DOI: http://dx.doi.org/10.21501/21454086.2938

Lámpsakos | No. 20 | pp. 30-39| julio-diciembre | 2018 | ISSN: 2145-4086 | Medellín–Colombia

Reconocimiento de Rostros en Tiempo Real sobre Dispositivos Móviles de Bajo Costo

Real Time Face Recognition on Low-Cost Mobile Devices

Alexander Cardona López, MsC.
Universidad Autónoma de Colombia
Bogotá, Colombia
alexander.cardona@fuac.edu.co

Franklin Pineda Torres, MsC.
Universidad Autónoma de Colombia
Bogotá, Colombia
franklin.pineda@fuac.edu.co

(Recibido el 20-09-2018, Aprobado el 02-12-2018, Publicado el 11-01-2019)

Estilo de Citación de Artículo:
A. Cardona, F. Pineda, "Reconocimiento de Rostros en Tiempo Real sobre Dispositivos Móviles de Bajo Costo", Lámpsakos, no. 20, pp 30-39, 2018
DOI: http://dx.doi.org/10.21501/21454086.2938

Resumen: Se prueban algunos de los métodos más conocidos de reconocimiento de rostros, para determinar su utilidad real en la construcción de aplicaciones en tiempo real que puedan ejecutarse sobre un dispositivo móvil de bajo costo. Con este fin, se realiza una breve descripción de los principales algoritmos utilizados en aplicaciones de reconocimiento de rostros y se muestra cómo la fase de detección de rostros es de vital importancia en cuanto a desempeño se refiere en estos dispositivos. Se demuestra además la imposibilidad de realizar el procesamiento de cada frame de un stream de video, a una rata de 30 frames por segundo, con los métodos revisados.

Palabras clave: Análisis de desempeño, Computación móvil, Reconocimiento de rostros.

Abstract: Some of the most recognized face recognition methods are tested to determine their usefulness in the construction of real-time mobile applications, intended to a low-cost mobile market. To this end, a brief description of the main algorithms used in face recognition applications is made. It is shown how face detection phase is vital in terms of performance on these devices. It is also demonstrated the impossibility of performing the processing of each frame of a video stream, which runs at a rate of 30 frames per second, using the considered methods.

Keywords: Face recognition, Mobile computing, Performance analysis.

1. INTRODUCCIÓN

El reconocimiento de rostros en imágenes digitales, tiene como finalidad lograr que una máquina determine sin equívocos la identidad de uno o más rostros dentro de una imagen. Una implementación adecuada del reconocimiento de rostros tiene un amplio espectro de aplicación, ya que reemplazaría la necesidad de otros métodos de identificación, tales como el uso de tarjetas, sensores de radiofrecuencia o la interacción del usuario mediante el empleo de claves.

Por supuesto, debido a su importancia, a lo largo de los años se han propuesto diversos métodos computacionales para realizar dicho reconocimiento, alcanzando en la última década resultados que se pueden asemejar a la identificación de rostros que realizaría una persona. Sin embargo, buena parte de los métodos más reconocidos no están enfocados a su aplicación en video en tiempo real, ni en dispositivos móviles de bajo costo (cuyo precio es de 100 dólares o menos). Dispositivos que no cuentan con capacidades de cómputo, ni con la calidad de cámaras con que cuentan los dispositivos de alta gama. Ambos aspectos importantes para el reconocimiento de rostros sobre imágenes digitales. Mucho menos cuentan con sensores diferentes a la ya tradicional cámara fotográfica. Trabajos como [1] o [2] reportan un éxito superior al 90% en el

reconocimiento de rostros sobre el conjunto de datos Labeled Faces in the Wild (LFW)[3]. Sin embargo, estos trabajos se centran en la precisión de los métodos, más que en los recursos necesarios para su ejecución, por lo que no es claro su desempeño en dispositivos móviles.

Este documento se centra en probar algunos de los algoritmos de reconocimiento de rostros más conocidos, que han mostrado una precisión superior al 85%, para determinar su utilidad real en aplicaciones que hagan uso del stream de video brindado por la cámara de un dispositivo móvil de baja gama. Con este fin, primero se realiza una descripción general del proceso de reconocimiento de rostros y de varios de los algoritmos comúnmente empleados en este proceso. Posteriormente se realizan pruebas sobre el conjunto de datos Labeled Faces in the Wild, con diferentes parámetros de configuración. Finalmente se presentan los resultados y conclusiones obtenidas.

2. REVISIÓN LITERARIA

Es difícil realizar una clasificación de todos los métodos de reconocimiento de rostros, en parte debido a la cantidad de métodos que se han propuesto. En este documento, se describen varios de los métodos más reconocidos, si desea conocer sobre la diversidad de métodos propuestos se sugiere la lectura del survey “A Survey on Face Recognition Techniques” [4]. De forma similar el trabajo [5] menciona algunas de las variaciones recientes que se han realizado a métodos conocidos. El documento [6], explica los métodos más populares antes del año 2009, mientras que el documento [7], se enfoca en el problema de encontrar puntos relevantes en el rostro (landmarks).

Las aplicaciones de reconocimiento de rostro normalmente implementan los siguientes procesos: captura, pre-procesamiento de la imagen, detección de rostros, corrección de postura y el reconocimiento en sí. Cada proceso cumple con una función específica:

- **Captura**: se obtiene la imagen del sensor (normalmente la cámara del dispositivo en aplicaciones móviles) y se almacena en memoria o en disco para su posterior procesamiento. La forma en que se realiza la captura normalmente es dependiente del hardware y del sistema operativo empleado.

- **Pre-procesamiento**: de ser necesario, cada imagen se escala al tamaño óptimo para el algoritmo seleccionado. Se cambia el formato de los pixeles y se mejora la calidad de la imagen mediante la modificación del valor de sus pixeles. Este último caso suele orientarse a la eliminación de ruido o reducir los efectos de una iluminación deficiente.

- **Detección**: si la imagen contiene algo más que rostros o muchos rostros, es necesario ubicar dentro de la imagen la posición y tamaño de cada uno de los rostros, e idealmente su orientación. Esto se debe a que la mayoría de los algoritmos de reconocimiento trabajan sobre la imagen de un único rostro.

- **Corrección de postura:** algunos algoritmos de reconocimiento funcionan mejor si la imagen trabajada contiene rostros en una postura específica o con características geométricas determinadas. En estos casos, puede ser mejor modificar la imagen para "intentar" colocar el rostro en la postura más adecuada para su reconocimiento. Esta fase algunas veces involucra la creación de un modelo tridimensional a partir de la información extraída de la imagen bidimensional.

- **Reconocimiento:** finalmente se busca reconocer a quién corresponde cada rostro de la imagen de entrada. Como es de esperarse, se debe contar con anterioridad con un registro de los rostros sobre los que se va a realizar la identificación.

2.1 Detección de rostros

La detección de rostros es el proceso de identificar las regiones de una imagen que corresponden a rostros. Este proceso produce como resultado una serie de regiones donde existe mayor probabilidad de encontrar un rostro, normalmente regiones rectangulares, aunque existen también propuestas para encontrar otro tipos de regiones que detallen no sólo la posición, sino la orientación del rostro. En esta fase no se realiza proceso alguno de identificación, sólo se determina si existe un rostro dentro de la imagen.

El proceso de detección de rostros permite reducir la complejidad del algoritmo de reconocimiento, ya que este último puede asumir que siempre cuenta con la imagen de un rostro.

La detección de rostros en sí misma es un problema complejo, debido a varios factores: los cambios de iluminación, el color de la piel es variable, el rostro

MsC. Alexander Cardona-López
MsC. Franklin Pineda-Torres
DOI: http://dx.doi.org/10.21501/21454086.2938

puede expresar diversidad de posturas, la textura facial varía no linealmente con el cambio de postura, entre otros aspectos.

A través de los años se han propuesto diversos métodos, algunos orientados específicamente a la detección, otros que pretenden realizar la detección y el reconocimiento de manera simultánea. En las siguientes secciones se presentarán algunos de los métodos de detección más conocidos, clasificados en: métodos que buscan reducir la dimensionalidad de los datos y métodos que buscan extraer las características relevantes de la imagen (características geométricas, texturas, etc.).

Reducción de dimensionalidad

Las imágenes son datos que por su naturaleza poseen un dominio muy grande de valores. Esto hace prácticamente imposible trabajar métodos que operen sobre toda la imagen, utilizando todo el dominio de valores. Los métodos de reducción de dimensionalidad, como su nombre lo indica, pretenden reducir la dimensionalidad de los datos sin perder sus características relevantes.

No es de extrañar que los métodos de detección de rostro se basaran en reconocidos procedimientos estadísticos para reducir los datos, siendo los métodos lineales como el Principal Component Analysis (PCA) o el Linear Discriminant Analysis (LDA), los más mencionados en la literatura, posiblemente debido a que éstos cuentan con estrategias de clasificación computacionalmente eficientes. En PCA cada componente es la combinación lineal de las dimensiones originales que presentan una mayor variabilidad y que son ortogonales con los primeros componentes principales. El LDA por otra parte busca encontrar las diferencias entre clases. Otro método, comúnmente referenciado es el Independent Component Analysis (ICA), el cual a diferencia del PCA intenta transformar los datos como combinaciones de datos estadísticamente independientes.

El método de detección y reconocimiento Eigenfaces [8] es un ejemplo del uso de PCA para la detección de rostros. En este caso el procedimiento generalmente consiste en utilizar un conjunto de imágenes de entrenamiento sobre las cuales se determinan los vectores principales, es decir las "eigenfaces" cuya combinación lineal describe mejor todas las imágenes del conjunto. Posteriormente, en el proceso de detección se procede a comparar secciones de la imagen de entrada con el valor promedio del espacio de Eigenfaces, utilizando para ello una medida de distancia.

De manera similar, el método Fisherfaces hace uso de LDA como método de reducción. La idea de usar LDA es aprovechar que este método busca discriminar por "clases" en vez de trabajar con el dominio total. Los proponentes del método afirman que éste se comporta mejor cuando se manejan imágenes con variaciones de iluminación y de expresiones. Sin embargo, algunos trabajos también muestran que cuando el conjunto de imágenes de entrenamiento es pequeño, PCA puede tener mejor desempeño que LDA [6].

Extracción de características

Existen gran cantidad de métodos para realizar la extracción de características de un rostro (En el survey [9] puede encontrar una clasificación de los mismos). Sin embargo, actualmente, los métodos basados en la apariencia son los más comunes, en estos casos se recolecta un conjunto de imágenes ejemplo, se aplica a cada una un algoritmo para resaltar características, y se adopta algún algoritmo de aprendizaje para aprender un modelo. Hay por tanto, dos pasos importantes en este proceso: la extracción de características y la aplicación del algoritmo de aprendizaje.

Los algoritmos de extracción de características buscan encontrar un descriptor preciso, invariante a las transformaciones y en lo posible con un espacio de valores pequeño que facilite la clasificación. En la literatura se destaca el uso de técnicas basadas en HAAR, en Gabor y las basadas en Histograma [10].

Posiblemente el más difundido de los algoritmos de detección es el Viola-Jones [11]. En este método la detección de las características relevantes del rostro se realiza mediante el uso de filtros que se asemejan a las funciones bases de la transformada HAAR, cuyos resultados se clasifican con la ayuda del algoritmo AdaBoost. La Fig. 1 muestra un ejemplo de los filtros usados.

El algoritmo Viola-Jones comienza con una fase de entrenamiento, en el que el algoritmo debe "aprender" las características que identifican un rostro. Cada filtro se aplica sobre la imagen realizando un cálculo sencillo sobre los pixeles afectados por el filtro. El valor de un filtro con dos rectángulos es la diferencia entre la suma de los pixeles dentro de cada región rectangular. El valor de un filtro con tres rectángulos se determina mediante la resta de la suma de los pixeles de las

DOI: http://dx.doi.org/10.21501/21454086.2938

regiones exteriores con la suma de la región central. Existe una gran cantidad de filtros rectangulares que se puede aplicar sobre una imagen, pero aplicarlos todos haría que el desempeño fuera muy bajo. Para acelerar el proceso Viola-Jones realiza dos propuestas: hacer uso de una imagen integral y emplear el algoritmo AdaBoost. Adicionalmente el Viola-Jones emplea una "cascada de clasificadores", que básicamente es un árbol de decisión en cascada.

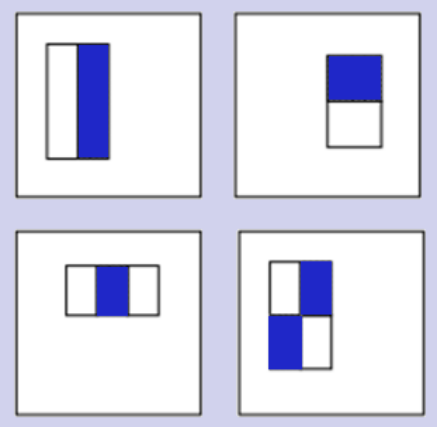

Fig. 1. Filtros HAAR

Un método de extracción de características más reciente es el Histogram of Oriented Gradients (HOG). A diferencia del Viola-Jones no es un método orientado exclusivamente a la detección de rostros, inclusive, el trabajo inicial donde se propone el algoritmo está enfocado a la detección de peatones en imágenes tradicionales [12]. El HOG es un método más robusto ante cambios de iluminación comparado con el Viola-Jones.

El descriptor del HOG es básicamente un histograma de los principales gradientes de la imagen. Para obtener dicho descriptor la imagen se divide en regiones rectangulares (celdas) y sobre cada una se calcula un histograma de gradientes. El descriptor final será la unión de todos estos gradientes.

Con el fin de lograr un histograma más resistente a los cambios de iluminación, en el descriptor final se hace uso de un histograma normalizado. Curiosamente, el proceso de normalización no se hace sobre los histogramas de cada celda, en vez de ello, las celdas se agrupan en bloques que se superponen y el histograma final es la concatenación de los histogramas del bloque dividido por su magnitud. Finalmente, el descriptor de la imagen es la unión de los histogramas normalizados de cada bloque. La Fig. 2 muestra los histogramas resultantes de aplicar el algoritmo HOG sobre una imagen del dataset LFW, en este caso cada componente del histograma se grafica como un vector para facilitar su visualización.

Aunque en teoría se puede emplear cualquier método de entrenamiento, comúnmente se hace uso de máquinas vectoriales (SVM) utilizando un kernel lineal. Este justamente es el método utilizado por los autores del HOG debido a su desempeño. También es común encontrar el uso de redes neuronales para el proceso de entrenamiento [13]. Sin importar el método escogido, la idea es realizar un entrenamiento previo con imágenes de rostros (muestras positivas) y no rostros (muestras negativas). Para cada imagen se procede a determinar el descriptor HOG explicado anteriormente, siendo los valores resultantes la entrada para el algoritmo de entrenamiento. Una vez se ha finalizado el entrenamiento, se puede realizar la detección de rostros sobre una imagen.

Como un rostro puede aparecer en cualquier parte de la imagen, normalmente se opta por un algoritmo de ventana deslizante. El esquema de ventana deslizante es costoso en procesamiento, por lo que se acostumbra a emplear un esquema en cascada para acelerar su funcionamiento [14]. También es factible paralelizar el proceso, los trabajos [15] y [16] son ejemplos del uso de la GPU (Graphics Processing Unit) para acelerar el algoritmo.

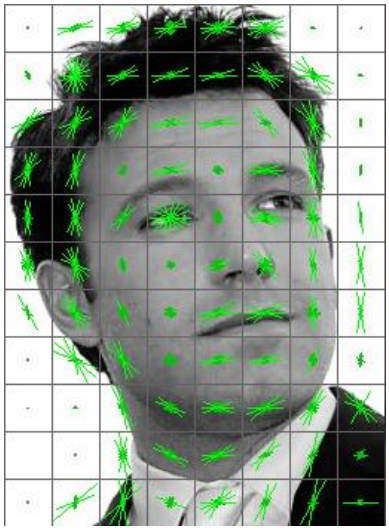

Fig. 2. HOG

Existen otros métodos de extracción de características de propósito más general, cuyo uso es más común en la fase de reconocimiento de rostros. Estos métodos se mencionarán más

MsC. Alexander Cardona-López
MsC. Franklin Pineda-Torres
DOI: http://dx.doi.org/10.21501/21454086.2938

adelante, sin embargo, es de aclarar que varios de ellos pueden emplearse también en la detección de rostros, siguiendo un proceso similar al de los métodos antes descritos.

2.2. Reconocimiento de rostros

El reconocimiento de rostros busca determinar si una imagen corresponde a un rostro previamente registrado, en otras palabras intenta identificar a quién pertenece el rostro que aparece en la imagen.

Una fuerte tendencia en el reconocimiento de rostros es el uso de métodos locales; dichos métodos no buscan una descripción de la imagen en su totalidad, en vez de ello se enfocan en obtener representaciones de sub-regiones de la imagen, comúnmente denominadas parches (patches en inglés). Al usar regiones pequeñas de la imagen se tiene un menor número de valores para representar, y por tanto, se puede hacer uso de un menor número de variables. Adicionalmente, el no tratar la imagen como un todo hace que los algoritmos se vean menos afectados por deformaciones u oclusiones en la imagen [4].

La forma de las regiones y su posición dentro de la imagen total del rostro, varía de un método a otro. Algunos métodos optan por hacer uso de regiones rectangulares formando una grilla sobre la imagen, otros métodos hacen uso de regiones que se superponen, otros hacen uso de regiones circulares, etc. En general la idea es: tomar regiones de la imagen, utilizar sobre cada región un algoritmo para detección de características y crear un descriptor que agrupe los resultados de cada región.

Entre los métodos de detección de características más usados se encuentran: Scale Invariant Feature Transform (SIFT) [17], Speed Up Robust Features (SURF) [18], Features from Accelerated Segment Test (FAST) [19], FAST Oriented FAST and Rotated BRIEF (ORB) [20] y los patrones binarios [21]. SIFT se basa en realizar una diferencia de kernels Gaussianos, a diferentes escalas de la imagen buscando las áreas cuya variación se asemeje más a una esquina. SURF se basa en el mismo principio que SIFT, pero emplea filtros rectangulares para aproximarse al resultado de la diferencia de Gauss, lo que mejora su velocidad de respuesta. Adicionalmente, emplea una matriz de Hessian para la orientación y la escala del punto evaluado. FAST realiza operaciones de diferencia de pixeles en un área circular alrededor del punto, empleando un método de selección de pixeles que permite utilizar sólo un conjunto de los pixeles que hacen parte del área circular. ORB es una fusión entre dos métodos: FAST y BRIEF, que usa este último para generar una cadena binaria que representa las características relevantes de la imagen. FREAK, al igual que BRIEF y ORB, producen una cadena binaria, pero en este caso la forma en que se realiza el recorrido sobre la imagen está inspirado en el comportamiento de la retina del ojo humano.

En el caso del histograma de patrones binarios locales (LBPH), las características se representan como un patrón de bits que corresponde a la estructura local de una parte de la imagen. En su idea inicial los patrones de bits surgen de la comparación de un pixel con sus vecinos, si la intensidad del pixel es mayor que la de su vecino esto se expresa con un "1", en caso contrario se expresa con un "0". La unión de los bits producidos al comparar el pixel central con cada vecino, siguiendo un orden pre-establecido, produce un número binario. Para el caso de reconocimiento de rostros, se subdivide la imagen en regiones rectangulares, para cada una de las cuales se calcula un histograma [21]. El descriptor final de la imagen del rostro es la concatenación de cada uno de los histogramas locales. Al igual que con otras propuestas, el descriptor se emplea como entrada para un método de entrenamiento.

El problema de usar métodos que se basan en información local, es que no se describe la relación geométrica entre las partes, información que puede ser relevante para la identificación de un rostro. Se han propuesto por tanto, métodos que generan descriptores que incluyen información global y local. Estos métodos, conocidos como métodos jerárquicos, pretenden obtener una descripción local invariante a los cambios de iluminación y una descripción global que ayude a las labores de identificación. En el caso de reconocimiento de rostros, es común el empleo de técnicas de Deep Learning para este tipo de representaciones. El survey [22] resume un conjunto de trabajos en los que se hace uso de Deep Learning para el reconocimiento de rostros. De forma similar el trabajo [13] muestra cómo se pueden usar diversos tipos de redes neuronales, en las diferentes fases del reconocimiento de rostros.

Otra alternativa es procurar ubicar las regiones en los puntos relevantes del rostro, como ojos, nariz y boca, y procurar realizar únicamente la descripción de estas regiones. Esto normalmente implica el uso inicial de algoritmos de detección de puntos clave (landmarks) como el Active Shape Model (ASM) [23]. El ASM permite identificar en una imagen de un

rostro la ubicación de características como nariz, cejas, ojos, boca o barbilla. El esquema general en reconocimiento es utilizar ASM o alguna de sus variaciones para obtener los puntos clave de la imagen, tomar una región alrededor de cada punto y describirla utilizando un algoritmo de detección de características (Gabor, ORB, Binary patterns, etc.), para posteriormente agrupar las características de cada región en un único descriptor

3 PRUEBAS

A pesar de que los métodos de reconocimiento son independientes de la fuente de las imágenes, las imágenes capturadas por un dispositivo móvil de forma cotidiana tienden a tener características diferentes a las imágenes destinadas a ser publicadas. En [24] se abordan estas diferencias y sus posibles implicaciones en la detección de rostros. Adicionalmente, los dispositivos móviles, en especial de bajo costo no cuentan con un hardware comparable con los equipos de escritorio. Trabajos como [25] o [26] son ejemplos de propuestas para afrontar el bajo desempeño de estos equipos. Aunque menos comunes, ya se observan trabajos orientados a mejorar la detección en imágenes de baja calidad, el survey [27] resume varios de los avances en este sentido.

En el caso de este documento, se analiza el desempeño de algunos de los métodos de reconocimiento mencionados en el mismo, cuando se emplean dispositivos móviles de bajo costo.

Para probar el desempeño del proceso de detección se escogió un método basado en extracción de características (el HOG) y para el reconocimiento, se eligió un método de características locales (el LBPH). Además se utilizó el ya tradicional Viola-Jones, como punto de referencia debido a que su eficiencia en tiempo de ejecución es ya reconocida. Para el caso del reconocimiento se escoge un método de extracción de características locales, ya que este tipo de métodos son lo que presentan una mayor cantidad de propuestas en los últimos años. Además como la mayoría de métodos de extracción de características se basan en operaciones de comparación entre pixeles vecinos, se puede pensar que la velocidad de ejecución será similar para todos los métodos (si no se cuenta la influencia del método de aprendizaje de máquina seleccionado).

Para evaluar los métodos antes descritos se crea una aplicación en Android que realiza las siguientes actividades: captura el frame de video de la cámara, pre-procesamiento de la imagen, detección y reconocimiento. La aplicación se realiza en Android debido a que es común el uso de este sistema operativo en equipos de bajo costo. No se realiza corrección de postura o tracking. Los aspectos que se tienen en cuenta para medir el desempeño del sistema son: tiempo de captura, tiempo de conversión y pre-procesamiento de la imagen, tiempo de detección, resultados de la detección y tiempo del reconocimiento.

No se tiene en cuenta el tiempo de entrenamiento, ya que este proceso se puede realizar en muchos casos antes de la distribución de la aplicación de software. Sin embargo, cabe resaltar que el proceso de entrenamiento aunque no tiene una influencia directa en los tiempos de ejecución, si tiene una gran influencia en la precisión de los resultados.

Aunque la medición de los tiempos de captura y conversión de una imagen se realizó directamente con la cámara del dispositivo, para la detección se empleó el conjunto de imágenes de la base FDDB [28], elaborada por la Universidad de Massachusetts. La base está organizada en diez secciones, cada una de las cuales corresponde a un listado de los rostros presentes en imágenes de la base LFW. Esta misma base se utilizó para el reconocimiento, pero sólo para la medición de tiempos. Para acelerar el procesamiento la imagen de entrada se convertía a una imagen en grises de 384 o de 480 pixeles de alto. Esto no afecta la calidad de los resultados ya que ninguno de los métodos utilizados hace uso del color como característica de descripción.

Para medir el desempeño de la aplicación se realizaron pruebas en dos equipos: un equipo con procesador de cuatro núcleos de 1.2 Ghz, con una cámara de 2 mp, y un equipo con procesador de dos núcleos de 1.2 Ghz con una cámara de sólo 0.3 mp. Ambos equipos cuentan con 1 Gb de memoria RAM.

4 RESULTADOS

En esta sección se presenta un resumen de los resultados de las pruebas realizadas. Estos resultados se dividen en tres partes: el tiempo requerido para la captura y pre-procesamiento de imágenes, los resultados de la fase de detección y los resultados de la fase de reconocimiento de rostros. En todos los casos, los valores mostrados son los valores promedio obtenidos en los dispositivos empleados para las pruebas.

MsC. Alexander Cardona-López
MsC. Franklin Pineda-Torres
DOI: http://dx.doi.org/10.21501/21454086.2938

4.1 Fase de Captura y Pre-procesamiento

Para la captura se empleó la cámara posterior del dispositivo, siendo posible acceder al buffer de pixeles de cada frame de video en tiempo real. Sin embargo, en los equipos utilizados para las pruebas, sólo era posible acceder a los valores comprimidos de cada frame, por lo que fue necesario dedicar tiempo del procesador para el proceso de descompresión. Una vez obtenida la imagen sin compresión, ésta se convertía a grises, se escalaba y se equalizaba (equalización del histograma). El tiempo promedio de todo este proceso fue de 16.75 ms por cada frame.

4.2 Fase de Detección

La Tabla 1 muestra los resultados de usar el Viola-Jones (con filtros HAAR) cuando las imágenes son escaladas a una altura de 384 pixeles. De manera similar la tabla 2 muestra los resultados del método HOG para estas mismas imágenes. La columna *Escala* hace referencia a la proporción en que se escala la imagen durante la aplicación de los filtros, *N* es el número de imágenes reconocidas, *VP* son los verdaderos positivos y *FP* son los falsos positivos. En el caso del método HOG la columna *Posturas* presenta el número de posturas pre-entrenadas en el modelo (donde 1 se refiere al empleo de un único modelo de entrenamiento, en posición frontal).

Tabla 1. Viola-Jones. 384 pixeles

Método	*Escala*	*N*	*VP*	*FP*	*Tiempo (ms)*
HAAR	1,2	470	207	128	99,4586206897
HAAR	1,5	412	170	134	41,7689655172
HAAR	2	322	111	93	25,3793103448

Tabla 2. HOG. 384 pixeles

Método	*Escala*	*Posturas*	*VP*	*FP*	*Tiempo (ms)*
HOG	10	1	227	71	187,5965517241
HOG	10	3	235	89	253,8724137931
HOG	10	5	237	90	334,6655172414
HOG	100	1	227	71	188,6482758621
HOG	100	3	235	89	254,975862069
HOG	100	5	237	90	330,6827586207
HOG	1000	1	227	71	188,2448275862
HOG	1000	3	235	89	253,1137931034
HOG	1000	5	237	90	329,6862068966

De manera similar, las Tablas 3 y 4 presentan los resultados para imágenes de 480 pixeles de alto.

Tabla 3. Viola-Jones. 480 pixeles

Método	*Escala*	*N*	*VP*	*FP*	*Tiempo (ms)*
HAAR	1,2	458	205	127	115,7482758621
HAAR	1,5	403	175	132	46,4517241379
HAAR	2	302	102	84	26,4689655172

Tabla 4. HOG. 480 pixeles

Método	*Escala*	*Posturas*	*VP*	*FP*	*Tiempo (ms)*
HOG	10	1	220	60	221,2931034483
HOG	10	3	231	73	300,024137931
HOG	10	5	236	72	397,5620689655
HOG	100	1	220	60	223,8379310345
HOG	100	3	231	73	302,5275862069
HOG	100	5	236	72	395,8310344828
HOG	1000	1	220	60	221,5
HOG	1000	3	231	73	299,5827586207
HOG	1000	5	236	72	396,6034482759

Se puede observar que el tiempo de ejecución del Viola-Jones es mejor que el tiempo del HOG en todos los casos, incluso cuando el HOG se emplea con un descriptor de una sola postura (el rostro frontal). Pero los resultados de la detección son mejores en el HOG, teniendo este último mayores "verdaderos positivos" y menos "falsos positivos". Por otra parte, no hay una diferencia importante en los resultados del método HOG cuando se varía el tamaño de la imagen de 384 pixeles a 480 pixeles, Fig. 3. Aunque este factor si es relevante en el tiempo de ejecución, junto con el número de posturas usadas, Fig. 4.

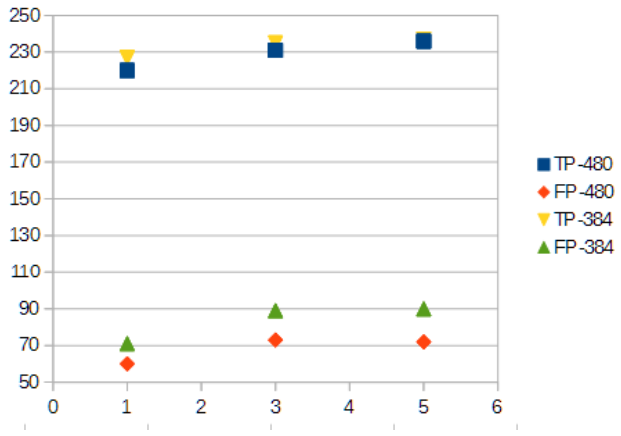

Fig. 3. Posturas vs detecciones positivas

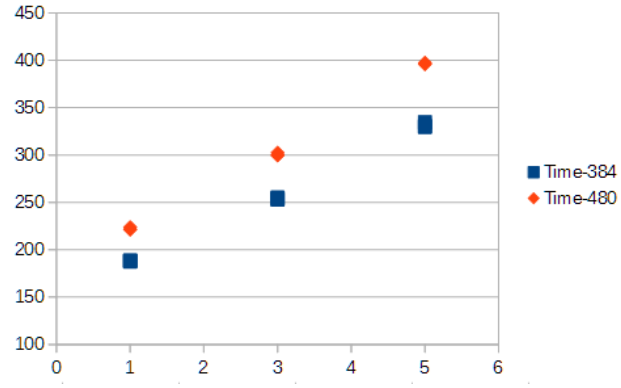

Fig. 4. Posturas vs tiempo de ejecución (ms)

Hay que tener en cuenta que las imágenes capturadas por una cámara de dos mega-pixeles, tendrán una mayor resolución horizontal que las utilizadas en el conjunto de prueba (aunque internamente la imagen se escale a una altura similar el ancho continúa siendo disímil) por lo que los tiempos de ejecución en imágenes capturadas por la cámara serán mayores.

4.3 Fase de Reconocimiento

La tabla 5 muestra el tiempo promedio utilizado por el algoritmo LBPH en procesar una imagen. En todos los casos se empleó un esquema de ocho vecinos. La columna radio de la tabla corresponde al radio en pixeles utilizado para determinar los vecinos, celdas es la cantidad de filas y columnas en que se divide la imagen, mientras que Escala es el tamaño al que se escala la imagen. Como puede observarse la diferencia entre el mejor y el peor tiempo es de sólo 0.95 milésimas de segundo, por lo que, si se mantiene el mismo número de vecinos la ejecución es prácticamente constante.

Tabla 5. Método LBPH

Radio	Celdas	Escala	Tiempo (ms)
1	5	70	11,1174698795
1	5	128	11,4608433735
1	8	128	10,9819277108
2	5	70	10,6777108434
2	5	128	11,0060240964
2	8	128	11,1596385542
4	5	70	11,6325301205
4	5	128	10,7228915663
4	8	128	10,9789156627

Como era de esperarse, el tiempo de ejecución y la precisión del proceso total dependen de los algoritmos seleccionados para realizar la detección y el reconocimiento. A pesar de ello, se puede observar que el tiempo del proceso de reconocimiento es bajo comparado con el tiempo de detección y también es menor que el tiempo de captura y pre-procesamiento. En cualquier caso, sin importar los métodos seleccionados, el proceso completo no toma un tiempo menor a los 50 milisegundos.

5 CONCLUSIONES

En cuanto a dispositivos móviles de consumo masivo se refiere, no parece posible lograr aplicaciones de reconocimiento de rostros que trabajen sobre un stream de video, sin asumir limitaciones en cuanto a la calidad del proceso de detección o sin la implementación de métodos de tracking. Por lo menos en el caso de los métodos probados.

El proceso de captura y pre-procesamiento por si sólo requiere 17 ms, lo que deja sólo 16 milisegundos para procesar cada frame de un video con una frecuencia de 30 frames por segundos. Esto de por sí hace poco probable poder realizar el procesamiento de cada frame de video en tiempo real sin importar cuáles son los algoritmos seleccionados, ya que estos 16 milisegundos deben repartirse entre el proceso de detección y reconocimiento.

En cuanto a las diferentes fases del sistema de reconocimiento, la detección de rostros es la más exigente, llegando a ser de hasta 300 ms cuando se

DOI: http://dx.doi.org/10.21501/21454086.2938

usa el algoritmo HOG tradicional con varias posturas. En general, parece conveniente utilizar un algoritmo como el Viola-Jones, que aunque no tan preciso tiene un buen tiempo de respuesta, o utilizar versiones optimizadas de otros algoritmos (que hagan uso de la GPU por ejemplo) junto con un algoritmo de tracking. Un algoritmo de tracking permitiría reducir el número de frames en los que es necesario el proceso de detección, ya que en algunos frames el espacio de búsqueda se limitaría a aquellas regiones de la imagen donde "posiblemente" aparece un rostro detectado en un frame anterior.

La selección del algoritmo de detección debe tener en cuenta además, que la calidad de las imágenes en un stream de video no es similar a la calidad de una imagen fija, sobre todo en cámaras utilizadas por dispositivos de bajo costo. Es por tanto más probable la aparición de imágenes borrosas.

También es necesario restringir la detección a un número limitado de posturas (idealmente sólo la frontal), teniendo en cuenta que incluso el método Viola-Jones requirió cerca de 40 milisegundos para dar resultados aceptables con una única postura. El uso de varias posturas no dejaría tiempo disponible para el proceso de reconocimiento incluso con el uso de un algoritmo de tracking.

REFERENCIAS

[1] D. Changxing, C. Jonghyun, T. Dacheng, and L. S. Davis, "Multi-Directional Multi-Level Dual-Cross Patterns for Robust Face Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 3, pp. 518-531, 2016.

[2] S. Rahimzadeh Arashloo and J. Kittler, "Fast pose invariant face recognition using super coupled multiresolution Markov Random Fields on a GPU," *Pattern Recognition Letters*, vol. 48, pp. 49-59, 2014.

[3] E. Learned-Miller, G.B. Huang, A.R. Chowdhury, H. Li, and G. Hua, "Labeled faces in the wild: A survey," *Advances in Face Detection and Facial Image Analysis*, pp. 189-248, 2016.

[4] V. Vijayakumari, "A Survey on Face Recognition Techniques," *World Journal of Computer Application and Technology*, vol. 1, no. 2, pp. 41-50, 2013.

[5] C. Zhang and Z. Zhang, "A Survey of Recent Advances in Face Detection," *Microsoft Research*, no. June, p. 17, 2010.

[6] R. Jafri and H. R. Arabnia, "A Survey of Face Recognition Techniques," *Journal of Information Processing Systems*, vol. 5, no. 2, pp. 41-68, 2009.

[7] N. Wang, X. Gao, D. Tao, H. Yang, and X. Li, "Facial feature point detection: A comprehensive survey," *Neurocomputing*, vol. 275, pp. 50-65, 2018. [Online]. Available on: http://www.sciencedirect.com/science/article/pii/S0925231217308202

[8] M.a. Ma Turk and Ap A.P. Pentland, Face Recognition Using Eigenfaces, 1991.

[9] M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting Faces In Image : A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34-58, 2002.

[10] R. Ahdid, K. Taifi, S. Safi, and B. Manaut, "A Survey on Facial Feature Points Detection Techniques and Approaches," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 10, no. 8, pp. 1504-1511, 2016.

[11] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. I-----511----I----518, 2001.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. I, pp. 886-893, 2005.

[13] M. M Kasar, D. Bhattacharyya, and T.H. Kim, "Face Recognition Using Neural Network: A Review," *International Journal of Security and Its Applications*, vol. 10, no. 3, pp. 81-100, 2016. [Online]. Available on: http://dx.doi.org/10.14257/ijsia.2016.10.3.08

[14] Q Zhu, S. Avidan, M. C. Yeh, and K.-T. Cheng, "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients," *Computer Vision and Pattern Recognition*, vol. 2, pp. 1491-1498, 2006.

[15] V. A. Prisacariu and I. Reid, "fastHOG - a real-time GPU implementation of HOG," *Science*, vol. 2310, no. 2310, pp. 1-13, 2009. [Online]. Available on: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.159.1161rep=rep1type=pdf

[16] M. Hirabayashi, S. Kato, M. Edahiro, K. Takeda, T. Kawano and S. Mita, "GPU implementations of object detection using HOG features and deformable models," 2013 IEEE 1st International Conference on Cyber-Physical Systems, Networks, and Applications, CPSNA 2013, pp. 106-111, 2013.

[17] D.G. Lowe, "Distinctive image features from scale invariant keypoints," *Int'l Journal of Computer Vision*, vol. 60, pp. 91-11020042, 2004.

[18] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," *Lecture Notes in Computer Science* (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3951 LNCS, pp. 404-417, 2006.

[19] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," *Lecture Notes in Computer Science* (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 3951 LNCS, pp. 430-443, 2006.

[20] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," Proceedings of the IEEE International Conference on Computer Vision, pp. 2564-2571, 2011.

[21] T. Ahonen, A. Hadid, M. Pietikäinen, and M. Pietik, "Face recognition with local binary patterns," *Proc. of the European Conference on Computer Vision* (ECCV), pp. 469-481, 2004. [Online]. http://www.springerlink.com/index/P5D9XP9GFKEX5GK9.pdf

[22] M. Wang and W. Deng, "Deep Face Recognition : A Survey," *ArXiv e-prints*, pp. 1-17, 2018. [Online], Avaiable on: http://adsabs.harvard.edu/abs/2018arXiv180406655W

[23] S. Milborrow and F. Nicolls, "Locating facial features with an extended active shape model," *Proc. of European Conf. on Computer Vision*, vol. 5305 LNCS, no. PART 4, pp. 504-513, 2008.

[24] R. D. Findling and R. Mayrhofer, "Towards face unlock: on the difficulty of reliably detecting faces on mobile phones," *Proceedings of the 10th International Conference on Advances in Mobile Computing*

[25] E. Kodirov, A. Fahmi Pn, G. S. Lee, D. J. Choi, and In Seop Na, "Robust Real Time Face Tracking in Mobile Devices," *Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication*, no. 82, pp. 88:1----88:8, 2013. [Online], Available on: http://doi.acm.org/10.1145/2448556.2448644

[26] A. El-mahdy, R. Elsersy, and Face Recognition, "A Large-Scale Mobile Facial Recognition System Using Embedded GPUs," HPC '14 Proceedings of the High Performance Computing Symposium, p. 23, 2014.

[27] Y. Zhou, D. Liu, and T. Huang, "Survey of Face Detection on Low-quality Images," *ArXiv e-prints*, 2018. [Online], Available on: http://adsabs.harvard.edu/abs/2018arXiv180407362Z

[28] V. Jain and E. Learned-Miller, "FDDB : A Benchmark for Face Detection in Unconstrained Settings," University of Massachusetts, 2010.

[29] I. Kemelmacher-shlizerman Steve, S. Daniel, and C V Dec, "The MegaFace Benchmark : 1 Million Faces for Recognition at Scale," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4873----4882, 2016.

[30] Proyecto Fin De Carrera and Ion Marques, "Face recognition algorithms," Euskal Herriko, Tech. rep. 2010.

[31] L. Hoang Thai, "Face Alignment Using Active Shape Model And Support Vector Machine The Active Shape Model (ASM) is one of the most popular local texture models for face alignment. It applies in many fields such as locating facial features in the accuracy of the classi," International Journal of Biometrics and Bioinformatics, vol. 4, no. 6, pp. 224-234, 2012. [Online]. http://arxiv.org/abs/1209.6151v15Cnpapers2://publication/uuid/CCF48C7F-E0FD-4833-BA98-FC58E1679211