

Universidad de las Ciencias Informáticas

Facultad 10



**Título: Propuesta de un sistema de reconocimiento de rostros para  
MOCIC**

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas

**Autor:** Maria Victoria Fornaguera Rodríguez

**Tutor:** Ing. Kiuver Kaddiel Ibañez Castro

**Ciudad de la Habana, 15 de junio del 2009**

“Codifica siempre como si la persona que finalmente mantendrá tu código  
fuera un psicópata violento que sabe dónde vives”

Martin Golding

## Declaración de autoría

Declaro ser la única autora del trabajo titulado:

---

---

y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
Maria Victoria Fornaguera Rodríguez  
Autor

\_\_\_\_\_  
Ing. Kiuver Kaddiel Ibañez Castro  
Tutor

## **Agradecimientos**

## **Dedicatoria**

## Resumen

En la Universidad de las Ciencias Informáticas se está implementando, desde el año 2005 a petición de la Oficina de Seguridad para las Redes Informáticas, un sistema de filtrado que pretende; de forma flexible, regular los contenidos inadecuados que provienen de Internet. Este sistema de filtrado tiene como componente principal una Base de Datos de URLs Categorizadas mediante la cual se toma la decisión de autorizar o denegar el acceso a los recursos solicitados. Dado que es inviable la categorización de las URLs de forma manual se necesita para esto un proceso automático por lo que surge Motor de Categorización Inteligente de Contenido. Este motor es el encargado de clasificar de forma automática las páginas HTML, para poder dar un criterio general de una página HTML es necesario la categorización tanto de su texto como de sus imágenes. Las imágenes constituyen un componente esencial en las páginas HTML, estas son mayormente usadas para hacer más atractivos los contenidos, sin embargo también están contribuyendo a la existencia de contenidos dañinos, que a su vez pueden ser ilegales, tales como la pornografía y la pedofilia. Motor de Categorización Inteligente de Contenido (MOCIC) necesita la detección e identificación de rostros para mejorar su efectividad. En este trabajo se realiza un estudio profundo de algunos métodos para este fin y se proponen los más factibles para MOCIC. También se implementa un prototipo funcional para la detección de rostros al cual se le realizaron pruebas que mostraron un correcto funcionamiento lo que demuestra que puede ser satisfactoriamente integrado a MOCIC.

# Índice general

---

<b>Introducción</b>	<b>1</b>
<b>1. Fundamentación teórica</b>	<b>5</b>
Introducción . . . . .	5
1.1. Actualización . . . . .	5
1.1.1. Internacional . . . . .	6
1.1.2. Nacional . . . . .	8
1.2. Historia del reconocimiento de rostros . . . . .	8
1.3. Detección de rostros . . . . .	10
1.3.1. Segmentación . . . . .	11
1.3.2. Áreas potenciales . . . . .	13
1.3.3. AdaBoost . . . . .	13
1.3.4. Detección Geométrica . . . . .	16
1.3.5. Detección utilizando elipses . . . . .	23
1.3.6. Redes Neuronales Artificiales para la detección de rostros . . . . .	24
1.4. Identificación de rostros . . . . .	25
1.4.1. Algoritmos para la reducción de vectores . . . . .	26
1.4.2. <i>k</i> Nearest Neighbour (k-NN) . . . . .	29
1.4.3. Redes Neuronales Artificiales para la identificación de rostros . . . . .	30
1.5. Herramientas . . . . .	31
Conclusiones . . . . .	32
<b>2. Solución propuesta</b>	<b>33</b>

Introducción . . . . .	33
2.1. Propuesta de métodos para la detección de rostros . . . . .	33
2.1.1. Entrenamiento del detector . . . . .	36
2.1.2. Colección de entrenamiento . . . . .	36
2.1.3. Pruebas . . . . .	40
2.2. Propuesta de métodos para la identificación de rostros . . . . .	42
2.2.1. Entrenamiento del identificador . . . . .	44
2.3. Integración con MOCIC . . . . .	46
2.3.1. Arquitectura . . . . .	47
2.3.2. Implementación de un prototipo funcional . . . . .	49
Conclusiones . . . . .	51
<b>Conclusiones</b>	<b>52</b>
<b>Recomendaciones</b>	<b>53</b>
<b>Referencias</b>	<b>54</b>
<b>Bibliografía</b>	<b>55</b>
<b>A. Prueba 1</b>	<b>58</b>
<b>B. Prueba 2</b>	<b>61</b>
<b>C. Prueba 3</b>	<b>64</b>
<b>D. Fichero de configuración</b>	<b>67</b>
<b>Glosario de términos</b>	<b>70</b>



# Introducción

---

Actualmente en la Universidad de las Ciencias Informáticas (UCI) se está desarrollando un filtro de contenidos llamado Filtrado de Paquetes por Contenido (FILPACON) para regular los contenidos inadecuados que provienen de Internet. Uno de los módulos principales de FILPACON es el MOCIC el cual posee como componentes fundamentales los categorizadores un Categorizador Automático de Texto (CAT), un Categorizador Automático de Imágenes Desnudas (CAID) y un Módulo decisor (MD) los cuales se comunican con el Módulo controlador (MC) el cual es el encargado de manejar el proceso de categorización.

Este último mediante reconocimiento de patrones es capaz de categorizar las imágenes en pornográficas y no pornográficas basado en siete características que son extraídas a las imágenes, hasta el momento estas características son insuficientes por lo que el CAID posee actualmente una efectividad de un 85 %, esto se pretende mejorar añadiendo dos características relacionadas con el rostro, mediante la detección de este.

La primera característica relacionada con el rostro, que es la cantidad de rostros que posee la imagen, esta característica está dada por el hecho de que una imagen pornográfica es poco probable que contenga muchos rostros, como muestra la Figura 1 además, el espacio ocupado por estos rostros aumentan la cantidad de piel en la imagen lo cual aumenta la probabilidad de que la imagen sea pornográfica sin serlo.

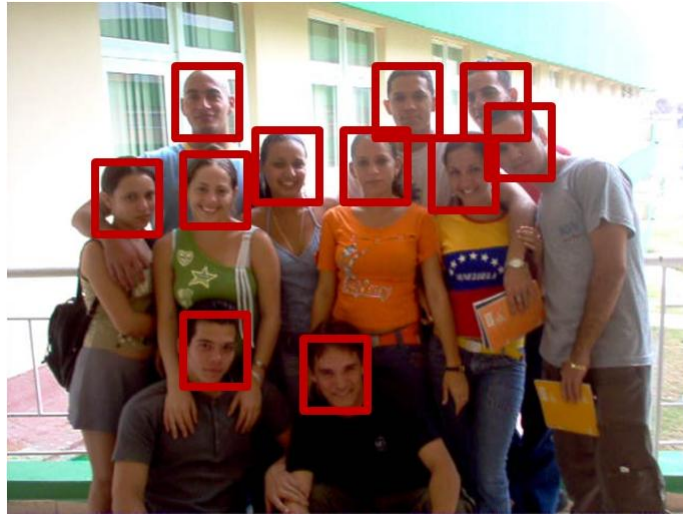


Figura 1: Rostro completo en una imagen

Las características extraídas para la categorización de las imágenes pornográficas están basadas en el color de la piel dado que estas imágenes muestran, generalmente, personas desnudas. Una gran cantidad de piel en una imagen aumenta la probabilidad de que esta sea pornográfica, esto da al traste con el hecho de que una imagen puede contener enteramente un rostro, dichas imágenes contienen gran cantidad de piel y no son pornográficas, como muestra la Figura 2, esto además es muy común. La segunda característica relacionada con el rostro, que es el espacio que ocupa el rostro en la imagen, pretende resolver la problemática planteada anteriormente.

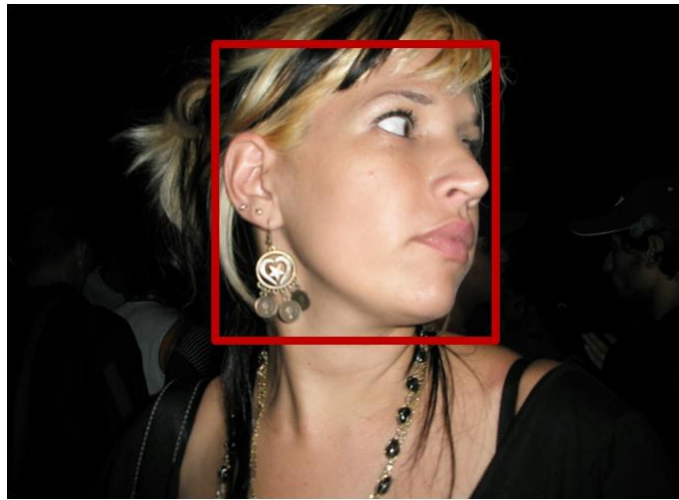


Figura 2: Rostro completo en una imagen

El MOCIC pretende ser una solución general para la categorización de documentos HTML por lo que en cierto momento necesita identificar los rostros que contiene una página HTML, para poder determinar a quién o quienes referencia dicha página HTML. Un componente que identifique el rostro de una persona en una imagen permitiría a este Motor dar un seguimiento a una persona específica en Internet, además mejoraría la efectividad de la categorización del sitio HTML en general ya que las personas pertenecen a ciertas categorías como son los actores, cantantes, deportistas, gobernantes, entre otros.

La detección de rostros se refiere al posicionamiento de un rostro en una imagen, o sea detectar en qué lugar de la imagen se encuentra un rostro, mientras que la identificación de rostros se refiere a determinar a quién pertenece un rostro detectado. Como se puede observar la identificación de rostros contiene de forma implícita la detección de este.

Dada la necesidad de un componente, en el MOCIC, que permita identificar el rostro de una persona en una imagen surge el siguiente problema científico, ¿Cómo detectar e identificar, en una imagen digital, el rostro de una persona para así contribuir a la efectividad de MOCIC?

Por lo anteriormente expuesto se defiende la idea de que la posibilidad de detectar e identificar los rostros, en imágenes digitales, podrá mejorar la efectividad de MOCIC siendo así más confiable la categorización.

El objeto de estudio de este trabajo se enmarca en el proceso de reconocimiento de rostros y su campo de acción en los métodos utilizados para la detección e identificación de estos.

El objetivo general de este trabajo es proponer los métodos adecuados para la detección e identificación rostros para de esta forma contribuir a la efectividad de MOCIC el cual se subdivide en los siguientes objetivos específicos:

- Seleccionar los algoritmos de categorización más eficientes para la detección e identificación de rostros.
- Analizar los métodos para la detección e identificación de rostros.
- Validar la efectividad de los métodos analizados mediante un prototipo funcional.

Para dar cumplimiento a los objetivos planteados se han definido las siguientes tareas de investigación:

- Estudiar los métodos de segmentación más utilizados para la detección e identificación de rostros.
- Estudiar los métodos de aprendizaje y categorización más utilizados para la detección e identificación de rostros.
- Estudiar las Redes Neuronales Artificiales para la detección de rostros.
- Estudiar el algoritmo de k-NN para la identificación de rostros.
- Estudiar los descriptores más relevantes para la detección e identificación de rostros.
- Estudiar las bibliotecas libres para agilizar la implementación de un prototipo funcional.
- Implementar un prototipo funcional para realizar las pruebas a los métodos seleccionados.

---

## Capítulo 1

# Fundamentación teórica

---

## Introducción

En el presente Capítulo se presenta un estudio realizado sobre el reconocimiento y la detección de rostros en el mundo, el cual se enfoca en las principales empresas productoras de este tipo de sistemas así como los métodos y herramientas utilizados para este fin. Se presenta el método utilizado para descartar los objetos que no corresponden con un rostro permitiéndole al detector actuar solo sobre los objetos que son probablemente rostros. Se explica el algoritmo AdaBoost para la detección de objetos, que en el caso específico de este trabajo se utilizará para la detección de rostros, tomando el rostro como un objeto. Se presentan dos técnicas utilizadas para el reconocimiento de rostros siendo estas *Eigenface* y *Fisherface*.

### 1.1. Actualización

Es necesario, antes de comenzar el desarrollo del presente trabajo de diploma, realizar un estudio del campo de acción en el cual se enmarca este trabajo, siendo en este caso el reconocimiento de rostros. Este estudio sirvió para tomar decisiones importantes en el desarrollo del presente trabajo, ya que partiendo de él se descubrieron los métodos y herramientas más apropiadas.

### 1.1.1. Internacional

El reconocimiento de rostros a escala internacional es una técnica avanzada ya que muchas empresas se han visto en la necesidad de implementar estos sistemas para su seguridad. Hoy en día existen muchas empresas dedicadas a desarrollar este tipo de sistema, seguidamente se muestra un listado de algunas de ellas<sup>1</sup>:

- Ex-Sight.Com: Es una compañía de software israelí que se especializa en sistemas precisos de reconocimiento, su tecnología permite un fuerte acabado de reconocimiento-detección facial basado en avanzadas tecnologías. La empresa provee a grandes servicios de desarrollo, a variada ingeniería biométrica, sistemas interactivos y aplicaciones de procesamiento de imágenes tanto para seguridad como para la venta al por menor. Además provee soluciones especializadas para la Banca, Seguridad y Sistemas minoristas. Esta protege la seguridad de las corporaciones e identidad personal usando algoritmos de autenticación biométrica con tecnología de punta.

La compañía esta comprometida a proveer a los clientes con herramientas de reconocimiento de rostros. Esta constantemente mejorando y extendiendo su plataforma para direccionarla a un gran rango de aplicaciones, ofrece una amplia gama de tecnología de reconocimiento de rostros designada para un rango de organizaciones. Cuando combinan su comprensiva tecnología de reconocimiento de rostros y su profunda aplicación de seguridad experta provee una solución en la cual se puede confiar totalmente. Disponible en: <http://www.ex-sight.com/>

- IITS: Es una compañía española especializada en la ingeniería biométrica. Desarrolla proyectos de biometría en niveles de consulta, ingeniería, integración y sistemas de acceso de control de ventas. Proporciona soluciones biométricas con un excelente balance entre la calidad y el precio, protegiendo la seguridad y la información usando autenticación biométrica. Las soluciones evitan el uso de contraseñas, usando tecnologías de reconocimiento de rostro y huellas digitales, virtualmente imposibles de duplicar. Colabora con compañías tecnológicas conocidas. Su fin en esta colaboración es el desarrollo de proyectos orientados al cliente así como proyectos de investigación nacional e internacional.

---

<sup>1</sup> Este listado fue extraído del sitio oficial de reconocimiento de rostros: <http://www.face-rec.org/ventors>

Uno de sus productos es:

IITS FRS ACCESS CONTROL(control de acceso): Asegura su entrada con autenticación biométrica. Es una tecnología de punta facilitando seguridad y autenticación biométrica. Es una tecnología de punta que facilita la seguridad del acceso físico biométrico de puertas eléctricamente controladas. ITS FRS ACCESS CONTROL corre examinando de forma automática la detección de la presencia física de un usuario y lo identifica utilizando una base de datos local o remota. Una vez que el usuario es localizado este guarda los detalles del usuario en un log. Disponible en: <http://www.iits.es/english/index.html>

- Smarti: Combina el control de acceso biométrico basado en el reconocimiento de rostro y voz, con una vista de otras características lo cual hace una elección mas fácil para los usuarios a través de un gran rango de categorías. Módulos para el control de acceso: tiempo y asistencia, construcciones automatizadas, intercomunicadores de video y teléfonos IP, videos de vigilancia y mensajería son actualmente incluidos asegurando el dinero. Disponible en: <http://www.tab-systems.com>
- XIDTECH: Usa una simple webcam, en el momento de autenticarse asegura a identidad del usuario mediante verificación biométrica en un proceso automático continuo, basado en la tecnología XID Face Recognition Artificial Intelligence. Disponible en: <http://www.xidtech.com/>
- X-PIN: Colabora con empresas conocidas tecnológica y científicamente. El objetivo de esta colaboración es desarrollar proyectos orientados al cliente así como proyectos de investigación nacional e internacional. Su sistema de identificación facial x-iden esta basado en las cámaras GigE con un procesador empotrado. Estas cámaras posibilitan la ejecución de aplicaciones directamente desde la cámara. En general, mediante el uso de cámaras inteligentes, existe la posibilidad de integrar programas. Disponible en: <http://www.x-pin.com/>

Estos sistemas poseen una alta calidad y eficiencia pero también tienen un costo elevado el cual nuestro país no esta en condiciones de afrontar, por ello se plantea una búsqueda profunda de alternativas mas factibles como son: Herramientas de Software Libre o desarrolladas en nuestro país.

### 1.1.2. Nacional

En el país no existe una solución factible para el reconocimiento de rostros, la referencia que se tiene del uso de las empresas cubanas son programas extranjeros de código propietario por lo que no se pueden reutilizar como un componente, además de esto estos software producen pérdida para estas empresas ya que los precios de este tipo de sistemas suelen ser elevados. Por lo antes expresado se llega a la conclusión de que a escala nacional no existe una solución al problema planteado y se toma como solución la investigación de este tema para el posterior desarrollo.

## 1.2. Historia del reconocimiento de rostros

Los humanos a menudo usan el rostro para reconocer individuos y avances en la computación en las últimas décadas ahora posibilitan reconocimientos similares de forma automática. Al principio los algoritmos de reconocimiento de rostros usaban simples modelos geométricos, pero el proceso de reconocimiento ha madurado como una ciencia de sofisticadas representaciones matemáticas. Mayores avances e iniciativas en los pasados diez a quince años tienen un impulso dentro del campo de la tecnología de reconocimiento de rostros. El reconocimiento de rostros puede ser utilizado para la verificación e identificación (circuito cerrado<sup>2</sup> y circuito abierto<sup>3</sup>).

El reconocimiento automático de rostros es un concepto relativamente nuevo. Desarrollado los 60, el primer sistema semi-automático para el reconocimiento de rostros requería un administrador para localizar las características (tales como los ojos, orejas, nariz y boca) en la fotografía antes de calcular la distancias y proporciones, los cuales eran comparados con datos almacenados. En los 70, Goldstein, Harmon y Lesk usaron 21 marcas específicas tales como el color del pelo y grosor del los labios para automatizar el reconocimiento. El problema de estas primeras soluciones fue que la medida y localización eran calculadas manualmente. En el 1988, Kirby y Sirovich aplicaron el método *Principal Component Analysis* (PCA) y la técnica de álgebra lineal estándar para el problema de reconocimiento de rostros.

---

<sup>2</sup>Cuando se tienen las personas que ya se van a identificar y las mismas tienen acceso a un local determinado

<sup>3</sup>Cuando no se tienen las personas que se van a identificar, el sistema identifica una nueva persona y esta es añadida



Esto es considerado uno de los hitos demostrando que menos de cien valores fueron requeridos para una adecuada normalización de rostro. En 1991, Turk y Pentland descubrieron que mientras usaban la técnica de EIGENFACES el error residual podía ser usado para detectar el rostro, un descubrimiento que permitió un sistema de reconocimiento de rostros automatizada en tiempo real. Aunque este método fue limitado por factores externos, no obstante su creación significa un interesante desarrollo futuro de la tecnología de reconocimiento de rostros. La tecnología primero captura la atención del público desde una aplicación demostrativa en enero del 2001, que capturaba imágenes de vigilancia que eran comparadas con una base de datos. Esta demostración iniciada necesitaba analizar como sería usada la tecnología para satisfacer necesidades nacionales mientras comenzaba a ser considerada como una preocupación pública y social.

Hoy día, la tecnología de reconocimiento de rostros se comienza a usar para combatir la falsificación de pasaporte, apoyar las leyes, identificación de niños perdidos, y minimizar la falsificación de identidad.

### **Principales métodos:**

Existen dos métodos predominantes para el problema de reconocimiento de rostros: geométrico(basado en características) y fotométrico(basado en vista). Como interés de los investigadores en el reconocimiento de rostros continuo e han desarrollado diferentes tipos de algoritmos, tres de los cuales han sido estudiados en la literatura del reconocimiento de rostros: PCA, *linear discriminant analysis* (LDA), y *Elastic Bunch Graph Matching* (EBGM).

- PCA: Comúnmente referido como EIGENFACES, es la técnica pionera desarrollada por Kirby y Sirovich en 1988. Con PCA, la prueba y la galería de imágenes debe ser del mismo tamaño y deben ser previamente normalizadas en la línea de los ojos y la boca y de los elementos dentro de la imagen. El método PCA es usado para reducir la dimensión de los datos por medio la compresión básica y relevar las leyes de estructura dimensional mas efectiva de los patrones faciales. Esta reducción de la dimensión elimina información que no es necesaria y precisamente descompone el rostro en una estructura dentro de los componentes ortogonales conocidos como eigenfaces. Cada rostro puede ser representado en una suma de pesos(vector de características) de los eigenfaces,

estos son almacenados en un arreglo de 1D. Una prueba de la imagen es comparado contra la galería de imágenes por la medición entre las características del vector. El PCA se enfoca en la representación de rostros de frente de lo contrario la imagen tendría pocas características para una identificación. La primera ventaja de esta técnica es que puede reducir los datos necesarios para identificar un individuo entre mil de la información presentada. cuando tratamos con una gran cantidad de rostros, esta técnica se enfrenta a un problema de muestras pequeñas que surge donde hay un numero pequeño disponible entrenando muestras comparadas por la dimensión del espacio de muestras.

- LDA: Es una método estadístico para clasificar muestras de clases desconocidas basadas en un entrenamiento de muestras de clases conocidas. El objetivo de esta técnica es maximizar la varianza entre clases(varios usuarios) y minimizar la varianza dentro de las clases(un mismo usuario). Esta técnica se enfrenta al pequeño tamaño de la muestra, problema que surge cuando hay un pequeño número de muestras de la formación en comparación con la dimensión espacial de la muestra.

### **1.3. Detección de rostros**

La detección de rostros es una etapa fundamental en cualquier aplicación donde se realice algún tipo de análisis facial como por ejemplo reconocimiento de rostros, codificación de vídeo en videoconferencias, interfaces inteligentes hombre-máquina. El objetivo de esta etapa consiste en detectar y localizar la posición de un número indefinido de rostros en una imagen. En general, la detección de rostros es un problema muy complejo ya que los objetos a detectar pueden ser de diferentes colores, expresiones, poses, tamaños relativos o tener condiciones de iluminación diferentes. Se han propuesto numerosas técnicas para la detección de rostros, pero básicamente se podrían agrupar en dos grandes categorías:

- Métodos basados en reglas.

Se establecen relaciones(reglas)entre las diferentes características faciales como por ejemplo la distancia entre ojos o “simetría” del rostro.

- Métodos estadísticos.

Este tipo de algoritmos no asumen ningún tipo de información previa de la tipología de un rostro; sino, que a partir de un conjunto de muestras (imágenes de rostros y de no-rostros) de entrenamiento extraen la información relevante que diferencia un objeto rostro de un objeto no rostro. Este grupo incluye uno de los métodos más referenciados y utilizados actualmente: el detector de rostros AdaBoost.

Existen factores que modifican la apariencia de un rostro en una imagen, esto hace la detección de rostros una tarea desafiante. Algunos de esos factores son la presencia de componentes estructurales en el rostro como barba, bigote, sombrero o gafas, la posición y orientación, la expresión facial, la variación de la iluminación, la oclusión y ruido [1, p. 17].

### 1.3.1. Segmentación

El objetivo principal de este paso es separar las partes principales de una imagen. La segmentación autónoma<sup>4</sup> es uno de los procesos más difíciles en el procesamiento de imagen. Por una parte una segmentación adecuada facilitará mucho la solución del problema y una segmentación errónea conducirá a fallos. Una completa segmentación de una imagen  $R$  comprende la identificación de un grupo finito de regiones  $\langle R_1, R_2, R_3, \dots, R_N \rangle$  tal que:

1.  $R = R_1 \cup R_2 \cup \dots \cup R_N$

2.  $R_i \cap R_j = \Phi, \forall i \neq j$

3.  $P(R_i) = \text{Verdadero}, \forall i$

4.  $P(R_i \cup R_j) = \text{Falso}, i \neq j$

---

<sup>4</sup>Sin necesidad de la intervención de un humano

### Modelado del color de la piel

Existen numerosos trabajos relacionados con el modelado del color de la piel, que abarcan definición implícita, métodos paramétricos y no paramétricos. También se usan diferentes espacios de color, siendo los más usados el RGB, RGB normalizado y HSV[2]. El objetivo de esta tarea es etiquetar cada píxel de la imagen con dos valores posibles: 1 si el píxel posee color de la piel, 0 si el píxel no posee color de la piel. Como resultado se obtiene una imagen binaria donde se distinguen claramente la regiones correspondientes a la piel humana. Este trabajo comenzó usando un modelo explícito del color de la piel en el espacio RGB, que clasificaba un píxel según el siguiente criterio:

$$\begin{aligned} \text{pixel}[R, G, B] \text{ tiene color de la piel si:} \\ R > 95 \text{ y } G > 40 \text{ y } B > 20 \text{ y} \\ \max\{R, G, B\} - \min\{R, G, B\} > 15 \\ \text{y } |R - G| > 15 \text{ y } R > G \text{ y } R > B \end{aligned}$$

Sin embargo el método propuesto por Jones y Regh ofrece mejores resultados; por lo cual se terminó escogiéndolo[3]. Este consiste en un modelo no paramétrico del color de la piel basado en un clasificador bayesiano. Como resultado de la aplicación de este método a una imagen se obtiene un mapa de probabilidad de la piel que no es más que una imagen en escala de grises donde cada píxel posee un valor entre 0 y 1 que indica su grado de pertenencia a la clase piel.

Para realizar la segmentación se define un umbral, entre cero y uno, y todo píxel cuyo valor sea igual o mayor que dicho umbral es etiquetado como piel y como no piel en caso contrario dando como resultado la imagen binaria descrita anteriormente. El umbral que hemos escogido es 0.7. Un umbral más cercano a 0 daría como resultado muchas zonas etiquetadas con color de la piel sin serlas y uno más alto produciría el efecto contrario.

### 1.3.2. Áreas potenciales

El primer paso en un sistema de detección es detectar los píxeles que con el color de la piel de la imagen. El resultado de esto es una imagen segmentada por el color de la piel para quitar pequeñas áreas de ruido. Así, las regiones conectadas con el color de la piel son calculadas usando un algoritmo de etiquetado. Una vez que los píxeles que pertenecen a la clase piel son agrupados en una región conexas, aquellos que su dimensión son claramente dimensiones diferentes a la de los rostros son descartadas. Para hacer este descarte son aplicados 4 diferentes pruebas a la regiones conectadas:

1. Prueba de la mínima y máxima área: Las regiones de piel son menores que el 1 % del total de las áreas de la imagen son descartadas. Las regiones que tienen un área mayor que el 80 % del total de la imagen son descartadas.
2. Prueba de las regiones largas: Toda área que bordea por un cuadrado el cual tiene un ancho menor que el 40 % del largo es descartada. Toda área que bordea por un cuadrado el cual tiene un largo menor que el 40 % del ancho es descartada.
3. Prueba de las regiones esparcidas: Toda región que su área es menor que el 50 % del área de su cuadrado es descartada.
4. Prueba de la proporción: Si la proporción ancho/largo de la región es mayor que 1.6, la altura es disminuida hasta que la proporción sea menor que 1.6.

Una vez descartadas las regiones explicadas, las otras regiones son las que se pasan al detector de rostros. [1, p. 20].

### 1.3.3. AdaBoost

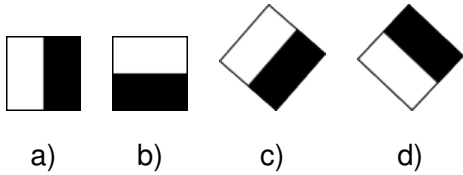
Primero el clasificador es entrenado con imágenes de cientos de muestras de un objeto en particular (e.j. un rostro o un carro) llamado muestra positiva, que son redimensionadas al mismo tamaño (e.j.  $20 \times 20$ ),

y muestras negativas que son imágenes arbitrarias del mismo tamaño.

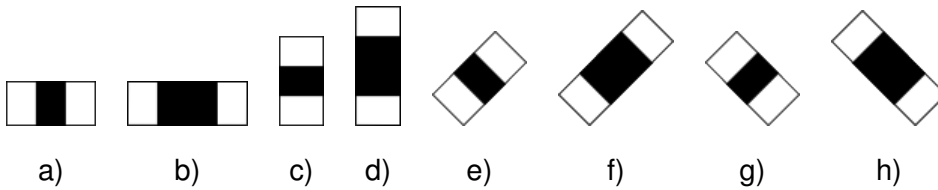
Después que el clasificador es entrenado puede ser aplicado a una región de interés (del mismo tamaño de las muestras usadas en el entrenamiento) de una imagen determinada. El clasificador devuelve 1 si la región es parecida al objeto para el cual fue entrenado, y 0 de otra forma. Para buscar el objeto en la imagen completa se mueve una ventana por toda la imagen y chequea en cada lugar usando el clasificador. El clasificador está diseñado para que pueda ser fácilmente redimensionado para que así pueda encontrar el objeto buscado con diferentes tamaños, lo cual es más eficiente que redimensionar la imagen en si. Para encontrar un objeto de un tamaño desconocido en una imagen el procedimiento debe de hacerse varias veces con diferentes tamaños de ventana.

La palabra “*cascade*” en el nombre del clasificador significa que la clasificación resultante consiste en varios clasificadores simples (*states*) que son aplicados sucesivamente a una región de interés hasta que en algún estado el candidato es rechazado o pasa todos los estados. La palabra “*boosted*” significa que los clasificadores en cada estado de la cascada son en si mismos complejos y son construidos de clasificadores básicos usando una de las cuatro diferentes técnicas de boosting (*Discrete Adaboost*, *Real Adaboost*, *Gentle Adaboost* y *Logitboost*). Los clasificadores básicos son clasificadores de arboles de decisión con al menos 2 salidas. Las características *Haar-like* son las entradas de los clasificadores básicos, y son calculados como se describe a continuación. El algoritmo actual usa las siguientes características de *Haar-like*.

## 1. Características de vértices



## 2. Características de líneas



## 3. Características de centro y borde

Figura 1.1: Características de *Haar-like*

Las características usadas en un clasificador en particular es especificada mediante la forma de la Figura 1.1 (1a, 2b etc), posición en la región de interés y tamaño (este tamaño no es el mismo que el usado en los estados de detección, aunque esos tamaños son multiplicados). Por ejemplo, en el caso de la primera línea de característica (2c) la respuesta es calculada como la diferencia entre la suma de los píxeles de la imagen en el rectángulo convergiendo en toda la característica (incluyendo las dos líneas blancas y la línea negra en el medio) y la suma de los píxeles de la imagen en la línea negra multiplicado por 3 en para compensar la diferencia en el tamaño de las áreas. Las sumas de los valores de los píxeles en la región rectangular son calculados rápidamente usando imágenes integrales [4, 5].

Este método ofrece una buena rapidez, 15 imagen por segundo en una PC de 700 MHz Intel Perntium III [1, 5].

Este método todavía no puede resolver el problema de rotación de rostros en imágenes de rostros reales. Aunque este problema puede ser resuelto mediante la rotación de la imagen y aplicando el método

muchas veces, eso significa un incremento de la complejidad computacional y falsas alarmas. Entonces se necesita desarrollar una técnica especial que mejore la eficiencia del detector.

### 1.3.4. Detección Geométrica

#### Detección de Ojos

Para poder encontrar la ubicación de los ojos y la boca dentro de un posible candidato a cara se realizaron mapas derivados directamente de la luma y de las cromas de la imagen original. Un “mapa” es una imagen en tonos de grises que resalta determinadas características relevantes de la imagen original, para facilitar la detección de algo concreto, en este caso de ojos y boca. Para la construcción de dichos mapas se trabajó en el espacio YCbCr (Luma, Cromo azul, Cromo roja). Para construir el mapa de ojos (*EyeMap*) se basamos en dos puntos:

- En los ojos hay poco rojo (bajo Cr) y bastante azul (alto Cb).
- En los ojos existen píxeles bien claros así como bien oscuros.

Considerando la primer sentencia, se forma el siguiente mapa de cromas:

$$EyeMapC = \frac{1}{3} \left( C_B^2 + (255 - C_R)^2 + \frac{C_B}{C_R} \right) \quad (1.1)$$

Cada término de la suma está normalizado al rango  $[0, 255]$ .

Para implementar la segunda sentencia, se utilizó morfología en escalas de grises. Para esto utilizamos un elemento de estructura hemisférico  $g_\sigma$ . La idea es resaltar aquellas zonas que tienen alto contenido de Y y que a su vez tienen vecinos cercanos con bajo contenido de Y. Para esto se dilató la componente



Y (imagen de luma) obteniendo así una imagen que resalta las zonas donde hay alto contenido de luma - brillos. Por otro lado se erosionó la Y, resaltando (hacia el 0) las zonas donde hay poco componente de luma. Si el elemento de estructura es lo suficientemente grande, las zonas que tienen alto y bajo componente de luma van a aparecer en las dos imágenes resaltadas (en el primer caso resaltadas positivamente y en el segundo negativamente). Luego se realizó la división entre la imagen dilatada y la erosionada obteniendo así el mapa de ojos considerando la luma. En resumen las operaciones realizadas son las siguientes:

$$EyeMapL = \frac{Y(i, j) \oplus g_s(i, j)}{Y(i, j) \ominus g_\sigma(i, j) + 1} \quad (1.2)$$

El tamaño del elemento de estructura utilizado es una medida del mínimo tamaño de ojo a encontrar en la imagen, se estima como :

$$\sigma = \frac{\sqrt{W \cdot H}}{2F_E} \quad (1.3)$$

Siendo:

- $W$ : Ancho de la imagen (Máscara)
- $H$ : Alto de la imagen (Máscara)
- $F_E$ : Cociente entre máximo tamaño de ojo sobre tamaño promedio de cara. Se toma  $F_E = 7$  píxeles
- $\oplus$ : Dilatación
- $\ominus$ : Erosión

Para juntar ambos mapas, primero se normaliza al rango  $[0, 255]$  y luego se combina mediante la multiplicación ( $AND$ ) y el resultado fue enmascarado y nuevamente normalizado al rango  $[0, 255]$ .

Para decidir cuales son las regiones de ojos, se realizó un umbralizado adaptivo. El umbral de decisión es una combinación lineal entre el promedio del mapa y el máximo valor del mapa, es decir:

$$Umb = \alpha \cdot promedio(EyeMap) + (1 - \alpha) \cdot Max(EyeMap) \quad (1.4)$$

Siendo  $\alpha$  un parámetro a ajustar.

Finalmente se obtiene un mapa binario con todos los píxeles que están por encima del umbral. Para terminar esta etapa se realiza un filtrado morfológico (mediante un elemento de estructura circular) para eliminar aquellas regiones que no tienen un mínimo de tamaño, aplicándole primero apertura y luego cerradura, obteniendo así el cierre de la apertura del mapa de ojos binario.

Luego se realizó una etapa de etiquetado, donde se identificaron aquellas regiones candidatas a ser ojos [6].

Una vez que los dos ojos son detectados, la posición de las cejas puede ser localizada dentro de una pequeña región arriba del centro del ojo. Seguidamente puede ser aplicado un mapa para buscar la ceja a lo largo de una columna del iris a la mitad de cada ceja. Algunas de las características de la imagen frontal son invariables, las cuales no cambian con la expresión facial, mientras que hay otras características variables que son las que varían cuando cambia la expresión del rostro. Entre todas las características faciales frontales, los ojos tienen un rol significativo en el proceso de reconocimiento [7, p. 255].

### **Detección de la boca**

Para construir el mapa de boca (*MouthMap*) nos basamos en un único punto:

- En la boca hay más componente roja comparada con la componente azul, que en el resto de la

cara.

Observando lo anterior y el hecho de que en la boca hay baja respuesta a  $C_r/C_b$ , se supone el siguiente MouthMap basado únicamente en la información de las cromas:

$$MouthMap = C_R^2 \left( C_R^2 - \eta \frac{C_R}{C_B} \right)$$

Siendo  $\eta$  el cociente entre la media de  $C_R^2$

$$\eta = 0,95 \frac{\text{promedio} (C_R^2)}{\text{promedio} \left( \frac{C_R}{C_B} \right)}$$

Todas las componentes de las ecuaciones anteriores son normalizadas a  $[0, 255]$ .

Finalmente se realizó una etapa de enmascarado (con la máscara de cara) y luego se dilató (utilizando un elemento de estructura hemisférico) buscando regularizar el mapa y resaltar las zonas mas claras.

### Umbralizado y Etiquetado

Para decidir cuáles son las regiones de boca, se realizó un umbralizado adaptivo similar al realizado en el *EyeMap*. El umbral de decisión es una combinación lineal entre el promedio del mapa y el máximo valor del mapa, donde los coeficientes de ponderación son determinados de forma experimental.

Finalmente obtenemos un mapa binario con todos los píxeles que están por encima del umbral. Para terminar esta etapa se realiza un filtrado morfológico para eliminar aquellas regiones que no tienen un mínimo de tamaño, aplicándole primero apertura y luego cerradura, obteniendo así el cierre de la apertura del mapa de ojos binario.

A continuación se realizó un etiquetado de las regiones encontradas como candidatos a boca, para luego facilitar el cálculo de algunas de sus características (área,centroide,ubicación).

### **Restricciones Geométricas: Validación**

Por último se realizó una etapa que combina los resultados encontrados en las etapas anteriores con el fin de validar si la región de piel considerada es una cara o no. En el caso en que se considere como cara, se debe indicar la (mejor) ubicación de los ojos y la boca. Para poder realizar lo anterior se forman todas las combinaciones posibles entre los candidatos a ojos y boca hallados (si tenemos  $n$  bocas,  $m$  ojos entonces hay  $n$  combinaciones de  $m$  tomadas de a dos ternas posibles). A cada combinación se denomina candidato. Para cada candidato, que supere un conjunto mínimo de restricciones se calcula un Valor, que contempla que tan buen candidato es.

Con cada par de ojos y cada boca se forma un triángulo cuyos vértices son los centroides de las regiones asociadas a los ojos y boca en consideración.

Primero se aplica una etapa, que es llamada de mínimas condiciones, en la cual si un candidato no las cumple es eliminado. Ésta consiste en:

- El triángulo formado por los tres centroides (2 ojos, 1 boca) debe ser agudo (no puede tener ninguno de sus ángulos mayor a  $90^\circ$ )
- El triángulo debe tener un área mínima de 100 píxeles (esta restricción está asociada al tamaño mínimo de cara a detectar)

Luego para cada candidato que cumpla las restricciones anteriores se le calculan los siguientes Valores:

- Valor de simetría y orientación.
- Valor de simetría de tamaño de ojos.

- Valor de coherencia de tamaño de cara.

Los valores están formados básicamente por términos de la forma de campanas tipo gaussianas. Dichas campanas están centradas en el valor considerado ideal y con un ancho de lóbulo tal que contemple que tan significativo es separarse de dicho valor ideal.

Valor de simetría y orientación

Para el cálculo de este valor se tuvo en cuenta dos aspectos:

- Simetría: que tan isósceles es el triángulo formado (ángulo  $\theta_1$ )
- Orientación: que tan inclinado respecto a la vertical se encuentra. (ángulo  $\theta_2$ )

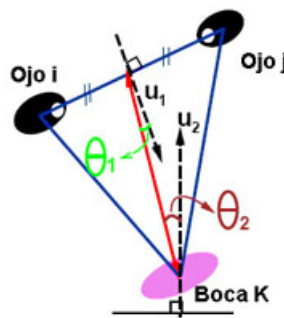


Figura 1.2: Triángulo de verificación

Para el cómputo del valor se realizó lo siguiente:

$$ScOrienta = \prod_{k=1,2} e^{-3 \sin^2 \frac{\theta_k}{2}}$$

Está formado básicamente por Gaussianas que dependen del cuadrado del seno del ángulo. La idea es favorecer a las caras simétricas y a las que se encuentran muy cerca de la vertical (mirando hacia arriba).

**valor de simetría de tamaño de ojos**

Se implementó un valor que contempla positivamente el hecho de que las regiones formadas por cada ojo tengan áreas similares. El valor es una campana de la forma:

$$ScAreaOjos = e^{-\frac{Area_i - Area_j}{\min\{Area_i, Area_j\}}}$$

**valor de coherencia de tamaño de cara**

Mediante un breve estudio estadístico se llegó a la conclusión de que el área del triángulo formado por los ojos y la boca, es el 10% del área total de la cara. Si se aproxima el área de la cara a la de la máscara, se puede implementar un valor para favorecer a los candidatos que cumplan esto, es decir:

$$ScTamano = e^{-\sqrt{Area(Trgulo) - Area(Mask)}}$$

Finalmente los tres valores son agrupados en un único valor mediante una combinación lineal de la siguiente manera:

$$Sc = 0,5 \cdot ScOrienta + 0,25 \cdot ScAreaOjos + 0,25 \cdot ScTamano$$

Se tuvo en cuenta que el valor más determinante para la validación de una cara es el de orientación y simetría.

Por último se devuelve como válido aquel candidato que tenga mayor valor y que su valor sea mayor a 0,6 Si no existen candidatos con valor mayor a 0,6 se considera que el candidato no es cara.

### 1.3.5. Detección utilizando elipses

Este algoritmo realiza procesamientos de la siguiente forma:

- Procesamiento(1):

Localización de áreas del color de la piel utilizando reglas de decisión en el espacio RGB normalizado [1]. Una vez localizadas dichas áreas, se somete a la imagen a operaciones morfológicas básicas para remover elementos que por su tamaño no pueden ser rostros y también para mantener las posibles áreas del rostro concatenadas.

- Procesamiento(2):

Después de segmentar la imagen empleando la información de color (localización de áreas del color de la piel) se la transforma a escala de grises y en esa nueva imagen se aplica la transformada de Hough para la detección de elipses. Debido a que esta técnica es computacionalmente costosa se la aplica bajo ciertas restricciones, además se emplea la imagen en escala de grises reescalada a la cuarta parte. La transformada de Hough trabaja sólo con píxeles de borde por lo que antes de utilizarla se debe realizar la detección de bordes. Para esta tarea se seleccionó el filtro de Sobel.

- Procesamiento(3):

Para cada elipse resultante se realiza lo siguiente:

1. Los parámetros de la elipse (coordenadas del centro y longitud de los semi-ejes) son reescalados con referencia al tamaño original de la imagen en escala de grises.
2. Se extrae de esta imagen el área que abarca la elipse y se le trata como una nueva imagen.
3. Para la nueva imagen se detecta el área de los ojos, el área de la boca, se realiza la validación geométrica del triángulo ojos-boca.
4. Si la puntuación obtenida por el triángulo es mayor a un valor umbral se normaliza la imagen en cuanto a orientación del rostro y luego se extrae el área del rostro como una nueva imagen.
5. Se ajusta la intensidad de la imagen del rostro.
6. Se proyecta esta imagen en el espacio de rostros (PCA) y se calcula la distancia entre la proyección y la imagen para confirmar si es un rostro.

7. Si la distancia es menor a un valor umbral el proceso de localización del rostro finaliza, caso contrario se repiten los pasos 1-7 con la siguiente elipse hasta encontrar un rostro o utilizar todas las elipses en cuyo caso se concluye que la imagen no contiene un rostro humano.

- **Procesamiento(4):**

Luego de tener las elipses candidatas, se les aplica el método de Detección Geométrica como se explica en sección anterior (1.3.4).

### **1.3.6. Redes Neuronales Artificiales para la detección de rostros**

Este algoritmo trabaja aplicando una o mas Redes Neuronales Artificiales (RNA) directamente a la porción de la imagen entrada y arbitrariamente sus resultados. Cada neurona es entrenada para una salida de presencia o ausencia de un rostro, Los algoritmos y los métodos de entrenamientos están diseñados para ser generales, con poca personalización para rostros.

Muchos investigadores tienen la idea que los rostros puede ser caracterizados directamente en términos de intensidad de píxeles. Estas imágenes pueden ser caracterizadas mediante métodos probabilísticos de un conjunto de imágenes de rostros o de forma implícita mediante una red neuronal u otro mecanismo. Los parámetros para estos modelos son ajustados de forma automática con imágenes de muestras o manualmente. Algunos autores han tomado estos métodos de extracción de características y aplicando cualquier regla generada manual o automáticamente para evaluar estas características.

Entrenar una red neuronal para la detección es una tarea desafiante dada la dificultad de caracterizar las imágenes que no son rostros. A diferencia del reconocimiento de rostros en las cuales las clases a ser discriminadas son diferentes rostros, las dos clases a ser discriminadas en la detección de rostros son imágenes que contienen rostro e imágenes que no contienen rostros. Es fácil obtener una muestra representativa de imágenes que contengan rostros, pero muy difícil obtener una muestra representativa de esas imágenes que no contengan rostros. Se evita este problema usando una gran colección de entrenamiento de imágenes que no contienen rostros. El método bootstrap reduce el tamaño de la colección de entrenamiento si es necesario. El uso arbitrario entre múltiples redes y heurísticas para limpiar los



resultados significativamente mejora la efectividad del detector.

### **Descripción del Sistema:**

El sistema opera en dos etapas: Primeramente se aplica a la imagen un conjunto de filtros basados en redes neuronales, y luego se usa un arbitro para combinar las salidas. Los filtros examinan en cada posición de la imagen en varias escalas, buscando la posición donde hay un rostro. El arbitro entonces fusiona las detecciones de los filtros individuales y elimina la superposición de las detecciones.

Este algoritmo puede detectar entre un 77.9% y un 90.3% de rostros en una muestra de 130 imágenes, con un numero aceptable de falsas detecciones. Dependiendo de la aplicación el sistema puede ser hecho mas o menos conservativo por la variación de las heurísticas arbitrarias o usando umbral. El sistema ha sido probado en una gran variedad de imágenes, con muchos rostros y fondos complejos. Una versión ligera del sistema puede procesar imágenes de  $320 \times 240$  píxel en un tiempo de 2 a 4 seg en una PC de 200MHz

Hay muchas áreas para trabajos futuros. La principal limitación de este sistema es que solo detecta rostros de frente y no inclinados. Diferentes versiones del sistema pueden ser entrenados para varias orientaciones de la cabeza. El trabajo preliminar en esta área indica que la detección de la vista de perfil es mas difícil que la detección de la vista frontal porque esta tiene menos características estables y la ventana de entrada contiene mas píxeles de fondo. Se aplica también el mismo algoritmo para la detección de ojos.

## **1.4. Identificación de rostros**

El resultado de la detección de rostros es un conjunto de áreas que representan las posiciones de los rostros detectados, cada una de estas áreas pasa por el proceso de identificación.

### 1.4.1. Algoritmos para la reducción de vectores

La imagen obtenida como resultado de la detección debe estar normalizada en cuanto a orientación y a iluminación. Luego de esto se la normaliza en cuanto a tamaño  $n \times n$  para extraer sus atributos más relevantes. El espacio del rostro es muy grande por lo cual se debe reducir, para esto utilizamos los algoritmos de reducción.

#### EIGENFACE

El método de representación *eigenface* para el reconocimiento de rostros esta basado en el análisis de los principales componentes. La idea principal es descomponer la imagen del rostro en un conjunto de eigenfaces (un pequeño conjunto de características), que son los principales componentes de la imagen original. Estas funciones como el vector ortogonal de un subespacio lineal llamado espacio del rostro. El método de reconocimiento implica la proyección del nuevo rostro en el espacio y comparar su posición en el espacio de rostros con los conocidos. En este método el conjunto de entrenamiento de patrones faciales son convertidos en un vector de  $M \times N \times K$  donde  $M \times N$  es el tamaño de la imagen y  $K$  es el número de muestras de entrenamiento. Este espacio de  $M \times N$ -dimensiones es realmente grande y es importante reducir la dimensiones de este espacio antes de hacer el reconocimiento. PCA y LDA son dos métodos comunes para reducir esta gran dimensionalidad.

La filosofía básica en PCA es mapear de todas las muestras de  $M \times N$ -dimensiones,  $x_i, i = \{1, \dots, k\}$  en un vector simple  $y_i, i = \{1, \dots, n\}$  tal que  $y_i$  representa  $x_i$ , ejemplo:

$$y_i = w^T x_i \quad (1.5)$$

Donde cada  $x_i$  representa un rostro y  $w$  es el peso del vector representado. El objetivo de la función es maximizar la varianza, ejemplo:

$$J = \max \sum_{i=1}^n (y_i, \bar{y})^2 \quad (1.6)$$

Donde:

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i \quad (1.7)$$

El conjunto de vectores de peso  $w_1, \dots, w_k$  representan los eigenvectores de la matriz de covarianza calculada del conjunto de muestra y  $k < MN$ . Esto implica que se seleccionan solo algunos eigenvectores correspondientes a eigenvalues dominantes y así se reduce las dimensiones de las características faciales.

Para un sistema de reconocimiento de rostros preciso, la precisión debe de ser realmente alta y al mismo tiempo el tiempo del procesamiento debe de ser bajo. El sistema debe de ser invariante para una rotación o traslación de cabeza, y también cambios de luminosidad.

## FISHERFACE

Otra técnica interesante, llamada fisherface también usa una proyección lineal de *eigenfaces* y *elastic branch graph matching*. Fisherface usa *Fisher Linear Discriminant Function* (FLDF) el cual permitirá una robusta y mejor precisión de la clasificación del patrón del vector. FLDF crea espacios separables linealmente con lo cual es eficiente en discriminar los patrones. FLDF un número de funciones lineales con las que parte el espacio del rostro en  $k$  clases distintas.

Mientras se usa el método Fisherfaces, los patrones de entrenamiento incluyen no solo un rostro sino varias expresiones del rostro. Que son  $n_i$  muestra de diferentes expresiones del  $i$ -ésimo clase de rostro y  $m^{(i)}$  representa el promedio del vector de característica de la  $n_i$  muestra de la  $i$ -ésimo clase. El objetivo de la función es maximizar si el caso es:

$$J_F = \max \left\{ \frac{W^T S_B W}{W^T S_W W} \right\} \quad (1.8)$$

Donde  $S_B$  está entre las clases de la matriz dada por:

$$S_B = \sum_{i=1}^k n_i (m^{(i)} - m)(m^{(i)} - m)^T \quad (1.9)$$

y  $S_W$  está dentro de las clases de la matriz dada por:

$$S_W = \sum_{i=1}^k \left[ \sum_{j=1}^n (X_j^{(i)} - m^{(i)})(X_j^{(i)} - m^{(i)})^T \right] \quad (1.10)$$

En las ecuaciones (1.8), (1.9), y (1.10),  $m$  representa la media del vector de muestra,  $n_i$  es el número de muestras en la  $i$ -ésima clase, y  $x_j^{(i)}$  representa la  $j$ -ésima expresión facial del  $i$ -ésimo rostro y  $W$  es la matriz de proyección.

El objetivo de esta función es maximizar la distancia Euclídea entre el rostro que pertenece a diferentes personas, mientras minimiza la distancia entre el rostro que pertenece a una misma persona. La dirección de la proyección que minimiza los  $J_F$  produce la columna del vector de la proyección de la matriz  $W$ . El objetivo de la función  $J_F$  es maximizar cuando la columna del vector de  $W$  es el eigenvector de  $S_W^{-1} S_B$ . Debe de notar que aquí si la dimensión del espacio de muestra es mas larga que el número de muestras en el conjunto de entrenamiento,  $S_W$  se convierte en singular y así su inverso no existe. Muchas estrategias han sido probadas para resolver esta situación.

Tanto el PCA como el LDA preservan la estructura global del espacio del rostro, pero el método *Fisherface* tiene mejor capacidad de discriminación comparado con método *Eigenface* [7, pp. 257-259].

### 1.4.2. k-NN

Primero son detectados un conjunto de puntos del rostro, que son usados luego para la medición de las características basada en el área, ángulo y distancia entre ellos. La extracción de características del frente del rostro están dada por los puntos seleccionados del lado frontal del rostro [7, p. 254].

Algunas características variables e invariables son mostradas a continuación:

#### Invariables

- Distancia entre el centro de los ojos.
- Distancia entre los puntos del ojo interno
- Distancia entre los puntos del ojo externo
- Distancia del centro (punto medio de la distancia entre el centro de los iris) de los ojos a la punta de la nariz.

#### Variables

- Distancia entre el centro del iris y la ceja de cada uno de los ojos (en la misma columna).
- Ancho de la cara a la altura de la punta de la nariz.

Para reducir la complejidad computacional los valores de los vectores de características con sus respectivos índices(clases) son guardados de forma ascendente. El algoritmo de clasificación k-NN para la búsqueda de rostros es como sigue:

**Paso 1.** Disponer los vectores de características con sus respectivos índices en forma ascendente.

- Paso 2.** Insertar cada vector de característica de la muestra de prueba a su lugar correspondiente.
- Paso 3.** Determinar de dos vecinos cercanos en cada columna y calcular la similitud de la muestra de ambos vecinos, cada uno con su índice.
- Paso 4.** Calcular la similitud de toda la muestra con todos a todos los vectores.
- Paso 5.** Asignar el índice que tiene la mayor similitud con el rostro.

Si el rostro del fichero de prueba no pertenece a ninguno de los que están en el fichero principal, se le asigna un índice fuera de las clases representadas. Este algoritmo descrito posee buenos resultados en el reconocimiento de rostros.

### **1.4.3. Redes Neuronales Artificiales para la identificación de rostros**

Una vez que se cuenta con el conjunto de vectores de características que representan al conjunto de imágenes de entrenamiento habiendo aplicado uno de los métodos anteriormente explicados, se crea una red neuronal de tipo *Radial Basis Function* (RBF) y se la entrena con dichos vectores. Para este tipo de red se debe definir previo a su entrenamiento el ancho más adecuado de la gaussiana para lograr una buena generalización a partir del conjunto de vectores proporcionados.

#### ***Radial Basis Function***

Son un método alternativo a el MLP para hacer ajuste a funciones no lineales. A diferencia de la disposición que se tiene en la funciones de activación que permite construir modelos de entrenamiento mediante backpropagation, estas nuevas redes basadas en RBF construyen sus modelos con funciones de activación que son diferente tanto en la capa oculta como la de salida. Una red RBF está diseñada con neuronas en la capa oculta activadas mediante funciones radiales de carácter no lineal con sus centros gravitacionales propios y en la capa de salida mediante funciones lineales.

A diferencia de las *Multilayer Perceptron* (MLP), el modelo clásico de las redes RBF está construido con una arquitectura rígida de tres capas: la de entrada, la oculta y la de salida.

En general, una red RBF tiene un mejor desempeño con un mayor volumen de datos de entrenamiento. La construcción de una red RBF requiere de una mayor cantidad de neuronas en los nodos ocultos que en las redes que usan backpropagation.

Aunque las redes RBF no son comúnmente utilizadas en aplicaciones que impliquen un alto volumen de patrones de entrenamiento, se le reconoce como una red con una alta eficiencia en la fase de entrenamiento [8].

## 1.5. Herramientas

Para el desarrollo de la investigación se utilizaron algunas herramientas para facilitar la implementación de un prototipo funcional, así como la evaluación de la efectividad del mismo a partir de los resultados arrojados en las pruebas realizadas.

Las ventajas principales de un IDE radica en la automatización de algunas tareas asociadas al desarrollo de aplicaciones y al poseer algunas funcionalidades que facilitan el desarrollo de software disminuyendo así el tiempo de construcción. La selección de un IDE para el desarrollo de aplicaciones en C++ en el sistema operativo GNU/Linux se hace hasta cierto punto difícil, ya que ninguno de los IDEs existentes completa código, lo cual es una de las características que más agiliza el desarrollo de software, al no tener que memorizar y teclear todas las funciones existentes en otras clases. Finalmente se encontró uno que si hace esta tarea y es el caso de Eclipse con un plugging llamado CDT, el cual tiene las siguientes características:

- Editor de C/C++ (Funcionalidades básicas, resaltamiento de sintaxis, completamiento de código, etc)

- Depurador C/C++ (Usando GDB)
- Lanzador C/C++ (Lanzadores y aplicaciones externas)
- Parser
- Herramienta de búsqueda
- Proveedor de asistencia de contenido
- Generador de Makefile

Es justo mencionar algunas herramientas auxiliares utilizadas para la confección del presente documento que, aunque no formaron parte del desarrollo del prototipo funcional, si contribuyeron a disminuir el tiempo y aumentar la calidad del trabajo; ellas son:

- LATEX: Es un sistema tipográfico para generar documentos científicos de alta calidad, lo cual permitió centrarse en el documento dejando a este la tarea de generar un documento que cumpliera con la calidad requerida para una tesis de grado.
- Kile: Es un IDE para LATEX el cual permitió un desarrollo fluido del documento.

## Conclusiones

Luego de analizar los softwares de detección y reconocimiento de rostros en los ámbitos internacional y nacional se evidencia la necesidad de diseñar e implementar uno propio, ya que los existentes, en el ámbito internacional, no cumplen con las características requeridas por FILPACON y en el ámbito nacional no se tienen referencias de la existencia de alguno. Mediante el estudio de las técnicas de IA, herramientas y tecnologías a utilizar, se han sentado las bases para el desarrollo de un prototipo funcional capaz de procesar de manera automática la detección de rostros.



---

## Capítulo 2

# Solución propuesta

---

## Introducción

En el presente capítulo, se brinda en general un estudio de los diferentes métodos tanto para la detección como para la identificación, sus principales características, y la propuesta a realizar en cuanto a los métodos que sean adaptables a MOCIC. Se precisan los elementos principales sobre dicha propuesta, y los aspectos fundamentales por los que fueron propuestos.

### 2.1. Propuesta de métodos para la detección de rostros

Proponer un método para la detección de rostros se hace hasta cierto punto difícil dado los múltiples caminos que se pueden tomar mediante la combinación de los métodos estudiados como muestra la Figura 2.1, se mencionaran algunas de estas combinaciones para ayudar a confirmar la propuesta.

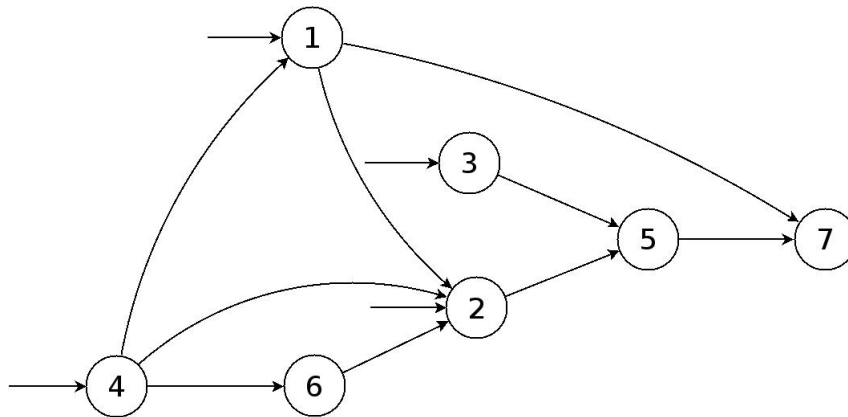


Figura 2.1: Combinaciones de métodos

A continuación se muestra la leyenda del grafo en el Cuadro 2.1 y los caminos del mismo que pueden ser tomados para la selección en el Cuadro 2.2.

1	AdaBoost
2	AdaBoost para detectar ojos y boca
3	Modelo del color para detectar ojos y boca
4	Segmentación de la imagen por el color de la piel
5	Validación del triangulo formado por los ojos y la boca
6	Detección de rostros candidatos mediante elipses
7	Detección de rostros

Cuadro 2.1: Leyenda

A	1 - 7
B	1 - 2 - 5 - 7
C	4 - 1 - 7
D	4 - 1 - 2 - 5 - 7
E	3 - 5 - 7
F	2 - 5 - 7
G	4 - 6 - 2 - 5 - 7
H	4 - 2 - 5 - 7

Cuadro 2.2: Caminos

Seleccionar la combinación E implica que no se puede utilizar la detección de las áreas potenciales mediante la segmentación de la imagen por el color de la piel ya que la imagen segmentada descartaría el color blanco de los ojos y el rojo fuerte de los labios, colores que necesita este modelo matemático para detectar los ojos y boca, esto podría significar la detección incorrecta de rostros en áreas que pudieron haber sido eliminadas para mejorar la detección. Además, tanto este como las combinaciones F , G y H realizan la validación entre todos los pares de ojos detectados con todas las bocas por lo que esto implicaría un modelo matemático muy complicado.

Los métodos B y H resuelven el problema de validar todos los pares de ojos con todas la bocas ya que enmarcan estas en regiones similares pero esto trae consigo un procesamiento adicional buscando primero un candidato a cara y luego comprobarlo mediante el método geométrico.

Para la selección del método a proponer se tomaron en cuenta algunos elementos como son la velocidad y efectividad de la detección así como el conocimiento del equipo de desarrollo de las herramientas necesarias para la implementación del método propuesto. De los métodos estudiados el más veloz es el AdaBoost(1.3.3) además de tener una excelente efectividad, por lo cual se seleccionó como propuesta. Existen varias combinaciones con este método como se dijeron anteriormente de estos la combinación de métodos A y C cumplen con los elementos necesarios para ser seleccionados.

### 2.1.1. Entrenamiento del detector

Como se dice en la Sección 1.3.3 el clasificador es entrenado con imágenes positivas y negativas de  $20 \times 20$  píxeles, las positivas son imágenes que contienen enteramente un rostro, como se muestra en la Figura 2.2, y las negativas imágenes arbitrarias como se muestra en la Figura 2.3. Luego de esto el clasificador es capaz de detectar el rostro de una persona de forma vertical y de frente.



Figura 2.2: Muestras imagenes positivas



Figura 2.3: Muestras imagenes negativas

### 2.1.2. Colección de entrenamiento

En la detección, el método AdaBoost utiliza una ventana que se mueve por toda la imagen comprobando si hay un rostro en esa posición y esta se va agrandando hasta el tamaño de la imagen y comenzando por el tamaño para el cual fue entrenado ( $20 \times 20$ ), esto resuelve el problema de la detección de rostros de diferentes tamaños pero no el de la rotación de estos. La rotación es muy importante ya que la gran mayoría de imágenes existentes hoy en día son tomadas por no profesionales y por esto muchas de las imágenes contienen rostros rotados. Este problema se puede resolver rotando la imagen varias veces

aplicando el detector en cada rotación, lo cual es computacionalmente costoso, o entrenado el sistema directamente para que detecte los rostros rotados quedando la colección de entrenamiento como se muestra en la Figura 2.4.



Figura 2.4: Imágenes rotadas

El clasificador fue entrenado con una muestra de 8000 imágenes positivas con diferentes ángulos de inclinación y 3000 imágenes negativas. Esto permite que el clasificador detecte un rostro rotado pero no en que posición se encuentra, esto trae consigo un problema a la hora de identificar el rostro ya que para esto debe estar normalizado en cuanto a orientación, tamaño y color, como muestra la Figura 2.5, y la normalización en cuanto a orientación es imposible de realizar si no se conoce la posición en la que se encuentra el rostro.



Figura 2.5: Normalización de una imagen

Para resolver el problema de la orientación de la imagen lo que se hace es dividir la colección por posiciones y entrenar el sistema varias veces y a la hora de detectar el rostro se le debe aplicar el detector con cada uno de los entrenamientos para de esta forma poder detectar el rostro rotado y conocer su posición esto tiene la desventaja de que en este caso se demoraría más en detectar.

Existen algunos aspectos importantes a valorar en cuanto a la colección para que el entrenamiento sea mas eficiente estos son:

- La colección que se utilizó detecta rostros hasta un mínimo de  $20 \times 20$  esto da al traste a la hora de identificar ya que para esto los rostros de este tamaño no son de utilidad, por lo que se debe valorar si es factible para MOCIC hacer un detector para la identificación y otro para obtener elementos como por ejemplo la cantidad de rostros en una imagen como se muestra en la Figura 2.6.
- Valorar si es factible para MOCIC detectar rostros con una inclinación mayor que 90 grados, ya que las gran parte de las imágenes con rostros rotados no exceden de este ángulo como se muestra en la Figura 2.6.



Figura 2.6: Varios Rostros

### 2.1.3. Pruebas

Se realizaron dos pruebas, una con imágenes de la vida real (ver Anexo A) y otra con imágenes tomadas para la identificación (ver Anexo B), con una muestra de 100 imágenes, a continuación se presentan los resultados obtenidos donde:

Falso Positivo (FP): Son las áreas detectadas como rostros que no lo son.

Falso Negativo (FN): Son las áreas que son rostros y no fueron detectadas.

Cant. Ros: Es la cantidad de rostros.

El porciento de efectividad es calculado de la siguiente manera:

$$P = 100 - \frac{(FP + FN) * 100}{Cant.Ros}$$

FP	FN	Cant. Ros.
182	19	440

Cuadro 2.3: Resultados con imágenes de la vida real

Esta prueba arrojó un 54,3 % de efectividad, este porciento es influenciado mayormente por la cantidad de FP. Como se evidencia en la Figura 2.7 esta efectividad podrá ser mejorada considerablemente mediante la búsqueda de áreas potenciales para lo cual se necesita un proceso de segmentación por el color de la piel que sea factible.





Figura 2.7: Detección de las áreas potenciales

FP	FN	Cant. Ros.
3	0	100

Cuadro 2.4: Resultados con imágenes tomadas para la identificación

Esta prueba arrojó un 97% de efectividad resultado el cual es satisfactorio puesto que estas imágenes fueron tomadas con el fin de la identificación lo cual hace que su detección sea más precisa.

Luego de estas pruebas se evidencia que para las imágenes de la vida real la efectividad no es buena por lo que se segmentaron las mismas en busca de áreas potenciales, y se realizó otra prueba la cual eliminó varios falsos positivos, con este resultado se realiza una tercera prueba (Anexo C) descartando los rostros que ocupan menos del 5% de la imagen la misma arrojó un 80% de efectividad el cual aumenta considerablemente, a continuación se muestran los resultados.

Totales			
FP	FN	Cant. Ros.	Por ciento
35	59	463	80

## 2.2. Propuesta de métodos para la identificación de rostros

En el Capítulo 1 se estudiaron dos métodos para el reconocimiento de rostros estos son: PCA y Distancias Invariantes los cuales han sido probados a escala internacional y poseen una excelente efectividad por lo cual se hace muy difícil su elección. Para la propuesta del método se tomó en cuenta el método de detección que se propuso ya que para el método de reconocimiento Distancias Invariantes se necesita utilizar la detección de ojos, las cejas y la nariz esto implicaría hacer otro detector para este método mientras que para el PCA esto no sería necesario por lo que se seleccionó como propuesta.

Como se estudió en la Sección 1.4.1 para identificar el rostro debe estar normalizado. Este proceso de normalización debe de ser independiente al de detección y reconocimiento por tanto se debe de ejecutar pasándole los rostros detectados y su inclinación para normalizarlo en cuanto a orientación y también en cuanto a tamaño y color como se muestra la siguiente figura.



Figura 2.8: Proceso de Normalización

Si el proceso de Normalización perteneciera al de detección y se quisiera solo detectar el rostro es se haría innecesariamente y si perteneciera al proceso de reconocimiento entonces en un caso dado de querer identificar rostros ya normalizados se haría innecesariamente también en cambio si se tienen separados le daría la posibilidad a MOCIC de ejecutar las dos situaciones anteriormente mencionadas independientemente de la Normalización.

Se hace importante analizar la diferencia que existe entre el reconocimiento de imágenes tomadas para este fin como se muestra en la Figura 2.9 y las imágenes de la vida real como se muestra en la Figura 2.10, ya que la efectividad del reconocimiento de imágenes de la vida real sería mas baja puesto que estas imágenes como son tomadas generalmente por personal no profesional pueden tener baja iluminación, oclusiones y otros aspectos lo cual dificultarían el reconocimiento mientras que las imágenes que son

tomada para este fin son generalmente con nitidez, normalizadas por lo que se hace mas efectivo. La mayoría de las imágenes que necesitaría reconocer MOCIC son imágenes de la vida real por lo que hay que tener en cuenta que su efectividad sería baja.

Sería de gran importancia analizar que las imágenes tomadas para la identificación son enteramente rostros y además no poseen fondos complejos, por lo que se debe valorar el ejecutar la segmentación ya que en estos casos las áreas potenciales no serían notables.



Figura 2.9: Imagenes tomadas para identificación



Figura 2.10: Imágenes de la vida real

El número de usuarios registrados y el número de imágenes por cada usuario también constituye un factor que influye en los resultados de la verificación de identidad. A un mayor número de usuarios registrados, existe una mayor probabilidad que la red neuronal cometa más errores en la verificación, en tanto que a un mayor número de imágenes por usuarios (imágenes lo más diferente posible entre sí) se mejoran las condiciones para que la red realice una mejor tarea de verificación.

### 2.2.1. Entrenamiento del identificador

Para el entrenamiento del método propuesto se necesitan imágenes de rostros ya normalizadas y de un tamaño de  $100 \times 100$  píxeles y además una cantidad de al menos 20 imágenes por cada persona, cada imagen representa una matriz  $M_{ij}$  donde  $i, j \in \mathbb{R}[1, 100]$ , esa matriz es convertida en un vector  $V_k$ , este vector tiene una gran dimensión(10000) por lo que se reducirá mediante el algoritmo EIGENFACE, después de aplicado este método se obtiene la matriz reducida y los eigenvectores que serían la cantidad

de imágenes por cada persona luego se entrena la red neuronal.

Actualmente no existe una arquitectura de red neuronal que sea adecuada para una tarea específica como es el caso de la verificación de rostros, considerando diversas condiciones de iluminación, rotación, apariencia, expresiones faciales, entre otras. Por tal razón, la selección de una u otra arquitectura no deja de ser un problema de prueba y error, donde se invierte gran cantidad de tiempo probando diferentes arquitecturas y formas de entrenamiento.

En una RNA de  $n$  entrada y  $m$  salidas algunos autores expresan que se debe de comenzar con una capa oculta  $P$  donde  $P = \frac{n}{2} + \frac{m}{2}$  como muestra la Figura 2.11, en cambio otros autores expresan de que debe de comenzar con una capa oculta  $P = \text{MAX}(n, m) + k$  donde  $k \in \mathbb{R}[1, 5]$  como muestra la Figura 2.12.

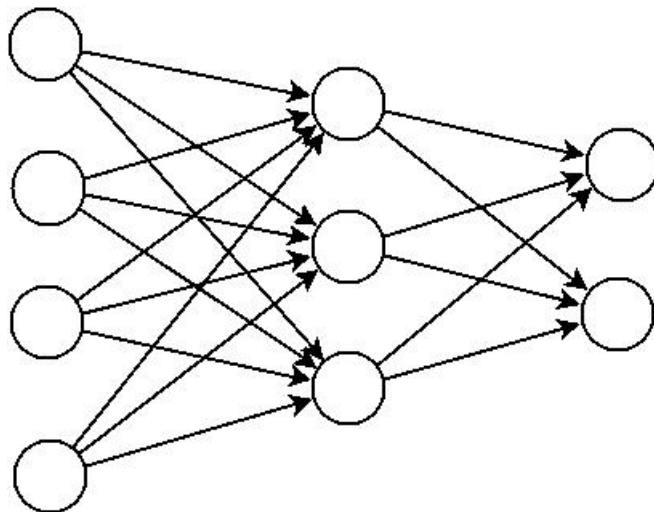


Figura 2.11: Red Neuronal

Una vez que se cuenta con el conjunto de vectores de características que representan al conjunto de imágenes de entrenamiento se crea una red neuronal de tipo Radial Basis Function y se le entrena con dichos vectores. Para este tipo de red se debe definir previo a su entrenamiento el ancho más adecuado de la gaussiana para lograr una buena generalización a partir del conjunto de vectores proporcionados.

La arquitectura que se propone para la red neuronal quedaría de la siguiente manera:

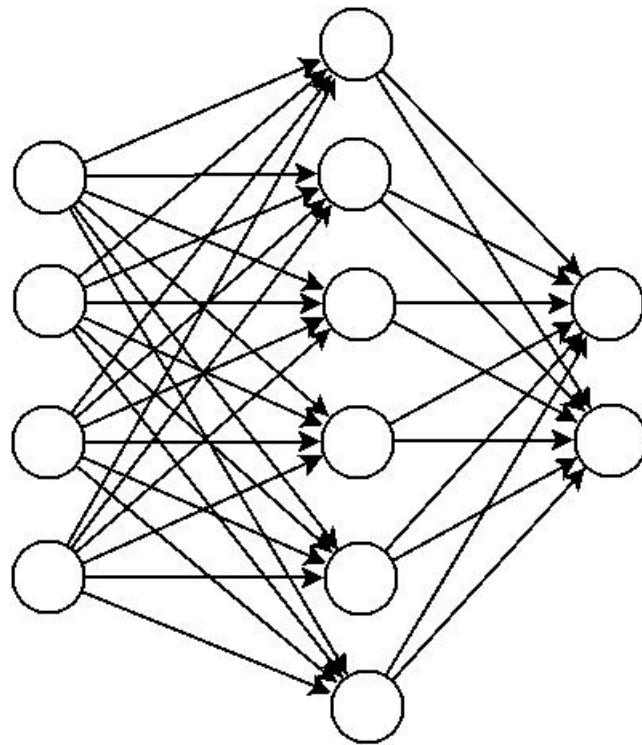


Figura 2.12: Red Neuronal

$n$  = la cantidad de eigenvectors

$m$  = la cantidad de personas a identificar +1

entonces:

$$P = \text{MAX}(n, m) + 5$$

### 2.3. Integración con MOCIC

Aunque el presente trabajo de diploma no es de implementación se definió en su alcance realizar un prototipo funcional del detector de rostros para ser integrado a MOCIC por lo cual se debe de analizar su arquitectura.

### 2.3.1. Arquitectura

El módulo central de MOCIC es el Controlador el cuál se encarga de unir los módulos restantes mediante el envío de mensajes. El Controlador toma los sitios del depósito, construye una petición y la envía a los clasificadores, estos según la URL especificada en el mensaje toman del depósito lo necesario para procesar la petición y una vez procesada estos envían la respuesta al controlador. En la Figura 2.13 se muestra el caso puntual del detector de rostros. La estructura de los mensajes de petición y respuesta serán explicados a continuación.

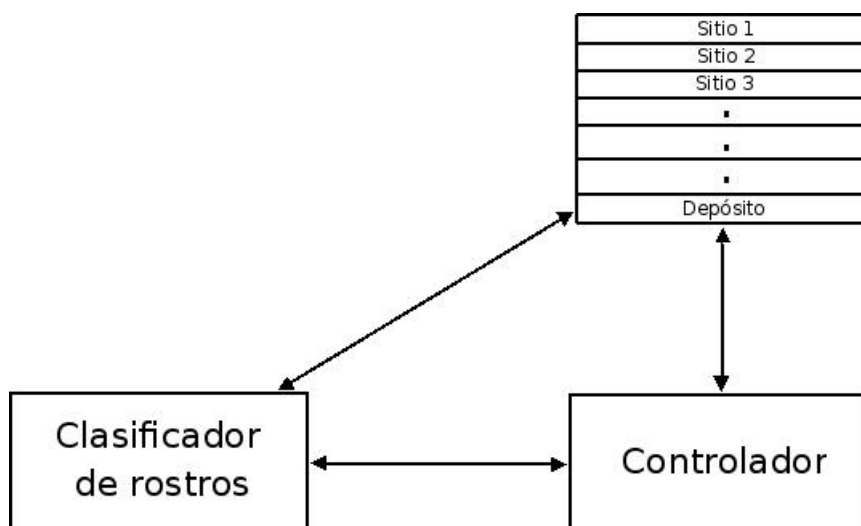


Figura 2.13: Arquitectura de MOCIC

La comunicación se hará a través de socket.

- Solicitud
- Respuesta
- Control

**Estructura del Mensaje:**

<Sello de tiempo><Nombre Filtro><Mensaje Parametrizado>

Sello de tiempo: Indica el día y la hora que fue generada la respuesta o solicitud (AA/MM/DD-hh:mm:ss)

**Nombre del Filtro:**

DROS: Detector de Rostros. **Parámetros:**

- i ( Indica localización del directorio de entrada ) El módulo controlador especifica la localización de la URL a procesar, esta localización se refiere al servidor de almacenamiento (depósito).
- p (Indica el ID del proceso) Es un número único generado en la primera creación del proceso.
- u (Indica la URL) Se especifica la dirección URL.
- c ( Indica mensaje de control ) Es aplicable a todos los Filtros + Decisor. Se utilizarán los mensajes:
  - ABORT: Abortar solo el proceso referenciado por su ID. Se emite cuando: TTL (proceso) expira o cuando el decisor da la respuesta del proceso.
  - RESET: Reiniciar el filtro completo. Esto ocurrirá cuando el número de ABORT exceda un valor predeterminado.
- R (Indica respuesta del filtro)

La respuesta del detector de rostros es como sigue:

- R xx xx xx xx xx xx xx xx. .... xx

El valor de xx representa la cantidad de rostros encontrados en la imagen.

Valores válidos para xx: [00–90]. Cuando xx es 99 indica que no se pudo clasificar esta imagen.

Ejemplo: -R 03 08 05 55 44 00 00 99

Las imágenes estarán ordenadas por nombre y cada xx corresponde a una imagen.



### **2.3.2. Implementación de un prototipo funcional**

Para la implementación del prototipo funcional se utilizó el diseño que propone el módulo de imágenes (MOCIC-IMG, Figura 2.14) que está basado en tres patrones de diseño los cuales son:

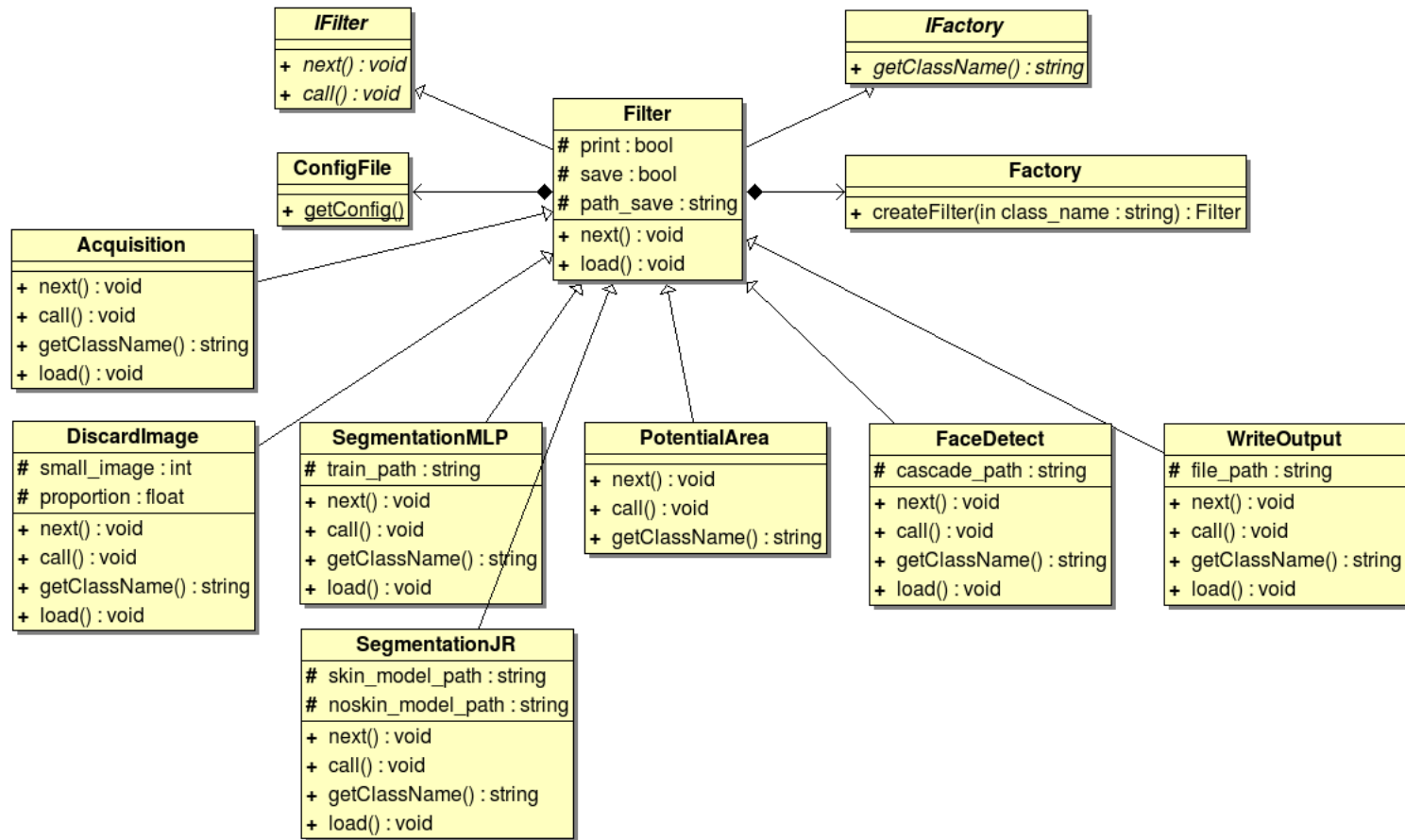


Figura 2.14: Diagrama de clases

**Singleton** Este patrón se utiliza para crear una única instancia de un objeto en toda la aplicación.

**Filter** Permite el flujo de un objeto por un proceso, modificándose este en cada paso.

**Factory** Este patrón permite crear una instancia de una clase específica mediante una configuración determinada.

## Conclusiones

Después de un análisis profundo de los métodos estudiados para la detección e identificación de rostros, se hicieron las propuestas de estos y se explicaron las razones para su elección. Se implementó un prototipo funcional al cual se le hicieron pruebas de integración mostrando un correcto funcionamiento. En las pruebas con imágenes de la vida real se evidenció la necesidad de reducir las áreas de búsqueda mediante la segmentación no siendo así para la prueba imágenes tomadas con el fin de la identificación ya que estas mostraron excelentes resultados.

# Conclusiones

---

Con la culminación del presente trabajo de diploma se cumple con el objetivo trazado en el mismo, mediante la propuesta de los métodos para la detección e identificación. Se cumplió además con los siguientes aspectos:

- Se seleccionó el algoritmo de categorización mas adecuado para la detección e identificación de rostros
- Se analizaron los métodos para la detección e identificación de rostros.
- Se validó la efectividad de los métodos seleccionados mediante un prototipo funcional al cual se le hicieron pruebas para ello.

# Recomendaciones

---

Para mejorar el proceso de detección de rostros se recomienda:

- Hacer entrenamientos con muestras de rostros rotados lateralmente para lograr una detección de perfil.
- La búsqueda de un segmentador en cuanto al color de la piel mas eficiente que el actual para lograr un mejor resultado en la extracción de las áreas potenciales.

Para mejorar el proceso de reconocimiento de rostros se recomienda:

- Implementar un prototipo funcional con el método propuesto en este trabajo para el reconocimiento de rostros.
- Hacer entrenamientos con muestras de rostros con un tamaño de  $100 \times 100$  píxeles y rotadas  $+ - 45$  grados para mejorar la efectividad y rapidez.

# Referencias

---

- [1] Final report. *ISIS*, 2006.
- [2] Detección automática del color de la piel en imágenes bidimensionales basado en el análisis de reguiones. Disponible en: <http://www.escet.urjc.es/~jjpantrigo/PFCs/MemoriaPielFeb05.pdf>.
- [3] Statitical color models with application to skin detection. *Cambridge Research Laboratory*, 1998.
- [4] Cv reference manual [en línea]. Marzo 2009. Disponible en: <http://opencv.willowgarage.com/wiki/CvReference#ObjectDetection> [visitada 1 abril de 2009].
- [5] Rapid object detection using a boosted cascade of simple features. *Presence*, 2001.
- [6] Cecilia A., Germán C., Mauricio D., y Matías M. Detección de ojos [en línea]. 2005. Disponible en: <http://iie.fing.edu.uy/investigacion/grupos/gti/timag/trabajos/2005/caras/Features/deteye.htm> [visitada 2 abril de 2009].
- [7] *Imagen Processing: Principles and applications*. John Wiley & Sons, 1frt edition, 2005.
- [8] Redes neuronales: Funcion de base radial.

# Bibliografía

---

- Luhong Liang Rainer Lienhart and Alexander Kuranov. A detector tree of boosted classifiers for real-time object detection and tracking. *Microcomputer Research Labs, Intel Corporation Santa Clara, CA, 95052.*
- Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. *Intel Labs, Intel Corporation, Santa Clara, CA 95052, USA.*
- Edgar Acuña. Clasificación usando el metodo k-nn.
- Jesús S. Aguilar-Ruiz Francisco J. Ferrer Gualberto Asencio, José C. Riquelme. Extendiendo la semántica de los datos en aprendizaje supervisado. *Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla.*
- Gerardo Colmenares. Redes neuronales: Funcion de base radial radial basis function (rbf).
- Universidad Tecnológica de Pereira. Principales tipos de redes neuronales.
- William Ardila Urueña Marta Lucía Guevara, Julian David Echeverry. Detección de rostros en imágenes digitales usando clasificadores en cascada. *Scientia et Technica Año XIV, No 38, Junio de 2008. Universidad Tecnológica de Pereira. ISSN 0122-1701.*
- Dr. Yan Meng Ryan White, James Ashenurst. Face detection and tracking.
- Daniel Schulz. Detección de ojos mediante adaboost.
- Ming-Hsuan Yang. Face detection. *University of California, Merced, CA 95344.*

- Tom Mitchell. Neural network and face images. *Carnegie Mellon University*.
- Jesse S. Jin John G. Allen, Richard Y. D. Xu. Object tracking using camshift algorithm and multiple quantized feature spaces. *School of Information Technologies, University of Sydney, Madsen Building F09, University of Sydney, NSW, 2006*.
- Francesc Tarrés Antonio Rama. Un nuevo método para la detección de caras basado en integrales difusas. *Dept. Teoria del Senyal i Comunicacions - Universitat Politècnica de Catalunya, Barcelona, Spain*.
- Karla Romero. Reconocimiento de rostros utilizando una red neuronal. *Departamento de Informática y Ciencias de Computación, ESCUELA POLITÉCNICA NACIONAL - Quito XVI Jornadas en Estadística e Informática ESPOL – Guayaquil, 2007*.
- Shumeet Baluja Henry A. Rowley and Takeo Kanade. Neural network-based face detection.
- Padhraic Smyth. Face detection using the viola-jones method. *Department of Computer Science University of California, Irvine*.
- José Luis Alba Castro Iago Landesa Vázquez. Detección de caras y localización de características faciales para reconocimiento biométrico. *Dpto. de Teoría do Sinal e Comunicacions, Universidade de Vigo ETSE Telecomunicación, Campus Universitario, 36310 Vigo*.
- Michael J. Jones Paul Viola. Robust real-time face detection. *Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA, Mitsubishi Electric Research Laboratory, 201 Broadway, Cambridge, MA 02139, USA*.
- Ajoy K. Ray Tinku Acharya. Image processing: Principles and applications.
- João P. Hespanha Peter N. Belhumeur and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 19, NO. 7, JULY 1997*.
- Max Welling. Fisher linear discriminant analysis. *University of Toronto 10 King's College Road Toronto, M5S 3G5 Canada*.
- Marian Stewart Bartlett Kresimir Delac, Mislav Grgic. Recent advances in face recognition. *In-Teh is Croatian branch of I-Tech Education and Publishing KG, Vienna, Austria*.



- 
- M. Beetz M. Wimmer, B. Radig. Adaptive skin color classifier for face outline models. *Informatik IX, Technische Universität München, Germany Boltzmannstr. 3, 87548 Garching, Germany.*
  - Jianbo Shi Dmitry Golomidov. Human detection in video. 2008.
  - Skin colour detection.
  - B. Radig M. Wimmer. Adaptive skin color classifier. *Informatik IX, Technische Universität München, Germany Boltzmannstr. 3, 87548 Garching, Germany.*
  - Ming-Hsuan Yang. Recent advances in face detection. *Honda Research Institute Mountain View, California, USA.*
  - Jasper Laagland Andrès Koetsier. Report on the extension of idesk.
  - Pedro Núñez Jörg Rett Antonio Bandera Francisco Sandoval Luis Molina, Juan P. Bandera. Visual perception system for a social robot(visor). *Grupo de Ingeniería de Sistemas Integrados, Dpto. Tecnología Electrónica, Universidad de Málaga, Campus de Teatinos 29071 Málaga (Spain).*
  - Alexander Hartmann Rainer Lienhart. Classifying images on the web automatically. *Intel Labs, Intel Corporation 2200 Mission College Boulevard, Santa Clara, California 95052-8119.*
  - Alan Roberts Adrian Ford. Colour space conversions.
  - The NSTC Subcommittee on Biometrics. Face recognition.
  - Joaquín Ataz López. Guia casi completa de BiBTeX.
  - Tobias Oetiker. An acronym environment for LaTeX.
  - Irene Hyna y Elisabeth Schlegl Tobias Oetiker, Hubert Partl. The not so short introduction to LaTeX.
  - Raúl Mata Botana. Tablas en LaTeX.
  - George Gratzer. Math into LaTeX.

---

## Anexo A

# Prueba 1

---

Imagen	FP	FN	Cant. Rost.	Imagen	FP	FN	Cant. Rost.
image1	3	0	2	image51	0	1	15
image2	1	2	2	image52	1	1	17
image3	3	2	2	image53	0	0	18
image4	3	0	2	image54	4	0	11
image5	1	0	2	image55	1	0	3
image6	2	0	2	image56	0	0	2
image7	2	0	2	image57	11	1	6
image8	1	0	1	image58	9	0	7
image9	5	0	3	image59	1	0	2
image10	2	0	2	image60	2	1	3
image11	4	0	3	image61	6	0	2
image12	3	1	7	image62	2	0	1
image13	2	0	3	image63	3	0	1
image14	4	0	2	image64	3	0	0
image15	2	0	5	image65	3	1	1
image16	1	0	3	image66	7	0	2

Continúa en la próxima página

image17	5	1	1	image67	2	0	0
image18	5	0	2	image68	1	0	2
image19	1	0	2	image69	3	0	2
image20	0	0	2	image70	3	0	2
image21	0	0	1	image71	4	0	2
image22	0	0	1	image72	3	0	3
image23	0	0	2	image73	4	1	5
image24	1	0	7	image74	1	0	4
image25	1	0	7	image75	1	0	2
image26	1	0	9	image76	3	0	3
image27	1	0	7	image77	3	0	6
image28	0	0	4	image78	1	1	4
image29	1	0	3	image79	1	1	2
image30	0	0	18	image80	0	0	3
image31	0	0	16	image81	1	0	7
image32	1	0	16	image82	2	0	2
image33	0	0	6	image83	1	0	2
image34	2	1	10	image84	1	0	2
image35	1	0	9	image85	3	1	4
image36	0	0	9	image86	2	0	2
image37	1	0	7	image87	1	0	1
image38	1	0	14	image88	2	0	3
image39	1	0	8	image89	3	0	4
image40	1	0	7	image90	2	0	1
image41	1	0	3	image91	0	1	3
image42	0	0	3	image92	2	0	4
image43	0	0	3	image93	3	1	2

Continúa en la próxima página

image44	0	0	3	image94	3	0	6
image45	0	0	6	image95	2	1	5
image46	0	0	6	image96	1	0	4
image47	0	1	22	image97	0	0	2
image48	0	0	11	image98	1	0	1
image49	2	0	5	image99	2	0	1
image50	1	0	2	image100	0	0	1

## Totales

FP	FN	Cant. Ros.
182	19	440

---

## Anexo B

# Prueba 2

---

Imagen	FP	FN	Imagen	FP	FN
image1	0	0	image51	0	0
image2	0	0	image52	0	0
image3	0	0	image53	0	0
image4	0	0	image54	0	0
image5	0	0	image55	0	0
image6	0	0	image56	0	0
image7	0	0	image57	0	0
image8	0	0	image58	0	0
image9	0	0	image59	0	0
image10	0	0	image60	0	0
image11	0	0	image61	0	0
image12	1	0	image62	0	0
image13	0	0	image63	0	0
image14	0	0	image64	1	0
image15	0	0	image65	0	0
image16	0	0	image66	0	0
Continúa en la próxima página					

image17	0	0	image67	0	0
image18	0	0	image68	0	0
image19	0	0	image69	0	0
image20	0	0	image70	0	0
image21	0	0	image71	0	0
image22	0	0	image72	0	0
image23	0	0	image73	0	0
image24	0	0	image74	0	0
image25	0	0	image75	0	0
image26	0	0	image76	0	0
image27	0	0	image77	0	0
image28	0	0	image78	0	0
image29	0	0	image79	0	0
image30	0	0	image80	0	0
image31	0	0	image81	0	0
image32	0	0	image82	0	0
image33	0	0	image83	0	0
image34	0	0	image84	0	0
image35	0	0	image85	0	0
image36	0	0	image86	0	0
image37	0	0	image87	0	0
image38	1	0	image88	0	0
image39	0	0	image89	0	0
image40	0	0	image90	0	0
image41	0	0	image91	0	0
image42	0	0	image92	0	0
image43	0	0	image93	0	0
Continúa en la próxima página					

image44	0	0	image94	0	0
image45	0	0	image95	0	0
image46	0	0	image96	0	0
image47	0	0	image97	0	0
image48	0	0	image98	0	0
image49	0	0	image99	0	0
image50	0	0	image100	0	0

## Totales

FP	FN	Cant. Ros.
3	0	100

---

## Anexo C

# Prueba 3

---

Imagen	FP	FN	Cant. Ros	Imagen	FP	FN	Cant. Ros
image1	0	0	2	image51	1	0	15
image2	0	2	2	image52	0	0	17
image3	0	0	2	image53	0	0	18
image4	1	0	2	image54	0	2	11
image5	0	0	2	image55	1	0	3
image6	1	0	2	image56	0	0	2
image7	0	0	2	image57	2	0	6
image8	0	0	1	image58	1	0	7
image9	0	0	3	image59	1	0	2
image10	1	0	2	image60	1	1	3
image11	1	0	3	image61	1	0	2
image12	1	1	7	image62	0	0	1
image13	1	0	3	image63	0	0	1
image14	0	0	2	image64	0	0	0
image15	1	5	5	image65	0	2	1
image16	1	0	3	image66	1	0	2
Continúa en la próxima página							



image17	2	1	1	image67	0	0	0
image18	4	0	2	image68	0	0	2
image19	1	0	2	image69	1	0	2
image20	0	0	2	image70	0	0	2
image21	0	0	1	image71	0	0	2
image22	0	1	1	image72	0	0	3
image23	0	0	2	image73	0	0	5
image24	1	0	7	image74	0	0	4
image25	1	0	7	image75	0	0	2
image26	1	0	9	image76	0	0	3
image27	1	0	7	image77	0	0	6
image28	0	0	4	image78	1	1	4
image29	0	0	3	image79	0	0	2
image30	0	9	18	image80	0	0	3
image31	0	7	16	image81	0	0	7
image32	0	6	16	image82	1	0	2
image33	0	3	6	image83	0	0	2
image34	0	1	10	image84	1	0	2
image35	0	1	9	image85	0	1	4
image36	0	0	9	image86	0	0	2
image37	0	0	7	image87	0	0	1
image38	0	0	14	image88	1	0	3
image39	0	0	8	image89	1	0	4
image40	0	0	7	image90	1	0	1
image41	0	0	3	image91	0	1	3
image42	1	0	3	image92	2	1	4
image43	0	0	3	image93	2	1	2

Continúa en la próxima página

image44	0	0	3	image94	0	0	6
image45	0	0	6	image95	2	1	5
image46	0	0	6	image96	0	0	4
image47	0	0	22	image97	0	0	2
image48	0	0	11	image98	1	0	1
image49	1	0	5	image99	1	0	1
image50	1	0	2	image100	0	0	1

## Totales

FP	FN	Cant. Ros.
35	59	463

---

## Anexo D

# Fichero de configuración

---

```
<Config>
  <Start>
    <type>CROS</type>
    <next>Acquisition</next>
  </Start>

  <Filter>
    <type>Acquisition</type>
    <print>>true</print>
    <path>ruta /imagenes/</path>
    <call>GetImg</call>
    <next>NULL</next>
  </Filter>

  <Filter>
    <type>GetImg</type>
    <save>>false</save>
    <path_save>copia/</path_save>
    <next>FaceDetect</next>
```

```
</ Filter>
```

```
< Filter>
```

```
  < type>DiscardImage</ type>  
  < small_image>20</ small_image>  
  < proportion>0.1</ proportion>  
  < next>NULL</ next>
```

```
</ Filter>
```

```
< Filter>
```

```
  < type>SegmentJR</ type>  
  < save>>false</ save>  
  < path_save>data / img / mask /</ path_save>  
  < skin>skin . dat</ skin>  
  < no_skin>no_skin . dat</ no_skin>  
  < segment>GRAY</ segment>  
  < channel>1</ channel>  
  < next>NULL</ next>
```

```
</ Filter>
```

```
< Filter>
```

```
  < type>SegmentMLP</ type>  
  < save>>false</ save>  
  < path_save>data / img / mask /</ path_save>  
  < path_train>train . net</ path_train>  
  < segment>GRAY</ segment>  
  < channel>1</ channel>  
  < next>NULL</ next>
```

```
</ Filter>
```

```
< Filter>
```

```
<type>FaceDetect</type>
<save>true</save>
<path_save>detected</path_save>
<print>true</print>
<draw_rectangle>true</draw_rectangle>
<cascade>cascade.xml</cascade>
<next>NULL</next>
</Filter>
</Config>
```

# Glosario de términos

---

**UCI** Universidad de las Ciencias Informáticas

**FILPACON** Filtrado de Paquetes por Contenido:

Es un sistema de filtrado que pretende ser flexible y fiable para regular los contenidos nocivos que provienen de Internet.

**IA** Inteligencia Artificial:

Rama de la informática que desarrolla procesos que imitan a la inteligencia de los seres vivos.

**MOCIC** Motor de Categorización Inteligente de Contenido

**OSRI** Oficina de Seguridad para las Redes Informáticas

**OSRI** Base de Datos de URLs Categorizadas

**HTML** *HyperText Mark Language:*

Lenguaje de Marcas de Hipertexto. Es el lenguaje de marcado predominante para la construcción de páginas HTML.

**HTML** Red:

La traducción literal de esta palabra inglesa es tela de araña, pero en términos informáticos significa mucho más que eso.

**k-NN** *k Nearest Neighbour:*

Proviene del idioma inglés y es traducido como los *k* vecinos más cercanos.

<b>RNA</b>	Redes Neuronales Artificiales
<b>URL</b>	<i>Uniform Resource Locator:</i> Se refiere al texto que identifica a una página HTML.
<b>GNU</b>	<i>GNU is Not Unix:</i> Sistema operativo cuyo nombre es un acrónimo recursivo que significa “GNU No es Unix”
<b>Linux</b>	Es el núcleo (kernel) del sistema operativo libre GNU
<b>CDT</b>	<i>C/C++ Development Tools:</i> Es un plugging para Eclipse el cual permite la programación en el lenguaje C/C++
<b>MLP</b>	<i>Multilayer Perceptron:</i> Perceptron multicapa. Topología de Red Neuronal definida por Rosenblatt en el año 1957.
<b>IDE</b>	<i>Integrate Development Enviroment:</i> Entorno de desarrollo integrado. Herramienta que se usa para facilitar el desarrollo de software.
<b>RGB</b>	<i>Red, Green and Blue:</i> Rojo, verde y Azul. Es un modelo de color en el cual es posible representar un color mediante la mezcla de estos tres colores primarios.
<b>HSV</b>	<i>Hue, Saturation and value:</i> Modelo de color basado en tres parámetros tono, saturación y valor.
<b>CAT</b>	Categorizador Automático de Texto: Es un categorizador de texto desarrollado por FILPACON.
<b>CAID</b>	Categorizador Automático de Imagenes Desnudas: Es un categorizador de imagenes desnudas desarrollado por FILPACON.
<b>MD</b>	Módulo decisor: Es un categorizador de imagenes desnudas desarrollado por FILPACON.
<b>MC</b>	Módulo controlador: Es un categorizador de imagenes desnudas desarrollado por FILPACON.

<b>PCA</b>	<i>Principal Component Analysis:</i> Por sus siglas en ingles...
<b>LDA</b>	<i>linear discriminant analysis:</i> Por sus siglas en ingles...
<b>PC</b>	<i>Personal Computer:</i> Por sus siglas en ingles...
<b>FLDF</b>	<i>Fisher Linear Discriminant Function:</i> Por sus siglas en ingles...
<b>EBGM</b>	<i>Elastic Bunch Graph Matching:</i> Por sus siglas en ingles...
<b>FP</b>	Falso Positivo
<b>FN</b>	Falso Negativo
<b>RBF</b>	<i>Radial Basis Function:</i> Por sus siglas en ingles...