Face Recognition and Spoofing Detection System Adapted To Visually-Impaired People

A. Fernández, J. L. Carús, R. Usamentiaga and R. Casado

***Abstract*— According to estimates by the World Health Organization, about 285 million people suffer from some kind of visual disability, of whom 39 million are blind, resulting in 0.7% of the world population. Computer vision techniques and image analysis can help improve visually-impaired people. In this project, a system that allows for facial recognition and detection of spoofing adapted to the needs of disabled people is proposed, implemented and validated. The architecture has been carefully selected and subsequently implemented following an innovative facial normalization algorithm in order to increase both the recognition rate of facial identification and spoofing detection. The information provided to the user is composed by the name of the person identified and whether it is real or fake image (photograph). This information is provided by means of a text-to-speech tool. This architecture can be integrated into video door-phone installations (videointercom installations), devices with reduced computing capabilities or the users´ mobile phones. The architecture has been validated in a real environment with both real users and printed images achieving very good results.**

***Keywords*— face recognition, spoofing detection, visually-impaired, system architecture.**

I. INTRODUCCION

LA CEGUERA es una discapacidad visual que afecta a un 0.7% de la población mundial. Según las últimas estimaciones, casi un millón de personas en España padecen algún tipo de discapacidad visual y debido a las enfermedades retinianas mencionadas, alrededor de 70.000 personas presentan ceguera total. Según estimaciones de la Organización Mundial de la Salud (OMS), alrededor de 285 millones de personas padecen algún tipo de discapacidad visual de las cuales 39 millones son ciegas, lo que supone un 0.7% de la población mundial [1].

La discapacidad visual afecta de manera desigual a los distintos grupos de edad siendo más incisiva en personas mayores de 50 años representando el 65% del total (a pesar de que este grupo sólo representa el 20% del total de la población) [2]. Entre los cambios que se producen en la visión a consecuencia de la edad podemos destacar [1]:

(1) Pérdida de la sensibilidad de la retina a la iluminación que origina una necesidad de utilizar iluminación más brillante.
(2) Opacidad del cristalino que ocasiona menor visión y reflejos molestos.
(3) Elasticidad del cristalino y pérdida de la capacidad para enfocar.
(4) Degeneración del vítreo que provoca la visión de manchas.
(5) Disminución de la capacidad de las conjuntivas y glándulas lagrimales para lubricar adecuadamente los ojos.

Todo ello provoca que con la edad se pierda parte de la capacidad visual y se desarrollen patologías como pueden ser las cataratas, el glaucoma, la degeneración macular, afecciones parpebrales o la sequedad de los ojos [1].

Además, las estimaciones apuntan a un mayor envejecimiento de la población en Europa. En la Unión Europea (UE), se estima que la población mayor de 65 años y susceptible de padecer algunas de las principales patologías de ceguera aumente del 17.4% actual al 29% en 2050 [1]. La situación de la ceguera en España es muy parecida a la que encontramos en Europa o en otros países desarrollados; se prevé que en el futuro ésta aumente como consecuencia del incremento de distintos factores de riesgo como el envejecimiento de la población o el aumento en prevalencia de la diabetes.

Actualmente la tasa de personas mayores de 65 años en España se sitúa en torno a 17%, muy parecida a la tasa media registrada en Europa que es del 17.4% [3]. Sin embargo, se estima que en el futuro España será uno de los países con mayores retos para enfrentar el envejecimiento de su población, ya que está previsto que para el año 2050, el 33% de las personas serán mayores de 65 años, 4 puntos porcentuales por arriba de la media de la Unión Europea que se situará en el 29% [3]. La prevalencia de diabetes en España se sitúa también por encima de la UE [4].

Las tecnologías de la información y la comunicación (TIC) suponen una gran oportunidad en el desarrollo de nuevos sistemas y soluciones que permitan en líneas generales aumentar la calidad de vida de las personas con discapacidad visual. En este sentido, la visión por computador puede ser de gran ayuda para mejorar el día a día de estas personas. En concreto, el análisis facial puede servir para extraer información muy útil y relevante con el objetivo de ayudar a las personas con discapacidad visual en varias de sus tareas diarias dotándoles de un mayor grado de autonomía y seguridad.

El reconocimiento facial ha recibido muchas mejoras en los últimos años y hoy en día se acerca a la perfección. Los avances en el reconocimiento facial no han sido ajenos a las personas con discapacidad. Por ejemplo, recientemente se ha

A. Fernández, Fundación CTIC Centro Tecnológico, Asturias, España, alberto.fernandez@fundacionctic.org
J. L. Carús, Fundación CTIC Centro Tecnológico, Asturias, España, juanluis.carus@fundacionctic.org
R. Usamentiaga, Universidad de Oviedo, Asturias, España rusamentiaga@uniovi.es
R. Casado, Treelogic Centro Tecnológico, Asturias, España, ruben.casado@treelogic.com

presentado un bastón inteligente para ciegos que utiliza reconocimiento facial [5]. El bastón viene equipado con un sistema de reconocimiento facial, GPS y Bluetooth. Al divisar la cara de cualquier conocido o amigo cuya foto está almacenada en la tarjeta SD del bastón, este vibrará y dará, a través de un auricular Bluetooth, las instrucciones necesarias para llegar hasta esta persona. El sistema funciona con cualquier persona que se encuentre a 10 metros o menos. Además, gracias al GPS, el usuario recibirá instrucciones para llegar a donde quiera, como con cualquier navegador GPS.

Sin embargo, además de realizar la tarea de reconocimiento, hoy en día los sistemas biométricos tienen que lidiar con otro tipo de problemas, como el spoofing. En términos de seguridad de redes, este término hace referencia al uso de técnicas a través de las cuales un atacante, generalmente con usos maliciosos, se hace pasar por una entidad distinta a través de la falsificación de los datos en una comunicación.

Los sistemas de reconocimiento facial son vulnerables a los ataques de tipo spoofing. Esto puede ocurrir cuando una persona presenta una fotografía de la persona deseada a la cámara en lugar de la suya propia. Esto presenta un gran problema, pues es relativamente fácil y sencillo hacerse con una fotografía para ser usado a posteriori. Creemos que este es un tema realmente importante en las personas que presentan discapacidad visual.

Basado en las premisas anteriores, el objetivo del presente artículo es el de proponer, construir y validar una arquitectura basada en reconocimiento facial y sistema anti-spoofing que pueda ser integrado tanto en un videoportero como una aplicación móvil. De esta manera, se quiere dotar a los ciegos y disminuidos visuales de un instrumento o herramienta que le permita en un último fin mejorar la calidad de vida y aumente tanto su seguridad como la sensación de la misma en su hogar o cuando tenga interacciones con otras personas. La arquitectura propuesta se ha validado con usuarios reales y en un entorno real simulando las mismas condiciones que se podrían dar tanto en las imágenes capturadas por un videoportero como las imágenes capturadas por una persona con discapacidad visual por medio de su dispositivo móvil. Las contribuciones se comentan a continuación:

En primer lugar se propone un algoritmo para la normalización de la cara del usuario robusto en cuanto a rotaciones y desajustes en el algoritmo de detección facial. Está demostrado que un algoritmo robusto de normalización puede aumentar considerablemente la tasa de acierto en un algoritmo de detección facial.

En segundo lugar se propone usar un algoritmo de detección de spoofing con el objetivo de aumentar la seguridad del sistema facial a reconocer. Para ello el algoritmo se basa en el análisis de texturas pues este tipo de algoritmos han presentado una gran alta tasa de acierto con la ventaja de haber aplicado el algoritmo de normalización de la cara en el paso previo.

En tercer lugar se ha diseñado e implementado una arquitectura que está especialmente pensada para ser ejecutada en dispositivos con reducidas capacidades de cómputo. El algoritmo ha sido diseñado e implementado en C++ y ha sido portado a la plataforma Android mediante JNI para la comunicación con los algoritmos de visión artificial.

Por último comentar que la arquitectura se complementa con una herramienta de "text to speech" para que la persona con discapacidad obtenga la información final del sistema: de quien se trata y si se trata efectivamente de una persona real o por el contrario se trata de una falsificación.

II. ESTADO DEL ARTE

A. Reconocimiento facial orientado a la discapacidad visual

El problema del reconocimiento facial adaptado a las personas con discapacidad visual ha sido investigado en sus diferentes formas. A continuación se resumen los trabajos más importantes, indicando para cada uno de ellos las características más importantes que han ido motivando el desarrollo de la arquitectura aquí propuesta.

En [6] se presenta un sistema de reconocimiento facial en dispositivos móviles para discapacitados visuales, pero se centra principalmente en reuniones con lo que aspectos como el campo visual capturado por el dispositivo móvil centran gran parte de la temática. En [7] se desarrolló un sistema de reconocimiento facial basado en Local Binary Pattern (LBP) [8]. Compararon este descriptor con otras alternativas (Local Ternary Pattern [9] o Histogram of Gradients [10]) y llegaron a la conclusión que el rendimiento de LBP es un poco superior, su coste computacional es menor y la representación de la información es más compacta. Y como se ha comentado anteriormente, en [5] se ha desarrollado un sistema de reconocimiento facial integrado en un bastón.

En ninguno de estos métodos se lleva a cabo la detección de spoofing, haciendo que el sistema tenga una vulnerabilidad alta ante este tipo de ataques. Creemos que es un punto muy importante sobre todo en personas con discapacidad visual. Además, ninguna de las alternativas antes comentadas está orientada a los videoporteros.

B. Detección de Spoofing

Como en ninguno de los casos anteriores se ha estudiado la detección de spoofing para ayudar a personas con discapacidad visual, procederemos a comentar los resultados más significativos en lo que a la detección de spoofing se refiere.

Existen muchos métodos diferentes para la detección de spoofing. Sin embargo, uno de los factores clave en una aplicación que debe funcionar en tiempo real y en un dispositivo embebido es que el método sea computacionalmente ligero. La mayoría de los algoritmos propuestos o bien son muy complejos y por tanto no son aptos para entornos reales, o bien no usan imágenes convencionales (por ejemplo multi-espectrales o termográficas) [11]. Los algoritmos basados en el análisis de micro-texturas ofrecen un buen resultado con un coste computacional relativamente bajo. En [11], se ha aplicado el algoritmo LBP a la detección de spoofing aplicando dicho operador a diferentes escalas. En [12], se analizan también diversas variantes del operador LBP para la detección de spoofing. En [13], también aplican el operador LBP junto con Máquinas de Soporte Vectorial (SVM) para la detección de spoofing. Tanto en [12] como en

[13], utilizan una división de la región facial en 9 regiones (3x3 regiones), pues comentan que es el número de divisiones que mejores resultados ha generado. Después aplican el operador LBP en cada una de las regiones y concatenan el histograma generado. Este factor será tenido en cuenta en nuestra arquitectura a la hora de generar el histograma LBP.

III. METODOLOGÍA

A continuación se comentan las principales fases del algoritmo. En los siguientes puntos de esta sección se irán detallando. Los principales pasos del algoritmo se pueden ver en la Fig. 1.

En primer lugar, este trabajo empieza con la localización de la cara de la persona en la imagen. Para esta tarea se usa un algoritmo basado en Viola & Jones [14], pero ampliado y modificado para lograr un algoritmo más robusto.

En segundo lugar y una vez localizada la cara en la imagen, un pre-procesamiento es necesario para lidiar con la pose, rotación e imprecisiones que provienen de la detección facial previa. Es por ello que se aplica un algoritmo de normalización facial. Nosotros hemos propuesto previamente este algoritmo de normalización facial para la detección robusta de gafas en personas en imágenes reales [15]. También lo hemos usado para la obtención de variables fisiológicas (frecuencia cardiaca y frecuencia respiratoria) obteniendo en ambos casos muy buenos resultados [16].

Una vez la región facial ha sido normalizada, LBP se emplea para obtener el conjunto de características que describirán la cara. Con el foco puesto en la complejidad computacional del algoritmo, el conjunto de características extraídas mediante el operador LBP ha sido cuidadosamente seleccionado, pues está produciendo muy buenos resultados tanto en la detección facial como en la detección de spoofing. Es por ello, que una vez extraído el conjunto de características faciales mediante el operador LBP, éste sirve como entrada en la etapa de clasificación a dos máquinas de soporte vectorial (SVM). SVM se aplica para clasificar el histograma obtenido de la región normalizada de la cara. La salida de los dos clasificadores SVM es por un lado la identificación facial y si dicha detección se corresponde con una falsificación o por el contrario es una cara real. SVMs son una técnica muy usada para la clasificación de los datos y ha sido propuesta en muchas ocasiones con temas relacionados con las tareas de reconocimiento de patrones, como por ejemplo reconocimiento facial [17]. Se usó LIBSVM para las tareas de entrenamiento y testeo de SVMs [18].

El sistema ha sido validado en un entorno real por medio de una aplicación desarrollada para la plataforma Android mediante llamadas JNI a los algoritmos de visión por computador que están desarrollados enteramente en C++. El motivo de desarrollar los algoritmos de visión por computador en C++ es debido a que es un lenguaje altamente compatible y recomendado para ser ejecutado en dispositivos con reducidas capacidades de cómputo.

La arquitectura ha sido desarrollada y validada en un videoportero y también se han portado los algoritmos al dispositivo móvil de una persona con discapacidad visual. Para validar la arquitectura en un videoportero ésta ha sido desarrollada meticulosamente teniendo en cuenta las posiciones, orientaciones y condiciones que presentan los videoporteros con el objetivo de que la captura de imágenes para validar el sistema fueran lo más fidedignas posibles a un entorno real. El entorno seleccionado para validar el sistema se encuentra a la puerta del Centro Tecnológico donde realizamos nuestra actividad investigadora.

Por otro lado, para validar la arquitectura en una aplicación móvil, la aplicación fue desarrollada teniendo en cuenta las características y recomendaciones propuestas por otros autores, además de características que faciliten el encontrar la cara del usuario para permitir tomar las fotografías de la persona con la que está interactuando la persona con discapacidad visual.

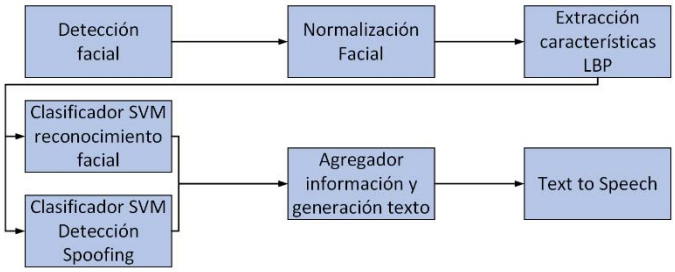

Figura 1. Principales pasos del algoritmo propuesto para la identificación facial, detección de spoofing y conversión a voz de la información agregada.

A. Detección facial

El detector facial propuesto por Viola & Jones [14] es comúnmente usado para realizar el seguimiento de la cara a lo largo del tiempo. Este detector presenta buenos resultados cuando la cara está prácticamente frontal, sin embargo, no puede lidiar cuando se presentan rotaciones a partir de 45 grados tanto en el eje vertical como en el eje horizontal. Es además un detector facial, es decir, no realiza el seguimiento de la cara a lo largo del tiempo (no es el algoritmo más adecuado para hacer el seguimiento o tracking facial). Además, es común que se encuentren falsos positivos o múltiples detecciones solapadas ante una misma cara por la forma que tiene de proceder dicho algoritmo. Por lo tanto, es necesario un algoritmo robusto para realizar un seguimiento de la cara a lo largo del tiempo. Con el objetivo de solucionar las dificultades anteriormente comentadas, se propone combinar tres detectores faciales en el primer frame o cuando se pierde una cara y se está buscando una nueva aparición. Cada detector está basado en el algoritmo de Viola & Jones antes comentado. De esta manera, se entrenan tres detectores: un detector frontal, un detector del perfil izquierdo y un detector del perfil derecho.

El resultado de aplicar el algoritmo de Viola & Jones a una imagen se corresponde con detecciones faciales en forma de rectángulos. En caso de que varios de los detectores faciales detecten una cara, la mínima región que recoge a ambos rectángulos es creada. El siguiente paso del algoritmo es el que trata en profundad estas posibles detecciones.

B. Seguimiento y normalización facial

El siguiente paso se corresponde con el seguimiento de la cara (tracking) y la normalización de la región facial. El algoritmo de Viola & Jones solamente encuentra las posiciones de las caras en la imagen que se corresponden con rectángulos. Esto no es suficientemente preciso, pues se pueden incluir en el rectángulo de detección píxeles que no se corresponden con

región facial, sobre todo en las esquinas del rectángulo. Por lo tanto, un algoritmo de normalización es necesario.

Es bastante común en algoritmos de procesamiento facial usar un detector de piel para construir un mapa de piel que después servirá para normalizar la región facial [19]. Sin embargo, estos detectores no son muy robustos a los cambios de iluminación. También es muy común el uso de la técnica Active Appearance Model (AAM) [20] para detectar los principales puntos característicos de la cara. Sin embargo, un detector basado en AAM no es muy robusto, sobre todo en imágenes con baja resolución [21].

Es por ello que en la presente publicación, se propone: (1) un algoritmo de normalización facial y (2) un algoritmo de seguimiento facial para solucionar los problemas antes comentados.

El algoritmo de normalización facial es necesario para lidiar con la pose, rotación, escala e inexactitudes de la cara localizada. Este algoritmo está basado en el algoritmo propuesto en [15] pero modificado para obtener toda la región facial necesaria para después proceder tanto al reconocimiento facial como a la detección de spoofing. A continuación se resumen los principales puntos del algoritmo de normalización.

1) Algoritmo de normalización facial

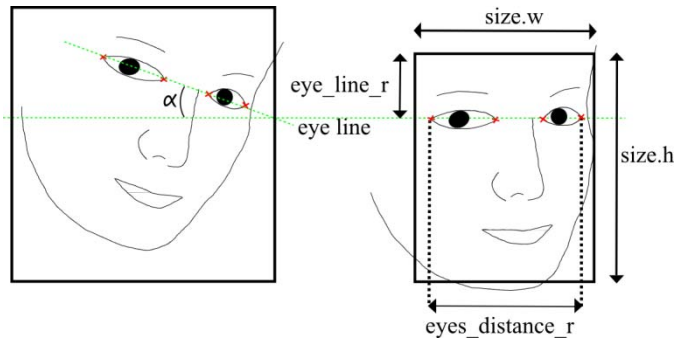

Figura 2. El algoritmo de normalización tiene en cuenta la rotación y realiza una corrección y normalización de la región facial.

A continuación se comentan los principales pasos del algoritmo. Una vez que la cara ha sido detectada, se aplica un detector robusto de características faciales basado en "Deformable Parts Models" (DPM) [21]. 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. 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. Esto puede verse en la Fig. 2. A continuación las caras son rotadas y alineadas de manera que los ojos siempre se encuentran en las mismas coordenadas en la imagen final. A continuación 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. Mediante este algoritmo de normalización hemos comprobado que la tasa de reconocimiento puede incrementarse significativamente. Aplicándolo al caso de reconocimiento de gafas por ejemplo, hemos obtenido una mejora del 1.55%. Como se ha comentado anteriormente, todos los detalles del algoritmo pueden verse en [15], aquí únicamente se muestran de manera resumida.

2) Algoritmo de seguimiento facial

Para realizar el seguimiento de las caras a lo largo del tiempo, se emplea un algoritmo de tracking muy robusto y recién publicado que está obteniendo muy buenos resultados [22]. Para el primer frame se emplea un rectángulo que envuelve el conjunto de características faciales devuelto por el detector. En caso de haber varias caras, se realiza el seguimiento de la cara que ocupe mayor espacio en la imagen, ya que se supone que es la principal. Dicho algoritmo de tracking realiza el seguimiento a lo largo del tiempo. La salida del algoritmo de tracking se utiliza después en el resto de frames y se aplica el algoritmo de normalización antes comentado.

C. Extracción de características mediante LBP

En primer lugar comentaremos en qué consiste el operador LBP y en segundo lugar comentaremos como se ha aplicado dicho operador para la extracción de las características faciales tanto para la identificación facial como para la detección de spoofing.

1) Operador LBP y mejoras incorporadas a dicho operador

El operador Local Binary Pattern (LBP) [23] es un tipo de operador que se suele utilizar para temas de clasificación. Es un operador muy potente en todo lo relacionado con el tema de clasificación de texturas. Dicho operador fue introducido en 1996 como un método para sintetizar la estructura del nivel de grises en imágenes. Dicho operador tiene en cuenta un vecindario local de píxeles alrededor de un píxel central. A continuación 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 3x3, con 8 bits para codificar los valores binarios, puesto que son precisamente 8 los píxeles vecinos. Formalmente el operador LBP presenta la siguiente forma:

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

donde en este caso p recorre los 8 vecinos con respecto al píxel central c,g_c y g_p son los valores del nivel de gris en c y p y:

$$s(x) = \begin{cases} 1, & si\ x \geq 0 \\ 0, & en\ otro\ caso \end{cases} \quad (2)$$

El proceso de codificación del operador original se ilustra en la Fig. 3.

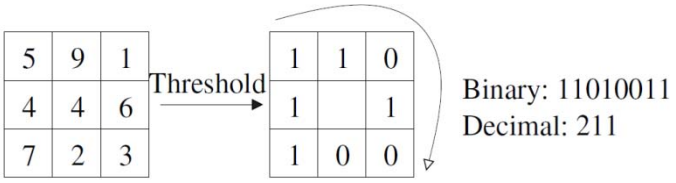

Figura 3. Operador LBP original (básico).

El operador fue posteriormente extendido para incorporar vecindarios de píxeles de diferentes tamaños, haciendo por tanto posible lidiar con las texturas a diferentes escalas [24].

Este hecho se denota por *(P,R)* donde *P* representa el número de puntos de muestreo (es decir el número de vecinos equi-espaciados alrededor del píxel central) y *R* representa el radio del vecindario. Cuando las posiciones de los puntos de muestreo no se corresponden con posiciones enteras en la imagen, se utiliza la técnica de interpolación bilineal. En la Fig. 4 se puede ver un ejemplo de un radio circular del tipo *(8,2)*. La implementación de este LBP circular ($LBP_{P,R}$) toma la siguiente forma:

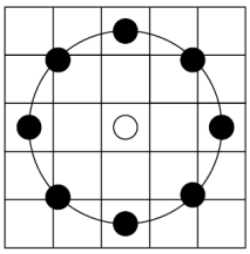

Figura 4. El operador circular (8,2). Los valores de los píxeles son bilinealmente interpolados siempre que el punto de muestreo no coincida en el centro de un píxel.

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

Otra extensión al operador original define como los patrones uniformes [24]. Un patrón LBP es uniforme cuando contiene como mucho dos transiciones a nivel de bit de 0 a 1 o viceversa visto como una cadena circular de bits cara uno de los patrones. Por ejemplo, los patrones 00000000, 00011110 and 10000011 son uniformes. El concepto de uniformidad es un concepto muy importante en la metodología de LBP, pues representa información estructural de primitivas como pueden ser los bordes o las esquinas en la imagen. A pesar de que únicamente hay 58 patrones uniformes de los 256 posibles patrones considerando un vecindario de 8 píxeles, cerca del 90% de los patrones en la región facial son uniformes. Es por ello que los patrones uniformes se pueden usar para reducir considerablemente la dimensionalidad de los datos manejados sin perder excesiva información. Para referirse a los patrones uniformes, se usa la siguiente notación: $LBP_{P,R}^{u2}$.

2) Extracción de características mediante el operador LBP para identificación facial y detección de spoofing

Con el objetivo de conseguir un algoritmo eficiente, robusto y computacionalmente ligero, se investigaron qué diferentes descriptores y operadores se podrían aplicar para representar de manera eficiente la región facial. Tras realizar una profunda investigación, analizar el estado del arte y hacer unas pruebas preliminares, se llegó a la conclusión que el operador LBP produce resultados excelentes tanto en el reconocimiento facial como en la detección de spoofing. Además, es un operador computacionalmente ligero. Por último y no por ello menos importante, sólo se computa una vez el operador, pues la información extraída de la región facial por medio del operador LBP se comparte por ambos módulos.

Después de etiquetar la imagen aplicándole el operador LBP, un histograma de esta imagen etiquetada $f_l(x,y)$ se puede definir como:

$$H_i = \sum_{x,y} I\{f_l(x,y) = i\}, i = 0, \dots, n-1 \quad (4)$$

donde *n* representa el número de valores diferentes producidos por el operador LBP y:

$$I\{A\} = \begin{cases} 1, & si\ A = verdadero \\ 0, & en\ otro\ caso \end{cases} \quad (5)$$

Para una representación eficiente de la información facial, las características extraídas mediante el operador LBP deberían disponer de información espacial. Es por ello que la imagen se ha dividido en m regiones $\{R_0, R_1, \dots, R_{m-1}\}$. De esta manera, el histograma básico descrito arriba puede extenderse en lo que se conoce como un "histograma mejorado espacialmente" [8], el cual es capaz de codificar tanto la apariencia y las relaciones espaciales de las distintas regiones faciales. Este nuevo histograma se define como:

$$H_{i,j} = \sum_{x,y} I\{f_l(x,y) = i\} I\{(x,y) \in R_j\} \quad (6)$$

donde *i=0,....,n-1*, *j=0,...,m-1*. El histograma por tanto sirve para describir la región normalizada de la cara teniendo en cuenta tres niveles de localidad: las etiquetas del histograma contienen información sobre los patrones a nivel de píxel, estas etiquetas son tenidas en cuenta para crear histogramas en regiones y en un último nivel, todos estos histogramas se concatenan para conseguir un histograma global.

D. Agregador información y generación de audio

Con el objetivo de minimizar la cantidad de información que se le proporciona al usuario y que ésta sea lo más fidedigna posible, los resultados se proporcionan al usuario al analizar un conjunto *N* de frames determinados. Por lo tanto, el sistema analiza los últimos N frames y en de que el algoritmo de tracking detecte la presencia de una cara, el sistema proporciona al usuario la información correspondiente a la detección facial y al sistema de spoofing.

Tras realizar varias pruebas y también tras analizar varias publicaciones [7], se ha llegado a la conclusión que un número reducido de frames (N = 5) es suficiente para minimizar la cantidad de información proporcionada al usuario y además, proporcionarle una información rápida, precisa y fiable al usuario.

Para el caso de uso de la aplicación móvil (ver Sección V), se proporciona además, información acústica acerca del estado del tracking con el objetivo de ayudar al usuario a "encontrar" a la persona que está hablando.

Para el caso del videoportero (ver Sección V) esto no es necesario, pues la detección facial del sistema no depende de la persona con discapacidad visual y no suele haber problemas para localizar y hacer el seguimiento de la cara en las imágenes capturadas por el videoportero.

IV. IMPLEMENTACIÓN

El sistema aquí propuesto ha sido desarrollado en C++ y haciendo uso de la librería de OpenCV. OpenCV (Open Source Computer Vision Library) 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 2.4.11 que es la que ha sido usada en la implementación del sistema.

Como herramienta para entrenar los modelos para el reconocimiento facial y detección de spoofing basados en SVM se hizo uso de la librería LibSVM [18].

El sistema ha sido construido de manera modular. Cada uno de los módulos que componen el sistema son los descritos anteriormente (ver Sección Metodología). Tanto la modularidad como el hecho de que el sistema estuviera desarrollado enteramente en C++ permitieron adaptar la el sistema a dos casos de uso diferentes.

Por un lado el sistema se adaptó para el caso de uso de los videoporteros.

El segundo caso de uso fue el de portar el sistema y los algoritmos a la plataforma Android para desarrollar una aplicación móvil que sirviera a las personas con discapacidad visual en su interacción diaria con otras personas.

A continuación comentamos de manera sucinta ambas implementaciones y en la Sección Casos de Estudio se comentan los resultados y detalles para cada uno de los dos casos de estudio.

A. Implementación en caso de uso de los videoporteros

En un primer lugar, el sistema se implementó bajo el sistema operativo Windows, haciendo uso de la librería OpenCV antes comentada y enteramente en C++. El objetivo es el de desarrollar una aplicación que pueda ser incluida en un videoportero. Puesto que el lenguaje C++ es uno de los más portables y con mejores rendimientos en cuanto a capacidad y velocidad de computación, la implementación desarrollada es perfectamente portable a un dispositivo embebido sin realizar apenas modificaciones al sistema.

B. Implementación en caso de uso de aplicación móvil

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 que pudiera realizar la autenticación facial y la detección de spoofing orientada a gente con discapacidad visual. De esta manera, la aplicación ayudaría a estas personas a sus interacciones diarias con otra gente.

V. CASOS DE ESTUDIO Y RESULTADOS OBTENIDOS

En esta Sección se muestran dos casos de estudio, donde la herramienta desarrollada sirve de soporte para la creación de dos aplicaciones diferentes.

En la primera de ellas, se comentan los detalles para adaptar el sistema a la identificación facial y detección de spoofing en un videoportero, teniendo en cuenta las características que esto implica.

En segundo lugar, se portaron los algoritmos construidos a la plataforma Android con el objetivo de construir una aplicación móvil que sirviera como soporte a las personas con discapacidad visual en su interacción con otras personas.

A. Caso de estudio 1: videoportero

En primer lugar, con el objetivo de validar el sistema en un entorno lo más realista posible, se decidió realizar las pruebas a la entrada del centro tecnológico donde realizamos nuestra actividad investigadora. Además, otro factor que ha sido tenido en cuenta para una correcta validación del sistema, fue la ubicación y tipo de la cámara a instalar. Las imágenes capturadas tienen que ser lo más realistas posibles pues son las que sirven tanto para entrenar como para validar el sistema. Para ello, se realizó un estudio en el que se recogieron dos factores principales: altura del videoportero, distancia a la que se suelen ubicar las personas (ver Fig. 5).

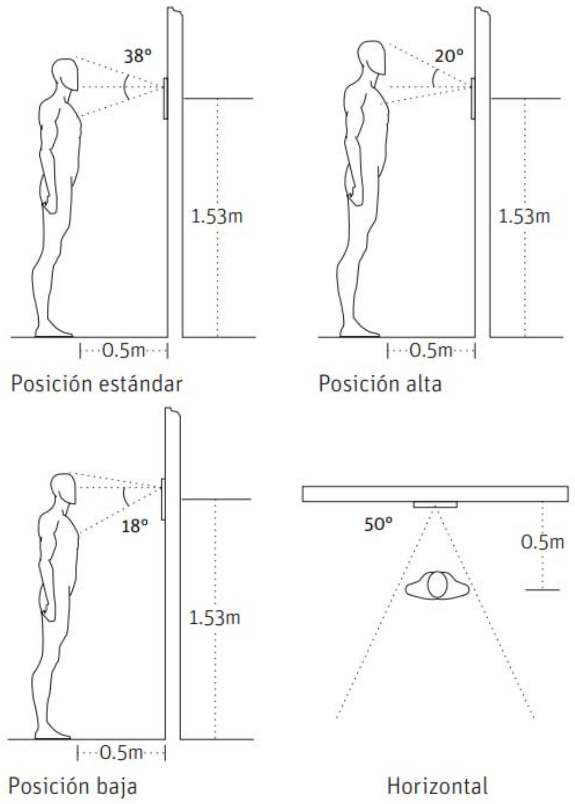

Figura 5. Alcance visual (posición y distancia) habituales de los usuarios respecto del videoportero

Como se puede apreciar, los usuarios se suelen colocar a una distancia de unos 0.5 metros. El videoportero ha sido colocado a una altura de unos 1.53m.

Puesto que el histograma LBP es calculado sobre cada división en la imagen y posteriormente concatenado, un número de divisiones pequeño consigue un histograma más pequeño pero tiene el inconveniente de que se pierde información espacial.

Es por ello que se realizaron diversas pruebas con el objetivo de ver el número de divisiones que mejores resultados aportaba tanto al reconocimiento facial como a la detección de spoofing así como otros parámetros del operador LBP. La mejor configuración de parámetros fue $LBP_{8,1}^{u2}$ (patrones uniformes, R = 1, P = 8) con SVM usando un kernel del tipo RBF, que está en consonancia con otras investigaciones relacionadas [12],[13],[25]. Un diagrama donde se muestra el proceso puede verse en la Fig. 6.

Para establecer los parámetros del algoritmo se hicieron uso

de las bases de datos de imágenes más comunes para la comparación de los algoritmos correspondientes al reconocimiento facial y a la detección de spoofing a partir de imágenes: Labeled Faces in the Wild (LFW) [26] y NUAA Photograph Imposter Dabase [27].

Con el objetivo de validar la arquitectura desarrollada, se instaló el prototipo a la entrada del Centro Tecnológico donde realizamos nuestra labor investigadora. Para ello se tuvieron en cuenta los detalles llevados a cabo por otros estudios, el alcance visual (posición y distancia) habituales de los usuarios respecto del videoportero.

Con esta configuración, lo habitual es que la cara detectada en la imagen tenga una resolución de unos 64x64 píxeles aproximadamente. Es por ello que las pruebas realizadas con las imágenes de las bases de datos antes comentadas se redimensionen a dicha resolución una vez la cara ha sido detectada (ver Fig. 6).

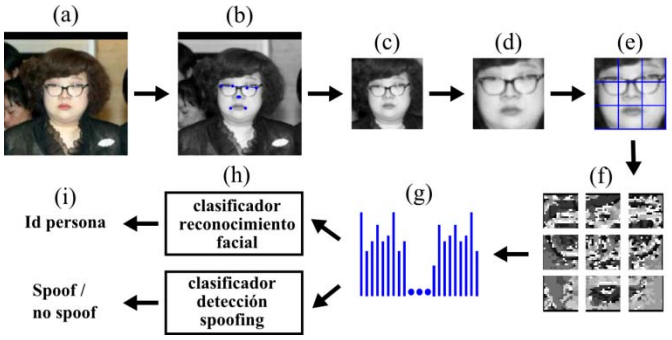

Figura 6. La imagen original (a) se convierte a escala de grises y se detectan los puntos característicos (b). A continuación se rota para alinearla en función del ángulo (c). Después obtenemos la región normalizada (64x64) de la cara (d). Aplicamos 3x3 divisiones a la región facial (e) y construimos la imagen $LBP_{8,1}^{u2}$ (f). Se construye el histograma dadas estas regiones para formar el vector de características (g). Por último se clasifica este vector usando ambos clasificadores (h) para obtener las respuestas finales (i).

Una vez seleccionados los parámetros del algoritmo, se hicieron pruebas durante 5 días consecutivos y con diferentes condiciones de iluminación (por la mañana y por la tarde). Cabe decir, que aunque la entrada del edificio no recibe luz directa es un sitio bastante iluminado. En total se realizaron pruebas con 25 usuarios.

Figura 7. Para las pruebas se seleccionó un Lenovo ThinkPad X200 Tablet que se posicionó encima del videoportero con la webcam en la parte inferior del mismo. Se puede ver un detalle de la imagen capturada.

Consideramos este número como un número suficiente de personas con las que más a menudo interaccione una persona con discapacidad visual (familiares, amigos más próximos, etc). La disposición del equipo y su ubicación puede verse en la imagen de la Fig. 7.

Para llevar a cabo las pruebas se seleccionó un Lenovo ThinkPad X200. Para la captura de imágenes se utilizó la webcam incorporada al equipo cuya resolución es de 640x480 píxeles.

Para probar el sistema de spoofing, se imprimieron fotografías de cada uno de los 25 usuarios. Para cada usuario se validaron ambos algoritmos (reconocimiento facial y detección de spoofing). Para ello el usuario se debe situar delante del videoportero. Con el objetivo de que el algoritmo obtenga una buena detección facial, el sistema de tracking facial procesa frames hasta que localiza uno donde las características faciales detectadas estén simétricas (dentro de unos umbrales). De esta manera garantizamos que la detección es frontal y en condiciones para que los algoritmos operen con propiedad.

En caso de que el algoritmo de tracking facial no detecte una detección frontal por el mecanismo antes comentado, se le proporciona al usuario un comando de voz para que se sitúe frontalmente al videoportero. A continuación se resume en forma de tabla los principales resultados a los que se ha llegado.

TABLA I. RESULTADOS VIDEOPORTERO EN ENTORNO REAL.

ALGORITMOS	D = 1	D = 2	D = 3	D = 4	D = 5
RECONOCIMIENTO FACIAL	97%	98%	97%	94%	97%
DETECCIÓN SPOOFING	86%	86%	89%	84%	90%

Como se desprende de los resultados, el reconocimiento facial presenta unos resultados bastante buenos. El sistema de detección de spoofing presenta unos resultados algo peores, pero aún así son unos resultados bastante buenos. Cabe destacar en el cuarto día de pruebas (D = 4) el tiempo fue bastante soleado. Creemos pudo afectar al rendimiento de los algoritmos.

B. Caso de estudio 2: Aplicación móvil

Con el objetivo de sacar el máximo partido a la arquitectura, se decidió portar los algoritmos a la plataforma Android. De esta manera, el usuario con discapacidad visual dispone de una herramienta portable y usable para sus interacciones.

En caso de la aplicación móvil se han tenido en cuenta diversos factores que no ha sido necesario contemplar en el caso de uso expuesto anteriormente y que se comentan a continuación. En este caso, es necesario tener en cuenta que el dispositivo móvil y por tanto las imágenes capturadas son tomadas por personas con discapacidad y como consecuencia, muchas imágenes pueden presentar ruido, desenfoque, borrosidad y diferentes condiciones de iluminación. En

segundo lugar, es necesario aportar al usuario información auditiva acerca del estado de la aplicación: si ha detectado a una persona, si ha perdido el tracking de dicha persona, si la ha conseguido identificar y por último si la identificación ha sido real con el objetivo de proporcionarle un "feedback" con el estado de la aplicación, pero sin abrumarle con un exceso de información innecesaria.

El principal punto a tratar es que la persona con discapacidad visual no sabe a donde está enfocando su dispositivo. Para solucionar este factor, proponemos dos medidas: (1) guía del estado del tracking por medio de pitidos sonoros; y (2) analizar varios frames antes de predecir un resultado. A continuación comentamos los dos puntos.

En el primero de ellos, en caso de que el usuario de la aplicación detecte una cara, se producirá un pitido. De esta manera, el usuario mantendrá el móvil en esa posición para "mantener" el tracking y que el algoritmo pueda funcionar correctamente.

Si se pierde el tracking antes de que el algoritmo haya terminado, se producirán dos pitidos que indicarán al usuario que el algoritmo ha finalizado sin concluir una respuesta. En cambio, si el algoritmo consigue dar una respuesta al usuario es que ha procesado un número suficiente de frames N, que se corresponde con el segundo de los puntos indicado antes. Tras realizar varias pruebas y consultar la bibliografía relacionada, se ha establecido que el número de frames *N* que se deben procesar antes de proporcionar un resultado es de N = 5. Se ha establecido este valor con el objetivo de: (1) proporcionar unos resultados más robustos; y (2) proporcionar un feedback rápido para mejorar la fluidez en la interacción. Esto mejorará los resultados en los casos en los que las imágenes estén borrosas o presenten ruido.

Por lo tanto para cada uno de los clasificadores (identificador facial y spoofing) se obtienen, para cada imagen procesada tanto el identificador de la clase predicha como el valor de confidencia para esa clase.

Es por ello que para un determinado número de frames N, antes de obtener el valor final, se calcula la clase ganadora, y por tanto el valor a predecir en función de éstos valores de confidencia.

A continuación se adjunta una tabla donde se recogen los tiempos de ejecución del algoritmo. Para ello se han seleccionado dispositivos móviles y tablets de gama media/baja que disponemos en nuestro laboratorio, pues el objetivo del sistema es que los algoritmos se ejecuten en dispositivos con baja potencia computacional.

TABLA II. TIEMPOS DE EJECUCIÓN PARA VARIOS DISPOSITIVOS DE GAMA MEDIA/BAJA.

DISPOSITIVO MOVIL	FPS	TPO EJECUCIÓN SEC PARA N=5	TPO DE EJECUCIÓN ACEPTABLE
HTC DESIRE X	1.82	2.75	SI
LG OPTIMUS L2	1.86	2.69	SI
WOXTER (TABLET)	0.88	5.68	NO
SAMSUNG GALAXY Y	0.76	658	NO

Actualmente se está usando el valor promedio de los valores de confidencia, aunque el sistema podría contemplar otras alternativas. Entendemos que un tiempo superior a 3 segundos representaría un tiempo inaceptable para un correcto funcionamiento de la aplicación móvil.

Hay que recordar que la aplicación tiene como objetivo ayudar a mejorar a las personas con discapacidad visual en su interacción. Mantener durante más de 3 segundos el móvil en una posición más o menos fija es complicado para un usuario con estas características. Los dos modelos que cumplen estas restricciones son dos modelos que tienen unas especificaciones similares. Son móviles con un procesador de más o menos 1Ghz de doble núcleo y unos 768MB de memoria RAM. En la actualidad prácticamente cualquier móvil duplica en prestaciones a los dispositivos antes comentados. Es por ello que creemos que la aplicación puede ser perfectamente usable en cuanto a características y requerimientos técnicos.

A modo de ejemplo, la aplicación fue instalada en una tablet Edison 2 3G Quad Core y tiene un rendimiento de unos 4.33fps. Esto significa que en poco más de 1 segundo, el usuario tiene el resultado requerido.

En la imagen 8 se puede ver la captura de pantalla de varias ejecuciones que se corresponden con imágenes reales (parte izquierda de la imagen) y con fotografías (parte derecha de la imagen).

Figura 8. Capturas de pantalla de la aplicación móvil. Parte izquierda: imágenes reales. Parte derecha: imágenes impresas (no reales).

VI. CONCLUSIONES

En el presente trabajo hemos presentado una arquitectura para la identificación facial y detección de spoofing orientado a las personas con discapacidad visual. La arquitectura ha sido diseñada y desarrollada con el objetivo de conseguir un resultado robusto y computacionalmente ligero que pudiera ser embebido en elementos con una capacidad de cómputo moderada, como pudiera ser un videoportero o el dispositivo móvil ambos de personas con discapacidad visual.

El sistema ha sido probado y validado en entornos y condiciones reales obteniendo unos resultados muy satisfactorios. Además comentar que se han tenido en cuenta conclusiones y elementos clave obtenidos de otras publicaciones relevantes con el objetivo de obtener unos resultados más robustos y que la arquitectura final fuera lo más usable y adaptada a las personas con discapacidad visual.

Como trabajo futuro comentar dos aspectos principalmente. Uno de ellos es mejorar el sistema para hacerlo más robusto ante diferentes condiciones de iluminación. En [9] se comenta un algoritmo de pre-procesamiento para mejorar el reconocimiento facial cuando las condiciones de iluminación son adversas. Habría que ver cómo influye dicha etapa de pre-procesamiento en el algoritmo de detección de spoofing, pues podría disminuir las texturas en las imágenes faciales que hacen diferenciar una imagen real de otra falsa. El segundo de los factores a mejorar radica en mejorar la tasa de reconocimiento en lo que al algoritmo de detección de spoofing se refiere.

En conclusión, creemos que las personas con discapacidad visual pueden verse beneficiadas por la presente solución con el objetivo final de mejorar su calidad de vida.

REFERENCIAS

[1] Gómez-Ulla de Irazazábal, F., & Ondategui-Parra, S. (2012). Informe sobre la ceguera en España.

[2] Vision 2020: The right to sight. World Health Organization (WHO) and International Agency for Blindness Prevention (IAPB). Action Plan (2006-2011).

[3] Portal web del Eurostat, 2011.

[4] Diabetes Atlas. International Diabetes Federation. 2010.

[5] Maidenbaum, S., Hanassy, S., Abboud, S., Buchs, G., Chebat, D. R., Levy-Tzedek, S., & Amedi, A. (2014). The "EyeCane", a new electronic travel aid for the blind: Technology, behavior & swift learning. Restorative neurology and neuroscience, 32(6), 813-824.

[6] C. Kramer, K. M., Hedin, D. S., & Rolkosky, D. J. (2010, August). Smartphone based face recognition tool for the blind. In Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE (pp. 4538-4541). IEEE.

[7] Balduzzi, L., Fusco, G., Odone, F., Dini, S., Mesiti, M., Destrero, A., & Lovato, A. (2010, September). Low-cost face biometry for visually impaired users. In Biometric Measurements and Systems for Security and Medical Applications (BIOMS), 2010 IEEE Workshop on (pp. 45-52). IEEE.

[8] Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 28(12), 2037-2041.

[9] Tan, X., & Triggs, B. (2010). Enhanced local texture feature sets for face recognition under difficult lighting conditions. Image Processing, IEEE Transactions on, 19(6), 1635-1650.

[10] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 1, pp. 886-893). IEEE.

[11] Maatta, J., Hadid, A., & Pietikainen, M. (2011, October). Face spoofing detection from single images using micro-texture analysis. In Biometrics (IJCB), 2011 international joint conference on (pp. 1-7). IEEE.

[12] Chingovska, I., Anjos, A., & Marcel, S. (2012, September). On the effectiveness of local binary patterns in face anti-spoofing. In Biometrics Special Interest Group (BIOSIG), 2012 BIOSIG-Proceedings of the International Conference of the (pp. 1-7). IEEE.

[13] Benlamoudi, A., Samai, D., Ouafi, A., Taleb-Ahmed, A., Bekhouche, S. E., & Hadid, A. Face Spoofing Detection From Single Images Using Active Shape Models with Stasm And LBP.

[14] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-511). IEEE.

[15] Fernández, A., García, R., Usamentiaga, R., & Casado, R. (2015). Glasses detection on real images based on robust alignment. Machine Vision and Applications, 26(4), 519-531.

[16] A. Fernandez, J.L. Carus, R. Usamentiaga, E. Alvarez, R. Casado, "Unobtrusive Health Monitoring System Using Video-Based Physiological Information and Activity Measurements", In IEEE International Conference on Computer, Information, and Telecommunication Systems, CITS 2015, IEEE, vol. 1, no. 1, pp. 100-104, Gijon (Spain), 2015.

[17] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. (2010, June). The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on (pp. 94-101). IEEE.

[18] 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.

[19] de Haan, G., & Jeanne, V. (2013). Robust pulse rate from chrominance-based rPPG. Biomedical Engineering, IEEE Transactions on, 60(10), 2878-2886.

[20] Cootes, T. F., Edwards, G. J., & Taylor, C. J. (2001). Active appearance models. IEEE Transactions on pattern analysis and machine intelligence, 23(6), 681-685.

[21] Uřičář, M., Franc, V., & Hlaváč, V. (2012). Detector of facial landmarks learned by the structured output SVM. VISAPP, 12, 547-556.

[22] Danelljan, M., Häger, G., Khan, F., & Felsberg, M. (2014). Accurate scale estimation for robust visual tracking. In British Machine Vision Conference, Nottingham, September 1-5, 2014. BMVA Press.

[23] 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.

[24] Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 24(7), 971-987.

[25] de Freitas Pereira, T., Anjos, A., De Martino, J. M., & Marcel, S. (2013, June). Can face anti-spoofing countermeasures work in a real world scenario?. In Biometrics (ICB), 2013 International Conference on (pp. 1-8). IEEE.

[26] Huang, G. B., Ramesh, M., Berg, T., & Learned-Miller, E. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments (Vol. 1, No. 2, p. 3). Technical Report 07-49, University of Massachusetts, Amherst.

[27] Tan, X., Li, Y., Liu, J., & Jiang, L. (2010). Face liveness detection from a single image with sparse low rank bilinear discriminative model. In Computer Vision–ECCV 2010 (pp. 504-517). Springer Berlin Heidelberg.

Alberto Fernández received his M.S. degree in Computer Science from University of Oviedo in 2009. Since 2007 he has been developing his professional career at the CTIC Foundation (Centre for the Development of Information and Communication Technologies in Asturias). He has been working in R&D projects. His research interests include both industrial and medical processes.

Juan Luis Carús. PhD in Industrial Engineering (2015) from the Spanish University for Distance Education (UNED) and MSc in Telecomnunications Engineering (2009) from the University of Oviedo. In 2007, he received the Thesis Award "Telefónica España" for the best record in engineering. Since July 2008, he is researcher with the R&D&i Area of CTIC – Technology Centre. His interests include ambient intelligence research projects focusing on signal processing, wearable computing and eHealth.

Rubén Usamentiaga Associate Professor in the Department of Computer Science and Engineering at the University of Oviedo. He received his M.S. and Ph.D. degrees in Computer Science from University of Oviedo in 1999 and 2005, respectively. In recent years he has been working on several projects related to computer vision and industrial systems. His research interests include real-time imaging systems and thermographic applications for industrial processes.

Rubén Casado received a B.Sc. Degree in Computer Science in 2005, a M.Sc. in Computing in 2008 and a PhD in Software Systems in 2013 from University of Oviedo, Spain. He has worked as a researcher and teaching assistant at the University of Oviedo, where he is currently a member of the Software Engineering Research Group. Currently he is the leader of the Big Data research program at Treelogic.