



Universidad Nacional del Litoral

Facultad de Ingeniería y Ciencias Hídricas

Proyecto Final de Carrera - Ingeniería en Informática

VERIFICACIÓN BIOMÉTRICA AUTOMÁTICA DE IDENTIDAD MEDIANTE RECONOCIMIENTO FACIAL

Autores: Long, Sebastián
Müller, Omar V.

Director: Bioing. Martínez, Cesar

Santa Fe - 2007

Prefacio

El reconocimiento facial humano consiste en una habilidad natural y robusta, capaz de soportar variaciones en la expresión, el transcurso del tiempo, la oclusión parcial de rasgos importantes, la pose, el punto de vista y la diversidad del fondo. Sin embargo, construir un sistema computacional que reproduzca esta tarea, con similares características y eficiencia, constituye un desafío que aún no ha sido resuelto, y es un área de investigación de gran interés actual.

Este proyecto pretende realizar un aporte a un sistema de registro de personal actualmente en utilización en la Facultad de Ingeniería de la Universidad Nacional de Entre Ríos, desarrollando un módulo de verificación de identidad mediante reconocimiento facial automático. Para ello, se utilizan técnicas *ad-hoc* en los pasos de preprocesamiento de las imágenes capturadas y en la extracción de características útiles en la etapa posterior de reconocimiento, donde se aplicarán algoritmos existentes y se probará la eficacia del sistema propuesto en un ambiente real.

Este documento se encuentra organizado como se indica a continuación.

El Capítulo 1 introduce al lector en los sistemas biométricos, haciendo hincapié en los de reconocimiento facial automático. Se explica la evolución de las técnicas, el estado del arte y se hace una descripción general de la motivación para el desarrollo del proyecto, los objetivos y la propuesta.

El Capítulo 2 expone los conceptos del procesamiento digital de imágenes utilizados en el proyecto, como así también se introduce al Análisis de Componentes Principales (PCA) y a las técnicas de clasificación de patrones experimentadas.

Los Capítulos 3 y 4 detallan el diseño y desarrollo del módulo. Específicamente, el primero de ellos describe el análisis y diseño del sistema propuesto en base a los conceptos del Lenguaje de Modelación Unificado (UML), mientras que el segundo describe con detalles al reconocedor completo –desde la captura de la imagen hasta la verificación de la identidad de la persona bajo estudio– justificando y ejemplificando gráficamente en cada paso.

En el Capítulo 5 se describe el diseño de los experimentos, su aplicación y los resultados obtenidos en las pruebas realizadas, mediante las cuales se comprueba el desempeño general del algoritmo y el análisis del comportamiento del mismo en situaciones particulares.

Finalmente, estos experimentos se evalúan y discuten, dando lugar a las conclusiones y delineado de tareas futuras expuestas en el Capítulo 6.

Índice general

| | |
|--|-------------|
| Prefacio | III |
| Índice de figuras | VIII |
| Índice de tablas | X |
| 1. Introducción | 1 |
| 1.1. Motivación | 1 |
| 1.2. Objetivos | 2 |
| 1.2.1. Objetivos generales | 2 |
| 1.2.2. Objetivos específicos | 2 |
| 1.3. Antecedentes en reconocimiento de caras | 3 |
| 1.3.1. El problema de la localización | 4 |
| 1.3.2. Algoritmos de reconocimiento en la actualidad | 5 |
| 1.4. Descripción general de la propuesta | 8 |
| 2. Fundamentos teóricos | 11 |
| 2.1. Procesamiento digital de imágenes | 11 |
| 2.1.1. Introducción | 11 |
| 2.1.2. Filtro de mediana | 12 |
| 2.1.3. Corrección Gamma | 13 |

| | | |
|-----------|---|-----------|
| 2.1.4. | Modelos de Color | 13 |
| 2.1.5. | Segmentación en el espacio vector RGB | 16 |
| 2.1.6. | Transformaciones morfológicas | 16 |
| 2.2. | Análisis de componentes principales | 19 |
| 2.2.1. | Introducción | 19 |
| 2.2.2. | Justificación matemática | 19 |
| 2.2.3. | Componentes principales | 21 |
| 2.3. | Clasificación de patrones | 23 |
| 2.3.1. | Métodos de la Teoría de la decisión | 24 |
| 3. | Diseño del módulo de reconocimiento facial | 25 |
| 3.1. | Introducción | 25 |
| 3.2. | Análisis de requerimientos | 27 |
| 3.2.1. | Requerimientos funcionales | 27 |
| 3.2.2. | Requerimientos no funcionales | 28 |
| 3.3. | Casos de Uso | 28 |
| 3.4. | Diagrama de Clase | 33 |
| 3.5. | Diagrama de Secuencia | 36 |
| 4. | Implementación del módulo de reconocimiento facial | 40 |
| 4.1. | Introducción | 40 |
| 4.2. | Preproceso | 42 |
| 4.2.1. | Restauración | 42 |
| 4.2.2. | Normalización de la iluminación | 42 |
| 4.3. | Extracción de características | 43 |
| 4.3.1. | Localización | 44 |
| 4.3.2. | Normalización de la geometría | 46 |
| 4.3.3. | Proyección Eigenfaces | 47 |
| 4.4. | Clasificación | 50 |
| 4.4.1. | Identificación | 50 |
| 4.4.2. | Verificación de la identidad | 51 |
| 5. | Marco experimental | 52 |
| 5.1. | Descripción de los datos | 52 |

| | |
|--|-----------|
| 5.2. Diseño de experimentos | 53 |
| 5.3. Implementación y resultados | 53 |
| 5.3.1. Experimento 1: Normalización de la iluminación | 53 |
| 5.3.2. Experimento 2: Detección de ojos | 55 |
| 5.3.3. Experimento 3: Tasa de reconocimiento | 55 |
| 5.3.4. Experimento 4: Sensibilidad al ruido y oclusión | 57 |
| 5.4. Análisis de resultados y discusión | 58 |
| 6. Conclusiones | 63 |
| 6.1. Conclusiones finales | 63 |
| 6.2. Desarrollos futuros | 64 |
| Bibliografía | 65 |
| A. Corpus | 69 |
| A.1. BADAFFI | 69 |
| B. Librería de Procesamiento de Imágenes Digitales | 72 |
| B.1. CImg | 72 |

Índice de figuras

| | |
|--|----|
| 1.1. Sistema genérico de reconocimiento facial | 8 |
| 2.1. Gráfico de la ecuación $s = cr^\gamma$ | 14 |
| 2.2. Modelo de color RGB | 15 |
| 2.3. Formas de representar el subespacio segmentado | 17 |
| 3.1. Diagrama de bloque del sistema existente | 26 |
| 3.2. Diagrama de bloque del sistema Propuesto | 26 |
| 3.3. Diagrama de casos de uso de alto nivel | 29 |
| 3.4. Diagrama de casos de uso detallado | 30 |
| 3.5. Diagrama de clase | 35 |
| 3.6. Diagrama de secuencia del módulo reconocedor | 37 |
| 3.7. Diagrama de secuencia del módulo de entrenamiento | 38 |
| 4.1. Sistema genérico de reconocimiento facial | 40 |
| 4.2. Preproceso | 42 |
| 4.3. Ejemplo de preproceso | 43 |
| 4.4. Extracción de características | 44 |
| 4.5. Ejemplo de localización | 45 |
| 4.6. Ejemplo de detección de ojos | 46 |
| 4.7. Normalización | 47 |

| | |
|--|----|
| 4.8. Eigenfaces e imagen media | 49 |
| 4.9. Clasificación | 50 |
| 5.1. Evaluación del método de normalización | 54 |
| 5.2. Evaluación del proceso de detección de ojos | 56 |
| 5.3. SCREE Test | 57 |
| 5.4. Tasa de reconocimiento para k -vecinos | 60 |
| 5.5. Evaluación de la sensibilidad al ruido y oclusiones parciales . . | 60 |
| 5.6. Representación de tres clases sobre las dos primeras eigenfaces. | 62 |

Índice de tablas

| | |
|---|----|
| 5.1. Valores correspondientes a la Figura 5.1 | 55 |
| 5.2. Rendimiento del algoritmo de reconocimiento | 58 |
| 5.3. Tasas de reconocimiento y rechazo para k -vecinos más cercanos | 59 |
| 5.4. Tasa de reconocimiento para clase mas cercana | 59 |

Introducción

1.1 Motivación

En el ámbito de las tecnologías de seguridad, desde el acceso seguro a computadoras, redes, hasta sistemas de fichaje de personal y control de ingreso a salas de acceso restringido, uno de los problemas fundamentales es la necesidad de autenticar de forma segura la identidad de las personas. Hasta el momento es común encontrar sistemas de autenticación basados en números de identificación personal (DNI, PIN), llaves electrónicas (tarjetas magnéticas), etc., con los consiguientes inconvenientes de seguridad que ello presenta, al no contar ninguno de estos métodos con una verificación eficaz de que el número ingresado (en el caso de los PIN) o el portador de la llave (en el segundo caso), sean efectivamente de la persona que se registra en el sistema [8].

En la actualidad están apareciendo técnicas basadas en características biológicas, intrínsecas a la propia persona, y por tanto difícilmente transferibles o sustraíbles. Estas técnicas reciben el nombre de *biométricas*, concepto que proviene de las palabras bio (vida) y metría (medida).

El proceso general de autenticación sigue unos pasos comunes a todos los métodos biométricos: captura de los datos del usuario a validar, extracción de ciertas características de la muestra, comparación de tales características con las registradas en una base de datos, y validación final del usuario. Por tanto, desde un enfoque informático, la biometría puede definirse como la aplicación

de técnicas matemáticas y estadísticas sobre rasgos físicos o de conducta de un individuo, a fin de “verificar” identidades o “identificar” individuos.

Aunque existen métodos biométricos muy confiables, como huellas digitales, retina o iris, el reconocimiento facial se destaca por ser un sistema discreto (no necesita de la colaboración del usuario), aceptado por los usuarios, de bajo costo, fácil de implementar y con un extenso rango de aplicaciones posibles [25]; lo que ha llevado a un continuo crecimiento, especialmente durante los últimos años, entre las técnicas de análisis y procesamiento de imágenes.

La Facultad de Ingeniería de la Universidad Nacional de Entre Ríos (FI-UNER) cuenta con un sistema actualmente en funcionamiento de registro de entrada y salida del personal, mediante el tipeo del DNI. A fin de proponer una mejora respecto a la comprobación de la identidad de la persona que se registra, este proyecto pretende realizar el diseño y la implementación de un módulo de software para autenticar la identidad mediante técnicas de reconocimiento automático de imágenes faciales. Este módulo podría servir como prototipo para el estudio de su integración definitiva por parte de los administradores del sistema en la FI-UNER.

1.2 Objetivos

1.2.1 Objetivos generales

- Adquirir nuevos conocimientos acerca de aplicaciones específicas innovadoras de técnicas de procesamiento digital de imágenes.
- Aplicar los conocimientos adquiridos en el transcurso de la carrera a un proyecto que realice un aporte ingenieril a la comunidad en general.

1.2.2 Objetivos específicos

- Profundizar en el estudio y desarrollo de técnicas de adquisición y pre-procesamiento de imágenes faciales.
- Conformar una base de datos con imágenes capturadas en el sistema de registro de entrada/salida de la FI-UNER.

- Diseñar e implementar un localizador y normalizador de caras a partir de técnicas de procesamiento digital de imágenes.
- Diseñar e implementar un clasificador para reconocimiento facial, que incorpore técnicas del estado del arte.
- Diseñar e implementar el módulo de verificación biométrica de identidad mediante reconocimiento facial, para el sistema de registro de entrada/salida del personal de la FI-UNER.

1.3 Antecedentes en reconocimiento de caras

A partir de 1950 se llevaron a cabo gran cantidad de investigaciones por psicofísicos, neurólogos e ingenieros sobre varios aspectos del reconocimiento facial humano y su automatización [37]. Los psicofísicos y neurólogos en particular han estudiado cuestiones tales como si la percepción facial es un proceso dedicado y si es realizado holísticamente o mediante análisis de características locales [9].

Por otro lado los ingenieros formularon el problema como reconocimiento de objetos 3D a partir de imágenes 2D. Como resultado, hasta la primera mitad de los años '70, se adoptaron técnicas típicas de clasificación de patrones que usaban medidas de características como atributos, logrando así sistemas semiautomáticos. Bledsoe en 1966 [4] realizó un clasificador híbrido humano-computadora en base a distancias y radios calculados a partir de marcas puntuales ingresadas a mano. Goldstein, Harmon y Lesk en 1971 [10] trabajaron sobre marcas subjetivas (longitud de la oreja, grosor de labios, etc.) lo que hacía difícil la automatización.

Durante los años '80, no hubo grandes avances en el área, pero a partir de 1990 los estudios tuvieron un importante crecimiento atribuible al interés comercial, la disponibilidad de tecnología y el incremento de las aplicaciones relacionadas a la seguridad.

En los últimos 20 años, el objetivo de las investigaciones fue lograr un sistema totalmente automático abordando problemas como la localización del rostro en una imagen dada y la extracción de características faciales (ojos, boca, etc.). Mientras tanto, se lograron avances significativos en el diseño de algoritmos de reconocimiento, tales como Eigenfaces o Fisherfaces.

1.3.1 El problema de la localización

Muchas de las técnicas de reconocimiento facial asumen la disponibilidad de imágenes ideales (buena luz, fondo uniforme y pocas variaciones en la pose). Sin embargo, en aplicaciones sobre escenarios reales la cara se encuentra en un fondo complejo y con variantes en la pose, necesitando de procesamientos previos sobre la imagen que localicen y extraigan la región facial del fondo [14].

Esta tarea es de las más críticas en los sistemas de reconocimiento facial. Hasta mediados de los años '90, los trabajos en detección facial se basaron en imágenes con una cara simple sobre un fondo controlado. Los métodos incluían la utilización de plantillas de la cara, plantillas deformables basadas en características, color de la piel y redes neuronales. En los últimos años, se buscó trabajar con escenas complejas (fondos desordenados, con condiciones de iluminación impredecibles) surgiendo gran cantidad de métodos, que se pueden clasificar en dos grandes áreas que serán explicadas a continuación, según utilicen información global o local de la cara.

Localización mediante representación global de la cara

Entre las técnicas más destacadas está la codificación de reglas que describen las características de la cara y sus relaciones (distancias relativas y posiciones). Presenta la dificultad de codificar nuestro conocimiento de caras efectivamente y el problema de generar muchos falsos positivos en fondos complejos [17].

Otras técnicas globales son las basadas en apariencia y el uso de plantillas que tratan la localización como un problema de clasificación de patrones, donde la cara es el patrón. Cuando se utilizan las plantillas se busca un patrón definido manualmente en la imagen de entrada. Es decir, las categorías *cara* y *no cara* son explícitamente definidas por un experto. En los métodos basados en apariencia, las plantillas se aprenden de las imágenes de entrenamiento. En general, estos métodos se basan en el análisis estadístico y aprendizaje automático para encontrar las características relevantes de las categorías *cara* y *no cara* [29].

El defecto de los algoritmos basados en plantilla es la dificultad de generalización del algoritmo para diferentes situaciones de imagen tales como iluminación, pose y oclusión parcial. Para superar este problema, el experto debe definir un conjunto de plantillas o el sistema debe ser entrenado con un conjunto de datos representativo de todos los casos. Ambos métodos hacen

una búsqueda exhaustiva de patrones, para cada posición y escala, extraen la región y la clasifican con los métodos clásicos.

También se encuentran las técnicas basadas en la distribución del color. Dado que el color de piel se mantiene en una estrecha banda en el espacio color y que la mayor información se concentra en los componentes de crominancia y no en luminancia, esa componente es descartada en muchos casos. Se usan diferentes modelos de color para segmentar: RGB, HSV, YCrCb, YIQ, CIE, XYZ, etc [5].

Localización mediante representación local de la cara

En estos métodos se busca detectar características faciales como ojos, cejas, boca, contorno de la cara, etc. y se combinan con procedimientos que identifican las relaciones y verifican la existencia de la cara. Se basan en la idea de que el hombre es capaz de localizar caras en diferentes poses y condiciones de iluminación, lo que supone que existen propiedades o características invariantes a estas variables. Por otro lado, la detección de características locales se utiliza para la normalización de las imágenes, es decir, obtener imágenes con tamaño y rotación semejantes, en métodos de combinación holística uniforme, como Eigenfaces o Fisherfaces, o bien como parámetros característicos al momento de la clasificación.

Para detectar las características locales se pueden usar técnicas similares a las presentadas anteriormente como las basadas en apariencia o combinación de plantillas, utilizando múltiples plantillas para los distintos componentes faciales con hipótesis para sus posibles apariciones [30].

1.3.2 Algoritmos de reconocimiento en la actualidad

Durante los últimos 10 años, muchas investigaciones se centraron en reconocimiento facial en videos y el reconocimiento facial 3D, sin embargo estos temas no son profundizados en el presente informe, porque exceden el marco de trabajo de este proyecto (más bibliografía en [18, 6, 32, 3, 21]). Por otro lado, el reconocimiento facial en imágenes 2D fijas, se ha convertido en una de las aplicaciones más investigadas en reconocimiento de patrones y análisis y procesamiento de imágenes, logrando avances significativos en el estado del arte. Este auge es evidente con el surgimiento de conferencias como la AVB-PA (del inglés *Audio and Video-based Biometric Person Authentication*) a partir de 1997 y AFGR (del inglés *Automatic Face and Gesture Recognition*)

desde 1995, ambas organizadas por la IAPR (del inglés *International Association for Pattern Recognition*) de la IEEE. Desde 2006 estas conferencias fueron integradas en la ICB (del inglés *International Conference on Biometrics*) de la misma organización. También surgieron evaluaciones sistemáticas empíricas como FERET (del inglés *Face Recognition Technology*), FRVT (del inglés *Face Recognition Vendor Tests*) 2000, etc. y la disponibilidad de muchos sistemas comerciales [37].

Entre los algoritmos de reconocimiento más eficientes en la actualidad, se hace la siguiente clasificación:

Métodos de concordancia holística

Estos métodos utilizan el rostro completo para el reconocimiento.

- *Eigenfaces* [34, 33, 23], es un método que utiliza la técnica de análisis de componentes principales (PCA, del inglés *Principal Component Analysis*) tomando como datos de entrada los valores de píxeles de imágenes faciales. A partir de un conjunto de imágenes faciales de entrenamiento tomadas como un punto en el espacio P dimensional, mediante PCA se buscan nuevas variables en un subespacio K dimensional ($K \ll P$) cuyos vectores de la base correspondan a la dirección de la máxima varianza en el espacio original, de forma tal que cada imagen concentre su información en unos pocos valores.
- *Análisis de componentes independientes* (ICA, del inglés *Independent Component Analysis*) [2] es una generalización de PCA. La base encontrada en este último método tiene en cuenta solo relaciones entre píxeles de a pares, sin embargo es de suponer que en el reconocimiento facial existen relaciones entre píxeles de alto orden. Basado en esto ICA considera las relaciones de segundo orden como las de orden superior en los datos de entrada e intenta encontrar la base en la cual los datos resultan estadísticamente independientes.
- El método adaptativo basado en eigenespacio *Evolutionary Pursuit* [22] tiene por objetivo la codificación y clasificación de imágenes. Implementa estrategias características de los algoritmos genéticos en la búsqueda del espacio de soluciones que determine el mejor conjunto de ejes de proyección, midiendo al mismo tiempo la precisión de clasificación y la habilidad de generalización del sistema. Como la dimensión del espacio solución es muy grande, se resuelve mediante un algoritmo genético llamado *Evolutionary Pursuit*.

Métodos de concordancia basados en características (estructural)

Usan características locales como ojos, nariz y boca, sus localizaciones y estadísticas locales (geometría y/o apariencia) como entrada a un clasificador estructural. Las técnicas de clasificación basadas en este paradigma, por ejemplo la correspondencia de grafos, han logrado buenos resultados, resultando menos sensibles a variaciones de iluminación y puntos de vista y a la inexactitud en localización. Sin embargo, las técnicas de extracción de características requeridas para este tipo de métodos no son lo suficientemente exactas.

- *Correspondencia de grafos elásticos* (EBGM, del inglés *Elastic Bunch Graph Matching*) [36]. Todas las caras humanas tienen una estructura topológica similar. Las caras son representadas como grafos, con nodos posicionados en puntos importantes (ojos, nariz) y aristas etiquetadas con vectores distancia 2D. Cada nodo contiene un conjunto de coeficientes de ondas complejas de Gabor a diferentes escalas y orientaciones (fase, amplitud), llamados *jets*. Un grafo etiquetado es un conjunto de nodos conectados por aristas, los nodos son etiquetados con *jets*, las aristas con distancias. El reconocimiento está basado en la comparación directa de los grafos.
- Los *modelos ocultos de Markov* (HMM, del inglés *Hidden Markov Models*) [24] constituyen una clase de modelo estadístico en el cual se asume que el sistema que genera las señales de salida (secuencias temporales, imágenes u otras) puede ser aproximado mediante un autómata finito probabilístico de parámetros desconocidos, y el objetivo es estimar estos parámetros a partir de las señales que modelan. Un HMM consta de un conjunto finito de estados cada uno de los cuales modela segmentos característicos de señal, una matriz de probabilidad de transición entre estados, un conjunto de funciones de densidad de probabilidad asociadas con cada estado, y una distribución de probabilidad de estado inicial.

Métodos híbridos

Basados en la idea de que la percepción humana combina características locales y globales, estos métodos buscan reconocer la cara usando ambos atributos en un único sistema. Entre ellos eigenfaces extendido a eigenfeatures [27] y otros [13, 35, 20].

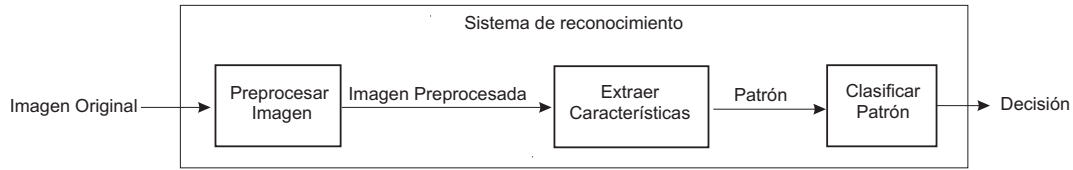


Figura 1.1: Sistema genérico de reconocimiento facial

1.4 Descripción general de la propuesta

La FI-UNER tiene en funcionamiento un sistema de registro de entrada/salida del personal donde los usuarios tipean el DNI, el mismo se busca en la base de datos y si es encontrado se registra el horario y se muestra en pantalla. Para ello se cuenta con una PC dedicada, en el ingreso al edificio. A fin de mejorar el sistema se propone desarrollar un módulo que realice una verificación biométrica de la identidad ingresada a partir de la captura de una imagen del rostro del usuario.

A tal efecto, en primer término se efectuó la instalación del dispositivo de captura (cámara web) conectado a la PC del sistema, que adquiere una muestra al ingresar el DNI. Con estas muestras, se hace una selección y un etiquetado, para conformar la base de datos BADAFFI (BASE de DATOS Faciales de la FI-UNER). Cabe destacar que la cámara se encuentra detrás de una vidrio en un pasillo amplio a pocos metros de la puerta de ingreso, con luz natural y sin ningún tipo de iluminación artificial, necesaria en este tipo de sistemas. Esto hace que las imágenes adquiridas sean de baja calidad, con problemas de luminosidad, con reflejo en muchos casos, sobre un fondo complejo y con variedad de poses y distancias de los usuarios.

Una vez que se dispone del material necesario se propone un algoritmo basado en técnicas avanzadas del procesamiento digital de imágenes y métodos del estado del arte que se adapten a las necesidades (tiempo real, bajo costo computacional) y características (calidad de imágenes) del sistema. Este sistema se compone de dos etapas, una de entrenamiento del cual se obtienen parámetros útiles para el reconocimiento y otra de prueba que valida el módulo de reconocimiento. Estas etapas tienen en común gran parte del algoritmo, por lo que se detalla el mismo, diferenciando cuando sea necesario a qué etapa corresponde el paso en cuestión, y omitiendo detalles en donde entrenamiento y prueba coincidan en la tarea.

El algoritmo propuesto consta de tres bloques fundamentales (ver Figura

1.1):

1. Preproceso: Esta etapa se encarga de restaurar y normalizar las imágenes de entrada, a fin de lograr una buena localización y un buen reconocimiento.
 - Restauración: Aplica un filtro de mediana para reducir el ruido generado por el dispositivo de captura.
 - Normalización de la iluminación: Aplica corrección gamma para aclarar las imágenes con baja luminosidad.
2. Extracción de características: En este paso se realizan todas las tareas necesarias para localizar el rostro y luego parametrizar la imagen facial obteniendo un vector característico representativo del rostro.
 - Localización: La detección del rostro se basa en una segmentación de la piel detectando posibles regiones cara, y luego decidiendo mediante combinación de plantillas, haciendo ampliaciones y rotaciones de la misma que permitan un rendimiento efectivo ante variantes en la pose y la distancia. Finalmente se recorta un rectángulo que incluya la cara y se transforma a escala de grises.
 - Normalización de la geometría: Aquí se efectúa la detección de los ojos mediante un algoritmo basado en las operaciones morfológicas de dilatación y erosión. Una vez detectados, se normaliza en base a la distancia entre los ojos y se ajusta la inclinación de la cara mediante una rotación que tiene en cuenta el ángulo del segmento entre ojos respecto al piso.
 - Proyección Eigenfaces: Eigenfaces es uno de los métodos más exitosos en reconocimiento facial, basado en el concepto de la teoría de la información, que descompone las imágenes en un pequeño conjunto de imágenes de características llamadas "eigenfaces". Estas representan las imágenes faciales como combinación lineal de un reducido número de imágenes de características, las cuales son las componentes principales del conjunto de imágenes de entrenamiento. El reconocimiento se realiza proyectando una imagen de entrada en el subespacio de las eigenfaces ("eigenespacio") y luego se clasifica comparando su posición en el eigenespacio con las posiciones de individuos conocidos. En el entrenamiento se calculan los vectores de la base a partir de los eigenvectores y eigenvalores de la matriz de covarianza. Con estos vectores tanto en el entrenamiento como en la prueba se proyecta al nuevo espacio obteniendo

el vector de características que representa la imagen de entrada. Finalmente en el caso del entrenamiento se etiqueta el vector característico con el DNI del usuario.

3. Clasificación: Sólo el proceso de prueba realiza esta etapa, en donde se hace la comparación entre la posición de la cara de prueba en el eigenspacio con las posiciones de los individuos conocidos.
 - Identificación: Se obtiene la etiqueta que corresponde al vector almacenado que minimiza la distancia con el vector de prueba, usando la distancia euclídea.
 - Verificación de la identidad: Se verifica que la identidad reconocida corresponda a la ingresada por el usuario.

Finalizado el desarrollo del algoritmo se ajustan los parámetros del sistema, verificando su desempeño y se elaboran las conclusiones.

Fundamentos teóricos

2.1 Procesamiento digital de imágenes

2.1.1 Introducción

Una imagen puede ser definida como una función bidimensional $f(x, y)$ representando intensidad de luz, donde x e y son las coordenadas espaciales y el valor de f en cualquier punto (x, y) se denomina intensidad o nivel de gris. Si x , y y f son valores discretos, llamamos a la imagen *imagen digital*. Por lo tanto, una imagen digital puede ser representada como una matriz de $M \times N$ elementos llamados *píxeles*.

$$f(x, y) = \begin{bmatrix} f(1, 1) & f(1, 2) & \cdots & f(1, N) \\ f(2, 1) & f(2, 2) & \cdots & f(2, N) \\ \vdots & \vdots & \ddots & \vdots \\ f(M, 1) & f(M, 2) & \cdots & f(M, N) \end{bmatrix}$$

El procesamiento digital de imágenes (PDI) es el conjunto de métodos y algoritmos para la manipulación de imágenes digitales por medio de una computadora. Posee un amplio campo de aplicaciones al permitir trabajar con todo el espectro electromagnético [11], mientras la visión humana solo percibe un rango del mismo. No hay un acuerdo general sobre los límites

entre PDI y áreas relacionadas tales como análisis de imagen y visión computacional. Una manera útil para diferenciar estas áreas, es considerar que los procesos de PDI efectúan transformaciones de imagen a imagen, es decir, realizan operaciones primitivas tales como reducción del ruido, realce del contraste y refinamiento de la imagen. El análisis de imágenes transforma una imagen en atributos (por ejemplo los bordes, texturas, y mediciones en los objetos presentes), lo que implica tareas como la segmentación (dividir una imagen en regiones u objetos) y descripción de un objeto en particular. Finalmente, el reconocimiento (y su campo relacionado, la visión computacional), se ocupa de la interpretación de un conjunto de objetos, tomando los atributos y asignándole etiquetas a los objetos reconocidos, realizando las funciones cognitivas normalmente asociadas con la vista.

2.1.2 Filtro de mediana

En el área de captura y procesamiento digital de imágenes es común encontrarse con imágenes que posean ruido, debido a defectos de los dispositivos de adquisición o los medios de transmisión. Estos ruidos pueden ser de diferente tipo: gaussiano, impulsivo, gamma, exponencial, uniforme, etc., distinguiéndose por la manera en que afectan la imagen. Existen métodos denominados *técnicas de restauración* que logran un adecuado desempeño para determinados tipos de ruidos y fallan en otros. Entre ellas se encuentran las técnicas de procesamiento local, ampliamente utilizadas ya que trabajan directamente sobre la imagen (dominio espacial), realizando cálculos sencillos en un entorno cercano de un píxel cualquiera. Dentro de este grupo de técnicas se hallan las máscaras o filtros espaciales.

Los filtros espaciales procesan todos los puntos de la imagen, trabajando sobre vecindades pequeñas de éstos mediante la aplicación de una máscara (también denominada núcleo o ventana). Los valores particulares en cada punto de la máscara (denominados coeficientes), determinan el resultado obtenido del procesamiento.

El proceso para realizar un filtrado consiste simplemente en mover la máscara sobre toda la imagen, y en cada punto (x, y) la respuesta del filtro se calcula usando una operación predefinida. Para el filtrado espacial lineal, la respuesta es la suma de los productos entre los coeficientes del filtro y su correspondiente píxel de la imagen en el área abarcada por la máscara del filtro. Para un filtro no lineal o de orden estadístico, el resultado está basado en el ordenamiento de los píxeles contenidos dentro del área de la máscara, y un posterior reemplazo con un valor resultante de la operatoria que se aplique

a la ventana ordenada.

Un filtro no lineal empleado en este trabajo es el de mediana, que toma un entorno de un píxel de la imagen y ordena los niveles de gris de menor a mayor, obteniendo como salida aquel que sea la mediana estadística de la muestra de valores.

$$f(x, y) = \text{mediana}_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (2.1)$$

donde S_{xy} representa el conjunto de coordenadas (s, t) de la máscara g centrada en (x, y) .

Este filtro obtiene buenos resultados en imágenes con ruido impulsivo (también conocido como *sal y pimienta*, por sus valores extremos negro y blanco), al eliminar el ruido sin provocar un desenfoque o nublado en la imagen.

2.1.3 Corrección Gamma

Una gran variedad de dispositivos usados para la impresión o visualización de imágenes responden a los estímulos de entrada de acuerdo a una ley de potencia, según la siguiente transformación de dominio:

$$s = cr^\gamma \quad (2.2)$$

donde r es el valor del píxel de entrada, s de la salida y c y γ son constantes positivas. El exponente se denomina “factor gamma”. El proceso utilizado para la corrección de estos fenómenos se lo conoce con el nombre de corrección gamma.

Las transformaciones de potencia son útiles para la manipulación de contraste. La Figura 2.1 muestra las transformaciones para distintos valores de γ , donde los valores de r y s están normalizados entre 0 y 1. Las curvas de potencia con valor de $\gamma < 1$ amplifican los valores oscuros y contraen los valores claros de entrada. Para valores de $\gamma > 1$ se produce el efecto contrario.

2.1.4 Modelos de Color

Los modelos de color son estándares de especificación de colores. Estos modelos especifican un sistema de coordenadas de 3 dimensiones y un subespacio en el cual un color queda definido por un punto. En términos de

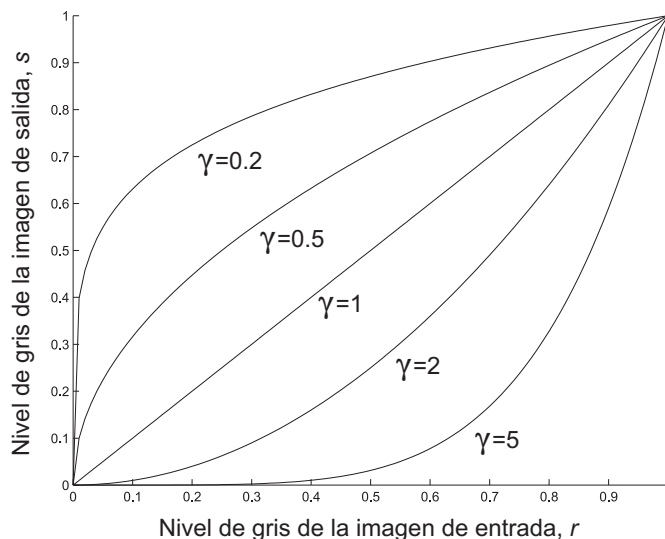


Figura 2.1: Gráfico de la ecuación $s = cr^\gamma$

PDI, los modelos orientados al hardware más usados son el RGB (rojo, verde y azul), CMY (cian, magenta y amarillo), el CMYK (cian, magenta, amarillo y negro), el HSI (tono, saturación y brillo) y el YCbCr (luminancia, crominancia azul, crominancia roja).

Modelo RGB

Cada color es una combinación de los colores primarios R (rojo), G (verde) y B (azul). Este modelo está basado en el sistema de ejes cartesianos y el subespacio es un cubo normalizado de $[0,1]$, en el cual los valores RGB se ubican en los tres vértices, $(1,0,0)$, $(0,1,0)$ y $(0,0,1)$ respectivamente; los otros tres vértices corresponden a los colores secundarios cian, magenta y amarillo, en el origen está ubicado el negro y el punto más lejano del origen corresponde al blanco (ver Figura 2.2). En este modelo los valores de grises (un punto con los tres valores iguales) se extienden sobre el segmento que va del origen $(0,0,0)$ al punto más lejano $(1,1,1)$. Los diferentes colores en este modelo son puntos dentro del cubo, que están definidos por un radio vector.

Las imágenes en el modelo RGB están compuestas por tres matrices, una para cada color primario. El número de bits para representar cada píxel se denomina profundidad.

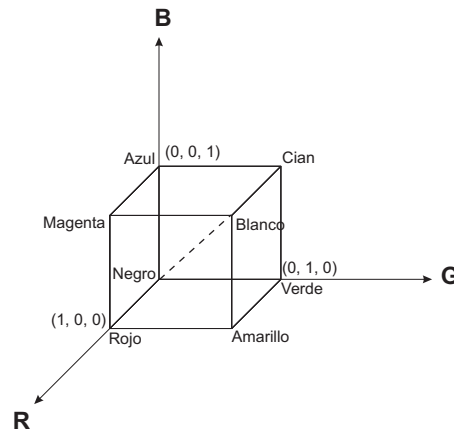


Figura 2.2: Modelo de color RGB

Modelo YCbCr

YCbCr es un modelo de color usado en sistemas de video, donde Y es el componente de luminancia¹ y Cb y Cr son los componentes de crominancia² azul y rojo.

Las ecuaciones para pasar de un sistema RGB a un YCbCr son las siguientes:

$$\begin{aligned}
 Y &= 0,3R + 0,6G + 0,1B \\
 Cb &= \frac{B - Y}{2} + 0,5 \\
 Cr &= \frac{R - Y}{1,6} + 0,5
 \end{aligned}
 \tag{2.3}$$

YCbCr es el equivalente digital del modelo YUV (sistema de TV). El formato YCbCr concentra la mayor parte de la información de la imagen en la luminancia y menos en la crominancia. El resultado es que los elementos de YCbCr están menos correlacionados y pueden ser codificados por separado.

¹Luminancia: intensidad aparente de la luz proveniente o reflejada por un objeto o punto determinado.

²Crominancia: Parte de la señal que contiene la información de color.

2.1.5 Segmentación en el espacio vector RGB

La segmentación es el proceso de partir una imagen en regiones. En el espacio RGB, la segmentación se basa en separar las regiones contenidas en un rango específico de color de una imagen. Dado un conjunto de muestras del rango de color a segmentar, se calcula el promedio de las mismas obteniendo un vector RGB \mathbf{m} . Luego se clasifica cada píxel según pertenezca al rango específico o no. Esta comparación se determina mediante la distancia Euclídea entre un punto cualquiera \mathbf{a} del espacio RGB y el vector promedio \mathbf{m} , si la distancia entre ellos es menor que un umbral D_0 , el píxel es segmentado. La distancia Euclídea entre \mathbf{a} y \mathbf{m} está dada por:

$$\begin{aligned} D(\mathbf{a}, \mathbf{m}) &= \|\mathbf{a} - \mathbf{m}\| \\ &= [(\mathbf{a} - \mathbf{m})^t(\mathbf{a} - \mathbf{m})]^{1/2} \\ &= [(a_r - m_r)^2 + (a_g - m_g)^2 + (a_b - m_b)^2]^{1/2} \end{aligned} \quad (2.4)$$

donde r , g , y b son las componentes de \mathbf{m} y \mathbf{a} . La imagen binaria resultante toma el valor 1 para los puntos que cumplen con la condición $D(\mathbf{a}, \mathbf{m}) \leq D_o$ y 0 para el resto. Existen tres formas de representar el subespacio (ver Figura 2.3), éstas son: una esfera, donde D_o es el radio de la misma, un cuerpo elíptico y una caja. Para este trabajo se utilizó la caja que está centrada en \mathbf{m} , y las dimensiones a lo largo de cada uno de los ejes RGB son escogidos proporcionalmente a la desviación estándar de las muestras a largo de los ejes. Dado un conjunto arbitrario de puntos, se segmenta de acuerdo a si está o no dentro del volumen de la caja, por medio del cálculo de la distancia. Al utilizar las dos primeras formas se requieren operaciones de potencia en el cálculo de la distancia, mas costosas computacionalmente con respecto a la caja.

2.1.6 Transformaciones morfológicas

Las transformaciones morfológicas brindan una manera no lineal de procesar imágenes digitales basándose en la forma de los objetos presentes en la imagen. Dentro de estas transformaciones se encuentran las de dilatación y erosión.

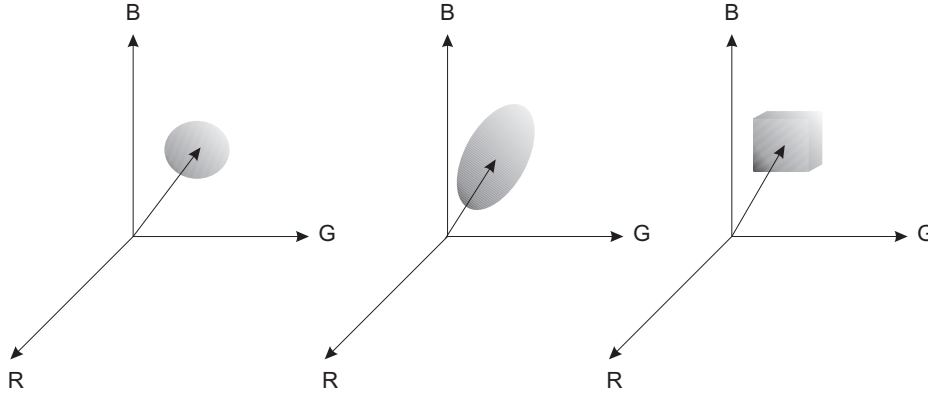


Figura 2.3: Formas de representar el subespacio segmentado

Dilatación

La dilatación de una imagen binaria A y un elemento estructurante B (máscara) en un espacio de enteros Z^2 , se denota $A \oplus B$ y se define como:

$$A \oplus B = \{z \mid [(\hat{B})_z \cap A] \subseteq A\} \quad (2.5)$$

donde \hat{B} es la reflexión de B sobre su origen. Esta reflexión se intercambia por el punto z .

Este operador es similar a la convolución y es comúnmente conocido como “relleno”, “expansión” o “crecimiento”. Puede ser usado para rellenar “huecos” de tamaño igual o menor que el elemento estructurante con la que se opera la dilatación. Sobre cada píxel de la imagen se superpone el origen de B , si el píxel de la imagen en ese punto es 1, cada píxel de A que cae en B es añadido al resultado aplicando el operador OR.

Erosión

La erosión del conjunto A y B en Z^2 se denota $A \ominus B$ y se define como:

$$A \ominus B = \{z \mid B_z \subseteq A\} \quad (2.6)$$

Este operador es lo opuesto a la dilatación, logrando efectos de “encogimiento”, “contracción”, o “reducción”. Puede ser utilizado para eliminar islas menores en tamaño que el elemento estructurante. Sobre cada píxel de A se superpone el origen de B , luego si B está completamente contenido en A , entonces el píxel de salida es puesto a 1.

Extensión a imágenes en escala de grises

La imagen en escala de grises se representa como $f(x, y)$ y el elemento estructurante como $b(x, y)$, donde se asume que estas funciones son discretas. La dilatación de f y b se expresa como $f \oplus b$ y se define según:

$$f \oplus b(s, t) = \text{máx}\{f(s-x, t-y) + b(x, y) \mid (s-x), (t-y) \in D_f; (x, y) \in D_b\} \quad (2.7)$$

donde D_f y D_b son el dominio de f y b respectivamente. Las condiciones de que $(s-x)$ y $(t-y)$ tienen que estar en el dominio de f , y x e y tiene que estar en el dominio de b , expresan que debe existir superposición al menos en un elemento. La ecuación 2.7 es similar a la convolución en 2D, con la operación máximo (denotada por *máx*) reemplazando a la suma de convolución y la suma reemplazando al producto de convolución.

En general, si todos los valores del elemento estructurante son positivos, la imagen de salida tiende a ser más brillante y a expandir las regiones claras. Los detalles oscuros se reducen o eliminan, dependiendo del elemento estructurante.

La erosión en imágenes en escala de grises se escribe como $f \ominus b$ y se define según:

$$f \ominus b(s, t) = \text{mín}\{f(s+x, t+y) - b(x, y) \mid (s+x), (t+y) \in D_f; (x, y) \in D_b\} \quad (2.8)$$

donde D_f y D_b son el dominio de f y b respectivamente. Las condiciones de que $(s+x)$ y $(t+y)$ tienen que estar en el dominio de f , y x y y tiene que estar en el dominio de b , expresa que el elemento estructurante tiene que estar completamente contenido en f . La ecuación 2.8 es similar a la correlación en 2D, con la operación mínimo (denotada por *mín*) reemplazando a la suma de correlación y la resta reemplazando al producto de correlación.

En general, si todos los elementos de b son positivos, la imagen de salida es más oscura y se reducen los detalles brillantes cuya área es menor que el área de b . Esta reducción se visualiza como un gradiente determinado por el valor de gris circundante a los detalles brillantes.

2.2 Análisis de componentes principales

2.2.1 Introducción

El análisis de componentes principales (PCA) es una técnica estadística con aplicaciones en áreas de reconocimiento facial y compresión de imágenes. Técnicamente, es una transformación lineal de los datos a un nuevo sistema de coordenadas tal que la máxima varianza queda proyectada sobre la primera coordenada, la segunda mayor varianza queda proyectada sobre la segunda coordenada, y así sucesivamente. Comúnmente se usa para la reducción de la dimensionalidad de un conjunto de datos reteniendo aquellas características del conjunto que mayor contribución tienen a su varianza, manteniendo las primeras componentes principales de orden más bajo.

2.2.2 Justificación matemática

Estadística

Tanto la varianza como el desvío estandar son medidas unidimensionales. Sin embargo, muchos conjuntos de datos tienen más de una variable, y el objetivo del análisis estadístico es ver la relación entre las variables. Por ejemplo, si se registran en un examen las notas obtenidas por N alumnos en la variable X y la cantidad de horas de estudio en Y , se puede hacer un análisis para ver si las horas de estudio tienen algún efecto sobre la nota. En este caso, el desvío estandar y la varianza solo operan sobre una dimensión, por lo que se pueden calcular sobre cada variable independientemente de la otra. Por otro lado, la covarianza permite operar sobre 2 variables mostrando la relación entre las mismas, y se define como:

$$\text{cov}(X; Y) = \frac{\sum_{i=1}^N (X_i - \bar{X})(Y_i - \bar{Y})}{(N - 1)} \quad (2.9)$$

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

Lo más importante en el análisis de la covarianza no es el valor exacto, sino más bien el signo. Si el valor es positivo, indica que ambas variables

crecen juntas, si es negativo, mientras una variable crece la otra decrece y si es cero las variables son independientes una de otra.

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

$$C = (c_{ij}, c_{ij} = \text{cov}(X_i; X_j)) \quad (2.10)$$

donde C es de $P \times P$.

Para un conjunto de 3 variables por ejemplo, se obtiene:

$$C = \begin{bmatrix} \text{cov}(X_1, X_1) & \text{cov}(X_1, X_2) & \text{cov}(X_1, X_3) \\ \text{cov}(X_2, X_1) & \text{cov}(X_2, X_2) & \text{cov}(X_2, X_3) \\ \text{cov}(X_3, X_1) & \text{cov}(X_3, X_2) & \text{cov}(X_3, X_3) \end{bmatrix}$$

Como sirve, a los valores de la diagonal corresponde la covarianza de una variable consigo misma, o lo que es igual, la varianza de la variable. Además $\text{cov}(a, b) = \text{cov}(b, a)$ por lo que la matriz resultante es simétrica y mantiene los valores mas altos en la diagonal.

Algebra matricial

En muchas aplicaciones es útil encontrar para una transformación lineal $T : V \rightarrow V$ un vector \mathbf{v} en el espacio vectorial V tal que $T\mathbf{v}$ y \mathbf{v} son paralelos. Es decir, se busca un vector \mathbf{v} y un escalar λ tal que

$$T\mathbf{v} = \lambda\mathbf{v} \quad (2.11)$$

Si $\mathbf{v} \neq \mathbf{0}$ y λ satisface 2.11, entonces λ se llama un *eigenvalor* de T y \mathbf{v} se llama un *eigenvector* de T correspondiente al eigenvalor λ . Si V tiene una dimensión finita, entonces T se puede representar por una matriz A_T [12].

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

Para una matriz hermítica³ $A_{N \times N}$ existen N eigenvalores y eigenvectores asociados, y los eigenvectores correspondientes a distintos eigenvalores son

³Una matriz A se dice que es *hermítica* si la traspuesta de la matriz compleja conjugada coincide con la matriz original. Si A es real y simétrica, entonces es hermítica.

linealmente independientes. Esto significa que se pueden expresar los datos en términos de estos eigenvectores perpendiculares.

2.2.3 Componentes principales

La técnica de PCA transforma un conjunto de variables correlacionadas en un conjunto de variables no correlacionadas, y simplifica la transformación encontrando los componentes más cercanos a las variables originales pero ordenadas en forma decreciente de varianza. Esta transformación de hecho es una rotación ortogonal en el espacio de las variables originales. Dada la dificultad de encontrar patrones en conjuntos de datos de grandes dimensiones, y la imposibilidad de representarlos gráficamente, PCA es una herramienta poderosa para el análisis de datos. La otra gran ventaja de PCA es que una vez encontrados los patrones, los datos se pueden comprimir sin grandes pérdidas de información.

Método

A continuación se detallan los pasos del método, según [16].

Datos: Dado un conjunto de datos de P variables con N individuos, estos se colocan en la matriz W , donde x_{ij} representa el valor que toma la variable X_j en el individuo w_i .

$$W = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1P} \\ x_{21} & x_{22} & \cdots & x_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{NP} \end{bmatrix}$$

Variables centradas: Para trabajar con datos estandarizados, se sustrae la media para cada una de las variables $X_j^0 = X_j - \overline{X_j}$. De esta forma, el conjunto de datos W^0 tiene media cero.

Matriz de covarianza: Se calcula la matriz de covarianza C a partir de (2.10), quedando:

$$C = \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1P} \\ c_{21} & c_{22} & \cdots & c_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ c_{P1} & c_{P2} & \cdots & c_{PP} \end{bmatrix}$$

donde $c_{ij} = cov(X_i; X_j)$.

Eigenvectores y eigenvalores: Dado que C es cuadrada, se pueden calcular los eigenvectores \mathbf{v}_i y eigenvalores λ_i con $i = 1, \dots, P$ y se ordenan de acuerdo a los eigenvalores de manera decreciente formando:

$$\Lambda = \begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_P \end{bmatrix}$$

$$V = \begin{bmatrix} \mathbf{v}_{11} & \mathbf{v}_{12} & \dots & \mathbf{v}_{1P} \\ \mathbf{v}_{21} & \mathbf{v}_{22} & \dots & \mathbf{v}_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{v}_{P1} & \mathbf{v}_{P2} & \dots & \mathbf{v}_{PP} \end{bmatrix}$$

donde cada columna de V es un eigenvector.

Los eigenvectores obtenidos son ortonormales, lo que permite a PCA expresar los datos como combinación lineal de los mismos, preservando las distancias en el espacio P dimensional.

Componentes principales: Las nuevas variables rotadas se obtendrán de acuerdo a la relación

$$Y^0 = W^0 V \quad (2.12)$$

siendo Y_i^0 la i -ésima *componente principal*, obtenida como combinación lineal de las variables originales: $Y_i = W \mathbf{v}_i$, que por ser V una matriz ortogonal, inducen una rotación de los ejes coordenados. Esto asegura que las distancias entre individuos se mantenga, tanto si se calculan con las coordenadas X_i como con las Y_i^0 .

Esta rotación se efectúa con el criterio de hacer coincidir el eje Y_1^0 con la dirección de mayor variabilidad, el eje Y_2^0 con la dirección ortogonal a Y_1^0 de mayor variabilidad residual y así sucesivamente.

Reducción de la dimensionalidad: PCA redistribuye la varianza total de los datos originales haciendo que las primeras componentes acumulen la máxima cantidad de varianza posible. Cada componente principal acumula una proporción de la varianza total igual a:

$$\frac{\lambda_i}{\sum \lambda_i} \quad (2.13)$$

Entonces, para las primeras K componentes principales el porcentaje de la

varianza total acumulada por las primeras K -ésimas componentes principales está dada por:

$$\frac{\sum_{i=1}^K \lambda_i}{\sum_{i=1}^n \lambda_i} \times 100 \quad (2.14)$$

Si para un determinado valor de K ($K \leq P$) se reconstruye un porcentaje elevado de la varianza total, se puede reemplazar las primeras P variables originales por las K primeras componentes principales, con poca pérdida de información.

Reconstrucción de los datos

Los datos originales de la matriz W pueden ser aproximados utilizando los K primeros eigenvectores y las componentes principales asociadas, de la forma:

$$\widehat{W}^0 = Y_K (V_K)^t \quad (2.15)$$

en otras palabras las variables originales pueden ser parcialmente reconstruidas como combinación lineal de las K primeras componentes principales. En caso de $K = P$ las variables originales se reconstruyen en su totalidad. Para llegar a W basta con sumar las \overline{X}_i restadas en el segundo paso.

2.3 Clasificación de patrones

Un vector patrón \mathbf{x} se define como un arreglo de características que describen una señal, en nuestro caso una imagen y puede ser representado como un punto en un espacio multidimensional. Una clase, denotada por ω_i con $i = 1, \dots, \Omega$, es un conjunto de patrones con propiedades en común. Las técnicas de clasificación de patrones son aquellas que asignan los patrones a sus respectivas clases de acuerdo a las características que presenta.

Los métodos de clasificación se dividen en: *clasificación estadística* o basados en la teoría de la decisión, donde se utilizan las características estadísticas de los patrones y se asume que los mismos han sido generados por un sistema probabilístico y *clasificación sintáctica* o estructural, la cual se basa en las relaciones estructurales de las características [11]. En este trabajo se utilizará el primer enfoque, que se explica a continuación.

2.3.1 Métodos de la Teoría de la decisión

Sea $\mathbf{x} = (x_1, x_2, \dots, x_n)^t$ un vector patrón n -dimensional y $\omega_1, \omega_2, \dots, \omega_\Omega$ las Ω clases. El problema en la teoría de la decisión es encontrar Ω funciones de decisión $d_1(\mathbf{x}), d_2(\mathbf{x}), \dots, d_\Omega(\mathbf{x})$ tal que, un patrón \mathbf{x} pertenece a la clase ω_i , si

$$d_i(\mathbf{x}) > d_j(\mathbf{x}) \quad j = 1, 2, \dots, \Omega; j \neq i \quad (2.16)$$

Es decir, un patrón desconocido \mathbf{x} pertenece a la i -ésima clase si, calculando todas las funciones de decisión para \mathbf{x} , $d_i(\mathbf{x})$ produce el máximo.

Clasificación por mínima distancia

Dentro de los métodos de la teoría de la decisión se encuentra la clasificación por mínima distancia, donde cada clase se representa por un vector patrón prototipo, luego se calcula la distancia entre el patrón desconocido y cada uno de los prototipos, eligiendo la menor distancia para asignar el patrón a una clase.

Se define el prototipo de cada clase como el vector media de los patrones de la clase:

$$\mathbf{m}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in \omega_j} \mathbf{x}_j \quad j = 1, 2, \dots, \Omega \quad (2.17)$$

donde N_j es el número de patrones de la clase ω_j . La pertenencia de un patrón desconocido \mathbf{x} a una clase se determina a partir del prototipo más cercano. Si se utiliza la distancia Euclídea para determinar la proximidad, dada por:

$$D_j(\mathbf{x}) = \|\mathbf{x} - \mathbf{m}_j\|_2 \quad j = 1, 2, \dots, \Omega \quad (2.18)$$

entonces, seleccionar la mínima distancia es equivalente a evaluar las funciones

$$d_j(\mathbf{x}) = \mathbf{x}^t \mathbf{m}_j - \frac{1}{2} \mathbf{m}_j^t \mathbf{m}_j \quad j = 1, 2, \dots, \Omega \quad (2.19)$$

y asignar \mathbf{x} a la clase ω_i si $d_i(\mathbf{x})$ es el máximo.

Diseño del módulo de reconocimiento facial

3.1 Introducción

Como se señaló en la Sección 1.4, la FI-UNER tiene en funcionamiento un sistema de registro de entrada/salida del personal donde los usuarios tipean el DNI, si el mismo es localizado en la base de datos, se registra el horario y se muestra en pantalla. Para ello se cuenta con una PC ubicada en el ingreso al edificio. En la Figura 3.1 se observa un diagrama de bloques esquematizando el sistema actual.

En este proyecto se pretende mejorar el sistema mediante la implementación de un módulo que realice la verificación biométrica de la identidad ingresada, a partir de la captura de una imagen del rostro del usuario. Para esto se instaló un dispositivo de captura (cámara web) conectado a la PC del sistema, que al ingresar el DNI adquiere una imagen del usuario. En la Figura 3.2 se observa como el módulo de reconocimiento se inserta en el sistema luego de localizar el usuario en la base de datos, arrojando un resultado positivo cuando el DNI del usuario reconocido a partir de la imagen, coincida con el ingresado, y negativo en caso contrario. En este último caso se elimina automáticamente la imagen tomada.

Un estándar para modelar los sistemas son las herramientas que proporciona UML (Unified Modeling Language), que es un lenguaje utilizado para escribir o modelar cualquier sistema de información. UML provee un vocabulario y reglas para utilizarlo que permiten representar grandes sistemas

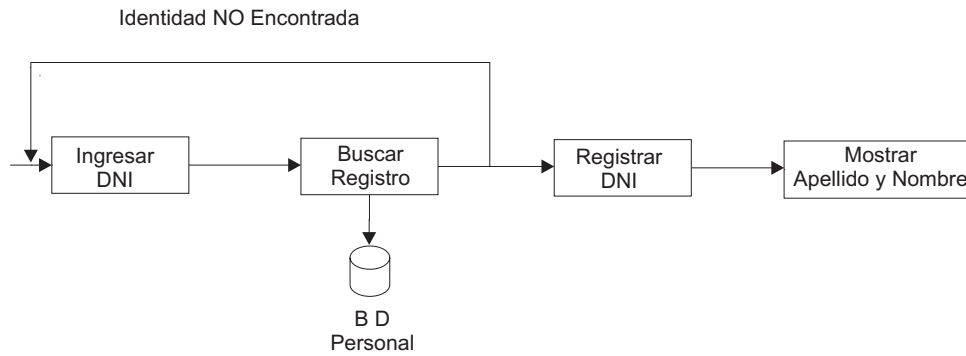


Figura 3.1: Diagrama de bloque del sistema existente

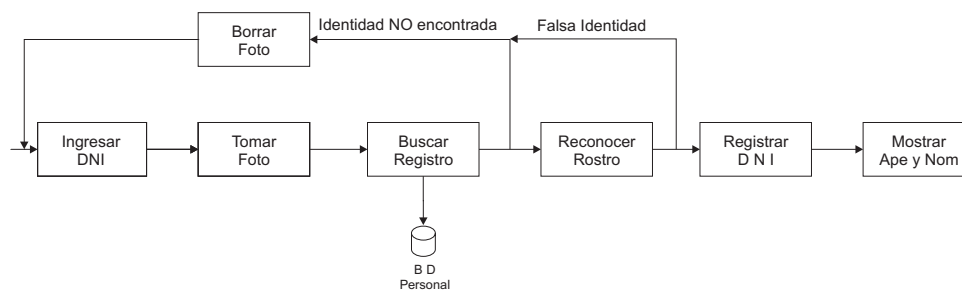


Figura 3.2: Diagrama de bloque del sistema Propuesto

mediante gráficos sencillos o texto obteniendo modelos explícitos. Estos modelos mejoran la comunicación cliente-desarrollador durante el desarrollo, al poder ser interpretados por personas que no participaron en su diseño (e incluso por herramientas) sin ninguna ambigüedad.

En este capítulo, se analiza y documenta las funcionalidades del módulo a partir de UML. Primero se especifican los requerimientos identificados, luego se explica el funcionamiento del módulo mediante los diagramas de casos de uso, clases y secuencia.

3.2 Análisis de requerimientos

Existen muchas definiciones para requerimiento, tales como: “Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo”, “una condición o capacidad que debe estar presente en un sistema para satisfacer un contrato, estándar, especificación u otro documento formal” o finalmente “una representación documentada de una condición o capacidad de los tipos definidos anteriormente”. A partir de estos conceptos, se realiza el análisis de requerimientos en función de las necesidades que tiene que satisfacer el sistema para darle más autenticidad, seguridad y robustez.

Por otro lado, los requerimientos pueden dividirse en funcionales y no funcionales. Los primeros definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema produce sobre las entradas para generar salidas. Los requerimientos no funcionales están relacionados con características que de una u otra forma pueden limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, etc.

A continuación se definen los requerimientos del módulo.

3.2.1 Requerimientos funcionales

- El módulo debe:
 1. Localizar el rostro en la imagen, teniendo en cuenta factores como luz, distancia de la persona a la cámara, fondo complejo, etc, proporcionando una subimagen normalizada de la cara.
 2. Verificar que la imagen normalizada pertenezca al usuario que ingresó el DNI.
 3. Mejorar el rendimiento del reconocedor agregando automáticamente imágenes a la base de datos del módulo.
 4. Añadirse al sistema vigente.

3.2.2 Requerimientos no funcionales

1. Se requiere que el dispositivo de captura se encuentre ubicado correctamente, sin obstáculos. La escena se debe iluminar con luz incidente ubicada junto a la cámara apuntando hacia el rostro del usuario y que este adquiera una pose definida en el momento de tomar la foto.
2. La cámara debe cumplir con características aceptables de velocidad, resolución, etc.
3. El módulo debe ejecutarse en tiempo real, por lo tanto, los cálculos deben estar depurados eficientemente.

3.3 Casos de Uso

Un caso de uso es la descripción de un conjunto de acciones que un sistema ejecuta y que produce un determinado resultado de interés para un usuario particular [7]. El diagrama de caso de uso sirve para capturar el comportamiento deseado del sistema en desarrollo, sin tener que especificar cómo se implementa ese comportamiento. En esta sección se explica la funcionalidad del módulo y las operaciones que realiza el usuario, por medio de los diagramas de caso de uso y sus descripciones textuales.

En la Figura 3.3 se observa el diagrama de caso de uso de alto nivel del módulo reconocedor y en la Figura 3.4 un diagrama más detallado (bajo nivel).

A continuación se explica en detalle cada caso de uso de la Figura 3.3.

| |
|---|
| Caso de uso: Cargar Foto + DNI |
| Actor: Sistema |
| Descripción: Carga la imagen de un directorio y el DNI que se encuentra en el nombre de la misma. |
| Caso normal: |
| 1) El módulo carga la o las imágenes almacenadas de un directorio. 2) El módulo registra el DNI que se encuentra como parte del nombre de la imagen. |

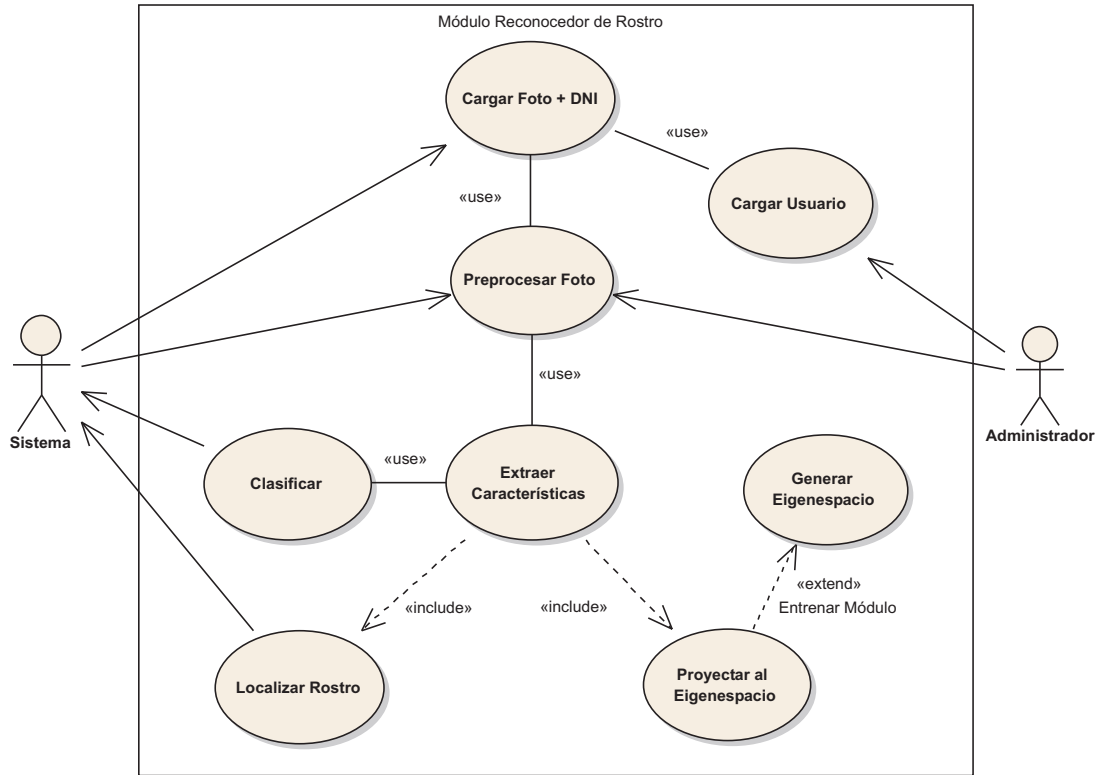


Figura 3.3: Diagrama de casos de uso de alto nivel

| |
|--|
| Caso de uso: Cargar Usuario |
| Actor: Administrador |
| Descripción: Da de alta usuarios o agrega mas imágenes de la persona al módulo. |
| Caso normal: |
| <ol style="list-style-type: none"> 1) El administrador guarda las imágenes con los DNI asociados en un directorio. 2) El módulo carga la imagen con el DNI. 3) El administrador toma una muestra de piel sobre las imágenes cargadas. 4) El módulo genera un archivo con las coordenadas de la muestra de piel y el DNI correspondiente. |

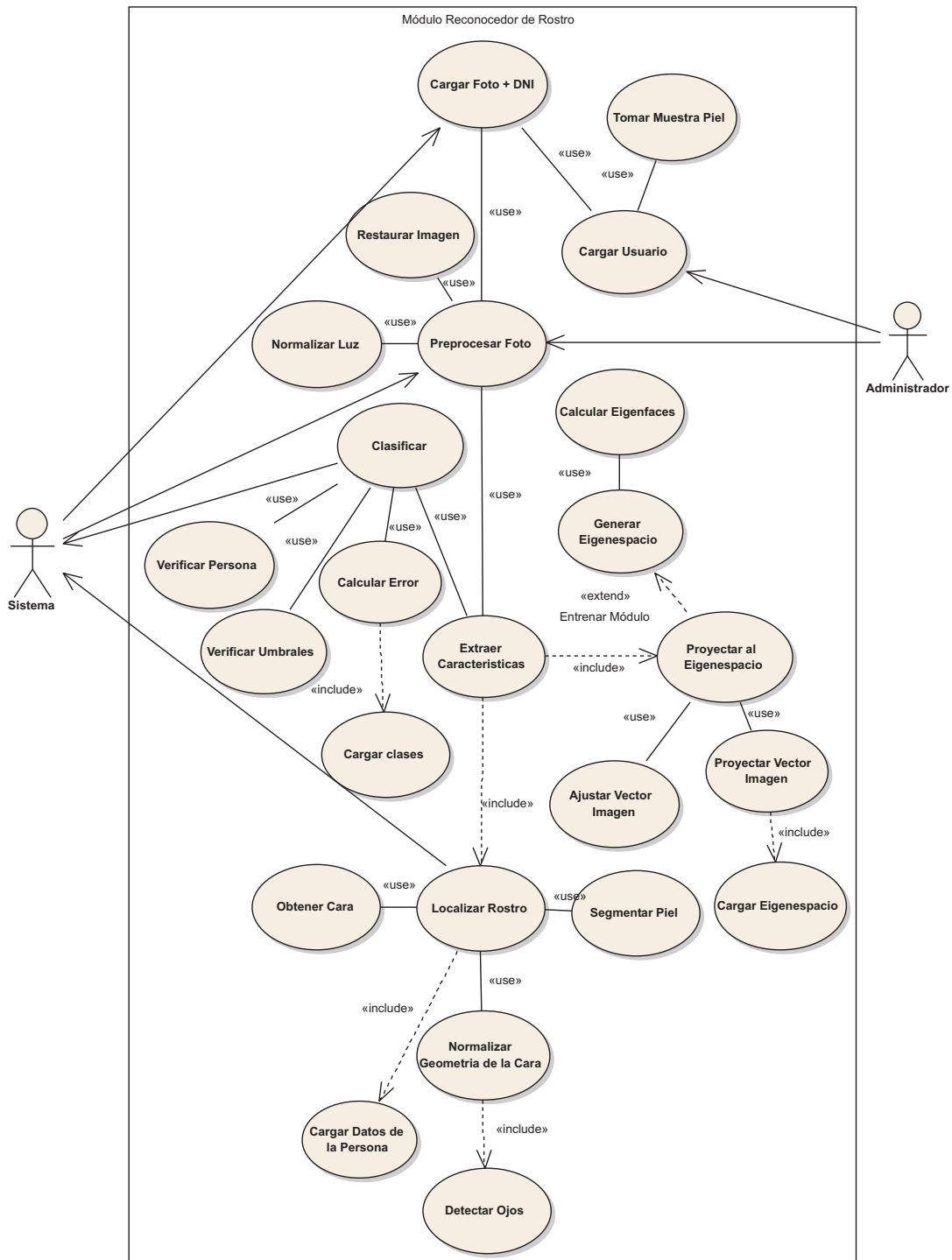


Figura 3.4: Diagrama de casos de uso detallado

| |
|--|
| Caso de uso: Preprocesar Foto |
| Actor: Sistema, administrador |
| Descripción: Mejorar la calidad de la imagen por medio de técnicas del PDI. |
| Caso normal: |
| 1) Se aplica un filtro de mediana. 2) Se aplica una transformación de potencia. |

| |
|--|
| Caso de uso: Extraer Características |
| Actor: — |
| Descripción: Parametrizar la imagen para poder clasificarla. |
| Caso normal: |
| 1) Se localiza el rostro dentro de la imagen. 2) Se proyecta al eigenespacio. |

| |
|--|
| Caso de uso: Localizar Rostro |
| Actor: Sistema |
| Descripción: Obtener una subimagen normalizada de la cara . |
| Caso normal: |
| 1) Se cargan los datos de la piel del usuario a partir del DNI ingresado. 2) Se segmenta la imagen con los datos cargados. 3) Se aplica una plantilla para obtener la cara y se recorta la imagen en un tamaño definido. 4) Se localizan los ojos dentro de la nueva imagen obtenida. 5) Con los ojos se normaliza la geometría y se vuelve a recortar la imagen original. |
| Alternativa: |
| 3.1) El módulo devuelve FALSO al sistema al no encontrar una región que pueda ser considerada cara luego de la segmentación. 4.1) Se recorta la imagen de un tamaño específico y no se normaliza, ya que no se pueden localizar los ojos. |

| |
|--|
| Caso de uso: Proyectar al eigenespacio |
| Actor: — |
| Descripción: Proyecta la imagen del rostro normalizada al espacio cara. |
| Caso normal: |
| <ol style="list-style-type: none"> 1) Se ajusta la imagen, sustrayendo la imagen media. 2) Se carga la base. 3) Se proyecta la imagen ajustada multiplicando por las eigenfaces generando un punto en el eigenespacio. |
| Alternativa: |
| <ol style="list-style-type: none"> 1.1) Cuando el módulo se entrena, se generan las eigenfaces y la imagen media y luego se ajustan las subimágenes, restando la imagen media. 3.1) Cuando el módulo se entrena, se proyectan las imágenes ajustadas multiplicándolas por las eigenfaces. Esto genera puntos en el eigenespacio (patrones) que se guardan para ser utilizados en la clasificación. |

| |
|--|
| Caso de uso: Generar eigenespacio |
| Actor: — |
| Descripción: Calcula las eigenfaces y la imagen media. |
| Caso normal: |
| <ol style="list-style-type: none"> 1) Calcula la imagen media de toda las imágenes cargadas y la guarda. 2) Calcula las eigenfaces y las guarda. |

| |
|---|
| Caso de uso: Clasificar |
| Actor: Sistema |
| Descripción: Verificar que la imagen cargada pertenezca al usuario que ingresó el DNI. |
| Caso normal: |
| 1) Se calcula el error entre el nuevo patrón (punto en el eigenespacio) y los patrones almacenados en el entrenamiento. 2) Se toma el menor de los errores y se verifica que sea menor a cierto umbral. 3) Se verifica que el DNI ingresado sea el mismo que el del usuario correspondiente al patrón con menor error, devolviendo <i>VERDADERO</i> . |
| Alternativa: |
| 2.1) El módulo devuelve <i>FALSO</i> , si el menor error obtenido es mayor que el umbral. 3.1) El DNI ingresado no coincide con el de la persona con menor error, el módulo devuelve <i>FALSO</i> . |

3.4 Diagrama de Clase

Un diagrama de clase muestra un conjunto de clases y sus relaciones, siendo muy utilizados en el modelado de sistemas orientados a objetos [28].

Estos diagramas se usan para modelar la vista de diseño estática de un sistema, incluyendo el vocabulario del mismo. Son importantes no sólo para visualizar, especificar y documentar modelos estructurales, sino también para construir sistemas ejecutables aplicando ingeniería directa e inversa.

En la Figura 3.5 se observa el diagrama diseñado para el módulo de reconocimiento de rostro, en donde se definieron 5 clases: CImg, Persona, Imagen, Base y Patron.

La clase *Persona* engloba las operaciones necesarias para poder almacenar y cargar los datos de la piel tales como la media y el desvío estandar, que luego son utilizados para localizar el rostro. Además guarda el identificador de la persona (DNI).

La clase *Imagen* incluye las operaciones sobre imágenes. Hereda todos los métodos y atributos de la clase de licencia libre (CImg) que se utiliza en este proyecto. Las funciones más importantes que realiza son:

- Restaurar la imagen.
- Segmentar la piel.
- Localizar la cara.
- Normalizar la geometría.

La clase *CImg* se utiliza para realizar operaciones básicas en el manejo de imágenes, tales como:

- Cargar imágenes en formato jpg, bmp, etc.
- Realizar operaciones de correlación, erosión, dilatación, etc.
- Manipular imágenes como matrices.
- Cortar imágenes, efectuar rotaciones, etc.
- Realizar las transformaciones a diferentes espacios de color.

La clase *Base* se encarga de todas las operaciones necesarias para generar el espacio cara, tales como:

- Calcular, guardar y cargar la imagen media.
- Calcular, guardar y cargar las eigenfaces.
- Calcular, guardar y cargar las proyecciones de los patrones en el nuevo espacio.

La Clase *Patron* engloba las funciones encargadas de la clasificación, tales como:

- Calcular los errores.
- Verificar los umbrales.
- Verificar la identidad de la persona.

Esta clase es la encargada de retornar un valor verdadero o falso según el resultado de dicha clasificación.

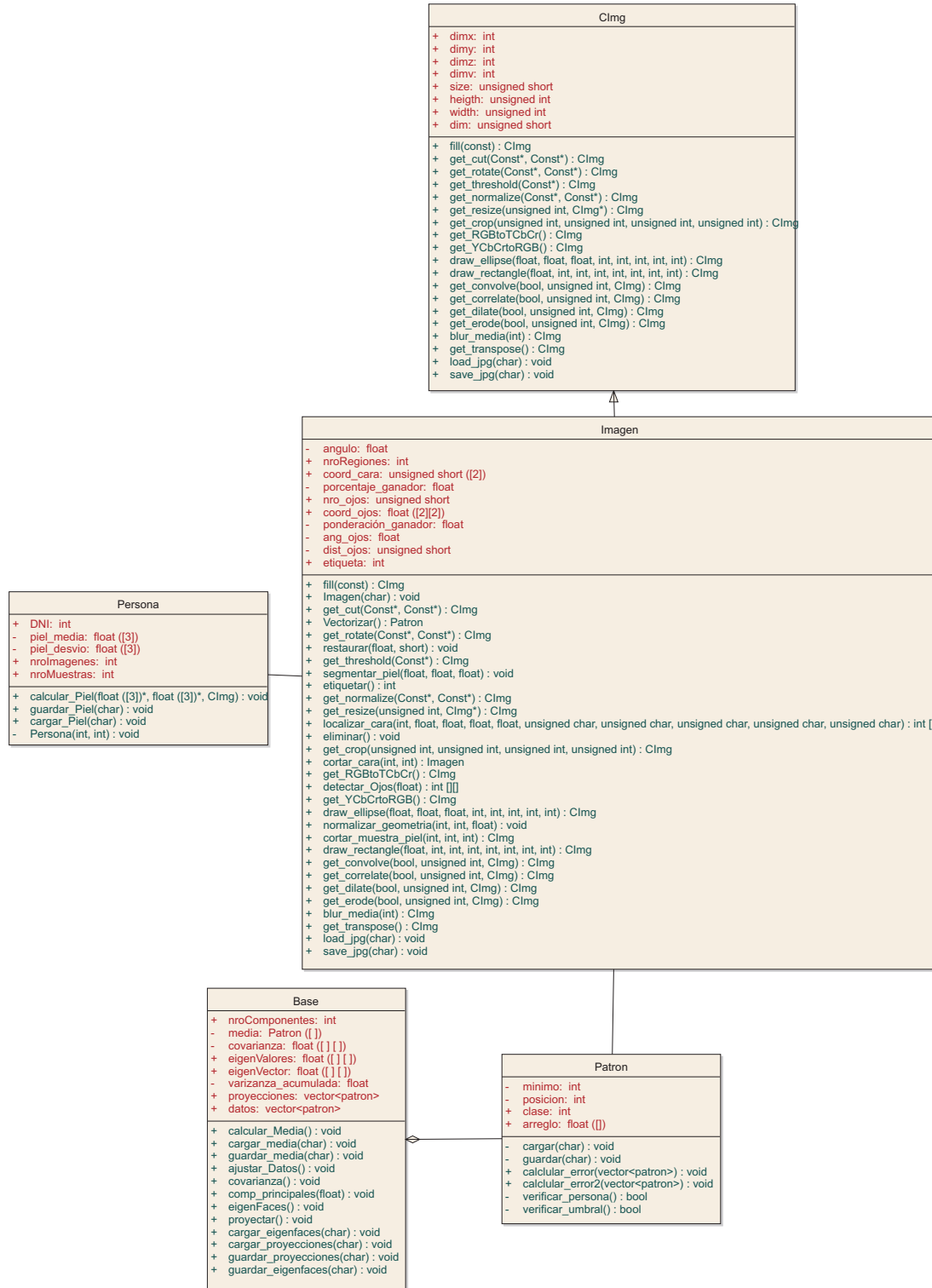


Figura 3.5: Diagrama de clase

3.5 Diagrama de Secuencia

Un diagrama de secuencia muestra una interacción ordenada según la secuencia temporal de eventos. En particular, muestra los objetos participantes en la interacción y los mensajes que intercambian ordenados según su secuencia en el tiempo [31]. El eje vertical representa el tiempo, y en el eje horizontal se colocan los objetos y actores participantes en la interacción. Cada objeto o actor tiene una línea vertical, y los mensajes se representan mediante flechas entre los distintos objetos. El tiempo fluye de arriba hacia abajo.

En el diagrama de secuencia de la Figura 3.6 se observa la fase de reconocimiento del rostro. El sistema inicia el módulo cuando un usuario ingresa su DNI instanciando la clase *Persona* encargada de cargar los datos de piel de la persona y registrar el DNI ingresado. Luego se instancia la clase *Imagen* que carga la imagen, la restaura y con los datos de piel la segmenta, obteniendo como resultado una subimagen de la cara normalizada. A dicha subimagen se la transforma en un vector instanciando la clase *Patron*. A continuación la clase *Base* carga la imagen media, centra el vector imagen, carga las eigenfaces y proyecta al nuevo espacio generando un patrón.

Finalmente, se cargan todas las proyecciones (patrones) que se obtuvieron en la fase de entrenamiento con sus respectivos DNI y la clase *Patron* –por medio de sus métodos–, calcula el error de los patrones con respecto al patrón proyectado anteriormente, busca el menor de todos los errores y obtiene el DNI del mismo para compararlo con el ingresado. Si los DNI coinciden el módulo retorna *verdadero*, en caso contrario *falso*.

La Figura 3.7 muestra la secuencia de eventos y mensajes en la fase de entrenamiento del módulo. El administrador genera un archivo de texto con los nombres de las imágenes que se quieren agregar al módulo, luego ejecuta la fase de entrenamiento. Lo primero que realiza el módulo es instanciar la clase *Persona* tantas veces como personas haya para entrenar. Esta clase registra el DNI obtenido del nombre de la imagen, a continuación se instancia la clase *Imagen* por cada imagen del usuario para cargar y restaurar las imágenes y cortar una muestra de piel. Luego la clase *Persona* calcula la media y el desvío estándar de las muestras y las almacena para su posterior utilización en la siguiente etapa.

Luego se pasa a la etapa de localización del rostro, donde se vuelve a usar la clase *Imagen* para cada imagen del usuario. Nuevamente carga, restaura, segmenta, localiza y se corta la cara y finalmente se localizan los ojos para

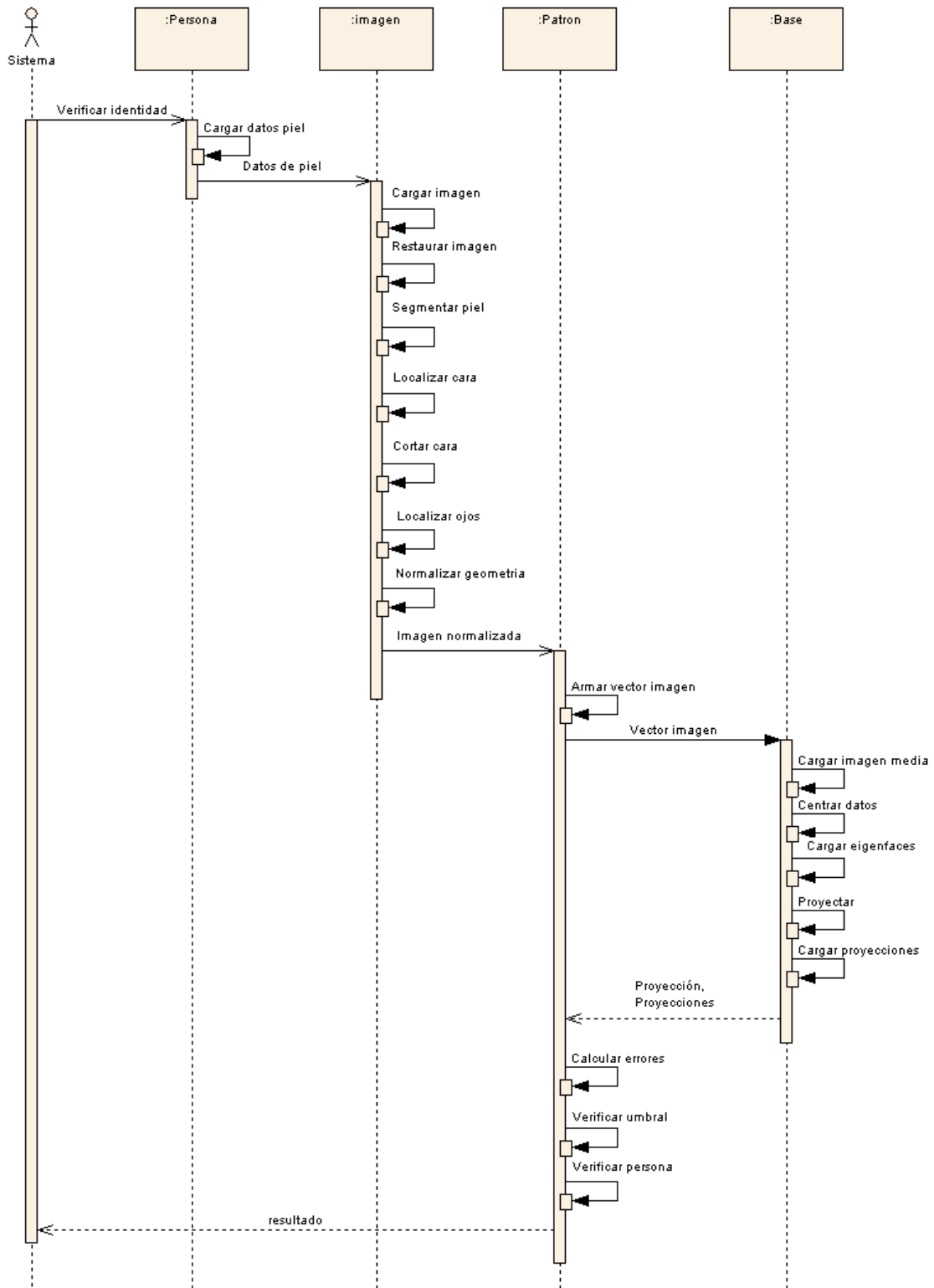


Figura 3.6: Diagrama de secuencia del módulo reconocedor

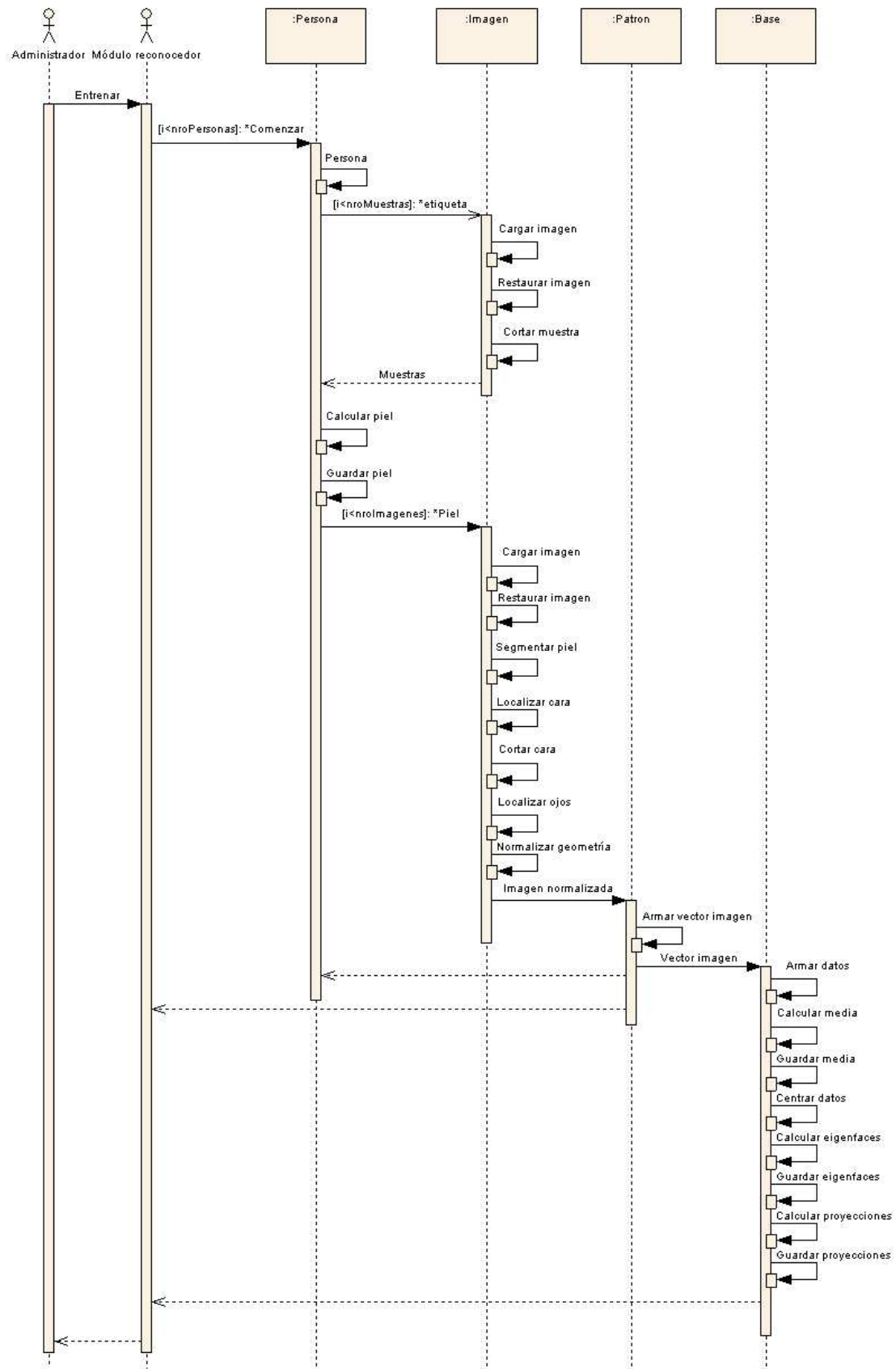


Figura 3.7: Diagrama de secuencia del módulo de entrenamiento

normalizar. Para finalizar se instancia la clase *Patron*, que arma un vector con cada imagen normalizada.

Todas las etapas enunciadas en el párrafo anterior se repiten por cada usuario distinto. Con los vectores de todas las imágenes de todos los usuarios, se instancia la clase *Base* que arma una matriz, donde cada columna es un vector imagen, calcula y guarda la imagen media de esta matriz, centra los datos restándole la imagen media a cada columna, calcula las eigenfaces y proyecciones y las guarda.

Implementación del módulo de reconocimiento facial

4.1 Introducción

Un sistema de reconocimiento facial consta básicamente de tres bloques fundamentales: (1) Preproceso, (2) Extracción de características y (3) Clasificación. En el primer paso se realizan, sobre la imagen de entrada, los procesos de PDI necesarios para resaltar los detalles de interés. En la segunda etapa se extraen las características a utilizar en el reconocimiento, formando uno o varios vectores patrón y por último se efectúa la clasificación, comparando el nuevo patrón con los patrones de usuarios registrados. El algoritmo desarrollado en este proyecto sigue estos pasos, y se muestra gráficamente en la Figura 1.1 (reproducida aquí para conveniencia del lector como Figura 4.1).

El preproceso de la imagen de entrada reduce el ruido introducido por el dispositivo de captura y normaliza la iluminación. La extracción de características localiza el rostro en la imagen y proyecta al eigenspacio, obteniendo un vector patrón que contiene las propiedades más relevantes de la imagen.

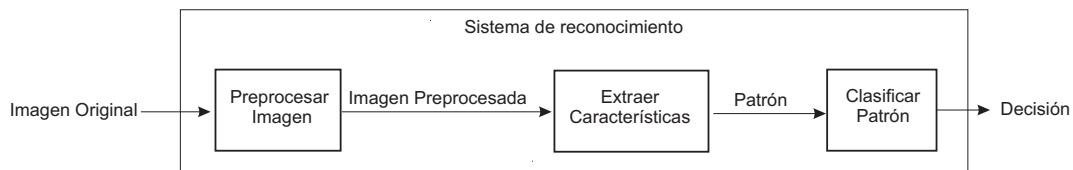


Figura 4.1: Sistema genérico de reconocimiento facial

La clasificación consiste en comparar el patrón con los patrones de usuarios registrados, determinando si la imagen capturada corresponde al DNI ingresado o no. De esta manera, el módulo toma como entrada una imagen y el DNI ingresado y devuelve como salida *verdadero* o *falso*.

Por otro lado, el algoritmo tiene dos fases de ejecución, una de entrenamiento (en tiempo diferido) y una de clasificación (sistema en funcionamiento). En la primera fase, un total de N imágenes de entrenamiento $I = \{I_1, \dots, I_N\}$ de un grupo de D usuarios diferentes se preprocesan. Para cada usuario se toman muestras de piel (subimágenes cuyos píxeles representen el tono de piel) sobre algunas de las fotos de entrenamiento. Luego, para cada usuario se forma una única imagen con las muestras de piel del mismo y se calcula la media \mathbf{m}_d y el desvío estándar σ_d con $d = 1, \dots, D$ de los valores de intensidad de cada componente de color de la imagen. Tanto \mathbf{m}_d como σ_d son vectores de tres componentes: R, G y B, y representan en qué rango del espacio color se distribuye el color de piel de cada usuario. Con estos parámetros, de cada imagen I_i inicial se extrae una subimagen F_i con el rostro localizado que se denomina *imagen facial*. A partir de *todas* las imágenes faciales obtenidas de I se calculan K eigenfaces ($K \ll N$) y se proyectan las F_i al eigenspacio, obteniendo un vector patrón \mathbf{x}_i de dimensión K para cada imagen del conjunto inicial. Cada vector \mathbf{x}_i tiene asociado una etiqueta de clase $\omega_c \in \{\omega_1, \dots, \omega_D\}$, heredada de la imagen I_i inicial. Obviamente, todas las imágenes de una misma persona comparten la misma etiqueta de clase.

La otra fase toma una imagen de entrada $I_p \notin I$, la preprocesa, localiza la imagen facial F_p en base a los datos de piel del usuario. Luego, proyecta con las eigenfaces calculadas en el entrenamiento obteniendo el vector característico \mathbf{x}_p y clasifica a partir de la distancia de \mathbf{x}_p con respecto a las proyecciones \mathbf{x}_i almacenadas previamente.

Cabe destacar que el dispositivo de captura utilizado se encuentra ubicado detrás de un vidrio, en un pasillo con luz natural y fondo complejo, y adquiere imágenes de 320×240 en RGB. En base a estas dificultades y a los requerimientos analizados en la Sección 3.2 se desarrolla el algoritmo detallado a continuación. El mismo fue programado en C++, utilizando una librería para el procesamiento de imágenes y el cálculo de autovectores y autovalores llamada CImg. El módulo implementado se detalla a continuación, justificando los pasos realizados a partir de los obstáculos presentados. Además, se ejemplifica con secuencias de imágenes a fin de facilitar el entendimiento.

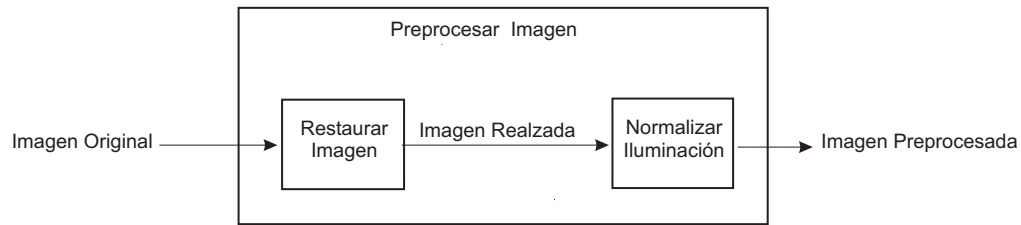


Figura 4.2: Preproceso

4.2 Preproceso

Esta etapa (ver Figura 4.2) se encarga de restaurar y normalizar las imágenes de entrada tanto en el entrenamiento como en las pruebas, a fin de lograr una correcta localización y un buen reconocimiento. En la Figura 4.3 se muestra un ejemplo.

4.2.1 Restauración

En las imágenes de la base BADAFFI se detectó presencia de ruido impulsivo (intensidades extremas en ciertos píxeles) generado por la cámara web. Por esto, se aplica un filtro de mediana:

$$I_j(x, y) = \text{mediana}_{(s,t) \in S_{xy}} \{g(s, t)\} \quad (4.1)$$

donde $I_j(x, y)$ es la imagen a procesar (I_i o I_p) y S_{xy} representa el conjunto de coordenadas de la máscara g centrada en (x, y) .

4.2.2 Normalización de la iluminación

Los cambios inducidos por la iluminación son a menudo más grandes que las diferencias entre los individuos y provocan malas clasificaciones en sistemas basados en comparación [1], por esto es crucial un proceso de normalización.

En este caso en particular, la falta de luz artificial produjo gran variabilidad en la iluminación de las imágenes capturadas, presentando mayores

problemas la falta de iluminación dependiendo fundamentalmente del horario de adquisición. Por esto, se compensa la luminosidad de las imágenes más oscuras normalizadas de 0 a 1, mediante un proceso sencillo como la transformación de potencia dada por:

$$I_j(x, y) = I_j(x, y)^\gamma \quad (4.2)$$

donde nuevamente $I_j(x, y)$ es I_i o I_p –según se entrene o pruebe– y $0 < \gamma < 1$ para amplificar los tonos oscuros y contraer los claros, logrando el efecto deseado. Para adaptar la transformación a la cantidad de luz de cada imagen, se toma un valor de γ variable de acuerdo a la siguiente ecuación:

$$\gamma = c_1\mu_I + c_2 \quad (4.3)$$

donde c_1 y c_2 son constantes definidas mediante experimentación y μ_I es el valor medio de la imagen a normalizar en escala de grises y en el rango $[0, 1]$. De esta forma, γ es directamente proporcional al brillo.

Luego se reacomodan los valores R, G, B en el rango $[0, 255]$.



Figura 4.3: Ejemplo de preproceso de imagen capturada a las 20:49 hs. a) Imagen original, b) imagen realzada, c) imagen con iluminación normalizada.

4.3 Extracción de características

La extracción de características realiza todas las tareas necesarias para localizar el rostro, normalizarlo y luego parametrizar la imagen facial, obteniendo un vector característico representativo del rostro.

4.3.1 Localización

Para el correcto funcionamiento del reconocedor es indispensable localizar el rostro del usuario eliminando el fondo, que no contiene información útil a los fines de la verificación. Se debe tener en cuenta que puede haber más de un rostro en la imagen y que las muestras fueron capturadas en un fondo complejo propenso a producir falsas localizaciones (por presencia de macetas, puertas u otros objetos con color similar a la piel). Dada la imposibilidad de efectuar la localización a partir de características locales por la complejidad y baja calidad de las imágenes tomadas *“in situ”*, la localización se basa en una representación global de la cara.

Las imágenes se someten a dos segmentaciones de piel sucesivas, ambas en el espacio color RGB. La primera segmentación opera por igual en todos los usuarios aplicando ciertas reglas definidas en [26], basadas en que el color de piel se mantiene en cierto rango del espacio color. Así, el píxel en la posición (x, y) de I_j es clasificado como piel y mantiene sus valores RGB si cumple con las siguientes condiciones:

$$\begin{aligned} R &> 95, G > 40, B > 20 \\ \max \{R, G, B\} - \min \{R, G, B\} &> 15 \\ |R - G| &> 15 \\ R &> G, R > B \end{aligned} \quad (4.4)$$

siendo R, G, B los tres componentes del píxel. En caso que el píxel no cumpla las condiciones, se clasifica como no piel y se pone a negro.

La ventaja de este método es la simplicidad en las reglas conformando un clasificador muy rápido. No obstante, los parámetros de las desigualdades pueden variar en base a ciertas características como la luz o la variabilidad de tonos de piel. La Figura 4.5a muestra el resultado de este proceso.

Después de las reglas presentadas, se refinan las regiones de piel con una

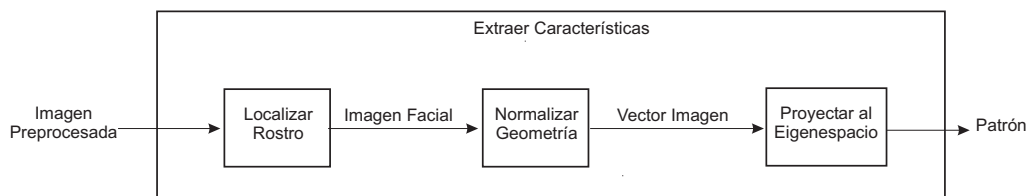


Figura 4.4: Extracción de características



Figura 4.5: Ejemplo de localización. a) Primera segmentación, b) segunda segmentación, c) combinación de plantilla sobre las regiones, d) subimagen con la cara localizada.

segunda segmentación tomando como parámetros el vector de medias \mathbf{m}_d y desvíos σ_d , calculados a partir de muestras de piel tomadas para cada usuario. Estos vectores son de 3 componentes correspondientes a R, G y B. Entonces para una imagen del usuario ω_d , cada píxel es clasificado como piel si:

$$\begin{aligned} R &\in [m_{d_R} - k\sigma_{d_R}, m_{d_R} + k\sigma_{d_R}], \\ G &\in [m_{d_G} - k\sigma_{d_G}, m_{d_G} + k\sigma_{d_G}] \text{ y} \\ B &\in [m_{d_B} - k\sigma_{d_B}, m_{d_B} + k\sigma_{d_B}] \end{aligned} \quad (4.5)$$

donde k es una constante definida por experimentación que permite ajustar las dimensiones del cubo de segmentación. Cabe destacar que es igual para todos los usuarios.

Esta segmentación tiene en cuenta parámetros específicos de los usuarios, logrando una clasificación más personalizada de las regiones de piel que resulta en una mejor localización (ver Figura 4.5b).

Una vez obtenidas las posibles regiones cara, éstas se etiquetan mediante un algoritmo recursivo denominado *flood fill*. Luego, se dibuja una plantilla elíptica de tamaño prefijado en el centro de cada región, obtenido a partir de las ecuaciones de centro de masa. Se calcula el porcentaje de píxeles de la región que caen dentro de la plantilla. Si este valor es muy bajo, o por el contrario es muy alto, la región se descarta por considerar que no puede representar una cara. Sobre las regiones que superan estos umbrales, se realizan pequeñas traslaciones de la plantilla en los cuatro sentidos y rotaciones a ambos lados para encontrar la elipse que mejor se ajuste a la región contemplando de esta manera variaciones en la pose y distancia (Figura 4.5c). Por último, se recorta un rectángulo (que contiene la cara) rotado con el ángulo

especificado por la elipse ganadora (Figura 4.5d).

4.3.2 Normalización de la geometría

La normalización posibilita la comparación de patrones estandarizados geoméricamente. Los ojos constituyen una característica facial importante debido a que la distancia ocular es relativamente constante [19]. Así, la detección de los mismos es un punto clave en la determinación de la orientación de la cara. Por otro lado, al no estar el sistema en funcionamiento las personas se colocaron frente a la cámara sin tener en cuenta la distancia a ésta ni la ubicación respecto al centro de la imagen, lo que dificulta la tarea para el sistema. Por lo anteriormente expuesto, se normaliza el tamaño y la orientación del rostro contenido en las imágenes en base a la distancia entre ojos.

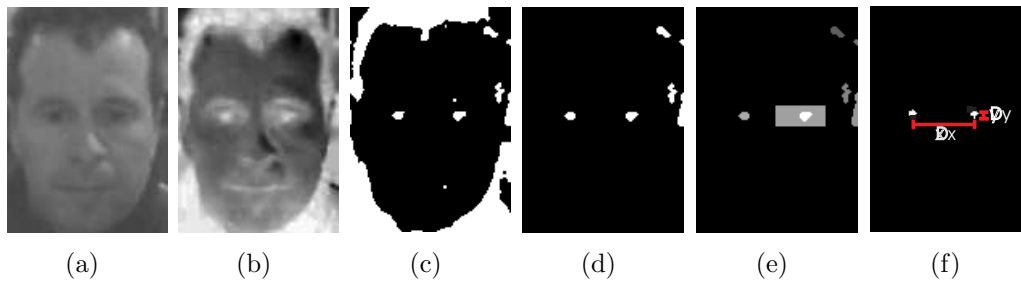


Figura 4.6: Ejemplo de detección de ojos. a) Componente Y de la imagen facial, b) mapa, c) mapa binarizado, d) filtrado de regiones, e) búsqueda del par de ojos, f) ojos detectados.

Puesto que los ojos generalmente contienen píxeles oscuros y claros en la luminancia, los operadores morfológicos de dilatación y erosión en escala de grises pueden ser usados para enfatizar los píxeles más brillantes y más oscuros alrededor de los ojos [15]. Utilizando las operaciones morfológicas mencionadas con un elemento estructurante cuadrado de 3×3 , se construye un mapa en base a la componente de luminancia Y del modelo $YCbCr$ explicado en la Sección 2.1.4, dado por:

$$Mapa = \frac{Y(x, y) \oplus g(x, y)}{Y(x, y) \ominus g(x, y) + 1} \quad (4.6)$$

donde \oplus y \ominus denotan los operadores de dilatación y erosión en escala de grises respectivamente y g representa el elemento estructurante. Como muestra la Figura 4.6b, este mapa resalta los ojos y otras zonas de contraste.

El mapa se binariza (Figura 4.6c) de acuerdo a un umbral fijado a partir del brillo medio del mapa, obteniendo un conjunto de posibles ojos. Estas regiones se etiquetan con diferentes valores para poder identificarlas, se filtran según tengan demasiados o muy escasos píxeles como para representar un ojo (Figura 4.6d) y se busca el par de ojos correctos tomando cada región de la mitad izquierda del mapa y buscando otra región que caiga dentro de una plantilla rectangular (de tamaño fijo) a la derecha de la región de referencia como se observa en la Figura 4.6e. Los pares que cumplen esta condición son los pares de ojos candidatos y se validan mediante la ponderación de ciertas propiedades del par como la posición donde aparecen y la distancia entre ellos. El par de ojos ganador será aquel que obtenga la mayor ponderación.

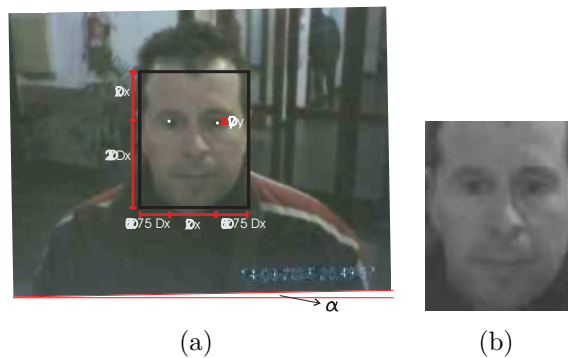


Figura 4.7: Normalización: a) Uso de los ojos como parámetros de normalización. b) Imagen normalizada.

Por último si la ponderación del par ganador supera un umbral preestablecido (Figura 4.6f) se calcula la distancia ocular horizontal Dx y vertical Dy y el ángulo $\alpha = \arctan \frac{Dy}{Dx}$. Con estos parámetros se toma la imagen realzada y se recorta la subimagen facial normalizada, como se muestra en la Figura 4.7. Cuando no se detectan los ojos se utiliza la imagen localizada. Llegado este punto, se pasa a escala de grises, se redimensiona a un tamaño constante obteniendo la imagen facial resultante F_i utilizada en el reconocimiento.

4.3.3 Proyección Eigenfaces

Como se mencionó con anterioridad, se hace imposible usar características locales dadas las cualidades de las imágenes disponibles. No obstante, existen muchas técnicas holísticas de reconocimiento que toman como parámetro la cara completa. Entre estos métodos se encuentra Eigenfaces, uno de los

más exitosos en reconocimiento facial, basado en conceptos de la teoría de la información. Básicamente, el método descompone las imágenes en un pequeño conjunto de imágenes de características llamadas “eigenfaces”, las cuales corresponden a las componentes principales del conjunto de imágenes de entrenamiento. El reconocimiento se realiza proyectando una nueva imagen dentro del subespacio de las eigenfaces (“eigenespacio”) y luego se clasifica comparando su posición en el eigenespacio con las posiciones de individuos conocidos.

El método de eigenfaces detallado en [34, 33, 23] se resume a continuación:

Una imagen de $X \times Y = P$ puede ser considerada como un punto en el espacio P dimensional. Sin embargo, como las imágenes faciales comparten en general las mismas características, pueden ser representadas en un subespacio de menor dimensión cuyos ejes son las *eigenfaces*. Así, se busca un conjunto de puntos (vectores) que contengan la mayor cantidad de información respecto a la distribución de imágenes faciales en el espacio imagen. Cada uno de estos vectores-imagen $X \times Y$ se constituye como combinación lineal de las imágenes originales. Al ser estos vectores los eigenvectores de la matriz de covarianza del conjunto de imágenes F de entrenamiento, se denominan usualmente eigenfaces. Cada uno de los eigenvectores se ordena en forma decreciente de acuerdo al eigenvalor asociado (Figura 4.8a).

El primer paso del método es calcular la imagen media $M = \sum_{i=1}^N F_i$ (Figura 4.8b) para centrar las imágenes del conjunto haciendo $A_i = F_i - M$. A este conjunto de imágenes se le aplica PCA en busca de K vectores ortonormales \mathbf{u}_l y sus eigenvalores asociados λ_l que mejor describan la variabilidad de los datos. Los vectores \mathbf{u}_l y los escalares λ_l son los eigenvectores y eigenvalores respectivamente, de la matriz de covarianza

$$\begin{aligned} C &= \frac{1}{N} \sum_{i=1}^N A_i A_i^t \\ &= AA^t \end{aligned} \quad (4.7)$$

donde $A = [A_1 A_2 \dots A_N]$ y la matriz C es por lo tanto de $P \times P$. Determinar los P eigenvectores y eigenvalores en una tarea muy costosa computacionalmente, por lo que se busca simplificar los cálculos.

Si el número de puntos en el espacio imagen es menor que la dimensión del espacio, o lo que es igual, si el número de imágenes es menor que la cantidad de píxeles de cada imagen ($N < P$) existen sólo N eigenvectores significativos, ya que los otros tienen asociados eigenvalores nulos. Entonces, se pueden calcular los eigenvectores resolviendo primero un problema matri-

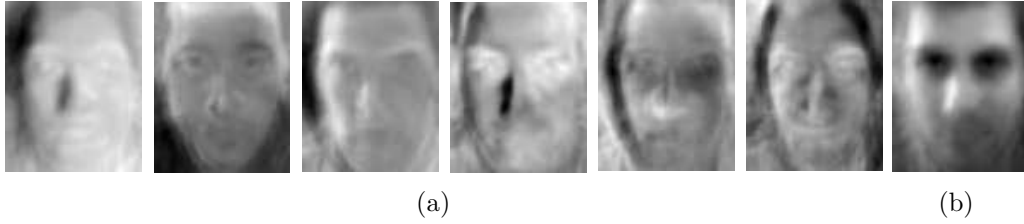


Figura 4.8: a) Primeras 6 eigenfaces de un conjunto de 72 imágenes de entrenamiento. b) Imagen media del conjunto.

cial de $N \times N$, y tomar combinaciones lineales de los vectores resultantes. Considerar los eigenvectores \mathbf{v}_l de $A^t A$ tal que

$$A^t A \mathbf{v}_l = \mu_l \mathbf{v}_l \quad (4.8)$$

Premultiplicando ambos lados por A , se obtiene

$$\begin{aligned} AA^t A \mathbf{v}_l &= \mu_l A \mathbf{v}_l \\ C A \mathbf{v}_l &= \mu_l A \mathbf{v}_l \end{aligned} \quad (4.9)$$

de donde se deduce que $A \mathbf{v}_l$ son los eigenvectores de $C = AA^t$.

Siguiendo este análisis, se construye la matriz $L = A^t A$ y se buscan los N eigenvectores \mathbf{v}_l de L . Estos vectores determinan combinaciones lineales de las N imágenes de entrenamiento para formar las eigenfaces \mathbf{u}_l :

$$\mathbf{u}_l = \sum_{k=1}^N \mathbf{v}_{lk} A_k, \quad l = 1, \dots, N \quad (4.10)$$

Los eigenvalores asociados permiten ordenar los eigenvectores de acuerdo a la cantidad de información de variabilidad que contengan.

Con esta simplificación los cálculos se reducen notablemente, ya que en general $N \ll P$. Por ejemplo, para un total de 200 usuarios y con imágenes faciales normalizadas de tamaño $P = 80 \times 100$, esta simplificación es válida si tenemos un promedio de imágenes por usuario Q dada por:

$$\begin{aligned} DQ &\leq P \\ 200Q &\leq 8000 \\ Q &\leq 40 \end{aligned}$$

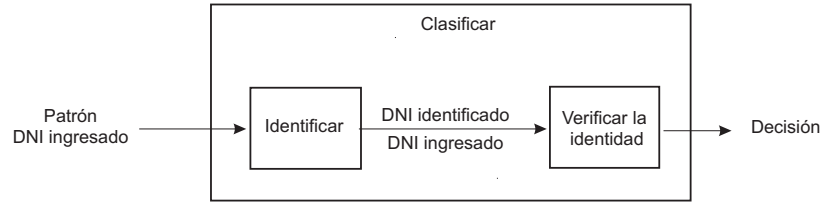


Figura 4.9: Clasificación

Las eigenfaces calculadas conforman un conjunto base con el cual se describe las imágenes faciales. Puesto que no se requiere la reconstrucción de la imagen, se usan tan sólo $K \ll N$ eigenfaces calculadas a partir de los K eigenvectores más significativos de L , elegidos como aquellos con mayor eigenvalor asociado.

Finalmente, las imágenes faciales F_i son proyectadas al eigenspacio mediante:

$$\mathbf{x}_i = U^t A_i \quad (4.11)$$

donde $U = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_K]$. Cada componente del vector \mathbf{x}_i describe la contribución de cada eigenface en la representación de la imagen de entrada. Los \mathbf{x}_i con sus etiquetas ω_i asociadas son almacenados para la clasificación.

Para la fase de prueba, dada la imagen F_p se la centra con la imagen media M y se la proyecta al eigenspacio aplicando (4.11) obteniendo el vector característico \mathbf{x}_p .

4.4 Clasificación

En esta etapa se hace la comparación entre la imagen facial F_p obtenida de I_p y proyectada al eigenspacio, con las proyecciones de los usuarios del sistema determinando si la identidad ingresada corresponde a la reconocida. Como es de esperar, esta etapa no se aplica durante el entrenamiento.

4.4.1 Identificación

El vector \mathbf{x}_p se utiliza para determinar cuál de las clases (usuarios) describe mejor la imagen de prueba. Para esto, se aplica la clasificación por mínima

distancia, que busca la clase c que provee la mejor descripción de una imagen facial entrante según se minimice la distancia euclídea:

$$e_c^2 = \|\mathbf{x}_p - \mathbf{x}_c\|^2 \quad (4.12)$$

donde \mathbf{x}_c es el promedio de las \mathbf{x}_i cuya etiqueta es ω_c , es decir, el promedio de las proyecciones de cada usuario. Una imagen es clasificada como perteneciente a la clase c cuando el mínimo e_c es menor que un umbral prefijado θ_e . Este parámetro, pretende definir cuán seguro es el sistema. En caso contrario, la cara es tratada como desconocida y se retorna una decisión negativa.

Otra manera sencilla de clasificar es tomar las k mínimas distancias, método conocido como *k-vecinos más cercanos*, el cual busca los k vectores \mathbf{x}_i que minimicen la distancia:

$$e_c^2 = \|\mathbf{x}_p - \mathbf{x}_i\|^2 \quad (4.13)$$

Entonces, si $k_c(\mathbf{x}_p)$ es el número de muestras de la clase c presentes en los k vecinos más próximos a \mathbf{x}_p , esta regla puede expresarse como:

$$\mathbf{x}_p \in c \quad \text{si} \quad k_c(\mathbf{x}_p) = \max_{i=1..d} k_i(\mathbf{x}_p) \quad (4.14)$$

Nuevamente, los vecinos son tenidos en cuenta si minimizan la distancia euclídea y si esta distancia es menor al umbral θ_e .

Ambas opciones han sido programadas y se evaluarán sus desempeños en el desarrollo de los experimentos.

4.4.2 Verificación de la identidad

Según se expuso previamente, como resultado de la identificación se obtiene la etiqueta de la clase c que mejor representa a la imagen facial analizada. Luego se compara el DNI del usuario c del sistema con el DNI ingresado y se asigna verdadero a la salida del sistema si son iguales, o falso si no lo son. De esta manera, se completa la verificación que realiza el módulo. Además, si la respuesta es positiva y e es muy pequeño, la imagen de entrada I_p se almacena en un directorio especial para formar parte de I en el próximo entrenamiento del sistema.

Marco experimental

5.1 Descripción de los datos

Para los experimentos a desarrollar se utilizan imágenes de la base de datos BADAFFI, conformada por imágenes del personal de la FI-UNER adquiridas con una cámara web en el registro de ingreso y egreso del establecimiento.

La base se creó con imágenes registradas en septiembre de 2005 y agosto de 2006. El total de las muestras fueron adquiridas de forma automática con fondo, luz, distancia, pose y expresión totalmente natural y en horarios que van de las 6 a las 23 hs., con una resolución de 320×240 píxeles en formato JPEG con una tasa de compresión de 40,1. En el Apéndice A.1 se describe detalladamente la base de datos generada. Para resumir la descripción de los datos empleados en los experimentos, podemos citar que del conjunto original se utilizó un subconjunto de 457 imágenes de 27 personas que tuvieran variaciones aceptables de pose, iluminación y calidad. Finalmente, el conjunto fue ampliado mediante espejado, logrando una extensión válida del conjunto original como se describe en [37], a los efectos de aumentar la cantidad de datos disponibles para la experimentación.

5.2 Diseño de experimentos

Con la meta de ajustar los parámetros c_1, c_2, k, θ_e, K del sistema y verificar el desempeño del algoritmo implementado se realizan pruebas varias mostrando el resultado de las técnicas desarrolladas en aspectos diversos de los procesos más importantes del sistema y en situaciones particulares.

En primer lugar se evalúa la normalización de la luz a través de los histogramas de frecuencia de imágenes capturadas en diferentes horarios del día. Para esto, se muestran las imágenes con sus histogramas correspondientes antes y después de la aplicación de la transformación de potencia.

A continuación se estudia el proceso de detección de ojos para los casos particulares en los que el usuario tiene anteojos, mostrando cómo afecta este hecho al intento de normalización de la geometría.

Concluidas las pruebas correspondientes a la localización y normalización de la cara, se verifica la tasa de reconocimiento del sistema con los métodos de clasificación presentados en la Sección 4.4, como así también el rendimiento ante la variación del umbral θ_e .

La última prueba consiste en distorsionar una imagen de diferentes maneras para comprobar el desempeño del reconocedor en situaciones de oclusiones parciales o presencia de ruido.

5.3 Implementación y resultados

5.3.1 Experimento 1: Normalización de la iluminación

Para evaluar el proceso de normalización de la iluminación se seleccionan imágenes tomadas a diferentes horarios del día y se calcula su histograma previo a la normalización. Luego se muestran las imágenes normalizadas con sus respectivos histogramas (Figura 5.1). En la Tabla 5.1 se muestran los valores de la media μ_0 de la imagen original en escala de grises, el valor de γ calculado según (4.3) con $c_1 = 1, 2$ y $c_2 = 0, 3$ y el valor de la media μ_f obtenida luego de la normalización.

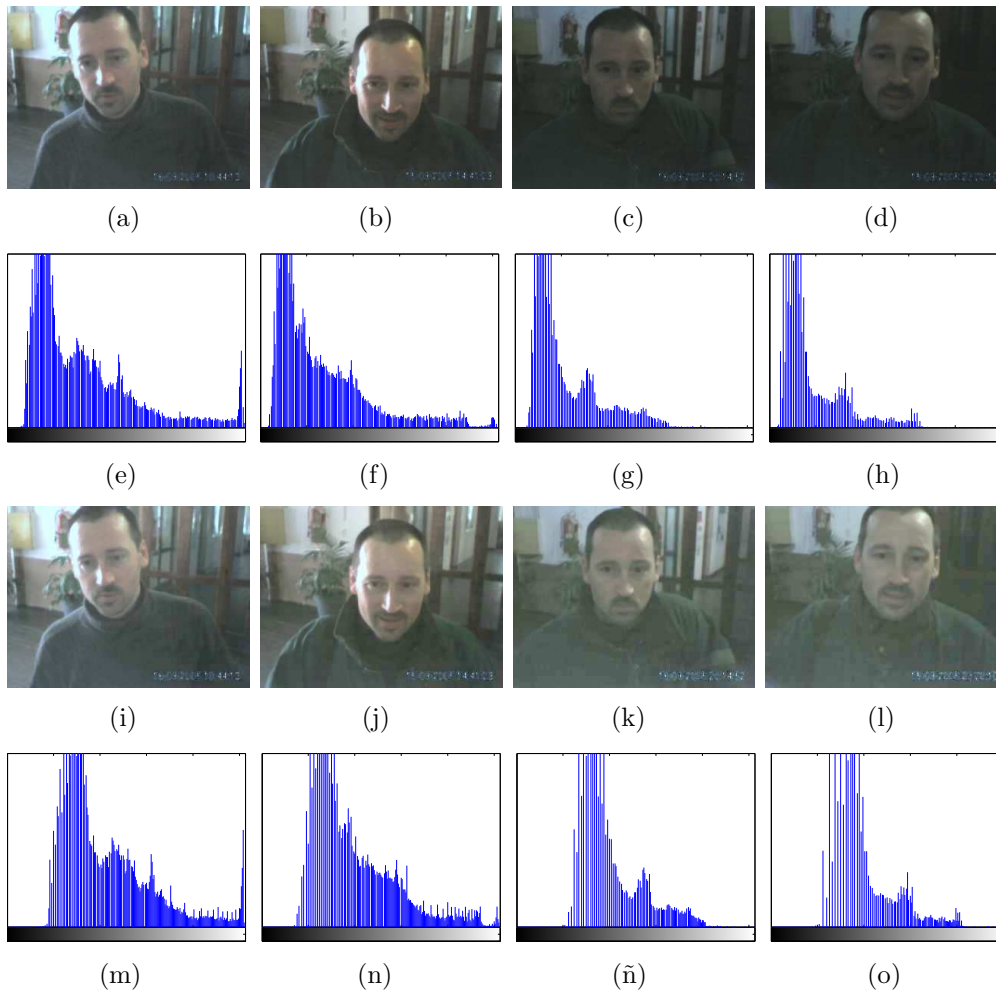


Figura 5.1: Imágenes capturadas a las: a) 10:44 hs., b) 14:41 hs., c) 20:14 hs., d) 23:20 hs., e-h) Histogramas respectivos. i-o) Imágenes con luz normalizada y sus correspondientes histogramas

| <i>Hora(hs.)</i> | μ_0 | γ calculado | μ_f |
|------------------|---------|--------------------|---------|
| 10:44 | 0,3 | 0,66 | 0,43 |
| 14:41 | 0,24 | 0,59 | 0,4 |
| 20:14 | 0,18 | 0,53 | 0,39 |
| 23:20 | 0,16 | 0,49 | 0,38 |

Tabla 5.1: Valores correspondientes a la Figura 5.1

5.3.2 Experimento 2: Detección de ojos

El sistema informático de registro de la FI-UNER trabaja con una población de alrededor 250 personas, por lo que es de esperar que se presente algún usuario que lleve anteojos, inclusive una misma persona puede usarlo en ciertas ocasiones y en otras no. Por lo tanto, el sistema debe ser capaz de resolver correctamente esta situación al momento de detectar los ojos durante el proceso de normalización de la geometría. Para comprobarlo, se toman imágenes de diferentes personas con anteojos y se evalúa si la detección de los ojos funciona o no en cada caso.

En la Figura 5.2 se muestran las imágenes faciales de las personas que se toman como entrada en la normalización y todo el proceso posterior hasta la detección de los ojos. Como se puede observar, en el primer caso no se logra detectar los ojos y sí en el resto.

5.3.3 Experimento 3: Tasa de reconocimiento

En este primer experimento se analiza el rendimiento del reconocedor tomando un conjunto de 15 personas con 16 o más imágenes cada una, de las cuales se seleccionan aleatoriamente para cada sujeto 13 imágenes para el entrenamiento y 3 para prueba.

El número de eigenfaces para estos ensayos se fija según el porcentaje de variabilidad de los datos que se desea representar. En la literatura, generalmente se opta por aplicar el test SCREE [16], término que se define como analogía al escombro en el fondo de un acantilado, es decir, las componentes principales usadas son el acantilado y el resto los escombros. Esta técnica que analiza una gráfica como la presentada en la Figura 5.3 donde se observa que con 13 eigenfaces de un total de 195, es decir un 6,67 %, se representa más del 80 % de la varianza acumulada. Notar que las eigenfaces siguientes se

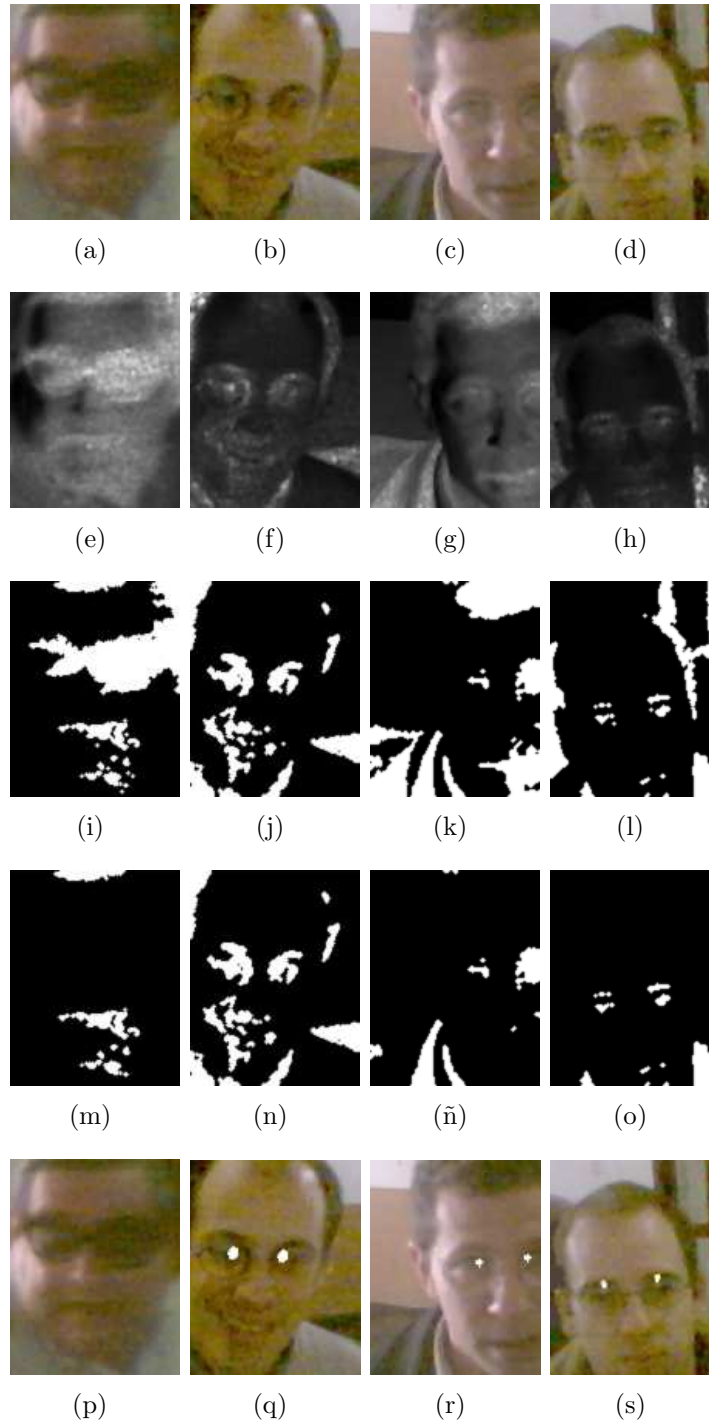


Figura 5.2: a-d) Imagen facial. e-h) Mapa. i-l) Mapa binarizado. m-o) Filtrado de regiones. p-s) Ojos detectados en la imagen facial.

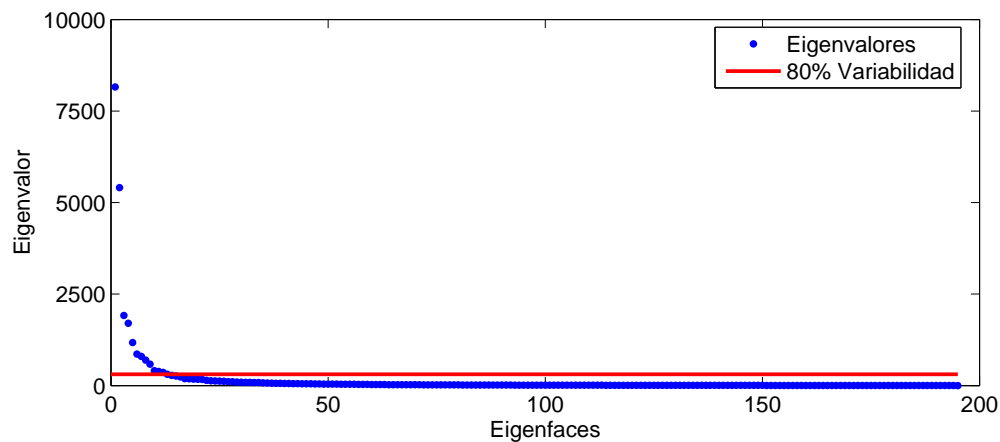


Figura 5.3: SCREE Test

mantienen cercanas en una estrecha línea, por lo que carecen de relevancia. En la Tabla 5.2 se detallan los eigenvalores, el porcentaje de la varianza y de la varianza acumulada por cada uno.

Luego se efectúan las pruebas de clasificación y para diferentes umbrales θ_ϵ inclusive $\theta_\epsilon = \infty$ donde no se rechaza ninguna persona, calculando la tasa de reconocimiento y de rechazo. En todos los casos, las imágenes de prueba son clasificadas como uno de los 15 individuos conocidos.

Los resultados se exponen en las Tablas 5.3 y 5.4. Los porcentajes de rechazos se contabilizan sobre el total de las imágenes de prueba y los de reconocimiento sobre el total de las imágenes no rechazadas. En la Figura 5.4 se observa la variación del porcentaje de reconocimiento y de rechazo con respecto al umbral para diferentes valores de k .

5.3.4 Experimento 4: Sensibilidad al ruido y oclusión

Dada una imagen del conjunto de entrenamiento, se la distorsiona de 8 formas diferentes mediante la adición de ruido gaussiano, desenfoque aplicando un filtro de promediado, la oclusión de características faciales claves, etc. (Figura 5.5). Se prueba el reconocimiento de estas nuevas imágenes y se observa que todas fueron reconocidas a excepción de la imagen 5.5h.

| <i>Eigenface</i> Nº | <i>Eigenvalor</i> | <i>Varianza</i> (%) | <i>Varianza acumulada</i> (%) |
|---------------------|-------------------|---------------------|-------------------------------|
| 1 | 8159,69 | 28,94 | 28,94 |
| 2 | 5410,07 | 19,19 | 48,13 |
| 3 | 1913,04 | 6,79 | 54,92 |
| 4 | 1702,02 | 6,04 | 60,96 |
| 5 | 1173,84 | 4,16 | 65,12 |
| 6 | 860,76 | 3,05 | 68,17 |
| 7 | 798,5 | 2,83 | 71,01 |
| 8 | 693,24 | 2,46 | 73,47 |
| 9 | 583,03 | 2,07 | 75,53 |
| 10 | 404,4 | 1,43 | 76,97 |
| 11 | 384,28 | 1,36 | 78,33 |
| 12 | 360,76 | 1,28 | 79,61 |
| 13 | 306,63 | 1,09 | 80,7 |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 195 | $3,84E^{-8}$ | $1,36E^{-10}$ | 100,0 |

Tabla 5.2: Rendimiento del algoritmo de reconocimiento

5.4 Análisis de resultados y discusión

Del experimento 1 se desprende que la normalización logra un correcto desempeño al dejar las imágenes con un brillo general similar. En la Figura 5.1a-d se observa con claridad las diferencias de iluminación que hay en las imágenes a partir de los horarios en que fueron tomadas. Esto también se ve reflejado en los histogramas 5.1e-h, donde se observan picos en valores oscuros (similares en las cuatro imágenes), pero las imágenes con mayor iluminación tienden a utilizar todo el rango de la escala de grises, mientras las más oscuras concentran sus grises en los valores más bajos del rango. Aplicando la transformación de potencia se busca que todas las imágenes posean un brillo general, una distribución y desplazamiento del histograma similar, de manera tal que se facilite el trabajo de localización del rostro. Las Figuras 5.1m-o demuestran que los histogramas adquieren una forma similar, desplazando sus picos hacia valores más altos y ampliando las concentraciones de oscuros en las imágenes con brillo bajo. En la Tabla 5.1 se ve que las medias tienden a igualarse luego de la transformación, obteniendo el resultado deseado.

| θ_ϵ | $k = 1$ | | $k = 3$ | | $k = 5$ | | $k = 7$ | |
|-------------------|---------|-------|---------|-------|---------|-------|---------|-------|
| | Recon. | Rech. | Recon. | Rech. | Recon. | Rech. | Recon. | Rech. |
| ∞ | 75,33 | 0 | 70,05 | 0 | 64,22 | 0 | 60,22 | 0 |
| 7000 | 77,14 | 4,67 | 71,85 | 5,11 | 68,97 | 5,11 | 67,71 | 5,11 |
| 6000 | 78,24 | 6,0 | 73,84 | 6,67 | 72,03 | 6,67 | 71,10 | 6,67 |
| 5000 | 79,78 | 8,67 | 75,68 | 9,56 | 73,57 | 9,56 | 72,88 | 9,56 |
| 4000 | 82,18 | 12,44 | 78,29 | 12,89 | 77,17 | 12,89 | 77,20 | 12,89 |
| 3000 | 85,69 | 19,78 | 81,95 | 20,67 | 81,9 | 20,67 | 81,19 | 20,67 |

Tabla 5.3: Tasas de reconocimiento y rechazo (en %) para k -vecinos más cercanos.

| θ_ϵ | Recon. | Rech. |
|-------------------|--------|-------|
| ∞ | 42,44 | 0 |
| 26000 | 45,91 | 15,11 |
| 24000 | 45,97 | 15,11 |
| 22000 | 45,91 | 15,11 |
| 20000 | 46,19 | 14,89 |
| 18000 | 46,17 | 16,44 |

Tabla 5.4: Ttasa de reconocimiento (en %) para clase más cercana

El segundo experimento muestra cómo el sistema resuelve satisfactoriamente en la mayoría de los casos el problema de la detección de ojos en rostros con anteojos. No obstante, se nota que en los primeros tres casos al binarizar el mapa, el antejo hace que tanto los ojos como las cejas sean parte de una misma región, lo que resta precisión al método de detección. En general este problema se repite en otras imágenes, ya que el mapa tiende a remarcar las zonas más oscuras del rostro como los ojos, cejas y otras características faciales como la boca y cabello. Estas últimas características son descartadas al no obtener grandes ponderaciones en la elección de los ojos; sin embargo, las cejas se confunden fácilmente con los ojos por su cercanía y similar aspecto. En particular, en la primer imagen el color y el grosor de los anteojos hacen que el mapa resalte toda la sección del cristal y marco, produciendo una única región en la binarización que imposibilita la detección correcta de los ojos. Si bien la situación ideal es la correcta detección de los ojos que permita una normalización precisa, cuando esto no ocurre, la mejor alternativa es dejar la imagen facial obtenida de la segmentación, ya que si se detectan falsos ojos la normalización resulta errónea.

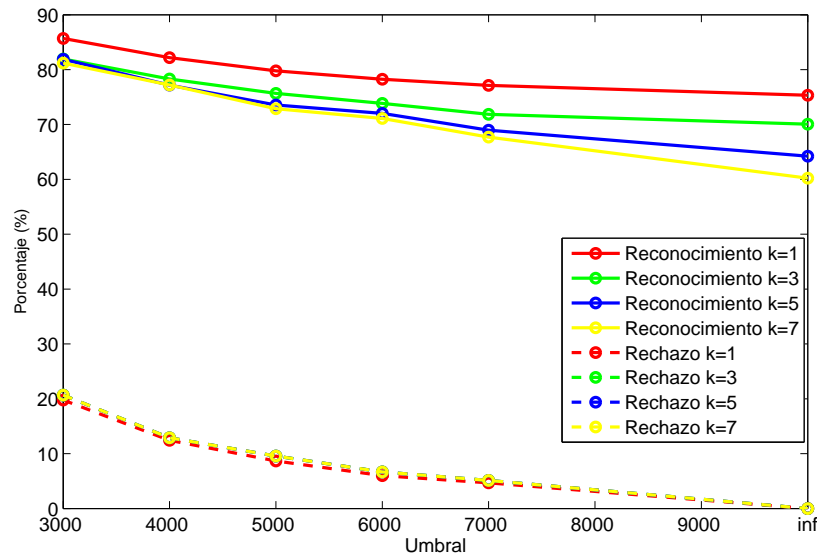


Figura 5.4: Tasa de reconocimiento y rechazo (en %) vs. umbral θ_ϵ para k -vecinos

Con respecto al reconocimiento, la clasificación por mínima distancia a la clase produce gran cantidad de errores debido a la dispersión de los miembros de cada clase, lo que hace difícil representar la clase con un único punto en el eigenspace. Esto se observa claramente en la Figura 5.6 para las personas etiquetadas con las clase 2 y 13, donde debido a la dispersión de los puntos, muchos de éstos pertenecen a la clase 13 (rojo) y se encuentran más próximos al centro de la clase 2.

Los métodos de clasificación basados en k -vecinos logran mejores desempeños. En este caso, como cada clase posee varios puntos de representación,

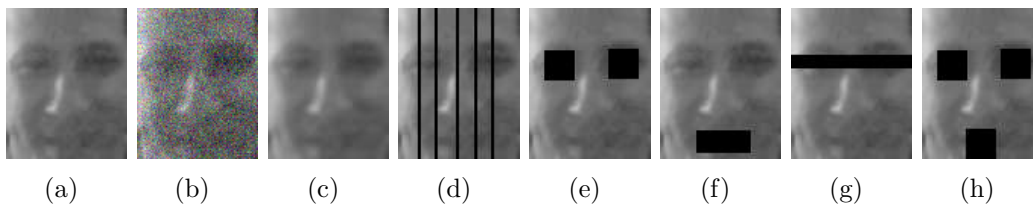


Figura 5.5: Distorsión: a) Imagen original, b) con ruido gaussiano (media 0, varianza 0,01), c) con desenfoque, d-h) con oclusiones parciales.

una nueva cara con características similares a las utilizadas en la etapa entrenamiento para esa persona caerá en proximidad de algunos de los puntos de la clase. De la Tabla 5.3 se advierte que $k = 1$ obtiene el mejor desempeño en comparación con $k > 1$. Esto se debe nuevamente a la complejidad del problema observada en la Figura 5.6, donde además de la dispersión de las clases se evidencia una escasa homogeneidad entre los individuos de algunas clases. En muchas ocasiones, esto hace que haya mayor cantidad de vecinos de una clase distinta a la del usuario ingresado. Por ejemplo, se notó en varias oportunidades que ingresado un usuario de la clase 1, el vecino más cercano es de la clase 1, pero al incrementar k los siguientes vecinos pertenecían a distintas clases, causando una falsa verificación de la identidad.

Ante la variación del umbral θ_ϵ , el reconocimiento mejora a medida que el umbral disminuye. Sin embargo, como se ve en la Figura 5.4, esto conduce a un aumento de la cantidad de rechazos ya que la mínima distancia no es lo suficientemente pequeña, suponiendo de esta manera que el o los vecinos más cercanos están fuera del radio en el cual se considera que una clase debe estar representada. De esta forma, el manejo del umbral queda a criterio del administrador, es decir, si se requiere un sistema más seguro se fija un umbral bajo que asegure un alto porcentaje de reconocimiento a costa de altos porcentajes de rechazos. En base al problema que pretende solucionar este módulo y teniendo en cuenta la aplicación del mismo, se recomienda fijar θ_ϵ en un valor alto.

Analizando detalladamente los resultados de la clasificación, se concluye que los principales problemas en el reconocimiento surgen en la etapa de normalización por: (a) escalamientos erróneos por falsa detección de ojos y (b) variación de la pose. Respecto a éste último punto, la Figura 5.6 ilustra este hecho en la dispersión en el plano de puntos correspondientes a un mismo usuario, cuando gira la cabeza a los lados sin inclinarla.

El último experimento demuestra que el método de eigenfaces utilizado reduce la sensibilidad al ruido y a oclusiones parciales (crecimiento de bigote, presencia de cicatrices, etc.), permitiendo reconocer exitosamente variaciones en el rostro de las personas.

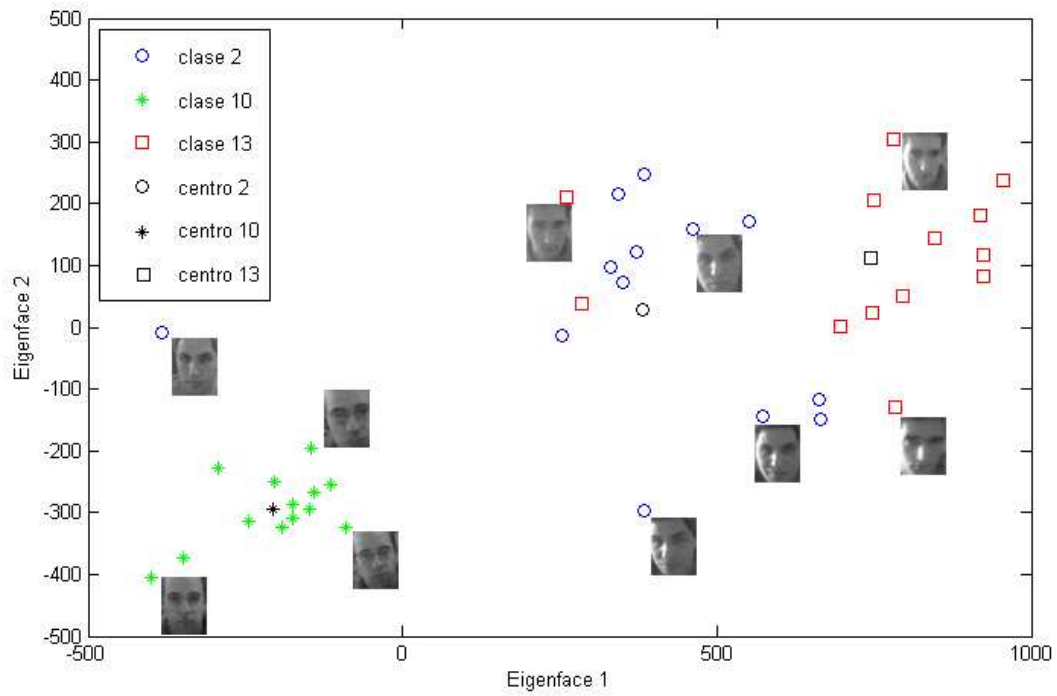


Figura 5.6: Representación de tres clases sobre las dos primeras eigenfaces.

Conclusiones

6.1 Conclusiones finales

Se desarrolló un módulo completo de reconocimiento facial que contempla todas las etapas necesarias desde la captura de la imagen en el entorno de un sistema de registro de personal, hasta la verificación final de la identidad del usuario, pasando por las etapas de preproceso, extracción de características y clasificación.

El módulo obtenido cumple con los objetivos planteados, mostrando un rendimiento aceptable para el escenario en estudio, logrando satisfacer las necesidades para las cuales se diseñó mediante pasos simples que superan la complejidad presentada por el problema a partir de la baja calidad y la variabilidad de las imágenes de entrada. La experimentación realizada muestra que el método de las eigenfaces representa adecuadamente las imágenes faciales con unos pocos valores, permitiendo una manipulación sencilla y rápida de los datos.

6.2 Desarrollos futuros

Como trabajos futuros podemos citar la integración definitiva del módulo al sistema existente, siendo necesario en este caso el desarrollo de una interfaz gráfica para la administración y manejo de usuarios.

Además, se requiere de un acondicionamiento del entorno, que redunde en una mejor captura de la imagen de entrada al módulo, permitiendo un mejor desempeño general del sistema completo. Para el reacondicionamiento de la escena, se recomienda la colocación de una luminaria blanca con disparo sincronizado a la captura (o continuo), colocación de la cámara delante del vidrio para evitar reflejos, fondo claro y homogéneo, y colocación del dispositivo de captura sobre un soporte fijo, en una posición adecuada a la altura de los usuarios.

Bibliografía

- [1] Y. Adini, Y. Moses, and S. Ullman. Face recognition: The problem of compensating for changes in illumination direction. *IEEE Transaction in Pattern Analysis and Machine Intelligence*, 19:144–148, 1997.
- [2] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, 13(6):1450–1464, 2002.
- [3] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–31074, 2003.
- [4] W. W. Bledsoe. The model method in facial recognition. *Panoramic Research Inc*, 1966.
- [5] D. Chai and K. Ngan. Locating facial region of head-and shoulders color image. *Proceedings of Automatic Face and Gesture Recognition*, 124–129, 1998.
- [6] T. Choudhury, B. Clarkson, T. Jebara, and A. Pentland. Multimodal person recognition using unconstrained audio and video. *Proceedings, International Conference on Audio- and Video-Based Person Authentication*, 176–181, 1999.
- [7] A. Cockburn. *Writing Effective Use Cases*. Addison - Wesley, 2000.

- [8] J. Daugman. Face and gesture recognition: Overview. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):675–676, 1997.
- [9] I. Gauthier, M. Behrmann, and M. J. Tarr. Can face recognition really be dissociated from object recognition? *Journal of Cognitive Neuroscience*, 11:349–370, 1999.
- [10] A. Goldstein, L. Harmon, and A. Lesk. Identification of human faces. *Proceedings of the IEEE*, 59(5):748–760, 1971.
- [11] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall, New Jersey, second edition, 2002.
- [12] S. Grossman. *Algebra Lineal*. McGraw-Hill, México, 5 edition, 1996.
- [13] B. Heisele, T. Serre, M. Pontil, and T. Poggio. Component-based face detection. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:I–657–I–662, 2001.
- [14] E. Hjelmås. Face detection: A survey. *Computer Vision and Image Understanding*, 83:236–274, 2001.
- [15] R. L. Hsu, M. Abdel-Mottaleb, and A. K. Jain. Face detection in color images. *IEEE Transaction in Pattern Analysis and Machine Intelligence*, 24(5):696–706, 2002.
- [16] J. E. Jackson. *A User's Guide To Principal Components*. John Wiley and Sons, New York, 1991.
- [17] A. Jacquin and A. Eleftheriadis. Automatic location tracking of faces and facial features in video signal. *Proceedings of International Workshop on Automatic Face and Gesture Recognition*, 142–147, 1995.
- [18] T. Jebara, k. Russell, and A. Pentland. Mixtures of eigenfeatures for real-time structure from texture. *Sixth International Conference on Computer Vision*, 128–135, 1998.
- [19] R. Kumar, S. Raja, and A. G. Ramakrishnan. Eye detection using color cues and projection functions. *IEEE International Conference on Image Processing 2002*, 3:III–337– III–340, 2002.
- [20] A. Lanitis, C. Taylor, and T. Cootes. An automatic face identification system using flexible appearance models. *Image and Vision Computing*, 13:393–401, 1995.

- [21] Y. Li, S. Gong, and H. Liddell. Modelling faces dynamically across views and over time. *Eighth International Conference on Computer Vision (ICCV'01)*, 01:554–559, 2001.
- [22] C. Liu and H. Wechsler. Evolutionary pursuit and its application to face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(6):570–582, 2000.
- [23] H. Moon and P. J. Philips. Computational and performance aspects of PCA-based face-recognition algorithms. *Perception*, 30(3):303–321, 2001.
- [24] A. Neffian and M. Hayes III. Hidden Markov models for face recognition. *International Conference on Acoustics, Speech, and Signal Processing*, 5:2721–2724, 1998.
- [25] R. Paredes, J. C. Pérez, A. Juan, and E. Vidal. Local representations and a direct voting scheme for face recognition. *Proceedings of the 1st International Workshop on Pattern Recognition in Information Systems, ICEIS Press*, 71–79, 2001.
- [26] P. Peer, J. Kovac, and F. Solina. Human skin color clustering for face detection. *Computer as a Tool, EUROCON 2003*, 2(7):721–732, 1997.
- [27] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 84–91, 1994.
- [28] S. R. Pressman. *Ingeniería del software. Un enfoque práctico*. Mc Graw Hill, 5 edition, 1997.
- [29] A Samal and P. A. Iyengar. Human face detection using silhouettes. *International Journal of Pattern Recognition and Artificial Intelligence*, 9(6):845–867, 1995.
- [30] H. Scheniderman and T. Kanade. Probabilistic modeling of local appearance and spatial relationships for object recognition. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 45–51, 1998.
- [31] J. Schmuller. *Aprendiendo UML en 24 horas*. Prentice Hall, 2004.
- [32] A. Shio and J. Sklansky. Segmentation of people in motion. *Proceedings of the IEEE Workshop on Visual Motion*, 325–332, 1991.

- [33] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
- [34] M. Turk and A. Pentland. Face recognition using eigenfaces. *IEEE Conference on Computer Vision and Pattern Recognition*, 1991.
- [35] B. Weyrauch, B. Heisele, J. Huang, and V. Blanz. Component-based face recognition with 3d morphable models. *Conference on Computer Vision and Pattern Recognition Workshop*, 85–85, 2004.
- [36] L. Wiskott, J-M. Fellous, N. Kruger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):776–779, 1997.
- [37] W. Zhao, R. Chellappa, P. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, 2003.

Corpus

A.1 BADAFFI

BADAFFI (BAse de DATos Faciales de la FI-UNER) es un conjunto de imágenes faciales del personal de la Facultad de Ingeniería de la Universidad Nacional de Entre Ríos, conformada especialmente para el desarrollo del presente proyecto final de carrera. Las imágenes se tomaron en forma automática entre las 6 y las 23 hs. durante el mes de septiembre de 2005 y el mes de agosto de 2006. En total se obtuvieron 1104 fotos de 84 personas con fondo, luz, distancia, pose y expresión totalmente natural, de las cuales se seleccionaron 457 de 27 personas con variaciones aceptables de pose, iluminación y calidad. Dado que en la selección realizada existían personas con escasa cantidad de fotos, se amplió el conjunto mediante espejado, logrando una extensión válida del conjunto original. De esta manera, se obtuvo un mínimo de 16 fotos por persona.

Cada imagen se almacenó en formato JPEG con una tasa de compresión de 40,1, con una resolución de 320×240 píxeles. La nomenclatura del etiquetado es de la forma ‘dddddddYYYYMMDDhhmmss.ext’, donde dddddddd es el DNI ingresado, YYYY es el año, MM el mes, DD el día, hh la hora, mm los minutos, ss los segundos y ext la extensión, en este caso, jpg. Por ejemplo, una imagen con el nombre ‘2920054820050912155518.jpg’ indica que el usuario con DNI 29200548 se registró en el sistema el 12/09/2005 a las 15 hs. 55 min. 18 seg.

En este apartado se exhibe un subconjunto de 30 imágenes originales en color seleccionadas arbitrariamente de BADAFFI y las imágenes faciales de cada una en escala de grises, obtenidas automáticamente por el bloque de preprocesamiento.

Es posible visualizar también algunos de los problemas introducidos por las condiciones de la escena: en las dos últimas se evidencia el reflejo del monitor en la foto y se observa que el machimbre de la pared en la parte izquierda de ambas, tiene color símil piel. Además, en varias fotos es visible la luz natural, que incide lateralmente en el rostro.





Librería de Procesamiento de Imágenes Digitales

B.1 CImg

La librería CImg¹ creada por David Tschumperlé, investigador del grupo IMAGE del laboratorio GREYC en Caen (Francia), perteneciente al instituto CNRS (del francés, *Centre national de la recherche scientifique*), y varios contribuyentes (H-E. Assemlal, J. Boulanger, E. Fausett, S. Fourey, y otros), es un conjunto de herramientas de código abierto en C++ destinado al procesamiento de imágenes. Ha sido diseñada principalmente para enseñanza y actividades de investigación. Es altamente portable y se puede utilizar en diferentes sistemas operativos (Unix/X11, Windows, MacOS X, FreeBSD) con diferentes compiladores de C++.

Consiste sólo de un único archivo de cabecera CImg.h que debe ser incluido en el código fuente del programa en C++. Se puede compilar utilizando solamente librerías estándar de C++, ya que no necesita de otras librerías y dependencias complejas.

CImg contiene funciones para cargar/guardar/visualizar imágenes, y algoritmos de procesamiento para redimensionar/filtrar/transformar imágenes, dibujar objetos (texto, líneas, lados, curvas, elipses, objetos 3D), entre otras. Es eficiente y fácil de utilizar, similar a una STL para procesamiento de imágenes, con numerosos ejemplos.

¹<http://cimg.sourceforge.net>

La librería CImg es distribuída bajo dos licencias distintas: la librería propiamente dicha es regida por la licencia CeCILL-C (similar a LGPL), mientras que los ejemplos de la librería son regidos por la licencia CeCILL License (similar a GPL). Ambas son licencias de código abierto. El código fuente ha sido registrado al APP (French Agency for the Protection of Programs) por el INRIA, Francia.