Revista Iberoamericana de Automática e Informática Industrial

Revista Iberoamericana de Automática e Informática industrial 14 (2017) 307–328

www.elsevier.es/RIAI

Sistema Automático Para la Detección de Distracción y Somnolencia en Conductores por Medio de Características Visuales Robustas

Alberto Fernández Villán[a,*], Rubén Usamentiaga Fernández[b], Rubén Casado Tejedor[b]

[a] *Grupo TSK, Parque Científico y Tecnológico de Gijón, 33203 Gijón, Asturias, España*
[b] *Universidad de Oviedo, Campus de Viesques, 33204 Gijón, Asturias, España*

Resumen

De acuerdo con un reciente estudio publicado por la Organización Mundial de la Salud (OMS), se estima que 1.25 millones de personas mueren como resultado de accidentes de tráfico. De todos ellos, muchos son provocados por lo que se conoce como inatención, cuyos principales factores contribuyentes son tanto la distracción como la somnolencia. En líneas generales, se calcula que la inatención ocasiona entre el 25 % y el 75 % de los accidentes y casi-accidentes. A causa de estas cifras y sus consecuencias se ha convertido en un campo ampliamente estudiado por la comunidad investigadora, donde diferentes estudios y soluciones han sido propuestos, pudiendo destacar los métodos basados en visión por computador como uno de los más prometedores para la detección robusta de estos eventos de inatención. El objetivo del presente artículo es el de proponer, construir y validar una arquitectura especialmente diseñada para operar en entornos vehiculares basada en el análisis de características visuales mediante el empleo de técnicas de visión por computador y aprendizaje automático para la detección tanto de la distracción como de la somnolencia en los conductores. El sistema se ha validado, en primer lugar, con bases de datos de referencia testeando los diferentes módulos que la componen. En concreto, se detecta la presencia o ausencia del conductor con una precisión del 100 %, 90.56 %, 88.96 % por medio de un marcador ubicado en el reposacabezas del conductor, por medio del operador LBP, o por medio del operador CS-LBP, respectivamente. En lo que respecta a la validación mediante la base de datos CEW para la detección del estado de los ojos, se obtiene una precisión de 93.39 % y de 91.84 % utilizando una nueva aproximación basada en LBP (LBP_RO) y otra basada en el operador CS-LBP (CS-LBP_RO). Tras la realización de varios experimentos para ubicar la cámara en el lugar más adecuado, se posicionó la misma en el salpicadero, pudiendo aumentar la precisión en la detección de la región facial de un 86.88 % a un 96.46 %. Las pruebas en entornos reales se realizaron durante varios días recogiendo condiciones lumínicas muy diferentes durante las horas diurnas involucrando a 16 conductores, los cuales realizaron diversas actividades para reproducir síntomas de distracción y somnolencia. Dependiendo del tipo de actividad y su duración, se obtuvieron diferentes resultados. De manera general y considerando de forma conjunta todas las actividades se obtiene una tasa media de detección del 93.11 %.

Palabras Clave:
Detección distracción y somnolencia, Visión por computador, Percepción y reconocimiento, Aprendizaje automático, Monitorización y supervisión

1. Introducción

La conducción es una actividad que requiere un alto grado de concentración por parte de la persona que la realiza, ya que un pequeño descuido es suficiente para sufrir un accidente con las consiguientes pérdidas materiales y/o humanas. De acuerdo al más reciente estudio publicado por la Organización Mundial de la Salud (OMS) en 2013, se estimó que 1.25 millones de personas mueren como resultado de accidentes de tráfico y entre 20 y 50 millones más sufren accidentes sin perder la vida pero pudiendo derivar en dolencias crónicas (Organization (2016)). Todas estas muertes y accidentes no sólo afectan de manera directa a los familiares de las víctimas, sino que, además, tienen un alto coste sobre los presupuestos de los gobiernos, que se estima entre un 3 y un 5 % del producto interior bruto (Peden et al. (2016)).

De todos estos accidentes, muchos son provocados por lo que se conoce como inatención. Este término engloba diferentes estados del conductor, como pueden ser la distracción y la somnolencia, siendo precisamente éstos los que más fatalidades ocasionan. Existen muchas publicaciones e investigaciones que intentan poner cifras que indiquen la cantidad producida por la

* Autor en correspondencia.
Correos electrónicos: alberto.fernandez@grupotsk.com (Alberto Fernández Villán), rusamentiaga@uniovi.es (Rubén Usamentiaga Fernández), rcasado@lsi.uniovi.es (Rubén Casado Tejedor)

http://dx.doi.org/10.1016/j.riai.2017.05.001
Disponible en www.sciencedirect.com

inatención (y sus subtipos), pero no existe una figura exacta sobre los accidentes causados por la inatención puesto que todos estos estudios están realizados en diferentes lugares, diferentes marcos temporales, y por tanto, en diferentes condiciones. En líneas generales, se calcula que la inatención ocasiona entre el 25 % y el 75 % de los accidentes y casi-accidentes (Talbot et al. (2013)).

Uno de los trabajos que mejor trata de definir estos conceptos y su relación es el propuesto por Regan et al. (2011), el cual define una taxonomía para la inatención, con sus diferentes subtipos y que la define como '*insuficiente o no atención a las actividades críticas para una conducción segura*'. En esta taxonomía, a) la somnolencia la engloban en el tipo '*Conductor con Atención Restringida*', definida como '*insuficiente o no atención a las actividades críticas para una conducción segura debido a factores biológicos en los que el conductor no es capaz de procesar la información crítica, como por ejemplo, en eventos de micro-sueños, parpadeos, o procesos de somnolencia*', y b) la distracción la engloban en el tipo '*Conductor con Atención Desviada*', que la definen como '*desviación de la atención de las actividades críticas para una conducción segura*'.

En un estudio realizado en 10 países europeos acerca de la somnolencia y la conducción, la fatiga incrementa el tiempo de reacción en un 86 % y es la cuarta causa de muerte en las carreteras españolas (RACE (2016)). Además, cabe destacar que el 75 % de los conductores españoles han sufrido episodios de somnolencia mientras conducían, muy superior a la media del 47 % que han admitido este hecho. Además, otro factor importante a tener en cuenta es que aunque los accidentes producidos por la somnolencia suelen ser muy graves (vistas las estadísticas anteriores de mortalidad), muchos conductores infravaloran esta situación y conducen aunque noten la presencia de sus síntomas. Bostezos frecuentes, cabeceos, visión borrosa, caída de párpados y esfuerzos por mantener tanto la atención como los ojos abiertos son signos habituales de somnolencia (RACE (2016)). Respecto a la distracción, éste es uno de los factores que más fatalidades ocasiona en España. Por ejemplo, de acuerdo con la Dirección General de Tráfico (DGT), la distracción es la primera infracción detectada en los accidentes con víctimas, con un 13,15 % de los casos (StopChatear (2016)).

Por todo esto, este campo ha sido vastamente explorado por la comunidad investigadora, donde los diferentes estudios y soluciones para luchar contra la inatención se pueden agrupar en tres grandes grupos.

El primero de ellos se corresponde con los métodos basados en el comportamiento vehicular. Estos métodos detectan el estado del conductor analizando constantemente ciertas métricas como pueden ser la posición del coche, los movimientos del volante, la presión del acelerador o del freno, el cambio de marchas (entre otros), y si en alguno se sobrepasa un determinado umbral, es probable que el conductor esté somnoliento o distraído (Liu et al. (2009); Forsman et al. (2013); Sahayadhas et al. (2012)). En líneas generales, el principal inconveniente de estos métodos es que su eficacia depende principalmente de las características individuales del vehículo, conductor y carretera (Sahayadhas et al. (2012); Selvakumar et al. (2015); Jo et al. (2014)). Dentro de los métodos basados en el comportamiento vehicular, empiezan a desarrollarse alternativas que requieren la comunicación entre vehículos para operar correctamente (Slawiñski et al. (2015)).

El segundo de los grupos se basa en el análisis de variables fisiológicas, principalmente para la detección de la somnolencia. Son métodos muy robustos pues permiten la detección de la somnolencia en sus fases tempranas con una baja tasa de falsos positivos (Sahayadhas et al. (2012)). En este grupo destacan los métodos basados en: a) electroencefalograma (EEG), b) electromiograma (EMG), c) electrocardiograma (ECG), d) electrooculograma (EOG). De entre todos estos métodos, el más común para la detección de la somnolencia es EEG, dónde se analizan diferentes bandas de frecuencia (Sahayadhas et al. (2012)). Sin embargo, estos métodos requieren contacto con el conductor para la realización de las medidas, lo que ocasiona que su implementación en entornos reales no sea ni lo más adecuado ni lo más práctico (Dasgupta et al. (2013); Sahayadhas et al. (2012)).

Finalmente, el tercero de los grupos se basa en el análisis de características visuales que presenta un conductor distraído o bajo un estado somnoliento. Un conductor distraído se caracteriza por no mantener la atención puesta en la carretera, por lo que son continuos los movimientos de cabeza hacia ambos lados, sin mantener fija la mirada en la carretera. En cuanto a la somnolencia, las características visuales que la describen son muy variadas, incluyendo movimientos faciales, parpadeos rápidos y constantes, cabeceos y bostezos frecuentes. Hacer constar que, estas características visuales de la somnolencia aparecen en espacios temporales diferentes y normalmente bien definidos (Jo et al. (2014)). De manera específica, los bostezos ocurren generalmente antes de que el conductor entre en somnolencia mientras que, normalmente, los cabeceos ocurren cuando el conductor se duerme. Es por ello que los métodos basados tanto en los bostezos como en los cabeceos no son capaces de detectar con exactitud cuando un conductor está empezando a estar somnoliento. Sin embargo, los métodos basados en obtener información de los ojos pueden detectar con precisión este punto, es decir, son los métodos visuales más adecuados para la detección de la somnolencia (Vural et al. (2007)). Sin embargo, cabe decir que, puesto que existen esas diferencias temporales entre los distintos signos visuales, un punto importante puede ser la combinación de varias de estas características para aumentar la robustez final de la solución (Jo et al. (2014); Sahayadhas et al. (2012)).

Basado en las premisas anteriores, el objetivo del presente artículo es el de proponer, construir y validar una arquitectura basada en el análisis de características visuales mediante el empleo de técnicas de visión por computador y aprendizaje automático para la detección tanto de la distracción como de la somnolencia en los conductores. En concreto, se propone una arquitectura de procesamiento especialmente diseñado para operar en entornos vehiculares, con una carga computacional muy baja y fácilmente integrable en dispositivos con reducidas capacidades de cómputo y capaz de lidiar con distintas condiciones de imágenes muy presentes en este tipo de entornos, como pueden ser las condiciones lumínicas, la resolución de la imagen y la apariencia y pose del rostro del conductor en la

imagen.

En resumen, la principal contribución es el hecho de presentar una solución completamente autónoma para detección de distracción y somnolencia pues son dos de los tipos en la inatención que más accidentes ocasionan. Además, dicha solución presenta las siguientes características: 1) detección automática de la presencia del conductor en el entorno vehicular para dirigir el flujo del algoritmo, 2) detección facial adaptada al entorno vehicular, 3) normalización facial para enfrentarse a características de imagen difíciles, 4) detección rápida y robusta de distracción y 5) detección rápida y robusta de somnolencia.

La organización del resto del trabajo es como se expone a continuación. En la Sección 2 se comentan los principales trabajos relacionados, los cuales sirven como base a la propuesta aquí presentada. A continuación, en la Sección 3 se comenta la metodología empleada, donde se detallan las principales fases del sistema acorde con las contribuciones antes comentadas. En la Sección 4 se comentan los principales puntos en cuanto a la implementación que puedan servir para replicar el trabajo aquí presentado. En la Sección 5 se comentan los resultados obtenidos, tanto con bases de datos de referencia como su validación en entornos reales. Por último, en la Sección 6 se comentan las conclusiones y el trabajo futuro.

2. Estado del arte

Puesto que el presente sistema trata de abordar tanto la distracción como la somnolencia utilizando técnicas de visión por computador, expondremos los principales trabajos en este ámbito, distribuidos en tres grupos acorde a su objetivo (distracción, somnolencia y ambos tipos). Además, extraeremos unas conclusiones de trabajos previos, que servirán de base para proponer nuestra arquitectura.

Para la detección de la distracción existen básicamente dos aproximaciones [1](Fernández et al. (2016)). Por un lado, están las aproximaciones que detectan la distracción por medio de cámaras de alta resolución colocadas por toda la cabina del conductor con el objetivo de poder observar los ojos del conductor independientemente de la posición de la cabeza del conductor. Existen varios inconvenientes en esta aproximación (Hansen and Ji (2010)) como son: a) necesidad de calibración periódica entre las cámaras, b) imposibilidad de captar los ojos con robustez debido a desviaciones en la cabeza del conductor que causan oclusión, o c) requerimiento computacional elevado. Sin embargo, muchos sistemas de ayuda a la conducción no necesitan disponer de la posición exacta de hacia dónde está mirando el conductor con precisión, sino que es suficiente con una aproximación de la misma (Lee et al. (2011)) para, por ejemplo, saber si el conductor está mirando al frente o a alguno de los lados. Estas aproximaciones se basan en que la posición de la cabeza puede ser suficiente para extraer la información de hacia dónde está mirando el conductor (Boyraz et al. (2012)). Esta aproximación funciona bien porque la pose de la cabeza es un indicador de hacia dónde tiene puesto el foco de atención el conductor y, por tanto, hacia donde está mirando (Murphy-Chutorian and Trivedi (2010)). Siguiendo esta aproximación, en Hattori et al. (2006) se propone un sistema que determina la distracción del conductor en caso de que no se detecte una cara frontal. En esta línea, existen otras publicaciones que se comentan a continuación. You et al. (2013) entrenan varios modelos para la detección facial, captando varias categorías: a) cara no encontrada, b) cara frontal, c) cara de perfil hacia la derecha, y d) cara de perfil hacia la izquierda. Otro sistema similar es el implementado en Flores et al. (2010) donde, con el objetivo de detectar la distracción, se genera una alerta si se detecta que el conductor no está mirando de frente.

La mayoría de los trabajos para la detección de la somnolencia se basan en la obtención, de la manera más robusta posible, de lo que se conoce como PERCLOS (Dinges and Grace (1998)) - PERcentaje of eye CLOSure - por considerarse la métrica más aceptada y extendida para la detección de la somnolencia (Dong et al. (2011)). Dicha métrica ha sido validada usando tanto EEG como evaluación subjetiva, por ejemplo, encuestas (Dong et al. (2011)). Con el ojetivo de aumentar esta robustez, Sigari (2009) propone la extracción de signos de fatiga en la parte mitad superior de la región facial del conductor. Es decir, captan signos de somnolencia de la región facial sin realizar un preprocesamiento de dicha región. En nuestra solución se utiliza dicha aproximación para establecer la zona de la región facial que usaremos para extraer a posteriori los signos de somnolencia. Otra opción, es combinar información de otros sensores. Recientemente, López Romero (2016) propone un sistema para la detección de la somnolencia. Utiliza técnicas de visión por computador para extraer del rostro información de somnolencia y se complementa con un sensor de oximetría para la extracción del pulso, que se coloca alrededor del dedo de la mano izquierda, lo que hace que el sistema sea intrusivo.

Algunos trabajos intentan extraer tanto la distracción como la somnolencia. Por ejemplo, en Flores et al. (2011) se detectan ambos factores de inatención tanto de día como de noche. Para ello, realizan el seguimiento de los ojos y el rostro para posteriormente extraer los índices de somnolencia y distracción.

Conclusiones de trabajos previos

El hecho de incorporar varias características permite aumentar la robustez de los sistemas para la detección tanto de distracción como de somnolencia. Para ello, se combinan diferentes técnicas y algoritmos (Jo et al. (2014); Sahayadhas et al. (2012); Noori and Mikaeili (2016)). A modo de ejemplo, se puede destacar la reciente publicación de Noori and Mikaeili (2016), donde abordan la detección de la somnolencia fusionando la información del encefalograma, electrooculograma y señales de la conducción. Sin embargo, los métodos basados en visión por computador son una buena alternativa para monitorizar al conductor de forma no intrusiva sin que interfiera con la conducción. Por ello, el objetivo del presente trabajo es el de proponer una arquitectura basada en diferentes características visuales extraídas mediante procesamiento de imagen y apren-

[1]En general cuando se habla de distracción sin especificar el tipo, los autores se refieren a distracción visual, aunque también existen otros tipos de distracción como son la cognitiva y la manual. Así que siempre que se hable de distracción nos referiremos a distracción visual

dizaje automático para aumentar la robustez del sistema de detección.

Para la detección de la somnolencia el factor clave es la detección del estado de los ojos de manera robusta. Algunas aproximaciones para la detección del estado de los ojos en los conductores se basan en la detección de ciertas características que permitan discriminar el estado del ojo. Estos algoritmos se pueden agrupar en lo que se conoce como '*métodos basados en características*'. Ejemplos de este tipo pueden ser la aproximación del iris por medio de elipses o la obtención del estado de los ojos contabilizando la distribución de ausencia/presencia de iris y del blanco ocular en función de si está abierto o cerrado. Para ello, se suele recurrir a acumular las intensidades de los píxeles tanto en el eje vertical como horizontal de la imagen, que reciben el nombre de proyecciones. Caracterizando las curvas resultantes, se puede obtener el estado del ojo (Zhang and Zhang (2006); Devi and Bajaj (2008); Lu et al. (2011)). Otra aproximación, conocida como '*métodos basados en apariencia*', se presenta como una alternativa más robusta para la detección de ciertas características y atributos faciales, entre los que está el estado de los ojos (Song et al. (2014)), y se basa en obtener la apariencia mediante operadores robustos como LBP, wavelets de Gabor o similares (Song et al. (2014)). En lo que a la detección del estado del ojo se refiere y utilizando los métodos basados en apariencia, antes comentados, es común el uso de aproximaciones de aprendizaje automático en la etapa de clasificación: Redes Neuronales (NN), Máquinas de Soporte Vectorial (SVM), Árboles de Decisión y Adaboost. De todas éstas, las SVM son las que se comportan de manera más robusta (Song et al. (2014)). Es por ello, que para la aproximación actual, se usa SVM como herramienta para el aprendizaje automático en la etapa final de clasificación del estado de los ojos.

Para la detección de la distracción, el factor clave es la detección facial. El algoritmo de Viola & Jones es un algoritmo que permite la detección facial con alta robustez y que puede ser ejecutado en tiempo real. Como muestra de su aplicabilidad, muchos de los sistemas aquí comentados para la detección de la distracción y somnolencia lo usan (Hattori et al. (2006); Sigari (2009); You et al. (2013); Flores et al. (2010, 2011); Hong and Qin (2007); López Romero (2016)). Sin embargo, este método es computacionalmente exigente y no es el más adecuado para su integración en dispositivos con reducidas capacidades de cómputo en entorno vehicular. Es por ello que en la presentación actual se usará, validará e integrará en el sistema, una alternativa más adecuada que el algoritmo de Viola & Jones.

3. Metodología

En esta sección se comentan los puntos principales de la arquitectura desarrollada, cuyo esquema de funcionamiento puede verse en la Figura 1. El funcionamiento del sistema se basa en tres módulos principales: '***detección de pose***', '***detección de ausencia conductor***' y '***detección de estado de los ojos***', que permiten la detección tanto de la distracción como de la somnolencia en entornos vehiculares. En primer lugar, se hará una descripción del flujo de las imágenes y cómo éstas se recogen en los *buffers* CABEZA y OJOS, que recogen los estados de la cabeza y de los ojos, para disparar, en último lugar, las reglas que permiten detectar tanto la distracción como la somnolencia respectivamente.

En concreto, el sistema propuesto va capturando imágenes de la cámara y las va procesando secuencialmente. En primer lugar, se ejecuta el módulo de '***detección de pose***', que a partir de tres detectores faciales (que abreviadamente identificaremos como frontal, izquierdo y derecho) computacionalmente muy ligeros, se consigue estimar la orientación de la cabeza del conductor.

El hecho de que ninguno de los detectores faciales consiga detectar la presencia de la cara del conductor es debido a: 1) el conductor no está presente, o 2) la cabeza del conductor presenta una inclinación o posición excesiva y por tanto refleja una pose que ocasiona que no sea 'detectable' por ninguno de los detectores faciales [2]. En este caso, se ejecuta el módulo '***detección de ausencia del conductor***', que permite discernir si se está en el caso 1) o caso 2) antes comentados.

En caso de que el módulo de detección de pose detecte la presencia del conductor, se comprueba si la pose (o detección) es frontal para determinar el estado de los ojos del conductor mediante el módulo '***detección de estado de ojos***'. Como se ha comentado, sólo se detecta el estado de los ojos si la pose es frontal, pues por pruebas realizadas, no se puede detectar correctamente (se obtiene un elevado número tanto de falsos positivos como de falsos negativos) el estado de los ojos si la pose no es frontal, en cuyo caso la detección facial ha sido 'proporcionada' por el detector de perfil izquierdo o derecho. Sin embargo, este factor no es importante, pues en ese caso, el conductor sería clasificado como distraído por no estar 'de frente' y, por tanto, dicha situación sería tenida en cuenta.

Por tanto, para la detección de la somnolencia, se tiene en cuenta el estado de los ojos del conductor, que se recoge en el buffer ***OJOS***. El estado de los ojos puede presentar tres estados (*abiertos, cerrados, indefinido*), donde el valor indefinido es establecido cuando la pose no es frontal, y por tanto, no se detecta el estado de los ojos. Sin embargo, hemos de notar que dicho caso es tenido en cuenta porque sería catalogado como un caso de distracción al no presentar una pose frontal.

Por otro lado, para la detección de la distracción y en función del flujo de ejecución, se tienen en cuenta cinco posibles estados en el conductor, que se registran en el buffer ***CABEZA***. Estos cinco posibles estados (*frontal, izquierda, derecha, ausencia e inatencion*) se obtienen como se comenta a continuación. Tres de ellos son obtenidos por el módulo 'detección de pose' y se corresponden con las detecciones positivas en los detectores faciales, identificados en la Figura 1 como *CABEZA.resultado*, donde CABEZA.resultado = frontal, izquierda, derecha. Existen dos estados adicionales que son proporcionados por el módulo 'detección de ausencia conductor', que se corresponden con *CABEZA.ausencia* - obtenido en caso de

[2] Por las pruebas realizadas cuando los detectores faciales no son capaces de detectar la cara del conductor (y éste está presente) es debido a que presenta una pose excesiva por estar distraído (por ejemplo buscando algo dentro del vehículo), o por estar demasiado somnoliento y la cabeza presenta una gran desviación - hecho conocido como 'cabezadas'

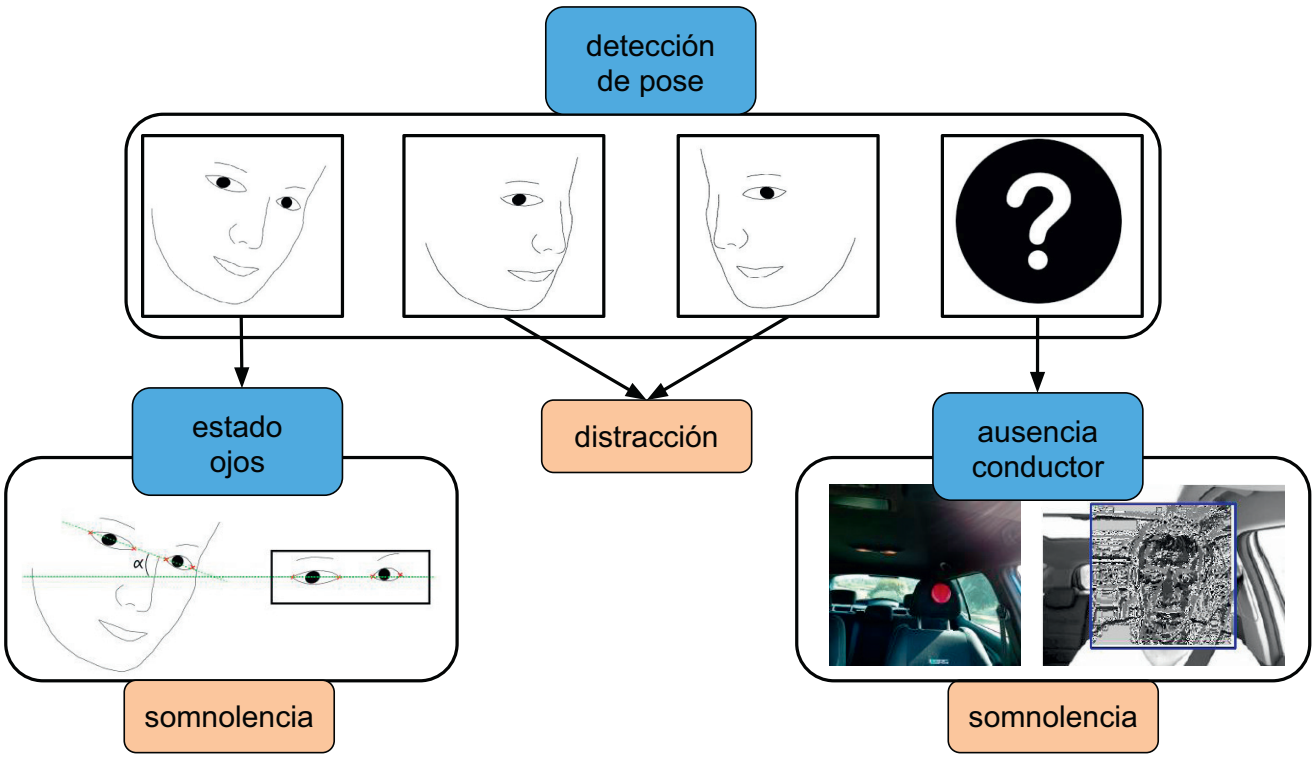


Figura 1: Principales puntos del sistema propuesto

que el conductor no se encuentre en el vehículo - y con *CABEZA.inatencion* - que se corresponde con el caso de que la cabeza del conductor presenta una pose demasiado excesiva y se puede deber tanto a un estado de distracción como de somnolencia (por ello se cataloga como inatención, pues es un factor que engloba a ambos tipos).

A continuación, se proponen y especifican los tres módulos antes comentados. Los detalles de implementación se especifican en la Sección 4, y en la Sección 5, cuando se validan en entorno real, se especifican los tamaños de los buffers (CABEZA y OJOS), así como las reglas que permiten caracterizar el estado del conductor.

3.1. Detección de estado de los ojos

Los principales signos de somnolencia aparecen en los ojos y, por tanto, se comentarán los puntos del algoritmo implementado para el cálculo de PERCLOS por ser el indicador visual más adecuado.

Una vez que la cara ha sido detectada, el siguiente punto radica en aplicar un detector robusto de características faciales basado en 'Modelos de Partes Deformables' (DPM) propuesto por Uřičář et al. (2012). La salida del detector se corresponde con estimaciones de localizaciones para un conjunto de puntos característicos en la imagen: esquinas de los ojos, esquinas de la boca y nariz. A continuación, se aplica un algoritmo para que las caras sean rotadas y alineadas de manera que los ojos siempre se encuentren en las mismas coordenadas en la imagen final. Para ello, con el objetivo de calcular el ángulo de desviación de la cara, se calcula una recta de regresión que utiliza los cuatro puntos de los ojos. Por último, se calcula la región facial por encima y por debajo de los ojos, para que únicamente información relevante se procese en las etapas siguientes del algoritmo. Para más detalles de este algoritmo, se puede consultar el Apéndice B. Pre-procesamiento facial para obtener el estado de los ojos, donde se describe dicha etapa de pre-procesamiento con mayor profundidad.

Una vez la región facial a procesar ha sido normalizada, se aplica un *framework* especialmente diseñado e implementado para la normalización de una imagen ante condiciones lumínicas adversas (Tan and Triggs (2010)), situaciones muy comunes en entornos vehiculares. Se trata de un *framework* computacionalmente muy ligero, pero que permite mejorar considerablemente los algoritmos posteriores que se apliquen sobre la región facial pues dicha región se encuentra normalizada en cuanto a la iluminación. Son tres las operaciones que se realizan en este *framework*: 1) corrección gamma, 2) diferencias gausianas, y c) ecualización de contraste.

A continuación, se aplican tanto el operador LBP (Ojala et al. (1996)), como el operador Center-Symmetric Local Binary Pattern (CS-LBP) (Heikkilä et al. (2009)), que es una modificación del operador LBP computacionalmente más ligera y produce prácticamente los mismos resultados. A continuación se hace una introducción a ambos operadores y para más información, se recomienda la lectura del Apéndice C. Operadores LBP y CS-LBP.

El operador Local Binary Pattern (LBP) (Ojala et al. (1996))

es uno de los descriptores de texturas más populares. Esto es debido principalmente a varios factores: a) fácil implementación, b) invariante a cambios monotónicos de iluminación, y c) complejidad computacional baja. Dicho operador fue introducido en 1996 como un método para sintetizar la estructura del nivel de grises en imágenes. A pesar de que originalmente fue propuesto para el análisis de texturas, el método LBP se ha propuesto para muy diversas tareas en lo que a la visión por computador y el aprendizaje automático se refiere. Relacionado con el procesamiento facial, ha sido empleado para muy diversas tareas de reconocimiento (Losada et al. (2013)), como por ejemplo, reconocimiento facial (Ahonen et al. (2006); Villan et al. (2016)), extracción de variables fisiológicas (Fernández et al. (2015a); Fernandez et al. (2017)) clasificación de género (Shan (2012)), clasificación de expresiones faciales (Shan et al. (2009)), clasificación de la edad (Hadid and Pietikäinen (2013)) e, incluso, detección de gafas (Fernández et al. (2015b)). Dicho operador tiene en cuenta un vecindario local de píxeles alrededor de un píxel central (Ojala et al. (1996)). Seguidamente, umbraliza los píxeles del vecindario con el valor del píxel central y usa el resultado como un número en binario como descriptor para ese vecindario y así sucesivamente para toda la imagen. Fue originalmente propuesto para un vecindario de 8, con 8 bits para codificar los valores binarios. El operador fue posteriormente extendido para incorporar vecindarios de píxeles de diferentes tamaños, haciendo por tanto posible lidiar con las texturas a diferentes escalas (Ojala et al. (2002)).

Para una representación facial eficiente, las características extraídas por el operador LBP deben considerar información espacial. Para ello, la imagen se divide en m regiones $\{R_0, ..., R_{m-1}\}$ y para cada una de esas regiones, se construye el histograma a partir de la imagen LBP generada tras aplicar el operador. De esta manera, el histograma básico se puede extender recibiendo el nombre de 'histograma espacial extendido' (Ahonen et al. (2006)), codificando tanto la apariencia como las relaciones espaciales de las regiones de la cara. El histograma tendrá una longitud de $B = 256 \times m$, siendo 256, el número de patrones diferentes que se pueden producir para el operador LBP con 8 vecinos y m, el número de regiones. Existe una variante del operador LBP, que se conoce como uniforme (Ojala et al. (2002)), y que reduce el número de patrones de 256 a 59, con lo que el histograma generado se reduce en dimensionalidad.

Si bien el operador es computacionalmente ligero y fácil de implementar, puede producir longitudes de histograma bastante grandes si el número de divisiones es considerable, lo cual, como hemos comprobado en una publicación anterior, tiene influencia directa en el grado de acierto (Fernández et al. (2015c)). Esto puede tener implicaciones de rendimiento en dispositivos con reducidas capacidades de computo al tener que tratar con estos histogramas. Se propone usar en vez del operador LBP, una modificación eficiente de este operador para obtener histogramas mucho más reducidos, pero sin perder poder discriminatorio, es decir, manteniendo el rendimiento. Para ello, se hace uso del operador CS-LBP (Heikkilä et al. (2009)), una modificación del operador LBP, que es: 1) computacionalmente más ligero que el operador LBP, 2) más fácil de implementar que el operador LBP, y 3) produce vectores de características más compactos. Para computar el operador CS-LBP se comparan pares de píxeles opuestos en vez de comparar todos los píxeles con el píxel central. El operador CS-LBP también se emplea para calcular el histograma espacial extendido. En la Figura 2 se puede ver la diferencia en la forma de computar ambos operadores. Como se puede ver en esta Figura, el operador CS-LBP produce 16 patrones diferentes. Si se compara con los 256 producidos por el operador LBP (ó los 59 producidos por la extensión uniforme), se trata de una reducción considerable. Además, añadiendo el umbral T se aumenta la robustez en regiones con poco contraste (Heikkilä et al. (2009)).

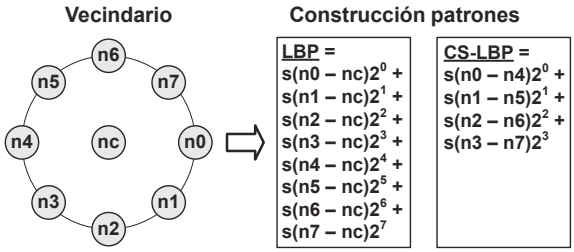


Figura 2: Comparación en la construcción del patrón LBP y CS-LBP para un vecindario de 8

Una vez construido el histograma (mediante alguno de los operadores antes comentados), el último paso consiste en aplicar la etapa de clasificación. Para ello, se hizo uso de las Máquinas de Soporte Vectorial (Support Vector Machines, SVMs). Por tanto, y a modo de resumen, se calcula el histograma espacial extendido (Ahonen et al. (2006)) de la región de los ojos (Fernández et al. (2015c)) normalizada respecto a la iluminación (Tan and Triggs (2010)) aplicando tanto los operadores LBP (Ojala et al. (1996)) como CS-LBP (Heikkilä et al. (2009)). Para clasificar dichos histogramas, se hace uso de la librería LIBSVM (A Library for Support Vector Machines) (Chang and Lin (2011)).

Se realizaron pruebas preliminares para establecer unos valores iniciales de configuración en los operadores LBP y CS-LBP. Tras analizar dichas pruebas, se llega a la conclusión de que, en el caso de LBP, los patrones uniformes $LBP^{u2}_{P,R}$ arrojan mejores resultados que otros tipos de patrones LBP. En cuanto al vecindario, $P = 8$ y un radio de $R = 2$ es la configuración que mejores resultados arroja tanto para LBP como para CS-LBP.

3.2. Módulo detección de pose

Existen muchas aproximaciones para la detección facial en general y otras, para entorno vehicular en particular. El algoritmo más común y extendido es el algoritmo de Viola & Jones (Viola and Jones (2004)). Esto es debido, entre otras cosas, a que es un algoritmo bastante robusto y también porque está listo para usarse en librerías de visión por computador, como OpenCV. Si bien es un algoritmo que se puede ejecutar en tiempo real en un PC, su rendimiento en dispositivos con reducidas capacidades de cómputo se decrementa considerablemente. En este trabajo se propone usar una variante del algoritmo de Viola & Jones, especialmente pensada para ejecutarse en este tipo de dispositivos y además, con una tasa de falsos positivos

menor. Dicho algoritmo recibe el nombre de PICO (Pixel Intensity Comparison-based Object detection Markuš et al. (2014)) y proporciona resultados comparables a algoritmos de vanguardia (Li et al. (2015)), pero con un coste computacional muy bajo. Para poner en perspectiva a este algoritmo, se adjunta la tabla 4, donde se puede observar el coste computacional comparado con las implementaciones que ofrece la librería de OpenCV para el algoritmo de Viola & Jones. Para una descripción más exhaustiva de PICO y su comparación con el algoritmo de Viola & Jones, se recomienda la lectura del Apéndice D. Comparación entre los algoritmos de Viola & Jones y PICO.

Siguiendo la aproximación propuesta por Asthana et al. (2011), en el sistema aquí propuesto se hace uso de tres clasificadores. El detector frontal detecta caras con un ángulo de *yaw* entre -40 grados y 40 grados, otro detector de perfil izquierdo que detecta caras con un ángulo de *yaw* entre 30 y 60 y un detector de perfil derecho que detecta caras con un ángulo de *yaw* entre -30 y -60 grados. Además, para estos detectores se estima que se detectan caras entre -30 y 30 grados en el *pitch*. De manera resumida, mediante estos tres detectores se abarca prácticamente todo el ángulo *yaw*, cuyo procesamiento nos daría la distracción. Sin embargo, estos detectores tienen una limitación, que para la detección de la inatención (somnolencia y distracción) puede ser una virtud que conviene explotar. Estos detectores no detectan caras si tienen un pitch excesivo (es lo que se identifica como CABEZA.inatención). Una cara presenta un ángulo *pitch* excesivo sobre todo producto de las típicas cabezadas cuando el conductor está demasiado somnoliento o completamente distraído. Estos ángulos se pueden ver de manera esquemática en la Figura 3.

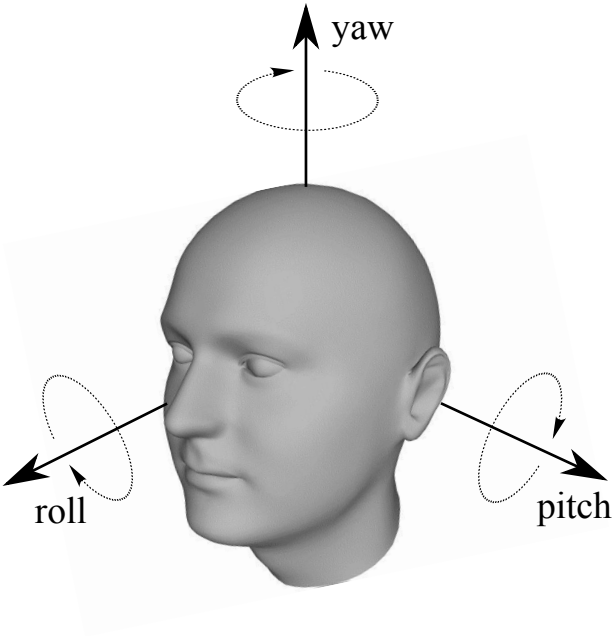


Figura 3: La pose se puede descomponer en los ángulos conocidos como roll, pitch y yaw. El ángulo yaw es especialmente útil para la detección de la distracción y el ángulo pitch para la somnolencia

3.3. Detección de ausencia del conductor

Este módulo detecta si el conductor se encuentra presente en el habitáculo. Como punto de partida, comentar que este módulo se ejecuta tras el módulo de obtención de pose en entorno vehicular. Mediante el presente módulo y el anterior, se permite caracterizar la cabeza del usuario en los cinco estados comentados en la introducción de esta sección: *frontal, izquierda, derecha, ausencia e inatención*). Para la detección de ausencia del conductor, se han implementado dos aproximaciones que se comentan a continuación: a) detección de ausencia de conductor mediante marcador visual en el asiento del conductor, y b) detección de ausencia mediante operador CS-LBP y aprendizaje supervisado.

3.3.1. Detección de ausencia de conductor mediante marcador visual en el asiento del conductor

El primer paso consiste en transformar la imagen del espacio de color RGB al HSV, donde la crominancia y la iluminación están separados, con lo que establecer umbrales para el filtrado posterior es más fácil y más robusto. En segundo lugar, establecemos dichos umbrales para la detección de zonas rojas en la imagen. Una vez filtrada la imagen, se calculan los contornos que envuelven a dichas zonas. Por último, se filtran por tamaño y forma los contornos detectados para obtener el marcador circular. De manera más específica en la etapa de filtrado, se calcula como primer paso el cuadrado mínimo que envuelve al potencial círculo y se considera que el potencial contorno se trata del marcador, si cumple las cuatro condiciones siguientes: a) el contorno tiene más de 6 vértices, b) el cociente entre el ancho y el alto del cuadrado envolvente es aproximadamente de 1, c) el área del contorno es aproximadamente πr^2 (siendo el radio - r - la mitad del ancho de la envolvente), y d) el área del contorno tienen un determinado tamaño en píxeles. Las pruebas del algoritmo se pueden ver en la Sección Pruebas. El hecho de utilizar un marcador, que se ubicaría en el reposacabezas del asiento, permitiría no sólo saber si el conductor está presente o no en su lugar de conducción, sino también realizar aspectos de calibración y establecer heurísticos de manera fácil. A modo de ejemplo, nos permitiría establecer una zona donde buscar la cara en la imagen, para limitar la búsqueda de la región facial.

3.3.2. Detección de ausencia mediante operadores LPB/CS-LBP y aprendizaje supervisado

Comentar que esta aproximación surge a raíz de la implementación de los operadores LBP y CS-LBP para la obtención del estado de los ojos. En concreto, el operador CS-LBP fue propuesto originalmente para describir regiones de interés en imágenes, donde demostró ser tolerante a cambios de iluminación, ruido en la imagen y pequeños cambios de perspectiva, factores bastante frecuentes en el interior de los vehículos. Es por ello que se optó por detectar la ausencia (o presencia) del conductor mediante la aproximación que se comenta a continuación. En este caso, se entrena un clasificador para discernir si el conductor está presente en el vehículo o por el contrario el conductor no está en el mismo. Para ello, se recolectaron imágenes para entrenar el sistema. El procedimiento es muy similar al

Tabla 1: Tiempo medio requerido para procesar una imagen con resolución de 640x480 píxeles

Dispositivo	CPU	Tiempo [ms]		
		PICO	Viola &Jones (OpenCV)	LBP (OpenCV)
PC1	3.4GHz Core i7-2600	2.4	16.9	4.9
PC2	2.53GHz Core 2 Duo P8700	2.8	25.4	6.3
iPhone 5	1.3GHz Apple A6	6.3	175.3	47.3
iPad 2	1GHz ARM Cortex-A9	12.1	347.6	103.5
iPhone 4S	800MHz ARM Cortex-A9	14.7	430.3	129.2

usado para el cálculo de los ojos, lo que cambia es: 1) las imágenes utilizadas para entrenar y validar el sistema, y 2) la región de la imagen utilizada. Por lo tanto, se calcula el histograma espacial extendido de la región de de interés normalizada respecto a la iluminación aplicando tanto los operadores LBP como CS-LBP. Respecto a las imágenes, se recopilaron 1000 imágenes negativas donde el conductor no está presente y se recopilaron 1000 imágenes positivas donde el conductor está presente. Con el objetivo de reducir el tamaño de la imagen a procesar y de que el clasificador aprenda mejor las características que le permitan discernir las dos clases, se restringe el espacio de búsqueda (ROI), pero teniendo en cuenta un cierto margen para facilitar el 'set up' inicial de la cámara. Como herramienta de clasificación se utilizó SVM. Los resultados del algoritmo y las imágenes utilizadas para entrenar y validar el sistema se pueden ver en la Sección 5.

4. Implementación

El sistema aquí propuesto ha sido desarrollado en C++ y haciendo uso de la librería de OpenCV - Open Source Computer Vision Library, que es una biblioteca de visión por computador multiplataforma, publicada bajo la licencia BSD, que permite ser usada tanto para uso académico como comercial. Incluye más de 500 algoritmos. La última versión estable es la 2.4.13, que es la que ha sido usada en la implementación del sistema. Como herramienta para entrenar los modelos basados en SVM, se hizo uso de la librería LibSVM Chang and Lin (2011). El sistema ha sido construido de manera modular. Cada uno de los módulos que componen el sistema son los descritos anteriormente. Para la detección facial, se hizo uso del framework PICO. Para la detección de los principales puntos faciales, se hizo uso de la librería Flandmarks. Para la normalización de la iluminación de la región facial, se hizo uso de la implementación del algoritmo Tan and Triggs (2010), que requiere únicamente tres llamadas a tres funciones de la librería OpenCV, que se corresponden con los tres pasos propuestos por el framework para la normalización de la iluminación. En relación al operador LBP como del operador, se hizo uso de un algoritmo implementado en C++ y OpenCV. Para el alineamiento de la región facial, y posteriormente quedarnos con la región del entorno de los ojos, se hizo uso también de un algoritmo implementado en C++ y OpenCV.

En un primer momento, el sistema se implementó bajo el sistema operativo Windows, haciendo uso de la librería OpenCV antes comentada y enteramente en C++. De esta manera, se realizaron los primeros tests y pruebas con bases de datos de referencia y con imágenes capturadas desde entornos vehiculares. Dichos tests se corresponden con la subsección 'Validación de los módulos que componen el sistema con diferentes bases de datos'.

Dada la modularidad del sistema, el siguiente paso fue portar los algoritmos de visión por computador a la plataforma Android usando el framework JNI. Es un framework que permite que partes de la aplicación en Android se comuniquen con los algoritmos de visión artificial cuya implementación seguiría estando en C++. De esta manera, el sistema no perdería excesivamente en rendimiento. El objetivo de portar los algoritmos a la plataforma Android fue el de construir una aplicación para los dispositivos móviles para validarla y ejecutarla en un entorno real. Dichos tests se corresponden con la subsección 'Validación en entorno real controlado'.

5. Resultados

Para evaluar el sistema actual con rigurosidad, pero sin poner en riesgo la integridad de cualquier persona involucrada en los experimentos o terceras personas ajenas a los mismos, se llevaron a cabo dos procedimientos. El primero de ellos fue testear con bases de datos los diferentes módulos que componen el sistema y con el segundo, se realizaron diferentes pruebas del sistema completo en un entorno real controlado. Cabe destacar que las pruebas realizadas fueron en condiciones diurnas.

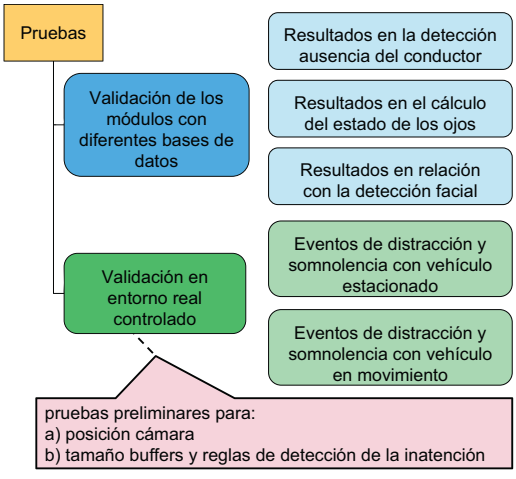
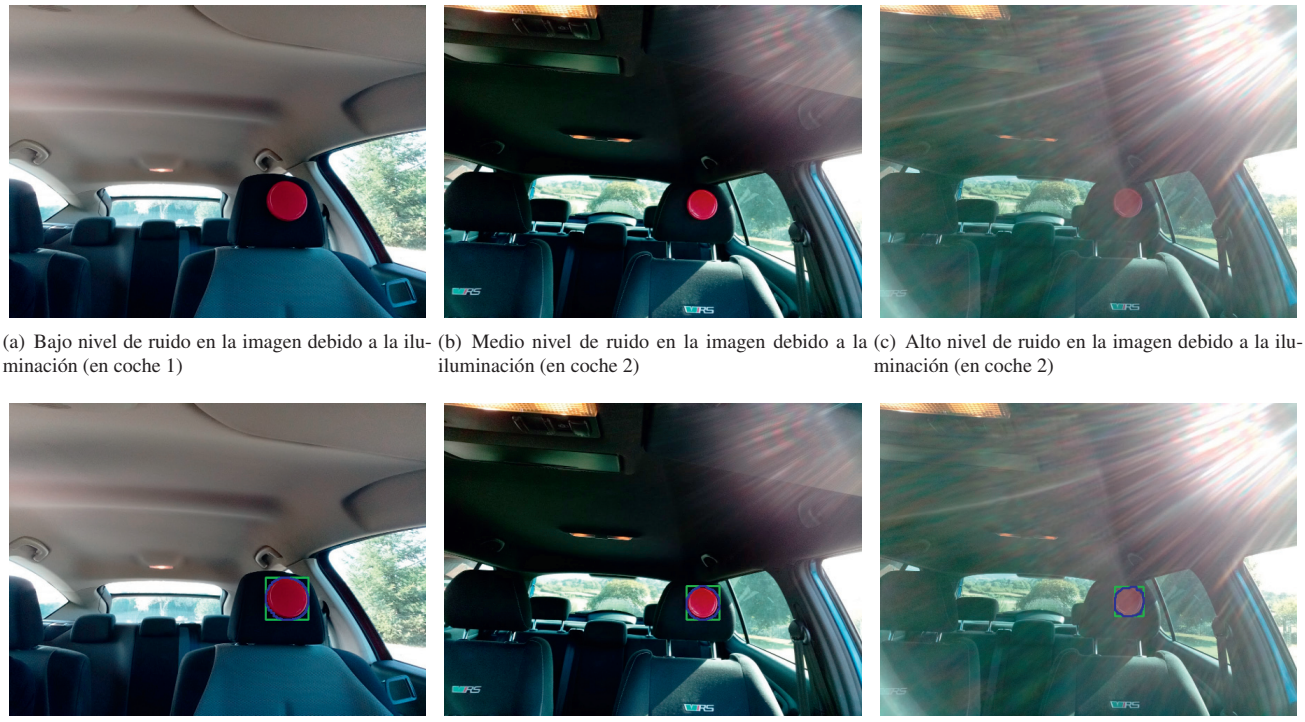


Figura 4: Organización y estructura de las pruebas realizadas



(a) Bajo nivel de ruido en la imagen debido a la iluminación (en coche 1)

(b) Medio nivel de ruido en la imagen debido a la iluminación (en coche 2)

(c) Alto nivel de ruido en la imagen debido a la iluminación (en coche 2)

(d) Resultado para un bajo nivel de ruido debido a la iluminación

(e) Resultado para un nivel de ruido medio debido a la iluminación

(f) Resultado para un nivel de ruido alto debido a la iluminación

Figura 5: Resultados en la detección ausencia del conductor

Todas estas pruebas se resumen en la Figura 4. A continuación, se comentan los dos procedimientos antes introducidos.

5.1. *Validación de los módulos que componen el sistema con diferentes bases de datos*

Para la validación tanto de los resultados de ausencia del conductor por medio de aprendizaje automático y la detección del estado de los ojos se hace uso de los operadores LBP, CS-LBP en la etapa de extracción de características y SVM en la etapa de clasificación. En los Apéndices C. Operador LBP y E. Entrenamiento con SVM se pueden ver de manera más detallada aspectos más técnicos y específicos con el objetivo de reproducir los resultados aquí obtenidos.

Resultados en la detección ausencia del conductor. En la Figura 5 se puede ver de manera gráfica cómo funciona el algoritmo para la detección de ausencia del conductor por medio de un marcador. Pruebas preliminares nos demostraron que el factor más influyente en este algoritmo eran las diferentes condiciones lumínicas en el interior vehicular, más que el propio interior propiamente dicho. Las diferentes condiciones lumínicas producen ruido en la imagen, que en la etapa de clasificación por forma circular es capaz de rechazar, con lo que se puede discriminar el contorno circular correspondiente al marcador. Es por ello que sólo se realizaron pruebas en dos vehículos, pero con entornos lumínicos muy diferentes. Se realizaron pruebas con $N = 200$ imágenes y en todas ellas se detectó el marcador correctamente. Podemos, por tanto, decir que este simple algoritmo es capaz de detectar con robustez la presencia o ausencia de un conductor por medio de un marcador, alcanzando una tasa de precisión del 100 %. El marcador mostrado en la Figura 5 presenta un radio $r = 4cm$, el cual es suficiente para su detección en condiciones diurnas. Un tamaño más pequeño (por ejemplo $r = 2cm$) podría presentar problemas, sobre todo cuando existe mucho ruido debido a la iluminación.

Por otro lado, a continuación se comentan las pruebas realizadas para la detección del conductor por medio de procesamiento de imagen y de aprendizaje supervisado. Para entrenar el algoritmo se recolectaron imágenes positivas (el conductor está presente) e imágenes negativas (el conductor no está presente y, por tanto, se ve únicamente el interior del vehículo). Para la recolección de imágenes positivas, se tomaron 1000 imágenes de las siguientes bases de datos: a) YawDD (Abtahi et al. (2014)), b) RS-DMV (Nuevo et al. (2010)), c) Set 11 (Daniluk et al. (2014)), y d) imágenes recogidas para la elaboración de la presente publicación y las diversas pruebas realizadas. Como imágenes negativas, se capturaron un total de 1000 imágenes desde el interior de cinco vehículos diferentes (200 imágenes de cada uno) para obtener cierta variabilidad, y que el clasificador no clasificara características irrelevantes. El *dataset*, por tanto, consta de 2000 imágenes en total.

Como herramienta de clasificación se hizo uso de SVM. Se realizaron pruebas variando parámetros en los diferentes operadores (LBP y CS-LBP), obteniendo resultados similares en ambos casos. De manera más específica, la tasa de acierto media del clasificador basado en el operador LBP fue de 90.56 %,

Tabla 2: Tasa detección para la detección de la ausencia del conductor utilizando tanto un marcador, como los operadores LBP y CS-LBP

Marcador	LBP	CS-LBP
100 %	90.56 %	88.96 %

y del 88.96 % utilizando el operador CS-LBP. En la Figura 6 se puede ver la imagen preprocesada que sirve de entrada al clasificador SVM, donde se puede ver el ROI establecido y el resultado de aplicar tanto el operador LBP como el CS-LBP. El ROI seleccionado nos permite, por un lado, que la variabilidad en la imagen sea ofrecida por la presencia/ausencia del conductor y no por otros elementos como pueden ser cambios en el exterior del vehículo debido al movimiento o la presencia de pasajeros. Sin embargo, con objeto de que no requiera de calibración, no hemos establecido una ROI bastante precisa, para que se pudiera adaptar a diferentes 'set up' de cámara.

Los resultados para la detección del conductor se pueden ver de manera resumida en la Tabla 2. No existe comparación posible, pues no se han encontrado publicaciones que lleven a cabo dicha detección.

Resultados en el cálculo del estado de los ojos. Para el cálculo del estado de los ojos, se hizo uso de la base de datos CEW (Closed Eyes in The Wild) Song et al. (2014), recientemente propuesta para este propósito. Esta base de datos constituye una buena base de test, pues contiene imágenes en condiciones no controladas, con iluminaciones muy variadas, baja resolución, diferentes poses o gafas, entre otras. A continuación, se comentan los resultados obtenidos puestos en contexto con otras aproximaciones, los cuales pueden verse de manera resumida en la Tabla 3. En la Figura 7 se puede ver un ejemplo

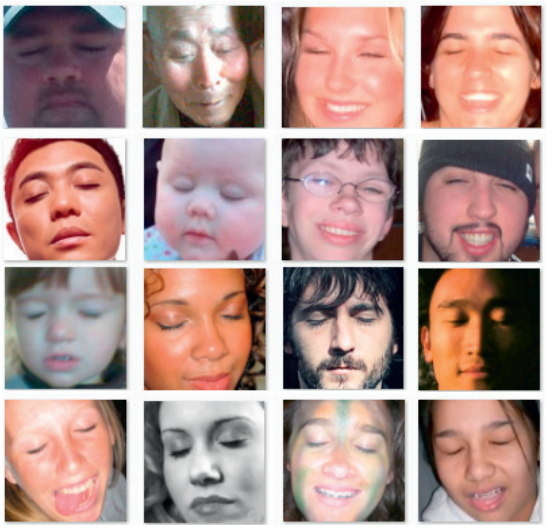


Figura 7: Ilustración de las imagénes de la base de datos CEW

visual de la variabilidad incluida en dicha base de datos.

Los métodos basados en proyecciones, obtienen una tasa de reconocimiento del 70.1 %, muy lejos de la aproximación más robusta que propone el uso de una modificación del algoritmo basado en la computación de Histogramas Orientados de Gradientes (HOG), originalmente propuesto para la detección de personas Dalal and Triggs (2005), pero que ha demostrado ser robusto para la detección de objetos en general. Dicha modificación del algoritmo HOG, que los autores llaman '*MultiHPHOG*', obtiene una tasa de acierto del 93.51 % Song et al. (2014). Nuestra aproximación, basada tanto en LBP como en CS-LBP, obtiene unas tasas de acierto del 93.39 % y del 91.84 %, respectivamente. Además, la arquitectura aquí propuesta tiene dos ventajas principales frente a otras aproximaciones: 1) la tasa de reconocimiento es de las más altas sin llevar a cabo una localización exhaustiva del ojo, es decir, es suficiente con detectar de forma más o menos precisa la región de los ojos, lo que aumenta su robustez frente a otros métodos que requieren localizar el ojo, y 2) es una aproximación computacionalmente muy ligera. Por poner en contexto la aproximación más robusta, basada en *MultiHPHOG*, con tasa de reconocimiento del 93.51 % (Song et al. (2014)), tiene un tiempo de ejecución aproximado de unos 40ms. Nuestra aproximación, basada tanto en LBP (93.39 %) como en CS-LBP (91.84 %), tiene un tiempo de ejecución aproximado de unos 2ms y 1ms aproximadamente. Aquí es donde entendemos que es la principal diferencia, pues los algoritmos son unas 20 veces más rápidos, con un ligero decremento de la efectividad, pero aún así, mucho más robustos que la mayoría de algoritmos. En la Figura 8 se puede ver el preprocesamiento y el procesamiento de la imagen para la aproximación que se usa para la detección de los ojos, donde se muestran todos los pasos para llevar a cabo dicha detección.

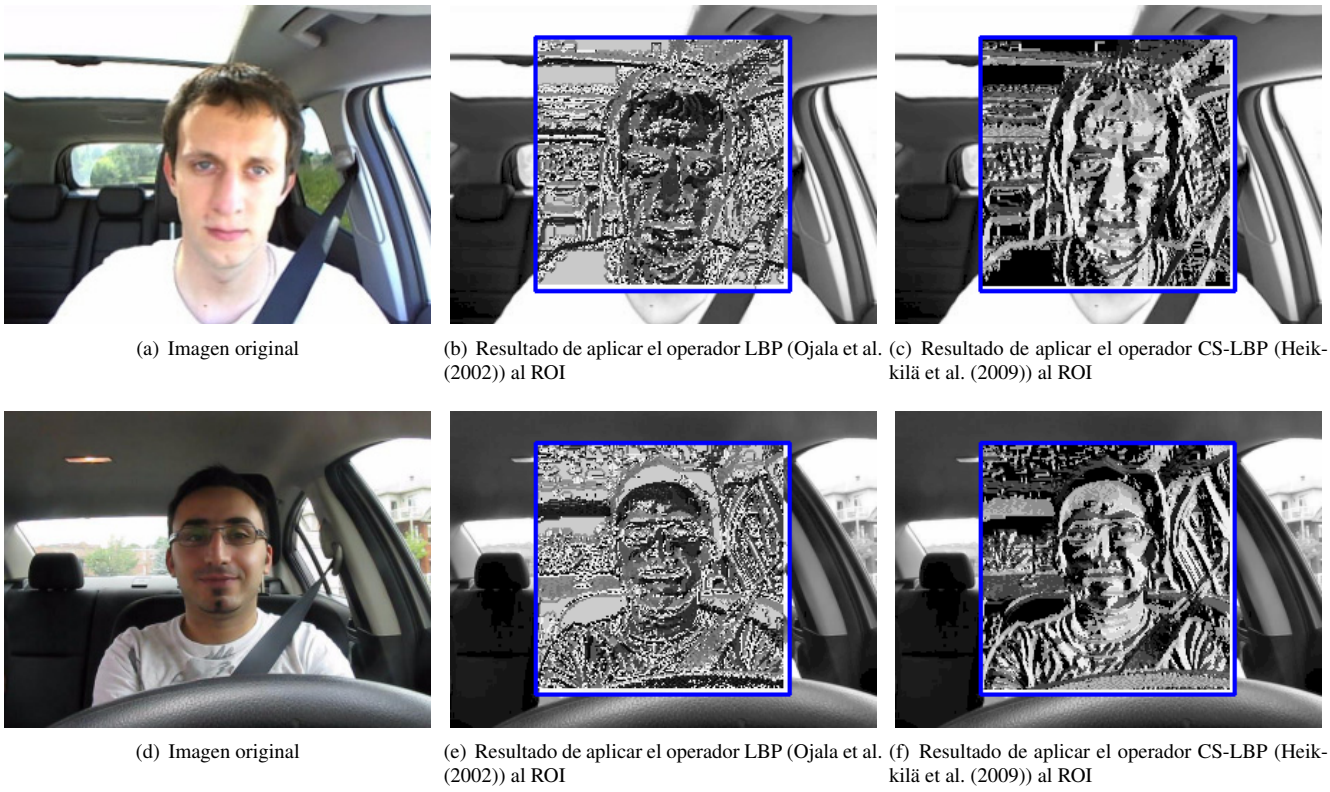
Tabla 3: Comparativa de rendimiento para la detección del estado de los ojos en la base de datos CEW y algoritmos *state-of-the-art*

Aproximación	Precisión (%)	Tiempo de ejecución [ms]
Proyección	70.1	-
Escala de grises + SVM	82.85	0.32
LBP + SVM	91.12	1.96
LTP + SVM	92.94	6.67
Gabor + SVM	91.16	17.65
HOG + SVM	93.10	12.61
MultiHOG + SVM	93.31	19.81
HPOG + SVM	93.13	18.55
MultiHPOG + SVM	**93.51**	**38.47**
LBP_RO + SVM	**93.39**	**2**
CS-LBP_RO + SVM	**91.84**	**1**

Resultados en relación con la detección facial. Pruebas realizadas arrojaron que la posición de la cámara juega un papel fundamental en la detección facial y, por tanto, en la detección posterior tanto de la distracción como la somnolencia. Es por ello que se realizaron pruebas para determinar la posición óptima de la cámara sin ocasionar oclusión o distracción al conductor por su posible ubicación. Las pruebas realizadas en este punto se incluyen en la subsección siguiente, pues han sido llevadas a cabo en entorno real.

5.2. Validación en entorno real controlado

La validación en entorno real controlado incluye dos tipos de pruebas: 1) eventos de distracción y somnolencia con el



(a) Imagen original

(b) Resultado de aplicar el operador LBP (Ojala et al. (2002)) al ROI

(c) Resultado de aplicar el operador CS-LBP (Heikkilä et al. (2009)) al ROI

(d) Imagen original

(e) Resultado de aplicar el operador LBP (Ojala et al. (2002)) al ROI

(f) Resultado de aplicar el operador CS-LBP (Heikkilä et al. (2009)) al ROI

Figura 6: Descripción gráfica de la detección ausencia del conductor mediante procesamiento de la imagen y aprendizaje supervisado

vehículo estacionado y 2) eventos de distracción y somnolencia con vehículo en movimiento en un recorrido preestablecido. En este sentido, tanto: a) posición de la cámara y b) tamaño de los buffers CABEZA y OJOS (introducidos en la Sección 3) juegan un papel fundamental y serán detallados antes de comentar los resultados de estas pruebas.

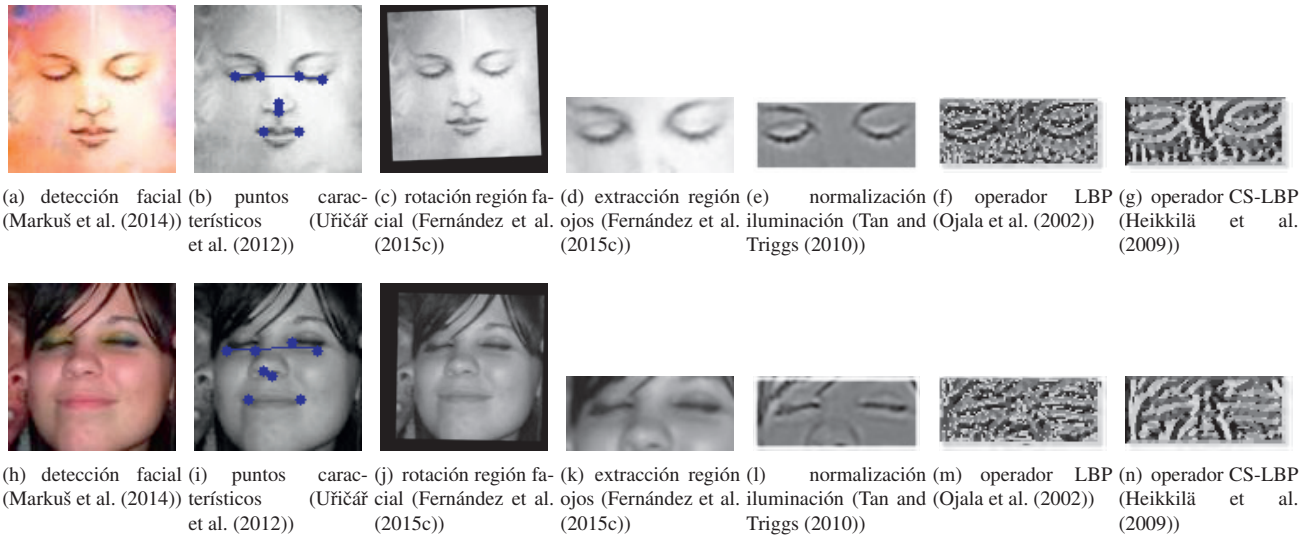
Para las pruebas se instaló la aplicación en un dispositivo móvil. En concreto, las pruebas fueron realizadas con un móvil de gama media con procesador Qualcomm APQ8064T Snapdragon 600 Quad-Core a 1.9 GHz y 2 GB de RAM. El dispositivo tiene instalada la versión Android 4.2.2. y la aplicación en este dispositivo funciona a unos 10 frames por segundo (fps).

Los vehículos usados fueron: Citroen C4, Skoda Fabia RS y Kia cee'd. Cada vehículo fue conducido por las mismas 16 personas, con una edad media (μ) de 34.5 años desviación estándar (σ) de 6.65. De esos 16 conductores, 8 de ellos ($\mu = 34.25$, $\sigma = 5.91$) eran hombres y los 8 restantes ($\mu = 34.75$, $\sigma = 7.81$) mujeres. A su vez, de los 8 hombres, 4 llevaban gafas en el momento de las pruebas ($\mu = 34$, $\sigma = 8.22$) y 4 no las llevaban ($\mu = 34.5$, $\sigma = 1.5$). En lo que respecta a las 8 mujeres, 4 llevaban gafas ($\mu = 39.5$, $\sigma = 8.29$) y 4 no las llevaban ($\mu = 30$, $\sigma = 2.83$). Ninguno de los conductores llevaba gafas de sol.

Posición de la cámara. Para ambos tipos de pruebas, se tuvo especial cuidado en la ubicación de la cámara. La ubicación de la misma, y en este caso por tanto, del dispositivo móvil, tiene un papel fundamental, pues dependiendo de la imagen adquirida, la cabeza del conductor será encontrada de forma más o menos robusta. Se realizaron pruebas con dos ubicaciones, que se pueden ver en la Figura 9. En la primera, el dispositivo móvil se sitúa en el espejo retrovisor, y en la segunda, sobre el salpicadero, tratando de capturar las imágenes del conductor lo más frontalmente posible. Respecto a este punto, en la Figura 10 se incluye la posición final escogida, por proporcionar una detección más robusta de la cabeza del conductor, como se comenta de forma más detallada a continuación. Dicha posición, por tanto, fue tenida en cuenta para la validación en entorno real controlado y los dos tipos de pruebas antes introducidos.

El hecho de contar con la cámara frente a la cabeza del conductor presenta varias ventajas. La primera de ellas es que el detector frontal presenta una tasa de reconocimiento mayor. En la Tabla 4, se presenta el rendimiento de ambas posiciones en lo que al grado de detección facial se refiere. Para ello, se grabaron varias secuencias cortas de vídeo contemplando distintas condiciones y el conductor manteniendo una conducción normal, comprobando los retrovisores en caso necesario o cambiando de marchas, pero sin mostrar signos de distracción (como, por ejemplo, mirar fuera del vehículo mediante un giro acusado de la cabeza, lo cual no sería ya una detección frontal). Otra de las ventajas es el hecho de detectar diferentes eventos de distracción, pues se corresponden directamente con la salida de los detectores, es decir, en cuanto la cabeza no se capture frontalmente, es que el conductor está distraído.

Tamaño de los buffers y reglas de detección de la inatención (somnolencia y distracción). La información capturada por el sistema se resume en los buffers CABEZA y OJOS de la Figura 1. El objetivo es la detección de eventos de somnolen-



(a) detección facial (Markuš et al. (2014)) (b) puntos característicos (Uřičář et al. (2012)) (c) rotación región facial (Fernández et al. (2015c)) (d) extracción región ojos (Fernández et al. (2015c)) (e) normalización iluminación (Tan and Triggs (2010)) (f) operador LBP (Ojala et al. (2002)) (g) operador CS-LBP (Heikkilä et al. (2009))

(h) detección facial (Markuš et al. (2014)) (i) puntos característicos (Uřičář et al. (2012)) (j) rotación región facial (Fernández et al. (2015c)) (k) extracción región ojos (Fernández et al. (2015c)) (l) normalización iluminación (Tan and Triggs (2010)) (m) operador LBP (Ojala et al. (2002)) (n) operador CS-LBP (Heikkilä et al. (2009))

Figura 8: Representación visual de los principales pasos para la detección del estado de los ojos

Tabla 4: Tasa detección facial para las dos ubicaciones de la cámara propuestas: 1) debajo del espejo retrovisor central y 2) en el salpicadero centrado al conductor

	Posición 1 (debajo espejo)	Posición 2 (salpicadero)
Detector facial frontal(Markuš et al. (2014))	86.88 %	96.46 %

cia y distracción de una manera robusta, pero también teniendo en cuenta una baja tasa de falsos positivos (por ejemplo, que el usuario compruebe los espejos retrovisores y el sistema detecte esta situación como de distracción). Para establecer el tamaño de los buffers, se realizaron: a) diversas pruebas preliminares, b) encuestas a los usuarios en función de resultados intermedios y c) análisis del estado del arte. En este sentido, las tres opciones arrojaron resultados muy similares, que se comentan a continuación.

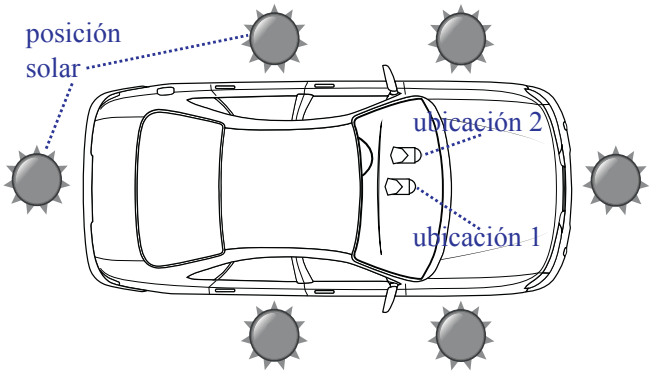


Figura 9: Ubicaciones seleccionadas para la instalación del dispositivo móvil. Además, las pruebas realizadas incluyeron diferentes posiciones solares para conseguir diferentes condiciones lumínicas. Las pruebas realizadas tuvieron lugar tanto en verano como en invierno

Debe ser tenido en cuenta, que del estudio '100-Car Study' cerca del 80 % de los accidentes y del 65 % de los casi-accidentes involucraban algún tipo de inatención por parte del conductor en los últimos 3 segundos antes del evento (Martin (2006)). Además, según otro estudio, se deben detectar eventos de inatención en ventanas de 4 segundos (of Transportation (2016)). Partiendo de estos datos, algunos autores han seguido estas aproximaciones para ajustar las ventanas temporales de sus algoritmos a 3 (You et al. (2013), Berri et al. (2014)) y 4 (Mbouna et al. (2013)) segundos respectivamente.

Tras realizar diversas pruebas y ajustes, se estableció el tamaño temporal de 4 segundos para la detección de eventos de distracción y somnolencia. Esto supone el análisis de unos 40 frames (la aplicación se ejecuta a unos 10fps) y, por tanto, el análisis de unos 40 elementos almacenados tanto para el buffer CABEZA como para el de OJOS. Cada elemento almacenado en los buffers tiene su *timestamp* para disparar los eventos de detección de distracción y somnolencia teniendo en cuenta los 4 segundos antes comentados. Es necesario almacenar el timestamp porque el tiempo de ejecución del sistema completo varía en función del flujo de ejecución (ver Figura 1). Cada vez que se almacena un frame (elemento X) en los buffers, se contabiliza el tiempo respecto a la última inserción en el buffer (elemento X-1) y se anota dicho tiempo al elemento X-1. Estos tiempos nos permiten computar después los estados del conductor para disparar las reglas. Las reglas que disparan las detecciones son bastante sencillas. Se estableció una para la somnolencia (ver regla 1) y otra para la distracción (ver regla 2).

La regla para la detección de la somnolencia se basa en procesar los frames contenidos en el buffer OJOS. En caso de que los frames almacenados en dicho buffer, etiquetados como cerrado (CER), superen los 3 segundos, el conductor se considera dormido. En caso contrario, se considera que está despierto. Para la distracción, se considera que el conductor está distraído si los frames etiquetados como izquierda (IZQ), derecha (DER) e inatención (INA) para el buffer CABEZA, superan los 3 segun-

Regla 1 Deteccion Somnolencia OJOS

1: **if** $tiempos(OJOS, [CER]) \geq 3s$ **then**
2: Conductor dormido
3: **else**
4: Conductor despierto
5: **end if**

Tabla 5: Actividades a realizar por los conductores tanto para las pruebas con el vehículo estacionado como para el vehículo en movimiento sobre recorrido preestablecido

Eventos de distracción y somnolencia a realizar por los conductores
✔A01_X: Cerrar los ojos, con X={4,5,6} seg
✔A02_X: Tocar con la barbilla en el pecho, con X={4,5,6} seg
✔A03_X: Tocar con la nuca en la espalda, con X={4,5,6} seg
✔A04_X: Girar la cabeza hacia la derecha, con X={4,5,6} seg
✔A05_X: Girar la cabeza hacia la izquierda, con X={4,5,6} seg

dos. La función $tiempos(X, [Y])$ devuelve el tiempo en el buffer X para los estados contenidos en el array Y.

Regla 2 Deteccion Distraccion CABEZA

1: **if** $tiempos(CABEZA, [IZQ, DER, INA]) \geq 3s$ **then**
2: Conductor distraido
3: **else**
4: Conductor atento
5: **end if**

En caso de que la aplicación se instale en un dispositivo con mejores capacidades de cómputo, la métrica *fps* tendrá un valor más elevado, y por tanto, se almacenarán más elementos en los buffers para llegar a cubrir los 4 segundos antes comentados. En caso de ser un dispositivo con menos recursos, se almacenarán menos elementos en los buffers. Por pruebas realizadas, hasta los 5 *fps* se obtienen unos resultados similares, pero para *fps* menores, la aplicación no funciona correctamente.

5.2.1. *Eventos de distracción y somnolencia con vehículo estacionado*

En parte, estas pruebas sirvieron para 'ajustar' ciertos aspectos de funcionamiento de la aplicación antes comentados: a) posición de la cámara, b) tamaño de los buffers, y c) ajustes más finos y precisos en la implementación de los algoritmos. Las pruebas fueron realizadas con el vehículo estacionado. La lista de actividades a realizar por los usuarios se puede ver en la Tabla 5. Se trata de 5 eventos con 3 duraciones diferentes. A modo de ejemplo, la actividad A01_4 consiste en cerrar los ojos durante 4 segundos. Los resultados de estas prueba se recogen en la subsección *5.2.3*.

5.2.2. *Eventos de distracción y somnolencia con vehículo en movimiento*

En la Figura 11 se puede ver el recorrido para las pruebas en entorno controlado en movimiento. Para facilitar que los conductores realizaran el mismo recorrido entre las distintas iteraciones, se dispusieron conos para delimitar el recorrido. Las pruebas en movimiento se realizaron a una velocidad de 4km/h durante todas las iteraciones y para todos los conductores. Para ello, se dispuso del Skoda Fabia RS con cambio automático, en el cual, introduciendo la marcha 'D', se inicia el movimiento del vehículo manteniendo dicha velocidad constante siempre y cuando no se altere mediante el acelerador o el freno. El hecho de establecer dicha velocidad no es otro que el de permitir, para el recorrido seleccionado, reproducir por parte del conductor varios episodios de distracción y somnolencia con suficiente tiempo, sin poner en riesgo la integridad del conductor debido a la baja velocidad alcanzada. Además, el recorrido se realizó en ambos sentidos. El hecho de capturar las imágenes a una mayor velocidad no supone ningún cambio en las características de las mismas. Los tramos azules se corresponden con zonas en las que está 'permitido' realizar las actividades de distracción y somnolencia encomendadas (ver Figura 11). Las actividades son las mismas que para el vehículo estacionado, y se corresponden con las de la Tabla 5. Los resultados de estas prueba se recogen en la subsección *5.2.3*.

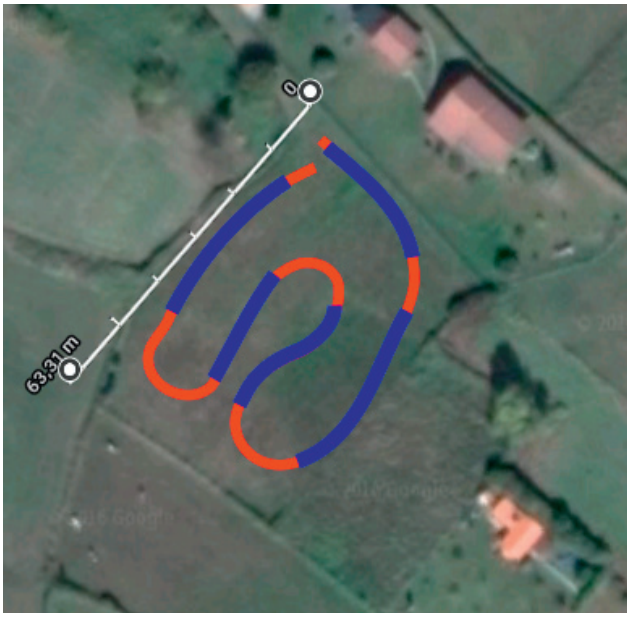


Figura 11: Ruta para las maniobras controladas realizadas en el experimento

5.2.3. *Resultados para entorno real controlado*

En este apartado se resumen los resultados de las pruebas comentadas anteriormente en las subsecciones *5.2.1* y *5.2.2*. Como ya se ha comentado, las pruebas se realizaron con 16 personas, 3 vehículos diferentes y en 2 condiciones diferentes (vehículo parado - *5.2.1* - y en movimiento - *5.2.2*). Las pruebas fueron realizadas a lo largo de un total 22 días, con lo que diferentes condiciones lumínicas fueron tenidas en cuenta al realizar las pruebas. Además las pruebas fueron realizadas en diferentes estaciones, tanto en verano (a lo largo de los meses de Agosto y Septiembre) y en invierno (a lo largo de los meses de Enero y Febrero).

Figura 10: Posición (posición 2 - salpicadero) finalmente seleccionada para la ubicación del dispositivo móvil. De esta manera se consigue una detección facial más robusta

Cada prueba (se denominará sesión para una mayor claridad), consistió en realizar las 5 pruebas descritas en la Tabla 5 con las 3 duraciones diferentes (4,5 y 6 segundos). Es decir, cada sesión involucró 15 actividades diferentes. Se realizaron un total de 96 sesiones (16 personas, 3 vehículos y 2 tipos - vehículo estacionado y en movimiento), con una duración aproximada para realizar todas las actividades de una sesión de 25 minutos. Los resultados del sistema para clasificar cada una de estas 96 sesiones se pueden ver en las Tablas 6, 7, 8, las cuales se comentan en detalle a continuación. De la Tabla 6 se puede extraer la media de las 96 sesiones ($\mu = 93.11$, $\sigma = 1.6$).

Como se puede apreciar, se obtienen resultados similares para los eventos de somnolencia A02 y A03, puesto que la mecánica de funcionamiento es la misma, es decir, en caso de que no se puede obtener la pose del conductor (detección negativa en los algoritmos de detección facial), se comprueba si el conductor está presente o no en el habitáculo. También se obtienen resultados similares para la detección de los eventos de distracción (A04 y A05), puesto que la cámara está posicionada justo en frente del conductor y la pose respecto a la cámara es la misma al realizar los eventos de distracción. Respecto al cálculo del estado de los ojos, se obtienen resultados muy robustos, que pueden llegar a una tasa de acierto del 95.43 %. Sin embargo, es en esta actividad donde se encuentran unas diferencias significativas para los conductores que llevaban gafas en el momento de realizar las pruebas. En la Tabla 7 se pueden ver los resultados correspondientes a los 8 conductores (4 hombres y 4 mujeres) con gafas y su tasa de detección en la actividad A01. En la Tabla 8 se puede ven los resultados para los otro 8 conductores (4 hombres y 4 mujeres) sin gafas.

Para los conductores que no llevan gafas se produce una tasa de detección media de 95.54 %, que puede llegar al 96.88 % si la actividad tiene una duración de 6 segundos. Si la actividad tiene una duración de 4 segundos la tasa de detección es de 94.42 %. Por el contrario, para los conductores que llevan gafas, la tasa de detección media se decrementa, y se obtiene un valor de 93.51 %, que puede llegar al 94.30 % si la duración de la actividad es de 6 segundos. Si la duración de la actividad es de 4 segundos la tasa de detección es de 92.44 %.

No se encontraron diferencias significativas en caso de que el conductor fuera un hombre o una mujer. En este sentido cabe comentar que las mujeres involucradas en el experimento no presentaban maquillaje y el pelo estaba recogido, de manera que no presentaba ninguna posible oclusión en este aspecto. Tampoco se encontraron diferencias significativas tanto si el vehículo estaba estacionado como en funcionamiento.

Tabla 6: Resumen de los resultados (tasa de acierto) para las pruebas en entorno controlado. Se muestra la precisión al clasificar las 96 sesiones como eventos de somnolencia o no (para A01/A02/A03) o de distracción o no (para A04/A05)

	4 seg	**5 seg**	**6 seg**
A01	93.50	94.66	95.43
A02	91.25	92.80	93.61
A03	92.42	92.94	93.27
A04	92.06	92.66	93.37
A05	92.14	93.08	93.39

Tabla 7: Resumen de los resultados (tasa de acierto) para la Actividad del tipo A01_X al clasificar el estado de los ojos para 8 conductores (4 hombres y 4 mujeres) con gafas

	4 seg	**5 seg**	**6 seg**
A01	94.42	95.33	96.88

6. Discusión y conclusiones

La comparación mediante datasets y bases de datos disponibles para la comparación de los diferentes algoritmos, como puede ser la base de datos CEW en este caso, permite una comparación 'justa' entre las diferentes aproximaciones. En este sentido, de los resultados desprendidos tras la realización de la validación de los módulos que componen el sistema con diferentes bases de datos (ver Tablas 2, 3), se puede decir que los diferentes módulos se comportan de forma robusta. La detección de ausencia del conductor por medio de un marcador es un método simple y efectivo para detectar si el conductor está presente en el puesto de conducción. En lo que respecta al cálculo del estado de los ojos, la base de datos utilizada para entrenar y validar los algoritmos faciales es un aspecto muy

Tabla 8: Resumen de los resultados (tasa de acierto) para la Actividad del tipo A01_X al clasificar el estado de los ojos para 8 conductores (4 hombres y 4 mujeres) sin gafas

	4 seg	**5 seg**	**6 seg**
A01	92.44	93.79	94.30

importante, pues aspectos como la pose, iluminación y resolución de las imágenes juegan un papel fundamental. De manera específica, y como se puede ver en Song et al. (2014), varias aproximaciones basadas en LBP, LTP, Gabor, HOG que en la base de datos ZJU (Pan et al. (2007)) obtienen una precisión de 89.19, 91.39, 85.04, 90.90 pero en cambio evaluados con la base de datos CEW obtienen 81.00, 83.59, 85.53, 90.35 respectivamente, lo que supone un decremento considerable. Es por ello que en la presente publicación se ha optado por emplear la base de datos CEW (ver Figura 3 para visualizar ejemplos de imágenes de dicha base de datos) para el cálculo del estado de los ojos, obteniendo el segundo algoritmo más robusto, sólo por detrás de una aproximación que presenta una carga computacional prácticamente 20 veces más. Además, mediante el empleo de LBP uniforme y CS-LBP se consiguen vectores con dimensionalidad relativamente baja, lo que conlleva una computación más fácil (Jung et al. (2016)).

De los resultados obtenidos tras la ejecución de las pruebas en entorno real controlado, se puede deducir que se detecta con bastante precisión los eventos de distracción y somnolencia, obteniéndose los resultados más elevados para la detección del estado de los ojos (A01). El hecho de que el conductor lleve gafas en el momento de las pruebas ocasionó que la tasa de reconocimiento se decrementara aproximadamente en un 2 %. Por otro lado, se trata de una solución completamente autónoma, apoyada por el módulo de detección de presencia, el cual se puede llevar a cabo, bien por medio de técnicas de aprendizaje automático (sin realizar ninguna modificación en el vehículo), o por medio de un marcador que se situaría en el reposacabezas del asiento; siendo esta alternativa más robusta. Por último, comentar que tampoco hay excesivas diferencias en relación a la duración del evento a detectar. Si bien es cierto que para los eventos con una duración de 4 segundos la tasa es menor, se achaca este decremento al instante temporal inicial y final marcado para dichos eventos, pues los conductores tardaban más en empezar a obedecer la orden dada (por ejemplo, cerrar los ojos) que en finalizar la orden dada (por ejemplo, abrir los ojos).

Por otro lado, la posición de la cámara juega un papel fundamental, pues la detección facial es el punto principal sobre el que pivota el resto del sistema. Esto es así porque los algoritmos de detección facial están entrenados con caras con una determinada pose para poder ser detectadas de frente o de perfil. Así pues, las caras capturadas en la imagen desde el interior vehicular deben tener una apariencia similar a aquellas con las que fueron entrenadas. Si la cámara se sitúa en el salpicadero en frente del conductor, se puede incrementar la detección facial en, aproximadamente, 8 puntos porcentuales respecto a si la cámara está en el espejo retrovisor (de un 86.88 % se puede pasar a un 96.46 %), lo cual está en consonancia con otros trabajos, que establecen que el salpicadero es el mejor lugar para la ubicación de la cámara (Vicente et al. (2015); Abtahi et al. (2014); López Romero (2016)).

El grado de detección facial logrado con la ubicación final de la cámara y utilizando el algoritmo de detección facial antes comentado, es superior al de todos los trabajos analizados (Hattori et al. (2006); Sigari (2009); You et al. (2013); Flores et al. (2010, 2011); Hong and Qin (2007); López Romero (2016)). Puesto que el resto de la arquitectura de estos sistemas trabaja sobre la detección de la cara del conductor, este es un factor determinante.

Además, el Algoritmo de Viola & Jones es frecuentemente usado por su facilidad de uso y por sus resultados más o menos robustos. En los trabajos analizados (Hattori et al. (2006); Sigari (2009); You et al. (2013); Flores et al. (2010, 2011); Hong and Qin (2007); López Romero (2016)) es usado como detector facial, lo que puede conllevar alguna restricción en el procesamiento. Por ejemplo, en You et al. (2013), con el objetivo de satisfacer los requisitos de 'tiempo real', se vieron obligados a bajar la resolución a 320×240 píxeles, lo que no ocurre en el sistema aquí propuesto.

Respecto a la detección del estado de los ojos, en You et al. (2013) se obtiene una precisión del 92 % utilizando para su evaluación 1780 imágenes. En Sigari (2009) se obtiene una precisión de 91.5 % evaluado con 817 imágenes. En la presente publicación se obtiene una precisión superior teniendo en cuenta 3000 imágenes (E.16). Flores et al. (2010) obtienen una tasa de clasificación del 94 % sobre una base de datos de 500 ojos cerrados y 1000 ojos abiertos. Si bien es una tasa bastante elevada, dicha base de datos ya se encuentra procesada, es decir, está compuesta por *patches* con imágenes de únicamente los ojos, con lo que la parte de pre-procesamiento no se tiene en cuenta.

En este sentido, en la alternativa aquí propuesta para detectar el estado de los ojos (LBP_RO) y (CS-LBP_RO) no se lleva a cabo una localización exhaustiva del ojo, es decir, es suficiente con detectar de forma más o menos precisa la región de los ojos, lo que aumenta su robustez frente a otros métodos que requieren localizar de forma específica el ojo.

Como se ha comentado anteriormente, el hecho de que las imágenes faciales presenten gafas, puede dificultar las tareas de clasificación. En la Figura 12 se pueden observar un conjunto de imágenes que han sido clasificadas de forma errónea en lo que corresponde al estado de los ojos debido a que las mismas presentan diferentes condiciones lumínicas y la presencia de gafas que dificulta las tareas de clasificación. Dichas imágenes han sido extraídas de la base de datos FERET (Phillips et al. (2000)), también usada para comparación de algoritmos de procesamiento facial.

También comentar que se han tenido en cuenta diversos estudios (Martin (2006); of Transportation (2016); You et al. (2013); Berri et al. (2014); Mbouna et al. (2013)) para establecer las ventanas temporales para la detección de los eventos de somnolencia y distracción, llegando a sus mismas conclusiones.

6.1. Limitaciones del sistema propuesto

Existen básicamente dos limitaciones en el sistema propuesto. El primero de ellos es que está orientado como sistema para la detección tanto de la distracción como de la somnolencia en condiciones diurnas. El sistema se debería completar con los algoritmos adecuados para contemplar condiciones nocturnas para ofrecer un funcionamiento ininterrumpido.

Por otro lado, y relacionado con la detección de la distracción visual, el sistema utiliza lo que se conoce como una apro-

Figura 12: Imágenes clasificadas erróneamente ya que en todas ellas los individuos presentan los ojos abiertos y han sido clasificadas con los ojos cerrados

ximación *coarse*, lo que podría suponer alguna limitación en alguna situación muy concreta. Por ejemplo, cuando el conductor esté realizando una tarea secundaria que involucre intervención visual, en la que presente una orientación en la cabeza detectada por el algoritmo de tracking como de distracción, pero pudiendo alternar de manera continua y constante la mirada entre la carretera y el foco de atención de dicha tarea secundaria.

Trabajo futuro

La arquitectura aquí propuesta podría completarse con las limitaciones antes comentadas (funcionamiento en condiciones nocturnas y detección precisa de la ubicación de los ojos para la obtención del foco de atención y poder establecer hacia donde está mirando el conductor) y con tres aspectos que contribuirían a mejorar la robustez. En primer lugar, se podría añadir al flujo de ejecución de la arquitectura un algoritmo para la detección de bostezos. Los bostezos aparecen como uno de los primeros síntomas de la somnolencia, con lo que su detección podría desencadenar recomendaciones y advertencias al conductor. Tanto para entrenar como para testear el grado de acierto del algoritmo a implementar, se podría hacer uso de la base de datos propuesta por Abtahi et al. (2014), que contiene dos bases de datos de vídeo para la comparación de modelos y algoritmos en la detección de bostezos.

En segundo lugar, se podría proponer el uso de un algoritmo de detección de gafas de sol en los conductores, pues en el caso de que las lleve, es imposible la detección del estado de los ojos. Por tanto, dicho algoritmo debería ser aplicado previamente a la detección del estado de los ojos y también antes de aplicar el algoritmo de detección del foco de atención del conductor (en caso de llevar a cabo su implementación).

Y en tercer lugar, el sistema se integrará con el GPS para la obtención de la posición real con el objetivo de posicionar el vehículo en un mapa. De esta manera, se podría obtener si el conductor está en un cruce. En tal situación, se podría diferenciar si el conductor está distraído o si, por el contrario, se está asegurando antes de cruzarlo.

English Summary

Automatic System to Detect Both Distraction and Drowsiness in Drivers Using Robust Visual Features.

Abstract

According to the most recent studies published by the World Health Organization (WHO) in 2013, it is estimated that 1.25 million people die as a result of traffic crashes. Many of them are caused by what it is known as inattention, whose main contributing factors are both distraction and drowsiness. Overall, it is estimated that inattention causes between 25 % and 75 % of the crashes and near-crashes. That is why this is a thoroughly studied field by the research community, where solutions to combat distraction and drowsiness, in particular, and inattention, in general, can be classified into three main categories, and, where computer vision has clearly become a non-obtrusive effective tool for the detection of both distraction and drowsiness. The aim of this paper is to propose, build and validate an architecture based on the analysis of visual characteristics by using computer vision techniques and machine learning to detect both distraction and drowsiness in drivers. Firstly, the modules have been tested with all its components independently using several datasets. More specifically, the presence/absence of the driver is detected with an accuracy of 100 %, 90.56 %, 88.96 % by using a marker positioned onto the headrest, the LBP operator and the CS-LBP operator, respectively. Regarding the eye closeness validation with CEW dataset, an accuracy of 93.39 % and 91.84 % is obtained using a new method using both LBP (LBP_RO) and CS-LBP (CS-LBP_RO). After performing several tests, the camera is positioned on the dashboard, increasing the accuracy of face detection from 86.88 % to 96.46 %. In connection with the tests performed in real-world settings, 16 drivers were involved performing several activities imitating different sings of sleepiness and distraction. Overall, an accuracy of 93.11 % is obtained considering all activities and all drivers.

Keywords:
Distraction and drowsiness detection Computer vision Perception and recognition Machine learning Monitoring and supervision

Agradecimientos

El origen de las actividades del presente trabajo ha sido realizado parcialmente gracias al apoyo tanto de la Fundación para el fomento en Asturias de la investigación científica aplicada y la tecnología (FICYT) y de la empresa SINERCO SL, por medio de la ejecución del proyecto '*Creación de algoritmos de visión artificial*', con referencia IE09-511.El presente trabajo se engloba en la tesis doctoral de Alberto Fernández Villán.

Apéndice A. Detección distracción visual en condiciones diurnas

En un trabajo previo recientemente publicado (Fernández et al. (2016)) analizamos un total de 1500 publicaciones con el objetivo de categorizar los diferentes métodos propuestos para la detección de la distracción en general, y de la distracción visual en particular, utilizando únicamente algoritmos de visión por computador. De acuerdo a dicha publicación y en lo que respecta a la distracción visual, existen principalmente dos aproximaciones.

La primera de ellas es la que se conoce como *coarse*, y se basa principalmente en asumir que el foco de atención del conductor (hacia dónde dirige la mirada) se puede aproximar utilizando la orientación de la cabeza. En la segunda aproximación, que se conoce como *fine*, los investigadores consideran tanto la orientación de la cabeza como la orientación de los ojos para obtener un foco de atención más preciso.

Cada una de las aproximaciones anteriores presenta ventajas e inconvenientes. En lo que respecta a la primera de las aproximaciones, se ha demostrado que la orientación de la cabeza es un indicador robusto del foco de atención del conductor (Murphy-Chutorian and Trivedi (2010)), y ambas medidas están estrechamente ligadas (Hammoud et al. (2005)).

Sin embargo, el conductor, en algunas ocasiones y sobre todo cuando lleva a cabo alguna tarea secundaria que requiere involucración manual o visual (por ejemplo buscar algo en la guantera del vehículo) es común que posicione su cabeza en una posición intermedia entre las dos situaciones que requieren su atención (en este caso, la guantera y la carretera) y mediante continuos y constantes movimientos oculares lleva a cabo la inspección visual de ambas tareas. En esta situación, un algoritmo de *tracking* facial reconocería esta situación como de distracción visual al basarse únicamente en la posición de la cabeza. Por lo tanto, en una situación ideal, se deberían considerar tanto la posición de la cabeza como la posición de los ojos.

Sin embargo, el hecho de obtener la posición de los ojos en todo momento no es una tarea fácil (Song et al. (2013)), debido principalmente a aspectos como: a) las expresiones faciales, b) oclusiones (por ejemplo si el conductor lleva gafas o gafas de sol, o parpadeos), c) pose (por ejemplo puede ser posible que no sea posible recoger la posición de los dos ojos en una posición de perfil), d) diferentes condiciones de la imagen y su calidad (iluminación o baja calidad de la imagen, entre otros). Es por ello que, para lidiar con estos aspectos antes comentados, se instalen diversas cámaras distribuidas por el habitáculo.

A medida que aumenta el número de cámaras, es posible obtener mayor información. Sin embargo, esta información habría que fusionarla (se obtienen datos de cada una de las cámaras) para obtener unos resultados finales. Además, estos sistemas son difíciles de configurar, requieren de calibración periódica para comprobar el estado de las cámaras, pues pequeños desajustes pueden ocasionar problemas (Ahlstrom and Dukic (2010)), y pueden presentarse como intrusivos al conductor. Además, existen muchas situaciones en las que es imposible obtener la posición exacta de los ojos debido a las condiciones propias del entorno vehicular: a) reflejos en las gafas, b) parpadeos, c) pose muy acusada del conductor (por ejemplo si presenta su cabeza mirando hacia el techo o el suelo del vehículo es imposible recoger la información de sus ojos - ambas posiciones posibles con un conductor dormido), d) diferentes condiciones de iluminación (Fernández et al. (2016)).

Sin embargo, y con el objeto de profundizar en algoritmos para la detección de los ojos en condiciones no controladas, se hicieron pruebas con dos algoritmos recientemente propuestos para la detección de los ojos. En este sentido, uno de los algoritmos que mejor se ha comportado en vista de los resultados obtenidos de las diversas publicaciones analizadas (Song et al. (2013)) es el algoritmo ASEF (Average of synthetic exact filters), el cual destaca por tener una carga computacional bastante baja, aspecto fundamental para su integración en un entorno embebido, propuesto en Bolme et al. (2009).

Es por ello que se realizaron pruebas preliminares a partir de su implementación en C++ que permite una integración directa con OpenCV. También se experimentó con el algoritmo propuesto en Timm and Barth (2011) para la detección de los ojos, pues por los resultados mostrados en el mismo, son más que prometedores. Además, el algoritmo destaca, como en el caso anterior, por su baja carga computacional. Es por ello que también se realizaron pruebas preliminares a partir de su implementación en C++ que permite una integración directa con OpenCV. En ambos casos, los resultados preliminares (utilizando imágenes con los ojos abiertos de la base de datos CEW) no fueron buenos, no considerándose adecuado para lidiar con las condiciones de iluminación y pose en los entornos vehiculares. En este sentido, los resultados fueron peores cuando:

- Las condiciones lumínicas afectaban la imagen
- La cara presentaba una pose acusada
- Las caras presentaban gafas

Además, comentar que ambos algoritmos (Timm and Barth (2011); Bolme et al. (2009)) fueron usados como etapa de preprocesamiento para el alineamiento facial, con el objetivo de obtener el ángulo de inclinación obtenido a partir de la posición de los ojos. Tras realizar pruebas con diferentes bases de datos faciales, se comprobó experimentalmente que es más robusto utilizar un detector de *landmarks* faciales (Uřičář et al. (2012)), el cual usamos finalmente en la etapa de pre-procesamiento.

Para más información en este punto, se recomienda leer el Apéndice Pre-procesamiento facial para obtener el estado de los ojos. Es por ello que, finalmente, en la presente publicación, se ha optado por utilizar la primera aproximación (*coarse*), al requerir una solución más compacta, más fácil de instalar y configurar, sin perder de vista los algoritmos para la detección de los ojos, con vistas a una futura integración en la arquitectura.

Finalmente y por completar la información presentada, la detección de distracción visual en condiciones nocturnas se realiza, generalmente, de forma más precisa, pues los sistemas de iluminación infrarroja hacen que la pupila adquiera un brillo característico que posibilita una fácil localización de los ojos. De día, dichos sistemas se ven perjudicados por la luz infrarroja solar (Fernández et al. (2016)).

Apéndice B. Pre-procesamiento facial para obtener el estado de los ojos

Es necesario realizar un pre-procesamiento facial antes de extraer las características de la región facial (en este caso, aplicar el operador LBP/CS-LBP) para obtener el estado de los ojos. Con el objetivo de lidiar con el escalado, rotación, pose e inexactitudes del algoritmo de detección facial, se propuso en Fernández et al. (2015c) un algoritmo para normalizar la región facial (Algoritmo 3). Los principales pasos se comentan a continuación. Una vez que la cara ha sido detectada, el siguiente punto radica en aplicar un detector robusto de características faciales basado en 'Modelos de Partes Deformables' (DPM) propuesto por Uřičář et al. (2012). La salida del detector se corresponde con estimaciones de localizaciones para un conjunto de puntos característicos en la imagen: esquinas de los ojos, esquinas de la boca y nariz. A continuación, se aplica un algoritmo para que las caras sean rotadas y alineadas de manera que los ojos siempre se encuentren en las mismas coordenadas en la imagen final. Para ello, con el objetivo de calcular el ángulo de desviación de la cara, se calcula una recta de regresión que utiliza los cuatro puntos de los ojos. Por último, se calcula la región facial por encima y por debajo de los ojos, para que únicamente información relevante se procese en las etapas siguientes del algoritmo. Para más detalles de este algoritmo, se recomienda la lectura de una publicación previa, donde se comprueba que este algoritmo de normalización incrementa la tasa de reconocimiento significativamente Fernández et al. (2015c). De esta manera, se consigue extraer la región facial normalizada alrededor de los ojos de manera totalmente automatizada para posteriormente extraer el vector de características mediante el operador correspondiente.

Algoritmo 3 Normalización facial del ROI
eyes_distance_r, eye_line_r, size

1: Se localizan 8 puntos faciales. Se calcula recta de regresión basada en los 4 puntos de los ojos. El ángulo α de desviación es calculado y la imagen se rota teniendo en cuenta dicho ángulo para alinear las imágenes
2: La distancia Euclídea d_0 se calcula entre los ojos en la imagen rotada
3: La distancia entre los ojos en la imagen redimensionada se calcula según la fórmula $d_t = size.w * eyes\_distance\_r$
4: El ratio r se calcula como $r = d_0/d_t$
5: La anchura w_0 and la altura h_0 de la región alrededor de los ojos se calcula como $w_0 = r * size.w$ y $h_0 = r * size.h$
6: Las coordenadas de las esquinas de la región facial en la imagen rotada se calculan como $x_l = x_e - w_0/2$, $y_t = y_e - h_0/eye\_line\_r$, $x_r = x_l + w_0$ y $y_b = y_t + h_0$, donde x_l es la coordenada x del borde izquierdo, x_e es la coordenada x del punto medio entre los ojos, y_t es la coordenada y del borde superior, y_e es la coordenada y de los ojos, x_r es la coordenada x del borde derecho e y_b es la coordenada y del borde inferior. Este *ROI* es extraído de la imagen
7: **return** *ROI*

Para establecer dicha región, los parámetros de entrada al algoritmo que se han usado son los siguientes: eyes_distance_r = 0.7, eye_line_r = 2, size = 80x36 píxeles, que de manera gráfica se pueden ver en la Figura B.13.

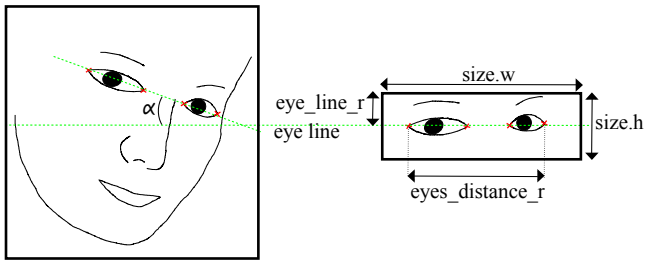


Figura B.13: Representación visual del algoritmo de pre-procesamiento de la región facial mostrando los parámetros de entrada del mismo

Apéndice C. Operadores LBP y CS-LBP

Formalmente, el operador LBP toma la siguiente forma:

$$LBP(x_c, y_c) = \sum_{p=0}^{7} 2^p s(g_p - g_c) \tag{C.1}$$

donde p recorre los 8 vecinos alrededor del pixel central c, g_c y g_p son los valores del nivel de gris en c y p respectivamente y

$$s(x) = \begin{cases} 1, & \text{if } x \geq 0 \\ 0, & otherwise \end{cases} \tag{C.2}$$

Este operador fue extendido para usar vecindarios de diferentes tamaños Ojala et al. (2002), posibilitando lidiar con texturas a diferentes escalas. Este hecho se denota mediante (P, R), donde P representa el número de vecinos y R representa el radio para el vecindario. Cuando los puntos no se corresponden con posiciones enteras, el valor de intensidad para un determinado punto es bilinealmente interpolado. A esta implementación se le conoce como LBP ($LBP_{P,R}$):

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} 2^p s(g_p - g_c) \tag{C.3}$$

Otra extensión al operador original es lo que se conoce como patrones uniformes Ojala et al. (2002). Un patrón LBP es uniforme si contiene, como máximo, dos transiciones de 0 a 1 (o viceversa) visto como un *buffer* circular de bits. Por ejemplo, 00000000, 00011110 and 10000011 son los tres patrones uniformes. La uniformidad de los patrones es un hecho importante, pues codifica información como esquinas y bordes. A pesar de que sólo 58 de los 256 patrones (para un vecindario de 8) son uniformes, alrededor del 90 % de los patrones observados son uniformes Ojala et al. (2002). Para identificar a los patrones uniformes, se utiliza la siguiente notación: $LBP^{u2}_{P,R}$. Es por ello que los patrones uniformes se pueden considerar un método efectivo para reducir la dimensionalidad de los datos. Otras extensiones al operador original, también propuestas en la mencionada publicación, se corresponden con: 1) los patrones invariantes a

la rotación, los cuales adquieren la siguiente notación: $LBP^{ri}_{P,R}$ y representan un total de 36 del total de 256 patrones para un vecindario de 8 píxeles; 2) la conjunción de patrones uniformes e invariantes a la rotación: $LBP^{riu2}_{P,R}$ y representan un total de 10 del total de los 256.

Se realizaron pruebas preliminares teniendo en cuenta estos tipos ($LBP^{u2}_{P,R}$,$LBP^{ri}_{P,R}$,$LBP^{riu2}_{P,R}$) para diferentes valores de P y R. Según las publicaciones analizadas y la experiencia previa en otras publicaciones (Fernández et al. (2015c); Villan et al. (2016); Losada et al. (2013)) $P = \{8, 16\}$ y $R = \{1, 2, 3\}$.

Tras analizar dichas pruebas, se llega a la conclusión de que los valores con mayor tasa de reconocimiento son los patrones uniformes $LBP^{u2}_{P,R}$ con un número de vecinos de $P = 8$ y un radio de $R = 2$.

Respecto al operador CS-LBP, formalmente toma la siguiente formulación:

$$CS - LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{(P/2)-1} 2^p s(g_p - g_{p+(P/2)}) \qquad \text{(C.4)}$$

Con este operador también se realizaron pruebas preliminares, obteniendo los mejores resultados, como en el caso del operador LBP con un número de vecinos de $P = 8$ y un radio de $R = 2$.

Una vez se han definidas las características de ambos operadores, el siguiente paso consiste en dividir la imagen en m regiones $\{R_0, ..., R_{m-1}\}$ y para cada una de esas regiones, se construye el histograma a partir de la imagen LBP generada tras aplicar el operador seleccionado. El número de divisiones juega, por tanto, un papel importante en la longitud final del histograma generado. Escoger este número de divisiones no es tampoco tarea fácil y se requieren diversas pruebas para establecer dicho número.

En lo que se refiere a la tarea de detección de ausencia del conductor mediante LBP y CS-LBP el número de divisiones que mejores resultados arrojó fueron de 5×5. Para la tarea de detección del estado de los ojos, el número de divisiones óptimo fue de 6×3.

Apéndice D. Comparación entre los algoritmos de Viola & Jones y PICO

En relación a las necesidades computacionales, el algoritmo de PICO presenta unas prestaciones mucho superiores. En relación a la precisión en dicha detección, en la propia publicación de PICO, comparan su algoritmo con el algoritmo de Viola & Jones. Para ello, hacen uso de la base de datos FDDB (Jain and Learned-Miller (2010)), que contiene 5171 imágenes de caras adquiridas en diferentes condiciones y su uso es común para la comparación de algoritmos de detección facial.

La comparación puede verse en las Figuras D.14,D.15, donde se representan, las curvas ROC discretas y continuas, respectivamente. Como se puede observar, el método de PICO presenta mejores resultados.

Si bien en la presente publicación no se realizaron pruebas cuantitativas para obtener de manera precisa la diferencia entre ambos algoritmos, de manera cualitativa sí que se puede concluir que el algoritmo de PICO presenta mejores resultados. Este hecho unido a que la carga computacional del algoritmo PICO es mucho inferior, lo hace mucho más interesante para su inclusión en sistemas con reducidas capacidades de cómputo.

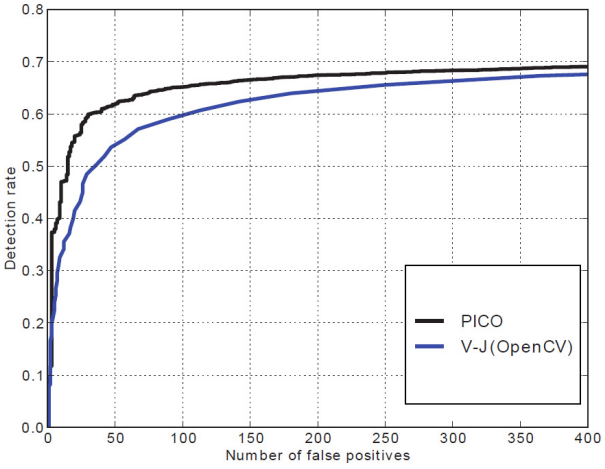


Figura D.14: Comparacion de las curvas ROC discretas tanto para los métodos de PICO y de Viola & Jones empleando la base de datos FDDB

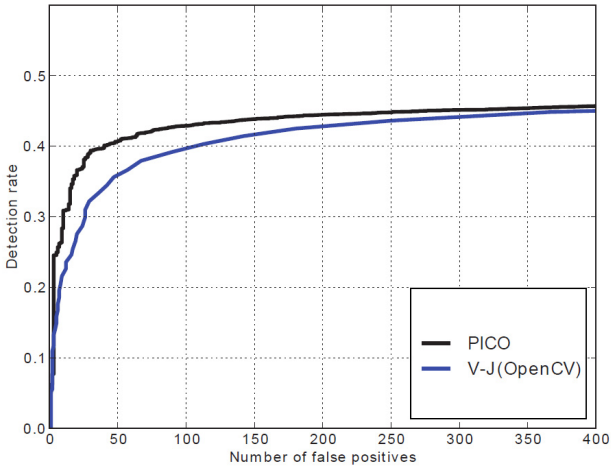


Figura D.15: Comparacion de las curvas ROC contínuas tanto para los métodos de PICO y de Viola & Jones empleando la base de datos FDDB

Apéndice E. Entrenamiento con SVM

Las Máquinas de Soporte Vectorial (SVMs, por sus siglas en inglés) representan un método de aprendizaje automático muy popular para tareas de clasificación, regresión y otras de aprendizaje Chang and Lin (2011), es también una poderosa herramienta para tareas relacionadas con la extracción de información de la región facial.

La SVM construye un hiperplano o conjunto de hiperplanos en un espacio de dimensionalidad muy alta (o incluso infinita) que, de forma óptima (hiperplano que tenga la máxima

distancia - margen - con los puntos que estén más cerca de él mismo), separe a los puntos de una clase de la de otra Vapnik (1998). Dado una equivalencia no lineal Φ que transforma los datos a uno con mayor dimensionalidad, los *kernels* presentan la siguiente formulación $K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$, siendo $\{(x_i, y_i), i = 1, ..., l\}$ un conjunto de datos de entrenamiento etiquetados, donde $x_i \in R^n$, $y_i \in \{1, -1\}$.

A pesar de que se están proponiendo nuevos *kernels* Shan et al. (2009), los más populares son: a) lineal, b) polinomial, y c) función de base radial (RBF, por sus siglas en inglés) (Hsu et al. (2003)).

Para trabajar con las SVMs, se hizo uso de la librería LIBSVM (Chang and Lin (2011)). A continuación, y como está sugerido en Hsu et al. (2003), se llevaron a cabo los siguientes pasos:

1. Transformar los datos de entrada al formato de la librería.
2. Realizar un escalado de los datos.
3. Experimentar con diferentes *kernels* y seleccionar el que mejores resultados arroje.
4. Realizar validación cruzada y lo que se conoce como *grid-search* para establecer los parámetros de la SVM.

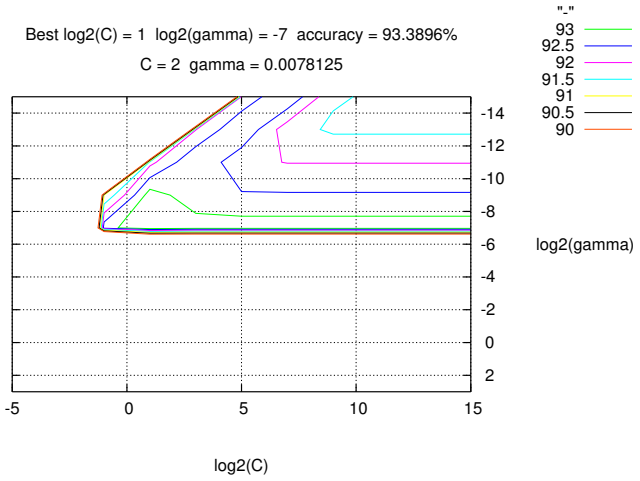


Figura E.16: *Grid-search* sobre los parámetros C y γ mediante el empleo de la técnica de *cross-validation* para el operador LBP sobre la región de los ojos utilizando el kernel de tipo RBF correspondiente a la SVM para la detección del estado de los ojos

Para el escalado de los datos de entrada se hicieron pruebas con dos rangos de datos recomendados (Hsu et al. (2003)): $[-1, +1]$ y $[0, +1]$, obteniendo mejores resultados con el intervalo $[-1, +1]$, con lo que se estableció dicho intervalo de escalado. Tras realizar diversas pruebas con los tres *kernels* antes comentados (lineal, polinomial y RBF), se seleccionó éste último por presentar mejores resultados. Una vez seleccionado, se realizó lo que se conoce como *cross-validation* y *grid-search* para establecer los mejores parámetros en dicho *kernel*, los cuales se gobiernan principalmente mediante dos parámetros: C y γ.

El hecho de identificar estos dos parámetros (C, γ) es con el objetivo de que el clasificador pueda predecir con robustez

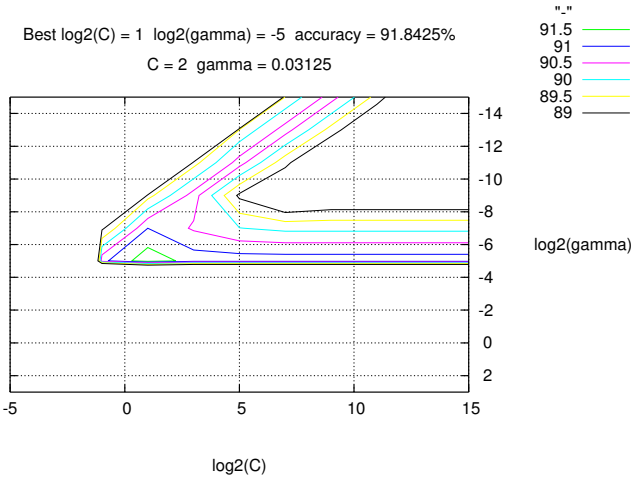


Figura E.17: *Grid-search* sobre los parámetros C y γ mediante el empleo de la técnica de *cross-validation* para el operador CS-LBP sobre la región de los ojos utilizando el kernel de tipo RBF correspondiente a la SVM para la detección del estado de los ojos

los datos desconocidos (por ejemplo, los datos de entrenamiento). De manera más específica, se llevó a cabo el proceso de *grid-search* de los parámetros C y γ usando *cross-validation*. Es decir, se cogen pares de valores de (C, γ) y para cada uno de ellos se realiza *cross-validation*. Al final del proceso, el par de valores que mejores resultados obtiene es el que se selecciona.

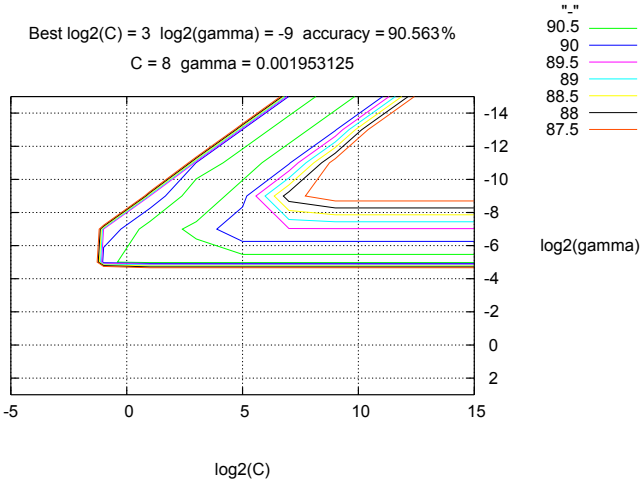


Figura E.18: *Grid-search* sobre los parámetros C y γ mediante el empleo de la técnica de *cross-validation* para el operador LBP sobre la ROI utilizando el kernel de tipo RBF correspondiente a la SVM para la detección de ausencia del conductor

Para evidenciar de manera visual este procedimiento, mostramos, en las Figuras E.16 y E.17 los resultados tras las diversas ejecuciones en lo que al cálculo del estado de los ojos se refiere. En la Figura E.16 se puede ver que el mejor resultado obtenido es de 93.39 % y se corresponde a aplicar el operador LBP sobre la región de los ojos.

En la Figura E.17 se puede ver que el mejor resultado es de 91.84 % y se corresponde a aplicar el operador CS-LBP sobre

la región de los ojos. Además, en relación a la detección de la ausencia del conductor utilizando tanto LBP como CS-LBP y SVM, en las Figuras E.18 y E.19 se muestran los resultados utilizando el mismo procedimiento antes comentado.

En la Figura E.18 se puede ver que el mejor resultado obtenido es de 90.56 % y se corresponde a aplicar el operador LBP sobre el ROI establecido para la detección automática de la presencia del conductor. En la Figura E.19 se puede ver que el mejor resultado obtenido es de 88.96 % y se corresponde a aplicar el operador CS-LBP sobre el ROI establecido.

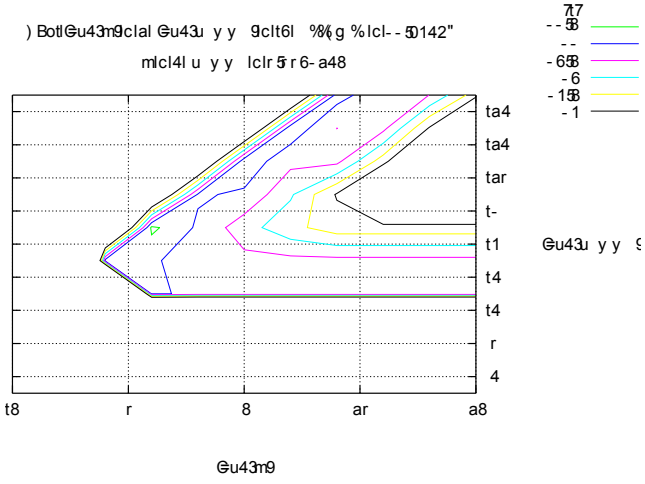


Figura E.19: *Grid-search* sobre los parámetros C y γ mediante el empleo de la técnica de *cross-validation* para el operador CS-LBP sobre la ROI utilizando el kernel de tipo RBF correspondiente a la SVM para la detección de ausencia del conductor

Referencias

Abtahi, S., Omidyeganeh, M., Shirmohammadi, S., Hariri, B., 2014. Yawdd: a yawning detection dataset. In: Proceedings of the 5th ACM Multimedia Systems Conference. ACM, pp. 24–28.

Ahlstrom, C., Dukic, T., 2010. Comparison of eye tracking systems with one and three cameras. In: Proceedings of the 7th International Conference on Methods and Techniques in Behavioral Research. ACM, p. 3.

Ahonen, T., Hadid, A., Pietikainen, M., 2006. Face description with local binary patterns: Application to face recognition. IEEE transactions on pattern analysis and machine intelligence 28 (12), 2037–2041.

Asthana, A., Marks, T. K., Jones, M. J., Tieu, K. H., Rohith, M., 2011. Fully automatic pose-invariant face recognition via 3d pose normalization. In: 2011 International Conference on Computer Vision. IEEE, pp. 937–944.

Berri, R. A., Silva, A. G., Parpinelli, R. S., Girardi, E., Arthur, R., 2014. A pattern recognition system for detecting use of mobile phones while driving. In: Computer Vision Theory and Applications (VISAPP), 2014 International Conference on. Vol. 2. IEEE, pp. 411–418.

Bolme, D. S., Draper, B. A., Beveridge, J. R., 2009. Average of synthetic exact filters. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, pp. 2105–2112.

Boyraz, P., Yang, X., Hansen, J. H., 2012. Computer vision systems for context-aware active vehicle safety and driver assistance. In: Digital Signal Processing for In-Vehicle Systems and Safety. Springer, pp. 217–227.

Chang, C.-C., Lin, C.-J., 2011. Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2 (3), 27.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 1. IEEE, pp. 886–893.

Daniluk, M., Rezaei, M., Nicolescu, R., Klette, R., 2014. Eye status based on eyelid detection: A driver assistance system. In: International Conference on Computer Vision and Graphics. Springer, pp. 171–178.

Dasgupta, A., George, A., Happy, S., Routray, A., Shanker, T., 2013. An on-board vision based system for drowsiness detection in automotive drivers. International Journal of Advances in Engineering Sciences and Applied Mathematics 5 (2-3), 94–103.

Devi, M. S., Bajaj, P. R., 2008. Driver fatigue detection based on eye tracking. In: 2008 First International Conference on Emerging Trends in Engineering and Technology. IEEE, pp. 649–652.

Dinges, D. F., Grace, R., 1998. Perclos: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance. US Department of Transportation, Federal Highway Administration, Publication Number FHWA-MCRT-98-006.

Dong, Y., Hu, Z., Uchimura, K., Murayama, N., 2011. Driver inattention monitoring system for intelligent vehicles: A review. IEEE transactions on intelligent transportation systems 12 (2), 596–614.

Fernandez, A., Carus, J., Usamentiaga, R., Alvarez, E., Casado, R., 2017. Wearable and ambient sensors to health monitoring using computer vision and signal processing techniques. Journal of Networks In press.

Fernández, A., Carús, J. L., Usamentiaga, R., Alvarez, E., Casado, R., 2015a. Unobtrusive health monitoring system using video-based physiological information and activity measurements. In: Computer, Information and Telecommunication Systems (CITS), 2015 International Conference on. IEEE, pp. 1–5.

Fernández, A., Casado, R., Usamentiaga, R., 2015b. A real-time big data architecture for glasses detection using computer vision techniques. In: Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on. IEEE, pp. 591–596.

Fernández, A., García, R., Usamentiaga, R., Casado, R., 2015c. Glasses detection on real images based on robust alignment. Machine Vision and Applications 26 (4), 519–531.

Fernández, A., Usamentiaga, R., Carús, J. L., Casado, R., 2016. Driver distraction using visual-based sensors and algorithms. Sensors 16 (11), 1805.

Flores, M. J., Armingol, J. M., de la Escalera, A., 2010. Real-time warning system for driver drowsiness detection using visual information. Journal of Intelligent & Robotic Systems 59 (2), 103–125.

Flores, M. J., de la Escalera, A., et al., 2011. Sistema avanzado de asistencia a la conducción para la detección de la somnolencia. Revista Iberoamericana de Automática e Informática Industrial RIAI 8 (3), 216–228.

Forsman, P. M., Vila, B. J., Short, R. A., Mott, C. G., Van Dongen, H. P., 2013. Efficient driver drowsiness detection at moderate levels of drowsiness. Accident Analysis & Prevention 50, 341–350.

Hadid, A., Pietikäinen, M., 2013. Demographic classification from face videos using manifold learning. Neurocomputing 100, 197–205.

Hammoud, R. I., Wilhelm, A., Malawey, P., Witt, G. J., 2005. Efficient real-time algorithms for eye state and head pose tracking in advanced driver support systems. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 2. IEEE, pp. 1181–vol.

Hansen, D. W., Ji, Q., 2010. In the eye of the beholder: A survey of models for eyes and gaze. IEEE Transactions on pattern analysis and machine intelligence 32 (3), 478–500.

Hattori, A., Tokoro, S., Miyashita, M., Tanaka, I., Ohue, K., Uozumi, S., 2006. Development of forward collision warning system using the driver behavioral information. Tech. rep., SAE Technical Paper.

Heikkilä, M., Pietikäinen, M., Schmid, C., 2009. Description of interest regions with local binary patterns. Pattern recognition 42 (3), 425–436.

Hong, T., Qin, H., 2007. Drivers drowsiness detection in embedded system. In: Vehicular Electronics and Safety, 2007. ICVES. IEEE International Conference on. IEEE, pp. 1–5.

Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al., 2003. A practical guide to support vector classification.

Jain, V., Learned-Miller, E. G., 2010. Fddb: A benchmark for face detection in unconstrained settings. UMass Amherst Technical Report.

Jo, J., Lee, S. J., Park, K. R., Kim, I.-J., Kim, J., 2014. Detecting driver drowsiness using feature-level fusion and user-specific classification. Expert Systems with Applications 41 (4), 1139–1152.

Jung, J.-Y., Kim, S.-W., Yoo, C.-H., Park, W.-J., Ko, S.-J., 2016. Lbp-ferns-based feature extraction for robust facial recognition. IEEE Transactions on Consumer Electronics 62 (4), 446–453.

Lee, S. J., Jo, J., Jung, H. G., Park, K. R., Kim, J., 2011. Real-time gaze estima-

tor based on driver's head orientation for forward collision warning system. IEEE Transactions on Intelligent Transportation Systems 12 (1), 254–267.

Li, H., Lin, Z., Shen, X., Brandt, J., Hua, G., 2015. A convolutional neural network cascade for face detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5325–5334.

Liu, C. C., Hosking, S. G., Lenné, M. G., 2009. Predicting driver drowsiness using vehicle measures: Recent insights and future challenges. Journal of safety research 40 (4), 239–245.

López Romero, W. L., 2016. Sistema de control del estado de somnolencia en conductores de vehículos.

Losada, D. G., López, G. A. R., Acevedo, R. G., Villán, A. F., 2013. Aviue-artificial vision to improve the user experience. In: New Concepts in Smart Cities: Fostering Public and Private Alliances (SmartMILE), 2013 International Conference on. IEEE, pp. 1–6.

Lu, L., Ning, X., Qian, M., Zhao, Y., 2011. Close eye detected based on synthesized gray projection. In: Advances in Multimedia, Software Engineering and Computing Vol. 2. Springer, pp. 345–351.

Markuš, N., Frljak, M., Pandžić, I. S., Ahlberg, J., Forchheimer, R., 2014. Object detection with pixel intensity comparisons organized in decision trees. arXiv preprint arXiv:1305.4537.

Martin, E., 2006. Breakthrough research on real-world driver behavior released. National Highway Traffic Safety Administration.

Mbouna, R. O., Kong, S. G., Chun, M.-G., 2013. Visual analysis of eye state and head pose for driver alertness monitoring. IEEE transactions on intelligent transportation systems 14 (3), 1462–1469.

Murphy-Chutorian, E., Trivedi, M. M., 2010. Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness. IEEE Transactions on intelligent transportation systems 11 (2), 300–311.

Noori, S. M. R., Mikaeili, M., 2016. Driving drowsiness detection using fusion of electroencephalography, electrooculography, and driving quality signals. Journal of medical signals and sensors 6 (1), 39.

Nuevo, J., Bergasa, L. M., Jiménez, P., 2010. Rsmat: Robust simultaneous modeling and tracking. Pattern Recognition Letters 31 (16), 2455–2463.

of Transportation, D., 2016. Pennsylvania driver's manual. `https://goo.gl/XCER8C`, accessed: 2016-09-018.

Ojala, T., Pietikäinen, M., Harwood, D., 1996. A comparative study of texture measures with classification based on featured distributions. Pattern recognition 29 (1), 51–59.

Ojala, T., Pietikainen, M., Maenpaa, T., 2002. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on pattern analysis and machine intelligence 24 (7), 971–987.

Organization, W. H., 2016. Global status report on road safety 2015. `http://goo.gl/jMoJ4l`, accessed: 2016-07-01.

Pan, G., Sun, L., Wu, Z., Lao, S., 2007. Eyeblink-based anti-spoofing in face recognition from a generic webcamera. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. IEEE, pp. 1–8.

Peden, M., Toroyan, T., Krug, E., Iaych, K., et al., 2016. The status of global road safety: The agenda for sustainable development encourages urgent action. Journal of the Australasian College of Road Safety 27 (2), 37.

Phillips, P. J., Moon, H., Rizvi, S. A., Rauss, P. J., 2000. The feret evaluation methodology for face-recognition algorithms. IEEE Transactions on pattern analysis and machine intelligence 22 (10), 1090–1104.

RACE, A. y. l. D., 2016. Los conductores españoles reconocen sufrir más somnolencia al volante que los usuarios europeos. `http://goo.gl/mui9S3`, accessed: 2016-07-01.

Regan, M. A., Hallett, C., Gordon, C. P., 2011. Driver distraction and driver inattention: Definition, relationship and taxonomy. Accident Analysis & Prevention 43 (5), 1771–1781.

Sahayadhas, A., Sundaraj, K., Murugappan, M., 2012. Detecting driver drowsiness based on sensors: a review. Sensors 12 (12), 16937–16953.

Selvakumar, K., Jerome, J., Rajamani, K., Shankar, N., 2015. Real-time vision based driver drowsiness detection using partial least squares analysis. Journal of Signal Processing Systems, 1–12.

Shan, C., 2012. Learning local binary patterns for gender classification on real-world face images. Pattern Recognition Letters 33 (4), 431–437.

Shan, C., Gong, S., McOwan, P. W., 2009. Facial expression recognition based on local binary patterns: A comprehensive study. Image and Vision Computing 27 (6), 803–816.

Sigari, M. H., 2009. Driver hypo-vigilance detection based on eyelid behavior. In: Advances in Pattern Recognition, 2009. ICAPR'09. Seventh International Conference on. IEEE, pp. 426–429.

Slawiñski, E., Mut, V., Penizzotto, F., 2015. Sistema de alerta al conductor basado en realimentación vibro-táctil. Revista Iberoamericana de Automática e Informática Industrial RIAI 12 (1), 36–48.

Song, F., Tan, X., Chen, S., Zhou, Z.-H., 2013. A literature survey on robust and efficient eye localization in real-life scenarios. Pattern Recognition 46 (12), 3157–3173.

Song, F., Tan, X., Liu, X., Chen, S., 2014. Eyes closeness detection from still images with multi-scale histograms of principal oriented gradients. Pattern Recognition 47 (9), 2825–2838.

StopChatear, 2016. Uso de los smartphones en la conducción. `http://goo.gl/67dvtn`, accessed: 2016-07-01.

Talbot, R., Fagerlind, H., Morris, A., 2013. Exploring inattention and distraction in the safetynet accident causation database. Accident Analysis & Prevention 60, 445–455.

Tan, X., Triggs, B., 2010. Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE transactions on image processing 19 (6), 1635–1650.

Timm, F., Barth, E., 2011. Accurate eye centre localisation by means of gradients. VISAPP 11, 125–130.

Uřičář, M., Franc, V., Hlaváč, V., 2012. Detector of facial landmarks learned by the structured output svm. VIsAPP 12, 547–556.

Vapnik, V., 1998. Statistical learning theory wiley new york google scholar.

Vicente, F., Huang, Z., Xiong, X., De la Torre, F., Zhang, W., Levi, D., 2015. Driver gaze tracking and eyes off the road detection system. IEEE Transactions on Intelligent Transportation Systems 16 (4), 2014–2027.

Villan, A. F., Candas, J. L. C., Fernandez, R. U., Tejedor, R. C., 2016. Face recognition and spoofing detection system adapted to visually-impaired people. IEEE Latin America Transactions 14 (2), 913–921.

Viola, P., Jones, M. J., 2004. Robust real-time face detection. International journal of computer vision 57 (2), 137–154.

Vural, E., Cetin, M., Ercil, A., Littlewort, G., Bartlett, M., Movellan, J., 2007. Drowsy driver detection through facial movement analysis. In: International Workshop on Human-Computer Interaction. Springer, pp. 6–18.

You, C.-W., Lane, N. D., Chen, F., Wang, R., Chen, Z., Bao, T. J., Montes-de Oca, M., Cheng, Y., Lin, M., Torresani, L., et al., 2013. Carsafe app: alerting drowsy and distracted drivers using dual cameras on smartphones. In: Proceeding of the 11th annual international conference on Mobile systems, applications, and services. ACM, pp. 13–26.

Zhang, Z., Zhang, J.-s., 2006. Driver fatigue detection based intelligent vehicle control. In: 18th International Conference on Pattern Recognition (ICPR'06). Vol. 2. IEEE, pp. 1262–1265.