

DESARROLLO DE UNA APLICACIÓN INTELIGENTE BASADA EN RECONOCIMIENTO FACIAL PARA CONTROLAR EL ACCESO A LA BIBLIOTECA DANIEL RODRÍGUEZ HOYLE

Development of a smart application based on face recognition to control the access to the Daniel Rodriguez Hoyle Library.

Francisco Lezcano¹ Aixa Quispe¹ David Ruiz¹ Edgard Vílchez¹

¹ Universidad Privada del Norte

Recibido jul. 2013; aceptado nov. 2013; versión final dic. 2013.

Resumen

En este artículo se presenta el desarrollo de una aplicación que permite controlar el acceso de los estudiantes de la Universidad Privada del Norte (UPN) a la biblioteca DRH, mediante el reconocimiento facial. El sistema se implementa en *C Sharp* con la plataforma de trabajo *Emgu CV*, y realiza las tareas de detección de rostros mediante el algoritmo "*Viola - Jones*" que posee dos etapas principales: una de entrenamiento del detector y otra de detección donde se emplea el detector entrenado en la primera etapa sobre cada imagen a analizar, permitiendo la identificación de los estudiantes en tiempo real.

Palabras clave: Reconocimiento Facial, Emgu CV, Viola Jones, detección.

Abstract

This article presents the development of an application to control the access of students from the Universidad Privada del Norte (UPN) to the DRH library using facial recognition. The application is implemented in C Sharp with CV Emgu framework which performs the face detection task by using the "Viola - Jones" algorithm. It has two main stages: training of the detector and the detection itself which employs the detector trained in the first step on each image to be analyzed, allowing the identification of students in real time.

Keywords: Facial Recognition, Emgu CV, Viola Jones, detection.

INTRODUCCIÓN

La detección de rostros es actualmente utilizada en diversos tipos de aplicaciones de seguridad, seguimientos de personas, entre otras. La aplicación se basa en primera instancia en detectar los rostros en una imagen para luego poder determinar la identidad de cada uno de ellos; pudiendo identificar la ubicación, tamaño y orientación de los rostros presentes en una imagen adquirida.

Existen diversos métodos para la detección de rostros, pero no todos logran hacerlo en tiempo real, por ello se eligió el algoritmo “*Viola – Jones*” que ofrece la detección en tiempo real, analizando la imagen en busca de características relevantes que aporten información acerca de la presencia del rostro (Open CV, 2013).

Como complemento en el Análisis de Componentes Principales (PCA) se construye una transformación lineal que escoge un nuevo sistema de coordenadas para el conjunto original de datos (imagen), en el que la varianza de mayor tamaño del conjunto de datos es capturada en el primer eje (llamado el Primer Componente Principal), la segunda varianza más grande es el segundo eje, y así sucesivamente. La transformación que lleva de las antiguas coordenadas a las coordenadas de la nueva base, es precisamente la transformación lineal necesaria para reducir la dimensionalidad de datos. Además las coordenadas en la nueva base dan la composición en factores subyacentes de los datos iniciales (Diez Palacio, 2010).

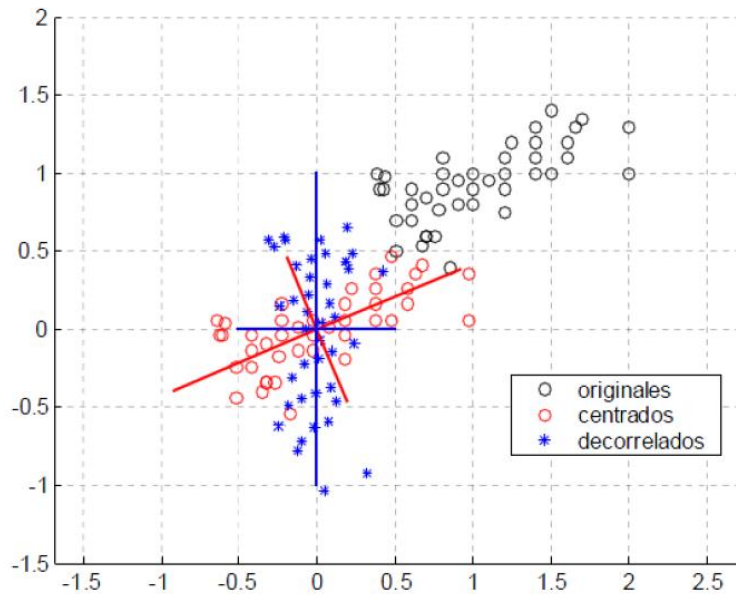


Figura 1. Transformación PCA

Una de las ventajas del PCA para reducir la dimensionalidad de un grupo de datos, es la retención de aquellas características del conjunto de datos que contribuyen más a su varianza, manteniendo un orden de bajo nivel de los componentes principales e ignorando los de alto nivel. El objetivo es que esos componentes de bajo orden (a veces con el aspecto más importante de esa información), permitan realizar el reconocimiento.

En este artículo se desarrolla un sistema con la capacidad de determinar la identidad de una persona mediante el reconocimiento del rostro a través de una cámara, a ser instalada en la Biblioteca Daniel Rodríguez Hoyle (DRH), con la finalidad de verificar si es un estudiante y permitirle el acceso a dicho espacio, sin el tradicional carné de biblioteca.

MATERIALES Y MÉTODOS

El sistema necesita de una pc de escritorio o laptop, que cuente con las características mencionadas en la Tabla 1.

Tabla 1. Especificaciones de equipo informático

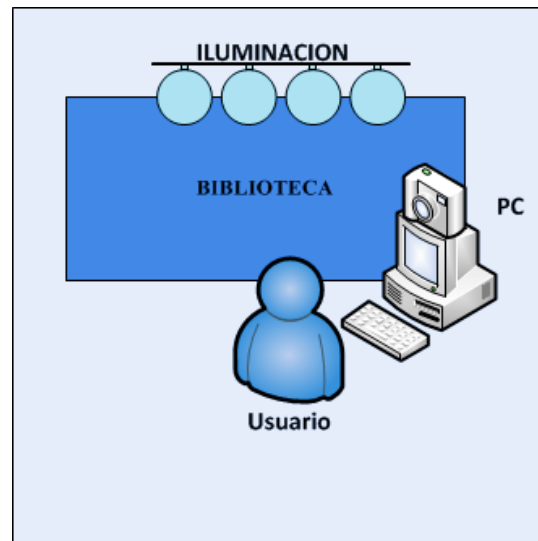
Especificaciones	Memoria RAM 4GB 2.4GHz.
	Procesador Intel Core Duo.
	Resolución de pantalla 1600 x 900 píxeles.
	Cámara VGA (640 x 480).

Fuente: Elaboración propia.

El proyecto se estructura en etapas:

Entorno

El entorno de trabajo debe contar con una pc de escritorio o laptop con las especificaciones mencionadas anteriormente, además de una cámara VGA externa o integrada, ubicada en un entorno iluminado (luz blanca) para mejorar la detección del rostro; y el usuario deberá permanecer en posición frontal y de preferencia en fondo blanco o equilibrado.



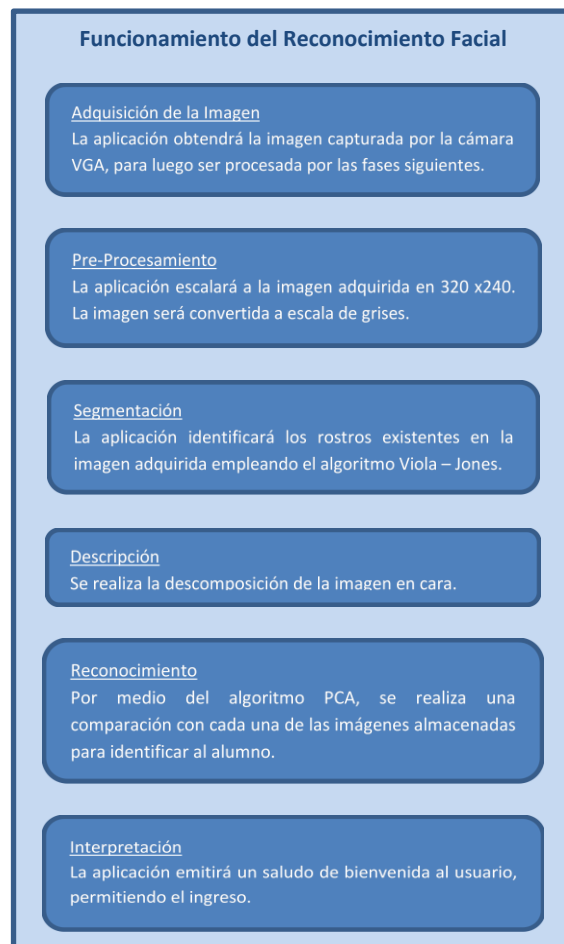
Fuente: Elaboración propia.

Figura 2. Entorno

Control de acceso

La aplicación de control de acceso (ACA), trabaja con luz blanca para permitirle a la cámara realizar la captura precisa del rostro como información para la etapa de entrenamiento. El proceso de toma de imágenes de los alumnos con acceso a la Biblioteca DRH será supervisado por el encargado del área.

Esta aplicación es una propuesta de implementación que se basa principalmente en la adquisición de la imagen, pre-procesamiento, segmentación, descripción, reconocimiento e interpretación, como se muestra en la Figura 3. La aplicación al reconocer el rostro del usuario y validar su existencia en la base de datos, emitirá un saludo de bienvenida al estudiante.



Fuente: Elaboración propia.

Figura 3. Procesamiento de imagen

Captura de Datos

La imagen será capturada a colores, es decir con una profundidad de color de 24 bits por píxel, que representa 8 bits por cada canal (RGB – rojo, verde y azul); el tamaño dependerá de la resolución de la cámara utilizada por la aplicación (Figura 4). Una vez capturada la imagen, se redimensiona usando interpolación bicúbica, con la nueva dimensión que es de 320 x 240 píxeles.

La interpolación bicúbica usa un algoritmo más complejo y otorga buenas aproximaciones, mientras que la interpolación ninguna y lineal otorgan menor calidad de solución.



Figura 4. Captura de datos

Escala de Grises

Una vez redimensionada la imagen, se pasa a escala de grises, para que su análisis en las siguientes fases se realice de una forma más rápida, ya que se trabajará solo en un canal.



Figura 5. Escala de grises

Seguidamente, en la segmentación se procede a analizar la imagen capturada con el fin de identificar todos los rostros que estén dentro de la imagen. Para esto se usa el algoritmo Viola-Jones, el cual a partir de descriptores llamados Haar, identifica si un sector es un rostro o no.

Algoritmo *Viola-Jones*

El algoritmo *Viola-Jones* es un método eficaz de detección y en tiempo real de objetos en una imagen, desarrollado por los investigadores Paul Viola y Michael Jones en el año 2001. Ellos proponen utilizar las características tipo *Haar*, que se calculan como la diferencia de la suma de los píxeles de dos o más zonas rectangulares adyacentes.

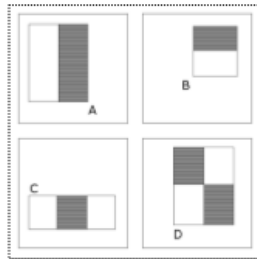
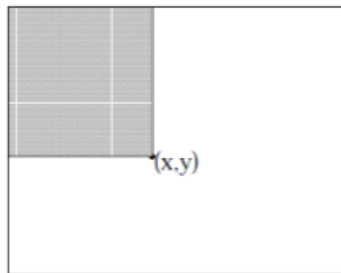


Figura 6. Características tipo *Haar*

Viola y Jones utilizan un algoritmo que busca entre una gran cantidad y variedad de características *Haar* de la imagen, aquellos que mayor información aportan acerca de una cara. Para el cálculo de estas características ellos proponen un nuevo método, la imagen integral, que es una representación en forma de imagen del mismo tamaño que la imagen original, donde cada uno de los puntos se calcula como la suma de los píxeles situados por encima de él y a su izquierda.



Nota. El valor de la imagen integral en un punto (\square) es la suma de los píxeles de la región superior izquierda.

Figura 7. Valor de la imagen

Mediante el uso de la imagen integral puede calcularse cualquier suma rectangular únicamente con cuatro referencias. El valor de la imagen integral en el punto 1 es la suma de los píxeles del rectángulo; en el punto 2 es $A + B$; en el 3 es $A + C$; y en el punto 4 es $A + B + C + D$.

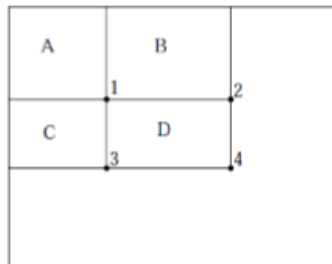
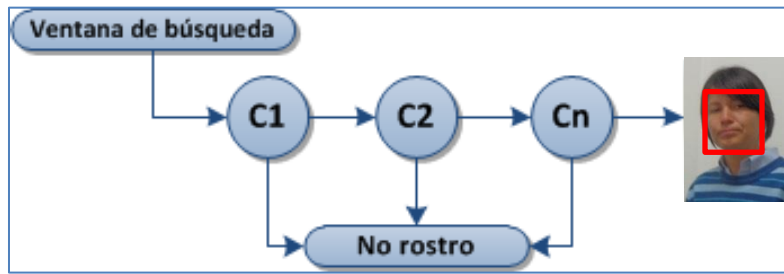


Figura 8. Suma del área rectangular

Algoritmo *Boosting*

El algoritmo de *Viola-Jones* utiliza también el algoritmo *Boosting*, que usa un método de clasificación que combina varios clasificadores básicos para formar un único clasificador más complejo y preciso. La idea se basa en la afirmación de que varios clasificadores

sencillos, pueden combinarse para formar un clasificador de mayor precisión, siempre y cuando se disponga de un número suficiente de muestras de entrenamiento. La aplicación de clasificadores en cascada permite obtener buenos resultados (Álvarez et al, 2010).



Fuente: Elaboración propia en base al algoritmo.

Figura 9. Procedimiento del algoritmo de *Boosting*

Después de la segmentación prosigue la descripción, que es donde se descomponen las imágenes de caras en un conjunto de imágenes características llamadas *eigenfaces*, los cuales son nuestros descriptores captados tanto de la imagen capturada como de la imagen almacenada. También representan las características principales de cada individuo y son los vectores propios del conjunto de imágenes.

Cálculo de las *Eigenfaces*

Sea una imagen $I(x, y)$ y un *array* bidimensional de $N \times N$ valores de intensidad (8 bits). Una imagen puede ser considerada como un vector de dimensión N^2 , de modo que la típica imagen de 100×100 se convierte en un vector de dimensión 10000. Por tanto, un conjunto de imágenes será un conjunto de puntos en un espacio de dimensión enorme.

La idea principal del análisis consiste en encontrar los vectores que formarán el *face space*. Cada vector tiene un tamaño de N^2 , y describe una imagen de $N \times N$. Estos vectores denominados *eigenfaces* son una combinación lineal de las imágenes de las caras originales.

Sea el conjunto de imágenes de entrenamiento $\Gamma_1, \Gamma_2, \dots, \Gamma_M$. La imagen media del conjunto viene definida por la siguiente expresión:

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n$$

Cada imagen difiere del resto por el vector: $\Phi_i = \Gamma_i - \Psi$

Este gran conjunto de vectores es el objeto de estudio del análisis, el cual busca un conjunto de M vectores ortonormales y que mejor definen la distribución de los datos.

El k -ésimo vector u_k es elegido de modo que:

$$\lambda_k = \frac{1}{M} \sum_{n=1}^M (\mathbf{u}_k^T \Phi_n)^2$$

Es un máximo, sujeto a:

$$\mathbf{u}_l^T \mathbf{u}_k = \delta_{lk} = \begin{cases} 1, & \text{si } l = k \\ 0, & \text{en otro caso} \end{cases}$$

Los vectores u_k y los escalares λ_k son los vectores propios y los valores propios, respectivamente, de la matriz de covarianzas:

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T$$

Donde la matriz $A = [\Phi_1, \Phi_2, \dots, \Phi_M]$. La matriz C de covarianzas es de dimensión $N^2 \times N^2$, por lo que determinar los N^2 vectores propios y valores propios resulta una tarea muy compleja a la hora de tratar con tamaños de imágenes normales.

Sin embargo, si el número de vectores es menor que la dimensión de éstos ($M < N^2$), habrá $M - 1$, en lugar de N^2 , vectores significativos (los vectores propios restantes tendrán asociados valores propios igual a cero). De esta forma, se permite reducir el tamaño de la matriz a una dimensión $M \times M$ y luego tomar combinaciones lineales de las imágenes de las caras Φ_i . Considerando los vectores propios, v_i de AA^T : $A^T A v_i = \mu_i v_i$

Multiplicando ambos lados de la igualdad por A, se tiene: $AA^T A v_i = \mu_i A v_i$

Donde se observa que son los vectores propios de $C = AA^T$.

Siguiendo este análisis, se construye la matriz $L M \times M = AA^T$, donde $L_{mn} = \Phi_m^T \Phi_n$ y se buscan los vectores propios, v_l de L. Estos vectores determinan combinaciones lineales de las M imágenes del conjunto de entrenamiento que forman las *eigenfaces* u_l .

$$u_l = \sum_{k=1}^M v_{lk} \Phi_k ; \quad l = 1, \dots, M$$

De esta forma se reduce el tamaño de las imágenes de N^2 píxeles al tamaño de M, correspondiente al número de imágenes del set de entrenamiento.

Reconocimiento

Se proyecta una nueva imagen sobre el espacio formado por las *eigenfaces*, denominado *face space*, y clasifica comparando su posición en ese espacio con las posiciones de los individuos ya conocidos.

En términos matemáticos, el objetivo consiste en encontrar los vectores propios de la matriz de covarianzas del conjunto de imágenes, tratando a la imagen como un punto (o vector) en un espacio de dimensión enorme, donde cada coordenada corresponde al valor de un píxel de la imagen original.

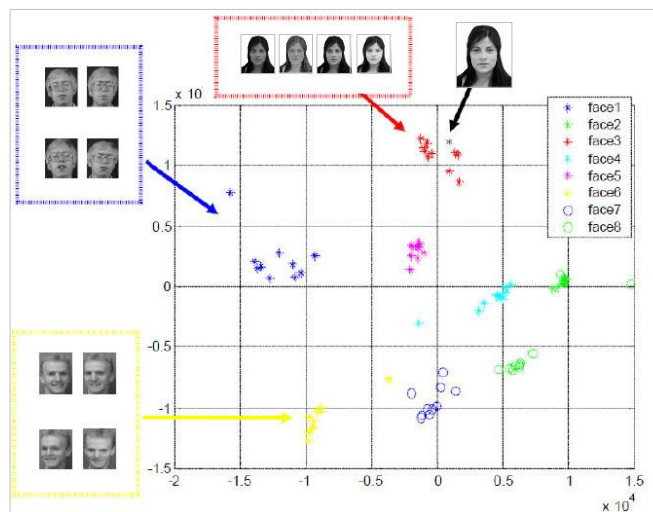
La imagen de cada individuo puede ser representada mediante una combinación lineal de las *eigenfaces*. Cada cara puede aproximarse utilizando únicamente las mejores *eigenfaces*, es decir, aquellas que tienen los mayores valores propios y por tanto, los de máxima varianza. Estas M mejores *eigenfaces* crean un subespacio M-dimensional de todas las imágenes posibles (Figura 10). Esta aproximación al reconocimiento de caras implica las siguientes operaciones iniciales:

Fase de entrenamiento

1. Adquisición de la colección de imágenes iniciales (conjunto de entrenamiento).
2. Calcular las *eigenfaces* del set de entrenamiento, teniendo en cuenta las M imágenes que corresponden a los mayores valores propios. Estas M imágenes definen el *face space*.
3. Calcular la correspondiente distribución en el espacio M-dimensional para cada individuo, proyectando las imágenes en el *face space*.

Fase de test

1. Proyectar la nueva imagen de entrada sobre el *face space* y determinar si ésta es una cara, comprobando si es próxima a este espacio.
2. Si es una cara, utilizar un clasificador para determinar la clase más próxima a la que pertenece la nueva imagen.



Nota. Se muestran las diferentes clases (correspondientes a las caras de los individuos) y la imagen de entrada a identificar.

Figura 10. Ejemplo de un *eigenespacio*

Clasificación de caras mediante *eigenfaces*

Llegados a este punto, una vez realizado el proceso anterior de cálculo de las *eigenfaces*, una nueva imagen de entrada Γ se transforma en su *eigenface* (proyectada en el *face space*) mediante una simple operación:

$$\omega_k = \mathbf{u}_k^T (\Gamma - \Psi); \quad k = 1, \dots, M'$$

Siendo M' el número de vectores propios más significativos de la matriz L . Esta operación es denominada reconstrucción. Estos componentes forman el vector:

$$\Omega^T = \{\omega_1, \omega_2, \dots, \omega_{M'}\}$$

Que describe la contribución de cada *eigenface* a la representación de la imagen de entrada. El método más simple utilizado para determinar qué clase de las existentes proporciona una mejor descripción de la imagen de entrada consiste en buscar la clase k que minimiza la distancia Euclídea:

$$\epsilon_k^2 = \|\Omega - \Omega_k\|^2$$

Donde Ω_k es el vector que describe la clase k -ésima. Las clases Ω_i se calculan promediando los resultados de la representación de la *eigenface* sobre un pequeño número de imágenes de cada individuo. Una cara se clasifica como perteneciente a una clase k cuando el mínimo ϵ_k está bajo un umbral $\theta\epsilon$, escogido previamente. En otro caso se clasifica como desconocida y puede utilizarse para crear una nueva clase.

La distancia ϵ al cuadrado entre una imagen de entrada y el *face space* es la diferencia al cuadrado de la imagen de entrada ajustada $\Phi = \Gamma - \Psi$ y su proyección en el *face space*.

$$\Phi_f = \sum_{i=1}^{M'} \omega_i \mathbf{u}_i$$

$$\epsilon^2 = \|\Phi - \Phi_f\|^2$$

Del cálculo de ϵ pueden obtenerse cuatro resultados respecto a la imagen de entrada:

1. Que sea próxima al *face space* y próxima a una clase (individuo reconocido).

2. Que sea próxima al *face space* pero no a una clase conocida (individuo desconocido pero presente).
3. Que sea lejana al *face space* y próxima a una clase.
4. Que sea lejana al *face space* y lejana a una clase.

RESULTADOS

Después de realizar diversas pruebas cambiando el valor (*Eigen Distance Threshold*), que representa el umbral trabajado en la aplicación, se obtiene la mínima distancia entre los *eigenfaces*, usando la distancia euclídeana (Emgu CV, 2013).

En la etapa de interpretación se obtienen dos resultados:

1. Cara Reconocida: “Puede Ingresar” en color verde (Figura 11).
2. Cara No Reconocida: “No Puede Ingresar” en color rojo (Figura12).



Figura 11. Detección de alumno con código correcto



Figura 12. No detección de alumno con código diferente

DISCUSIÓN

Después de realizar diversas pruebas y contrastar las bases históricas recolectadas en el marco teórico, la aplicación de reconocimiento tiene algunas restricciones como:

- Ángulo e intensidad de la luz.
- *Background* de imagen.
- Oclusiones del rostro entrante.
- Edad.

Debido a esas restricciones, se afirma que la aplicación implementada trabaja en base a un historial de imágenes del usuario en diferentes ángulos, contrarrestando los problemas futuros de oclusión del rostro de los estudiantes registrados.

CONCLUSIONES

1. Al término del proyecto se ha logrado desarrollar una aplicación que funciona en tiempo real y permite la identificación de rostros, para permitir controlar el acceso a la Biblioteca Daniel Rodríguez Hoyle.
2. La aplicación es una propuesta de desarrollo que de ser implementada en un entorno de trabajo grande, podría funcionar correctamente sin modificaciones significativas.

Agradecimientos

Esta investigación fue realizada bajo la supervisión del Ing. Juan Orlando Salazar Campos, a quien expresamos nuestro más profundo agradecimiento.

Nota. Todas las imágenes contenidas en este artículo tienen la respectiva autorización de sus autores y protagonistas.

BIBLIOGRAFÍA

- Álvarez, D. A., Fetecua, J. G., Orozco, Á. Á. y Castellanos, C. G. (2010). *Feature extraction of facial action units combining kernel methods and independent component analysis*. Revista Facultad de Ingeniería Universidad de Antioquia, (56), 130-140.
- Diez Palacio, I. (2010). *Reconstrucción de escenarios virtuales y desarrollo de aplicaciones de realidad aumentada para rehabilitación cognitiva* (Tesis de Máster). Universidad del País Vasco, Donostia.
- Emgu CV. (2013). Emgu CV: OpenCV in .NET (C#, VB, C++ and more). Recuperado 25 de noviembre de 2013, a partir de http://www.emgu.com/wiki/index.php/Main_Page
- Open CV. (2013). OpenCV | OpenCV. Recuperado 25 de noviembre de 2013, a partir de <http://opencv.org/>