DISEÑO DE UN SISTEMA DE RECONOCIMIENTO FACIAL EN UN AMBIENTE CONTROLADO

FABIÁN STIVEN PÉREZ
LILIANA SAAB CANO

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
NEIVA
2012

DISEÑO DE UN SISTEMA DE RECONOCIMIENTO FACIAL EN UN AMBIENTE CONTROLADO

FABIÁN STIVEN PÉREZ
LILIANA SAAB CANO

Trabajo de grado presentado
Para optar al título de
Ingeniero Electrónico.

Director del proyecto
JOSE DE JESUS SALGADO PATRÓN
Ingeniero Electrónico.

UNIVERSIDAD SURCOLOMBIANA
FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA
NEIVA
2012

Nota de aceptación:

_____

_____

_____

_____

_____

Firma del Presidente del jurado

_____

Firma del jurado

_____

Firma del jurado

Neiva, 26 de octubre de 2012

Este trabajo de grado se lo dedico principalmente a Dios.
A mis padres y hermanos, por el apoyo que me han dado.
Por guiar mí camino para que sea una profesional.
A mi compañero de tesis, por apoyarme en este proyectó con toda su dedicación.

LILIANA SAAB CANO

Este trabajo de grado se lo dedico principalmente a Dios;
A mis abuelos y mi madre, por el acompañamiento que me han dado.
Por apoyarme en mi formación personal y profesional.

FABIÁN STIVEN PÉREZ

AGRADECIMIENTOS

Agradezco a Dios, a mis padres, hermanos, amigos, a mi jefe, a mis compañeros de trabajo y compañero de tesis por el apoyo que me han brindado en todo este tiempo, por permitirme compartir una etapa de mi vida llena de alegrías y obstáculos que con ayuda de ellos logre superar y de los cuales aprendí muchas cosas valiosas que me permitieron para llegar a ser la persona que soy hoy en día y gracias ha eso he logrado finalizar esta etapa muy importante con éxito.

Liliana Saab

En primer lugar, doy gracias a Dios, a mi familia, especialmente a mis abuelos y a mi madre por el apoyo brindado durante mis estudios, hago extensivo mi agradecimiento a todas las personas que de alguna forma contribuyeron en formación personal y profesional que hoy tengo: mis hermanos, compañera de tesis, compañeros de universidad y amigos. Finalmente agradezco al Tecnoparque del SENA y al ingeniero Héctor Sánchez por el apoyo brindado en el proyecto de grado.

Fabián Stiven Pérez

TABLA DE CONTENIDO

Pág.

LISTA DE FIGURAS

pág.

LISTA DE TABLAS

pág.

LISTAS DE ANEXOS

pág.

GLOSARIO

CLASIFICADOR DÉBIL: Es un clasificador de objetos que tiene un rendimiento muy bajo, levemente mejor que el azar.

CLASIFICADOR FUERTE: Es un clasificador de objetos compuesto por varios clasificadores débiles y del que se espera un mejor desempeño, en comparación a un clasificador débil.

COVARIANZA: La covarianza es una medida de la variación común a dos variables y, por tanto, una medida del grado y tipo de su relación.

DATOS DE ENTRENAMIENTO: Están compuestos por las imágenes de entrenamiento, la lista de etiquetas de usuario y los datos generados por Análisis de los Componentes Principales (eigenfaces, eigenvalores, eigenvectores y pesos) de las imágenes de entrenamiento.

ECUALIZACIÓN DEL HISTOGRAMA: Consiste en encontrar una transformación con la cual el histograma tenga una representación uniforme, es decir, con la ecualización trataremos de igualar lo más posible el histograma de una imagen al histograma ideal.

EIGENFACES: Son un conjunto de vectores propios derivados de un conjunto de rostros.

EIGENVALORES: Es el factor de escala por el que ha sido multiplicado el vector propio.

EIGENVECTORES: Son los vectores no nulos que cuando son transformados por el operador, dan lugar a un múltiplo escalar de sí mismos, con lo que no cambian su dirección. También es llamado vector propio.

FALSOS POSITIVOS: Son aquellos objetos que son detectados pero que no deberían ser detectados.

HISTOGRAMA: El histograma de una imagen es la representación gráfica de la distribución que existe de las distintas tonalidades de grises con relación al número de pixeles o porcentaje de los mismos.

IMAGEN DE ENTRADA: Imagen de prueba capturada por la cámara web que se desea reconocer.

IMAGEN DE ENTRENAMIENTO: Son las imágenes que se encuentran en la base de datos del sistema que se toman como referencia para la identificación de personas.

INTERPOLACIÓN LINEAL: Es un método numérico (y grafico) que permite encontrar datos desconocidos entre o en medio de otros datos ya conocidos que representan una relación lineal entre sí.

MÉTODO DE BOOSTING: Es un método de aprendizaje estadístico el cual une los clasificadores débiles para convertirlos en un clasificador fuerte.

OPENCV: es una biblioteca libre de visión artificial originalmente desarrollada por Intel.

XML: siglas de Extensible Markup Language, es un estándar para el intercambio de información estructurada entre diferentes plataformas.

RESUMEN

Este trabajo presenta el diseño y desarrollo de un sistema de identificación de personas utilizando reconocimiento facial.

Dentro del diseño del sistema de reconocimiento facial se tiene en cuenta la posible variación en la iluminación de la escena y los efectos que esto podría presentar en el momento de la identificación, así como la posibilidad de que el sistema pudiera añadir nuevos usuarios en escenarios distintos.

Adicionalmente se explica detalladamente el método utilizado, los algoritmos manejados en el desarrollo del sistema de reconocimiento facial y cada una de las etapas asociadas en el funcionamiento del sistema.

También se presentará el proceso de instalación del software Visual C++ 2008 Express y la biblioteca de visión artificial OpenCV 2.3.1 utilizados en el desarrollo de este proyecto y se mostrará un manual de usuario donde se explicará de manera sencilla el manejo del sistema de reconocimiento facial realizado.

Finalmente se realizan algunas conclusiones y recomendaciones referentes al diseño y las dificultades presentadas en el proceso de desarrollo y realización del sistema de reconocimiento facial.

ABSTRACT

This work presents the design and development of a system for identifying people using face recognition.

Within the design of face recognition system was taken into account the possible variation in the illumination of the scene and the impact this could present at the moment of the identification, as well as the possibility that the system could add new users in different scenarios .

Additionally explain in detail the method used, algorithms handled in the development of facial recognition system and each of the associated stages in the operation of the system.

Also will present the software installation process Visual C++ 2008 Express and OpenCV computer vision library 2.3.1 used in the development of this project and will display a user's manual which will explain in simple handling facial recognition system performed.

Finally we then make some conclusions and recommendations for the design and the difficulties presented in the process of development and implementation of the face recognition system.

INTRODUCCIÓN

La biometría facial ha venido desarrollándose a nivel mundial como una herramienta que puede ser aplicada en sistemas de identificación de personas en operaciones legales o de seguridad.

La gran ventaja de la implementación de sistemas basados en biometría facial se debe a que los datos biométricos son diferentes e independientes en cada persona, maximizando la seguridad y fiabilidad a la hora de verificar la identidad de una persona y evitando los fallos más comunes (robo, olvido, transferencia) de los sistemas de identificación que utilizan contraseñas o documentos.

Un sistema de reconocimiento facial consta básicamente de cinco etapas. En la primera etapa se captura la imagen de la persona mediante una cámara digital. En la segunda etapa se realiza sobre la imagen obtenida, el pre-procesamiento necesario para resaltar los detalles de interés (por ejemplo reducción de ruido introducido por el dispositivo de captura). En la tercera etapa se realiza la detección del rostro eliminando los objetos no deseados. En la cuarta etapa se extraen las características a utilizar en el reconocimiento. En la quinta etapa se efectúa la clasificación, comparando el nuevo patrón con los patrones de usuarios registrados en la base de datos.

A pesar de que el reconocimiento facial es un tema bastante difundido, no es sencillo crear sistemas que realicen este trabajo, debido a la influencia de muchos factores externos como la iluminación de la escena, la calidad de la imagen, los gestos y la postura de la persona, que interfieren en el proceso de extracción de características del rostro.

En este trabajo, se desarrollo un sistema de reconocimiento facial con ciertas restricciones en la obtención de las imágenes faciales, generalmente requiriendo un fondo simple, buena iluminación y una posición frontal de la cara sin realizar gestos extraños para evitar posibles fallos en la identificación. Por otra parte el sistema de reconocimiento facial cuenta con la posibilidad añadir usuarios a la base de datos en distintos escenarios mediante una interfaz gráfica sencilla y de fácil manejo.

Finalmente cabe destacar que el sistema de reconocimiento facial podría implementarse a su vez con otros sistemas biométricos (reconocimiento de voz, huella dactilar, reconocimiento de iris.) para realizar una identificación más confiable.

1. MARCO TEÓRICO

1.1 BIOMETRÍA

La biometría es un conjunto de métodos que permiten reconocer a una persona basándose en sus características físicas (huella dactilar, rostro, iris, retina, geometría de la mano) o conductuales (escritura, voz). La gran ventaja que tiene este tipo de métodos es que los parámetros biométricos no pueden ser robados.

En este trabajo de tesis, se trabaja el método de reconocimiento facial que consiste en una aplicación dirigida por un computador, diseñada para reconocer automáticamente a una persona teniendo de entrada una imagen digital, por medio de una comparación entre características faciales en la imagen de entrada y las imágenes de la base de datos que contiene los rostros de las personas que se pretenden identificar.

1.2 SISTEMA DE RECONOCIMIENTO FACIAL

El sistema de reconocimiento facial consiste en una aplicación dirigida por un computador que identifica a una persona automáticamente en una imagen. Esto se puede desarrollar mediante el análisis de las características de la cara del sujeto que se extraen de la imagen o de una fuente de video comparándolas con las imágenes de rostro que se encuentran en la base de datos del sistema de reconocimiento facial.

El principal objetivo de un sistema de reconocimiento facial consiste en que dada una imagen de un rostro desconocido, se pueda encontrar una imagen del mismo rostro en un conjunto de imágenes ya conocidas o imágenes que han sido entrenadas. La mayor dificultad de este sistema es que el reconocimiento facial se pueda realizar en tiempo real y la variabilidad de iluminación que tenga el lugar donde es aplicado. El sistema pretende identificar los rostros presentes en una imagen o un video automáticamente, lo cual se puede realizar de dos maneras.

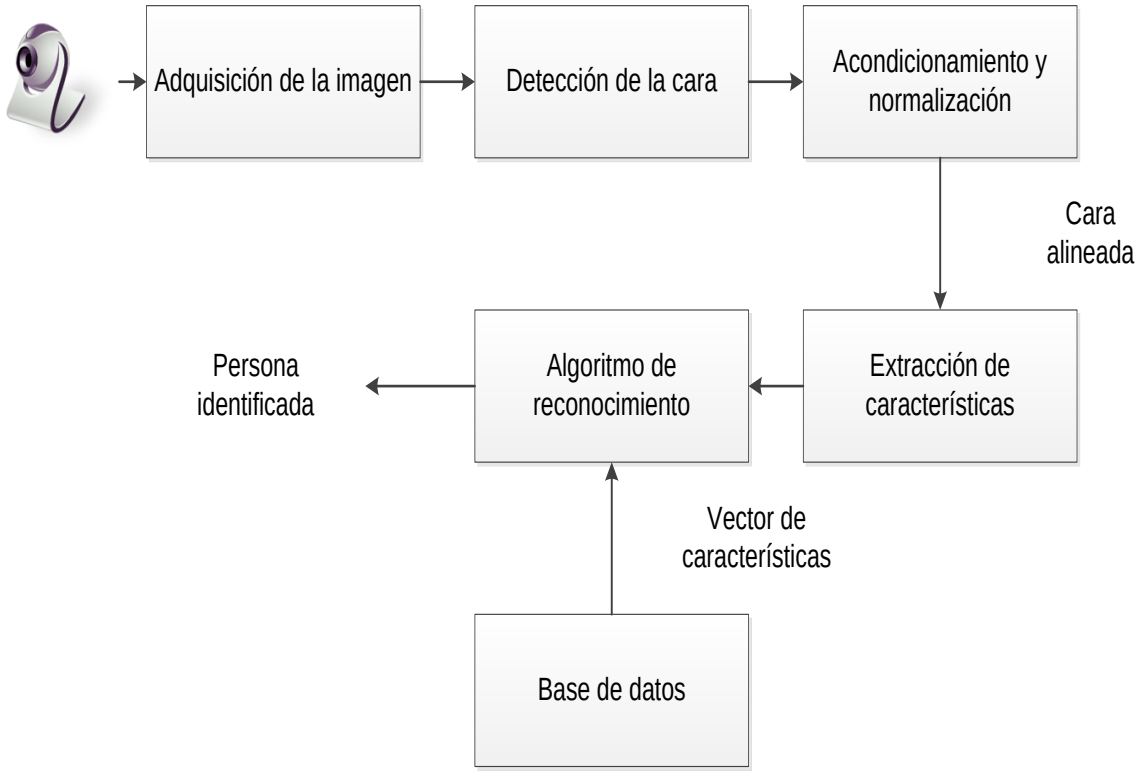
- **Verificación del rostro**: en esta etapa se compara una imagen del rostro con el rostro del cual se quiere conocer la identidad. El sistema lo que hará es confirmar o negar la identidad del rostro.

- **Identificación del rostro**: se compara la imagen del rostro desconocido con todas las imágenes de los rostros que se encuentran en una base de datos creada con anterioridad para determinar la identidad de la persona.

1.2.1 Aplicaciones. Este sistema se utiliza por lo general para autentificación de usuarios, videos de vigilancia avanzada, seguimiento de sospechosos, inicio de sesión en un computador, seguridad en determinadas aplicaciones, control de acceso en determinadas empresas, cajeros automáticos, pasaporte, registro de votantes, entre otros. En este sistema se utiliza un descifrador que determina las características de la cara y cuando este solicita el acceso se verifica comparando los datos que se obtuvieron con la base de datos creada.

1.2.2 Funcionamiento. El proceso del sistema de reconocimiento facial general consta de seis módulos principales que se muestra en la Figura 1.

Figura 1 Proceso de un sistema de reconocimiento facial



Fuente: http://es.wikipedia.org/wiki/Sistema_de_reconocimiento_facial.

1.2.2.1 Adquisición de la imagen. En este bloque del sistema de reconocimiento facial se obtienen las imágenes a las que se les quiere hacer el proceso de reconocimiento por medio de una cámara web.

1.2.2.2 Detección de la cara. En este bloque del sistema de reconocimiento facial se muestra que hay una cara en una imagen sin identificarla. Si se trata de un video se hace un seguimiento de la cara mediante un recuadro que enmarca la cara de la persona proporcionando la localización.

1.2.2.3 Acondicionamiento y normalización. En el bloque de alineación se localizan las principales componentes del rostro, mediante la normalización respecto a ciertas propiedades como lo son el tamaño, la posición y la iluminación. La normalización de la imagen que contiene el rostro se puede realizar de muchas formas, como lo son la distancia entre pupilas, la posición de la nariz o la distancia entre las junturas de los labios o simplemente normalizando el tamaño de la imagen. También se debe establecer la gama de colores, ya que se acostumbra a utilizar imágenes pequeñas en escala de grises disminuyendo la carga computacional del sistema. En este caso también se realiza una ecualización del histograma que se usa para obtener en una imagen un histograma con una distribución uniforme que implica que exista el mismo número de pixeles para cada nivel del histograma.

1.2.2.4 Extracción de características. Se adquiere la información que caracteriza a cada rostro, para posteriormente distinguirlo entre los rostros de diferentes personas dependiendo sus diferencias fotométricas.

1.2.2.5 Reconocimiento. Se compara el vector de características extraídas con los vectores de las características extraídas de los rostros que se encuentran en la base de datos. Si se encuentra un vector con una similitud muy alta muestra la identidad de dicha persona, en cambio, si no se encuentra un vector con parecido, este indica que el rostro de la persona no se conoce.

1.2.2.6 Base de datos. En este bloque del proyecto se guardan las imágenes de ejemplos que van a ser comparadas con una nueva imagen de entrada.

1.3 ESTADÍSTICA

Según Sebastián Long y Omar Müller, tanto la varianza como el desvió estándar son medidas unidimensionales. Sin embargo, muchos conjuntos de datos tienen más de una variable, y el objetivo del análisis estadístico es ver la relación entre las variables. Por ejemplo, si se registran en un examen las notas obtenidas por N alumnos en la variable X y la cantidad de horas de estudio en Y, se puede hacer un análisis para ver si las horas de estudio tienen algún efecto sobre la nota. En este caso, el desvió estándar y la varianza solo operan sobre una dimensión, por

lo que se pueden calcular sobre cada variable independientemente de la otra. Por otro lado, la covarianza permite operar sobre 2 variables mostrando la relación entre las mismas, y se define como

$$\text{cov}(X;Y)=\frac{\sum_{i=1}^{N}(X_i-\bar{X})(Y_i-\bar{Y})}{(N-1)} \quad \text{Ec. (1.1)}$$

Donde $\bar{X}$ es la media de X y $\bar{Y}$ es la media de Y.

Lo más importante en el análisis de la covarianza no es el valor exacto, sino más bien el signo. Si el valor es positivo, indica que ambas variables crecen juntas, si es negativo, mientras una variable crece la otra decrece y si es cero las variables son independientes una de otra.

Si se tiene un conjunto de datos con N variables X_1, X_2,….., X_n con N>2, se puede calcular una covarianza por cada par de variables. En estos casos, se calculan todos los valores de covarianza entre las diferentes variables y se ordenan en una matriz C denominada matriz de covarianza y se define según

$$C= (c_{ij}, c_{ij} = \text{cov}\ (X_i; X_j)) \quad \text{Ec. (1.2)}$$

Donde C es de N*N.

Para un conjunto de 3 variables por ejemplo, se obtiene

$$C = \begin{bmatrix} \text{cov}(X_1,X_1) & \text{cov}(X_1,X_2) & \text{cov}(X_1,X_3) \\ \text{cov}(X_2,X_1) & \text{cov}(X_2,X_2) & \text{cov}(X_2,X_3) \\ \text{cov}(X_3X_1) & \text{cov}(X_3,X_2) & \text{cov}(X_3,X_3) \end{bmatrix} \quad \text{Ec. (1.3)}$$

Como sirve, a los valores de la diagonal corresponde la covarianza de una variable consigo misma, o lo que es igual, la varianza de la variable. Además cov(a, b) = cov(b, a) por lo que la matriz resultante es simétrica y mantiene los valores más altos en la diagonal[1].

[1] Long, Sebastián y Müller, Omar. Verificación Biométrica Automática de Identidad Mediante Reconocimiento Facial. Santa Fe. 2006. p. 19-20

1.4 ÁLGEBRA MATRICIAL

Según Sebastián Long y Omar Müller, en muchas aplicaciones es útil encontrar para una transformación lineal T: V → V un vector v en el espacio vectorial V tal que Tv y v son paralelos. Es decir, se busca un vector v y un escalar λ tal que

$$T_v = \lambda_{\mathbf{v}} \quad \text{Ec. (1.4)}$$

Si v≠0 y λ satisface la Ecuación 1.4, entonces λ se llama un eigenvalor de T y v se llama un eigenvector de T correspondiente al eigenvalor λ. Si V tiene una dimensión finita, entonces T se puede representar por una matriz A_T

Dicho de otra forma, los eigenvectores de un operador lineal son los vectores no nulos que, cuando son transformados por el operador, dan lugar a un múltiplo escalar de sı mismos, con lo que no cambian su dirección.

Para una matriz hermítica $A_{N\times N}$ existen N eigenvalores y eigenvectores asociados, y los eigenvectores correspondientes a distintos eigenvalores son linealmente independientes. Esto significa que se pueden expresar los datos en términos de estos eigenvectores perpendiculares[2].

1.5 ALGORITMOS

En esta parte se explicaran los algoritmos escogidos para la realización del sistema de reconocimiento facial.

1.5.1 Haar Cascade Clasiffier o Clasificador en Cascada Haar. EL Haar Cascade Clasiffier es el detector de objetos (en este caso rostros) usado por OpenCV, este fue propuesto inicialmente por “Paul Viola y Michael Jones”[3] y mejorado por “Rainer Lienhart y Jochen Maydt”[4].

---

2 Long, Sebastián y Müller, Omar. Verificación Biométrica Automática de Identidad Mediante Reconocimiento Facial. Santa Fe. 2006. p. 19-20.

3 Paul Viola y Michael J. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE CVPR, 2001.

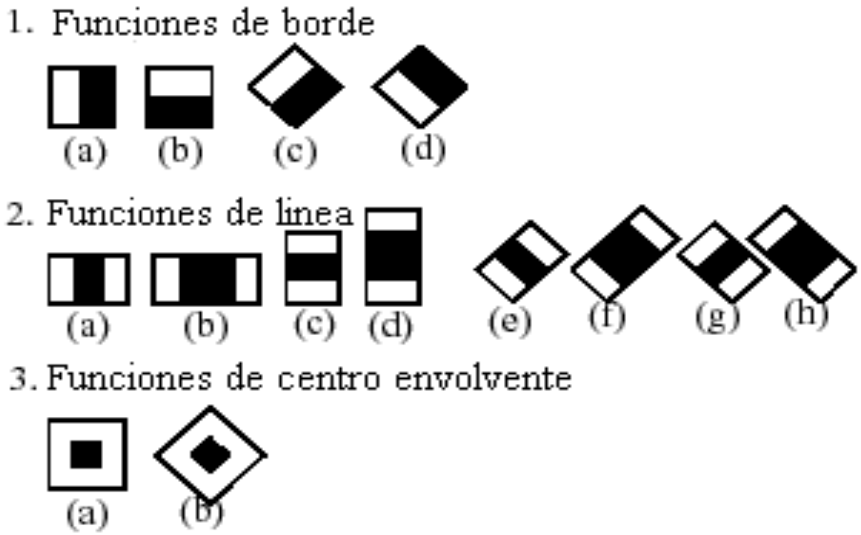
4 Lienhart, Rainer y Maydt Jochen. An Extended Set of Haar-like Features for Rapid Object Detection. IEEE ICIP 2002, Vol. 1, Sep. 2002, p. 900-903.

Este método utiliza cuatro conceptos claves los cuales son:

- Funciones rectangulares simples, llamadas funciones Haar.
- Una imagen integral para una rápida detección de características
- El método de aprendizaje *AdaBoost.*
- Un clasificador en cascada para combinar muchas funciones de manera eficiente.

Las funciones usadas por Viola y Jones están basadas en Wavelets de Haar. Estos clasificadores están conformados por simples ondas cuadradas en dos dimensiones, en el cual una onda cuadrada es un par de rectángulos adyacentes (uno claro y otro oscuro) como se muestra en la Figura 2.

Figura 2. Diferentes funciones Haar-like usadas en OpenCV



Fuente:http://opencv.itseez.com/modules/objdetect/doc/cascade_classification.

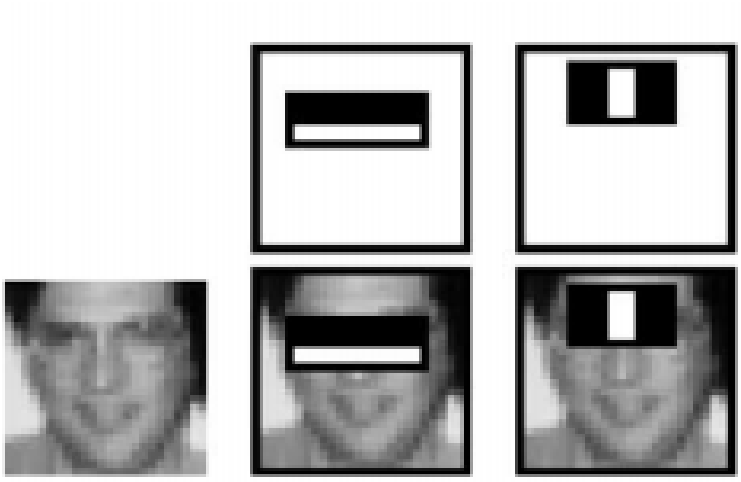
Inicialmente, el clasificador es entrenado tomando como base un objeto en particular en este caso imágenes de rostros, tomando alrededor de cien muestras de diferentes vistas del objeto, las cuales son llamadas muestras positivas y las coloca todas en un mismo tamaño, también se ponen imágenes arbitrarias del mismo tamaño las cuales se denominan muestras negativas.

En la Figura 3, se puede observar en la primera fila dos funciones Haar y en la segunda fila se observan la imagen de una cara y las dos funciones superpuestas a la cara; la primera función de izquierda a derecha diferencia la intensidad entre la región de los ojos y las mejillas, debido a que generalmente los ojos presentan una tonalidad oscura, mientras que las mejillas presentan una tonalidad clara. En

la segunda función superpuesta compara la tonalidad oscura en las regiones de los ojos con la de la tonalidad clara que presenta el canal de la nariz. Este es un ejemplo general de la caracterización de la imagen por medio de funciones Haar. En el entrenamiento se aplican en las diferentes subregiones de la imagen del rostro las funciones Haar, si la diferencia entre el valor de la función Haar y el valor promedio de los pixeles bajo las franjas de esta función es pequeña o superan un umbral característico se dice que la función Haar caracteriza la región.

Para determinar la presencia o ausencia de las funciones Haar en cada posición de la imagen, Viola y Jones aplicaron el concepto de imagen integral, en este caso el valor integral para cada pixel es la suma de todos los pixeles por encima de ella y a su izquierda. Comenzando en la parte superior izquierda atravesando hacia la derecha y hacia abajo, toda la imagen se puede integrar con unas pocas operaciones de enteros por pixel.

Figura 3. Funciones Haar superpuestas sobre las imágenes de rostro



Fuente: Viola, Paul y Jones, Michael. Rapid Object Detection using a Boosted Cascade of Simple Features.

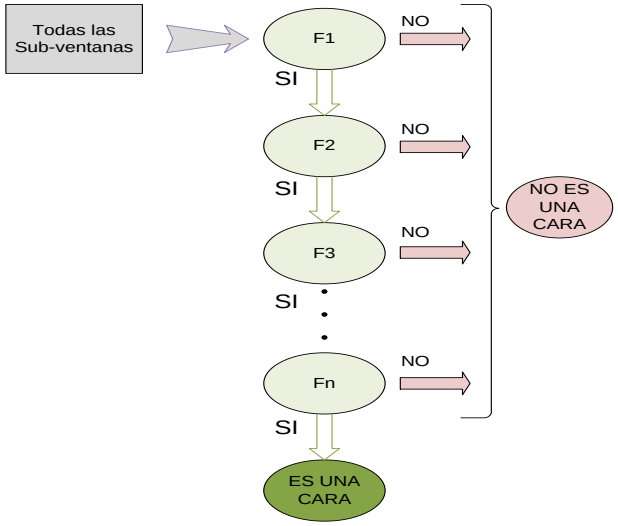
Una vez el clasificador ha sido entrenado puede ser aplicado en diferentes regiones de interés a una imagen de entrada. La salida del clasificador es 1 si el área esta dentro de los parámetros estipulados en el entrenamiento y 0 si no coincide.

La palabra clasificador en cascada significa que la clasificación resultante consiste de varias clasificaciones más sencillas. En este caso cada función Haar que caracteriza el rostro representa un clasificador sencillo (estados), que son aplicadas subsecuentemente en una región de interés (el rostro). La imagen es aceptada si todos los clasificadores sencillos son admitidos, de lo contrario si uno

de ellos no es conocido la imagen es rechazada. Los clasificadores de cada estado de la cascada están construidos en base a clasificadores básicos (funciones Haar), usando el método de Boosting. En la Figura 4, se muestra la forma como trabaja el método Boosting, donde cada filtro es un clasificador AdaBoost separado con un número bastante reducido de clasificadores débiles, creando unánimemente un clasificador fuerte.

Los filtros en cada nivel están capacitados para clasificar imágenes que han pasado todas las etapas anteriores. Durante su ejecución, si alguna región de la imagen no pasa alguno de estos filtros, esa región no califica como una cara. Cuando la región de la imagen pasa por toda la cadena de filtros es considerada una cara.

Figura 4. Clasificador en cascada como una cadena de filtros



Fuente: http://www.cognotics.com/opencv/servo_2007_series/part_2/sidebar.html.

El orden de los filtros en cascada se basa en la importancia de los pesos que asigna AdaBoost, los filtros de mayor peso se aplican primero para eliminar las regiones de la imagen que no son consideradas caras lo más rápido posible como se muestra en la Figura 4.

El clasificador utiliza los datos almacenados en un archivo XML para decidir cómo clasificar cada región de la imagen. El clasificador utilizado se llama haarcascade_frontalface_alt.xml.

1.5.2 Método AdaBoost. Este es un algoritmo de aprendizaje que puede ser utilizado para construir clasificadores sólidos usando una combinación lineal de

clasificadores simples. El método consiste en utilizar varias iteraciones para generar un aprendizaje único y fuerte.

El método de AdaBoost crea y llama a un clasificador débil cada uno en una serie de rondas, para cada llamada usa una distribución de pesos actualizados que indican la importancia de los ejemplos en el conjunto de datos para la sistematización. En cada ronda, los pesos de cada ejemplo clasificados incorrectamente se incrementan y los pesos correctos se reducen, por lo que el nuevo clasificador se centra en los ejemplos que se escapan a la correcta clasificación según Iván Cantador Gutiérrez [5].

1.5.3 PCA Y EIGENFACES. Basados en el documento realizado por "Matthew Turk y Alex Petland"[6], se considera que el método PCA consiste en analizar los datos y aplicar el reconocimiento del rostro con alguna variación en el método, el cual es llamado *eigenfaces*.

Este método se utiliza para el sistema de reconocimiento facial, ya que es fácil de implementar. Los pasos generales a seguir para hacer este reconocimiento son:

Paso 1. Se adquieren las imágenes de rostros de la base de datos. Se cargan las imágenes de cara que constituyen el conjunto de entrenamiento, estas deben tener las condiciones de iluminación, pose y resolución, soportados por el sistema para el reconocimiento.

Paso 2. Se calculan Eigenvectores o Eigenfaces del conjunto de imágenes de entrenamiento, se retienen únicamente las N imágenes concernientes a los Eigenvalores mayores. Estas Eigenfaces definen al espacio de las caras y deben ser actualizadas si se añaden nuevas caras.

Paso 3. Se proyectan las imágenes de los rostros de entrenamiento sobre el espacio de las caras N-dimensional, esta proyección da como resultado los pesos de cada imagen.

_____

5 Cantador Gutiérrez, Iván. Aplicación de Perceptrones Paralelos y AdaBoost a Problemas de Clasificación de Muestra Extrema, Universidad Autónoma de Madrid, 2005, p. 25-40.

6 Turk, Matthew, Pentland, Alex. Eigenfaces for Recognition, Journal of Cognitive Neurosicence, Vol. 3, No. 1, 1991, p. 71-86.

Paso 4. Se calcula el conjunto de pesos de la imagen de entrada basados en las N eigenfaces, es decir, se proyecta la imagen de entrada sobre cada una de las eigenfaces obtenidas de las imágenes de entrenamiento.

Paso 5. Se determina si los pesos de la imagen de entrada son cercanos (son similares) a los pesos de alguna de las imágenes de la base de datos, si su cercanía pasa un umbral de reconocimiento se define que la persona es conocida.

Teniendo en cuenta los pasos generales que se deben aplicar para el reconocimiento, se hace un análisis más a fondo de PCA exponiendo paso a paso el algoritmo matemático llevado a cabo en su procedimiento.

Primero que todo se asimila que una imagen puede suponerse como un vector de pixeles donde el valor de cada componente es un valor en escala de grises, por ejemplo una imagen de 128*128 (N*N) será un vector de dimensión 16384(N^2), es decir la imagen estará en un área de dimensión 16384. Las imágenes de las caras no están distribuidas aleatoriamente en este espacio y pueden ser referidas como un sub-espacio de menor dimensión. Lo que se pretende con este método es encontrar los vectores que mejor constituyen las imágenes de caras dentro del área completa de imágenes.

Cada vector de longitud N^2 es una estructura lineal de las imágenes únicas. Los cuales son los eigenvectores de la matriz de covarianza del espacio original de imágenes de caras.

Si se tiene un conjunto M de imágenes de caras x_1, x_2,…, x_M, se puede formar la matriz X poniendo como vectores fila cada imagen.

Luego se obtiene la media de las imágenes **Ψ** como se muestra en la Ecuación 1.5**.**

$$\Psi = \frac{1}{M} \sum_{i=1}^{M} X_i \quad \text{Ec. (1.5)}$$

En la Figura 5, se muestra el resultado de la media arrojado por el sistema de reconocimiento facial.

Figura 5. Media del conjunto de imágenes

A continuación en la Ecuación 1.6, se obtiene la diferencia entre la imagen de entrenamiento y la media.

$$\Phi_i = X_i - \Psi \qquad \text{Ec. (1.6)}$$

Seguidamente en la Ecuación 1.7, se calcula la matriz de covarianza C.

$$C = \frac{1}{M}\sum_{n=1}^{M} \Phi_n \Phi_n^T = AA^T \qquad (N^2 x N^2) \qquad \text{Ec. (1.7)}$$

donde A es la matriz $A = [\Phi_1\, \Phi_2 \ldots . \Phi_M]$ $(N^2 x M)$.

El siguiente paso es encontrar los Eigenvectores u de C.

Pero se tiene que la matriz de covarianza da dimensiones muy elevadas, por lo que no es viable calcular los eigenvectores de esta, debido a que implicaría un elevado costo computacional. Ahora si el número de puntos de datos en el espacio de imágenes es menor que la dimensión del espacio ($M < N^2$), habrá solamente M - 1, *eigenvectores* significativos, los demás eigenvectores están asociados a los eigenvalores de cero.

Lo anterior se soluciona encontrando los eigenvectores v_i de la matriz L (MxM) donde L tiene la siguiente forma y propiedad como se muestra en la Ecuación 1.8:

$$L = A^T A \quad , \quad A^T A v_i = \mu_i v_i \quad , \quad L = \mu_i \qquad \text{Ec. (1.8)}$$

donde μ_i hace referencia al eigenvalor de la matriz L.

Continuadamente se observa en la Ecuación 1.9, la relación que existe entre u_i y v_i, cabe resaltar que es importante diferenciar bien las variables u_i, v_i, μ_i.

$$A^T A v_i = \mu_i v_i \rightarrow AA^T A v_i = \mu_i A v_i \rightarrow$$
$$CAv_i = \mu_i Av_i \quad o \quad Cu_i = \mu_i u_i \quad \text{donde } u_i = Av_i \qquad \text{Ec. (1.9)}$$

Observando que $L_{Mn} = \Phi_M^T \Phi_n$ y que en esta se encuentran los M eigenvectores v_k de L. Estos vectores determinan la combinación lineal de las M imágenes del conjunto de caras de entrenamiento para formar las *eigenfaces* u_i como se observa en la Ecuación 1.10.

$$u_l = \sum_{k=1}^{M} v_{lk} \Phi_k \; ; \quad \text{para } l = 1, \ldots, M \qquad \text{Ec. (1.10)}$$

De los M eigenvectores que se encuentren con la anterior Ecuación, se calculan los K mejores de ellos que maximicen la Ecuación 1.11, para el eigenvalor.

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}(u_k^T \Phi_n)^2 \qquad \text{Ec. (1.11)}$$

Sea máximo, sujeto a:

$$u_l^T u_k = \delta_{lk} = \begin{cases} 1 & \text{si } i=k \\ 0 & \text{en otro caso} \end{cases} \qquad \text{Ec. (1.12)}$$

Donde Los vectores u_k y los escalares λ_k, son los *eigenvectores* y *eigenvalores* proporcionalmente de la matriz de covarianza

Para explicar las Eigenfaces en términos simples, se dice que las Eigenfaces muestran las principales diferencias entre las imágenes de entrenamiento, y luego representan cada imagen de entrenamiento mediante una combinación de esas diferencias. En la Figura 6, se observan las Eigenfaces del sistema de reconocimiento facial.

Figura 6. Diferentes eigenfaces de la base de datos

Así, por ejemplo una de las imágenes de entrenamiento puede estar compuesta de: (imagen media) + (14.5% de la eigenface0) - (35.3% de la eigenface1) + (5.7% de la eigenface2) +…+ (0.05% de la eigenface299).

Seguidamente se encuentran los pesos, como se sabe, cada imagen del conjunto de pruebas puede ser representado como una combinación lineal de los K mejores eigenvectores (eigenfaces), donde cada eigenface viene asociado a un peso (w_j); entonces si se tiene K eigenfaces por consiguiente se tendría K pesos, los cuales forman un vector de longitud Kx1; para calcular los pesos se hace uso de las siguientes formulas:

$$\Phi_i = \sum_{j=1}^{k} w_j u_j \qquad \text{Ec. (1.13)}$$

$$w_j = u_j^T \Phi_i \qquad \text{Ec. (1.14)}$$

$$donde\ u_j = eigenfaces.$$

$$\Omega_i = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix} \qquad \text{Ec. (1.15)}$$

El proceso de reconocimiento se hace mediante una simple operación como se muestra en la Ecuación 1.16, dada una imagen de entrada X es el siguiente:

$$w_k = u_k^T(X - \Psi) \qquad \text{Ec. (1.16)}$$

Para k = 1,..., M, obteniéndose un conjunto de pesos que conforman el vector $\Omega = [\omega_1, \omega_2, ..., \omega_m]$ que definen los pesos de la imagen de entrada proyectada en los eigenvectores. El valor de Ω sería el que minimice la distancia euclídea como se muestra en la Ecuación 1.17.

$$\varepsilon_K^2 = ||(\Omega - \Omega_k)||^2 \qquad \text{Ec. (1.17)}$$

Donde Ω_k es el vector que describe la k-ésima cara.

1.5.4 Técnica de K Vecinos más cercanos. Esta técnica se ha utilizado desde hace mucho tiempo para el reconocimiento de patrones. Ya que esta técnica es muy sencilla, rápida y se puede aplicar a todos los algoritmos que se utilicen para buscar por aproximación y eliminación ofreciendo una excelente información sobre el problema de clasificación que se quiere tratar según Moreno Seco[7].

Los clasificadores que utilizan los vecinos más cercanos han sido aplicados para el reconocimiento facial. El éxito de este clasificador tan sencillo se basa en que la base de datos pensada depende de la luz, orientación y pose, ya que este método no es invariante frente a estas causas. Esta técnica es utilizada cuando las imágenes son procesadas con el algoritmo de PCA y Fisherfaces. Estos métodos tienen buenos resultados cuando las imágenes de ejemplo son parecidas a las imágenes de entrenamiento, pero la desventaja que tienen es que la iluminación o los gestos afectan la eficiencia del sistema.

_____

[7] Moreno Seco, Francisco. Clasificadores eficaces basados en algoritmos rápidos de búsqueda del vecino más cercano, 2004, p.17.

En cada paso de este método, se selecciona una imagen a vecino más cercano por medio de una función de aproximación, y se halla su distancia a la de la muestra X. Si esta distancia es menor que la del vecino más cercano hasta el momento, se actualiza su valor y se excluyen todos lo ejemplos del conjunto de entrenamiento que no puedan estar dentro de una esfera de radio X centralizada en la muestra; para determinar si un ejemplo puede estar o no en la esfera se suele utilizar una medida inferior de su distancia a la muestra según María Luisa Mico[8].

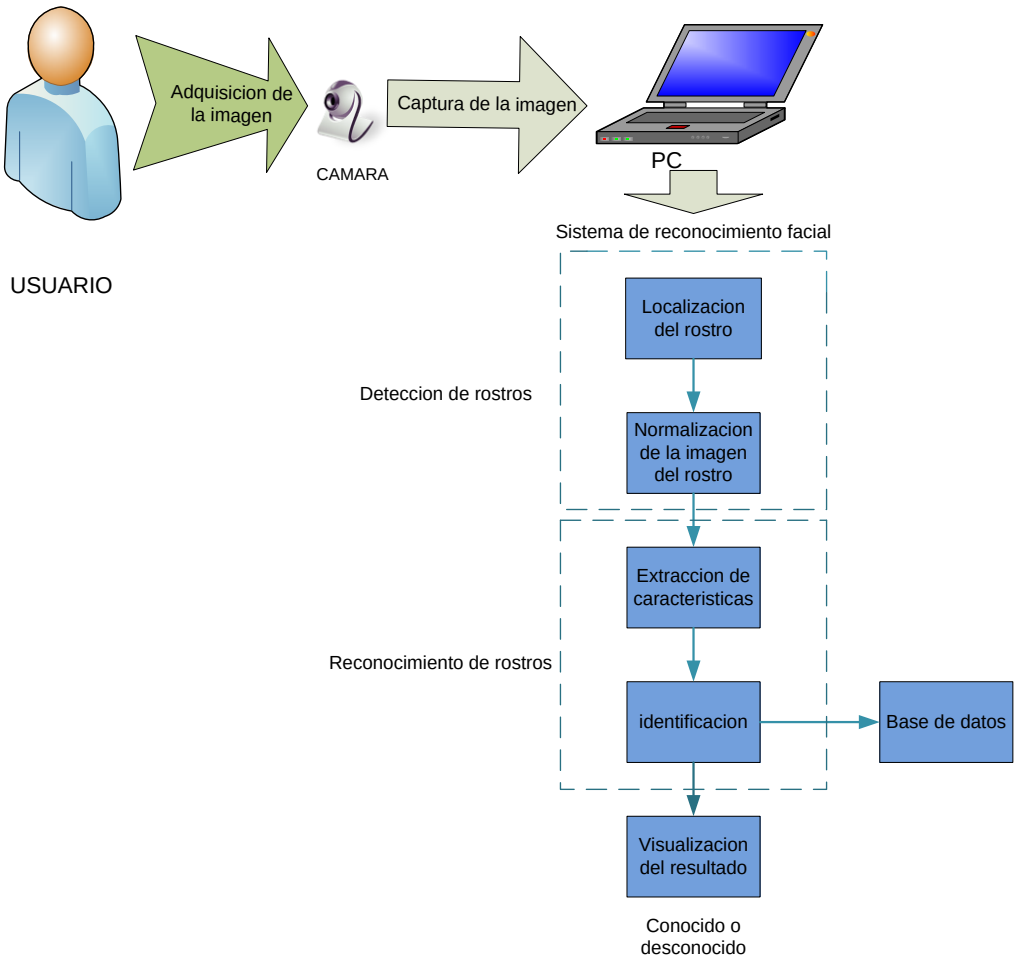
[8] Mico Andrés, María Luisa. Algoritmos de búsqueda de vecinos más próximos en espacios métricos, 1996, p.9-10.

2. DISEÑO DEL SISTEMA DE RECONOCIMIENTO FACIAL

Para el diseño del sistema de reconocimiento facial en un ambiente controlado, se parte de la visión general, de que el sistema debe adquirir la imagen del usuario mediante una cámara web, luego la aplicación o software debe localizar el rostro en la imagen, normalizarlo, adquirir sus características e identificar al usuario si se encuentra en la base de datos, mostrando el resultado del reconocimiento al usuario. En la Figura 7, se observa el diagrama de bloques general del sistema de reconocimiento facial que muestra las acciones mencionadas anteriormente.

El sistema de reconocimiento facial se desarrolla en Microsoft Visual C++ 2008 Express Edition con ayuda de la librería OpenCV versión 2.3.1 que son librerías creadas para aplicaciones de visión artificial en tiempo real.

Figura 7. Diagrama de bloques general del sistema de reconocimiento facial



2.1 REQUERIMIENTOS DEL SISTEMA

Una parte importante a la hora de realizar el diseño del sistema de reconocimiento facial, son los requerimientos que se definen como “una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo”. Teniendo en cuenta lo anterior, se analizan los requerimientos respecto a las necesidades que debe cumplir el sistema de reconocimiento facial para que tenga una buena fiabilidad, autenticidad y robustez. Para tener una mejor claridad en el diseño se dividen los requerimientos en funcionales y no funcionales. En los requerimientos funcionales se definen las funciones que el sistema es capaz de realizar, mientras que los requerimientos no funcionales se relacionan con características que pueden limitar el sistema, tales como la fiabilidad, el rendimiento, la interfaz de usuario, etc.

A continuación se definen los requerimientos del sistema de reconocimiento facial.

2.1.1 Requerimientos funcionales. El sistema debe:

- Establecer una conexión exitosa con la cámara web para adquirir la imagen de entrada del usuario.

- Localizar el rostro en la imagen, teniendo en cuenta factores externos, como la iluminación, la distancia de la cámara, el fondo, etc., generando una subimagen normalizada del rostro detectado.

- Identificar o rechazar la imagen normalizada del usuario dependiendo si se encuentra o no en la base de datos.

- Modificar la base de datos mediante la adicción y eliminación de usuarios.

2.1.2 Requerimientos no funcionales.

- Se requiere que la cámara se encuentre ubicada correctamente, sin obstáculos. La iluminación del lugar donde se ejecute el sistema debe presentar características iguales o similares durante el día, debido a que el sistema se pretende diseñar para ambientes controlados.

- el rostro del usuario al momento de tomar la foto, debe encontrarse frente a la cámara, con una posición definida y sin hacer ningún tipo de gesto.

- La cámara debe tener características aceptables de resolución y velocidad.

- Los cómputos deben ser ejecutados eficientemente, para obtener un buen rendimiento del sistema de reconocimiento facial.

2.2 CASOS DE USO

"El caso de uso es una estructura para describir la forma en que un sistema lucirá para los usuarios potenciales. Es una colección de escenarios iniciados por una entidad llamada actor (una persona, un componente de hardware, un lapso u otro sistema)". En esta sección se explican las funciones y operaciones que realiza el sistema de reconocimiento facial en un ambiente controlado al usuario, mediante diagramas de casos de uso como se muestra en la Figura 8.

A continuación se realiza la descripción textual de cada caso de uso que envuelve al sistema de reconocimiento facial en las siguientes tablas.

Tabla 1. Detectar rostro

Caso de uso: Detectar Rostro
Actor: Usuario
Descripción: Adquiere la imagen del rostro
Caso normal:
1) El módulo inicializa la cámara web, la configura y realiza la captura de la imagen almacenándola en una variable del sistema. 2) Localiza el rostro en la imagen, mediante el uso de clasificadores en cascada, específicamente se utilizaran clasificadores Haar, que son manejados por las librerías de OpenCv.

Tabla 2. Prepocesar imagen

Caso de uso: Preprocesar la imagen
Actor:------
Descripción: se normaliza la iluminación y el tamaño de la imagen del rostro.
Caso normal:
1) Se toma la matriz de la imagen del rostro y se ecualiza su histograma para normalizar la iluminación. 2) Se normaliza el tamaño de la imagen mediante una interpolación bilineal, para reducir el costo computacional en el reconocimiento y reducir el espacio de memoria que ocupan las imágenes que se almacenan en la base de datos. 3) Se crea la subimagen del rostro localizado.

Figura 8. Diagrama de caso de uso del sistema de reconocimiento facial

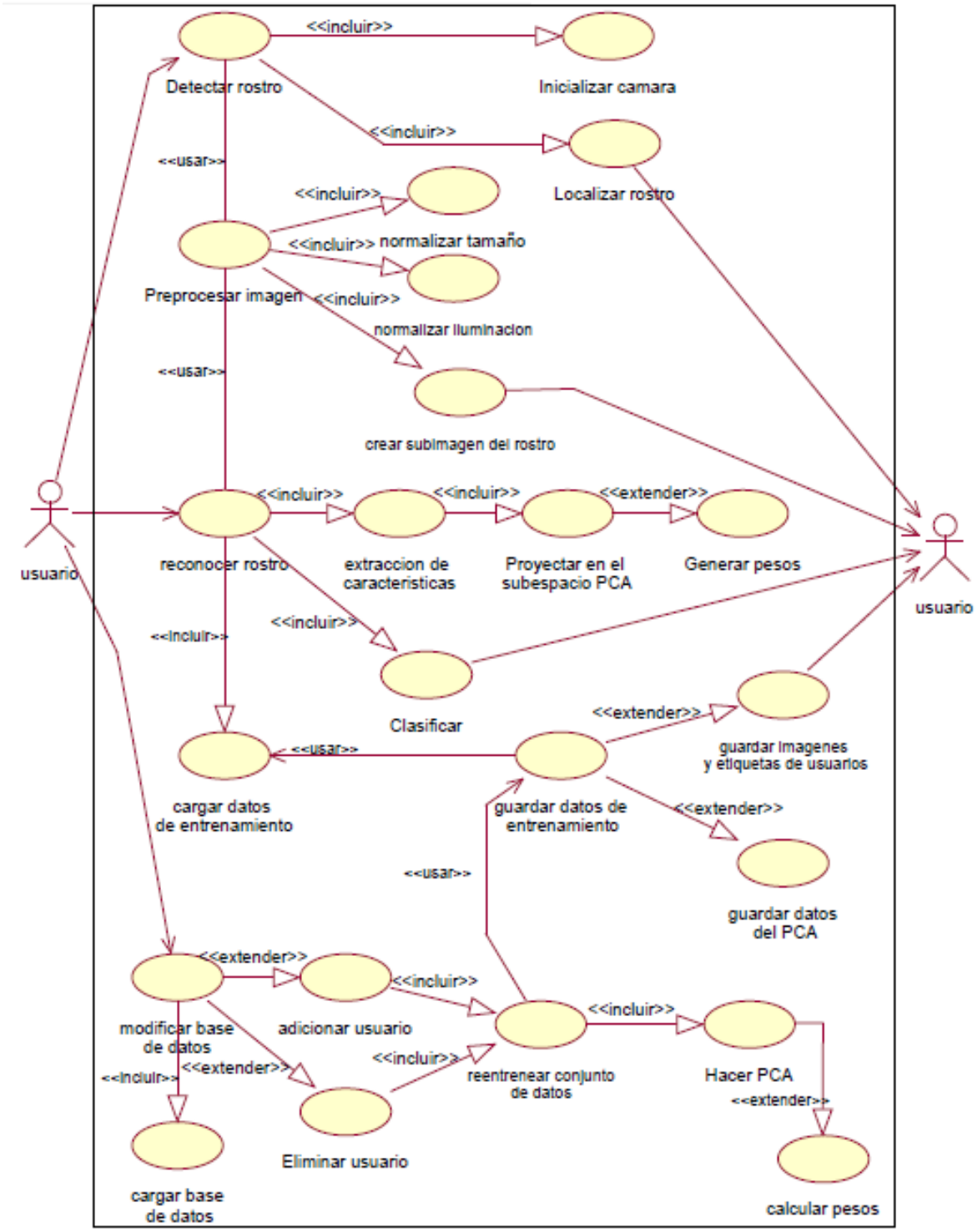


Tabla 3. Reconocer rostro

Caso de uso: Reconocer Rostro
Actor: Usuario
Descripción: se identifica la imagen del rostro.
Caso normal:
1) Se cargan los datos de entrenamiento de la base de datos necesarios para realizar la proyección en el subespacio PCA y la clasificación del peso de la imagen de entrada (Eigenvectores, eigenvalores, la media y pesos de las imágenes de entrenamiento). 2) Se extraen las características de imagen de entrada proyectándolas en el subespacio PCA y calculando su respectivo peso. 3) Se clasifica la imagen comparando su peso con los pesos de las imágenes de la base de datos.

Tabla 4. Tabla de modificación de la base de datos

Caso de uso: Modificación de la base de datos.
Actor: Usuario.
Descripción: Se añaden o eliminan usuarios del sistema.
Caso normal:
1) Se carga la base de datos del sistema con las imágenes de entrenamiento, sus respectivas etiquetas. 2) Se adiciona o elimina el usuario según sea el requerimiento. 3) Se entrena el sistema realizando el PCA para obtener los nuevos datos de entrenamiento (eigenvectores, eigenvalores, la media y pesos del conjunto de imágenes de entrenamiento). 4) Se guardan los datos de entrenamiento y las imágenes con sus respectivas etiquetas.
Alternativa 1: Adición de usuario. 2.1) Se agrega la imagen y la etiqueta del usuario al conjunto de imágenes de entrenamiento y lista de etiquetas respectivamente.
Alternativa 2: Eliminación de usuario. 2.2) Se elimina la imagen de usuario y su etiqueta del conjunto de imágenes de entrenamiento y lista de etiquetas respectivamente.

2.3 DIAGRAMA DE CLASES

Un diagrama de clase muestra un conjunto de clases, sus atributos y sus relaciones modelando la vista de diseño estática de un sistema. Se utiliza el diagrama de clases para diseño del sistema de reconocimiento facial, creando el diseño conceptual de la información que se manejará en el sistema, los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

En la Figura 9, se observa el diagrama de clases diseñado para el sistema de reconocimiento facial en donde se definieron 6 clases: Video capture, Cascade classifier, Preprocessing, PCA, CvKnearest y FileStorage.

La clase Video Capture engloba las operaciones necesarias para la adquisición de la imagen de entrada, realizando la inicialización de la cámara con “cvCaptureFromCAM()”, luego se configura la resolución de la captura mediante “cvSetCaptureProperty”, por último se realiza la captura de video mediante “cvQueryFrame”.

La clase “CascadeClassifier” incluye las operaciones para la detección del rostro, su función principales localizar los rostros en la imagen de la captura realizada previamente en múltiples escalas, para realizar esto utiliza la función “face_cascade.detectMultiScale()” valiéndose del clasificador de rostros "haarcascade_frontalface_alt.xml".

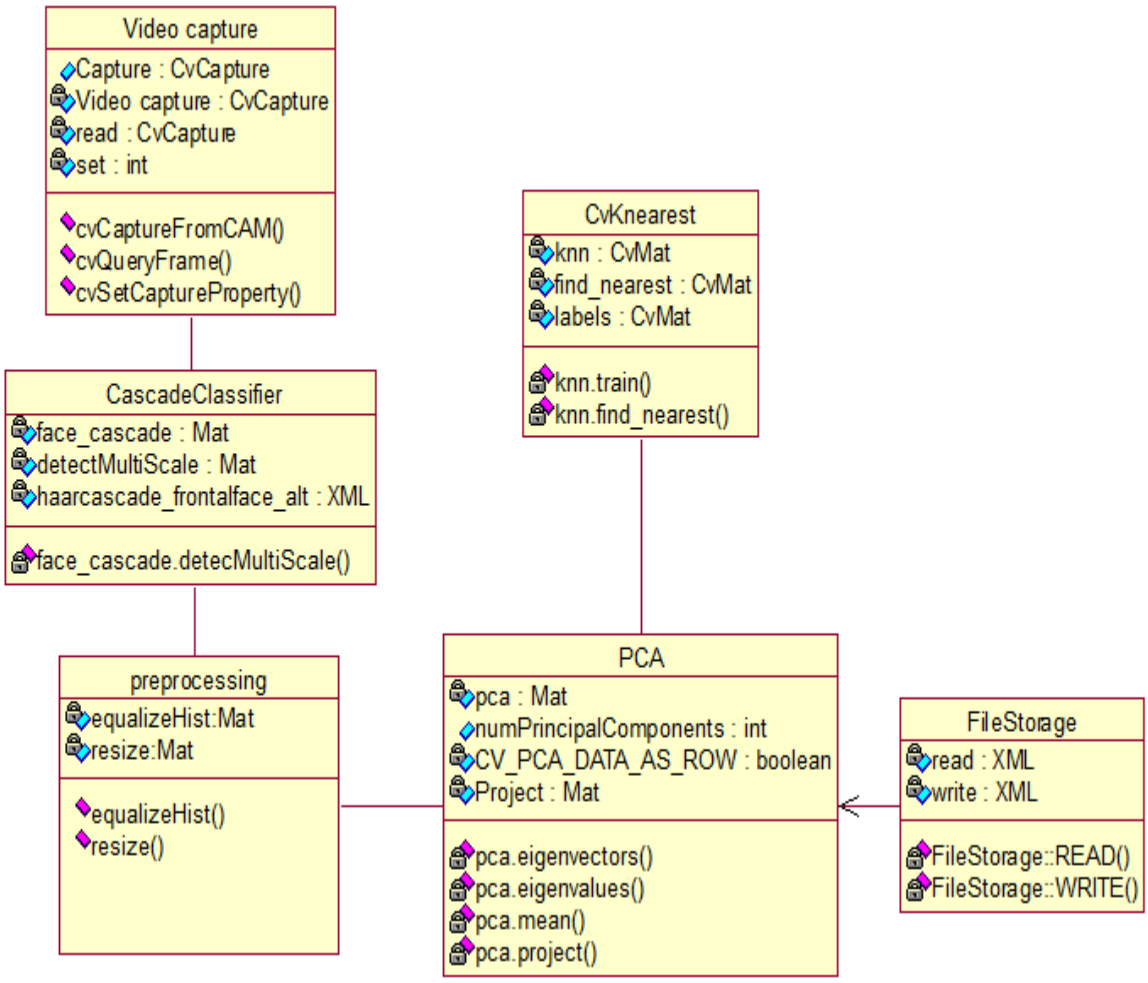
La clase preprocessing se utiliza para realizar la operación de equalizacion del histograma haciendo uso de la funcion “equalizeHist()” y para normalizar el tamaño de la imagen del rostro mediante la funcion “resize()”.

La clase PCA engloba las funciones encargadas de realizar el Analisis de Componentes Principales (PCA), como es generar los Eigenvectores, los Eigenvalores, la media de las imágenes y proyecccion de las imágenes en el espacio PCA, esto lo realiza con las funciones “pca.eigenvector”, “pcaeigenvalues”, “pca.mean” y “pca.project”, respectivamente.

La clase CvKnearest es la encargada de las operaciones de clasificacion, le asigna etiquetas a los pesos de las imágenes de entrenamiento con la funcion “knn.train()” y luego clasifica la imagen de entrada con la imagen de entrenamiento que tenga el valor de peso similar usando la funcion “knn.find_nearest()”.

La clase FileStorage se encarga de escribir y leer los datos de entrenamiento de la base de datos usando respectivamente las funciones “FileStorage::WRITE()” y “FileStorage::READ()”.

Figura 9. Diagrama de clases del sistema de reconocimiento facial



2.4 DIAGRAMA DE SECUENCIA

Un diagrama de secuencia se utiliza para observar la interacción ordenada entre las clases según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba hacia abajo.

En la Figura 10, se observa el diagrama de secuencia de la operación de identificación del sistema de reconocimiento facial. La operación de identificación

se inicia cuando el usuario activa la detección del rostro empezando el funcionamiento de la clase “video capture”, encargada inicializar la cámara, configurarla y realizar el proceso de captura de la imagen. Luego la clase “CascadeClassifier” recibe la imagen capturada y le aplica el clasificador en cascada que utiliza detectores haar entrenados, para localizar el rostro en la imagen dando como resultado la matriz del rostro, dicha matriz del rostro pasa a la clase “Preprocessing” que tiene como función ecualizar el histograma y normalizar el tamaño de dicha matriz creado una subimagen preprocesada del rostro. La subimagen mencionada anteriormente pasa a la clase “PCA”, esta clase necesita que previamente la clase “FileStorage” cargue los eigenvectores, Eigenvalores y la imagen media del conjunto de imágenes de la base de datos para proyectar dicha subimagen en el subespacio PCA, esta proyección genera el peso de la imagen que posteriormente es trasmitido a la clase “CvKnearest”, que se encarga en primera instancia de tomar los pesos de las imágenes de la base de datos que son cargados mediante la clase “FileStorage”, les asigna etiquetas a cada uno y luego realizar la operación de clasificación encontrando la imagen de la base de datos que más se asimile a la imagen de entrada mediante el valor de sus pesos clasificándola con la respectiva etiqueta.

Finalmente si la diferencia entre los pesos más cercanos es menor que el umbral de reconocimiento el sistema mostrará la etiqueta al usuario de lo contrario si la diferencia entre los pesos más cercanos supera el umbral de reconocimiento el sistema mostrará que el usuario no es conocido.

En la Figura 11, se observa la operación de entrenamiento del sistema de reconocimiento facial, esta etapa consiste en la creacion de los datos de entrenamiento que el sistema necesita para la identificacion de un usuario, e inicia guardando las imágenes de los usuario del sistema denomindas imágenes de entrenamiento y sus respectivas etiquetas con la clase “FileStorage”, luego con esta misma clase se cargan los datos anteriormente mencionados, para realizar el Analisis de los Componentes Principales con la clase “PCA”,generando la media, los Eigenvalores, Eigenvectores, y pesos de proyeccion en el subespacio PCA, de las imágenes de entrenamiento, finalmente estos datos se guardar haciendo uso nuevamente de la clase “FileStorage”, para que puedan ser llamados en la operación de identificacion. Esta operación es realizada originalmente por los desarrolladores del sistema de reconocimiento facial para crear la base de datos, pero tambien es una operación que realiza el sistema cuando el usuario modifica la base de datos existente, adicionando o eliminando un usuario de esta.

Figura 10. Diagrama de secuencia del sistema de reconocimiento facial

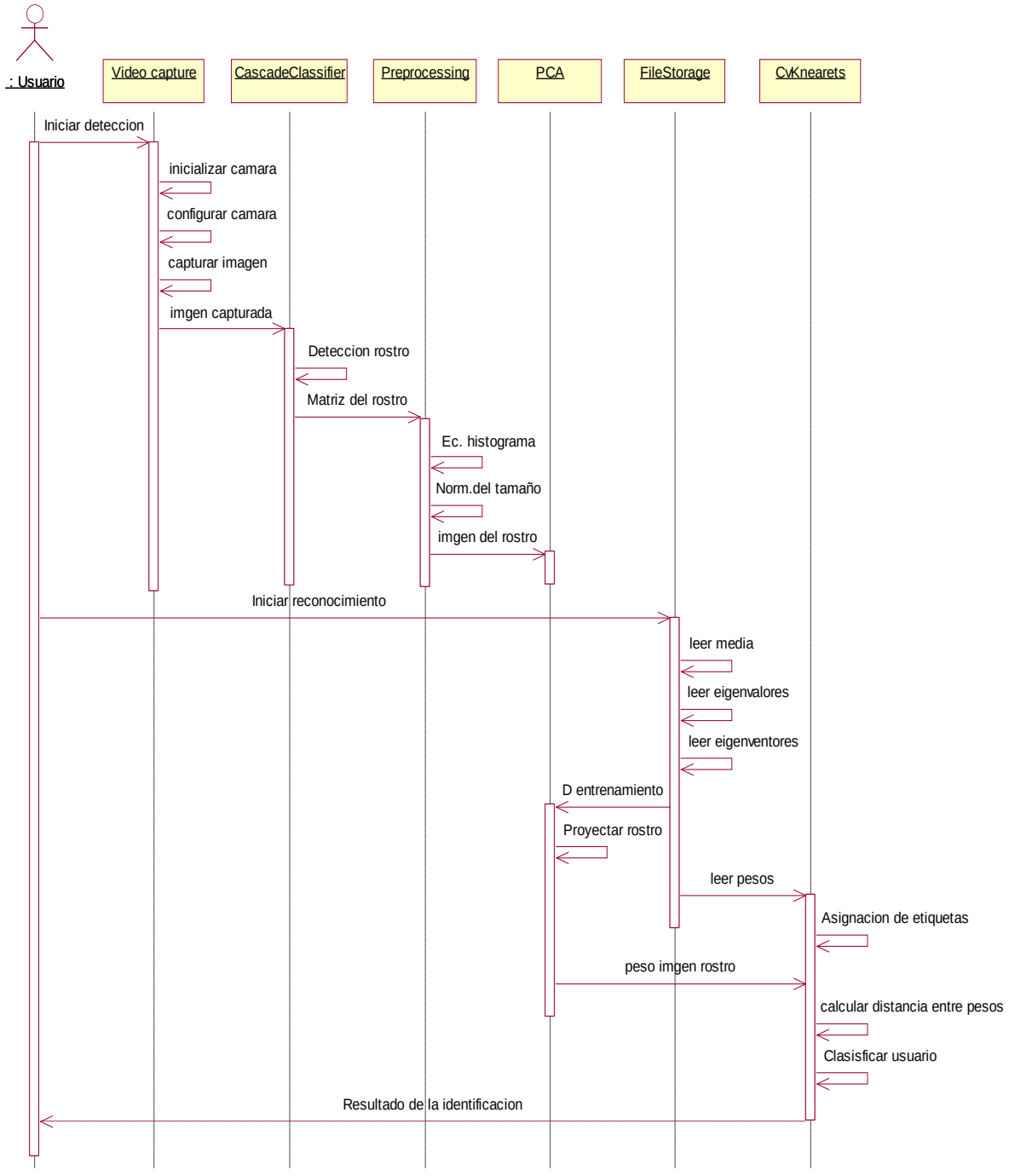
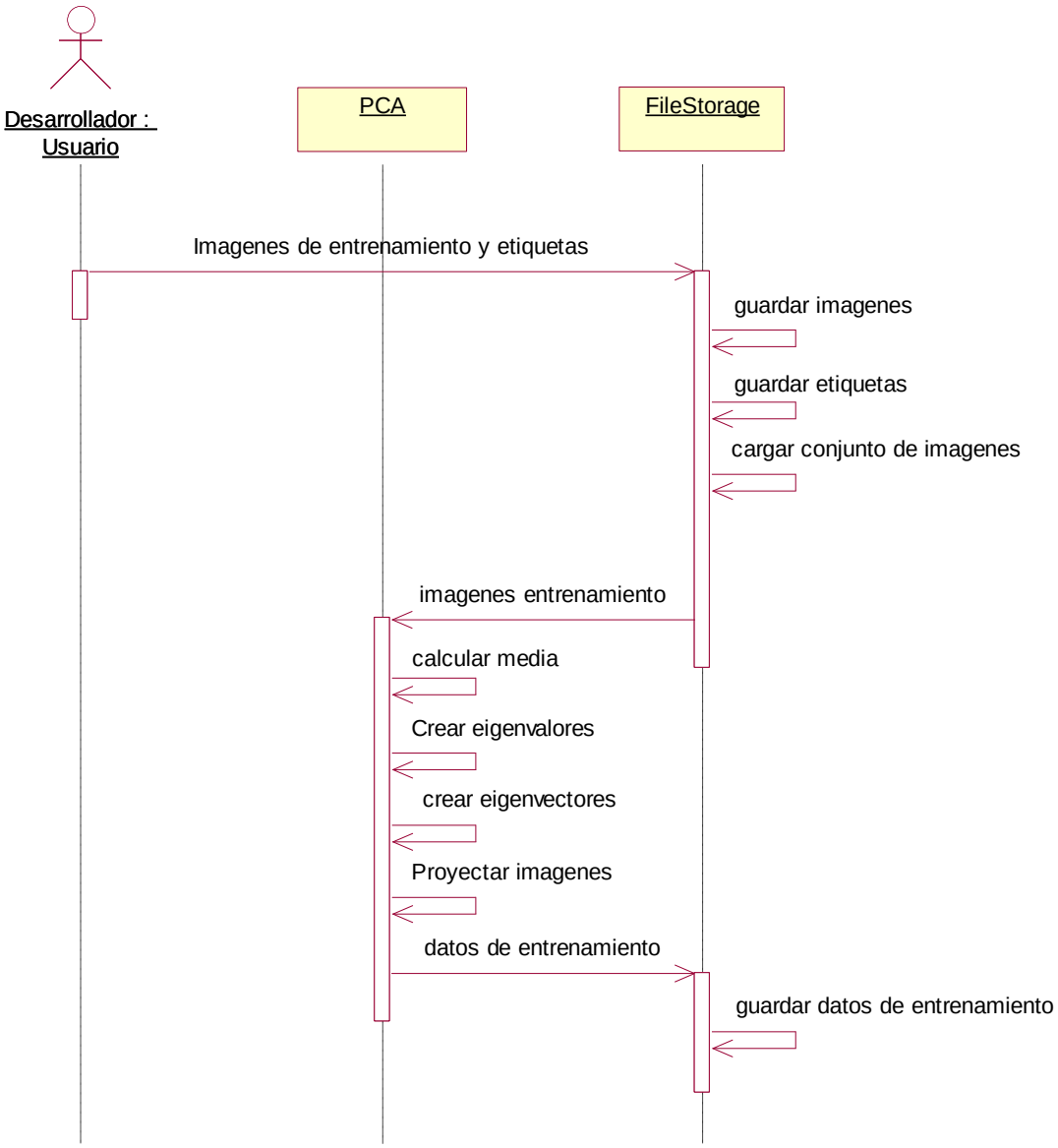


Figura 11. Diagrama de secuencia de la operación de Entrenamiento del Sistema de Reconocimiento Facial



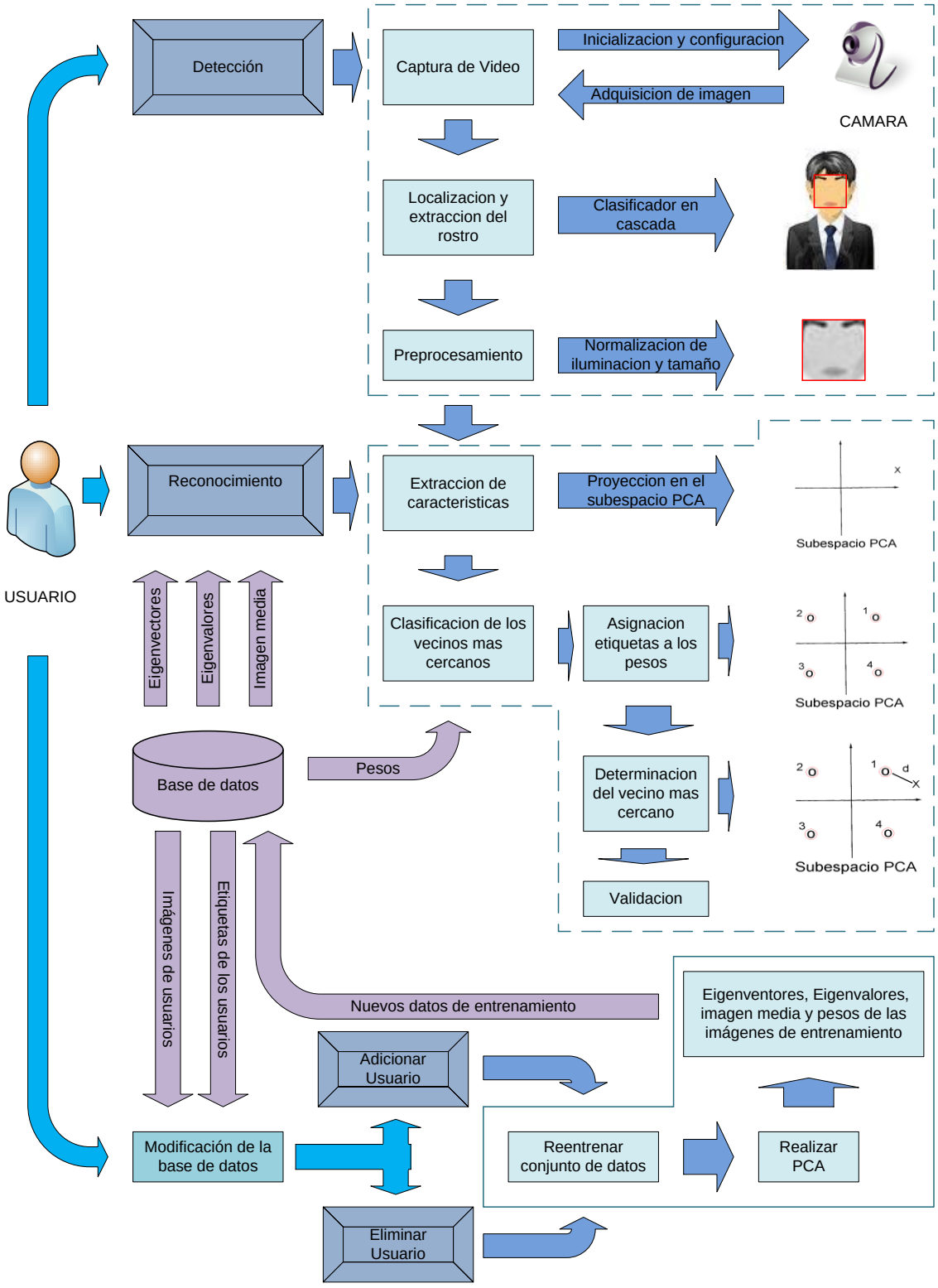
3. IMPLEMENTACIÓN DEL SISTEMA DE RECONOCIMIENTO FACIAL

El Sistema de Reconocimiento Facial en un ambiente controlado como se muestra en la Figura 12, consta básicamente de tres módulos principales: Detección, Reconocimiento y modificación de la base de datos. El módulo de Detección se encarga de realizar la captura de video, la localización del rostro haciendo uso del clasificador en cascada para la detección de rostro de OpenCV y el pre procesamiento que consiste en la normalización de la iluminación y tamaño de la imagen del rostro localizado. El módulo de Reconocimiento realiza la extracción de las características de la imagen de rostro detectada, cargando desde la base de datos los datos de entrenamiento necesarios para realizar la proyección en el subespacio PCA (Eigenvectores, Eigenvalores e imagen media) que genera el peso característico de dicha imagen. Luego se clasifica la imagen mediante la determinación de su vecino más cercano, este procedimiento empieza cargando los pesos de las imágenes de entrenamiento y asignándole etiquetas, para luego comparar el peso de la imagen de entrada con los pesos de las imágenes de entrenamiento, clasificándola con la etiqueta de su vecino más cercano, validando el reconocimiento si su distancia de cercanía es menor que el umbral de reconocimiento.

Finalmente se tiene el módulo de modificación de la base de datos, que presenta dos opciones: la adición y eliminación de usuarios, para realizar este proceso se carga en primer lugar las imágenes y etiquetas de los usuarios de la base de datos, si se requiere adicionar a un usuario el sistema agrega la imagen y la etiqueta al conjunto de imágenes y lista de etiquetas respectivamente. Si por el contrario se opta por eliminar un usuario el sistema busca la imagen del respectivo usuario mediante su etiqueta, una vez encontrado elimina la imagen del conjunto de imágenes y la etiqueta de lista de etiquetas de la base de datos, luego sin importar si se adiciono o elimino un usuario el sistema se reentrena realizando el PCA para generar los nuevos datos de entrenamiento (Eigenvectores, Eigenvalores, Imagen media y pesos), debido a que la adición o eliminación de una imagen cambia los componentes principales que caracterizan el conjunto de imágenes, luego los nuevos datos de entrenamiento, el nuevo conjunto de imágenes y la nueva lista de etiquetas son almacenados en la base de datos remplazando los datos anteriores. Los datos de entrenamiento se almacenan con el fin de reducir el tiempo de ejecución del sistema a la hora de identificar un usuario, esto hace que el módulo de reconocimiento acceda a los datos de entrenamiento desde la base de datos y no tenga la necesidad de realizar el proceso matemático para generarlos.

El Sistema de Reconocimiento Facial se realiza en Microsoft Visual C++ 2008 Express Edition con ayuda de la librería OpenCV versión 2.3.1 desarrolladas originalmente por Intel para aplicaciones de visión artificial.

Figura 12. Diagrama de bloques del sistema de reconocimiento facial

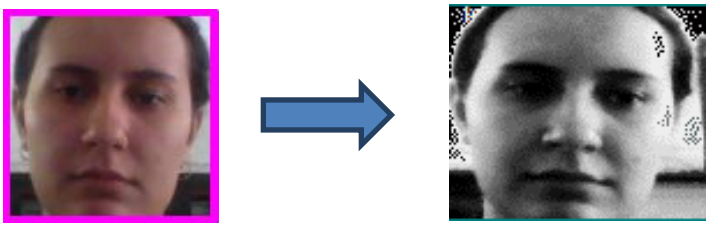


3.1 MÓDULO DE DETECCIÓN

El bloque de Detección se encarga en primer lugar de hacer la captura de video utilizando la clase "**Video Capture**", que inicializa la cámara con la función "**cvCaptureFromCAM()**", estableciendo desde código la resolución de la respectiva cámara haciendo uso de "**cvSetCaptureProperty()**", Posteriormente se captura la imagen de video con "**cvQueryFrame()**" y se almacena en una matriz variable denominada "**frame**".

Luego se aplica la función "**FaceDetect()**" a la variable "**frame**", esta función realiza la tarea conjunta de localización, extracción y pre procesamiento del rostro haciendo uso de la clases "**CascadeClassifier**" y "**Preprocessing**". La función "**FaceDetect()**" se inicializa si se detecta el "**frame**", e inmediatamente carga el archivo "**haarcascade_frontalface_alt.xml**" que es el Haar cascade classifier o clasificador en cascada para la detección del rostro de OpenCV, posteriormente con la función "**cvtColor()**" se procede a pasar a escala de grises el **frame,** como se observa en la Figura 13, debido a que el clasificador en cascada trabaja con este único canal, también se ecualiza el histograma para mejorar la detección, luego se aplica al "**frame**" la función "**CascadeClassifier::detectMultiScale**" la cual detecta objetos (en este caso los rostros) de diferentes tamaños en la imagen de entrada, y devuelve cada objeto como una matriz rectangular, una vez se ha detectado el rostro la función muestra un rectángulo alrededor de ellas para que el usuario se percate de que se ha hecho una detección exitosa. El resultado de los procesos de captura de video y localización del rostro se observan en la Figura 14.

Figura 13. Imagen de entrada convertida a escala de grises



Consecutivamente para el preprocesamiento se toma la matriz del rostro detectado y como se observa en la Figura 15, se realza realizando la ecualización de su histograma con la función "**equalizeHist()**", luego para disminuir la carga del procesamiento que se llevara a cabo en el reconocimiento se normaliza el tamaño de la imagen del rostro a 128x128 mediante función "**resize()",** una vez realizada la normalización se crea la subimagen del rostro, se muestra en un "picturebox" y se guarda en disco. Ahora cabe resaltar que si se detecta más de un rostro el usuario debe percatarse de que el rostro visualizado en el "picturebox" sea el que se desea identificar.

Figura 14. Captura de video y localización del rostro

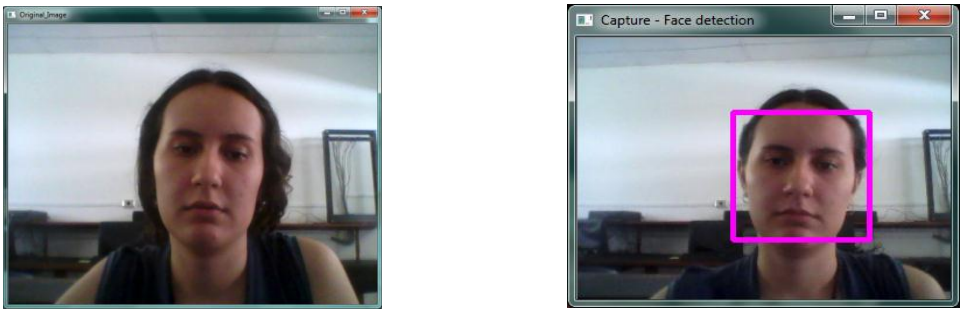
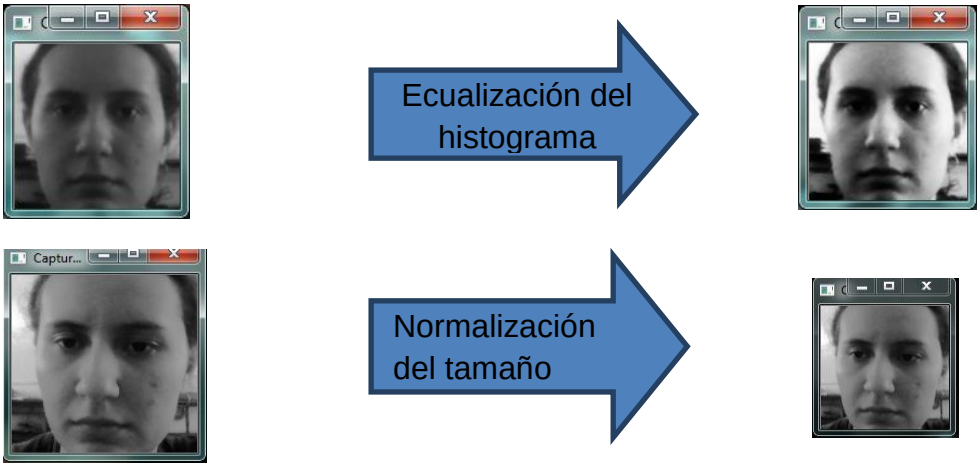


Figura 15. Ecualización del histograma y normalización del tamaño de la imagen del rostro



Finalmente se observa en la Figura 16, el resultado del módulo de Detección del sistema de reconocimiento facial en un ambiente controlado.

Figura 16. Resultado del módulo de Detección

3.2 MÓDULO DE RECONOCIMIENTO

El módulo de reconocimiento es el encargado de identificar al usuario mediante el rostro detectado, para realizar este proceso, el módulo extrae las características que se definen como un vector de pesos o valores que se obtienen de la proyección de la subimagen del rostro en el subespacio PCA y los clasifica respecto a los vectores de pesos de las imágenes de la base de datos. Para realizar este proceso el módulo de reconocimiento inicia haciendo uso de la clase "**FileStorage**", atreves de su función "**FileStorage::READ**" se cargan los datos de entrenamiento de la base de datos necesarios para realizar la proyección y clasificación de la subimagen del rostro detectado (Eigenvectores, Eigenvalores, la media, los pesos y etiquetas de las imágenes de entrenamiento), una vez se han cargado los datos mencionados anteriormente, se realiza la proyección en el subespacio PCA de la imagen de rostro detectada mediante la aplicación de la ecuación 3.1, que es creada a partir de la ecuación 1.16, que se encuentra en el capítulo 1, sección 1.3.3.

$$\text{Peso(i)} = \text{Eigenvector(i)} * (\text{rostro} - \text{media imagenes}) \quad \text{Ec. 3.1}$$

Con la ecuación anterior se genera un vector de pesos que caracteriza el rostro reduciendo su dimensionalidad, este vector de pesos tendrá un número de componentes igual al número de eigenvectores que hallan. Luego de haber calculado los pesos se utiliza la clase "**CvKnearest**" para realizar la clasificación del rostro detectado, en primer lugar se entrenan los pesos de las imágenes de entrenamiento cargados de la bases de datos con la función "**CvKnearest::knn()"**, asignándole inicialmente etiquetas numéricas para posteriormente clasificar el peso del rostro detectado con una de las etiquetas de los pesos de las imágenes de entrenamiento, esto se realiza recurriendo a la función "**CvKnearest::find_nearest()**", que mediante el método de los vecinos cercanos encuentra cual es el peso de las imágenes de entrenamiento que más se asimila al peso del rostro detectado, dando como resultado la diferencia entre el puntaje de la suma ponderada de los pesos más cercanos, y la etiqueta numérica que lo clasifica. Finalmente se realiza el proceso de validación que relaciona la etiqueta numérica con la etiqueta del nombre correspondiente y la visualiza al usuario, solo si la distancia resultante es menor que el umbral de reconocimiento. El valor del umbral de reconocimiento se determina realizando pruebas de identificación en el lugar donde se va a utilizar el sistema de reconocimiento facial, cuando el ambiente es controlado, es decir, el reconocimiento se realiza en el lugar donde se toman las imágenes de entrenamiento, el valor ideal del umbral es igual a 100, este resultado se justifica en los análisis de resultados que se encuentran en el siguiente capítulo.

3.3 MÓDULO DE LA MODIFICACIÓN DE LA BASE DE DATOS

La base de datos es un archivo de extensión XML llamado “trainingdata”, creado haciendo uso de la clase “**FileStorage**” con la función “**FileStorage::WRITE**”, este archivo contiene los datos de entrenamiento, que son: el conjunto de imágenes de usuarios del sistema, sus etiquetas o nombres, así como también, los Eigenvectores, los Eigenvalores, la media y pesos de proyección en el subespacio PCA que se generan de dichas imágenes de usuarios con la clase “**PCA**”. El sistema de reconocimiento facial en un ambiente controlado cuenta en su interfaz gráfica con un “listbox” que permite ver la lista de nombres de los usuarios que se encuentra en la base de datos y seleccionar, también tiene la propiedad de mostrar la imagen almacenada del usuario seleccionado en el listbox como se puede observar en la Figura 17.

Figura 17. Acceso a la base de datos

El módulo de modificación de usuario me permite adicionar o eliminar usuarios de la base de datos. El proceso empieza cargando todos los datos de entrenamiento de la base de datos con la función "**FileStorage::READ**". Cuando se realiza la operación de adición de un usuario, el sistema tiene como prerrequisito que se halla detectado la imagen del rostro del respectivo usuario y se halla consignado su etiqueta o nombre, porque si no se ha detectado el rostro del usuario o su etiqueta el sistema no podrá adicionarlo a la base de datos, suponiendo que se ha hecho la detección previa del rostro y se ha colocado su etiqueta, el módulo continua su operación adicionando la etiqueta a la lista de etiquetas y la imagen al conjunto de imágenes. En caso contrario de que la operación a realizar sea la de eliminar un usuario, el sistema toma la imagen de la etiqueta seleccionada en la interfaz gráfica y la elimina junto con la etiqueta creando un nuevo conjunto de imágenes.

Luego independientemente de cuál sea la operación realizada anteriormente (adición o eliminación de un usuario) el módulo realiza un proceso de reentrenamiento, a causa de que al modificar el conjunto de imágenes, cambian los componentes principales que las caracterizan, este proceso consiste en tomar el nuevo conjunto de imágenes y realizarle mediante la clase "**PCA**" el análisis de sus componentes principales, haciendo uso de la función "**pca()**" que genera los Eigenvectores, Eigenvalores y la media de las imágenes, el proceso continua generando también los pesos de las de proyecciones en el subespacio PCA de dichas imágenes utilizando para esto la función "**pca.project**". Una vez generados todos los nuevos datos de entrenamiento se actualiza la base de datos con la función "**FileStorage::WRITE**" que la almacena en disco.

3.4 AMBIENTE CONTROLADO

Se tuvo en cuenta que la iluminación del ambiente donde se va a utilizar el sistema, afecta la eficiencia del reconocimiento. Por lo tanto, esta aplicación funciona de forma óptima en un lugar cerrado donde se tenga preferiblemente iluminación artificial con el fin de reducir los efectos de sombra y brillos no deseados, mejorando de esta manera el contraste de la imagen. También se debe tener en cuenta que la fuente de iluminación artificial no debe estar al frente de la cara, debido a que se pueden presentar problemas con los brillos en la imagen del rostro.

Para que el sistema de reconocimiento facial tenga la mejor eficiencia, se debe actualizar la base de datos de los usuarios, en el sitio donde se instalará o ejecutará el programa de reconocimiento facial, esto hace que el reconocimiento sea muy preciso, puesto que el sistema se evalúa sobre el mismo ambiente en el que se toman las imágenes de la base de datos.

4. ANÁLISIS DE RESULTADOS

En este capítulo se mostrarán los resultados obtenidos en cuanto a la eficacia, fiabilidad, desempeño y robustez del sistema de reconocimiento facial diseñado de una manera cuantitativa, evaluando las etapas fundamentales en el proceso de reconocimiento.

Para el Sistema de Reconocimiento Facial se creó una base de datos con los cuatro usuarios para el reconocimiento y se tuvieron en cuenta tres usuarios que no se encuentran en dicha base, tratando de cumplir con ciertas condiciones que limitan el reconocimiento facial, como lo son:

- Requiere un fondo simple o una iluminación especial.
- El rostro se debe encontrar a cierta distancia de la cámara y en cierta posición.
- Que los rostros no estén haciendo gestos extraños, ni estén enmascarados.

4.1 ANÁLISIS DE LA ETAPA DE DETECCIÓN FACIAL

En la etapa de la detección facial se tomaron en cuenta más de cinco personas en diferentes entornos e iluminación para poder observar la eficacia de la función “FaceDetection". Para realizar la detección de rostro en el Sistema de Reconocimiento Facial, se utilizo el clasificador “haarcascade_frontalface_alt.xml” de OpenCV, para llevar a cabo este trabajo.

La detección tiene un 97% de eficacia, ya que a veces se presentan unos falsos positivos en la escena y en muy pocas ocasiones detecta algunos objetos que no corresponden a un rostro, debido a que en el entorno se presentan en ocasiones objetos o formas parecidas a la de un rostro. La etapa de detección funciona satisfactoriamente, puesto que detecta el rostro de toda persona sin importar sus gestos, color de la piel, tamaño de la cara, gafas que pueda tener dicha persona, entre otros como se muestra en la Figura 18.

La única limitante del bloque de detección es que no puede haber objetos que oculten partes representativas de una cara como los ojos, la nariz o la boca. Esto hace que se pierda la información que diferencia un rostro de otros objetos.

La cantidad de veces que se presentaron falsos positivos representa un 3% de las pruebas realizadas, cabe resaltar que este problema se minimiza totalmente si se tiene un ambiente controlado. En la Tabla 5, se observan los resultados obtenidos en las pruebas realizadas.

Figura 18. Resultado de la detección facial

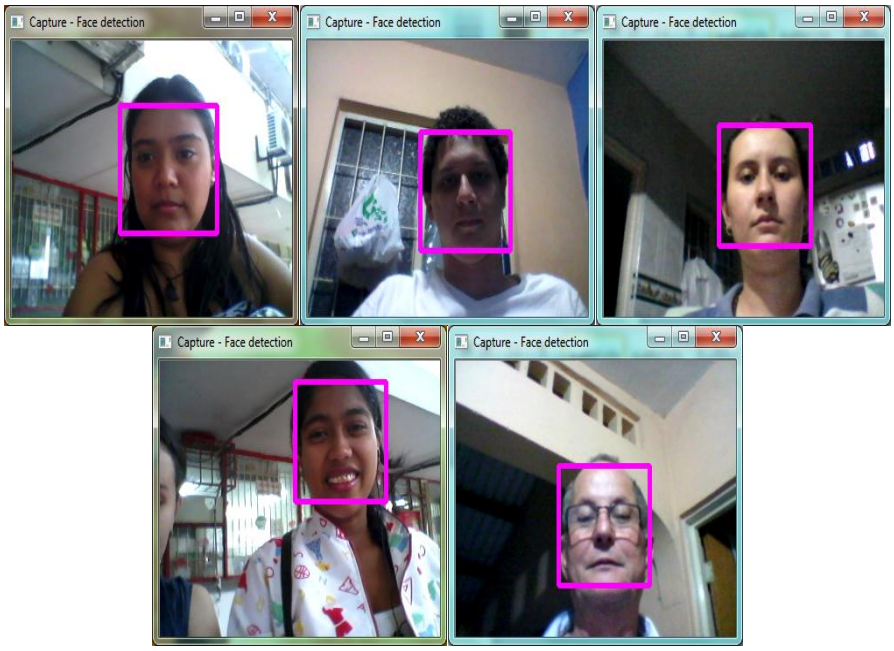


Tabla 5. Análisis de la detección facial

Objeto a detectar	% Detección de cada objeto
Rostro	97%
Falsos positivos	3%

4.2 ANÁLISIS DE LA ETAPA DE RECONOCIMIENTO FACIAL

El reconocimiento facial se diseñó en este caso para un ambiente controlado, teniendo en cuenta que la iluminación y el fondo del lugar donde se ejecuta el sistema debe ser igual o similar al entorno donde se han extraído las imágenes de la base de datos, adicionalmente, la posición de la cara de la persona a identificar debe estar derecha, encontrarse a un metro de la cámara, se debe además evitar gestos y obstrucción con objetos que hagan que se pierdan ciertas características del rostro. El sistema de reconocimiento facial acepta algunas variantes del rostro como lo son gafas y barba, siempre y cuando el rostro se encuentre con dichas características en la base de datos. Se hicieron pruebas en el sistema de reconocimiento facial con 7 personas de las cuales solo 4 se encontraban en la base de datos.

El sistema de reconocimiento facial diseñado utiliza como discriminador un filtro el cual define como desconocido el rostro entrada que tenga una diferencia de distancia entre dicha imagen y la imagen más cercana de la base de datos, mayor

al valor de este filtro, se estableció que el sistema de reconocimiento facial diseñado tiene un 100% de acierto con un filtro con valor igual a 200, cuando se identifica una persona en un ambiente controlado como se muestra en la Tabla 6.

Figura 19. Reconocimiento facial en un ambiente controlado

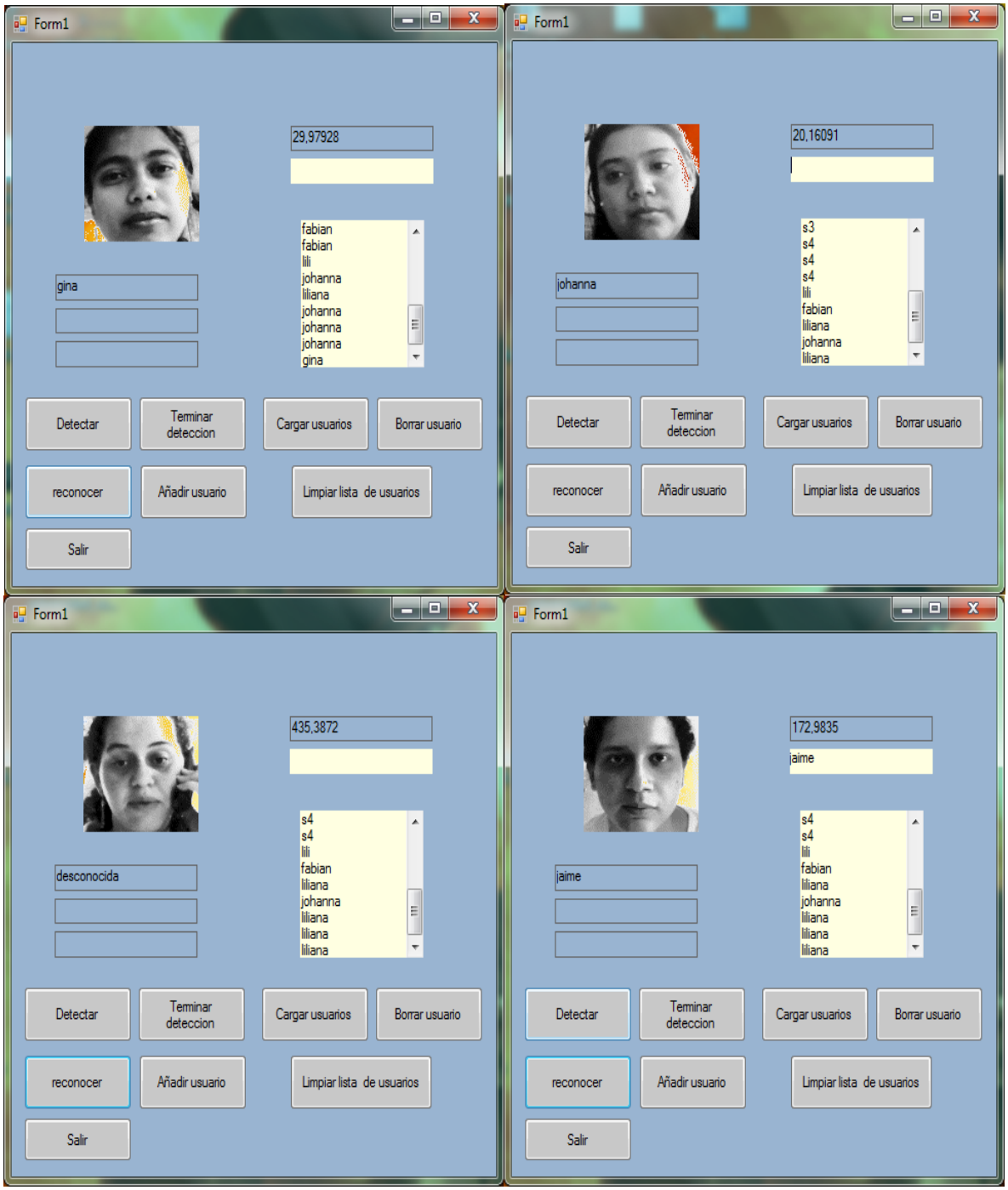


Tabla 6. Resultados del análisis de la etapa de reconocimiento facial en un ambiente controlado con el filtro con valor de 200

USUARIOS	EXITO	RECONOCIMIENTO CORRECTO
2	100%	20/20
3	100%	20/20
4	100%	20/20
7	100%	20/20

A continuación se analiza la variación de porcentaje de acierto a medida que se varía el filtro de reconocimiento. Los resultados de esta prueba se encuentran consignados en el Tabla 7.

Tabla 7. Reconocimiento facial en un ambiente controlado con diferente filtro.

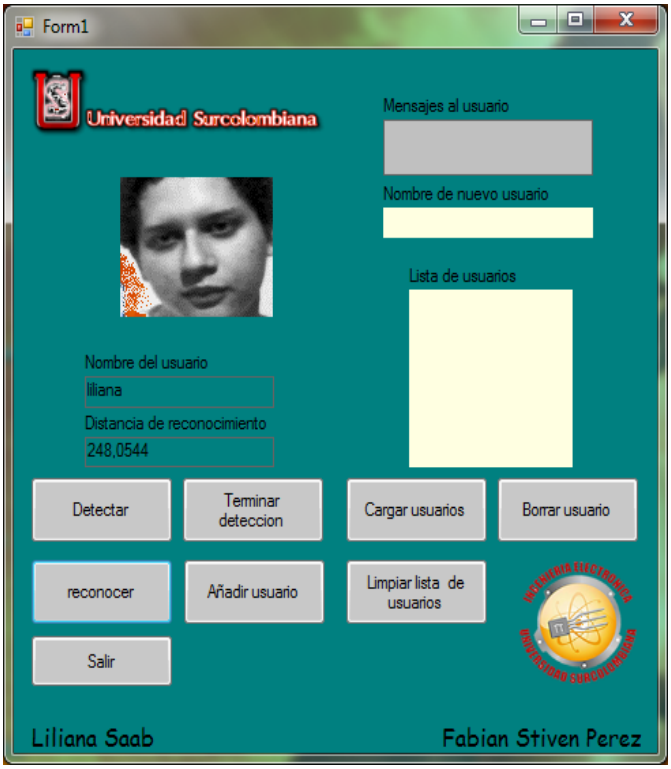
FILTRO	EXITO	RECONOCIMIENTO CORRECTO
100	90%	18/20
200	100%	20/20
300	75%	15/20
400	25%	5/20

Como se puede observar en el Tabla 7, cuando el filtro tiene un valor igual a 100, la tasa de aciertos baja un 10%, esto se debe a que las variaciones mínimas de la imagen de entrada del usuario a identificar respecto a la imagen que se encuentra en la base de datos de este mismo usuario, representan una diferencia mayor a 100, estas variaciones mínimas se presentan debido a la variación de la posición o de distancia del usuario a la cámara. Con el filtro en este valor el sistema no tuvo problemas para discriminar a las personas desconocidas.

Luego se observa que cuando el filtro de reconocimiento tiene un valor de 200, la tasa de aciertos es del 100%, por lo que se concluye que en el ambiente controlado donde se probo el sistema de reconocimiento facial las variaciones de la imagen de entrada del usuario respecto a su imagen almacenada en la base de datos, están representadas con una distancia de diferencia menor que 200, también al probarse con personas que no se encontraban en la base de datos el sistema de reconocimiento facial no tuvo ningún problema en etiquetarlas como desconocidas, por lo que se analiza que las personas desconocidas presentan una variación pesos de reconocimiento mayor a 200.

Siguiendo con las variaciones del filtro, cuando este tiene un valor de 300 la tasa de aciertos se reduce al 75%, la razón de esta reducción en la precisión del sistema de reconocimiento facial se debe a que el valor del filtro está muy elevado y etiqueta la persona desconocida con el nombre del usuario más cercano que tenga una diferencia menor de 300 respecto a la imagen que falsamente se supone como resultado de reconocimiento. A continuación en la Figura 20, se observa el problema descrito anteriormente.

Figura 20. Identificación de una persona que no se encuentra en la base de datos



Finalmente se analiza la última variación realizada en el filtro a el cual se le asigna un valor igual a 400, cuando el filtro tiene este valor la tasa de aciertos del sistema de reconocimiento facial baja críticamente a un 25%, este problema ocurre por la misma razón que cuando el valor del filtro es 300, el filtro es demasiado grande y deja pasar personas que no se encuentran en la base de datos, como personas conocidas, las variaciones entre los usuarios en la base de datos y las personas desconocidas no representa una variación mayor de 200, motivo por lo que se presentan estos falsos positivos.

Como se ha mencionado anteriormente, la iluminación del entorno afecta la fiabilidad del sistema de reconocimiento facial a la hora de realizar la identificación de la persona deseada, debido este sistema se diseñó para un ambiente controlado. A continuación se muestra el Tabla 8, donde se hace reconocimiento

facial en tres entornos con diferente iluminación, con la base de datos tomada en el ambiente controlado y el filtro de reconocimiento igual a 200.

Tabla 8. Reconocimiento facial con diferentes iluminaciones y fondos

ILUMINACIÓN	ÉXITO	RECONOCIMIENTO CORRECTO
Cuarto cerrado con iluminación artificial	100%	20/20
Cuarto con iluminación natural	85%	17/20
En un espacio abierto	50%	10/20

Inicialmente se realizan las pruebas en el ambiente controlado, el cual dispone de iluminación artificial, se observa nuevamente que el sistema da como resultado una tasa de aciertos del 100%.

Luego se somete el sistema de reconocimiento facial a un cuarto con iluminación natural, en este entorno la iluminación cambia un poco, por lo que se presentan variaciones de brillo en la imagen del usuario que se desea identificar, como el sistema no ha sido entrenado en estas condiciones de iluminación, la tasa de aciertos del mismo baja a un 85%, la solución a este problema es actualizar la base de datos en dichas condiciones de iluminación natural y los cambios que presente a medida que avanza el día.

Finalmente se ejecuta el sistema de reconocimiento facial en un espacio abierto, donde cambian drásticamente las condiciones de iluminación, respecto al ambiente controlado, allí se presenta una baja en la tasa de aciertos del 50%, la razón de esta baja en la fiabilidad es el cambio excesivo de brillo de las imágenes de entrada de usuario que se quiere reconocer en comparación con las imágenes de la base de datos, para minimizar este inconveniente se deben añadir un buen número de imágenes de cada usuario debido a que la variación mínima de pose o distancia a la cámara genera una variación significativa en el brillo de la imagen.

5. CONCLUSIONES

El reconocimiento facial se esta convirtiendo en una herramienta importante para la identificación de personas, debido a que la identificación se realiza de una manera cómoda y rápida, además con esta herramienta se puede aumentar la seguridad o el control de acceso a un lugar, al validar la identidad de una persona mediante los rasgos de la cara, los cuales son únicos e irrepetibles y no pueden ser olvidados o perdidos.

Hay muchos algoritmos para hacer el reconocimiento facial, donde algunos son muy buenos pero muy costosos, otros pocos costosos pero con muchas limitaciones, y otros como el PCA que fue el algoritmo utilizado que no es elevadamente costoso y tiene muy pocas limitaciones.

El reconocimiento facial se ve afectado por los cambios de iluminación, posición y distancia entre la persona y la cámara utilizada. Esto se debe a que las características de bordes, sombras, brillos y contraste de la imagen del rostro están dadas por la iluminación que incide sobre dicho rostro, por ende si cambian estas características en la imagen de entrada, difícilmente el sistema encontrará una imagen en la base de datos que se relacionen entre si.

Las imágenes capturadas que se utilizan en el sistema de reconocimiento facial deben ser normalizadas a un tamaño de 128*128, para que la ejecución del sistema sea más rápida y no se ocupe demasiado espacio de almacenamiento en disco. El tamaño de normalización no debe ser demasiado pequeño, debido a que se puede perder información de pixeles que caracterizan las imágenes y las hace diferenciable de las demás.

El sistema de detección de rostro funciona con una alta precisión, ya que no presenta problemas al detectar la cara en diferentes condiciones de iluminación, por lo que se puede concluir que el clasificador utilizado para esta función es robusto y confiable, aunque en muy pocas ocasiones si el ambiente no es controlado se detectan objetos o entornos que no son caras, como caras, aunque esto no representa ningún problema para el sistema, si la cara se encuentra en primer plano.

La robustez y precisión del sistema de reconocimiento facial depende enormemente de la variación de ruido que puede presentar las diferentes condiciones de iluminación del entorno donde se ejecute el sistema y a las pocas imágenes de muestra de cada usuario en la base de datos, por lo que se trata de minimizar este problema haciendo en lo posible que las condiciones de entorno e iluminación sean iguales o muy similares a las presentadas en la toma de las muestras de la base de datos. Esta solución da buenos resultados y se puede constatar en que el porcentaje de acierto es del 85% en la situación más

desfavorable. Otra solución que se puede implementar es agregar un mayor número de muestras en la base de datos tratando de abarcar el mayor número de diferentes condiciones de iluminación posibles presentadas al usuario, pero esto aumentaría el costo computacional a medida que se añaden nuevas imágenes.

Se mejoró el desempeño del Sistema Reconocimiento Facial, aplicando un buen preprocesamiento a las imágenes de entrada y de entrenamiento para reducir el ruido que puede generar la iluminación en dichas imágenes. Este preprocesamiento consiste en la ecualización del histograma que minimiza los brillos no deseados de las imágenes.

El Sistema de Reconocimiento Facial implementado en este trabajo de grado, no se compara con el sistema de reconocimiento facial innato de un humano, sin embargo da una idea de lo que puede traer la visión por computador en el futuro, y permite concluir que un sistema de reconocimiento facial robusto debe ser insensible a los cambios de iluminación, cambios de orientación de la cabeza y la escala, presencia de detalles faciales (como barba, gafas, gorras, etc.), y el ruido. Sin embargo, la aplicación creada es muy eficaz para detectar y reconocer rostros en ambientes controlados.

6. RECOMENDACIONES Y TRABAJO A FUTURO

En el transcurso del desarrollo de este trabajo de grado observamos limitaciones que se pueden evitar con las siguientes recomendaciones.

La primera recomendación es realizar un estudio exhaustivo de cada uno de los algoritmos que permiten efectuar la etapa de reconocimiento del rostro para diseñar un sistema de reconocimiento facial donde las limitaciones como la iluminación, posición y distancia no afecten para reconocer a un individuo.

Para hacer más eficiente el proceso de identificación se realiza previamente el análisis de los componentes principales (PCA) de las imágenes de entrenamiento, almacenando los datos que se generan de este proceso (eigenvectores, eigenvalores y la imagen media), para que al realizar la identificación se carguen esos datos sin que el sistema deba calcularlos cada vez que se haga el reconocimiento de la persona, reduciendo así el tiempo de ejecución del programa y el costo computacional.

Para realizar el sistema de reconocimiento facial en Visual C++ 2008 es recomendable utilizar las librerías de OpenCV, porque permite realizar el sistema de una manera más sencilla, rápida y robusta, gracias al gran número de clases y funciones que tiene para realizar aplicaciones de visión artificial.

Realizar un sistema de reconocimiento facial que permita identificar a la persona sin importar los gestos que adopte su rostro. Una posible solución seria agregar a la base de datos del sistema de reconocimiento facial una buena cantidad de imágenes con diferentes gestos de los rostros que pueden adoptar las personas que se desean identificar.

Estos son los posibles trabajos a futuro que se pueden realizar a partir del Sistema de Reconocimiento Facial desarrollado.

Combinar el reconocimiento facial con otras técnicas de reconocimiento biométrico. Esto permite tener un sistema de identificación de personas muy robusto, debido si una de las técnicas de reconocimiento biométrico presenta una fiabilidad baja a causa de factores externos, la identificación podrá ser validada por otra de las técnicas biométricas que se encuentren en el sistema. Por ejemplo, si una persona tiene la huella dactilar desgastada el sistema de reconocimiento por huella no lo podrá reconocer, por lo tanto la identificación de la persona se haría mediante el reconocimiento facial.

En caso de que el sistema de reconocimiento facial en un ambiente controlado tenga una aplicación comercial, seria necesario limitar la operación de adición y

eliminación de usuarios mediante una contraseña alfanumérica y se podría integrar con otros sistemas de reconocimiento biométrico.

El sistema de reconocimiento facial se podría realizar para el registro de acceso a lugares de trabajo donde el sistema registraría la hora de llegada y salida de la persona, validando el ingreso mediante el reconocimiento del rostro.

Se podría integrar el sistema de reconocimiento facial con sistemas de seguridad de cerraduras y electrónicos. La idea es que en un futuro se establezca el sistema de reconocimiento facial a nivel de usuario para que pueda ser utilizado en un supermercado o establecimiento donde se podría llevar a cabo un control sobre la persona que accede a una caja registradora mediante un reconocimiento facial previo, y así poder evitar los robos, ya que si el sistema no reconoce al usuario la caja registradora no podría ser abierta, otra opción podría ser que este sistema se desarrollará para los cajeros automáticos, debido a que esto permitiría realizar transacciones por medio del cajero de una manera más cómoda.

Realizar un sistema de reconocimiento facial como herramienta de seguridad para dispositivos móviles. Donde el dispositivo además de tener un bloqueo desactivado por contraseña alfanumérica, también sea desactivado por la imagen del rostro del propietario. Cabe resaltar que este sistema de reconocimiento facial tendría el gran desafío de realizar una verificación exitosa sin importar el lugar donde se encuentre el dispositivo móvil.

Diseñar un sistema de reconocimiento facial utilizando imágenes 3D tanto en el entrenamiento como en el reconocimiento. Puesto que con el algoritmo 3D se puede reconocer a una persona desde diferentes ángulos.

BIBLIOGRAFÍA

Armengot Iborra, Marcelo J; “Análisis comparativo de métodos basados en subespacios aplicados al reconocimiento de caras”. España. Universidad de Valencia; 2006, p 68.

Ballester Tomas, Felipe. Computer visión and face recognition. Tesis de licenciatura. Alemania. University of Applied Sciences. 2010, p 50.

Como trabaja la detección facial. Disponible en: http://www.cognotics.com/opencv/servo_2007_series/part_2/sidebar.html, visitado: Enero 20, 2012.

Devis Botella, Ricardo. C++/OOP un enfoque practico. Lugar de publicación desconocido. s.f, p 297.

Eigenface Tutorial. Disponible en: http://www.pages.drexel.edu/~sis26/Eigenface%20Tutorial.htm, visitado: Enero 23, 2012

González, Rafael y Woods, Richard. *Digital Image Processing*. Ed. Addison-Wesley. 1992.

HAAR CASCADE. Disponible EN: http://opencv.itseez.com/modules/objdetect/doc/cascade_classification.html, visitado: Enero 23, 2012.

El Manual de referencia de OpenCV versión 2.3. Disponible en: https://code.ros.org/trac/opencv/export/7294/branches/2.3/opencv/doc/opencv2refman.pdf, visitado: Enero 25, 2012.

Long, Sebastián y Müller, Omar. Verificación Biométrica Automática de Identidad Mediante Reconocimiento Facial. Santa Fe. 2006, p 19-20

Mico Andrés, María Luisa. Algoritmos de búsqueda de vecinos más próximos en espacios métricos. Universidad de alicante. 1996, p 166.

Moreno Seco, Francisco. Clasificadores eficaces basados en algoritmos rápidos de búsqueda del vecino más cercano. Tesis doctoral. Universidad de alicante. 2004, p 150.

OpenCV. Disponible en: http://opencv.willowgarage.com/wiki/, visitado: Febrero 10, 2012.

Proceso de un sistema de reconocimiento facial. Disponible en: http://es.wikipedia.org/wiki/Sistema_de_reconocimiento_facial, visitado: Febrero 10, 2012

Rainer Lienhart y Jochen Maydt. An Extended Set of Haar-like Features for Rapid Object Detection. IEEE ICIP 2002, Vol. 1, Sep. 2002, p 900-903.

Reconocimiento facial. Disponible en: http://www.face-rec.org/general-info/, visitado: Febrero 12, 2012.

Reyes López, cesar. Reconocimiento facial mediante visión artificial. Tesis de grado de ingeniería de comunicación. Sevilla 2005, p 134.

SZELISKI, Richard; “Computer Visión: Algorithms and Applications”; Springer 2010.

Turk, Matthew. Pentland, Alex. Eigenfaces for Recognition, Journal of Cognitive Neurosicence, Vol. 3, No. 1, 1991, p 71-86.

Viola, Paul y Jones Michael. Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE CVPR, 2001.

Visión artificial algoritmos. Disponible en: http://visiónartificial-poligran.blogspot.com/p/algoritmos.html, visitado: Febrero 11, 2012.

ANEXOS

ANEXO A. MANUAL DE INSTALACIÓN DE LAS LIBRERÍAS DE OPENCV USANDO CMAKE

En este manual se explica detalladamente como instalar las librerías de **OpenCV 2.3.1** (librería que se utiliza para realizar los proyectos de visión computacional) en el entorno de desarrollo de **Visual Studio 2008 Express Edition** y en un sistema operativo **Windows 7.** Los pasos que se presentan a continuación son el resultado del resumen de la información hallada en diferentes foros de internet.

1. Obtener la librería OpenCV 2.3.1 del link http://sourceforge.net/projects/opencvlibrary/files/opencv-win/2.3.1/ descargando el archivo OpenCV-2.3.1.

2. Obtener el compilador CMake 2.8. del link http://www.cmake.org/cmake/resources/software.html descargando el archivo llamado cmake-2.8.0-win32-x86.exe.

3. Obtener el Visual Studio C++ Express Edition 2008 de la página oficial de Microsoft que se encuentra en el link http://www.microsoft.com/express/downloads/#2008-Visual-CPP descargando vcsetup.exe.

4. Instalación de Visual Express 2008.
Ejecutar el archivo vcsetup.exe para instalar en Visual Express 2008, seguir el manual de instalación.

5. Instalación de CMake
Ejecutar el CMake-2.8.7-win32-x86.exe y seguir el asistente de instalación tomando todas las opciones que se encuentran por omisión.

6. Instalación OpenCV.

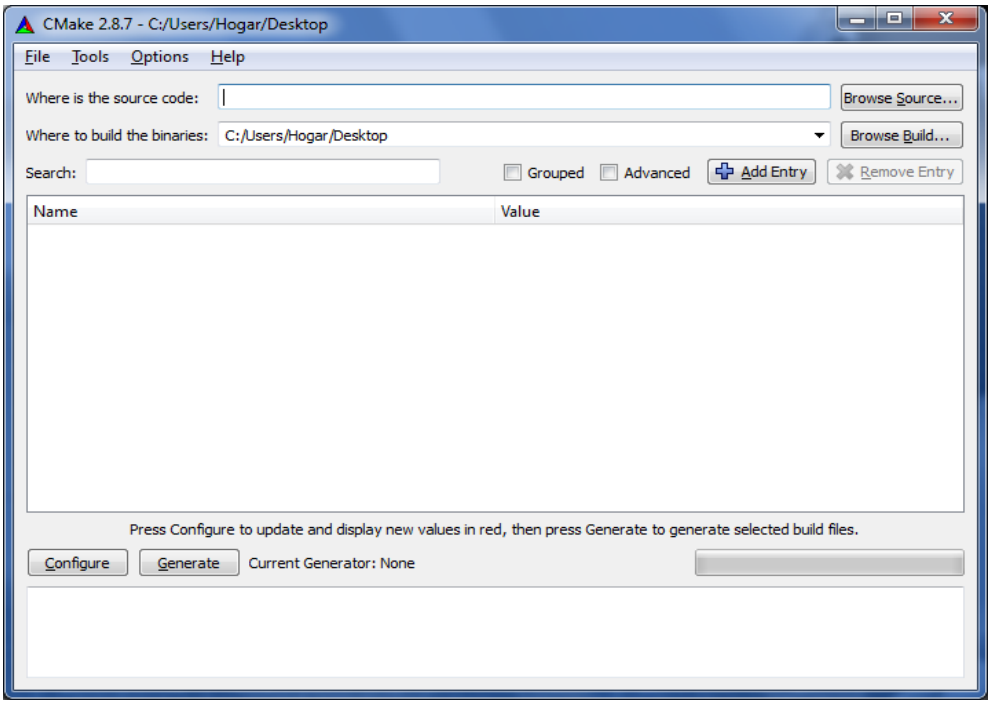
Ejecutar OpenCV-2.3.1-win-superpack.exe para instalar OpenCV asegurándose que el directorio de instalación sea en C:\OpenCV, y se descomprime las librerías en dicha carpeta.

7. Configuración de CMake.

A continuación se muestran los pasos para configurar el CMake (si esto se hace bien el resto del proceso es muy fácil).

a. Ir a Inicio -> Todos los programas -> CMake 2.8.7 -> CMake (CMake-gui) y configurar como se muestra en la Figura 21.

Figura 21. CMake 2.8.7



b. Para la primera opción (source code) se elige el directorio donde se encuentra instalado el OpenCV.

c. Para la otra opción (build binaries) se elije el folder donde van a quedar los archivos compilados, en nuestro caso la carpeta donde quedo fue C:\cmake. Si el folder no existe CMake pregunta si se quiere crear el folder.

d. Después se presiona el botón Configure en la esquina inferior izquierda. Entonces el CMake muestra una serie de opciones como se ve en la siguiente Figura 22.

e. La opción que se elije es la del Visual Studio 9 2008 y se da click en finish. Entonces el CMake muestra un menú de configuración.

f. Este paso es el más importante de todos porque un error acá puede dañar el resto del proceso. Se deben escoger las opciones como se muestra en la Figura 23.

g. Después se presiona Configure y el color rojo debe desaparecer (en nuestro caso se debió presionar dos veces Configure).

h. Se presiona el botón Generate para completar el proceso y se cierra el CMake.

Figura 22. Opciones para guardar las librerías de OpenCV en Visual C++ desde CMake

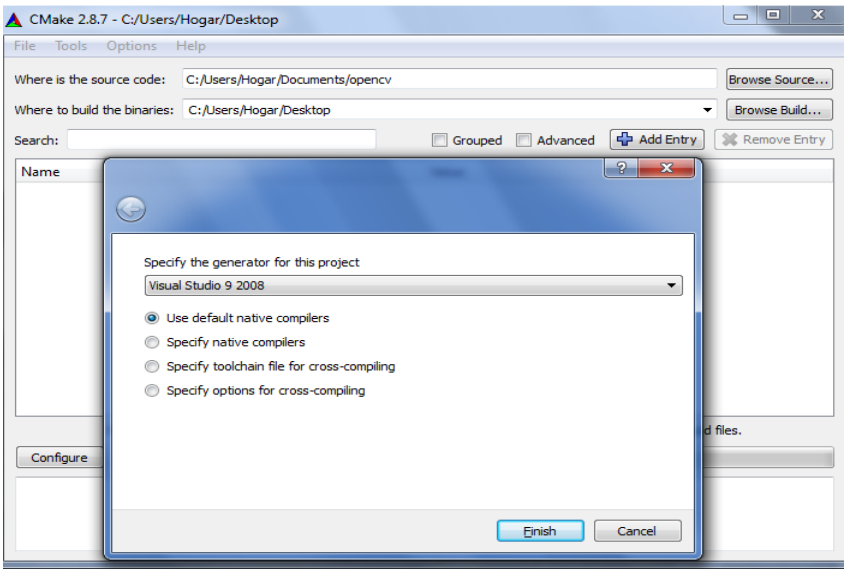
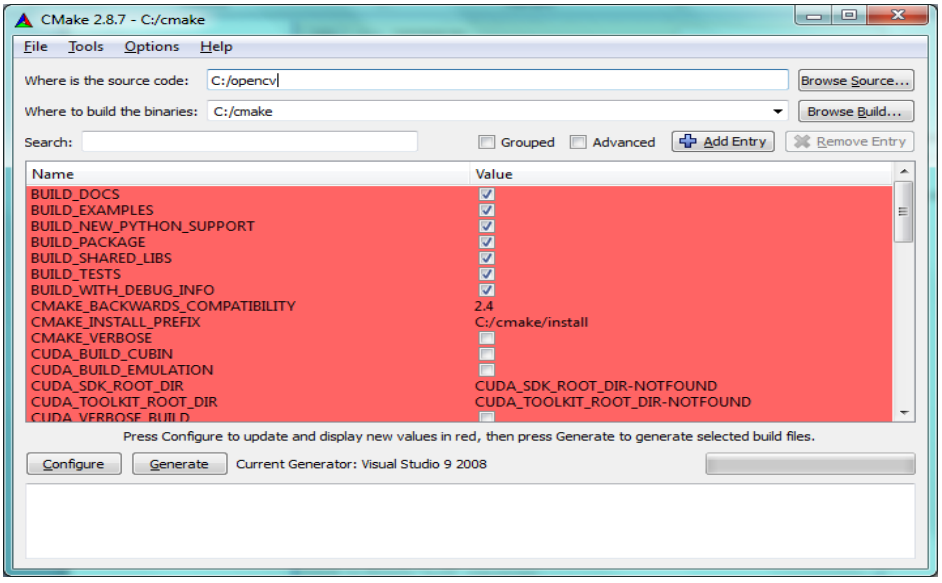


Figura 23. Opciones para crear los archivos necesarios



Las opciones que se eligen son: **BUILD_EXAMPLES, BUILD_PACKAGES, BUILD_TEST, INSTALL_C_EXAMPLES, PENCV_BUILD_3RDPARTY_LIBS, OPENCV_WHOLE_PROGRAM_OPTIMIZATION**.

i. Con esto se crean los siguientes archivos en el directorio de destino que se observan en la Figura 24.

j. Abrir el archivo OpenCV.sln, con Microsoft Visual Studio 2008

k. Luego se construye el proyecto (**Build-> Build**) escogiendo primero la opción Build y luego la opción Debug and Release).

Figura 24. Archivos creados en la carpeta de destino

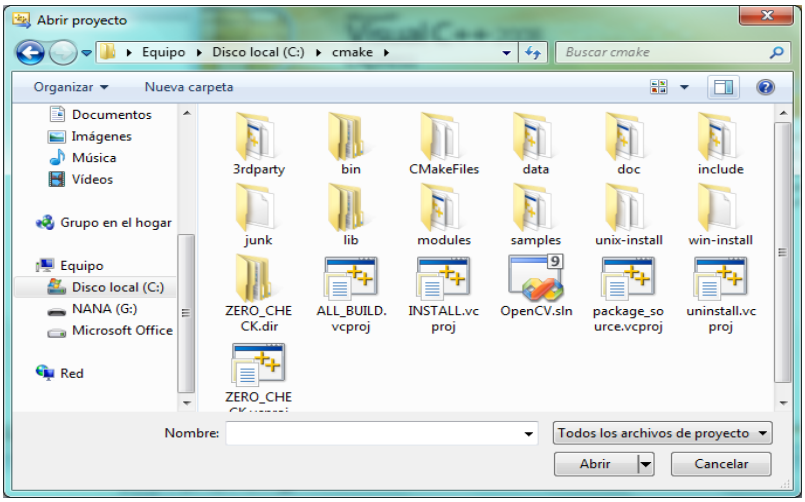
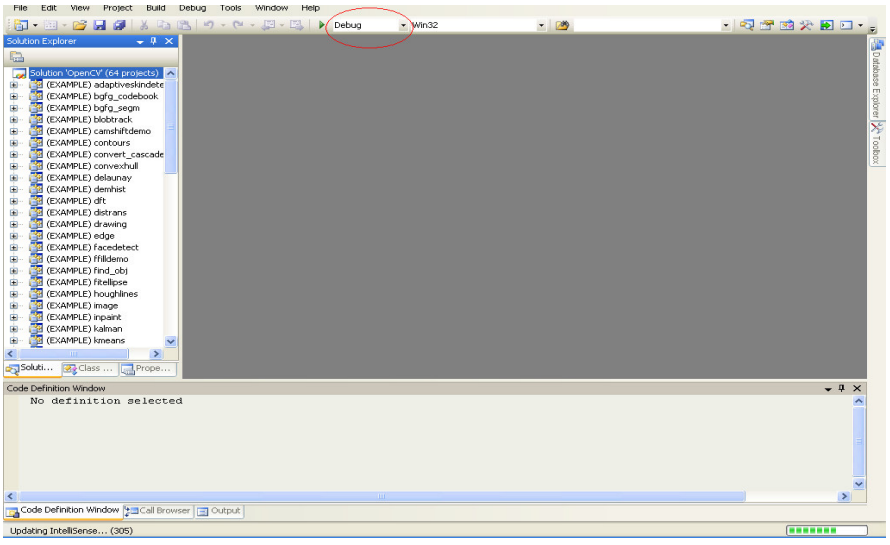


Figura 25. Compilador del archivo opencv.sln



l. Después de hacer cada compilación que se muestra en la Figura 25, se debe asegurar que el mensaje en output message Windows tenga Warning Messages

(se ignoran), 0 errors, Number of build, Number of up-to-date, and number of skipped.

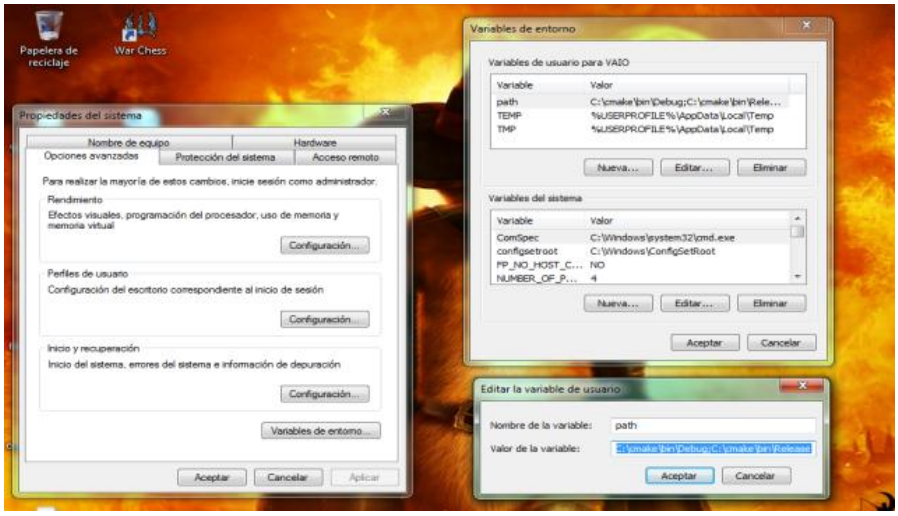
m. Lo importante es que la compilación no tenga ningún error. Si aparece algún error probablemente es porque las opciones del CMake no fueron seleccionadas apropiadamente, en este caso se debe borrar la carpeta vs2008 y se debe repetir toda la parte de CMake.

n. Cuando la compilación sea correcta se cierra el proyecto (no el IDE).

o. Verificar que los archivos opencv_core231d.lib, opencv_highgui231d.lib, opencv_video231d.lib, opencv_ml231d.lib, opencv_legacy231d.lib, opencv_imgproc231d.lib, opencv_objdetect231d.lib, opencv_ts231d.lib, opencv_features2d231d.lib, opencv_flann231d.lib, opencv_gpu231d.lib, opencv_haartraining_engine.lib, opencv_contrib231d.lib, opencv_calib3d231d.lib, se hayan creado en C:\cmake\lib\Debug y C:\cmake\lib\Release.

p. Después se debe agregar estas direcciones C:\cmake\bin\Debug; C:\cmake\bin\Release a las variables del sistema. Para ello se va a Mi PC-> Propiedades->Opciones Avanzadas->Variables de entorno->PATH->Modificar y se agregar sin borrar nada las direcciones especificadas como se muestra en la Figura 26.

Figura 26. Variables del sistema

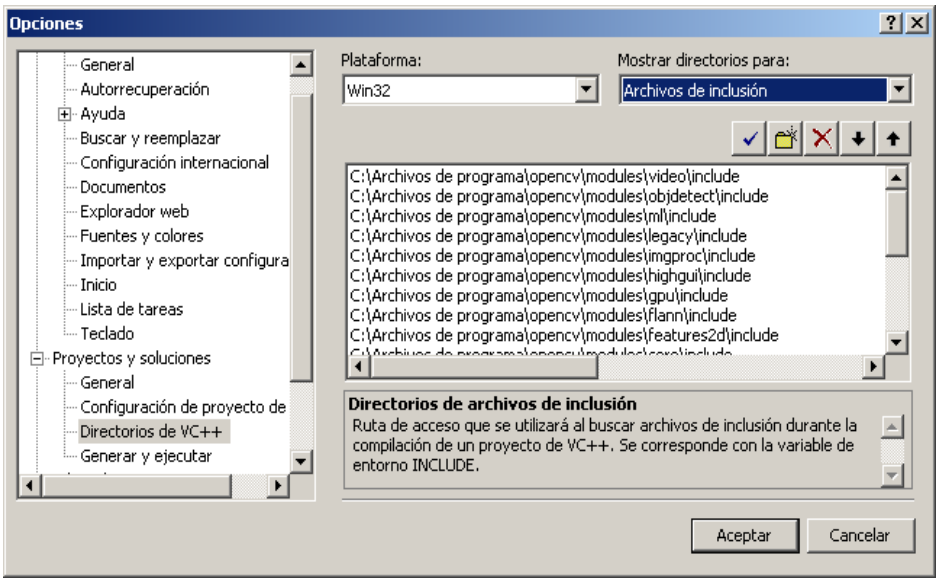


8. Configuración de Visual Studio 2008.

A continuación se presentan los pasos que se deben realizar para incluir las librerías que se han compilado al Visual Studio 2008.

a. En el Visual C++ 2008 Express Edition ir a Herramientas->Opciones->Proyectos y soluciones -> directorios VC++ como se muestra en la Figura 27.

Figura 27. Inclusión de los archivos de inclusión



b. En la opción Mostrar directorios para: seleccionar archivos de inclusión e incluir C:\opencv\include\opencv sin borrar nada de lo que ya se encuentra allí.

C:\Archivos de programa\opencvcvs\include\opencv
C:\Archivos de programa\opencvcvs\include

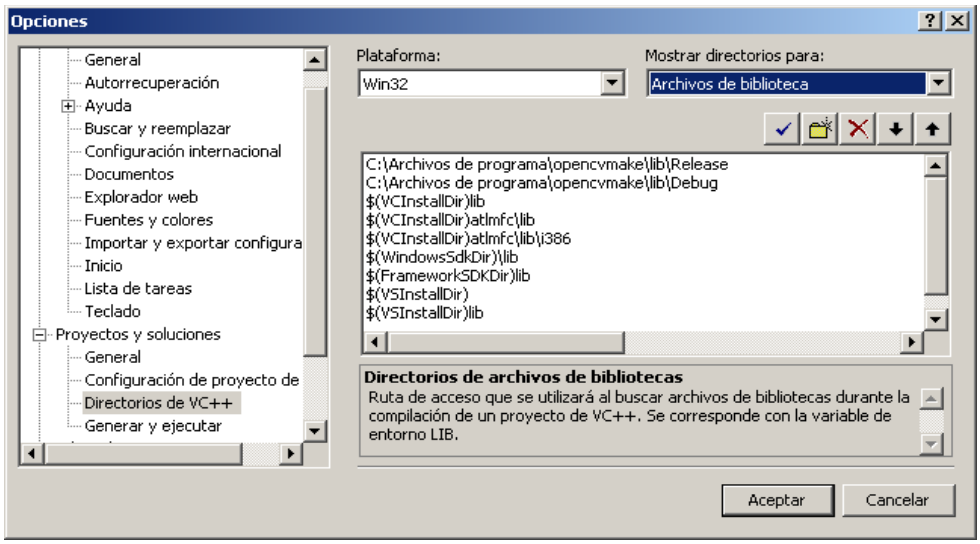
Si genera error respecto a los include añadir también:

C:\Archivos de programa\opencvcvs\modules\video\include
C:\Archivos de programa\opencvcvs\modules\objdetect\include
C:\Archivos de programa\opencvcvs\modules\ml\include
C:\Archivos de programa\opencvcvs\modules\legacy\include
C:\Archivos de programa\opencvcvs\modules\imgproc\include
C:\Archivos de programa\opencvcvs\modules\highgui\include
C:\Archivos de programa\opencvcvs\modules\gpu\include
C:\Archivos de programa\opencvcvs\modules\flann\include
C:\Archivos de programa\opencvcvs\modules\features2d\include
C:\Archivos de programa\opencvcvs\modules\core\include
C:\Archivos de programa\opencvcvs\modules\contrib\include
C:\Archivos de programa\opencvcvs\modules\calib3d\include

c. Después seleccionar la opción de archivos de biblioteca e incluir los archivos

C:\cmake\lib\Release, C:\cmake\lib\Debug como se muestra en la Figura 28:

Figura 28. Inclusión de los archivos de biblioteca



d. Después se repite el proceso escogiendo la opción archivos de código fuente como se observa en la Figura 29.

e. Después seleccionar la opción de archivos ejecutables e incluir los archivos

C:\cmake\bin\Release, C:\cmake\bin\Debug como se muestra en la Figura 30.

f. Después de completar correctamente todos los anteriores pasos, la instalación está lista y ya se puede hacer algún programa con OpenCV.

Figura 29. Inclusión de los archivos de código fuente

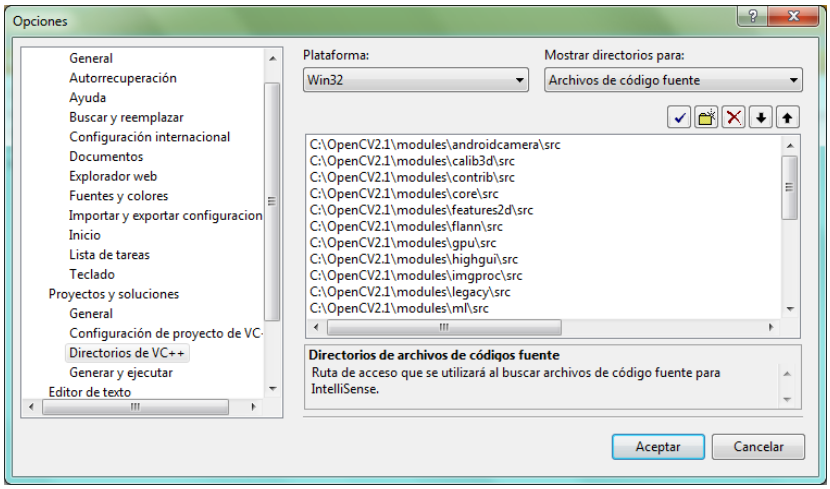
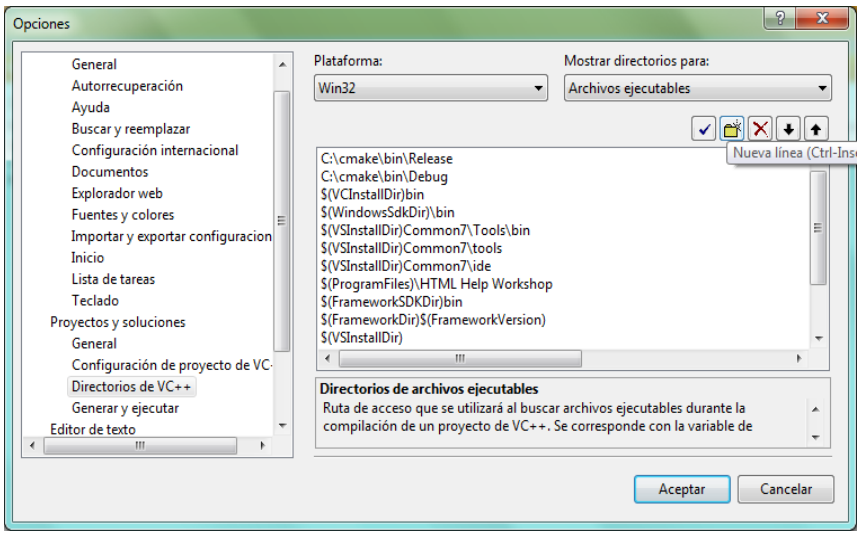


Figura 30. Inclusión de los archivos ejecutables



9. Correr un programa en Visual usando las librerías de OpenCV como se muestra en la Figura 31.

Archivo -> Nuevo -> Proyecto.

Acá se selecciona Aplicación de Consola de Win 32 y se nombra el proyecto, se da clic en aceptar, siguiente y luego en finalizar.

Figura 31. Creación de un proyecto.

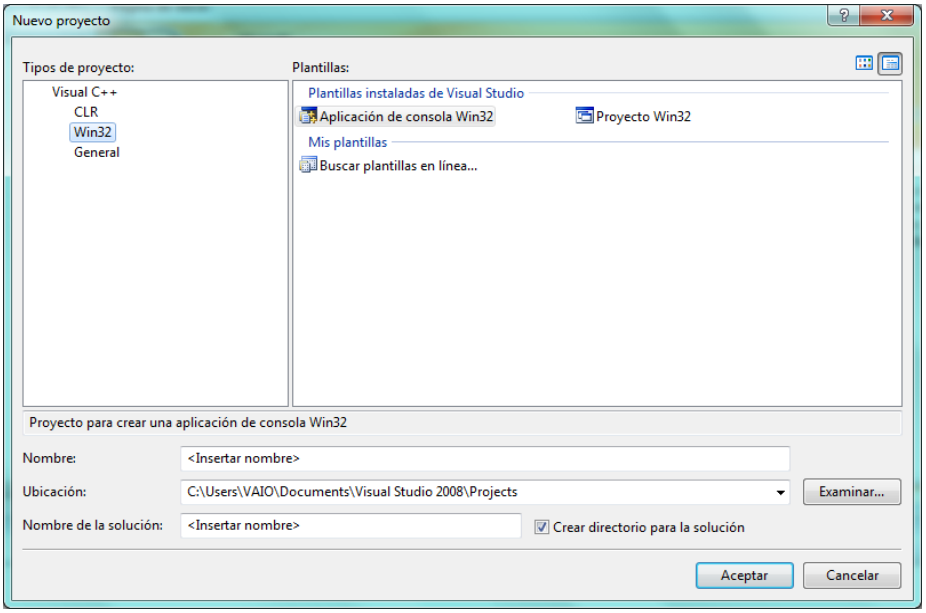
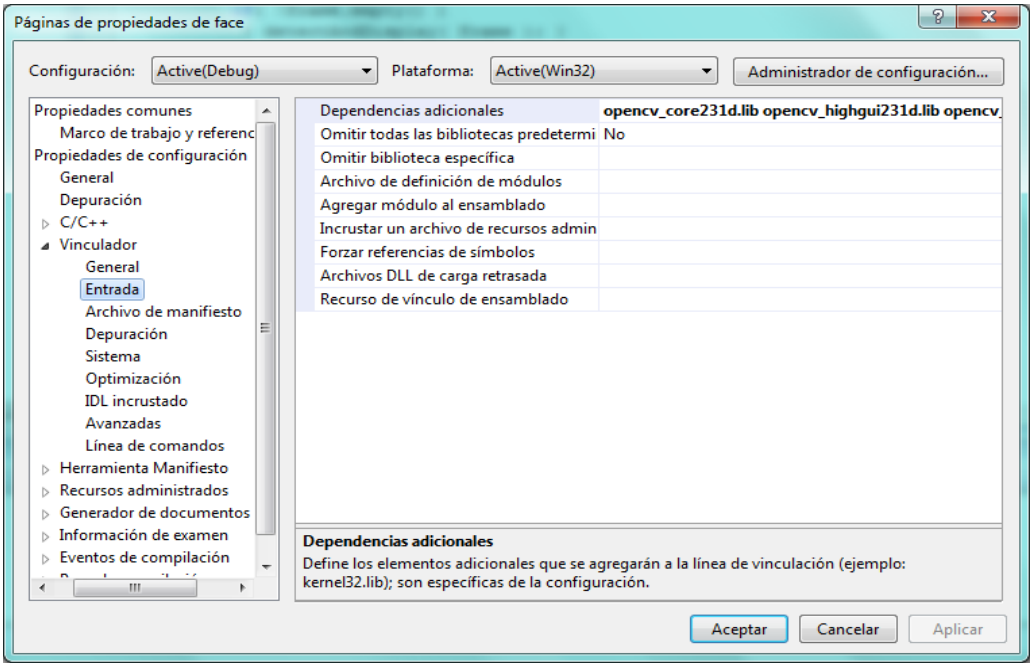
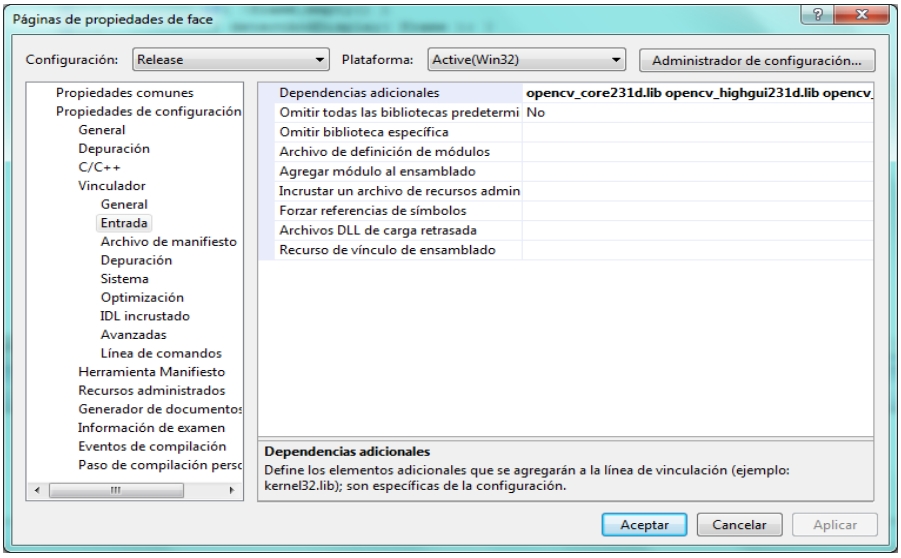


Figura 32. Inclusión de las librerías de OpenCV en el proyectó creado en debug



opencv_core231d.lib opencv_highgui231d.lib opencv_video231d.lib opencv_ml231d.lib opencv_legacy231d.lib opencv_imgproc231d.lib opencv_objdetect231d.lib opencv_ts231d.lib opencv_features2d231d.lib opencv_flann231d.lib opencv_gpu231d.lib opencv_haartraining_engine.lib opencv_contrib231d.lib opencv_calib3d231d.lib

Figura 33. Inclusión de las librerías de opencv en el proyectó creado en Release



Nota: para este caso:
String face_cascade_name = "haarcascade_frontalface_alt.xml";
String eyes_cascade_name = "haarcascade_eye_tree_eyeglasses.xml";
Se deben poner los dos archivos dentro de la carpeta del proyecto en este caso el proyecto se llama facedetecvisión: C:\Documents and Settings\Stiven\Mis documentos\Visual Studio 2008\Projects\facedetecvision\facedetecvision.

ANEXO B. MANUAL DE USUARIO DEL SISTEMA DE RECONOCIMIENTO FACIAL

El sistema de reconocimiento facial desarrollado, identifica a las personas en las imágenes a partir de una base de datos previamente creada. Tiene dos etapas principales como lo son la etapa de la detección del rostro y la etapa del reconocimiento facial, permitiendo también añadir usuarios, borrar usuarios, cargar usuarios, limpiar la lista de usuarios y salir del sistema. Este manual pretende explicar al usuario como utilizar de una manera sencilla el sistema de reconocimiento facial que se ha diseñado.

Primeramente se deben cumplir ciertos requisitos tanto a nivel de hardware como a nivel de software, a nivel de hardware se requiere de una cámara web correctamente instalada con sus respectivos drivers, se requiere que el computador que se va a utilizar a la hora de implementar el sistema de reconocimiento facial tenga características similares o superiores al computador donde se desarrollo dicho sistema como se muestra a continuación.

Para obtener las imágenes se utilizo una cámara web, la cual esta integrada al portátil sony vaio VPCEG-33FL con Resolución 640 x 480, .3 mega pixeles, procesador Intel ® Pentium ® 4 2,40 GHz, caché de 512 K, 533 MHz FSB, Memoria: 1GB[3], son los requerimientos mínimos para el funcionamiento del sistema de reconocimiento facial.

Cabe destacar que la iluminación del ambiente donde se va a utilizar el sistema, afecta la eficiencia del reconocimiento. Por lo tanto, esta aplicación funciona de forma óptima en un lugar cerrado donde se tenga preferiblemente iluminación artificial esto con el fin de reducir los efectos de sombra y brillos no deseados, mejorando de esta manera el contraste de la imagen. También se debe tener en cuenta que la fuente de iluminación artificial no debe estar al frente de la cara, debido a que se pueden presentar problemas con los brillos en la imagen del rostro.

Para que el sistema de reconocimiento facial tenga la mejor eficiencia de debe actualizar la base de datos de los usuarios, en el sitio donde se instalará o ejecutara el programa de reconocimiento facial, esto hace que el reconocimiento sea muy preciso, puesto que el sistema se evalúa sobre el mismo ambiente en el que se toman las imágenes de la base de datos.

En cuanto al software el sistema operativo debe pertenecer a la familia de Microsoft Windows. Se necesita instalar el Microsoft Visual C++ 2008 Express Edition y las librerías OpenCV, que se pueden obtener de forma gratuita.

Una vez instalado el Visual C++ 2008 Express Edition y las librerías de OpenCV, se procede a la ejecución del sistema de reconocimiento facial en un ambiente controlado desde el ejecutable nana que se encuentra en C:\Users\VAIO\Documents\Visual Studio 2008\Projects\nana\Debug.

El estado inicial de la interfaz del sistema muestra la función principal del programa, como se puede observar en la Figura 34, este contiene los botones de nuevo usuario, borrar usuario, limpiar lista, detectar, terminar detección, reconocer y cerrar, también se ven las cajas de texto donde se mostrará el resultado del reconocimiento, la distancia de sumas ponderadas entre la imagen de entrada y la imagen de la base de datos más cercana, donde se imprimen los mensajes de error o advertencia al usuario y la caja de texto donde se ingresa el nombre identificador de un nuevo usuario, además se observa una ventana de imagen donde se visualizará la imagen de la cara detectada y una caja de lista donde se pueden cargar los usuarios de la base de datos.

Figura 34. Interfaz gráfica del sistema de reconocimiento facial

A continuación se explica cada uno de los componentes de la interfaz gráfica.

Nombre de usuario: caja de texto que visualiza el resultado del proceso de reconocimiento.

Distancia de reconocimiento: caja de texto que visualiza la distancia de sumas ponderadas de la imagen que se quiere reconocer, con la imagen de la base de datos que más se asemeja a esta.

Mensajes al usuario: caja de texto que muestra mensajes que guían al usuario, cuando no se utiliza el programa de reconocimiento facial de forma adecuada.

Lista de usuarios: caja de lista que muestra los nombres de los rostros almacenados en la base de datos.

Detectar: activa la cámara y ejecuta el proceso de detección de rostro.

Detener detección: detiene el proceso de detección dejando la imagen detectada en la interfaz gráfica.

Reconocer: ejecuta el proceso de reconocimiento y muestra los resultados en las cajas de texto de “Nombre de usuario” y “distancia de reconocimiento”.

Añadir usuario: adiciona una imagen nueva y el nombre de un usuario a la base de datos del sistema de reconocimiento facial.

Cargar usuarios: muestra los nombres de las imágenes de los usuarios que se encuentran almacenados en la caja de lista.

Borrar usuario: elimina al usuario seleccionado en la caja de lista de la base de datos.

Limpiar lista de usuarios: pone en blanco la caja de lista.

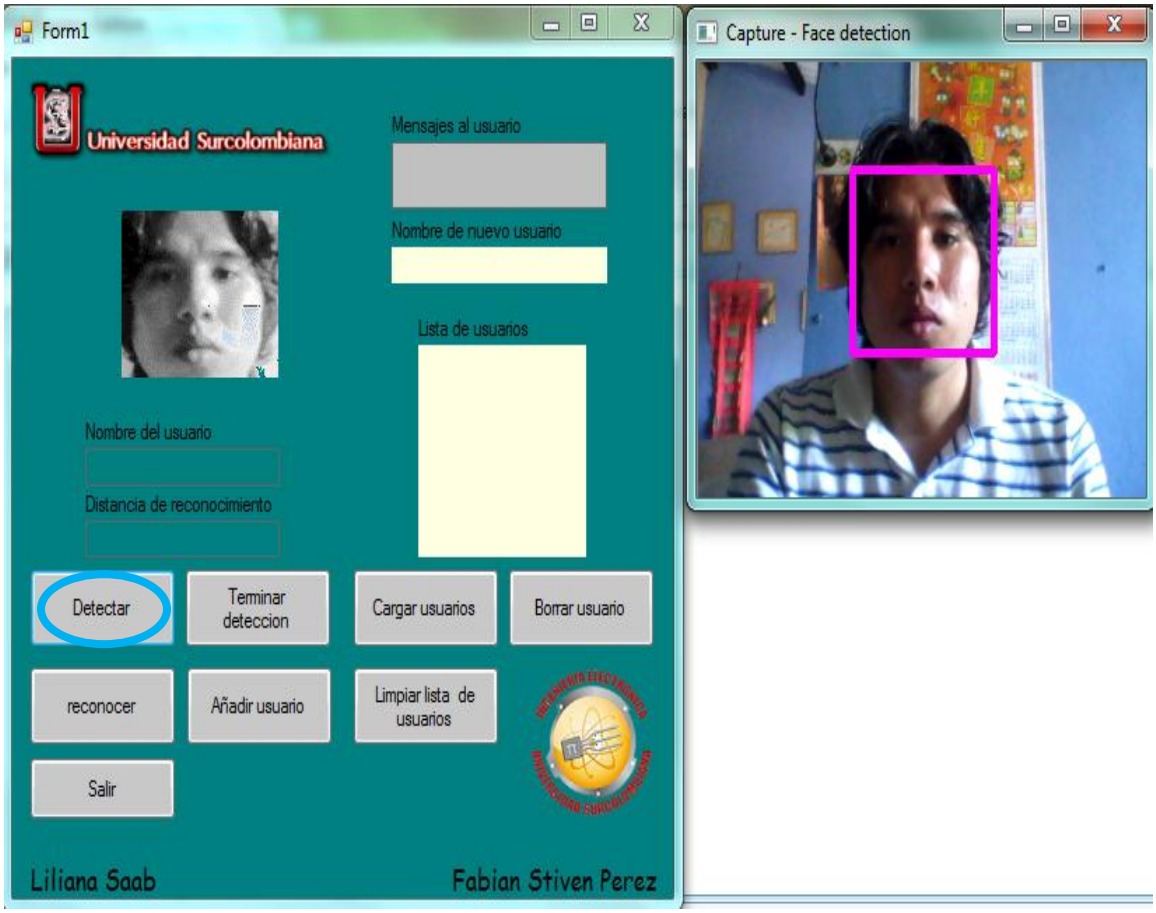
Salir: cierra el programa de reconocimiento facial.

Una vez se tiene claro la función que cumple cada componente de la interfaz gráfica, se exponen los pasos generales que se deben realizar para la ejecución correcta del sistema de reconocimiento facial.

Proceso de detección facial: La primera opción en el menú principal es la detección. Esta parte de la aplicación es capaz inicializar la captura de imagen desde la cámara y encontrar las caras en dicha imagen, estas se detectan automáticamente y muestran un recuadro alrededor de ellas, además muestra la

cara detectada en una imagen aparte, esto con el fin de que si se detectan más de una cara el usuario verifique que se está mostrando y almacenando la cara que se desea reconocer, a su lado se encuentra el botón detener detección, que finaliza la detección pero deja en la ventana de imagen la imagen detectada. En la Figura 35, se observa el sistema ejecutando el proceso de detección facial.

Figura 35. Detección facial



Proceso de adición. En primer lugar se añaden los usuarios a la base de datos (el programa de reconocimiento facial viene por defecto con dos usuarios cada uno con una imagen de su respectivo rostro, esto debido a que el sistema de reconocimiento facial no permite tener menos de dos imágenes de rostro en la base de datos), esta adición de usuarios se debe hacer preferiblemente en el lugar donde se realizará posteriormente el reconocimiento facial o en un entorno similar y deben estar a 1meto de la cámara, esto con el fin de que la precisión del sistema sea cercana al 100%. Para añadir un nuevo usuario primeramente se debe detectar el rostro que se quiere adicionar al sistema, para detectar un rostro se da clic en el botón “Detectar” e inmediatamente se activa la cámara y detecta los

rostros en la imagen, luego se escribe el nombre del nuevo usuario en la caja de texto llamada “Nombre de nuevo usuario” y finalmente se debe dar clic en el botón “Añadir usuario”, una vez realizado este proceso, se ha adiciona la cara detectada que se encuentra visualizada en la interfaz gráfica en la base de datos, con su respectivo nombre. En la Figura 36, se puede observar un ejemplo de adición de un nuevo usuario.

Figura 36. Adición de un nuevo usuario al sistema de reconocimiento facial

Proceso de eliminación de usuarios: Los usuarios añadidos se pueden borrar con el botón “Borrar usuario”, para realizar este proceso primero se deben visualizar los usuarios en la caja de lista llamada “Lista de usuarios”, utilizando el botón “cargar usuario”, una vez cargado los usuarios se selecciona el usuario que se desea borrar (cada vez que se seleccione un nombre usuario se mostrará su imagen en una caja de imagen) y se da clic en el botón “borrar usuario”, inmediatamente se borra el usuario de la base de datos y se actualiza la lista de usuarios.

Figura 37. Eliminar usuario al sistema de reconocimiento facial

Nota: si solo hay dos imágenes de rostro en la base de datos, el sistema de reconocimiento facial no permitirá borrar ninguno, debido a que el proceso de reconocimiento debe tener mínimo dos imágenes para su correcta ejecución, así, que si desea borrar alguno de las únicas dos imágenes que hallan, primero debe añadir una tercera imagen de rostro, para poder ejecutar el proceso de eliminación de usuario.

Proceso de cargar usuario y limpiar lista de usuarios: El botón de cargar usuarios muestra los usuarios que se encuentran actualmente en la base de datos, este proceso es requerido previamente para ejecutar la función del botón borrar usuario, este actúa sin mostrar errores siempre y cuando se hallan cargado los usuarios y uno de los usuarios este seleccionado, la función de este botón es eliminar de la base de datos el usuario seleccionado, e imprime en la caja de lista la nueva lista de usuarios que se encuentran en la base de datos. Debajo del

botón cargar usuarios se encuentra el botón limpiar lista de usuarios, el cual tiene la sencilla función de dejar en blanco la caja de texto.

Figura 38. Cargar usuario

Proceso de reconocimiento facial en un ambiente controlado: este proceso se inicia con la detección del rostro que se desea identificar, para esto se da clic en el botón “Detectar”, el usuario debe ubicarse a 1 metro de la cámara, una vez detectada la cara, se comprueba que el rostro que se visualiza en la interfaz gráfica, sea el rostro que se desea identificar, luego de verificado esto, se da clic en el botón “Reconocer” como se muestra en la Figura 38, ejecutando así, el proceso de reconocimiento a la imagen que se encuentre visualizada en la interfaz gráfica, arrojando los resultados del reconocimiento en las cajas de texto de “Nombre de usuario” y “Distancia de reconocimiento”, si el usuario es reconocido mostrará el nombre del usuario y la distancia de reconocimiento, este valor es un valor técnico del sistema de reconocimiento facial, por lo que no debe ser de gran

importancia para el usuario, en caso de que el usuario no sea reconocido, el nombre de usuario aparecerá como desconocido.

Nota: si el usuario no es reconocido y se desea que si lo sea, se debe añadir una nueva imagen a la base de datos de dicho usuario, para así, aumentar la precisión del sistema y obtener los resultados deseados.

Figura 39. Ejemplo de reconocimiento facial del sistema de reconocimiento facial en un ambiente controlado