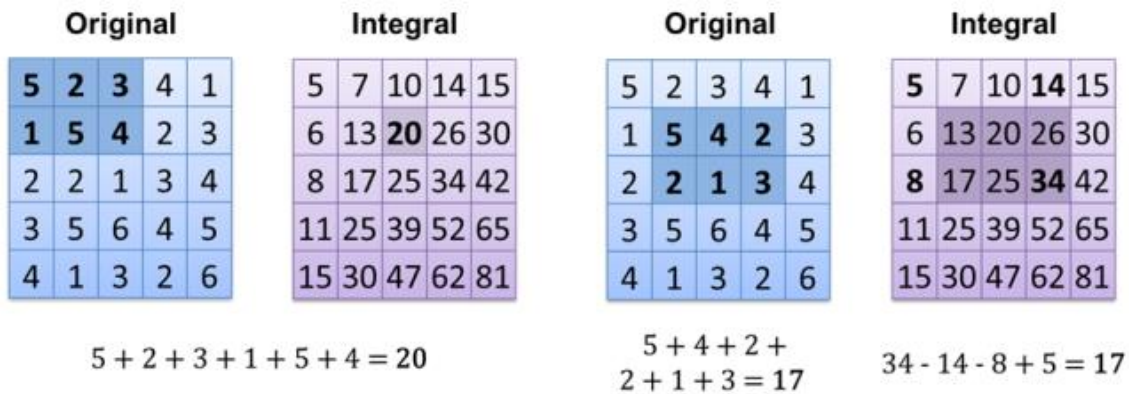


Figura 8. Ejemplo de creación de la imagen integral a través de la imagen original. El primer caso es la creación de la imagen integral en base a la frontera y la segunda en cualquier otra posición interior.

Las matrices azules representan las imágenes originales, mientras que los púrpuras representan las imágenes después de la transformación integral. Si tuviéramos que calcular el área sombreada en la primera imagen, tendríamos que haber sumado todos los píxeles de forma individual, obteniendo el resultado de 20 después de 6 accesos a memoria. Utilizando la imagen integral, todo lo que se necesita es un único acceso, si bien sólo en este caso por estar situados exactamente en la frontera. En caso de que no estar en la frontera, todo lo que se requiere es un máximo de cuatro accesos a la matriz, independientemente del tamaño de la región. Por lo tanto se produce una reducción efectiva de la complejidad computacional de $O(n)$ a $O(1)$. Se requerirá sólo dos restas y una suma para recuperar la suma de la zona sombreada en la imagen de la derecha, tal y como se expresa en la siguiente expresión:

$$\sum_{x=x_b}^{x_b} \sum_{y=y_b}^{y_b} i(x, y) = ii(x_b, y_b) + ii(x_{a-1}, y_b) - ii(x_b, y_{a-1}) + ii(x_{a-1}, y_{a-1}) \quad (3)$$

3.3.2.2. CARACTERÍSTICAS

Una característica se define como algo que se espera que proporcione suficiente información en el proceso de decisión del clasificador.

Una característica representa diferencias entre los valores de intensidad (escala de grises) entre dos o más áreas rectangulares adyacentes en una imagen.

El procedimiento de Viola y Jones clasifica imágenes en base al valor de características simples, esto es a nivel de píxel. Existen diversas razones para el uso de características en lugar de los píxeles directamente. El motivo más común es que las características pueden actuar para codificar el conocimiento *ad hoc* que sea difícil de aprender utilizando una cantidad finita de datos de entrenamiento. En el caso del probador que se plantea existe otra motivación crítica para utilizar características, ya que los sistemas basados en éstas operan con mayor rapidez que los basados en píxeles.

Motivado en parte por el trabajo de Papageorgiou y col. [36] el método propuesto por Viola y Jones [1] no trabaja directamente con las intensidades de las imágenes. Se utiliza un conjunto de características que recuerdan a las funciones base de la transformada 2D de wavelets dentro de la familia de Haar. Un ejemplo de cuatro características utilizadas en la detección facial se muestra en la Figura 9. El valor de una característica obtenida a partir de dos rectángulos (*two-rectangle feature*) se obtiene calculando la diferencia entre la suma de los píxeles dentro de las dos regiones rectangulares. Las regiones tienen el mismo tamaño y forma y son adyacentes horizontal y verticalmente (Figura 9). La característica de tres rectángulos (*three-rectangle feature*) calcula la suma dentro de los dos rectángulos externos, cuyo resultado se resta de la suma obtenida en un rectángulo central. Por último la característica de cuatro rectángulos (*four-rectangle feature*) calcula la diferencia entre pares diagonales de los rectángulos.

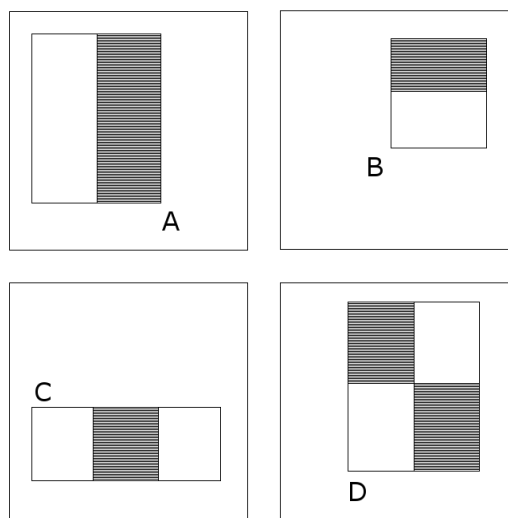


Figura 9. Ejemplo de características rectangulares mostradas relativas a la ventana principal de detección. La suma de los píxeles que se encuentran dentro de los rectángulos blancos se resta de la suma de los píxeles en los rectángulos grises. Características de dos rectángulos (*two-rectangle feature*) se muestran en (A) y (B). La figura (C) muestra una característica de tres rectángulos (*three-rectangle feature*), y (D) una característica de cuatro rectángulos (*four-rectangle feature*).

De esta manera, la función que permite obtener el valor de una característica, para una imagen integral en la posición de píxel x e y es la siguiente,

$$f(x, y) = \sum_i p_n(i) - \sum_i p_b(i) \quad (4)$$

donde $\sum_i p_n(i)$ es el sumatorio del valor de los píxeles dentro de la zona sombreada de la zona definida por la característica y $\sum_i p_b(i)$ es el sumatorio del valor de los píxeles dentro de la zona blanca de la característica.

Esta función nos permitirá determinar si la zona que se está explorando en un determinado instante se puede considerar como candidata a ser una cara u objeto de interés.

Para ilustrar el funcionamiento de las características se expone el siguiente ejemplo, Figura 10. Consideremos que colocamos una de las características definidas anteriormente sobre una imagen. El valor de la característica sería el resultado de la suma de todos los píxeles de intensidad en el lado blanco del rectángulo, la suma de los píxeles en las partes sombreadas del rectángulo, para después calcular su diferencia.



Figura 10. Ejemplo de aplicación de algunas características para el reconocimiento facial sobre una sub-ventana

3.3.2.3. CLASIFICACIÓN

Antes de definir el proceso de clasificación basado en las características a continuación se describe brevemente en qué consiste un proceso de clasificación de naturaleza estadística en general.

Según el capítulo 14 de [14] una muestra de entrada $x = \{x_1, x_2, \dots, x_d\}$ necesita ser clasificada en uno y sólo uno de los c grupos o clases c_1, c_2, \dots, c_c . La existencia de los grupos se conoce a priori. El vector de entrada x representa atributos de alguna característica de la imagen y además se desconoce la clase a la que pertenece. Supongamos que la variable y identifica la clase a la que pertenece, de modo que $y = c_j$ significa que pertenece a la clase c_j . La clasificación se refiere a la relación entre la etiqueta de pertenencia a las clases etiquetadas como y y el vector de características x . El objetivo es estimar la relación $x \rightarrow y$ utilizando los datos de entrenamiento $(x_i, y_i) = 1, \dots, n$. Esta relación llamada regla de decisión se utiliza luego para clasificar muestras futuras. Tanto las muestras de entrenamiento como las muestras futuras pertenecen a la misma distribución estadística desconocida.

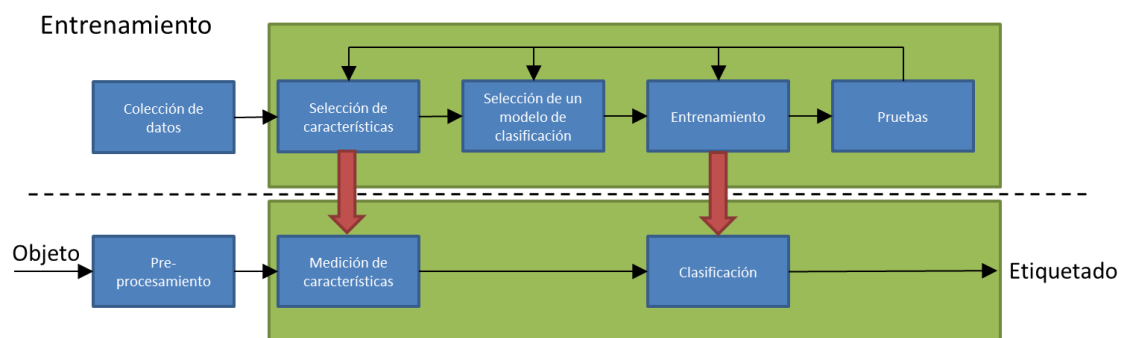


Figura 11. Fases de reconocimiento

Un sistema de reconocimiento tiene dos fases: el entrenamiento y la clasificación (Figura 11). En el entrenamiento, se propone un modelo para el clasificador y los parámetros desconocidos del modelo se estiman a partir de un conjunto de objetos de entrenamiento etiquetados. Los objetos se describen mediante un conjunto de características. Las características pueden no ser igualmente relevantes, algunas de ellas pueden ser importantes sólo en relación con las demás y algunas pueden ser sólo ruido en un contexto o aplicación particular. Por lo tanto, la función de selección se utiliza para mejorar la calidad de la descripción. La selección de características, entrenamiento y prueba de un modelo clasificador forman el núcleo del reconocimiento supervisado de patrones. El ciclo, que se muestra en la parte superior de la Figura 11, se puede realizar varias veces, cambiando las características, los parámetros o incluso el modelo clasificador hasta que se alcance una solución satisfactoria [33,34].

En la clasificación, el sistema de reconocimiento entrenado previamente asigna el objeto de entrada a una de las clases conocidas, basado en las características medidas. El papel del módulo de pre-procesamiento que aparece en la Figura 11 consiste en segmentar el patrón del fondo, esto es eliminar el fondo para quedarse con el patrón que contiene el objeto de interés, eliminar el ruido, normalizar el patrón y realizar cualquier otra operación necesaria para definir una representación compacta del objeto de interés, en nuestro caso la cara o los ojos.

En el algoritmo de Viola y Jones [1] se utiliza una versión de la clasificación llamada “en cascada”. La clasificación en cascada es en realidad una estrategia para combinar clasificadores cada vez más complejos en diferentes etapas en una estructura escalonada que incrementa la velocidad del detector centrando la atención en regiones prometedoras de la imagen, es decir, regiones que pueden ser candidatas a contener una cara. El procesado más complejo está reservado sólo para estas regiones.

El problema es cómo seleccionar esas características en cada etapa. Aquí es donde entra en juego AdaBoost [32], que para el caso que nos ocupa resulta ser un método de aprendizaje efectivo y de alto rendimiento. Dentro de cualquier sub-ventana de una imagen el número total de características de tipo Haar es muy grande, mucho más grande que el número de píxeles. Con el fin de garantizar una clasificación rápida, el proceso de aprendizaje debe excluir una gran mayoría de las características disponibles, y se centran en un pequeño conjunto de características que resultan críticas o relevantes.

3.3.2.3.1. ADABOOST ADPATADO

Adaboost es un algoritmo de *boosting* (aceleración) presentado en el contexto de la aplicación que se propone por Freund y Sachapire [32]. En este *framework* se utiliza una variante de AdaBoost [33,34] tanto para elegir un pequeño conjunto de características como para entrenar al clasificador.

En su forma original, el algoritmo de aprendizaje AdaBoost se utiliza para aumentar el rendimiento de la clasificación de un algoritmo de aprendizaje sencillo o regla. Esto se hace mediante la combinación de una colección de funciones de clasificación “débiles” para formar un clasificador “fuerte”. La idea subyacente es obtener un clasificador más preciso combinando clasificadores menos precisos que el producto final. En el lenguaje de *boosting*, se llama clasificador “débil” al algoritmo de aprendizaje simple. El algoritmo se llama “débil” porque no se espera de él la mejor precisión en la clasificación en contraposición a la combinación de clasificadores que la mejora, de ahí el nombre de clasificador fuerte.

Para acelerar el proceso de aprendizaje del clasificador “fuerte”, se le pide que resuelva una secuencia de problemas de aprendizaje. Después de la primera ronda de aprendizaje, los ejemplos difíciles son reponderados con el fin de hacer hincapié en las que se clasifican de forma incorrecta por el clasificador “débil” anterior. El clasificador “fuerte” final puede tomar la forma de un perceptrón, en el sentido de ser una combinación ponderada de clasificadores “débiles” seguidos de un umbral, que determina la clasificación.

Las garantías formales previstas por el procedimiento de aprendizaje AdaBoost [32] son mejores que las de los clasificadores individuales. Freund y Schapire [32] demostraron que el error de entrenamiento del clasificador “fuerte” se aproxima a cero de manera exponencial con el número de rondas. Se ha de tener en cuenta el problema general de la aceleración, en el que se combina un vasto conjunto de funciones de clasificación utilizando un voto ponderado de la mayoría. El reto consiste en asociar un peso significativo a cada función de clasificación que proporcione buenos resultados y un peso más pequeño a las funciones cuyos resultados sean peores.

3.3.2.3.1.1. CLASIFICADOR DÉBIL

Dado un conjunto de características y un conjunto de imágenes positivas y negativas de entrenamiento, se podría utilizar cualquier número de métodos de aprendizaje automático para estimar los parámetros de una función de clasificación o decisión en la fase de aprendizaje. En su forma original, el clasificador débil en AdaBoost [32] es usado para aumentar el rendimiento de clasificación de un simple algoritmo de aprendizaje.

En el caso relativo al problema de detección de caras, el clasificador débil se diseña para decidir si una determinada característica se ajusta a la imagen sobre la que se está tratando de identificar la cara y en su caso los ojos. Un clasificador “débil” $h_{débil}(x, y)$ consiste en una característica f_j , un umbral θ_j y una polaridad p_j indicando la dirección del signo de la desigualdad,

$$h_{débil}(x, y) = \begin{cases} 1 & \text{si } p_j f_j(x, y) < p_j \theta_j \\ 0 & \text{en otro caso} \end{cases} \quad (5)$$

La polaridad p_j (positiva o negativa) determina que no cambie el sentido de la desigualdad ante la posibilidad de cambio de signo de f_j según la ecuación (4). Por lo tanto el clasificador nos dará un resultado de 1 ó 0 dependiendo de si la característica está contenida o no en la imagen. El valor del umbral está especificado en el entrenamiento. Es decir, las características serán entrenadas previamente para determinar los valores de umbral. Por ejemplo, para la detección facial, antes de aplicar estos clasificadores, existirá una fase de entrenamiento donde, a partir de imágenes positivas (imágenes que contienen caras con distintos valores de brillo, contraste, etc.) y negativas (imágenes que no contienen caras) se obtienen los valores del umbral para cada característica.

Existen en la literatura diferentes implementaciones de este algoritmo de aprendizaje, una de ellas es la que se utiliza en la fase de pruebas del presente proyecto, correspondiente a los métodos desarrollados bajo las librerías OpenCV [37]. El algoritmo ha sido entrenado con una amplia diversidad de ejemplos de imágenes de caras bajo distintas condiciones. Esto garantiza el hecho de que los resultados de detección de las nuevas imágenes reales utilizadas en el probador virtual puedan ser contrastados con las de esta base de datos, evitando así la inclusión de ejemplos de imágenes similares tanto en la fase de entrenamiento como de clasificación y consiguiendo verificar la robustez de la estrategia propuesta. Por el contrario, debido a la opacidad en la documentación de los métodos utilizados por las propias OpenCV no es posible determinar los umbrales θ_j utilizados durante el entrenamiento.

3.3.2.3.2. CLASIFICACIÓN EN CASCADA

La forma general del proceso de detección es la de un árbol de decisión degenerado, lo que Viola y Jones [1] llaman una “cascada” (Figura 12).

Este proceso consiste en aplicar en varias etapas la clasificación de características a la sub-ventana. En la primera etapa se utiliza un conjunto reducido de características, por ejemplo las características A y B de

la Figura 9. Si la primera etapa es superada se aplica una segunda etapa añadiendo nuevas características, por ejemplo la característica C en la Figura 9. Si supera esta segunda etapa se aplican las siguientes etapas de la misma manera. Si se han superado todas las etapas se puede concluir que la sub-ventana contiene el objeto que se está detectando. De esta manera, el proceso cada vez es más complicado y está llamado a aplicar mayor complejidad sólo en las sub-ventanas que sean candidatas a tener el objeto que se está detectando. Por lo tanto, las regiones que no contienen el objeto a detectar se descartan en la primera fase, lo que hace que se aumente la velocidad de procesamiento.

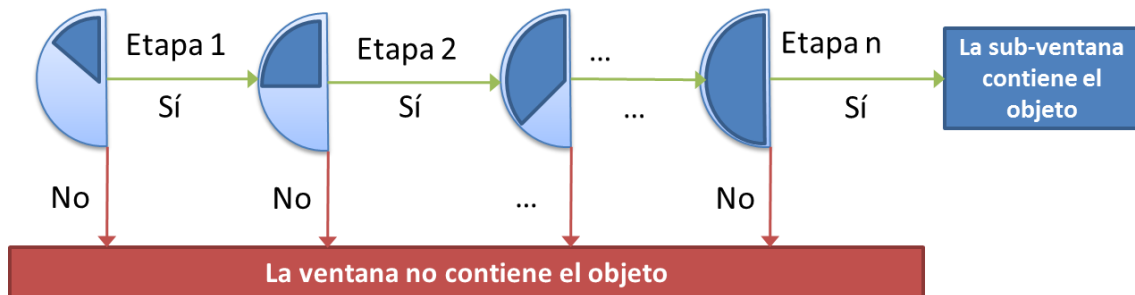


Figura 12. Representación esquemática de clasificación en cascada.

En cada etapa se aplican las siguientes operaciones:

1. Se evalúan las funciones de las características de la etapa correspondiente para la sub-ventana (x, y) . Es decir, se aplica la función $f(x, y) = \sum_i p_n(i) - \sum_i p_b(i)$ para cada característica.
2. Se determina el resultado del clasificador “débil” para cada característica, mediante la función:

$$h_{débil}(x) = \begin{cases} 1 & \text{si } p_j f_j(x, y) < p_j \theta_j \\ 0 & \text{en otro caso} \end{cases}$$

El umbral viene determinado por el valor que se ha obtenido en el entrenamiento de dicha característica.

3. Se combinan los clasificadores “débiles” para obtener un clasificador “fuerte” para dicha etapa. La función utilizada para calcular el clasificador débil es la siguiente:

$$h_{fuerte}(x, y) = \begin{cases} 1 & \text{si } \alpha_1 h_1(x, y) + \dots + \alpha_n h_n(x, y) \geq \frac{1}{2}(\alpha_1 + \dots + \alpha_n) \\ 0 & \text{en otro caso} \end{cases}$$

Donde α es un peso que se le da a cada característica (especificado en la fase de entrenamiento) y n es el número de características de la etapa.

Si el resultado del clasificador fuerte es 1, se aplican las mismas operaciones para la siguiente etapa. En caso de ser 0, la sub-ventana se moverá a la siguiente posición.

Como se ha indicado previamente se ha utilizado la base de ejemplos entrenados de OpenCV [37], de suerte que los pesos finalmente allí obtenidos son los que se utilizan en la clasificación. De nuevo, dada la falta de documentación al respecto no se identifican convenientemente sus valores en los ficheros XML proporcionados por los desarrolladores al efecto. Sin embargo, sí se documenta el uso de 212 características, ecuación (4) y 21 etapas.

Las siguientes imágenes (Figura 13 y Figura 14) muestran un ejemplo simplificado aplicando la cascada de clasificadores a una sub-ventana que no contiene el elemento a detectar (una cara) (a) y una sub-ventana que sí lo contiene (b).

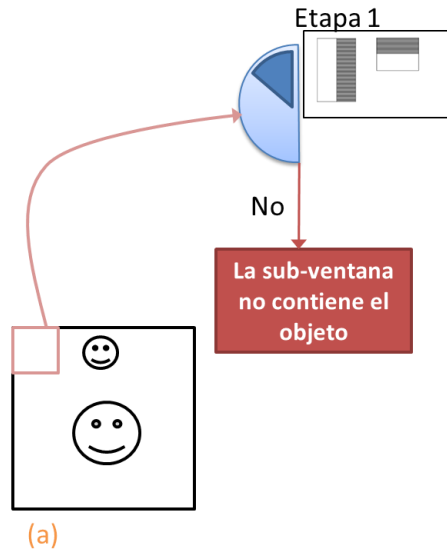


Figura 13. Ejemplo simplificado de aplicación de cascada de clasificadores a una sub-ventana que no contiene una cara

En la Figura 13 se muestra cómo al calcular las operaciones correspondientes al clasificador “débil” de las características de la primera etapa, aplicando el correspondiente umbral para esas características en la fase de entrenamiento, y calcular más tarde el clasificador “fuerte”, se puede concluir que la sub-ventana no contiene una cara. Por lo tanto, se procede a mover la ventana a la siguiente posición.

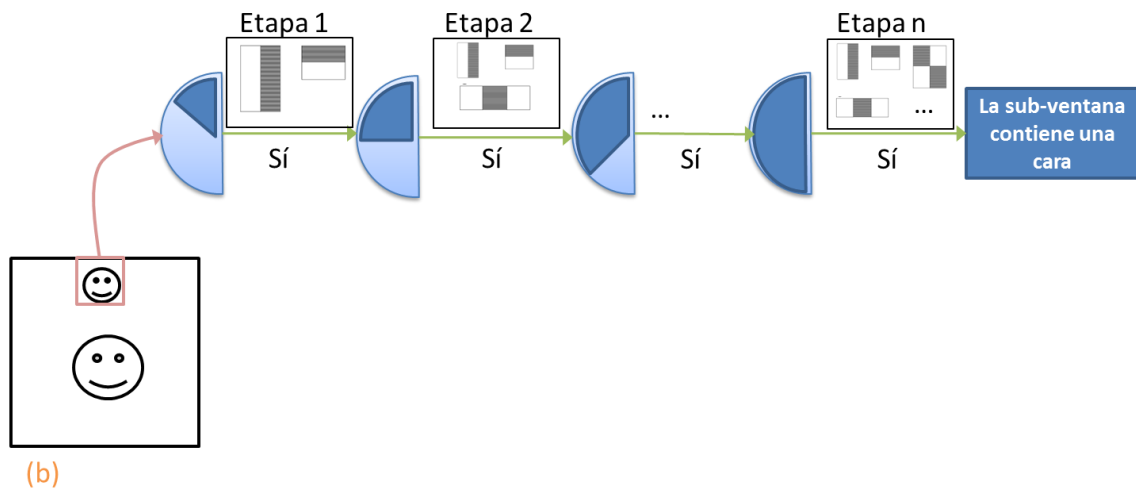


Figura 14. Ejemplo simplificado de aplicación de cascada de clasificadores a una sub-ventana que sí contiene una cara

En la Figura 14 se muestra cómo aplicar las operaciones correspondientes al clasificador “débil” de las características de la primera etapa, aplicando el umbral especificado para esas características en la fase de entrenamiento, y calcular más tarde el clasificador “fuerte”, se puede concluir que la sub-ventana es candidata a tener una cara. Por consiguiente, se procede a hacer la misma operación en la segunda etapa sobre un conjunto más amplio de características, que en realidad es añadir una sub-ventana más. El proceso se acaba cuando al aplicar todo el conjunto de características de la etapa enésima se concluye que la sub-ventana contiene una cara, por haber superado todas las etapas. En nuestro caso, al utilizar sólo cuatro características, el proceso consta de tres etapas, ya que se utilizan dos características en la primera etapa y luego se añade una característica más por etapa.

3.3. FUSIÓN DE IMÁGENES

Una vez detectada la posición de la cara y los ojos en la imagen recibida por el servidor, se fusiona la imagen de las gafas (Figura 15) en la posición de los ojos detectados con la imagen original y con un tamaño proporcional a la cara.



Figura 15. Imagen de ejemplo de gafas

3.3.1. RE-ESCALADO DE LA IMAGEN DE LAS GAFAS

La primera operación antes de superponer las gafas sobre la imagen original es re-escalarla para que tenga un tamaño adecuado al tamaño de la cara.

Para ello, se calcula la proporción entre el ancho de la imagen de las gafas y el ancho de la cara. Este factor de escala es el que finalmente se utiliza para re-escalar la imagen de las gafas.

El factor de escala se obtiene mediante la siguiente expresión,

$$\text{Factor de escala} = \frac{\text{ancho de la cara}}{\text{ancho imagen de la cara}}$$

Después se aplica el re-escalado a la imagen de las gafas con los siguientes parámetros:

$$\text{Ancho imagen gafas} = \text{Ancho de la cara}$$

$$\text{Alto imagen gafas} = \text{Alto imagen gafas} \times \text{Factor de escala}$$

$$\text{Tipo de interpolación} = \text{remuestreo usando relación área de píxeles}$$

Se aplica este tipo de interpolación ya que es un método rápido que obtiene el promedio de varios píxeles sobre el área considerada.

3.3.2. SUPERPOSICIÓN DE LA IMAGEN DE LAS GAFAS

La superposición de las gafas sobre la imagen original se realiza siempre en función de la posición de los ojos, tal y como se describe más adelante. En este sentido conviene reseñar el hecho que la estrategia propuesta tiene la posibilidad de detectar bien las caras en su conjunto o los ojos de forma separada utilizando el método descrito anteriormente a través de las características de Haar. En el primer caso, es necesario todavía un proceso adicional de forma que se detecten los ojos dentro de la cara. Esto se puede realizar bien estimando la posición de los mismos sobre la cara detectada, o bien volviendo a aplicar de nuevo el proceso de detección, en este caso sólo sobre la región de la cara, que es el finalmente elegido por cumplir con los requisitos de tiempo real a la vez que resulta más fiable, debido a la variabilidad de caras.

En cualquier caso, una vez detectados los ojos, la superposición de las gafas sobre la imagen original se realiza de la siguiente manera:

1. La posición a partir de la cual se superpondrán las gafas siempre será la que marque el detector considerando como origen la esquina superior izquierda del rectángulo que identifica la región de los ojos.

Posición de los ojos obtenida
en la fase de detección

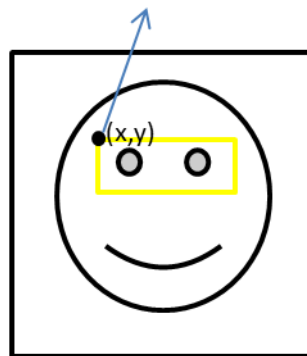


Figura 16. Representación esquemática de la posición aproximada de los ojos

- Una vez que se posiciona en el sitio correcto, se recorre la imagen de las gafas por filas y columnas y se superpone el píxel correspondiente de las imágenes de las gafas, teniendo en cuenta la opacidad de dicho píxel, en la imagen que contiene la cara, que también se recorre por filas y columnas, a partir de la posición indicada en el paso 1. Para fusionar los píxeles de ambas imágenes se aplica la siguiente operación:

$$Píxel_f(x, y) = Píxel_{ic}(x, y) \times (1 - opacidad(x, y)) + Píxel_g(x, y) \times opacidad(x, y)$$

Donde $Píxel_{ic}(x, y)$ es el píxel de la imagen de la cara en la posición (x, y) , $Píxel_g(x, y)$ es el píxel de la imagen de las gafas en la posición (x, y) , $opacidad$ es la opacidad que marca el canal Alfa de la imagen de las gafas en la posición (x, y) y $Píxel_f(x, y)$ es el píxel fusionado en la posición (x, y) .

4. RESULTADOS OBTENIDOS

4.1. DATOS UTILIZADOS

4.1.1. DESCRIPCIÓN DE LAS IMÁGENES

Las imágenes “testUnaPersona01.png” a “testUnaPersona05.png”, “testNoHumano.png” y “testDosPersonas.png” han sido captadas con una cámara web Logitech c310 HD webcam. Las imágenes “testUnaPersona06.png” y “testUnaPersona07.png” han sido obtenidas con la cámara web integrada en un portátil ACER Aspire 5810T. La imagen “testUnaPersona08.png” ha sido obtenida con la cámara delantera del teléfono móvil LG L70.

La primera imagen (“testUnaPersona01.png”) corresponde a una imagen en condiciones ideales, tanto de luminosidad como de posición de la cara.

Las siguientes cuatro imágenes (“testUnaPersona02.png” a “testUnaPersona05.png”) corresponden a imágenes tomadas con distintos ángulos y posición de la cara, para comprobar hasta qué punto de rotación de la cara es capaz de detectar el algoritmo.

Las imágenes “testUnaPersona06.png” y “testUnaPersona07.png” están tomadas con distintas condiciones de luminosidad, tanto artificial como natural., y con un entorno más complejo que las anteriores

La siguiente imagen (“testUnaPersona08.png”) corresponde a una imagen tomada con la cámara frontal de un teléfono móvil para comprobar la iluminación y resolución de la misma.

Las imágenes “testNoHumano01.png” y “testNoHumano02.png” corresponden a imágenes tomadas con una cámara web, también con diferentes condiciones de luminosidad, y que contienen elementos que no son caras humanas pero podrían inducir a un falso positivo.

La siguiente imagen (“testDosPersonas.png”) es una imagen con la presencia de dos personas en la misma.

Con la imagen “testUnaPersona-NoHumano.png” se pretende comprobar si la presencia de un humano con un objeto que no es humano influye en la detección del humano.



Figura 17. testUnaPersona01.png



Figura 18. testUnaPersona02.png



Figura 19. testUnaPersona03.png

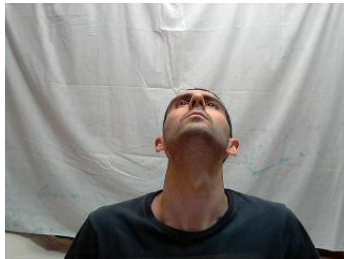


Figura 20. testUnaPersona04.png

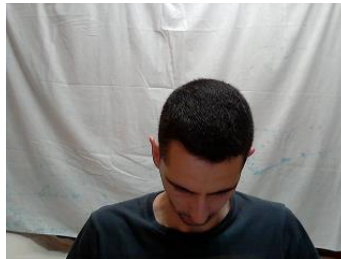


Figura 21. testUnaPersona05.png



Figura 22. testUnaPersona06.png



Figura 23. testUnaPersona07.png

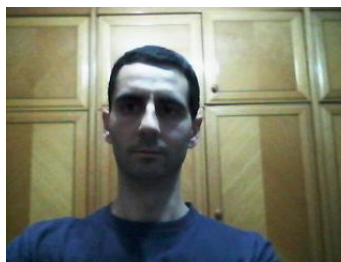


Figura 24. testUnaPersona08.png



Figura 25. testNoHumano01.png



Figura 26. testNoHumano02.png



Figura 27. testDosPersona.png



Figura 28. testUnaPersona-NoHumano.png

4.1.2. FICHEROS CLASIFICADORES DE HAAR

Para detectar las caras y los ojos se utilizan los ficheros XML contenidos en la base de datos de las OpenCV [37], que abarcan la información de las características a utilizar en las fases de la clasificación en cascada, así como los valores de umbrales para cada característica.

- **frontalEyes35x16.xml**: proporciona detección de los dos ojos. Fichero creado por Yusuf Bediz [28], entrenado con una base de imágenes de ambos ojos de forma frontal.

- **haarcascade_frontalface_default.xml**: proporciona detección de la cara de forma frontal.

4.1.3. PARÁMETROS

- Factor de escala: con posibles valores de 1.1 a 1.4
- Número mínimo de vecinos: con posibles valores de 1 a 4. Aunque se puede aplicar al número de vecinos el valor 0, se descarta este valor para nuestras pruebas ya que como se comenta en el apartado 4.3.1.2. Número mínimo de vecinos (*Min neighbors*), seleccionando dicho valor no se agrupan los resultados adecuadamente, pudiendo llegar a producir una importante confusión en relación al número de objetos detectados.

4.2. DETALLES DEL SERVIDOR Y MÁQUINAS UTILIZADAS

Los equipos que se han utilizado para realizar las pruebas son los siguientes:

- Equipo cliente:
 - Navegador Web: Google Chrome
 - Sistema Operativo: Windows 7
 - CPU: Intel iCore i5
 - RAM: 4 GB
 - Red: Banda Ancha de 50 Mbit/s
- Equipo servidor:
 - Servidor Web: Tomcat 7
 - Sistema Operativo: Ubuntu 13
 - CPU: 3 vCores
 - RAM: 4GB

4.3. MÉTODOS UTILIZADOS

El algoritmo de Viola y Jones [1] que se aplica es el contenido en la librería En JavaCV [38] mediante el método `cvHaarDetectObjects`. Este método permite detectar los objetos en una imagen en escala de grises según el fichero de clasificador XML de Haar seleccionado y los parámetros correspondientes que se pueden definir para modificar el comportamiento del algoritmo.

4.3.1. PARÁMETROS AJUSTABLES DEL ALGORITMO DE VIOLA Y JONES

4.3.1.1. FACTOR DE ESCALA (*SCALE FACTOR*)

Es un parámetro que especifica cuánto debe aumentar la escala para las detecciones de objetos con cada paso que se realiza sobre la imagen. El valor por defecto de este parámetro es 1.1, que quiere decir que la escala se incrementa en un 10% en cada paso.

Los posibles valores que debe tener este parámetro son 1.1, 1.2, 1.3 o 1.4.

Cuanto más bajo sea el valor, más exploraciones se requerirán para comprobar la imagen con el fin de detectar el objeto de interés, pero naturalmente tomará más tiempo.

4.3.1.2. NÚMERO MÍNIMO DE VECINOS (*MIN NEIGHBORS*)

Indica el mínimo número (menos 1) de rectángulos vecinos que componen un objeto. Todos los grupos de un número menor de rectángulos que el valor de este parámetro, menos 1, se rechazan. Si el valor es 0, la función no agrupa nada y retorna todos los rectángulos candidatos detectados, lo cual puede ser útil si se quiere aplicar un procedimiento personalizado de agrupación.

Es un parámetro de control para prevenir falsas detecciones. Las localizaciones reales de objetos en una imagen tienden a ser múltiples “éxitos” en la misma área porque los píxeles de los alrededores y las escalas a menudo indican que se ha detectado el objeto. Estableciendo al valor por defecto (3) en el código de detección de caras indica que sólo decidirá si una cara está presente en una localización si al menos se sobreponen 3 detecciones.

Los valores de este parámetro deben estar entre 0 y 4.

4.3.1.3. *FLAG*

El parámetro *flag* es usado para establecer preferencias específicas.

Los cuatro posibles valores que puede tener son:

- **CV_HAAR_DO_CANNY_PRUNNING**. Estableciendo este valor hace que regiones con niveles de intensidad similares, esto es regiones planas en intensidad y sin líneas con bordes significativos, sean omitidas por el clasificador.
- **CV_HAAR_SCALE_IMAGE**. Proporciona el valor para el escalado de la imagen en lugar del detector (esto puede dar algunas ventajas de rendimiento en términos de cómo se utilizan la memoria y la caché).
- **CV_HAAR_FIND_BIGGEST_OBJECT**. Informa que se debe retornar sólo el objeto más grande encontrado (de ahí que el número de objetos devueltos sea uno o ninguno).
- **CV_HAAR_DO_ROUGH_SEARCH**. Este valor se utiliza para terminar la búsqueda a cualquier escala en la que sea encontrado el primer candidato (con suficientes vecinos para ser considerado un “éxito”).

4.3.1.4. TAMAÑO MÍNIMO (*MIN SIZE*)

Establece la región más pequeña en la cual se procederá a buscar el objeto. Estableciendo a un valor alto reducirá el tiempo de computación a costa de perder caras pequeñas.

Los valores por defecto vienen determinados en los ficheros XML de cada clasificador de Haar, especificados bajo la etiqueta `<size>`.

4.3.1.5. TAMAÑO MÁXIMO (*MAX SIZE*)

Tamaño máximo de los posibles objetos. Los objetos mayores que este valor serán ignorados.

4.3.2. MÉTODOS IMPLEMENTADOS

El método `cvHaarDetectObjects` acepta dos versiones del mismo, una que recibe todos los argumentos posibles (factor de escala, número mínimo de vecinos, *flag*, tamaño mínimo y tamaño máximo) y otra que sólo recibe tres de los argumentos (factor de escala, número mínimo de vecinos y

flag). Para simplificar el número de variantes de cara a las pruebas realizadas, se utilizará la versión que recibe tres argumentos. En este caso el tamaño mínimo viene determinado por el valor por defecto explicado en la sección anterior y el tamaño máximo será el tamaño total de la imagen recibida. También se ha puesto por defecto el parámetro *flag* al valor **CV_HAAR_DO_CANNY_PRUNNING** para reducir el número de combinaciones en las pruebas.

A continuación se muestran los pseudocódigos de los distintos métodos creados utilizando las llamadas a `cvHaarDetectObjects` para la detección de caras y sus características.

4.3.2.1. MÉTODO 1

Para este método se utiliza el fichero clasificador de Haar **frontalEyes35x16.xml** para detectar la posición de los ojos.

1. Decodificar la imagen a un objeto que puede manejar OpenCV.
2. Transformar el objeto del paso 1 a escala de grises.
3. Cargar el fichero clasificador de Haar recibido.
4. Ejecutar el método `cvHaarDetectObjects` con los parámetros recibidos (factor de escala, número mínimo de vecinos) sobre la imagen en escala de grises.
5. Recorrer el número total de objetos encontrados en el paso 5 y dibujar un rectángulo de color amarillo sobre cada objeto encontrado.
6. Calcular el factor de re-escalado de la imagen de las gafas
 - a. Factor de re-escalado = Ancho del objeto detectado / Ancho original de la imagen de las gafas.
7. Re-escalado de la imagen de las gafas al siguiente tamaño:
 - a. Ancho de las gafas = ancho del objeto detectado.
 - b. Alto de las gafas = Alto original de imagen de las gafas * factor de re-escalado.
8. Superposición de la imagen de las gafas en la posición (x,y) desde el origen del objeto detectado (este origen es la posición x más hacia la izquierda y la posición más hacia arriba).
9. Devolver la imagen modificada y el número total de objetos encontrados.

4.3.2.3. MÉTODO 2

Para este método se utiliza el fichero clasificador de Haar **haarcascade_frontalface_default.xml** como primer clasificador para detectar la posición de la cara y **frontalEyes35x16.xml** como segundo clasificador para detectar la posición de los ojos.

1. Decodificar la imagen a un objeto que puede manejar OpenCV.
2. Transformar el objeto del paso 1 a escala de grises.
3. Cargar el fichero clasificador de Haar recibido.
4. Ejecutar el método `cvHaarDetectObjects` con los parámetros recibidos (factor de escala, número mínimo de vecinos, *flag*) sobre la imagen en escala de grises.
5. Recorrer el número total de objetos encontrados en el paso 5
 - a. Pintar un rectángulo de color amarillo sobre el objeto encontrado.
 - b. Establecer la región de interés (ROI) en la imagen en escala de grises cubriendo el objeto detectado.

- c. Ejecutar el método `cvHaarDetectObjects` con los parámetros recibidos (factor de escala, número mínimo de vecinos, *flag*) sobre la región de interés.
 - d. Dibujar un rectángulo de color rojo sobre los objetos encontrados en la región de interés.
 - e. Calcular el factor de re-escalado de la imagen de las gafas
 - i. Factor de re-escalado = Ancho de la cara detectada / Ancho original de la imagen de las gafas.
 - f. Re-escalado de la imagen de las gafas al siguiente tamaño:
 - i. Ancho de las gafas = Ancho de la cara.
 - ii. Alto de las gafas = Alto original de imagen de las gafas * factor de re-escalado.
 - g. Superposición de la imagen de las gafas en la posición (x,y) desde la siguiente posición:
 - i. Posición x = Posición más hacia la derecha de la cara detectada
 - ii. Posición y = Posición más hacia arriba de los ojos detectados.
6. Devolver la imagen modificada y el número total de objetos encontrados, que será la suma de los objetos encontrados de la primera ejecución de `cvHaarDetectObjects` y de la segunda ejecución.

4.3.2.5. MÉTODO 3

Para este método se utiliza el fichero clasificador de Haar `haarcascade_frontalface_default.xml` para detectar la posición de la cara.

1. Decodificar la imagen a un objeto que puede manejar OpenCV.
2. Transformar el objeto del paso 1 a escala de grises.
3. Cargar el fichero clasificador de Haar recibido.
4. Ejecutar el método `cvHaarDetectObjects` con los parámetros recibidos (factor de escala, número mínimo de vecinos, *flag*) sobre la imagen en escala de grises.
5. Recorrer el número total de objetos encontrados en el paso 5.
 - a. Dibujar un rectángulo de color amarillo sobre cada objeto encontrado.
 - b. Calcular la posición estimada de los ojos mediante medidas antropométricas de la cara, mediante la fórmula descrita en [4.3.2.7. Fórmula para el cálculo de la posición estimada de los ojos].
 - c. Dibujar un recuadro de color rojo para la posición estimada del ojo derecho y otro para la posición estimada del ojo izquierdo.
 - d. Calcular el factor de re-escalado de la imagen de las gafas
 - i. Factor de re-escalado = Ancho de la cara detectada / Ancho original de la imagen de las gafas.
 - e. Re-escalado de la imagen de las gafas al siguiente tamaño:
 - i. Ancho de las gafas = Ancho de la cara.
 - ii. Alto de las gafas = Alto original de imagen de las gafas * factor de re-escalado.
 - f. Superposición de la imagen de las gafas en la posición (x,y) desde la siguiente posición:

- i. Posición x = Posición más hacia la derecha de la cara detectada
 - ii. Posición y = Posición y_1 del ojos derecho estimado.
6. Devolver la imagen modificada y el número total de objetos encontrados.

4.3.2.7. FÓRMULA PARA EL CÁLCULO DE LA POSICIÓN ESTIMADA DE LOS OJOS

Para determinar la posición de los ojos en la cara, en el método 3, se utilizan las medidas de alto y ancho de una cara.

$$h = [\text{alto de la cara}] * 0.5$$

$$c = [\text{ancho de la cara}] * 0.5$$

$$x_1 = c - (0.6 * [\text{ancho de la cara}] * 0.5)$$

$$y_1 = h - (0.4 * [\text{alto de la cara}] * 0.5)$$

$$x_2 = c - (0.2 * [\text{ancho de la cara}] * 0.5)$$

$$y_2 = h$$

$$x_3 = c + (0.2 * [\text{ancho de la cara}] * 0.5)$$

$$y_3 = y_1$$

$$x_4 = c + (0.6 * [\text{ancho de la cara}] * 0.5)$$

$$y_4 = y_1$$

En la siguiente imagen se muestra la correspondencia de cada coordenada en una imagen de una cara frontal.

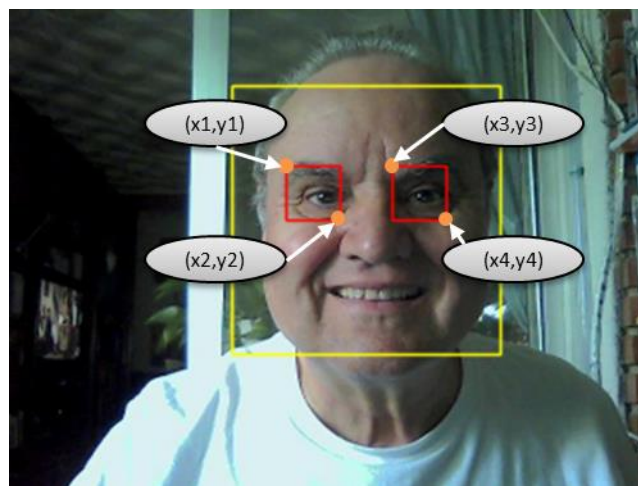
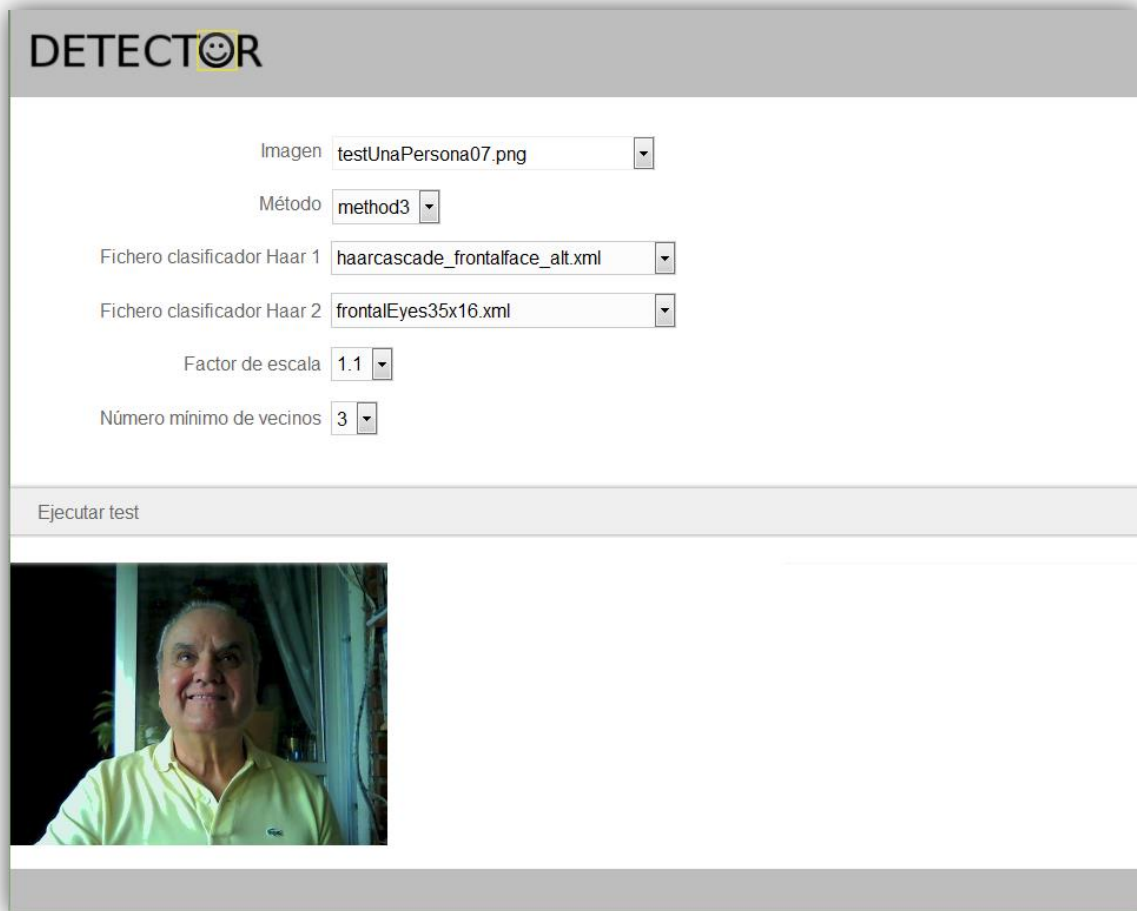


Figura 29. Posición estimada de los ojos basada en las medidas del ancho y alto de la cara.

4.4. RESULTADOS

La ejecución de las pruebas se hace a través de una herramienta creada para tal fin. Esta herramienta nos permite seleccionar una imagen y ejecutar cada método, con los parámetros seleccionados (Figura 30), mostrando los resultados de dicha ejecución en un fichero CSV.



The image shows a web application interface titled "DETECTOR" with a smiley face icon. The interface is designed for configuring a face detection test. It features several dropdown menus and input fields for parameter selection:

- Imagen:** A dropdown menu with the selected value "testUnaPersona07.png".
- Método:** A dropdown menu with the selected value "method3".
- Fichero clasificador Haar 1:** A dropdown menu with the selected value "haarcascade_frontalface_alt.xml".
- Fichero clasificador Haar 2:** A dropdown menu with the selected value "frontalEyes35x16.xml".
- Factor de escala:** A dropdown menu with the selected value "1.1".
- Número mínimo de vecinos:** A dropdown menu with the selected value "3".

Below the configuration options, there is a button labeled "Ejecutar test". Underneath the button, a small rectangular image of a person's face is displayed, representing the input image for the test.

Figura 30. Selección de parámetros en la herramienta de test.

Este fichero CSV creado contiene la siguiente información en columnas:

- Nombre del fichero seleccionado.
- Factor de escala.
- Número mínimo de vecinos
- Tiempo de envío de la imagen al servidor
- Tiempo medio de procesamiento
- Tiempo de recepción de la imagen en el dispositivo.
- Número de objetos detectados.

Aparte del fichero CSV generado, una vez que se pulsa sobre "Ejecutar test", se muestra por pantalla el resultado de la detección y superposición de las gafas en la imagen (Figura 31).

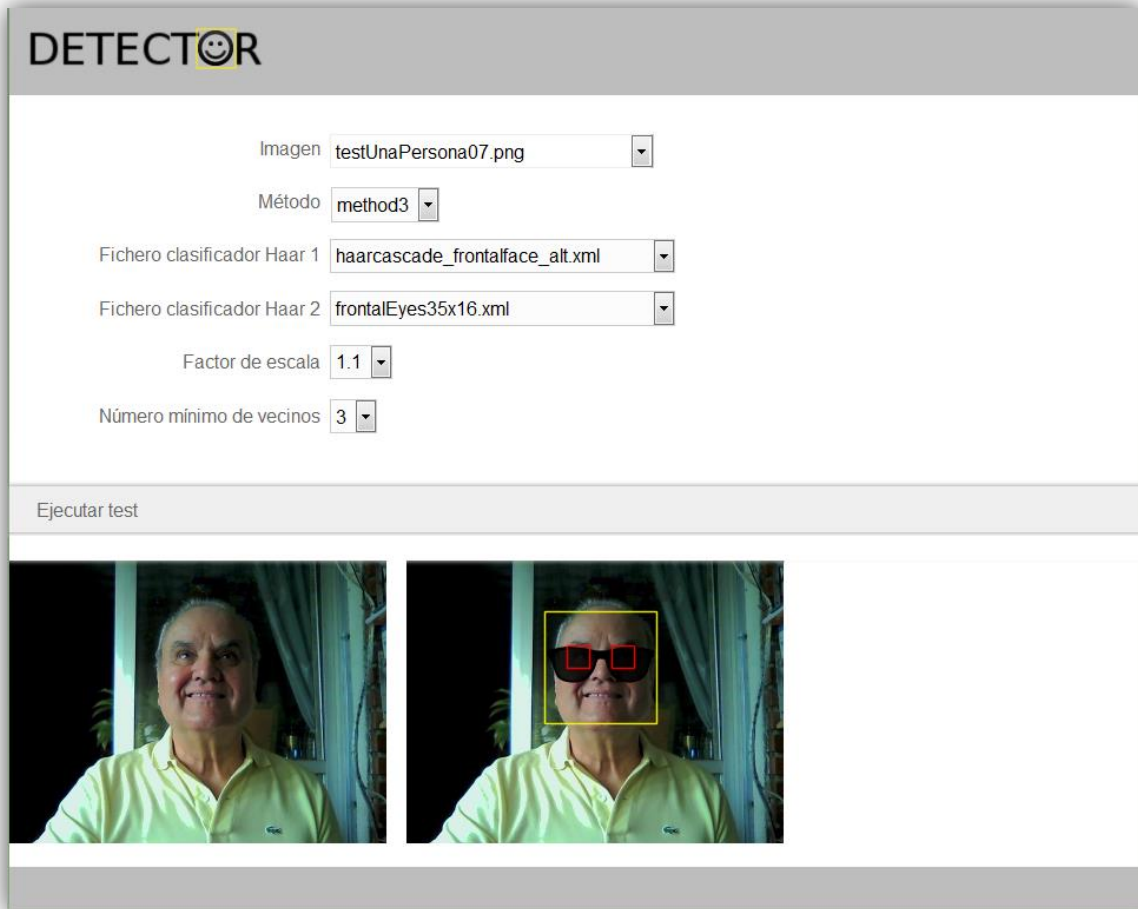


Figura 31. Resultado de un test en la herramienta de test.

4.4.1. PRUEBAS PARA EL MÉTODO 1

A continuación se muestra una tabla resumen con los resultados obtenidos del fichero CSV para la ejecución del método 1 sobre todas las imágenes y con todos los parámetros posibles.

La tabla contiene los siguientes elementos:

- **Imagen:** Imagen sobre la que se aplica el método.
- **F.E.:** Factor de Escala.
- **N.M.V.:** Número Mínimo de Vecinos
- **T.E.:** Tiempo de Envío en segundos (desde que el cliente envía la imagen hasta que el servidor la recibe).
- **T.P.:** Tiempo de Procesamiento en segundos (media de tiempos del proceso de detección y superposición).
- **T.R.:** Tiempo de Recepción en segundos (desde que el servidor envía la imagen procesada hasta que la recibe el cliente).
- **O.D.:** Objetos Detectados en la fase de procesamiento, relativos a caras.
- **O.R.:** Objetos Reales (caras) que posee la imagen.
- **R.:** Resultado. Se realiza una resta entre la columna O.D. y O.R. Si el resultado es 0 se han detectado los objetos esperados. Si es distinto de 0 el método ha detectado más objetos o menos de los esperados.

Imagen	F.E.	N.M.V.	T.E.	T. P.	T.R	O. D.	O. R.	R.
testUnaPersona01	1.1	1	1,432	0,073	1,186	1	1	0
testUnaPersona01	1.1	2	1,435	0,072	1,182	1	1	0
testUnaPersona01	1.1	3	1,431	0,08	1,19	1	1	0
testUnaPersona01	1.1	4	1,447	0,157	1,09	1	1	0
testUnaPersona01	1.2	1	1,431	0,085	0,822	1	1	0
testUnaPersona01	1.2	2	1,432	0,066	1,177	1	1	0
testUnaPersona01	1.2	3	1,445	0,155	1,098	1	1	0
testUnaPersona01	1.2	4	1,437	0,15	1,099	1	1	0
testUnaPersona01	1.3	1	1,436	0,202	1,162	2	1	-1
testUnaPersona01	1.3	2	1,494	0,146	1,154	1	1	0
testUnaPersona01	1.3	3	1,432	0,063	1,18	1	1	0
testUnaPersona01	1.3	4	1,432	0,196	1,045	1	1	0
testUnaPersona01	1.4	1	1,459	0,164	1,175	2	1	-1
testUnaPersona01	1.4	2	1,433	0,087	1,175	2	1	-1
testUnaPersona01	1.4	3	1,434	0,064	1,186	1	1	0
testUnaPersona01	1.4	4	1,432	0,064	1,194	1	1	0
testUnaPersona02	1.1	1	1,465	0,059	1,187	0	0	0
testUnaPersona02	1.1	2	1,478	0,163	0,999	0	0	0
testUnaPersona02	1.1	3	1,465	0,11	1,207	0	0	0
testUnaPersona02	1.1	4	1,464	0,057	1,147	0	0	0
testUnaPersona02	1.2	1	1,462	0,055	1,215	0	0	0
testUnaPersona02	1.2	2	1,461	0,05	1,216	0	0	0
testUnaPersona02	1.2	3	1,465	0,051	1,166	0	0	0
testUnaPersona02	1.2	4	1,464	0,056	1,062	0	0	0
testUnaPersona02	1.3	1	1,465	0,046	1,213	0	0	0
testUnaPersona02	1.3	2	1,464	0,048	1,22	0	0	0
testUnaPersona02	1.3	3	1,461	0,048	1,222	0	0	0
testUnaPersona02	1.3	4	1,465	0,05	1,215	0	0	0
testUnaPersona02	1.4	1	1,466	0,049	1,196	0	0	0
testUnaPersona02	1.4	2	1,466	0,046	1,204	0	0	0
testUnaPersona02	1.4	3	1,463	0,042	1,211	0	0	0
testUnaPersona02	1.4	4	1,464	0,049	1,215	0	0	0
testUnaPersona04	1.1	1	1,569	0,139	1,201	0	0	0
testUnaPersona04	1.1	2	1,464	0,062	1,183	0	0	0
testUnaPersona04	1.1	3	1,464	0,051	1,214	0	0	0
testUnaPersona04	1.1	4	1,463	0,062	1,217	0	0	0
testUnaPersona04	1.2	1	1,469	0,055	1,222	0	0	0
testUnaPersona04	1.2	2	1,467	0,151	1,063	0	0	0
testUnaPersona04	1.2	3	1,482	0,081	1,231	0	0	0
testUnaPersona04	1.2	4	1,463	0,048	1,225	0	0	0
testUnaPersona04	1.3	1	1,464	0,048	1,215	0	0	0
testUnaPersona04	1.3	2	1,464	0,052	1,187	0	0	0
testUnaPersona04	1.3	3	1,465	0,051	1,223	0	0	0
testUnaPersona04	1.3	4	1,463	0,047	1,216	0	0	0
testUnaPersona04	1.4	1	1,465	0,047	1,202	0	0	0
testUnaPersona04	1.4	2	1,462	0,05	1,22	0	0	0
testUnaPersona04	1.4	3	1,465	0,05	1,227	0	0	0
testUnaPersona04	1.4	4	1,462	0,044	1,223	0	0	0
testUnaPersona08	1.1	1	1,468	0,063	1,179	0	0	0
testUnaPersona08	1.1	2	1,463	0,055	1,234	0	0	0
testUnaPersona08	1.1	3	1,464	0,069	1,177	0	0	0

testUnaPersona08	1.1	4	1,463	0,06	1,213	0	0	0
testUnaPersona08	1.2	1	1,466	0,09	1,132	0	0	0
testUnaPersona08	1.2	2	1,479	0,107	0,968	0	0	0
testUnaPersona08	1.2	3	1,464	0,127	1,223	0	0	0
testUnaPersona08	1.2	4	1,464	0,049	1,199	0	0	0
testUnaPersona08	1.3	1	1,468	0,046	1,212	0	0	0
testUnaPersona08	1.3	2	1,463	0,045	1,235	0	0	0
testUnaPersona08	1.3	3	1,466	0,047	1,244	0	0	0
testUnaPersona08	1.3	4	1,468	0,046	1,208	0	0	0
testUnaPersona08	1.4	1	1,468	0,044	1,225	0	0	0
testUnaPersona08	1.4	2	1,466	0,046	1,179	0	0	0
testUnaPersona08	1.4	3	1,463	0,045	1,222	0	0	0
testUnaPersona08	1.4	4	1,462	0,045	1,219	0	0	0
testUnaPersona09	1.1	1	1,465	0,063	1,206	1	0	-1
testUnaPersona09	1.1	2	1,466	0,061	1,216	1	0	-1
testUnaPersona09	1.1	3	1,466	0,061	1,196	1	0	-1
testUnaPersona09	1.1	4	1,463	0,059	1,184	1	0	-1
testUnaPersona09	1.2	1	1,466	0,061	1,143	1	0	-1
testUnaPersona09	1.2	2	1,464	0,058	1,23	1	0	-1
testUnaPersona09	1.2	3	1,463	0,055	1,192	1	0	-1
testUnaPersona09	1.2	4	1,464	0,109	1,057	0	0	0
testUnaPersona09	1.3	1	1,476	0,168	1,148	1	0	-1
testUnaPersona09	1.3	2	1,466	0,051	1,215	0	0	0
testUnaPersona09	1.3	3	1,476	0,241	1,217	0	0	0
testUnaPersona09	1.3	4	1,467	0,052	1,227	0	0	0
testUnaPersona09	1.4	1	1,463	0,048	1,225	0	0	0
testUnaPersona09	1.4	2	1,466	0,047	1,22	0	0	0
testUnaPersona09	1.4	3	1,467	0,046	1,223	0	0	0
testUnaPersona09	1.4	4	1,464	0,047	1,219	0	0	0
testUnaPersona10	1.1	1	1,604	0,175	1,126	2	1	-1
testUnaPersona10	1.2	1	1,475	0,089	1,154	1	1	0
testUnaPersona10	1.1	2	1,474	0,108	1,234	2	1	-1
testUnaPersona10	1.1	3	1,479	0,11	1,214	2	1	-1
testUnaPersona10	1.1	4	1,469	0,089	1,176	1	1	0
testUnaPersona10	1.2	1	1,475	0,083	1,111	1	1	0
testUnaPersona10	1.2	2	1,471	0,088	1,214	1	1	0
testUnaPersona10	1.2	3	1,478	0,084	0,988	1	1	0
testUnaPersona10	1.2	4	1,47	0,084	1,096	1	1	0
testUnaPersona10	1.3	1	1,472	0,067	0,966	1	1	0
testUnaPersona10	1.3	2	1,472	0,068	1,181	1	1	0
testUnaPersona10	1.3	3	1,469	0,07	1,203	1	1	0
testUnaPersona10	1.3	4	1,472	0,072	1,199	1	1	0
testUnaPersona10	1.4	1	1,471	0,088	1,186	1	1	0
testUnaPersona10	1.4	2	1,469	0,074	1,195	1	1	0
testUnaPersona10	1.4	3	1,478	0,077	1,204	1	1	0
testUnaPersona10	1.4	4	1,469	0,05	1,211	0	1	1
testUnaPersona13	1.1	1	1,474	0,095	1,216	1	1	0
testUnaPersona13	1.1	2	1,472	0,086	1,206	1	1	0
testUnaPersona13	1.1	3	1,47	0,086	1,129	1	1	0
testUnaPersona13	1.1	4	1,471	0,082	1,167	1	1	0
testUnaPersona13	1.2	1	1,47	0,071	1,217	1	1	0
testUnaPersona13	1.2	3	1,469	0,07	1,185	1	1	0

testUnaPersona13	1.2	2	1,471	0,067	1,191	1	1	0
testUnaPersona13	1.2	4	1,476	0,07	1,217	1	1	0
testUnaPersona13	1.3	1	1,471	0,066	1,214	1	1	0
testUnaPersona13	1.3	2	1,47	0,069	1,192	1	1	0
testUnaPersona13	1.3	3	1,473	0,052	1,137	0	1	1
testUnaPersona13	1.3	4	1,47	0,051	1,24	0	1	1
testUnaPersona13	1.4	1	1,471	0,049	1,212	0	1	1
testUnaPersona13	1.4	2	1,471	0,047	1,222	0	1	1
testUnaPersona13	1.4	3	1,47	0,053	1,219	0	1	1
testUnaPersona13	1.4	4	1,471	0,048	1,235	0	1	1
testUnaPersona14	1.1	1	1,472	0,061	1,209	0	1	1
testUnaPersona14	1.1	2	1,469	0,06	1,228	0	1	1
testUnaPersona14	1.1	3	1,469	0,061	1,221	0	1	1
testUnaPersona14	1.1	4	1,472	0,057	1,222	0	1	1
testUnaPersona14	1.2	1	1,482	0,05	1,181	0	1	1
testUnaPersona14	1.2	2	1,471	0,056	1,219	0	1	1
testUnaPersona14	1.2	3	1,468	0,054	1,249	0	1	1
testUnaPersona14	1.2	4	1,474	0,053	1,219	0	1	1
testUnaPersona14	1.3	1	1,471	0,05	1,236	0	1	1
testUnaPersona14	1.3	2	1,467	0,055	1,229	0	1	1
testUnaPersona14	1.3	3	1,473	0,051	1,219	0	1	1
testUnaPersona14	1.3	4	1,471	0,05	1,254	0	1	1
testUnaPersona14	1.4	1	1,471	0,047	1,213	0	1	1
testUnaPersona14	1.4	2	1,473	0,049	1,211	0	1	1
testUnaPersona14	1.4	3	1,472	0,054	1,217	0	1	1
testUnaPersona14	1.4	4	1,472	0,112	1,057	0	1	1
testNoHumano01	1.1	1	1,551	0,139	1,204	0	0	0
testNoHumano01	1.1	2	1,47	0,06	1,198	0	0	0
testNoHumano01	1.1	3	1,512	0,123	1,196	0	0	0
testNoHumano01	1.1	4	1,468	0,056	1,21	0	0	0
testNoHumano01	1.2	1	1,469	0,052	1,222	0	0	0
testNoHumano01	1.2	2	1,472	0,168	1,204	0	0	0
testNoHumano01	1.2	3	1,471	0,053	1,178	0	0	0
testNoHumano01	1.2	4	1,468	0,076	1,203	0	0	0
testNoHumano01	1.3	1	1,47	0,052	1,199	0	0	0
testNoHumano01	1.3	2	1,47	0,053	1,211	0	0	0
testNoHumano01	1.3	3	1,47	0,052	1,216	0	0	0
testNoHumano01	1.3	4	1,469	0,05	1,192	0	0	0
testNoHumano01	1.4	1	1,468	0,05	1,207	0	0	0
testNoHumano01	1.4	2	1,473	0,05	1,224	0	0	0
testNoHumano01	1.4	3	1,472	0,055	1,024	0	0	0
testNoHumano01	1.4	4	1,476	0,052	1,168	0	0	0
testNoHumano04	1.1	1	1,471	0,123	1,182	1	0	-1
testNoHumano04	1.1	2	1,472	0,076	1,202	0	0	0
testNoHumano04	1.1	3	1,468	0,074	1,209	0	0	0
testNoHumano04	1.1	4	1,469	0,07	1,193	0	0	0
testNoHumano04	1.2	1	1,47	0,067	1,164	1	0	-1
testNoHumano04	1.2	2	1,471	0,065	1,208	1	0	-1
testNoHumano04	1.2	3	1,575	0,069	1,091	0	0	0
testNoHumano04	1.2	4	1,476	0,061	1,191	0	0	0
testNoHumano04	1.3	1	1,475	0,051	1,216	0	0	0
testNoHumano04	1.3	2	1,476	0,053	1,218	0	0	0

testNoHumano04	1.3	3	1,474	0,051	1,009	0	0	0
testNoHumano04	1.3	4	1,478	0,05	1,194	0	0	0
testNoHumano04	1.4	1	1,473	0,049	1,188	0	0	0
testNoHumano04	1.4	2	1,514	0,138	1,151	0	0	0
testNoHumano04	1.4	3	1,487	0,12	1,04	0	0	0
testNoHumano04	1.4	4	1,481	0,095	1,11	0	0	0
testDosPersonas02	1.1	1	1,513	0,487	1,011	2	2	0
testDosPersonas02	1.1	1	1,469	0,091	1,208	2	2	0
testDosPersonas02	1.1	2	1,468	0,077	1,19	1	2	1
testDosPersonas02	1.1	3	1,472	0,076	1,211	1	2	1
testDosPersonas02	1.1	4	1,47	0,092	1,144	1	2	1
testDosPersonas02	1.2	1	1,47	0,076	1,172	1	2	1
testDosPersonas02	1.2	2	1,469	0,081	1,213	1	2	1
testDosPersonas02	1.2	3	1,469	0,069	1,222	1	2	1
testDosPersonas02	1.2	4	1,472	0,071	1,164	1	2	1
testDosPersonas02	1.3	1	1,765	0,175	1,175	1	2	1
testDosPersonas02	1.3	2	1,471	0,086	1,213	1	2	1
testDosPersonas02	1.3	3	1,469	0,054	1,15	0	2	2
testDosPersonas02	1.3	4	1,469	0,048	1,222	0	2	2
testDosPersonas02	1.4	1	1,489	0,067	1,169	1	2	1
testDosPersonas02	1.4	2	1,467	0,06	1,221	1	2	1
testDosPersonas02	1.4	3	1,471	0,067	1,197	1	2	1
testDosPersonas02	1.4	4	1,468	0,068	1,069	1	2	1
testUnaPersona- NoHumano01	1.1	1	1,473	0,078	1,182	1	1	0
testUnaPersona- NoHumano01	1.1	2	1,47	0,078	1,218	1	1	0
testUnaPersona- NoHumano01	1.1	3	1,47	0,08	1,191	1	1	0
testUnaPersona- NoHumano01	1.1	4	1,471	0,076	1,164	1	1	0
testUnaPersona- NoHumano01	1.2	1	1,468	0,065	1,146	1	1	0
testUnaPersona- NoHumano01	1.2	2	1,468	0,079	1,218	1	1	0
testUnaPersona- NoHumano01	1.2	3	1,472	0,069	1,183	1	1	0
testUnaPersona- NoHumano01	1.2	4	1,472	0,076	1,2	1	1	0
testUnaPersona- NoHumano01	1.3	1	1,469	0,066	1,116	1	1	0
testUnaPersona- NoHumano01	1.3	2	1,48	0,063	1,197	1	1	0
testUnaPersona- NoHumano01	1.3	3	1,468	0,063	1,217	1	1	0
testUnaPersona- NoHumano02	1.1	1	1,468	0,074	1,199	1	1	0
testUnaPersona- NoHumano02	1.1	2	1,468	0,083	1,214	1	1	0
testUnaPersona- NoHumano02	1.1	3	1,472	0,083	1,208	1	1	0
testUnaPersona- NoHumano02	1.1	4	1,469	0,083	1,218	1	1	0
testUnaPersona-	1.2	1	1,469	0,068	1,186	1	1	0

NoHumano02								
testUnaPersona- NoHumano02	1.2	2	1,471	0,064	1,224	1	1	0
testUnaPersona- NoHumano02	1.2	3	1,469	0,069	1,218	1	1	0
testUnaPersona- NoHumano02	1.2	4	1,499	0,09	1,226	1	1	0
testUnaPersona- NoHumano02	1.3	1	1,472	0,072	1,172	1	1	0
testUnaPersona- NoHumano02	1.3	2	1,469	0,072	1,2	1	1	0
testUnaPersona- NoHumano02	1.3	3	1,467	0,065	1,232	1	1	0
testUnaPersona- NoHumano02	1.3	4	1,47	0,066	1,226	1	1	0
testUnaPersona- NoHumano02	1.4	1	1,479	0,07	1,16	1	1	0
testUnaPersona- NoHumano02	1.4	2	1,474	0,135	0,389	1	1	0
testUnaPersona- NoHumano02	1.4	3	1,473	0,148	1,19	1	1	0
testUnaPersona- NoHumano02	1.4	4	1,469	0,075	1,217	1	0	-1
			1,472	0,077	1,179	205	205	149

Tabla 1. Resultados de las pruebas de ejecución del método 1.

La última fila muestra para las columnas T.E., T.D. y T.R. el tiempo promedio en segundos, mientras que para las columnas O.D. y O.R. muestra el conteo de número de filas totales y para la columna R, se muestra el número total de celdas con valor 0.

4.4.2. PRUEBAS PARA EL MÉTODO 2

A continuación se muestra una tabla resumen con los resultados obtenidos del fichero CSV para la ejecución del método 2 sobre todas las imágenes y con todos los parámetros posibles.

La tabla contiene los mismos elementos que la tabla del apartado 4.4.1. En este caso la columna **O.R.** y **O.D.** cambian respecto al método 1 ya que en una imagen con una cara se detectan dos objetos, la cara y los ojos.

Imagen	F.E.	N.M.V.	T. E.	T. P.	T. R.	O. D.	O. R.	R.
testUnaPersona01	1.1	1	1,473	0,353	1,188	2	2	0
testUnaPersona01	1.1	2	1,475	0,32	1,181	2	2	0
testUnaPersona01	1.1	3	1,475	0,396	1,167	2	2	0
testUnaPersona01	1.1	4	1,475	0,338	1,189	2	2	0
testUnaPersona01	1.2	1	1,474	0,316	1,093	2	2	0
testUnaPersona01	1.2	2	1,478	0,231	1,162	2	2	0
testUnaPersona01	1.2	3	1,475	0,296	1,188	2	2	0
testUnaPersona01	1.2	4	1,473	0,339	1,196	2	2	0
testUnaPersona01	1.3	1	1,474	0,3	1,149	2	2	0
testUnaPersona01	1.3	2	1,444	0,198	1,19	2	2	0
testUnaPersona01	1.3	3	1,477	0,356	0,986	2	2	0

testUnaPersona01	1.3	4	1,446	0,346	0,404	2	2	0
testUnaPersona01	1.4	1	1,443	0,446	1,103	3	2	-1
testUnaPersona01	1.4	2	1,444	0,585	0,591	3	2	-1
testUnaPersona01	1.4	3	1,464	0,84	1,15	2	2	0
testUnaPersona01	1.4	4	1,44	0,322	1,18	2	2	0
testUnaPersona02	1.1	1	1,442	0,293	1,176	0	0	0
testUnaPersona02	1.1	2	1,442	0,336	1,11	0	0	0
testUnaPersona02	1.1	3	1,443	0,374	1,179	0	0	0
testUnaPersona02	1.1	4	1,443	0,31	1,146	0	0	0
testUnaPersona02	1.2	1	1,443	0,292	1,159	0	0	0
testUnaPersona02	1.2	2	1,443	0,301	1,176	0	0	0
testUnaPersona02	1.2	3	1,445	0,266	1,171	0	0	0
testUnaPersona02	1.2	4	1,446	0,298	1,166	0	0	0
testUnaPersona02	1.3	1	1,443	0,272	1,187	0	0	0
testUnaPersona02	1.3	2	1,445	0,254	1,168	0	0	0
testUnaPersona02	1.3	3	1,441	0,318	1,172	0	0	0
testUnaPersona02	1.3	4	1,444	0,784	0,966	0	0	0
testUnaPersona02	1.4	1	1,45	0,33	1,084	0	0	0
testUnaPersona02	1.4	2	1,441	0,209	1,162	0	0	0
testUnaPersona02	1.4	3	1,449	0,216	1,108	0	0	0
testUnaPersona02	1.4	4	1,447	0,211	1,179	0	0	0
testUnaPersona04	1.1	1	1,456	0,257	1,144	0	0	0
testUnaPersona04	1.1	2	1,444	0,258	1,144	0	0	0
testUnaPersona04	1.1	3	1,441	0,258	1,166	0	0	0
testUnaPersona04	1.1	4	1,443	0,254	1,188	0	0	0
testUnaPersona04	1.2	1	1,447	0,262	1,133	0	0	0
testUnaPersona04	1.2	2	1,445	0,253	1,161	0	0	0
testUnaPersona04	1.2	3	1,441	0,228	1,16	0	0	0
testUnaPersona04	1.2	4	1,442	0,212	1,155	0	0	0
testUnaPersona04	1.3	1	1,495	0,251	1,194	0	0	0
testUnaPersona04	1.3	2	1,447	0,236	1,018	0	0	0
testUnaPersona04	1.3	3	1,443	0,225	1,152	0	0	0
testUnaPersona04	1.3	4	1,445	0,228	1,189	0	0	0
testUnaPersona04	1.4	1	1,444	0,268	1,18	0	0	0
testUnaPersona04	1.4	2	1,445	0,25	1,096	0	0	0
testUnaPersona04	1.4	3	1,583	0,275	1,046	0	0	0
testUnaPersona04	1.4	4	1,483	0,278	1,076	0	0	0
testUnaPersona08	1.1	1	1,443	0,253	1,108	0	0	0
testUnaPersona08	1.1	2	1,443	0,228	1,179	0	0	0
testUnaPersona08	1.1	3	1,443	0,212	1,144	0	0	0
testUnaPersona08	1.1	4	1,445	0,228	1,166	0	0	0
testUnaPersona08	1.2	1	1,446	0,268	1,188	0	0	0
testUnaPersona08	1.2	2	1,444	0,25	1,133	0	0	0
testUnaPersona08	1.2	3	1,45	0,278	1,161	0	0	0
testUnaPersona08	1.2	4	1,441	0,253	1,16	0	0	0
testUnaPersona08	1.3	1	1,449	0,228	1,155	0	0	0
testUnaPersona08	1.3	2	1,447	0,212	1,194	0	0	0
testUnaPersona08	1.3	3	1,464	0,268	1,188	0	0	0
testUnaPersona08	1.3	4	1,44	0,25	1,196	0	0	0
testUnaPersona08	1.4	1	1,442	0,278	1,149	0	0	0
testUnaPersona08	1.4	2	1,442	0,253	1,19	0	0	0
testUnaPersona08	1.4	3	1,443	0,253	0,986	0	0	0

testUnaPersona08	1.4	4	1,443	0,228	0,404	0	0	0
testUnaPersona09	1.1	1	1,447	0,212	1,103	1	0	-1
testUnaPersona09	1.1	2	1,445	0,228	0,591	0	0	0
testUnaPersona09	1.1	3	1,441	0,278	1,15	0	0	0
testUnaPersona09	1.1	4	1,442	0,253	1,18	0	0	0
testUnaPersona09	1.2	1	1,495	0,228	1,176	0	0	0
testUnaPersona09	1.2	2	1,444	0,278	1,188	0	0	0
testUnaPersona09	1.2	3	1,445	0,253	1,133	0	0	0
testUnaPersona09	1.2	4	1,583	0,253	1,161	0	0	0
testUnaPersona09	1.3	1	1,443	0,228	1,16	0	0	0
testUnaPersona09	1.3	2	1,443	0,278	1,155	0	0	0
testUnaPersona09	1.3	3	1,443	0,253	1,194	0	0	0
testUnaPersona09	1.3	4	1,442	0,228	1,018	0	0	0
testUnaPersona09	1.4	1	1,442	0,212	1,152	0	0	0
testUnaPersona09	1.4	2	1,443	0,268	1,189	0	0	0
testUnaPersona09	1.4	3	1,443	0,228	1,16	0	0	0
testUnaPersona09	1.4	4	1,495	0,253	1,155	0	0	0
testUnaPersona10	1.1	1	1,444	0,339	1,194	1	2	1
testUnaPersona10	1.2	1	1,445	0,3	1,188	1	2	1
testUnaPersona10	1.1	2	1,583	0,198	1,196	1	2	1
testUnaPersona10	1.1	3	1,443	0,356	1,149	1	2	1
testUnaPersona10	1.1	4	1,445	0,346	1,19	1	2	1
testUnaPersona10	1.2	1	1,446	0,446	1,179	2	2	0
testUnaPersona10	1.2	2	1,444	0,253	1,144	2	2	0
testUnaPersona10	1.2	3	1,45	0,339	1,166	2	2	0
testUnaPersona10	1.2	4	1,441	0,3	1,188	2	2	0
testUnaPersona10	1.3	1	1,449	0,198	1,19	2	2	0
testUnaPersona10	1.3	2	1,444	0,346	0,986	2	2	0
testUnaPersona10	1.3	3	1,445	0,446	0,404	2	2	0
testUnaPersona10	1.3	4	1,583	0,396	1,103	2	2	0
testUnaPersona10	1.4	1	1,443	0,338	0,591	2	2	0
testUnaPersona10	1.4	2	1,445	0,316	1,15	2	2	0
testUnaPersona10	1.4	3	1,446	0,346	1,18	2	2	0
testUnaPersona10	1.4	4	1,444	0,446	1,155	2	2	0
testUnaPersona13	1.1	1	1,443	0,253	1,194	1	2	1
testUnaPersona13	1.1	2	1,443	0,339	1,018	2	2	0
testUnaPersona13	1.1	3	1,445	0,3	1,152	2	2	0
testUnaPersona13	1.1	4	1,446	0,198	1,189	2	2	0
testUnaPersona13	1.2	1	1,444	0,356	1,16	2	2	0
testUnaPersona13	1.2	3	1,45	0,346	1,149	2	2	0
testUnaPersona13	1.2	2	1,441	0,446	1,19	2	2	0
testUnaPersona13	1.2	4	1,449	0,396	1,179	2	2	0
testUnaPersona13	1.3	1	1,495	0,338	1,176	2	2	0
testUnaPersona13	1.3	2	1,444	0,316	1,188	2	2	0
testUnaPersona13	1.3	3	1,445	0,446	1,133	2	2	0
testUnaPersona13	1.3	4	1,583	0,253	1,161	2	2	0
testUnaPersona13	1.4	1	1,443	0,339	1,16	2	2	0
testUnaPersona13	1.4	2	1,445	0,346	1,155	2	2	0
testUnaPersona13	1.4	3	1,446	0,446	1,194	2	2	0
testUnaPersona13	1.4	4	1,445	0,396	1,018	2	2	0
testUnaPersona14	1.1	1	1,583	0,338	1,152	1	2	1
testUnaPersona14	1.1	2	1,443	0,316	1,144	1	2	1

testUnaPersona14	1.1	3	1,445	0,346	1,166	1	2	1
testUnaPersona14	1.1	4	1,446	0,396	1,188	1	2	1
testUnaPersona14	1.2	1	1,444	0,338	1,19	1	2	1
testUnaPersona14	1.2	2	1,583	0,316	0,986	1	2	1
testUnaPersona14	1.2	3	1,443	0,446	0,404	2	2	0
testUnaPersona14	1.2	4	1,445	0,253	1,103	2	2	0
testUnaPersona14	1.3	1	1,446	0,339	0,591	2	2	0
testUnaPersona14	1.3	2	1,441	0,346	1,15	2	2	0
testUnaPersona14	1.3	3	1,449	0,338	1,155	2	2	0
testUnaPersona14	1.3	4	1,444	0,316	1,194	2	2	0
testUnaPersona14	1.4	1	1,445	0,346	1,188	2	2	0
testUnaPersona14	1.4	2	1,583	0,316	1,196	2	2	0
testUnaPersona14	1.4	3	1,443	0,446	1,152	2	2	0
testUnaPersona14	1.4	4	1,445	0,253	1,189	2	2	0
testNoHumano01	1.1	1	1,446	0,253	1,16	0	0	0
testNoHumano01	1.1	2	1,444	0,228	1,149	0	0	0
testNoHumano01	1.1	3	1,443	0,212	1,19	0	0	0
testNoHumano01	1.1	4	1,445	0,268	1,179	0	0	0
testNoHumano01	1.2	1	1,446	0,25	0,591	0	0	0
testNoHumano01	1.2	2	1,444	0,278	1,15	0	0	0
testNoHumano01	1.2	3	1,583	0,253	1,155	0	0	0
testNoHumano01	1.2	4	1,443	0,253	1,194	0	0	0
testNoHumano01	1.3	1	1,445	0,228	1,16	0	0	0
testNoHumano01	1.3	2	1,449	0,212	1,149	0	0	0
testNoHumano01	1.3	3	1,444	0,228	1,19	0	0	0
testNoHumano01	1.3	4	1,445	0,278	1,179	0	0	0
testNoHumano01	1.4	1	1,444	0,253	1,176	0	0	0
testNoHumano01	1.4	2	1,443	0,212	1,188	0	0	0
testNoHumano01	1.4	3	1,443	0,268	1,133	0	0	0
testNoHumano01	1.4	4	1,445	0,25	1,161	0	0	0
testNoHumano04	1.1	1	1,446	0,228	1,16	0	0	0
testNoHumano04	1.1	2	1,444	0,278	1,155	0	0	0
testNoHumano04	1.1	3	1,45	0,253	1,194	0	0	0
testNoHumano04	1.1	4	1,441	0,212	1,018	0	0	0
testNoHumano04	1.2	1	1,443	0,268	1,152	0	0	0
testNoHumano04	1.2	2	1,445	0,228	1,144	0	0	0
testNoHumano04	1.2	3	1,446	0,278	1,166	0	0	0
testNoHumano04	1.2	4	1,444	0,253	1,16	0	0	0
testNoHumano04	1.3	1	1,583	0,253	1,149	0	0	0
testNoHumano04	1.3	2	1,449	0,212	1,19	0	0	0
testNoHumano04	1.3	3	1,444	0,268	1,179	0	0	0
testNoHumano04	1.3	4	1,445	0,25	0,591	0	0	0
testNoHumano04	1.4	1	1,583	0,268	1,15	0	0	0
testNoHumano04	1.4	2	1,443	0,228	1,155	0	0	0
testNoHumano04	1.4	3	1,445	0,278	1,194	0	0	0
testNoHumano04	1.4	4	1,446	0,253	1,16	0	0	0
testDosPersonas02	1.1	1	1,444	0,253	1,149	3	4	1
testDosPersonas02	1.1	1	1,443	0,546	1,19	3	4	1
testDosPersonas02	1.1	2	1,445	0,689	1,179	3	4	1
testDosPersonas02	1.1	3	1,446	0,52	1,188	4	4	0
testDosPersonas02	1.1	4	1,444	0,652	1,133	4	4	0
testDosPersonas02	1.2	1	1,583	0,495	1,161	4	4	0

testDosPersonas02	1.2	2	1,443	0,598	1,16	4	4	0
testDosPersonas02	1.2	3	1,445	0,563	1,155	4	4	0
testDosPersonas02	1.2	4	1,449	0,758	1,194	4	4	0
testDosPersonas02	1.3	1	1,443	0,654	1,018	4	4	0
testDosPersonas02	1.3	2	1,445	0,654	1,155	4	4	0
testDosPersonas02	1.3	3	1,446	0,652	1,194	4	4	0
testDosPersonas02	1.3	4	1,444	0,495	1,16	4	4	0
testDosPersonas02	1.4	1	1,583	0,598	1,149	4	4	0
testDosPersonas02	1.4	2	1,449	0,758	1,19	4	4	0
testDosPersonas02	1.4	3	1,444	0,654	1,179	4	4	0
testDosPersonas02	1.4	4	1,445	0,758	1,161	4	4	0
testUnaPersona- NoHumano01	1.1	1	1,583	0,356	1,16	2	2	0
testUnaPersona- NoHumano01	1.1	2	1,443	0,346	1,155	2	2	0
testUnaPersona- NoHumano01	1.1	3	1,444	0,446	1,194	2	2	0
testUnaPersona- NoHumano01	1.1	4	1,443	0,585	1,018	2	2	0
testUnaPersona- NoHumano01	1.2	1	1,445	0,84	1,155	2	2	0
testUnaPersona- NoHumano01	1.2	2	1,446	0,346	1,179	2	2	0
testUnaPersona- NoHumano01	1.2	3	1,444	0,446	0,591	2	2	0
testUnaPersona- NoHumano01	1.2	4	1,583	0,84	1,15	2	2	0
testUnaPersona- NoHumano01	1.3	1	1,443	0,346	1,155	2	2	0
testUnaPersona- NoHumano01	1.3	2	1,445	0,495	1,194	2	2	0
testUnaPersona- NoHumano01	1.3	3	1,449	0,598	1,16	2	2	0
testUnaPersona- NoHumano02	1.1	1	1,443	0,346	1,149	2	2	0
testUnaPersona- NoHumano02	1.1	2	1,583	0,446	1,19	2	2	0
testUnaPersona- NoHumano02	1.1	3	1,449	0,346	1,179	2	2	0
testUnaPersona- NoHumano02	1.1	4	1,444	0,346	1,188	2	2	0
testUnaPersona- NoHumano02	1.2	1	1,445	0,446	1,133	2	2	0
testUnaPersona- NoHumano02	1.2	2	1,583	0,585	1,161	2	2	0
testUnaPersona- NoHumano02	1.2	3	1,443	0,346	1,16	2	2	0
testUnaPersona- NoHumano02	1.2	4	1,444	0,495	1,155	2	2	0
testUnaPersona- NoHumano02	1.3	1	1,443	0,598	1,194	2	2	0
testUnaPersona- NoHumano02	1.3	2	1,444	0,346	1,16	2	2	0
testUnaPersona- NoHumano02	1.3	3	1,443	0,446	1,149	2	2	0

testUnaPersona- NoHumano02	1.3	4	1,445	0,346	1,19	2	2	0
testUnaPersona- NoHumano02	1.4	1	1,446	0,84	1,179	2	2	0
testUnaPersona- NoHumano02	1.4	2	1,444	0,346	1,161	2	2	0
testUnaPersona- NoHumano02	1.4	3	1,583	0,446	1,16	2	2	0
testUnaPersona- NoHumano02	1.4	4	1,443	0,585	1,16	2	2	0
Total			1,460	0,345	1,120	205	205	187

Tabla 2. Resultados de las pruebas de ejecución del método 2.

4.4.3. PRUEBAS PARA EL MÉTODO 3

A continuación se muestra una tabla resumen con los resultados obtenidos del fichero CSV para la ejecución del método 2 sobre todas las imágenes y con todos los parámetros posibles.

La tabla contiene los mismos elementos que la tabla del apartado 4.4.1.

Imagen	F.E.	N.M.V.	T. E.	T.P.	T. R.	O. D.	O. R.	R.
testUnaPersona01	1.1	1	1,922	0,353	0,903	1	1	0
testUnaPersona01	1.1	2	1,474	0,345	1,149	1	1	0
testUnaPersona01	1.1	3	1,478	0,42	1,19	1	1	0
testUnaPersona01	1.1	4	1,475	0,368	0,986	1	1	0
testUnaPersona01	1.2	1	1,473	0,353	0,404	1	1	0
testUnaPersona01	1.2	2	1,474	0,302	1,103	1	1	0
testUnaPersona01	1.2	3	1,444	0,356	0,591	1	1	0
testUnaPersona01	1.2	4	1,477	0,401	1,15	1	1	0
testUnaPersona01	1.3	1	1,446	0,353	1,18	1	1	0
testUnaPersona01	1.3	2	1,443	0,345	1,176	1	1	0
testUnaPersona01	1.3	3	1,444	0,298	1,11	1	1	0
testUnaPersona01	1.3	4	1,464	0,345	1,179	1	1	0
testUnaPersona01	1.4	1	1,44	0,324	1,146	1	1	0
testUnaPersona01	1.4	2	1,442	0,321	1,159	1	1	0
testUnaPersona01	1.4	3	1,442	0,358	1,176	1	1	0
testUnaPersona01	1.4	4	1,443	0,398	1,171	1	1	0
testUnaPersona02	1.1	1	1,443	0,198	1,166	0	0	0
testUnaPersona02	1.1	2	1,443	0,23	1,187	0	0	0
testUnaPersona02	1.1	3	1,443	0,234	1,168	0	0	0
testUnaPersona02	1.1	4	1,445	0,301	1,172	0	0	0
testUnaPersona02	1.2	1	1,446	0,199	0,966	0	0	0
testUnaPersona02	1.2	2	1,443	0,234	1,084	0	0	0
testUnaPersona02	1.2	3	1,445	0,256	1,162	0	0	0
testUnaPersona02	1.2	4	1,441	0,289	1,108	0	0	0
testUnaPersona02	1.3	1	1,444	0,236	1,179	0	0	0
testUnaPersona02	1.3	2	1,45	0,298	1,179	0	0	0
testUnaPersona02	1.3	3	1,441	0,236	1,146	0	0	0
testUnaPersona02	1.3	4	1,449	0,214	1,159	0	0	0
testUnaPersona02	1.4	1	1,447	0,268	1,176	0	0	0
testUnaPersona02	1.4	2	1,456	0,261	1,171	0	0	0
testUnaPersona02	1.4	3	1,444	0,291	1,166	0	0	0

testUnaPersona02	1.4	4	1,441	0,302	1,187	0	0	0
testUnaPersona04	1.1	1	1,444	0,234	1,168	0	0	0
testUnaPersona04	1.1	2	1,477	0,301	1,172	0	0	0
testUnaPersona04	1.1	3	1,446	0,199	0,966	0	0	0
testUnaPersona04	1.1	4	1,443	0,234	1,084	0	0	0
testUnaPersona04	1.2	1	1,444	0,256	1,162	0	0	0
testUnaPersona04	1.2	2	1,464	0,236	1,108	0	0	0
testUnaPersona04	1.2	3	1,44	0,214	1,179	0	0	0
testUnaPersona04	1.2	4	1,442	0,268	1,179	0	0	0
testUnaPersona04	1.3	1	1,442	0,256	1,146	0	0	0
testUnaPersona04	1.3	2	1,443	0,289	1,159	0	0	0
testUnaPersona04	1.3	3	1,478	0,268	1,168	0	0	0
testUnaPersona04	1.3	4	1,477	0,261	1,172	0	0	0
testUnaPersona04	1.4	1	1,443	0,291	0,966	0	0	0
testUnaPersona04	1.4	2	1,445	0,302	1,084	0	0	0
testUnaPersona04	1.4	3	1,441	0,234	1,162	0	0	0
testUnaPersona04	1.4	4	1,456	0,256	1,108	0	0	0
testUnaPersona08	1.1	1	1,444	0,236	1,179	0	0	0
testUnaPersona08	1.1	2	1,441	0,214	1,179	0	0	0
testUnaPersona08	1.1	3	1,444	0,256	1,146	0	0	0
testUnaPersona08	1.1	4	1,477	0,289	1,187	0	0	0
testUnaPersona08	1.2	1	1,446	0,236	1,168	0	0	0
testUnaPersona08	1.2	2	1,443	0,298	1,172	0	0	0
testUnaPersona08	1.2	3	1,464	0,236	0,966	0	0	0
testUnaPersona08	1.2	4	1,44	0,214	1,084	0	0	0
testUnaPersona08	1.3	1	1,442	0,234	1,162	0	0	0
testUnaPersona08	1.3	2	1,442	0,256	1,108	0	0	0
testUnaPersona08	1.3	3	1,443	0,236	1,179	0	0	0
testUnaPersona08	1.3	4	1,443	0,214	1,179	0	0	0
testUnaPersona08	1.4	1	1,443	0,268	1,146	0	0	0
testUnaPersona08	1.4	2	1,44	0,291	1,159	0	0	0
testUnaPersona08	1.4	3	1,445	0,302	1,176	0	0	0
testUnaPersona08	1.4	4	1,441	0,234	1,171	0	0	0
testUnaPersona09	1.1	1	1,477	0,256	1,166	0	0	0
testUnaPersona09	1.1	2	1,443	0,256	1,187	0	0	0
testUnaPersona09	1.1	3	1,446	0,289	1,168	0	0	0
testUnaPersona09	1.1	4	1,443	0,236	1,172	0	0	0
testUnaPersona09	1.2	1	1,444	0,298	0,966	0	0	0
testUnaPersona09	1.2	2	1,464	0,236	1,084	0	0	0
testUnaPersona09	1.2	3	1,44	0,236	1,162	0	0	0
testUnaPersona09	1.2	4	1,442	0,214	1,108	0	0	0
testUnaPersona09	1.3	1	1,442	0,268	1,179	0	0	0
testUnaPersona09	1.3	2	1,443	0,234	1,179	0	0	0
testUnaPersona09	1.3	3	1,478	0,256	1,146	0	0	0
testUnaPersona09	1.3	4	1,477	0,256	0,966	0	0	0
testUnaPersona09	1.4	1	1,443	0,289	1,084	0	0	0
testUnaPersona09	1.4	2	1,444	0,236	1,162	0	0	0
testUnaPersona09	1.4	3	1,441	0,236	1,108	0	0	0
testUnaPersona09	1.4	4	1,444	0,214	1,179	0	0	0
testUnaPersona10	1.1	1	1,477	0,365	1,179	0	1	1
testUnaPersona10	1.2	1	1,446	0,256	1,146	0	1	1
testUnaPersona10	1.1	2	1,443	0,398	1,187	1	1	0

testUnaPersona10	1.1	3	1,464	0,356	1,168	1	1	0
testUnaPersona10	1.1	4	1,445	0,401	1,172	1	1	0
testUnaPersona10	1.2	1	1,464	0,353	0,966	1	1	0
testUnaPersona10	1.2	2	1,443	0,345	1,179	1	1	0
testUnaPersona10	1.2	3	1,443	0,298	1,179	1	1	0
testUnaPersona10	1.2	4	1,477	0,256	1,146	1	1	0
testUnaPersona10	1.3	1	1,443	0,398	1,159	1	1	0
testUnaPersona10	1.3	2	1,446	0,353	1,176	1	1	0
testUnaPersona10	1.3	3	1,443	0,345	1,171	1	1	0
testUnaPersona10	1.3	4	1,444	0,298	1,166	1	1	0
testUnaPersona10	1.4	1	1,464	0,356	1,187	1	1	0
testUnaPersona10	1.4	2	1,44	0,401	1,168	1	1	0
testUnaPersona10	1.4	3	1,442	0,353	1,172	1	1	0
testUnaPersona10	1.4	4	1,442	0,345	0,966	1	1	0
testUnaPersona13	1.1	1	1,443	0,298	0,966	0	1	1
testUnaPersona13	1.1	2	1,478	0,398	1,084	0	1	1
testUnaPersona13	1.1	3	1,477	0,353	1,162	0	1	1
testUnaPersona13	1.1	4	1,443	0,353	1,108	1	1	0
testUnaPersona13	1.2	1	1,444	0,345	1,179	1	1	0
testUnaPersona13	1.2	3	1,444	0,298	1,179	1	1	0
testUnaPersona13	1.2	2	1,477	0,256	1,146	1	1	0
testUnaPersona13	1.2	4	1,446	0,298	0,966	1	1	0
testUnaPersona13	1.3	1	1,443	0,356	1,084	1	1	0
testUnaPersona13	1.3	2	1,464	0,401	1,162	1	1	0
testUnaPersona13	1.3	3	1,445	0,353	1,108	1	1	0
testUnaPersona13	1.3	4	1,464	0,345	1,179	1	1	0
testUnaPersona13	1.4	1	1,44	0,298	1,187	1	1	0
testUnaPersona13	1.4	2	1,442	0,398	1,168	1	1	0
testUnaPersona13	1.4	3	1,442	0,353	1,172	1	1	0
testUnaPersona13	1.4	4	1,443	0,353	0,966	1	1	0
testUnaPersona14	1.1	1	1,478	0,345	1,179	0	1	1
testUnaPersona14	1.1	2	1,477	0,298	1,179	0	1	1
testUnaPersona14	1.1	3	1,443	0,356	1,146	1	1	0
testUnaPersona14	1.1	4	1,445	0,401	1,159	1	1	0
testUnaPersona14	1.2	1	1,464	0,353	1,176	1	1	0
testUnaPersona14	1.2	2	1,443	0,345	1,171	1	1	0
testUnaPersona14	1.2	3	1,444	0,298	1,166	1	1	0
testUnaPersona14	1.2	4	1,446	0,398	1,176	1	1	0
testUnaPersona14	1.3	1	1,443	0,353	1,187	1	1	0
testUnaPersona14	1.3	2	1,464	0,298	1,168	1	1	0
testUnaPersona14	1.3	3	1,445	0,356	1,172	1	1	0
testUnaPersona14	1.3	4	1,464	0,401	0,966	1	1	0
testUnaPersona14	1.4	1	1,44	0,353	1,179	1	1	0
testUnaPersona14	1.4	2	1,442	0,356	1,179	1	1	0
testUnaPersona14	1.4	3	1,442	0,401	1,162	1	1	0
testUnaPersona14	1.4	4	1,443	0,298	1,108	1	1	0
testNoHumano01	1.1	1	1,444	0,298	1,179	0	0	0
testNoHumano01	1.1	2	1,444	0,236	1,179	0	0	0
testNoHumano01	1.1	3	1,477	0,214	1,146	0	0	0
testNoHumano01	1.1	4	1,446	0,234	0,966	0	0	0
testNoHumano01	1.2	1	1,443	0,256	1,084	0	0	0
testNoHumano01	1.2	2	1,464	0,236	1,162	0	0	0

testNoHumano01	1.2	3	1,445	0,214	1,108	0	0	0
testNoHumano01	1.2	4	1,464	0,214	1,179	0	0	0
testNoHumano01	1.3	1	1,44	0,234	1,187	0	0	0
testNoHumano01	1.3	2	1,442	0,256	1,168	0	0	0
testNoHumano01	1.3	3	1,442	0,234	1,172	0	0	0
testNoHumano01	1.3	4	1,443	0,214	1,179	0	0	0
testNoHumano01	1.4	1	1,478	0,256	1,187	0	0	0
testNoHumano01	1.4	2	1,477	0,236	1,168	0	0	0
testNoHumano01	1.4	3	1,445	0,214	1,172	0	0	0
testNoHumano01	1.4	4	1,464	0,256	0,966	0	0	0
testNoHumano04	1.1	1	1,443	0,236	1,179	0	0	0
testNoHumano04	1.1	2	1,444	0,214	1,179	0	0	0
testNoHumano04	1.1	3	1,446	0,214	1,146	0	0	0
testNoHumano04	1.1	4	1,443	0,234	1,159	0	0	0
testNoHumano04	1.2	1	1,464	0,236	1,176	0	0	0
testNoHumano04	1.2	2	1,445	0,214	1,171	0	0	0
testNoHumano04	1.2	3	1,464	0,256	1,166	0	0	0
testNoHumano04	1.2	4	1,44	0,236	1,176	0	0	0
testNoHumano04	1.3	1	1,442	0,214	1,187	0	0	0
testNoHumano04	1.3	2	1,442	0,234	1,168	0	0	0
testNoHumano04	1.3	3	1,443	0,214	1,172	0	0	0
testNoHumano04	1.3	4	1,478	0,256	1,179	0	0	0
testNoHumano04	1.4	1	1,477	0,236	1,179	0	0	0
testNoHumano04	1.4	2	1,445	0,214	1,162	0	0	0
testNoHumano04	1.4	3	1,464	0,234	1,108	0	0	0
testNoHumano04	1.4	4	1,443	0,236	1,179	0	0	0
testDosPersonas02	1.1	1	1,443	0,689	1,179	0	2	2
testDosPersonas02	1.1	1	1,445	0,656	1,146	0	2	2
testDosPersonas02	1.1	2	1,444	0,598	0,966	2	2	0
testDosPersonas02	1.1	3	1,446	0,698	1,084	2	2	0
testDosPersonas02	1.1	4	1,464	0,656	1,162	2	2	0
testDosPersonas02	1.2	1	1,464	0,698	1,108	2	2	0
testDosPersonas02	1.2	2	1,445	0,632	1,187	2	2	0
testDosPersonas02	1.2	3	1,464	0,654	1,168	2	2	0
testDosPersonas02	1.2	4	1,44	0,697	1,172	2	2	0
testDosPersonas02	1.3	1	1,442	0,698	0,966	2	2	0
testDosPersonas02	1.3	2	1,442	0,656	1,179	2	2	0
testDosPersonas02	1.3	3	1,443	0,698	1,179	2	2	0
testDosPersonas02	1.3	4	1,478	0,632	1,146	2	2	0
testDosPersonas02	1.4	1	1,477	0,654	1,159	2	2	0
testDosPersonas02	1.4	2	1,445	0,697	1,176	2	2	0
testDosPersonas02	1.4	3	1,445	0,698	1,171	2	2	0
testDosPersonas02	1.4	4	1,444	0,697	1,166	2	2	0
testUnaPersona- NoHumano01	1.1	1	1,446	0,598	1,176	0	1	1
testUnaPersona- NoHumano01	1.1	2	1,464	0,456	1,187	0	1	1
testUnaPersona- NoHumano01	1.1	3	1,464	0,65	1,168	0	1	1
testUnaPersona- NoHumano01	1.1	4	1,445	0,63	1,172	1	1	0
testUnaPersona- NoHumano01	1.2	1	1,464	0,456	1,162	1	1	0

testUnaPersona- NoHumano01	1.2	2	1,44	0,65	1,108	1	1	0
testUnaPersona- NoHumano01	1.2	3	1,442	0,65	1,179	1	1	0
testUnaPersona- NoHumano01	1.2	4	1,442	0,456	1,179	1	1	0
testUnaPersona- NoHumano01	1.3	1	1,443	0,63	1,146	1	1	0
testUnaPersona- NoHumano01	1.3	2	1,443	0,456	0,966	1	1	0
testUnaPersona- NoHumano01	1.3	3	1,445	0,65	1,084	1	1	0
testUnaPersona- NoHumano02	1.1	1	1,444	0,598	1,162	1	1	0
testUnaPersona- NoHumano02	1.1	2	1,446	0,456	1,108	1	1	0
testUnaPersona- NoHumano02	1.1	3	1,464	0,65	1,187	1	1	0
testUnaPersona- NoHumano02	1.1	4	1,464	0,456	1,168	1	1	0
testUnaPersona- NoHumano02	1.2	1	1,445	0,456	1,172	1	1	0
testUnaPersona- NoHumano02	1.2	2	1,464	0,65	0,966	1	1	0
testUnaPersona- NoHumano02	1.2	3	1,44	0,698	1,179	1	1	0
testUnaPersona- NoHumano02	1.2	4	1,442	0,63	1,179	1	1	0
testUnaPersona- NoHumano02	1.3	1	1,442	0,654	1,179	1	1	0
testUnaPersona- NoHumano02	1.3	2	1,443	0,659	1,146	1	1	0
testUnaPersona- NoHumano02	1.3	3	1,478	0,658	1,159	1	1	0
testUnaPersona- NoHumano02	1.3	4	1,477	0,698	1,176	1	1	0
testUnaPersona- NoHumano02	1.4	1	1,445	0,698	1,171	1	1	0
testUnaPersona- NoHumano02	1.4	2	1,445	0,63	1,166	1	1	0
testUnaPersona- NoHumano02	1.4	3	1,442	0,654	1,176	1	1	0
testUnaPersona- NoHumano02	1.4	4	1,442	0,659	1,187	1	1	0
Total			1,453	0,360	1,134	205	205	193

Tabla 3. Resultados de las pruebas de ejecución del método 3.

4.4.4. COMPARACIÓN DE MÉTODOS

A continuación, se muestra una tabla resumen y las gráficas correspondientes relativas a la comparación de métodos en promedios de tiempos de envío, de procesamiento y de recepción, así como tasa de aciertos.

	T.E.	T.P.	T.R.	Tasa de Aciertos
Método 1	1,472	0,077	1,179	72,68%
Método 2	1,46	0,345	1,12	91,22%
Método 3	1,453	0,36	1,134	94,15%

Tabla 4. Resumen de los resultados obtenidos en la ejecución del método 1, 2 y 3.

4.4.4.1. GRÁFICA COMPARATIVA DE TIEMPO DE ENVÍO

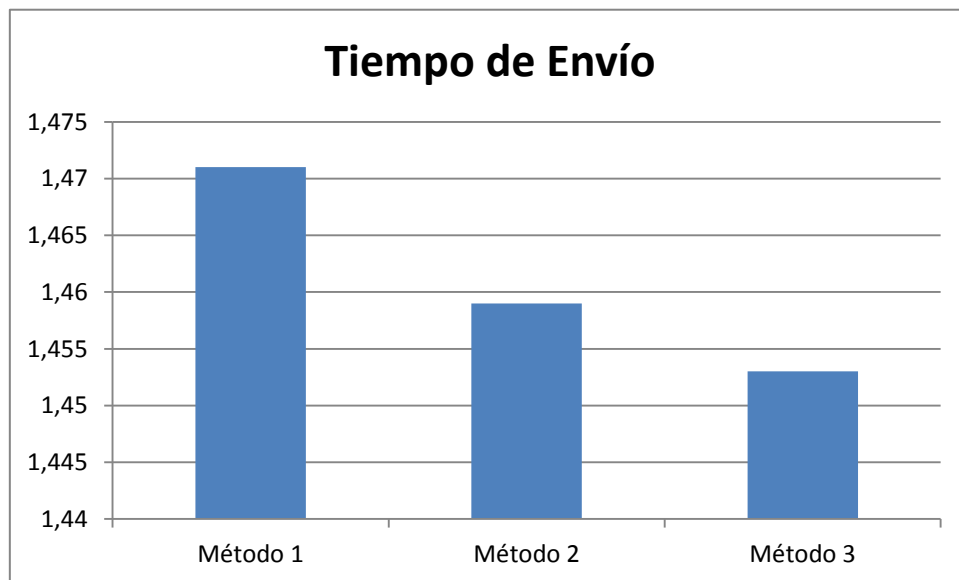


Figura 32. Gráfica comparativa del Tiempo de Envío

4.4.4.2. GRÁFICA COMPARATIVA DE TIEMPO DE PROCESAMIENTO

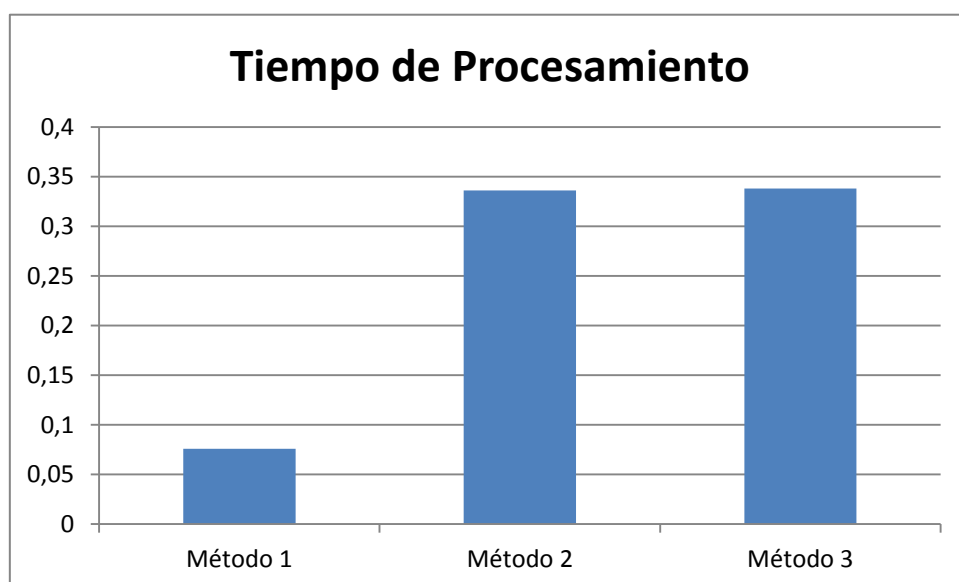


Figura 33. Gráfica comparativa del Tiempo de Procesamiento.

4.4.4.3. GRÁFICA COMPARATIVA DE TIEMPO DE RECEPCIÓN

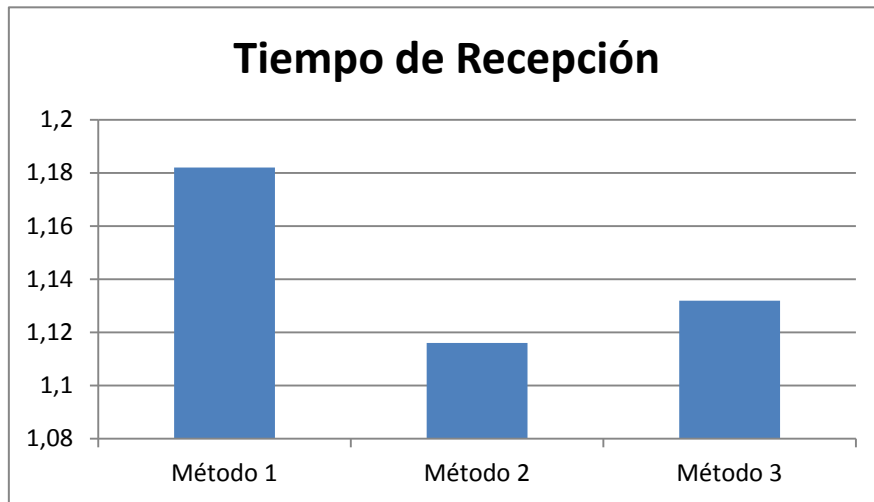


Figura 34. Gráfica comparativa del Tiempo de Recepción.

4.4.4.4. GRÁFICA COMPARATIVA DE TASA DE ACIERTO

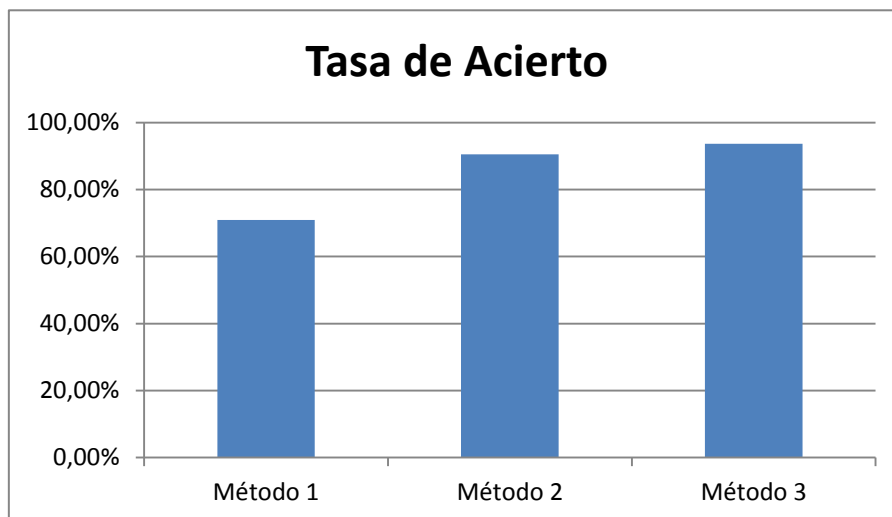


Figura 35. Gráfica comparativa de la Tasa de Acierto.

A la vista de los resultados obtenidos tras la realización de las pruebas nos aportan la siguiente información:

El método con **mayor tasa de acierto** es el **método 3**, aquel en el cual se detecta la cara para posteriormente estimar la posición de los ojos. También es el método, junto al método 2, que **más tiempo de procesamiento consume**. Es cierto que el método 1 posee un tiempo de procesamiento muy bajo, pero el 70% en la tasa de acierto le aleja de ser un método útil para este trabajo. La diferencia de tiempo de procesamiento y tasa de acierto entre el método 2 y 3 es muy pequeña, por lo que cualquiera de los dos puede ser válido para este trabajo.

En cuanto a los tiempos de envío y recepción se puede observar cómo no existen grandes diferencias entre los distintos métodos, ya que no depende tanto del método sino de la red en la que se ejecute, y como se puede comprobar son tiempos bastante elevados (por encima de 1 segundo). Se puede observar cómo el **tiempo total** (sumando tiempo de envío, más tiempo de procesamiento junto al

tiempo de recepción) también es bastante **elevado**, debido a los tiempos de envío y recepción, que son los que aumentan este tiempo total. Este es el punto débil que se ha encontrado en el sistema implementado de probador virtual. Esos tiempos no deberían ser tan altos si se quiere producir un efecto de tiempo real.

En las figuras Figura 36, Figura 37, Figura 38 y Figura 39 se muestran los resultados finales de la superposición de las gafas sobre imágenes originales, que son devueltos al usuario para su visualización. Dichas imágenes son representativas del conjunto de imágenes reales utilizadas y pretenden mostrar algunas deficiencias del método propuesto en su conjunto con vistas a las mejoras de futuro.



Figura 36. Superposición de gafas 1.

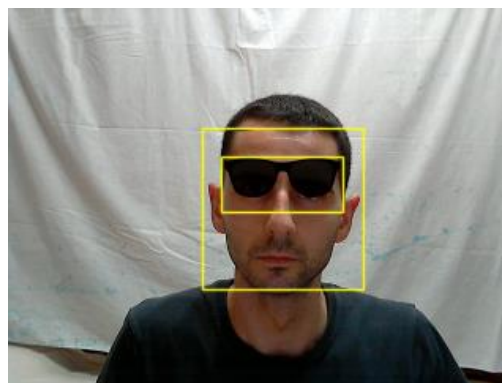


Figura 37. Superposición de gafas 2.

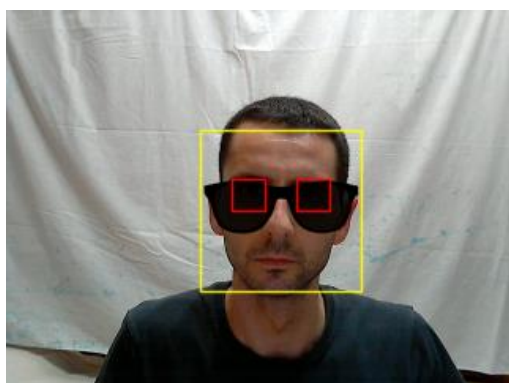


Figura 38. Superposición de gafas 3.

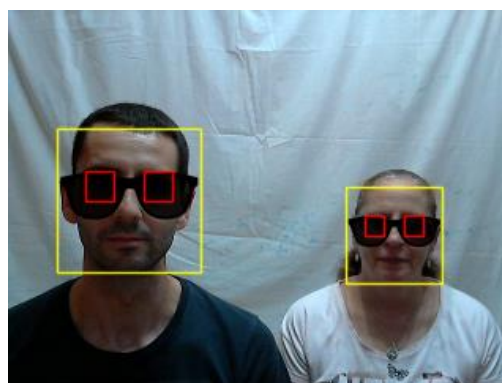


Figura 39. Superposición de gafas 4.

La primera imagen (Figura 36) muestra el resultado de aplicar el método 1 a la imagen “testUnaPersona01.png”. Se observa cómo la superposición y el re-escalado no resulta del todo correcto, si bien la detección de los ojos sí se hace correctamente. Esta situación se presenta en aproximadamente un 60% de las imágenes procesadas con el método 1, cuyo origen se sitúa en el propio método de detección.

La siguiente imagen (Figura 37) muestra el resultado de aplicar el método 2 también a la imagen “testUnaPersona01.png”. En este caso, aparte de detectar los ojos, primero se detecta la cara. Al igual que en el método 1, la superposición y re-escalado no son del todo correctos.

Las últimas imágenes (Figura 38 y Figura 39) muestran los resultados de aplicar el método 3 a las imágenes “testUnaPersona01.png” y “testDosPersonas.png”. En este caso el posicionamiento y re-escalado se realizan mejor que en el método 1, aunque el re-escalado no resulta ser el más óptimo como se puede comprobar en la Figura 37 y Figura 38 donde las gafas son de un tamaño algo mayor del que cabría esperar. En este caso, el problema procede de la integración entre detección y re-escalado,

requiriendo un esfuerzo en este sentido, que vendría por adaptar el re-escalado aplicado en el método 1 para el método 3. Este tipo de situaciones aparecen en aproximadamente el 70% de las imágenes procesadas.

5. CONCLUSIONES Y TRABAJO FUTURO

5.1. CONCLUSIONES

A modo de conclusión general, tras la realización del proyecto, se han obtenido las siguientes conclusiones:

- Para la realización del proyecto se han utilizado trabajos solventes, publicados y reputados en el ámbito del mismo.
- Se ha definido el problema y se ha establecido la base de conocimiento necesaria para su diseño atendiendo al estado de la cuestión.
- La solución de diseño propuesta constituye una aportación eficiente para el problema planteado, de forma que hasta donde se ha podido explorar, no se han encontrado desarrollos previos similares, que aborden la misma problemática relativa a un probador virtual. Si bien aunque existen soluciones próximas, tal como se ha descrito en el estado del arte, la presente aproximación realiza las aportaciones allí descritas.

5.2. TRABAJO FUTURO

Aunque se han conseguido resultados satisfactorios, todavía existen algunos aspectos y cuestiones dignos de consideración en el futuro.

De esta forma, como trabajo futuro, se podrían ampliar y/o corregir los siguientes aspectos, que se distinguen desde dos puntos de vista cuales son los aspectos relacionados con la imagen, su captura, procesamiento y acondicionamiento final y los relacionados con los sistemas que llevan a cabo las tareas de envío y recepción entre los dispositivos móviles y el servidor.

Desde el punto de vista de la imagen destacan:

- Se podría mejorar el posicionamiento de las gafas, no sólo con una estimación de los ojos o la detección de los mismos, sino buscar alternativas para un eficaz posicionamiento.
- En la línea anterior, resulta necesario analizar a fondo los tres métodos propuestos para aprovechar las mejoras de los tres en uno sólo.
- Otra de las mejoras a considerar es conseguir mayores niveles de automatización en el sentido de que sea el propio sistema el que detecte una imagen con baja calidad o un posicionamiento incorrecto del usuario frente a la cámara, con el fin de recomendar la captura de una nueva imagen antes de su envío al servidor para su procesamiento. Respecto de la baja calidad se plantea un tratamiento basado en el análisis de los histogramas de los canales de color. En lo que se refiere al posicionamiento, es necesario considerar puntos de referencia sobre el fondo o bien un procesamiento secuencial de varias imágenes de forma que si en ninguna de ellas se detecta la presencia de la cara, se realice la correspondiente recomendación de corrección.
- Toda la problemática identificada en la sección 2.3 y relacionada con el pose, rasgos y expresiones faciales, oclusiones, condiciones de captura o incluso la presencia de varias caras son temas para abordar en el futuro para ampliar las posibilidades de uso del probador virtual

Desde el punto de vista de los dispositivos y el servidor, cabe mencionar:

- La implementación en un entorno real (desplegando la aplicación en un servidor virtual) muestra carencias en cuanto al tiempo total del proceso (probablemente por la latencia del servidor o porque no acepte un número alto de conexiones en tan corto espacio de tiempo) y la

sensación de tiempo real no se consigue. Este problema todavía no ha sido identificado y no ha sido solucionado, por lo que queda pendiente para el futuro.

- Acceder a la cámara web o a la cámara móvil a través del navegador web es todavía una tecnología que está en fase experimental y existen ciertos dispositivos que no pueden incorporar dicha tecnología.
- En el sistema diseñado actualmente, el servidor devuelve la imagen transformada al cliente. Se podría devolver la posición donde se deben posicionar las gafas y ser el cliente el que las superponga directamente en la imagen captada por este dispositivo.
- En este trabajo se ha diseñado un sistema cliente-servidor para repartir el trabajo entre ambas partes (el cliente capta las imágenes y es el servidor el que la procesa). Existe la posibilidad de que todo el proceso lo haga el cliente, sin necesidad de enviar datos al servidor. Esta solución proporcionaría ventajas ya que no existirían latencias en el envío de datos, pero sobrecargaría de cálculos al cliente. Con la mejora de los dispositivos actuales (sobre todo la capacidad de procesamiento de los dispositivos móviles) en un futuro esta solución podría ser una alternativa a considerar.

APÉNDICE I. BIBLIOGRAFÍA

1. P. Viola, M.J. Jones. "Robust Real-Time Face Detection". International Journal of Computer Vision, vol. 57, n. 2, p. 137-154, 2004.
2. Virtual Mirror, <http://www.ray-ban.com/spain/virtual-mirror>. Accedido en Agosto de 2014.
3. PROBADOR VIRTUAL con tu webcam, <http://www.direct-optic.es/probador-virtual-gafas>, Accedido en Agosto de 2014.
4. O. Déniz y col. "Computer vision based eyewear selector". Journal of Zhejiang University, vol. 11, n.2, p. 79-91, 201.
5. PROBADOR mó, <http://www.multipopticas.com/probador-gafas>. Accedido en Agosto de 2014.
6. Kinect for Windows, <http://www.microsoft.com/en-us/kinectforwindows/>. Accedido en Agosto de 2014.
7. TICbeat "El crecimiento de la telefonía móvil – infografía". <http://www.ticbeat.com/sim/crecimiento-telefonía-movil-infografía/>. Accedido en Agosto de 2014.
8. CondisLife "Un formato en crecimiento: dispositivos móviles, smartphones y tablets". <http://condislife.com/2011/12/28/un-formato-en-crecimiento-dispositivos-moviles-smartphones-y-tablets/>. Accedido en Agosto de 2014.
9. A. Scantlin. "How long before smartphone cameras overtake digital cameras?" <http://www.phonedog.com/2013/10/25/how-long-before-smartphone-cameras-overtake-digital-cameras/> Accedido en Agosto de 2014.
10. G. McDowell. "How Does A Digital Camera Work? [Technology Explained]" <http://www.makeuseof.com/tag/technology-explained-how-does-a-digital-camera-work/>. Accedido en Agosto de 2014.
11. HowStuffWorks.com. "What is the difference between CCD and CMOS image sensors in a digital camera?". <http://electronics.howstuffworks.com/cameras-photography/digital/question362.htm>. Accedido en Agosto de 2014.
12. QuinStreet Inc. "What is digital camera?". Accedido en Agosto de 2014.
13. M. Pielot y col. "Evaluation of Camera Phone Based Interaction to Access Information Related to Posters". Mobile HCI 2008: Amsterdam, The Netherlands – MIRW, p. 61-72, 2008.
14. G. Pajares, J. M. Cruz "Visión por Computador. Imágenes digitales y aplicaciones 2ª Edición". Ra-Ma. 2007.
15. H. Zhu, F.H.Y. Chan y F.K. Lam "Image Contrast Enhancement by Constrained Local Histogram Equalization." Computer Vision Image Understanding, vol. 73, n. 2, p.281-290, 1999.
16. A. Pentland, "Looking at People," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 22, n. 1, p. 107-119, 2000.
17. A. Pentland, "Perceptual Intelligence," Comm. ACM, vol. 43, n. 3, p. 35-44, 2000.
18. A. Pentland and T. Choudhury, "Face Recognition for Smart Environments," IEEE Computer, p. 50-55, 2000.
19. I. Craw, D. Tock, and A. Bennett, "Finding Face Features," Proc. Second European Conf. Computer Vision, p. 92-96, 1992.
20. A. Samal y P.A. Iyengar, "Automatic Recognition and Analysis of Human Faces and Facial Expressions: A Survey," Pattern Recognition, vol. 25, n. 1, pp. 65-77, 1992.
21. R. Chellappa, C.L. Wilson, and S. Sirohey, "Human and Machine Recognition of Faces: A Survey," Proc. IEEE, vol. 83, n. 5, p. 705-740, 1995.
22. R. Frischholz "The Face Detection Homepage" <http://www.facedetection.com/facedetection/techniques.htm>. Accedido en Agosto de 2014.
23. J. P. Kapur "Face Detection in Color Images" Pattern Analysis and Machine Intelligence, IEEE Transactions, vol. 24, pp. 696-706, 2002.

24. M. Störring, H. J. Andersen, E. Granum "Skin colour detection under changing lighting conditions". 7th Symposium on Intelligent Robotics System, p. 187-195. 1999.
25. G. Darrell, M. Gordon, M. Harville, J. Woodfill "Integrated person tracking using stereo, color, and pattern detection". Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference, p. 601-608, 1998.
26. K. Fröba. "Audio- and Video-Based Biometric Person Authentication". 5th International Conference, AVBPA 2005, Hilton Rye Town, NY, USA. 2005.
27. O. Jesorsky y col. "Robust Face Detection Using the Hausdorff Distance". Third International Conference on Audio- and Video-based Biometric Person Authentication, Springer, Lecture Notes in Computer Science, LNCS-2091, pp. 90-95, 2001.
28. Haar Cascades. <http://alereimondo.no-ip.org/OpenCV/34>. Accedido en Agosto de 2014.
29. M. Wertheimer. "Experimentelle Studien über das Sehen von Bewegung (Experimental studies on the seeing of motion)". H. Münsterberg: The Photoplay: A Psychological Study. 1912.
30. 100fps.com. http://www.100fps.com/how_many_frames_can_humans_see.htm. Accedido en Agosto de 2014.
31. W3C. "JPEG JFIF". <http://www.w3.org/Graphics/JPEG/>. Accedido en Agosto de 2014.
32. Y. Freund y R.E. Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". Journal of computer and system sciences, vol. 55, p. 119-139. 1997.
33. R.O. Duda, P.E. Hart and D.G. Stork. "Pattern Classification". John Wiley & Sons, New York. 2001.
34. L. I. Kuncheva. "Combining Pattern Classifiers: Methods and Algorithms". John Wiley & Sons, Inc. New Jersey. 2004.
35. C. de Souza. "Haar-feature Object Detection in C#". <http://www.codeproject.com/Articles/441226/Haar-feature-Object-Detection-in-Csharp>. 2012. Accedido en Agosto de 2014.
36. O. Papageorgiou, y Poggio. "A general framework for object detection". International Conference on Computer Vision. 1998.
37. OpenCV API Reference. Object Detection. http://docs.opencv.org/modules/objdetect/doc/cascade_classification.html. Accedido en Agosto de 2014.
38. S. Audet. "JavaCV". <https://github.com/bytedeco/javacv>. Accedido en Agosto de 2014.

APÉNDICE II. GLOSARIO DE TÉRMINOS Y DEFINICIONES

A

AdaBoost (Adaptative Boost): Meta-algoritmo de aprendizaje automático formulado por Yoav Freund y Robert Schapire. Puede ser usado junto con otros algoritmos de aprendizaje automático para mejorar su rendimiento. El proceso de entrenamiento de AdaBoost selecciona sólo las características conocidas para mejorar la potencia predictiva del modelo, reduciendo la dimensión y potencialmente mejorando el tiempo de ejecución ya que las características irrelevantes no necesitan ser calculadas.

B

Boosting: Meta-algoritmo de aprendizaje automático para reducir el margen de error en el aprendizaje supervisado. Se basa en la combinación de clasificadores (llamados clasificadores “débiles”) de manera iterativa. Esta sucesión de clasificadores “débiles” da como resultado final un clasificador “fuerte”.

C

Canal Alfa: Define la opacidad de un píxel en una imagen. El canal alfa actúa como una máscara de transparencia que permite, de forma virtual, componer (mezclar capas) imágenes o fondos opacos con imágenes con un cierto grado de transparencia.

CCD (charge-coupled device): Circuito integrado que contiene un número determinado de condensadores enlazados o acoplados. Bajo el control de un circuito interno, cada condensador puede transferir su carga eléctrica a uno o a varios de los condensadores que estén a su lado en el circuito impreso. En la actualidad los CCD son mucho más populares en aplicaciones profesionales y en cámaras digitales.

Clasificador de Haar: Método desarrollado por Viola y Jones [1], siendo una versión adaptada del algoritmo AdaBoost. Es un clasificador basado en árboles de decisión con entrenamiento supervisado.

CMOS (Complementary metal-oxide-semiconductor): Familia lógica empleada en la fabricación de circuitos integrados. En la actualidad, la mayoría de los circuitos integrados que se fabrican utilizan la tecnología CMOS. Esto incluye microprocesadores, memorias, procesadores digitales de señales y muchos otros tipos de circuitos integrados digitales cuyo consumo es considerablemente bajo.

CSV (Comma-Separated Values): Son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por comas y las filas por saltos de línea.

E

Escala de grises: Escala empleada en la imagen digital en la que el valor de cada píxel posee un valor equivalente a una graduación de color en el modelo RGB según la diagonal principal del cubo que representa este modelo [2]. Las imágenes representadas mediante este modelo están compuestas por valores de intensidad iguales en los tres canales espectrales R, G y B.

F

Filtro paso alto: Tipo de filtro electrónico en cuya respuesta en frecuencia se atenúan las componentes de baja frecuencia pero no las de alta frecuencia.

Framework: Estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, que constituye la base de otro proyecto software en lo que respecta a su organización y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros, para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Frames per second: Medida de la frecuencia a la cual un reproductor de imágenes genera distintos fotogramas (frames).

H

Histograma: Función discreta que representa el número de píxeles de la imagen en función de los niveles de intensidad.

HTTP (Hypertext Transfer Protocol): Protocolo usado en cada transacción de la World Wide Web.

I

Imagen integral (Integral image – Summed area table): Estructura de datos y un algoritmo para generar rápida y eficientemente la suma de los valores en un subconjunto rectangular de una cuadrícula.

Internet de las cosas (Internet of things): Concepto que se refiere a la interconexión digital de objetos cotidianos con Internet.

J

JavaCV: Librería para poder utilizar las funciones de OpenCV en Java. Contiene funcionalidades comúnmente utilizadas en la visión por computador.

M

Machine learning (Aprendizaje automático): El objetivo del aprendizaje automático es desarrollar técnicas que permitan a las computadoras *aprender*. Es decir, se trata de crear programas capaces de generalizar comportamientos a partir de una información no estructurada suministrada en forma de ejemplos.

Matriz de convolución (Convolution kernel): La matriz de convolución en gráficos por ordenador se utiliza para aplicar filtros a las imágenes. A través de la aplicación de la convolución de dos matrices bidimensionales de las cuales la primera es la imagen original y la segunda, llamado el núcleo, es el filtro a aplicar. El núcleo puede estar constituido por matrices de dimensiones 3x3, 5x5, 7x7, etc.

O

OpenCV: Biblioteca libre de visión artificial originalmente desarrollada por Intel. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, tales como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estereoscópica y visión robótica.

P

Perceptrón: Unidad básica de inferencia en forma de discriminador lineal, es decir, un algoritmo capaz de generar un criterio para seleccionar un sub-grupo, de un grupo de componentes más grande.

Píxel: Menor unidad homogénea en color que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de vídeo o un gráfico.

R

ROI (Region Of Interest): Subconjunto seleccionado de muestras dentro de un conjunto de datos determinado y más amplio para un propósito particular. En visión por computador ROI define los bordes o regiones (como su nombre indica) del objeto que se está estudiando.

S

Streaming: Distribución de multimedia a través de una red de computadoras de manera que el usuario consume el producto, generalmente archivo de video o audio, en paralelo mientras se descarga. La palabra streaming se refiere a: una corriente continua (que fluye sin interrupción).

T

Tasa de transferencia (bit rate): Número de bits que se transmiten por unidad de tiempo a través de un sistema de transmisión digital o entre dos dispositivos digitales. Así pues, es la velocidad de transferencia de datos.

W

WebSocket: Protocolo que proporciona canales de comunicación en ambas direcciones (cliente y servidor) a través de una sola conexión TCP.

X

XML (eXtensible Markup Language): Lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible.