UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

FACULTAD DE CIENCIAS

RECONOCIMIENTO POR BIOMETRÍA FACIAL PARA APLICACIONES EN CIUDADES INTELIGENTES

TESIS

PARA LA OBTENER EL TÍTULO DE:
FÍSICO

PRESENTA:
MARTÍN ADRIÁN GARDUÑO SANTANA

ASESOR:
DR. LUIS ENRIQUE DÍAZ SÁNCHEZ
COASESOR:
DR. ISRAEL TABAREZ PAZ

El Cerrillo Piedras Blancas, Toluca, Estado de México. Enero 2018

Resumen

Este trabajo se presenta dentro del contexto de las ciudades inteligentes, pues la implementación de las mismas se ha convertido en un objetivo por parte de las instituciones mexicanas, incluida la Universidad Autónoma del Estado de México, donde la Facultad de Ciencias se encuentra involucrada a través de la creación del Centro de Datos[1] y su participación dentro del programa Fiware-México[2].

En esta búsqueda de crear tecnología para su implementación en las ciudades inteligentes se presenta un sistema de reconocimiento facial basado en el algoritmo Eigenfaces para aplicaciones potenciales de seguridad y control de acceso.

Dentro del trabajo de investigación documental se da una visión del concepto de ciudades inteligentes y se introduce de manera general la plataforma Fiware, cuyo objetivo es la implementación de aplicaciones en las ciudades y transformarlas en “smartcities".

El desarrollo del sistema de reconocimiento por biometría facial se hace mediante lenguaje de programación Python, y se ha evaluado usando la base de datos generada por el programa Face Recognition Grand Challenge[3]. Mediante procesamiento de imágenes se generan dos conjuntos de datos: imágenes de rostro completo e imágenes de la región central del rostro. Este último conjunto de imágenes se ha reportado en la literatura con más robustez a variaciones en cambios de expresión[4]. Para determinar la identidad de los sujetos, se utilizan las métricas Euclidiana y Mahalanobis con las que se halla la distancia mínima en el espacio multidimensional también llamado espacio de rostros generado por el algoritmo Eigenfaces[5].

[1] Ver [16]
[2] Ver [26]
[3] Programa Face Recognition Grand Challenge [48]
[4] Ver [14]
[5] Ver [63]

Los resultados obtenidos son satisfactorios al usar la métrica Euclidiana, ya que se acercan a resultados reportados en trabajos anteriores. Se ha registrado una tasa del 75 % en tres de las pruebas realizadas en esta tesis. Por el otro lado, usando la métrica de Mahalanobis se logra una tasa de reconocimiento del 17.8 %, siendo un resultado no esperado en este caso.

El sistema de reconocimiento facial usando la métrica Euclidiana se incluye en esta tesis, con la intención de que más alumnos de la Facultad de Ciencias y de la Universidad Autónoma del Estado de México se interesen en la generación de tecnología de este tipo y sus múltiples aplicaciones.

Además, se presenta un sistema de detección y medición del tamaño de nanopartículas basado en el procesamiento de imágenes, con el objetivo de aplicar los conocimientos adquiridos en el área de la Física.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

El ser humano siempre ha tratado de facilitar la manera en la que vive, los físicos nombran a esto el principio de mínima energía, dicho de otra manera, se busca emplear la menor cantidad de energía para realizar un trabajo. En materia tecnológica, este concepto es utilizado en los sistemas inteligentes, automatización, aplicaciones móviles, etc; más aún, se busca este principio con la creación de ciudades inteligentes.

El principio de una ciudad inteligente es la captación de información mediante dispositivos o sensores para su análisis, modelamiento y posteriormente tomar decisiones para un desarrollo en todos los ámbitos de la ciudad. Una plataforma que toma en cuenta este principio, fue desarrollada por la Unión Europea, denominada Fiware [25]. Esta plataforma sirve para implementar sensores, hacer cómputo en la nube, y generar aplicaciones para creación de ciudades inteligentes. Por otra parte, en nuestra universidad se encuentra en desarrollo el Centro de Datos ubicado en la Facultad de Ciencias con el objetivo de tener una infraestructura con estándares de calidad internacional ofreciendo alta conectividad y ancho de banda a universitarios y organizaciones, gestión de datos en tiempo real, alertas y procesamiento de información, además de la creación de aplicaciones que permiten la recopilación y el procesamiento de datos [16].

Sin embargo, debido a la gran cantidad de datos e información personal que circula, algunos delitos como robo de identidad o acceso a información privada sin autorización se han vuelto un problema, por lo que el desarrollo de sistemas de seguridad que no violen la privacidad y al mismo tiempo sean efectivos es vital. Una alternativa que cubre esta situación son los sistemas biométricos.

Los métodos biométricos realizan la identificación a través de características fisiológicas o sociales del ser humano como el rostro, huellas dactilares, venas, iris, retina, voz, etc; o en rasgos de comportamiento como el movimiento ondulatorio del paso, dinámica de firma y de pulsación de teclas entre otros [1], [4]. La identificación o verificación de una persona se realiza comparando una o más de estas biometrías con una base de datos de las mismas previamente almacenadas, en lugar de autenticar la identidad o garantizar acceso mediante métodos tradicionales como contraseñas, pines, tarjetas, tokens y llaves las cuales conllevan algunos riesgos, tales como pérdidas, olvido de contraseñas, clonación y/o robo.

Además, los sistemas biométricos se han popularizado con la introducción de nuevas tecnologías y el creciente desarrollo de algoritmos, el claro ejemplo son los teléfonos inteligentes o smartphones, cuyos terminales más recientes cuentan con sensores para aplicaciones biométricas [6].

En especial el estudio de reconocimiento facial se ha dado por presentar ciertas ventajas sobre otras biometrías. Una de las principales ventajas es que puede ser completamente no intrusivo, es decir, que el reconocimiento facial puede usarse sin participación o conocimiento del usuario debido a que las imágenes son obtenidas desde una cámara a distancia, mientras que la mayoría de los métodos biométricos requieren la participación de los sujetos para ser identificados. Por ejemplo, para la detección de iris o retina, es necesario situar el ojo a cierta distancia de un sensor para su lectura [37].

Otro beneficio es la adquisición de datos, los sistemas de reconocimiento facial pueden adquirir las imágenes a través de dispositivos de bajo costo, como cámaras de vídeo, cámaras digitales (incluyendo la cámara de los teléfonos inteligentes), contrario a otros sistemas como reconocimiento de iris que requiere de sensores costosos. Por otro lado, otros sistemas como reconocimiento a través de huella dactilar o geometría de la mano son susceptibles a cambios o variaciones como daños en la piel o manos mojadas; así como el reconocimiento de voz es susceptible a ruidos de fondo o fluctuaciones en el ambiente. Por otra parte, los sistemas que usan el mismo equipo para capturar las características biométricas de los sujetos los exponen a la transmisión de enfermedades. En este sentido, la biometría facial es completamente no intrusiva y no conlleva riesgos para la salud [37].

Debido a esta ventajas, el reconocimiento por biometría facial es una de las opciones para afrontar problemas actuales de la sociedad. Uno de los principales problemas

con los que se lidia actualmente es el robo de identidad. En 2017, en México se observa un incremento en los robos de identidad mediante la clonación o hurto de identificaciones oficiales [64]. De la misma manera, la usurpación o extravío de tarjetas bancarias ha llevado a los bancos a mirar nuevas formas de identificación de los usuarios. A partir de 2018, en el uso de cajeros automáticos o al realizar un trámite en ventanilla se podrá realizar la verificación de los clientes mediante la biometría facial; más aún, con los cambios a la Ley de Instituciones de Crédito, publicada en el Diario Oficial de la Federación el 29 de agosto de 2017, todas las instituciones bancarias se verán comprometidas a realizar una base de datos biométrica de los clientes para identificarlos, incluyendo biometría facial [19].

A pesar de los avances registrados en los sistemas de reconocimiento facial, este continúa siendo un reto debido a las múltiples variaciones del objeto de estudio y el entorno en el que se encuentra, mismos que afectan el rendimiento de los algoritmos. Entre las principales variaciones están los cambios en la posición del rostro, variaciones en la iluminación, expresiones faciales, oclusiones debido a la presencia de vello facial o accesorios (lentes, bufandas, etc.), así como el envejecimiento del rostro [69]. Debido a estos retos y al campo de aplicaciones creciente, es necesario continuar con la investigación de estos sistemas, además de tomar en cuenta la importancia en la creación de tecnología propia para desarrollo científico y tecnológico del país.

Objetivo General

Implementar un sistema de reconocimiento por biometría facial para su uso en ciudades inteligentes.

Alcances

Se logra el desarrollo de un sistema de reconocimiento facial, que puede ser implementado en aplicaciones de seguridad y/o control de acceso con un a eficiencia del 75 %.
Además, con base en el procesamiento de imágenes se ha desarrollado un sistema de detección y medición del tamaño de nanopartículas, que es de suma importancia para la caracterización de sus propiedades magnéticas.

Organización de la tesis

La organización de esta tesis está dada de la siguiente manera:

En el Capítulo 2 se presentan las generalidades de este trabajo, introduciendo el concepto de ciudades inteligentes y de la plataforma Fiware, además de presentar los trabajos más recientes que involucran una aplicación directa en ciudades de este tipo. Continúa con los principios generales de un sistema de reconocimiento, seguido del estado del arte y termina con los conceptos matemáticos involucrados en el algoritmo que se va a utilizar llamado Eigenfaces.

El Capítulo 3 describe el desarrollo experimental para la creación del sistema de reconocimiento facial, comenzando con el escenario de experimentación y el procesamiento de imágenes. Se explican el procedimiento seguido para la implementación del algoritmo Eigenfaces y la generación del espacio de rostros. Por último, se presenta la evaluación del sistema y los resultados obtenidos.
El capítulo 4 muestra las conclusiones y el trabajo futuro de esta tesis.

Además, al final de este trabajo se incluyen dos anexos. El Anexo A muestra el código desarrollado para el sistema de reconocimiento facial en lenguaje Python, y el Anexo B muestra el sistema de detección y medición de tamaño de nanopartículas creado con base en el procesamiento de imágenes.

Capítulo 2

Generalidades

Este Capítulo comienza con la descripción de las ciudades inteligentes introduciendo un concepto general y los factores principales que involucran su creación, y con ello los esfuerzos que actualmente se están realizando en México, en especial en nuestro estado y con la participación de la Universidad Autónoma del Estado de México en la plataforma Fiware. En seguida se muestran las investigaciones más recientes acerca de biometría facial cuya aplicación se encuentra dentro del marco de las ciudades inteligentes. La siguiente sección muestra los aspectos necesarios para la creación de un sistema de reconocimiento facial, incluyendo los programas internacionales que han surgido para impulsar el desarrollo de estos sistemas. Por último, se muestra el estado del arte del reconocimiento por biometría facial.

2.1. Ciudades Inteligentes

Existen diversos conceptos para definir a las ciudades inteligentes, todos ellos involucran un mejor desarrollo de la economía, gobierno, ambiente, transporte, seguridad y calidad de vida, mediante el uso de la tecnología para interconectar infraestructura, sobre todo mediante las tecnologías de información y comunicación (TICs) y el internet de las cosas (IoT, por sus siglas en inglés) [68], [54]. En [13] se da una definición con base en los conceptos antes mencionados, ellos mencionan que una ciudad es inteligente cuando las inversiones en capital humano y social, en la infraestructura de comunicación moderna (TICs) y tradicional (transporte), alimentan el crecimiento económico sostenible y otorgan alta calidad de vida, con una gestión inteligente de los recursos naturales, a través del gobierno participativo.

El concepto de ciudades inteligentes visto desde la perspectiva de las tecnologías se centra en los últimos avances en informática móvil, la computación generalizada, redes inalámbricas y demás tecnologías a medida que se integran en los espacios físicos de las ciudades. El énfasis en los dispositivos inteligentes representa una característica distintiva de las ciudades inteligentes [54].

Por otra parte, existen ocho factores principales por los cuales se busca la creación de ciudades inteligentes alrededor del mundo, que son: gestión y organización, tecnología, gobierno, política, sociedad, economía, infraestructura y medio ambiente [15]. Aunque la primera tarea que las ciudades deben abordar al convertirse en inteligentes es crear un entorno rico en redes de banda ancha que admitan aplicaciones digitales, y esto incluye tres metas principales: (1) el desarrollo de una infraestructura de banda ancha que combina cable, fibra óptica y redes inalámbricas, ofreciendo alta conectividad y ancho de banda a ciudadanos y organizaciones, (2) el enriquecimiento del espacio físico y las infraestructuras de ciudades con sistemas incorporados, dispositivos inteligentes, sensores y actuadores, ofreciendo tiempo real gestión de datos, alertas y procesamiento de información, y (3) la creación de aplicaciones que permiten la recopilación y el procesamiento de datos [54].

Figura 2.1: Centro de Datos de la UAEMéx [16].

Como parte de esto, la Universidad Autónoma del Estado de México ha puesto en marcha la creación del Centro de Datos en la Facultad de Ciencias (Ver Figura 2.1), con la cual se busca tener una infraestructura con certificados de estándares de calidad internacionales y cubrir las tres metas antes mencionadas. Este proyecto pretende poner a la ciudad de Toluca como una de las primeras ciudades inteligentes de la república, pues plantea la interconexión de la ciudad y de las empresas ubicadas en el corredor industrial [16]. De igual manera este trabajo cae dentro de las metas principales, al desarrollar un sistema de reconocimiento facial con la recopilación y

procesamiento de imágenes que pueden servir para posibles aplicaciones en seguridad y de control de accesos.

Otra parte fundamental para el desarrollo de las ciudades inteligentes en México es la inclusión de la plataforma Fiware. Fiware es una plataforma creada por la Unión Europea y un conjunto de empresas tecnológicas públicas y privadas, en el marco del Programa Internet del Futuro [36].

Fiware es una infraestructura abierta cuyo objetivo es crear y desplegar de manera rentable, aplicaciones y servicios basados en la conexión IoT, el almacenamiento, acceso, procesamiento, publicación y análisis de datos a gran escala, además el desarrollo de interfaces de usuario avanzadas con capacidades 3D y de realidad aumentada, con el fin de desarrollar infraestructuras inteligentes y su implementación en las ciudades [25].

La estructura de Fiware está basada en módulos con distintos objetivos que van desde la creación de aplicaciones y prestación de servicios, hasta el apoyo a pequeñas y medianas empresas y la expansión en todos los continentes.

La plataforma Fiware, contiene el Catálogo Fiware, que proporciona un conjunto de interfaces de programación de aplicaciones (APIs) públicas y libres que facilitan el desarrollo de aplicaciones inteligentes. La idea para el desarrollo de aplicaciones se basa en los bloques de Lego, en el catálogo existen los "Generic Enablers", que son programas de código abierto que sirven como bloques para la creación de aplicaciones, gestionando los recursos básicos de computación, de red y de almacenamiento o que sirven como herramientas para el análisis de gran cantidad de datos conocido como "Big Data", Existen también bloques para la integración, que proporcionan elementos para integrar aplicaciones, permitir su publicación, su venta, entre otras cosas. Otros permiten el manejo del Internet de las Cosas mediante el acceso a la red de comunicaciones y control de terminales y por último se tienen bloques para proporcionar seguridad y privacidad a las aplicaciones.

Fiware Lab ó FI-LAB es una instancia gratuita de la plataforma Fiware que está disponible para la experimentación y la prueba de las tecnologías Fiware. Los investigadores pueden realizar experimentos, desarrollar aplicaciones y comenzar a ampliar el ecosistema mediante el uso de los distintos módulos de Fiware, explotando datos abiertos publicados por ciudades y otras organizaciones.

Fiware Ops es el módulo que se utiliza para construir, operar y ampliar FI-LAB, es una colección de herramientas para facilitar el despliegue, configuración y operación de instancias por los proveedores de la plataforma. Está diseñado para ayudar a expandir la infraestructura asociada a una instancia y permite la cooperación de múltiples proveedores de la plataforma.

Fiware Accelerate tiene como objetivo promover la asimilación de tecnologías Fiware entre los desarrolladores de aplicaciones, con especial atención a empresas de nueva creación y a pequeñas y medianas empresas (PyMEs). Dentro de este programa la Unión Europea puso en marcha una campaña con una inversión de más de 80 millones de euros para apoyar a las PyMEs y las instituciones que desarrollan aplicaciones innovadoras basadas en Fiware enfocadas en ciudades inteligentes, salud, electrónica, transporte, energía y medio ambiente.

Fiware Mundus está diseñado para dar cobertura al esfuerzo de ser una plataforma usada a nivel global.

En nuestro país una red de colaboración es la encargada del proyecto Fiware-México (Ver Figura 2.2) [26], en la que la Universidad Autónoma del Estado de México se encuentra participando. Las instituciones encargadas del proyecto son el Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación (INFOTEC) y la Corporación Universitaria para el Desarrollo de Internet A.C (CUDI) [18], [46]. En esta colaboración uno de los principales objetivos es la creación de ciudades inteligentes. Este trabajo está involucrado como parte de la participación de nuestra universidad.

Figura 2.2: Logo del proyecto Fiware México [26].

Como continuación de esta Sección, se presentan algunas de las aplicaciones más recientes basadas en biometría facial, haciendo énfasis en la conexión para crear ciudades inteligentes.

Aplicaciones para implementación de ciudades inteligentes

El uso cotidiano de los dispositivos móviles se ha vuelto una herramienta y a la vez un marco de desarrollo para aplicaciones de reconocimiento facial, aunado al poder de procesamiento, almacenamiento y a los sensores integrados con los que los dispositivos actuales se presentan, y como se ha mencionado anteriormente, representan una característica de las ciudades inteligentes.

Los investigadores se han valido de esta y otras tecnologías actuales para desarrollar aplicaciones de reconocimiento en tiempo real y usarlo para el desarrollo de ciudades inteligentes. En el Cuadro 2.1 se muestran las áreas de mayor crecimiento y algunas de las aplicaciones implicadas en el desarrollo inteligente de las urbes [69].

Cuadro 2.1: Aplicaciones del reconocimiento facial en ciudades inteligentes [69].

Áreas	Aplicaciones específicas
Entretenimiento	Videojuegos, realidad virtual, Interacción humano-robot y humano-computadora.
Tarjetas Inteligentes	Licencias de conducir, ID's, pasaportes, inmigración.
Seguridad	Inicio de sesión o desbloqueo en dispositivos personales, encriptamiento de archivos, bases de datos.
Cumplimiento de la ley y vigilancia	Video-vigilancia avanzada, control de circuitos cerrados, control de accesos.

En [59], se desarrolla el sistema Cloud-Vision, el cual combina la tecnología de los teléfonos inteligentes con el computo en la nube. Este sistema usa primeramente la detección del rostro mediante el algoritmo de Viola-Jones [65] y el reconocimiento facial a través de Eigenfaces.

Otra investigación se enfoca en la autenticación activa de los usuarios en los teléfonos inteligentes para la protección de información personal. Usa la cámara frontal del dispositivo, tomando video del portador. Dichos videos son analizados y se hicieron públicos como base de datos para evaluar algoritmos similares. En su análisis comparan 9 algoritmos de reconocimiento facial con tasas de reconocimiento que varían

desde 7 % hasta 96 %. Para el algoritmo basado en eigenfaces se tienen tasas que varían desde el 22 % al 93 % de reconocimiento [24].

En [58], se realiza el diseño e implementación de un sistema de seguridad para el hogar basado en reconocimiento facial junto con monitoreo remoto para confirmar la identidad de las personas y controlar los accesos. Los algoritmos de reconocimiento son usados para reconocer a los visitantes y enviar una alerta al propietario mediante correo electrónico. El sistema se realiza con un controlador *Raspberry pi* y aplicaciones web para el control de manera remota, lo que convierte a este sistema en una aplicación de internet de las cosas.

Para combatir acciones ilegales como la reventa de boletos se ha propuesto el sistema TicketID en Japón. Es un sistema de verificación a gran escala para corroborar que el portador de un boleto es la misma persona que lo adquirió. Utiliza el sistema comercial NeoFace, cuyo desempeño es de los más altos en el en Face Recognition Vendor Test 2014. TicketID ya ha sido usado en un concierto arrojando una tasa de reconocimiento de 90 % con un proceso de verificación promedio de 6 segundos [45].

Recientemente, en 2017 el reconocimiento facial se puso en manos de todo el mundo, debido a la introducción del sistema FaceID en el smartphone Iphone X creado por la compañía norteamericana Apple.
FaceID funciona con un sistema TrueDepth compuesto por varios sensores, entre ellos una cámara de luz visible, una cámara de luz infrarroja y un proyector de 30,000 puntos infrarrojos. El modo de reconocimiento comienza detectando el rostro, una vez que confirma la presencia de una cara atenta, se proyectan y leen los puntos de infrarrojos para formar un mapa de profundidad de la cara, junto con una imagen infrarroja 2D. Esta información se usa para crear una secuencia de imágenes 2D y mapas de profundidad y proyectar un patrón aleatorio específico, mediante redes neuronales transforman estos datos en su representación matemática y compara esa representación con los datos faciales inscritos. Para entrenar la red neuronal usaron más de un billón de imágenes de comparación. Funciona con distintas oclusiones como gorras, lentes, vello facial, etc. y su aplicación va desde el desbloqueo del teléfono hasta para validar las compras por internet [6].

2.2. Sistema de Reconocimiento Facial

De manera general, el reconocimiento facial tiene dos escenarios: verificación o autenticación e identificación o reconocimiento.

La verificación es el escenario de una correspondencia uno a uno, formulado por la pregunta ¿eres quién dices ser?, donde un sujeto se presenta con cierta identidad, y el sistema de reconocimiento compara la biometría presentada con la biometría previamente almacenada de esta identidad, como se muestra en Figura 2.3 Basado en las comparaciones entre la biometría de entrada y la almacenada, el sistema puede aceptar al sujeto afirmando que es quien dice ser, o puede rechazarlo. Para tomar esta decisión se establece un límite o un umbral.

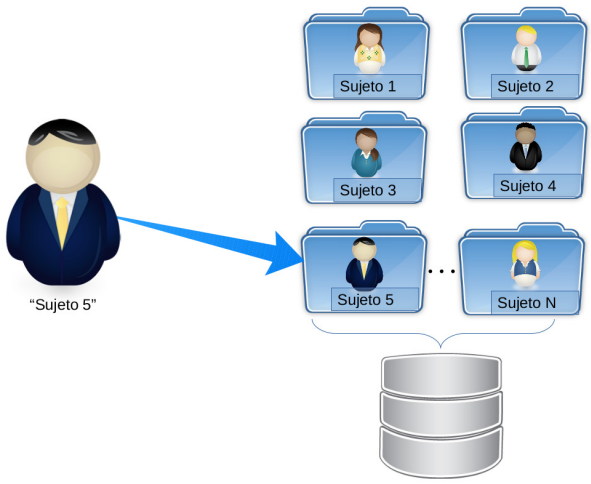


Figura 2.3: Escenario de verificación o autenticación en el reconocimiento facial.

Por otro lado, la identificación es una búsqueda de uno en muchos, iniciando con la pregunta ¿quién eres?, donde la biometría de un sujeto desconocido es comparada con todas las biometrías dentro de base de datos, de cada una de estas comparaciones se toma aquella biometría más cercana y así se asigna dicha identidad, tal como se muestra en la Figura 2.4. Si las comparaciones no alcanzan un límite establecido, el sistema puede decir que la identidad del sujeto desconocido no esta dentro de la base de datos.

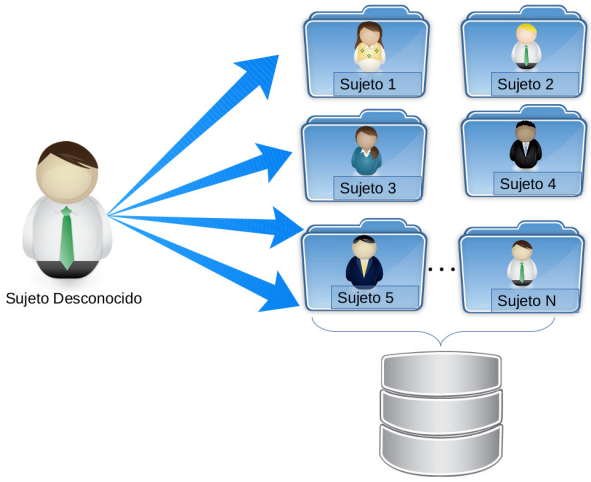


Figura 2.4: Escenario de Identificación en el reconocimiento facial.

Ciertos sistemas de identificación son capaces de aprender de manera automática, por lo que es posible identificar un rostro que antes era desconocido luego de que el sistema lo haya visto en previas ocasiones añadiéndolo a la base de datos [1], [4], [42], [69].

Existe un tercer escenario denominado "the watchlistz puede ser descrito mediante la pregunta ¿Te encuentras en una lista de identidades de alta prioridad?. Es un caso particular del escenario de identificación [42], [53].

Para evaluar el desempeño de un sistema de reconocimiento se tienen algunas variables que dependen de los rangos de aceptación o de rechazo, dichos rangos se consideran de acuerdo a ciertos parámetros, principalmente del límite o umbral establecido. Las principales variables consideradas son, Tasa de Aceptación Falsa (FAR, por sus siglas en inglés) y Tasa de Aceptación Verdadera (TAR, por sus siglas en inglés), estas tasas se definen a través de la clasificación de las pruebas mediante la llamada matriz de confusión mostrada en el Cuadro 2.2 [17], y donde cada una de sus opciones se describe a continuación:

- Real positivo (VP): pruebas que son correctamente etiquetadas como positivas.
- Falso positivo (FP): pruebas que son incorrectamente etiquetadas como positivas.

- Real negativo (VN): pruebas correctamente etiquetadas como negativas.
- Falso negativo (FN): pruebas incorrectamente etiquetadas como negativas.

Cuadro 2.2: Matriz de confusión para clasificación de pruebas en un sistema de reconocimiento [17].

	Positivo	Negativo
Etiquetado Positivo	Real Positivo VP	Falso Positivo FP
Etiquetado Negativo	Falso Negativo FN	Real Negativo VN

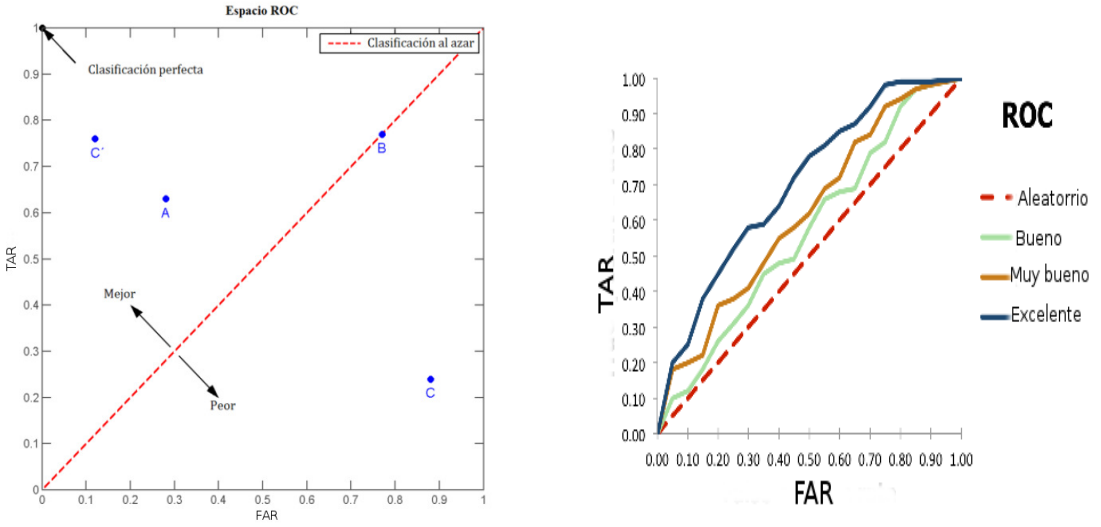
Mediante la clasificación propuesta por la matriz de confusión, se define el FAR como la fracción de pruebas negativas que son incorrectamente etiquetadas como positivas, mientras que el TAR mide la fracción de pruebas positivas que son etiquetadas correctamente.

$$FAR = \frac{FP}{FP + VN} \tag{2.1}$$

$$TAR = \frac{VP}{VP + FN} \tag{2.2}$$

Al variar los parámetros de aceptación o rechazo se obtienen diferentes pares de valores FAR y TAR, que al momento de ser graficados, trazan la curva Característica Operativa del Receptor (ROC, por sus siglas en inglés) [29]. Así, un sistema de reconocimiento ideal debe alcanzar el punto (0,1), lo que significa que ninguna prueba negativa es etiquetada como positiva y todas las pruebas positivas son etiquetadas correctamente, en otras palabras todos los usuarios del sistema son aceptados, mientras que todos los intrusos son rechazados, esto se muestra en la Figura 2.5.

Ya que la condición de etiquetado depende de un límite o umbral establecido para la prueba, la curva ROC permite observar de manera gráfica el comportamiento del clasificador cuando este límite varía y evaluar el desempeño del mismo, como se puede ver en la Figura 2.5a, además se puede realizar la comparación de diversos



(a) Curva ROC para clasificación de algoritmos de reconocimiento.

(b) Comparación entre algoritmos de reconocimiento.

Figura 2.5: Evaluación de algoritmos mediante la Curva ROC [27].

clasificadores al mismo tiempo, como se observa en la Figura 2.5b [27].

Para realizar la evaluación de las técnicas de reconocimiento se han creado diversos programas que incluyen sus propias bases de datos, consistentes en imágenes de rostros de varios sujetos, que dependiendo de la aplicación pueden variar en formato, posición, resolución, tipo, etc.

Por ejemplo, uno de los primeros programas fue el denominado Tecnología de Reconocimiento Facial[1] (FERET, por sus siglas en inglés), se llevó a cabo entre los años 1993 a 1997, con la misión de desarrollar capacidades de reconocimiento facial automático que pudieran ser empleados para asistencia de seguridad, inteligencia y cumplimiento de ley. Para lograr su propósito, el programa proporciona la base de datos FERET y establece la prueba FERET, siendo ambos los requerimientos necesarios para soportar la producción de sistemas de reconocimiento facial. Tres pruebas son llevadas a cabo: Agosto 1994, Marzo 1995 y Septiembre 1996. La primera de estas, establece por primera vez una base para evaluar el rendimiento de los algoritmos de reconocimiento capaces de localizar, normalizar e identificar rostros. La segunda prueba mide el progreso de los sistemas hasta ese momento y evalúa algoritmos con

[1] Face Recognition Technology

galerías más extensas que el de su sucesor, que constaba de 817 imágenes. Por último, la prueba de 1996, tiene tres objetivos: evaluar métodos del estado del arte, identificar futuras áreas de investigación y medir el rendimiento de los algoritmos. En total la base de datos contiene 14,126 imágenes de 1199 individuos [50].

Posteriormente, el programa Prueba de Vendedores del Reconocimiento Facial[2] (FRVT, por sus siglas en inglés) continúa con los trabajos del programa FERET, para la evaluación de los algoritmos desarrollados a partir de 1997. Este programa es reportado para los años 2000 (FRVT2000), 2002(FRVT2002), 2006(FRVT2006-ICE2006) y 2014 (FRVT2014).

Desde el año 2000 los prototipos de reconocimiento facial se transformaron en sistemas comerciales, por ello el programa realiza las pruebas FRVT2000 [9] y FRVT2002 [49] para evaluar el progreso y rendimiento en la vida real con bases de datos a gran escala, además de introducir nuevos experimentos para un mejor entendimiento del reconocimiento facial.

En un punto intermedio surge el programa Gran Reto del Reconocimiento Facial[3] (FRGC, por sus siglas en inglés), con el propósito de evaluar los métodos a partir del 2002 y desarrollar algoritmos que sobrepasen en un orden de magnitud el rendimiento registrado por el FRVT2002 (tasa de verificación de 80 % a una tasa de falsa aceptación (FAR) de 0.1 %). Este incremento en un orden de magnitud significaría un 98 % en la tasa de verificación para la misma tasa de falsa aceptación de 0.1 %, para lograrlo se fijaron como meta desarrollar una base de datos de imágenes estáticas (2D y 3D) para medir dichas tasas [48].

En el año 2006 el programa FRVT es más ambicioso, pues junto con el FRVT2006 se realiza el Reto de Evaluación de Iris[4] (ICE2006) que reporta por primera vez el rendimiento de algoritmos de reconocimiento a través de iris, siendo la primera comparación multimodal biométrica. Además, en un experimento se compara el rendimiento de estos algoritmos contra el reconocimiento humano, dando como resultado que siete algoritmos de reconocimiento facial con mejor desempeño son comparables e incluso sobrepasan el rendimiento humano, tomando diferentes condiciones de iluminación [51].

[2] Face Recognition Vendor Test
[3] Face Recognition Grand Challenge
[4] Iris Challenge Evaluation

La última evaluación del programa, FRVT2014 se presenta con el propósito de evaluar el rendimiento de los algoritmos en el escenario de identificación (uno a muchos), principalmente con imágenes de alta resolución que asemejan condiciones reales, por ejemplo aquellas imágenes tomadas para credenciales como pasaportes, visas, o licencias de conducir, e imágenes de baja resolución tomadas con cámaras web como las usadas para videovigilancia para mostrar el decremento en la eficiencia en condiciones no controladas. En su mayoría los algoritmos evaluados son de proveedores comerciales. Como resultado de los experimentos, concluyen que el rendimiento de los algoritmos depende de la aplicación para la que estos son diseñados, así, un algoritmo puede tener una mejor eficiencia en imágenes de baja calidad si está diseñado para trabajar con imágenes web, mientras que otros algoritmos aumentan la velocidad de búsqueda dependiendo del tamaño de la población en la base de datos [31].

Una de las bases de datos más extensas es la generada por el programa FRGC, actualmente es de acceso abierto para investigadores y desarrolladores de la industria, academia e instituciones. El programa da inicio en mayo de 2004 y termina en marzo de 2006, aunque la colección de imágenes da comienzo desde enero de 2003. El FRCG es estructurado en dos versiones o retos. La versión 1 es diseñada para introducir a los participantes al problema del FRGC, su formato y su infraestructura, mientras que la versión 2 es el reto para investigadores de alcanzar la meta de rendimiento de 98 % de verificación. La base de datos consiste en 50,000 imágenes, conteniendo imágenes estáticas de alta resolución tomadas bajo condiciones controladas y no controladas de iluminación, además de escaneos 3D, por lo que se considera una de las primeras bases de datos con imágenes 3D [48].

2.3. Estado del Arte

Los primeros trabajos de reconocimiento facial computarizado se reportan en la década de los 70's. Harmon y Kanade son los primeros en implementar sistemas capaces de reconocer rostros humanos mediante computadora. En su tesis doctoral, Kanade desarrolla un sistema para reconocer rostros con una alta precisión, detectando 608 rostros contenidos en una base de datos de 670 imágenes [38]. Por su parte, Harmon [34], [33] identifica perfiles del rostro dibujados por un artista. Ellos son considerados como los pioneros del reconocimiento facial.

Desde entonces, decenas de algoritmos han surgido con el amplio desarrollo en el ámbito computacional, por lo que se clasifican de acuerdo a la metodología y al tipo

de imágenes que usan. Dependiendo del algoritmo se dividen en métodos holísticos, basados en características e híbridos [69], y de acuerdo al tipo de imágenes en que se basan son métodos 2D o 3D [11].

En las aproximaciones holísticas se tienen representaciones globales del rostro mediante un arreglo bidimensional de valores de intensidad y el reconocimiento se realiza a través de correlaciones comparadas entre la imagen de entrada y la base de datos [37].

Uno de los métodos holísticos más utilizados hasta ahora, es presentado por Kirby y Sirovich, que implementa un modelo estadístico conocido como análisis de componentes principales (PCA por sus siglas en inglés). La principal ventaja de PCA es la reducción de las dimensiones, a través de la creación de un subespacio obtenido mediante los vectores propios de la matriz de covarianza de los datos originales, reteniendo la mayor variación. [39].

Los vectores propios obtenidos mediante PCA son denominados, por Turk y Pentland, como *eigenrostros*. En su trabajo, también presentan una aproximación utilizando dichos eigenrostros para la detección e identificación de rostros, además de un sistema de reconocimiento que sigue a la persona y es capaz de identificarla si se encuentra dentro de base de datos [63].

Otros modelos holísticos representativos son:
Análisis de discriminante lineal (LDA por sus siglas en inglés), propuesto por Etemad y Chellapa [23] quienes proponen este método semejante al usado por Pentland, la diferencia radica en el análisis de los eigenvectores, que son obtenidos de la matriz de separación en lugar de la matriz de covarianza, lo que permite tener una clasificación entre clases (sujetos), pues dichos eigenvectores muestran la máxima discriminación entre clases.
Fisherfaces es otro algoritmo de clasificación entre clases o sujetos, usando dos matrices de dispersión maximiza la diferencia entre imágenes de diferentes sujetos (diferencia entre clases) y minimiza la diferencia de imágenes del mismo sujeto (diferencia intra-clases) [8].
Análisis de componentes independientes (ICA por sus siglas en inglés), es una generalización del análisis de componentes principales que toma en cuenta las relaciones de alto orden de los píxeles, por ejemplo, los bordes en la forma o curvas del rostro son relaciones de este tipo, además, este algoritmo no restringe a los nuevos ejes a ser ortogonales [7].

Además existen investigaciones que analizan estos métodos holísticos con partes del rostro de manera independiente [47], [14].

Las aproximaciones basadas en características identifican, extraen y miden los rasgos faciales como ojos, nariz, boca, u otros puntos para después calcular relaciones geométricas y reducir la imagen a un vector, llamado vector de características para poder realizar la comparación [37].

Algunas extraen los rasgos específicos y usan plantillas deformables del rostro en 3D [10], [22] o grafos elásticos para la comparación [66]. En [12] se realiza una comparación concluyendo que los algoritmos basados en plantillas pueden ser más eficientes que algoritmos basados en comparaciones geométricas, al tener una tasa de reconocimiento de 100 % y 90 % respectivamente en una base de datos de 188 imágenes de 47 individuos.

Otros métodos incluyen redes neuronales [41], [61], wavelets [57],modelos de Markov [44], clasificadores de soporte vectorial [32], representación dispersa [67], patrones locales binarios [3], y algoritmos basados en video secuencias [60].

La clasificación de los métodos híbridos son métodos que combinan dos o más de estos algoritmos, y reportan en algunos casos un mejor desempeño que los algoritmos por separado [20], [43], [69].

Entre los sistemas más eficientes se tienen los métodos Deep Learning [55], [61], sin embargo generalmente requieren de cierta cantidad de datos y hardware especializado para entrenamiento del sistema y ponerlo en práctica [28].

En los resúmenes más actuales se muestran comparaciones en rendimiento y eficiencia, además de las ventajas y desventajas de cada uno de los algoritmos [52], [40], [56].

2.4. Eigenfaces

La idea básica del método a utilizar es extraer la información relevante en la imagen de un rostro y codificarla, para después compararla con una base de datos previamente tratada de la misma forma conocida como conjunto de entrenamiento. Matemáticamente, se logra mediante los componentes principales, es decir, obteniendo los

eigenvectores de la matriz de covarianza de un grupo de imágenes y formar un espacio de imágenes llamado espacio de rostros. Dichos eigenvectores pueden ser mostrados como una especie de "rostros fantasma" los cuales han sido denominados eigenrostros y se observan en la Figura 2.6 [63], [39].

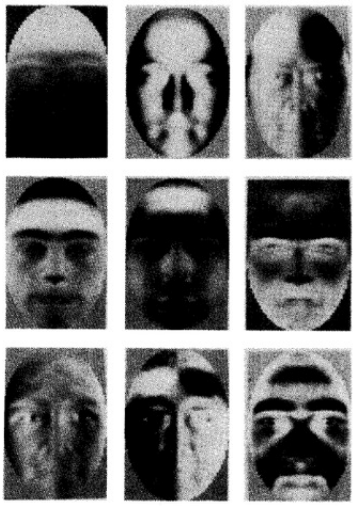


Figura 2.6: Eigenrostros generados mediante PCA [39].

Para comenzar, se define a la imagen como una función bidimensional $f(x, y)$, donde x y y son coordenadas espaciales y la amplitud de f en cualquier par de coordenadas (x, y) es llamada intensidad o nivel de grises. Una imagen digital está compuesta por una cantidad finita de elementos con una ubicación y valor particular. Estos elementos son llamados píxeles, y se consideran los elementos básicos de una imagen [30].

De esta manera, se puede tomar una imagen $\varphi(x, y)$ como una matriz de PxQ valores de intensidad para cada píxel o bien un vector de dimensión PxQ. Por ejemplo, una imagen de tamaño $256x128$ describe un vector de dimensión $32,768$ o equivale a un punto en un espacio $32,768 - D$, por lo tanto, un ensamble de imágenes puede verse como una colección de puntos en este espacio.

A partir del conjunto de entrenamiento que contiene M imágenes de rostros $\Gamma_1, \Gamma_2, \Gamma_3, ..., \Gamma_M$, es posible determinar el rostro promedio, y éste se define como:

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

por lo que ahora cada rostro varía de este promedio mediante el vector $\Phi_i = \Gamma_i - \Psi$, al realizar esta resta, obtenemos un conjunto de datos centrado con media igual a cero.

Este conjunto de vectores es el centro de estudio del análisis de componentes principales, que busca un conjunto de R vectores ortonormales u_k y sus eigenvalores asociados λ_k para describir la mejor distribución de los datos. Los vectores u_k y los escalares λ_k son los eigenvectores y eigenvalores, respectivamente de la matriz de covarianza:

$$C = \frac{1}{M} \sum_{n=1}^{M} \phi_n \phi_n^T = AA^T, \tag{2.3}$$

donde la matriz $A = [\Phi_1, \Phi_2, ..., \Phi_M]$.

Por la ecuación 2.3, la matriz C es de tamaño $PxQxPxQ$ y determina los PxQ eigenvectores y sus respectivos eigenvalores.
Para determinar estos vectores propios se utilizan métodos lineales, teniendo en cuenta que los eigenvectores de C son aquellos vectores que satisfacen la ecuación:

$$CU = \Lambda U, \tag{2.4}$$

donde cada columna de la matriz U es un eigenvector de C, además sujetos a la restricción $UU^T = I$, (donde I corresponde a la matriz identidad), y Λ es una matriz diagonal, donde los elementos de la diagonal son los eigenvalores.

Un método que determina los eigenvectores de manera simplificada es la descomposición de valores singulares (SVD, por sus siglas en inglés). El método SVD descompone una matriz cuadrada en tres matrices más simples:

$$C = P\Delta Q^T, \tag{2.5}$$

con P la matriz de eigenvectores normalizados de la matriz AA^T, llamados *vectores singulares izquierdos* de A, Q la matriz de eigenvectores normalizados de la matriz A^TA, llamados *vectores singulares derechos* y Δ la matriz de *valores singulares* $\Delta = \Lambda^{1/2}$, ordenados de mayor a menor sobre la diagonal [2].

La importancia de ordenar los eigenvalores de mayor a menor se debe a que estos valores representan la varianza de los datos de la matriz de covarianza y de esta manera se asegura que sus eigenvectores correspondientes tengan la dirección de mayor

variación.

Una vez que los eigenvectores han sido determinados, estos establecen el espacio de rostros PxQ, sin embargo, es posible utilizar un número M' de eigenrostros menor ($M' << PxQ$) para generar un espacio más pequeño, debido a que pueden considerarse únicamente los eigenvectores que representen la mayor cantidad de varianza de los datos originales. Esta es una de las mayores ventajas del método de eigenfaces, al reducir las dimensiones del espacio de rostros, facilitando el cálculo computacional, un ejemplo se muestra en la Figura 2.7.

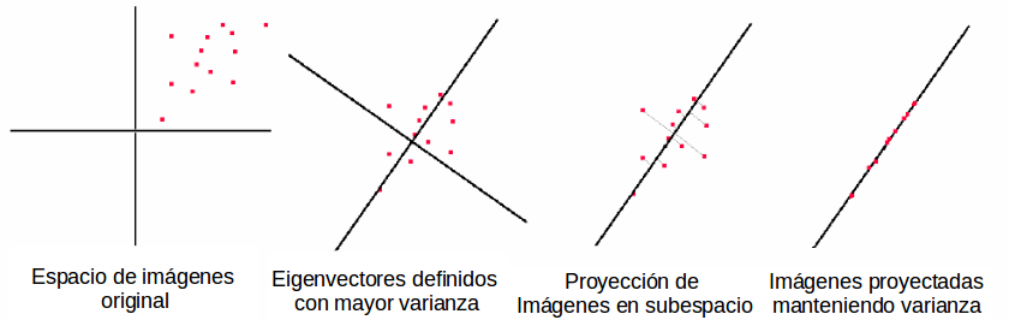


Figura 2.7: Definición geométrica del análisis de componentes principales, para la reducción de dimensiones del espacio de imágenes manteniendo la mayor variación [35].

Para realizar la selección de los M' eigenvectores, se utilizan los M' eigenvalores correspondientes, que sumen el 90 % de la varianza total [53].

Por último, para determinar si un nuevo rostro (Γ) pertenece a un sujeto dentro de la base de datos, la imagen del rostro es proyectada en el espacio de rostros a través de la operación $\omega_k = u_k^T(\Gamma - \Psi)$ para k desde 1 hasta M'. Un vector $\Omega = [\omega_1, \omega_2, ..., \omega_{M'}]$ describe la imagen de entrada Γ proyectada, y este es usado para comparar con cada uno de los rostros del conjunto de entrenamiento previamente proyectados de la misma manera.

Para realizar la comparación entre la imagen de prueba y las imágenes de entrenamiento, se utilizan distintas métricas. Las métricas estudiadas para realizar la comparación de las imágenes de prueba con las imágenes o conjunto de entrenamiento se denominan técnicas del vecino-más-cercano. El clasificador de vecino más cercano se

basa en una función métrica o de "distancia.entre patrones. Una metrica $D(a, b)$ es una función que da como resultado una distancia escalar generalizada entre dos argumentos. Para el caso general se tratará un espacio n-dimensional, donde se tienen un conjunto de M puntos (imágenes o sujetos) de entrenamiento proyectadas en el espacio R^n y se desea encontrar el más cercano al punto de prueba x [21].
Para ello, se hace uso de dos métricas: distancia Euclidiana y distancia Mahalanobis.

Distancia Euclidiana

En esta métrica se analiza cada punto de entrenamiento p_j contra el punto de prueba x, calculando la distancia euclidiana para cada uno de ellos:

$$D(x, p_j) = \sqrt{\sum_{i=1}^{M'} (x_i - p_{ij})^2}, \tag{2.6}$$

donde el valor j va desde 1 hasta M, al ser la cantidad de imágenes en el conjunto de entrenamiento, y el valor M' es la dimensión del espacio de rostros.

La identidad seleccionada será aquella imagen del sujeto con la que se obtenga la distancia mínima, la representación en un espacio bidimensional se observa en la Figura 2.8.

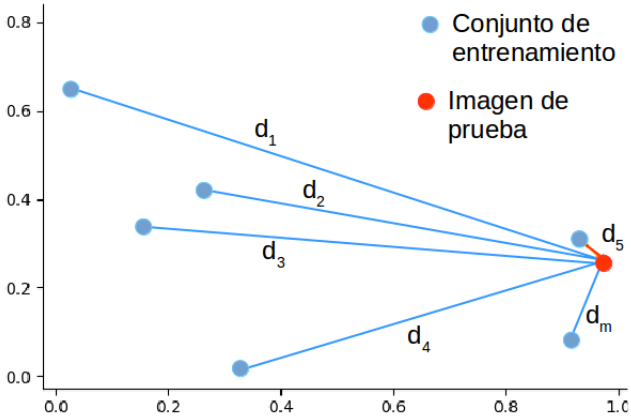


Figura 2.8: Representación de la distancia Euclidiana de un punto de prueba a cada punto del conjunto de entrenamiento.

Distancia Mahalanobis

La definición de esta métrica es la distancia de un punto $x \epsilon R^n$ (punto de prueba), a la media μ de un conjunto de puntos (conjunto de entrenamiento) existente en R^n. Debemos considerar que para nuestro caso en particular $n = M'$.

$$D(x, \mu) = \sqrt{(x - \mu)^t C^{-1} (x - \mu)} \tag{2.7}$$

Geométricamente se observa que a partir del conjunto de puntos en R^n, se define un sistema de coordenadas $R^{n'}$ en términos de su varianza, y este conjunto representa una hiper-elipse en el mismo espacio n'-dimensional, cuyo origen se centra en la media μ y los nuevos ejes están definidos por los eigenvectores de la matriz de covarianza C, en cuyo caso el primer eje se define en la dirección de mayor varianza. Este sistema está normalizado y los datos se encuentran dentro de una distribución Gaussiana, por lo que la escala está dada en términos de la desviación estándar, ver Figura 2.9 [21].

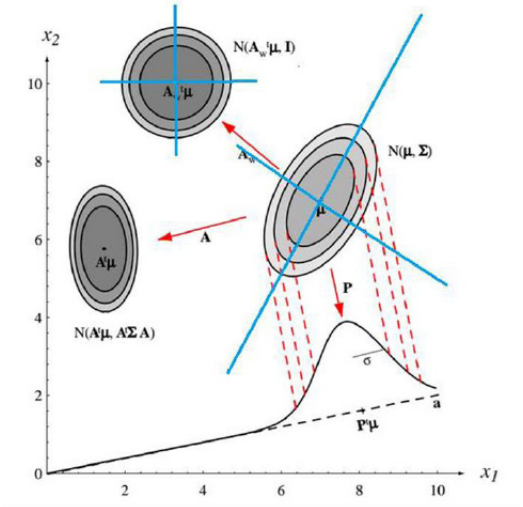


Figura 2.9: Representación de una elipse creada por un conjunto de puntos A con media μ y matriz de covarianza Σ en R^2, y la creación de un sistema de coordenadas en $R^{n'}$ [21].

En ambas métricas, es posible establecer el límite o umbral para el valor de la distancia para determinar si la decisión tomada es de aceptación o rechazo. Por ejemplo, si un sujeto dentro de la base de datos desea ser identificado, el sistema otorgará la

identificación del sujeto más cercano, si este valor está dentro del límite establecido, se dice que si es el sujeto, en cambio, si el valor de la distancia sobrepasa dicho límite, es posible que no sea el sujeto, por lo tanto puede hacerse una nueva lectura con otra imagen del mismo sujeto para validar nuevamente su identidad.

Capítulo 3

Desarrollo Experimental

En este Capítulo se explica el proceso que se lleva a cabo para crear el sistema de reconocimiento facial basado en Eigenfaces. Primeramente se da pauta para describir el escenario en el que se lleva a cabo, tomando en cuenta resultados obtenidos en trabajos previos y describiendo la base de datos con la cual se realiza la evaluación del algoritmo. Después se muestra el procesamiento que se les da a las imágenes para mejorar el desempeño del algoritmo, además de explicar su almacenamiento en dos conjuntos diferentes: imágenes de rostro completo e imágenes de la parte central del rostro. Como continuación se describe la manera de evaluación y por último se muestran los resultados obtenidos.

3.1. Escenario Experimental

El algoritmo de reconocimiento facial fue implementado en el lenguaje de programación Python, aunque cabe destacar que también se desarrolló con el software científico Matlab, dicho algoritmo es el llamado Eigenfaces, basado en PCA [63]. De acuerdo a resultados previamente documentados, este algoritmo tiene una eficiencia que va desde el 66 % hasta el 77 %. Dicho porcentaje varía pues de utilizan distintas partes del rostro para realizar el reconocimiento [14].

El procedimiento seguido para el desarrollo del sistema de reconocimiento se muestra en la Figura 3.1.

Como se muestra en la Sección 2.2, las bases de datos sirven para evaluar el rendimiento de los algoritmos. Para evaluar nuestro sistema de reconocimiento facial se utilizó la base de datos Face Recognition Grand Challenge (FRGC) [48]. Tomada en

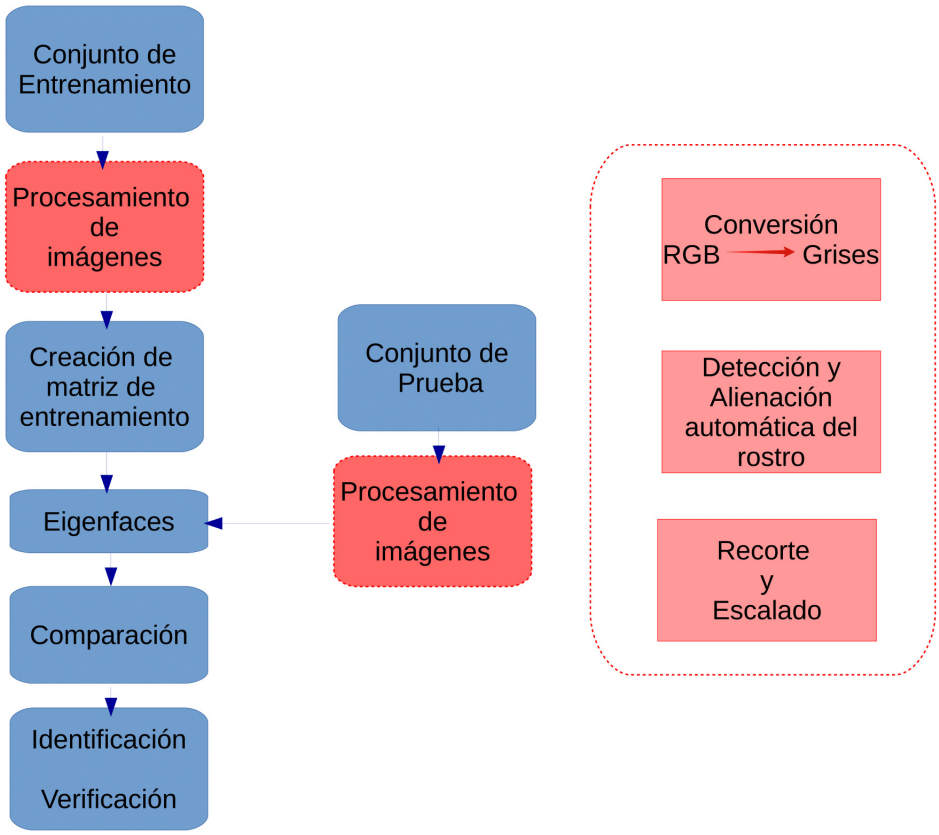


Figura 3.1: Procedimiento para elaborar un sistema de reconocimiento facial

tres sesiones (Spring-2003, Fall-2003 and Spring-2004), esta base de datos contiene 4950 imágenes 2D y sus correspondientes imágenes de profundidad.
El conjunto de imágenes tomadas en Spring-2003 fue colectada bajo condiciones controladas, aunque los sujetos fueron colocados a diversas distancias de la cámara, como consecuencia, muchas de las imágenes incluyen no solo el rostro, si no además también la parte superior del cuerpo y el fondo de la habitación. De manera general, todas las imágenes presentan una pose frontal, sin oclusiones con una pose neutral y bajo condiciones controladas de iluminación.

Por las razones anteriores, el conjunto Spring-2003 es seleccionado para realizar la evaluación. En el Cuadro 3.1 se muestra la organización del conjunto de imágenes

seleccionado, donde se tienen 275 sujetos, y las fotografías por sujeto varían desde 1 hasta 8.

Cuadro 3.1: Cantidad de imágenes por sujeto en el conjunto Spring-2003 de la base de datos FRGC.

Imágenes por sujeto	Spring-2003
1	77
2	32
3	47
4	33
5	28
6	30
7	15
8	13
Imágenes	943
Sujetos	275

De la misma manera, para realizar la identificación, cada una de las imágenes es etiquetada para saber la identidad de los sujetos. Se asignan 275 identificaciones, que corresponden a cada sujeto, y a sus respectivas imágenes se les coloca un subíndice, del uno al ocho, dependiendo cuantas imágenes del mismo sujeto contenga la base de datos. Por ejemplo, para el primer sujeto que tiene ocho imágenes, la manera de etiquetar queda de la siguiente manera: 1_1, 1_2, ...,1_7, 1_8.

Para los posteriores experimentos se dividen las imágenes en dos conjuntos: conjunto de entrenamiento y conjunto de prueba. El conjunto de entrenamiento es la base de datos de los usuarios registrados previamente etiquetados, mientras que el conjunto de prueba se refiere a las imágenes de entrada de los sujetos que desean ser identificados.

En nuestros experimentos se realiza el reconocimiento usando imágenes del rostro completo e imágenes de la región central del rostro, comprendiendo únicamente la nariz, desde el labio superior, hasta debajo de las cejas. Esta parte del rostro se considera más robusta a variaciones en expresión [14]. En la Figura 3.2 se muestra un ejemplo de las imágenes utilizadas tomadas de la base de datos del FRGC, en 3.2a

se muestra la imagen del rostro completo, mientras que en 3.2b se muestra la región central del rostro. En la siguiente sección se describe el procesamiento que se lleva a cabo para generar ambas imágenes.

(a) Ejemplo de una imagen del rostro completo

(b) Ejemplo de una imagen de la región central del rostro

Figura 3.2: Conjuntos de imágenes para realizar el reconocimiento facial

3.2. Procesamiento de Imágenes

El procesamiento previo que sufrieron las imágenes se realiza mediante lenguaje Python, y se describe a continuación.

El algoritmo de eigenfaces, utiliza imágenes en escala de grises; es decir, el valor de la intensidad de los pixeles está contenido en 256 niveles de grises, en otras palabras varía de 0 a 255. Sin embargo, las imágenes originales de la base de datos del FRGC se encuentran disponibles a color, por lo que es necesario realizar la conversión del canal RGB a escala de grises, en este mismo paso, se realizó una alineación para ubicar los ojos en posición totalmente horizontal, para beneficiar al desempeño del algoritmo [35].

El siguiente paso consta de la detección automática del rostro y de la región central, mediante el clasificador cascadas de Haar integrado en la librería de visión artificial OpenCV [5].

Las librerías de detección proporcionan un recuadro en la zona deseada, siendo esta zona la que se procede a recortar de las imágenes para su almacenamiento. De esta manera se almacenaron dos conjuntos distintos de la base de datos: región completa del rostro y región central del rostro.

El siguiente paso fue el escalado de ambos conjuntos de imágenes, se determinó usar una resolución de $75x112$ pixeles para todas las imágenes, y de esta manera tener un espacio $8400 - dimensional$, mismo que es utilizado en trabajos anteriores [35]. La Figura 3.3 muestra el procesamiento de imagen de un sujeto de la base de datos.

Figura 3.3: Procesamiento de imágenes de la base de datos FRGC.

3.3. Generación del Espacio de Rostros

Una vez que se tienen las imágenes procesadas, el siguiente paso es obtener los eigenvectores que mejor expresan la varianza y generar el espacio de rostros, para ello, es necesario generar la matriz de datos correspondiente, a partir del denominado conjunto de entrenamiento. En esta parte se tomó cada imagen de tamaño $75x112$ del conjunto de entrenamiento y se transformó en un vector de dimensión 8400 para formar la matriz de entrenamiento, luego de que cada vector sea una columna en dicha matriz, este proceso se muestra en la Figura 3.4.

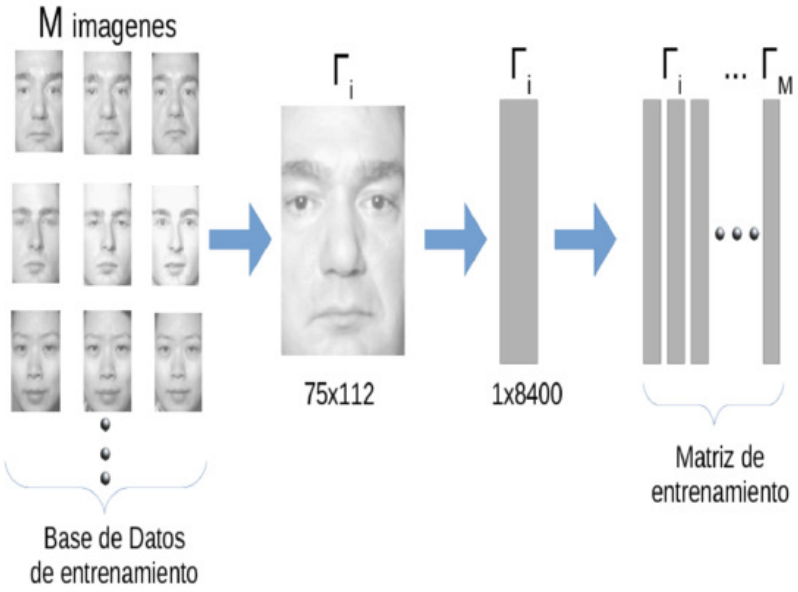


Figura 3.4: Construcción de la matriz de entrenamiento a partir del conjunto de imágenes de entrenamiento.

A partir de la matriz de entrenamiento se obtiene la matriz de covarianza, restando el rostro medio obtenido a cada imagen, para tener una distribución de datos con media cero. En la Figura 3.5 se observa el rostro medio obtenido en una de nuestras pruebas a partir de 415 imágenes de entrenamiento.
El siguiente paso fue la obtención de los eigenvectores o eigenrostros a partir de la matriz de covarianza del conjunto de entrenamiento. Para ello se utilizó el método lineal Descomposición de Valores Singulares (SVD, por sus siglas en inglés). Este método da como resultado la matriz diagonal de eigenvalores ordenados de mayor a menor y la matriz de eigenvectores respectivos a su eigenvalor, por lo que facilita el algoritmo, debido a que se deben tomar los eigenvectores que representen la mayor varianza de los datos originales.

En este paso se realiza la reducción de dimensiones, al seleccionar únicamente los eigenrostros con mayor variación. El procedimiento seguido para determinar cuantos eigenrostros usar para la creación del subespacio, es ocupar el número de eigenvectores correspondiente al número de eigenvalores para los que su suma se igual al 90 % de la variación de los datos originales, en otras palabras, se realiza la suma uno por uno de los eigenvalores y se divide entre la suma total, y aquel en donde se iguale

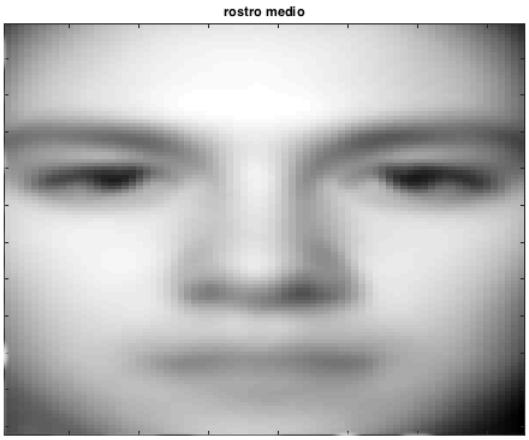


Figura 3.5: Rostro medio obtenido a partir del conjunto de entrenamiento de la prueba 3 de nuestra experimentación.

o se supere el valor 0.9, ese será el número de eigenvectores a ocupar. La representación de los eigenvectores es semejante a la de un rostro, debido a esto, se le da la denominación eigenrostros, sin embargo, tienen un aspecto fantasmal. En la Figura 3.6 se muestran los eigenrostros obtenidos en una de nuestras pruebas.

Al determinar la cantidad de dimensiones del espacio, mediante los eigenrostros, es momento de proyectar las imágenes de entrenamiento en dicho subespacio, para su posterior comparación con la(s) imagen(es) de prueba, mismas que también deberán ser proyectadas en este espacio.

Por último se realiza la comparación mediante el uso de las métricas, la distancia euclidiana y la distancia mahalanobis. Para determinar la identidad de un sujeto de prueba se obtiene la distancia mínima obtenida entre todas las mediciones. Se han establecido límites para determinar si la identidad señalada por la distancia mínima corresponde o no al sujeto de estudio. Al ser una distribución normal, los límites establecidos están en unidades de desviación estándar.

Figura 3.6: Visualización de los dos primeros eigenvectores o eigenrostros que generan el espacio de rostros en la prueba tres de nuestros experimentos.

3.4. Evaluación Experimental

La forma de evaluación consiste en dos experimentos distintos. La primer experimentación utiliza la distancia Euclidiana como métrica de clasificación, y está dividida en tres pruebas con la finalidad de determinar que prueba es más eficiente y comprobar si la zona central del rostro es más robusta, como se menciona en la literatura. Las mismas tres pruebas se realizan tanto para imágenes del rostro completo, como para la región central del rostro. Las pruebas establecidas en este experimento se mencionan a continuación y se describe de igual manera en el Cuadro 3.2:

Prueba 1.
Se toman en cuenta a los 275 sujetos en la base de datos, (con una imagen por sujeto), es decir, 275 imágenes para generar el conjunto de entrenamiento.
Prueba 2.
Se toman a los 197 sujetos con dos o más imágenes en la base de datos, generando una galería de entrenamiento de 394 imágenes con 2 imágenes por sujeto.
Prueba 3.
Se toman a 83 sujetos con cinco o más imágenes en la base de datos para generar el conjunto de entrenamiento con 415 imágenes, es decir, 5 imágenes por sujeto.

Cuadro 3.2: Cantidad de imágenes en los conjuntos de entrenamiento para cada prueba.

Prueba	Conjunto de Entrenamiento	Imágenes por sujeto
1	275	1
2	394	2
3	415	5

Es necesario destacar que para estas pruebas, todas las imágenes en el conjunto de entrenamiento se encuentran almacenadas en una carpeta, en otras palabras, cada imagen es proyectada de manera independiente en el espacio de rostros, por lo que no existen clases o clasificación por sujetos.

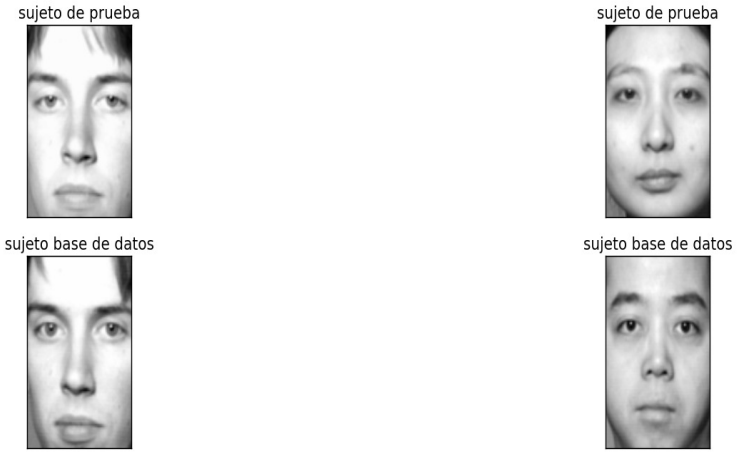
El conjunto de prueba se establece de la misma manera para cada una de las pruebas. Se seleccionan 20 sujetos que estén incluidos en el conjunto de entrenamiento, pero se selecciona una imagen distinta de las que forman dicho conjunto. De esta manera el conjunto de entrenamiento queda comprendido por 20 imágenes de 20 sujetos distintos.

El segundo experimento utiliza la métrica de Mahalanobis para la clasificación. Este experimento se lleva a cabo usando a los 28 sujetos de la base de datos que contienen 7 o más imágenes del rostro completo. Para formar el conjunto de entrenamiento se tomaron 6 imágenes por sujeto. Como la comparación de la métrica Mahalanobis es de una imagen de prueba a la media de un conjunto de entrenamiento, este experimento si realiza la clasificación por clases o sujetos. Para ello, las seis imágenes del mismo sujeto se almacenan en una carpeta, por lo que se tienen 28 carpetas correspondientes a cada sujeto.

El conjunto de prueba de este experimento está formado a partir de una imagen por sujeto de los 28 que están incluidos, y cinco imágenes más de cinco sujetos desconocidos por la base de datos. En total se tiene un conjunto de prueba de 33 imágenes, una imagen por sujeto.

Nuestro sistema de reconocimiento es capaz de dar una identificación a todos los sujetos dentro del conjunto de prueba, al mismo tiempo. Dicho en otras palabras, se pueden determinar distintos usuarios en un solo paso.

El último paso de este proceso es la visualización de resultados, a partir de la asignación de una identidad a un sujeto. Una vez que se ha calculado la distancia mínima de una imagen de prueba a una imagen en el conjunto de entrenamiento, es posible asignar la identidad del sujeto que corresponde a esta última, a través del nombre con el que es etiquetado. Además, dicha comparación es mostrada, tal como se ve en la Figura 3.7, que muestra una evaluación acertada (3.7b) y una errónea (3.7a), mismas que pueden ser tomadas para aceptar al sujeto como usuario genuino en el primer caso, y como un impostor en el segundo.



(a) Comparación correcta (b) Comparación incorrecta

Figura 3.7: Resultados de las comparaciones del sistema de reconocimiento facial

3.5. Análisis de Resultados

A continuación se presentan los resultados obtenidos de los experimentos de evaluación para el sistema de reconocimiento facial.

La evaluación comienza a partir de cuantos sujetos son comparados correctamente, es decir, cuantos sujetos del conjunto de prueba han sido identificados de manera verdadera con su respectiva etiqueta en el conjunto de entrenamiento.
En el Cuadro 3.3 se muestra el porcentaje de reconocimiento obtenido por nuestro

sistema en cada una de las pruebas para las imágenes del rostro completo y de la región central del mismo. Se han obtenido tasas cercanas a las registradas en la literatura, siendo el 75 % el valor más alto registrado.

Cuadro 3.3: Tasa de reconocimiento

Prueba	Reconocimiento	
	Rostro Completo	Región Central
1	50	55
2	75	60
3	75	75

En la Figura 3.8 se muestra la curva ROC comparando las pruebas utilizando ambos conjuntos de imágenes, esta curva es generada mediante las ecuaciones (2.1) y (2.2), variando el límite de aceptación.

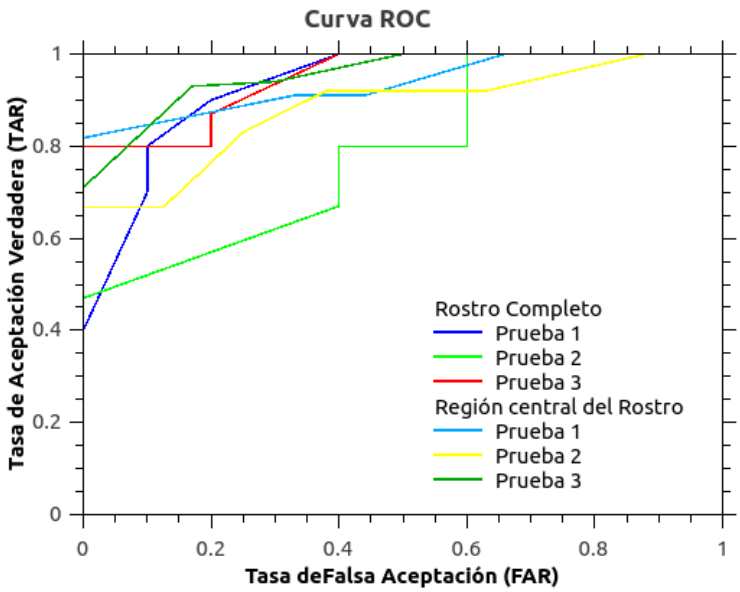


Figura 3.8: Curva ROC para las cuatro pruebas con la distancia euclidiana como métrica de clasificación.

Teniendo en cuenta que el punto $(0, 1)$ es punto ideal de reconocimiento, la prueba

número uno con imágenes de la zona central es la que más se acerca a este punto. A pesar de tener una de las tasas más bajas de reconocimiento, con el 55 %, lo que indica la curva ROC para esta prueba es que podemos estar seguros que los usuarios que fueron identificados de forma correcta son en verdad dichos usuarios.

Bajo la comparación anterior, la prueba tres en ambos conjuntos de imágenes, se acercan al punto ideal, por lo que representan las pruebas más confiables en nuestro sistema, al registrar 75 % de reconocimiento.

En la prueba 2 usando el rostro completo se presenta cierta desventaja a comparación de las demás, a pesar de tener una de las tasas de reconocimiento más elevadas.

Por otro lado, la experimentación usando la distancia Mahalanobis como métrica de clasificación no muestra el desempeño que esperábamos, teniendo una tasa de reconocimiento de 17.8 % al identificar correctamente a 5 usuarios de los 28 registrados en el conjunto de prueba y la curva ROC que se describe, está muy lejos del punto ideal, incluso mostrándose por debajo de un sistema aleatorio, tal como se observa en la Figura 3.9, por lo que tampoco se asegura que estos sean verdaderamente los usuarios.

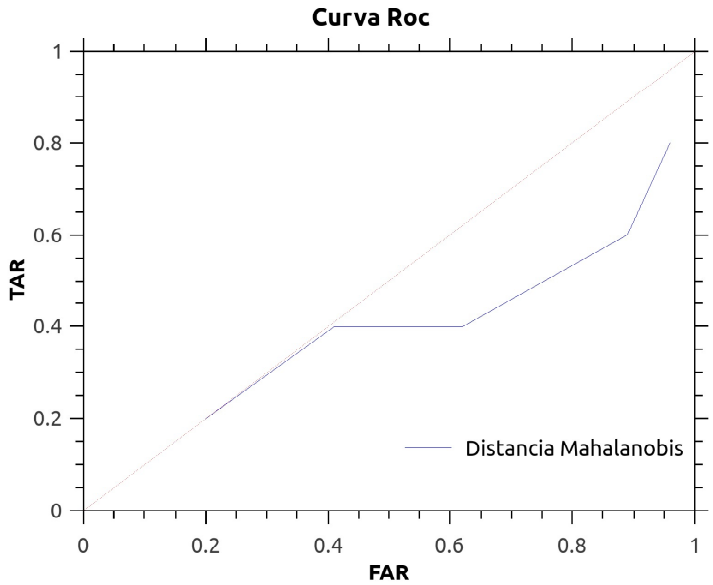


Figura 3.9: Curva ROC del experimento 2 usando distancia Mahalanobis como métrica de clasificación.

La baja precisión obtenida al usar esta métrica se debe a la baja cantidad de imágenes usadas por sujeto. En este caso ocurre una pérdida de información al generar el espacio de rostros, pues los eigenrostros utilizados para ello, deben ser una cantidad menor o igual que las imágenes de entrenamiento. Se esperaría un mejor desempeño usando más imágenes por sujeto.

Capítulo 4

Conclusiones y trabajo futuro

En este último capítulo se dan las conclusiones del trabajo realizado, entrando en detalles del uso del sistema de reconocimiento en posibles aplicaciones para la creación de ciudades inteligentes, y dando paso al trabajo futuro para mejorar el desempeño, con el objetivo de incrementar la tasa de reconocimiento.

Se ha logrado el objetivo principal de desarrollar un sistema de reconocimiento tomando en cuenta el algoritmo de Eigenfaces. A través de varias pruebas se ha evaluado el desempeño del mismo y se ha obtenido una tasa de reconocimiento del 75 %, valor cercano al reportado en la literatura. Además con este trabajo se abre paso a investigaciones de este tipo dentro de la Facultad de Ciencias de la UAEMéx.

Dentro del marco de investigación se ha introducido la plataforma Fiware, dando una idea general de sus alcances y objetivos principales. Además se hace énfasis de las aplicaciones recientes generadas a partir del reconocimiento facial y por sus características son ideales para su implantación dentro de las ciudades inteligentes.

Debido a la naturaleza con la que se ha implementado el algoritmo en este trabajo, las aplicaciones de nuestro sistema pueden ser vastas, al ser programado en lenguaje Python puede modificarse e implementarse de manera relativamente sencilla, por ello se pone a disposición el código. Por ejemplo, puede crearse una base de datos local para controlar el acceso a un sitio; con dicha base de imágenes se crea el conjunto de entrenamiento, mientras que a los usuarios que deseen ingresar se les toma una fotografía en el momento, la cuál es depositada en la carpeta del conjunto de prueba. La comparación realizada por el sistema de reconocimiento podrá permitir o denegar el acceso, si el usuario es genuino o es un impostor, según sea el caso.

Como parte del trabajo futuro, se tiene contemplado incluir otro método presentado en el estado del arte, como puede ser redes neuronales o usar imágenes de profundidad o 3D para elevar el porcentaje de la tasa de reconocimiento y aumentar el desempeño del sistema.

Otro punto a tratar será el reconocimiento facial basado en video, por lo que se pretende extender el sistema usando secuencias de imágenes capturadas por cámaras de video-vigilancia, en lugar de usar imágenes estáticas como hasta ahora.

Además, un punto clave que se tiene en mente es incluirlo como parte de la plataforma Fiware y pueda ser usado en las aplicaciones contempladas por la plataforma.

Por otra parte, con la creación del Centro de Datos se podrá trabajar en algoritmos más robustos que requieren de una gran cantidad de recursos computacionales y de almacenamiento.

Por último, en el Anexo B, se ha incluido el sistema de medición del tamaño de nanopartículas en micrografías, el cual se presentó como parte del Taller Alemania-Francia-México sobre magnetismo de nanoestructuras y grupos de aleaciones de materiales de transición, y es una aplicación directa del procesamiento de imágenes al ramo de la Física.

Apéndice A

Código

El código del sistema está dividido en dos módulos, el primero es el encargado de generar el espacio de rostros a partir de las imágenes en el conjunto de entrenamiento, mientras que el segundo proyecta las imágenes de los sujetos de prueba y realiza la comparación y otorga las identidades.

A continuación se muestra el primer módulo que lleva por nombre Eigenfaces_svd. Se explica cada linea del código mediante los signos #:

#Primera línea para aceptar caracteres especiales en Python
#!/*usr/bin/envpython*

#Librerías utilizadas en la creación del algoritmo, mismas que se encuentran disponibles en Python.
import os #Acceder a las carpetas del sistema
import glob #Acceder a los archivos
import cv2 #Librería OpenCV
import numpy as npy #Manejo de arreglos numéricos
from matplotlib import pyplot as plt #Graficador

#Directorio de las imágenes de entrenamiento (Se muestra la carpeta de la prueba 3 con 5 imágenes por sujeto)
$path = os.getcwd() + "/exp3\_5imgxperson"$

#Cantidad de imágenes en formato *ppm* en el conjunto de entrenamiento
$numero = len(glob.glob(path + "/ * .ppm"))$

```
# Mostrar en pantalla la cantidad de imágenes en el conjunto de entrenamiento
print 'Existen', numero, 'imágenes en la carpeta'

#Resolución de las imágenes
m = 75
n = 112
mxn = m * n

#Definición de la matriz de entrenamiento
Training = npy.empty(shape = (mxn, numero), dtype =' float64')

#Bucle para leer las imágenes de entrenamiento de manera ordenada.
columna = 0
for filename in sorted(os.listdir(path)):
                    pathname = os.path.join(path, filename)
                    print pathname
                    #Leer archivo y dar formato B/N
                    im = cv2.imread(pathname, 0)
                    #Transformar archivo en array de tipo flotante y linealizarlo
                    imagen_arr = npy.array(im, dtype =' float64').flatten()
                    #Escribir los arrays en columnas de la matriz de entrenamiento
                    Training[:, columna] = imagen_arr[:]
                    columna+ = 1

#Creación del rostro promedio, sumando las columnas de la matriz y dividiendo
entre el número de imágenes
medio = npy.sum(Training, axis = 1)/numero

#Visualización del rostro promedio
rostromedio = medio.reshape(n, m)
plt.imshow(rostromedio, cmap =' gray')
plt.title('RostroPromedio')
plt.xticks([]), plt.yticks([])
plt.show()

#Restar el rostro promedio a cada rostro de la matriz de entrenamiento
Trainingzm = Training.transpose() - npy.ones(shape = (numero, 1), dtype ='
float64') * medio
```

Obtención de eigenvalores y eigenvectores de la matriz de covarianza $C = B'B$ mediante SVD
#SVD calcula las matrices $[U, S, V]$
#U es la matriz unitaria de eigenvectores de la matriz $B * B'$
#S es la matriz de los eigenvalores, ordenados de mayor a menor
#V es la matriz unitaria de eigenvalores de matriz $B' * B$
$U, S, V = npy.linalg.svd(Trainingzm)$

#Tomar únicamente hasta el k-ésimo eigenvector que representa el 90 % de la varianza de los datos originales, tomando en cuenta que la varianza esta representada por los eigenvalores

$S = S.reshape(numero, 1)$
$sumaeigenval = sum(S[:])$
$eigenval\_k = 128$
$eigenval\_k = 0$
$sumvari = 0{,}0$
$varianza = 0{,}90$

for *eigenvalor* **in** $range(0, numero)$**:**
    $eigenval\_k += 1$
    $sumvari += sum(S[eigenvalor, :])/sumaeigenval$
    if $sumvari >= varianza$**:**
        break

#Mostrar en pantalla la cantidad de eigenvectores para crear el espacio de rostros
print*eigenval*$\_k$,*primeroseigenvectores*

#Mantener la cantidad de eigenvectores seleccionada
$eigenfaces = V[:, 0 : eigenval\_k]$

#Visualización de los primeros dos eigenrostros
$primPC = eigenfaces[:, 0]$
$primerPC = primPC.reshape(n, m)$
$segPC = eigenfaces[:, 1]$
$segPC = segPC.reshape(n, m)$

$plt.figure(1)$
$plt.subplot(211)$
$plt.imshow(primerPC, cmap =' gray')$
$plt.title('Primereigenvector')$
$plt.xticks([]), plt.yticks([])$
$plt.subplot(212)$
$plt.imshow(segPC, cmap =' gray')$
$plt.title('segundoeigenvector')$
$plt.xticks([]), plt.yticks([])$
$plt.show()$

#Proyección de las imágenes de entrenamiento
$Trainproyec = eigenfaces.transpose().dot(Trainingzm.transpose())$
$Trainproyec = Trainproyec.transpose()$

El segundo módulo requiere de las variables que se han generado en el primer módulo, por lo que se importa con el nombre asignado, tal como se muestra a continuación:

#!/*usr/bin/envpython*
Librerías de Python usadas en el segundo módulo; se incluye el módulo uno y las variables que se generaron en el. *import os*
import glob
import cv2
import numpy as npy
import math
from matplotlib import pyplot as plt
from Eigenfaces_svd import medio, m, n, mxn, eigenfaces, numero, eigenval_k, path, Trainproyec, Training

#Directorio del conjunto de prueba
$pathtest = os.getcwd() + "/test\_exp3"$
#Numero de sujetos de prueba
$numero\_t = len(glob.glob(pathtest + "/ * .ppm"))$

#Matriz de imágenes del conjunto de prueba
$test = npy.empty(shape = (mxn, numero\_t), dtype =' float64')$

#Bucle para crear la matriz con las imágenes de los sujetos de prueba

```
columna = 0
for filename in sorted(os.listdir(pathtest)):
                    pathnam = os.path.join(pathtest, filename)
                    printpathnam
                    imagentest = cv2.imread(pathnam, 0)
                    imag_t = npy.array(imagentest, dtype =' float64').flatten()
                    test[:, columna] = imag_t[:]
                    columna+ = 1

#Restar rostro promedio de la matriz de entrenamiento a imagen de prueba
testzm = test.transpose() − npy.ones(shape = (numero_t, 1), dtype =' float64') ∗
medio

#Proyección de imágenes test
testproyec = eigenfaces.transpose().dot(testzm.transpose())
testproyec = testproyec.transpose()

#Calculo de distancia
for i in range(0, numero_t) :
                    dif_e = Trainproyec − (npy.ones(shape = (numero, 1), dtype ='
float64') ∗ (testproyec[i, :]))
                    dist_e = npy.sqrt(npy.fabs(sum(dif_e.dot(dif_e.transpose()))))
                    minim = npy.amin(dist_e)

                    persona = 0
                    for p in dist_e:
                    persona+ = 1
                              if p == minim:
                                        break
                              printpersona, minim

                    #Visualización para comparación de ambos rostros
                    imagentest = test[:, i]
                    imagentesting = imagentest.reshape(n, m)
                    imagentrain = Training[:, persona]
                    imagentraining = imagentrain.reshape(n, m)

                    plt.figure(1)
```

```
                    plt.subplot(211)
                    plt.imshow(imagentesting, cmap =' gray')
                    plt.title('sujetodeprueba')
                    plt.xticks([]), plt.yticks([])
                    plt.subplot(212)
                    plt.imshow(imagentraining, cmap =' gray')
                    plt.title('sujetobasededatos')
                    plt.xticks([]), plt.yticks([])
                    plt.show()

num = 0
for filename in sorted(os.listdir(path)):
                    pathname = os.path.join(path, filename)
                    num+ = 1
                    printnum, pathname
```

Apéndice B

Sistema de medición de tamaño de nanopartículas

Con las bases aprendidas del procesamiento de imágenes se ha establecido un sistema de detección y medición del tamaño de nanopartículas en micrografías. Este trabajo fue presentado como parte del Taller Alemania-Francia-México sobre magnetismo de nanoestructuras y grupos de aleaciones de materiales de transición, el pasado 13 de octubre de 2017, en la Facultad de Ciencias de la UAEMéx.

La importancia de medir el tamaño de nanopartículas, ya sea en clusters o de manera individual se debe a que sus propiedades están extremadamente relacionadas a su morfología, de manera particular, el tamaño está relacionado con la anisotropía magnética del material. Su diseño y sus propiedades juegan un papel fundamental en numerosas aplicaciones como almacenamiento de la información y dispositivos magnetoelétricos [62].

El principio de la medición se basa en la comparación de los pixeles de cada nanopartícula con una métrica previamente establecida. De manera general, en las micrografías se puede observar una escala de tamaño nanométrico para comparar el tamaño de manera visual, ver Figura B.1. Dicha escala es la que se usa en este sistema para obtener la métrica deseada.

El sistema está divido en dos subprogramas, en el primero se obtiene la métrica a partir de la escala de la micrografía, mientras que en el segundo, se detecta cada una de las nanopartículas con base en el procesamiento de imágenes y con la métrica establecida se calcula su tamaño.

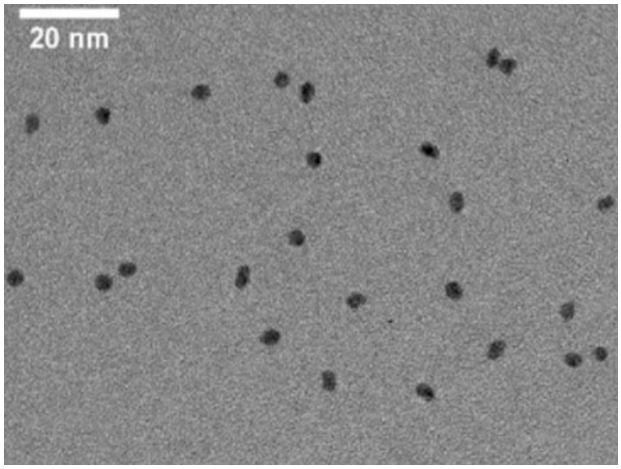
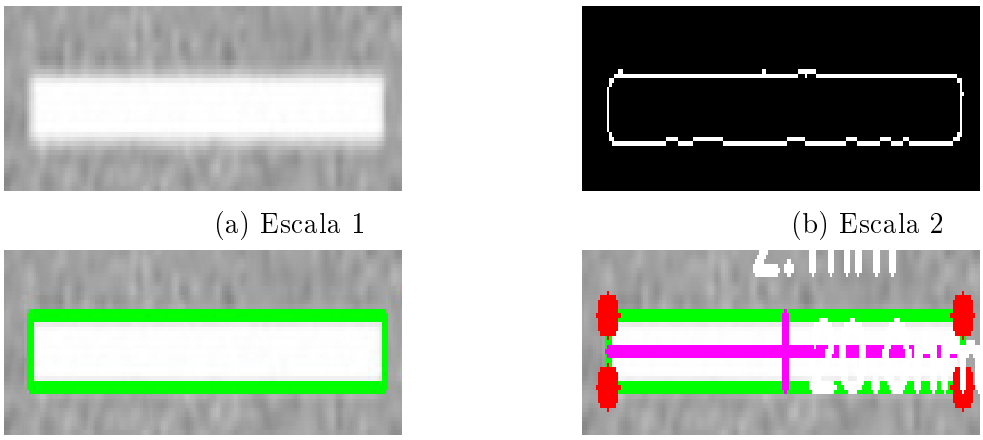


Figura B.1: CoPt nanoalloys

El subprograma para obtener la métrica consta de tres pasos:

1. Cortar manualmente la barra de la escala en la micrografía
2. Procesamiento de la imagen
3. Obtener métrica



(a) Escala 1

(b) Escala 2

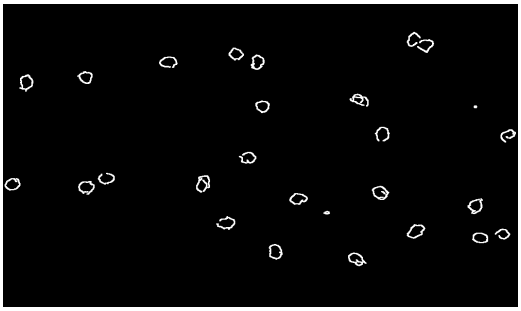
(c) Escala 3

(d) Escala 4

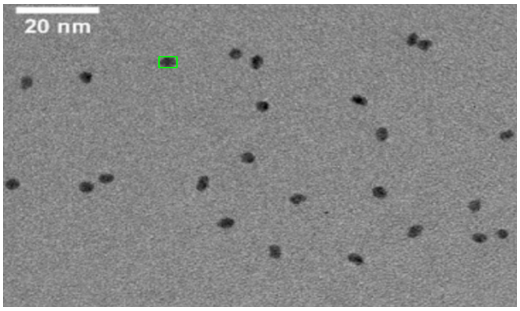
Figura B.2: Cálculo de la métrica de micrografía

En la Figura B.2 se muestra la escala recortada de la micrografía B.2a, el procesamiento de la imagen recortada para detectar y delimitar el borde de la escala B.2b, B.2c, y por último obtener la métrica B.2d. En este caso, la escala de la micrografía es de 20 nanómetros.

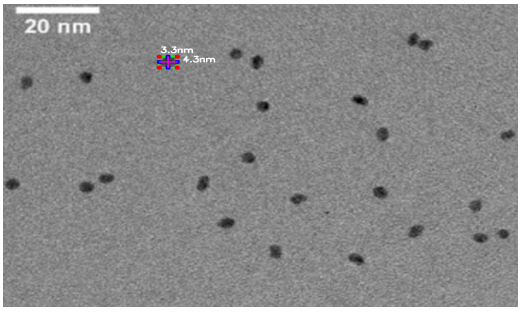
Una vez que se obtiene la métrica de la micrografía, el siguiente subprograma detecta el borde de las nanopartículas B.3a, selecciona una por una B.3b y les asigna su respectivo tamañoB.3c.
Al igual que el sistema de reconocimiento facial, este sistema está escrito en lenguaje Python basado en las librerías OpenCV [5].



(a) Detección de nanopartículas



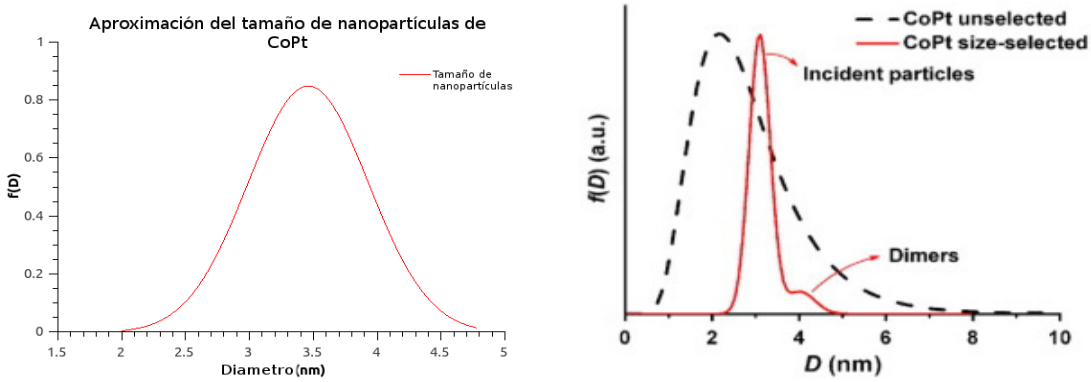
(b) Selección de la i-ésima nanoparticula



(c) Tamaño de la i-esima partícula

Figura B.3: Detección de nanopartículas y medición de su respectivo tamaño

Los resultados obtenidos son comparables con los establecidos en [62], tal como se muestra en la distribución de tamaños en la Figura B.4. En su trabajo, [62], consideran la aglutinación de dos o más nanopartículas, llamados Dímeros, y a ellos corresponde el segundo pico en la Figura B.4b.
Es necesario mencionar que este es un primer prototipo de este sistema, y para la



(a) Distribución de tamaños de nanopartícu- (b) Distribución de tamaño de nanopartículas las obtenidos con nuestro sistema de detección en [62]

Figura B.4: Resultados de las comparaciones del sistema de reconocimiento facial

detección de las nanopartículas se requiere que estas se encuentren separadas, pues al existir aglomeraciones, el sistema detecta todo el conjunto como una sola partícula y a este le asigna un tamaño, el cual sería incorrecto.

Como parte del trabajo futuro se planea la mejora de este sistema, para detectar efectivamente partículas dentro de aglomeraciones y pueda ser usado en laboratorios nacionales e internacionales.

Bibliografía

[1] Andrea F Abate, Michele Nappi, Daniel Riccio, and Gabriele Sabatino. 2d and 3d face recognition: A survey. *Pattern recognition letters*, 28(14):1885–1906, 2007.

[2] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[3] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(12):2037–2041, 2006.

[4] Lale Akarun, B Gokberk, and Albert Ali Salah. 3d face recognition for biometric applications. In *Signal Processing Conference, 2005 13th European*, pages 1–5. IEEE, 2005.

[5] M Alexander and K Abid. Opencv-python tutorials documentation, 2014.

[6] Apple. Face id security. http://images.apple.com/business/docs/FaceID_Security_Guide, 2017. [Web; accedido el 14-11-2017].

[7] Marian Stewart Bartlett, Javier R Movellan, and Terrence J Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on neural networks*, 13(6):1450–1464, 2002.

[8] Peter N. Belhumeur, João P Hespanha, and David J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):711–720, 1997.

[9] Duane Michael Blackburn, P Jonathon Phillips, and Mike Bone. *Facial recognition vendor test 2000 evaluation report*. US Department of Defense, 2001.

[10] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on pattern analysis and machine intelligence*, 25(9):1063–1074, 2003.

[11] Kevin W Bowyer, Kyong Chang, and Patrick Flynn. A survey of approaches and challenges in 3d and multi-modal 3d+ 2d face recognition. *Computer vision and image understanding*, 101(1):1–15, 2006.

[12] Roberto Brunelli and Tomaso Poggio. Face recognition: Features versus templates. *IEEE transactions on pattern analysis and machine intelligence*, 15(10):1042–1052, 1993.

[13] Andrea Caragliu, Chiara Del Bo, and Peter Nijkamp. 10 smart cities in europe. *Smart cities: governing, modelling and analysing the transition*, page 173, 2013.

[14] Kyong I Chang, Kevin W Bowyer, and Patrick J Flynn. Multiple nose region matching for 3d face recognition under varying facial expression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1695–1700, 2006.

[15] Hafedh Chourabi, Taewoo Nam, Shawn Walker, J Ramon Gil-Garcia, Sehl Mellouli, Karine Nahon, Theresa A Pardo, and Hans Jochen Scholl. Understanding smart cities: An integrative framework. In *System Science (HICSS), 2012 45th Hawaii International Conference on*, pages 2289–2297. IEEE, 2012.

[16] Agencia Informativa Conacyt. Centro de datos de la uaemex, alternativa para las empresas. http://www.conacytprensa.mx/index.php/tecnologia/tic/9299-centro-de-datos-de-la-uaemex-alternativa-para-las-empresas, 2016. [Web; accedido el 01-12-2017].

[17] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

[18] Centro de Investigación e Innovación en Tecnologías de la Información y Comunicación. Infotec. https://www.infotec.mx/, 2017. [Web; accedido el 22-07-2017].

[19] Gobierno de la Republica de los Estados Unidos Mexicanos. Diario oficial de la federación. http://www.dof.gob.mx/nota_detalle.php?codigo=5495299&fecha=29/08/2017, 2017. [Web; accedido el 22-11-2017].

[20] Oscar Déniz, M Castrillon, and Mario Hernández. Face recognition using independent component analysis and support vector machines. *Pattern recognition letters*, 24(13):2153–2157, 2003.

[21] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification*. John Wiley & Sons, 2012.

[22] BA Echeagaray-Patron, VI Kober, VN Karnaukhov, and VV Kuznetsov. A method of face recognition using 3d facial surfaces. *Journal of Communications Technology and Electronics*, 62(6):648–652, 2017.

[23] Kamran Etemad and Rama Chellappa. Discriminant analysis for recognition of human face images. *JOSA A*, 14(8):1724–1733, 1997.

[24] M. E. Fathy, V. M. Patel, and R. Chellappa. Face-based active authentication on mobile devices. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1687–1691, April 2015.

[25] Fiware. Fiware. https://www.fiware.org/, 2016. [Web; accedido el 22-07-2017].

[26] Fiware. Fiware mexico. http://www.fiwaremexico.org/, 2016. [Web; accedido el 22-07-2017].

[27] César Flores Lovera. *Prototipo de un sistema biométrico por reconocimiento facial 3D utilizando el sensor Kinect*. PhD thesis, Universidad Autónoma del Estado de México, 2016.

[28] Xavier Fontaine, Radhakrishna Achanta, and Sabine Süsstrunk. Face recognition in real-world images. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[29] Giorgio Fumera, Gian Luca Marcialis, Battista Biggio, Fabio Roli, and Stephanie Caswell Schuckers. Multimodal anti-spoofing in biometric recognition systems. In *Handbook of Biometric Anti-Spoofing*, pages 165–184. Springer, 2014.

[30] Rafael C Gonzalez and Richard E Woods. Image processing. *Digital image processing*, 2, 2007.

[31] Patrick Grother and Mei Ngan. Face recognition vendor test (frvt). *Performance of Face Identification Algorithms, NIST Interagency Report*, 8009:84, 2014.

[32] Guodong Guo, Stan Z Li, and Kapluk Chan. Face recognition by support vector machines. In *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pages 196–201. IEEE, 2000.

[33] LD Harmon, SC Kuo, PF Ramig, and U Raudkivi. Identification of human face profiles by computer. *Pattern Recognition*, 10(5-6):301–312, 1978.

[34] Leon D Harmon and Willard F Hunt. Automatic recognition of human face profiles. *Computer Graphics and Image Processing*, 6(2):135–156, 1977.

[35] Thomas David Heseltine. *Face recognition: two-dimensional and three-dimensional techniques*. PhD thesis, University of York, 2005.

[36] Future Internet. Future internet ppp. https://www.fi-ppp.eu/, 2015. [Web; accedido el 22-07-2017].

[37] Rabia Jafri and Hamid R Arabnia. A survey of face recognition techniques. *Jips*, 5(2):41–68, 2009.

[38] Takeo Kanade. *Computer recognition of human faces*. Birkhäuser Basel, 1977.

[39] Michael Kirby and Lawrence Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern analysis and Machine intelligence*, 12(1):103–108, 1990.

[40] K Krishna Prasad and PS Aithal. A conceptual study on user identification and verification process using face recognition techniques. *International Journal of Applied Engineering and Management Letters (IJAEML), ISSN(Applied)*, 1(1):6–17, 2017.

[41] Shang-Hung Lin, Sun-Yuan Kung, and Long-Ji Lin. Face recognition/detection by probabilistic decision-based neural network. *IEEE transactions on neural networks*, 8(1):114–132, 1997.

[42] Xiaoguang Lu. Image analysis for face recognition. *Personal notes*, page 36, 2003.

[43] Ajmal Mian, Mohammed Bennamoun, and Robyn Owens. An efficient multimodal 2d-3d hybrid approach to automatic face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 29(11), 2007.

[44] Ara V Nefian and Monson H Hayes. Hidden markov models for face recognition. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 5, pages 2721–2724. IEEE, 1998.

[45] Akitoshi Okumura, Takamichi Hoshino, Susumu Handa, Yugo Nishiyama, and Masahiro Tabuchi. Identity verification of ticket holders at large-scale events using face recognition. *Journal of Information Processing*, 25:448–458, 2017.

[46] Corporación Universitaria para el Desarrollo de Internet A.C. Cudi. http://www.cudi.edu.mx/#/, 2017. [Web; accedido el 22-07-2017].

[47] Alex Pentland, Baback Moghaddam, Thad Starner, et al. View-based and modular eigenspaces for face recognition. In *CVPR*, volume 94, pages 84–91, 1994.

[48] P Jonathon Phillips, Patrick J Flynn, Todd Scruggs, Kevin W Bowyer, Jin Chang, Kevin Hoffman, Joe Marques, Jaesik Min, and William Worek. Overview of the face recognition grand challenge. In *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on*, volume 1, pages 947–954. IEEE, 2005.

[49] P Jonathon Phillips, Patrick Grother, Ross Micheals, Duane M Blackburn, Elham Tabassi, and Mike Bone. Face recognition vendor test 2002. In *Analysis and Modeling of Faces and Gestures, 2003. AMFG 2003. IEEE International Workshop on*, page 44. IEEE, 2003.

[50] P Jonathon Phillips, Hyeonjoon Moon, Syed A Rizvi, and Patrick J Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE Transactions on pattern analysis and machine intelligence*, 22(10):1090–1104, 2000.

[51] P Jonathon Phillips, W Todd Scruggs, Alice J O'Toole, Patrick J Flynn, Kevin W Bowyer, Cathy L Schott, and Matthew Sharpe. Frvt 2006 and ice 2006 large-scale results. *National Institute of Standards and Technology, NISTIR*, 7408(1), 2007.

[52] M Janga Reddy. A survey of face recognition techniques. *International Journal of research in Computer Applications and Robotics*, 2(1):47–55, 2014.

[53] Marcelo Romero Huertas. *Landmark Localisation in 3D face data*. The University of York, 2010.

[54] Hans Schaffers, Nicos Komninos, Marc Pallot, Brigitte Trousse, Michael Nilsson, and Alvaro Oliveira. Smart cities and the future internet: Towards cooperation frameworks for open innovation. *The future internet*, pages 431–446, 2011.

[55] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[56] Muhammad Sharif, Farah Naz, Mussarat Yasmin, Muhammad Alyas Shahid, and Amjad Rehman. Face recognition: A survey. *Journal of Engineering Science and Technology Review*, 10(2):166–177, 2017.

[57] Linlin Shen and Li Bai. A review on gabor wavelets for face recognition. *Pattern analysis and applications*, 9(2-3):273–292, 2006.

[58] N Sivasankari, B Aravindhan, Suriya Kumar, et al. Smart security system with remote authentication. *Imperial Journal of Interdisciplinary Research*, 3(3), 2017.

[59] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman. Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture. In *2012 IEEE Symposium on Computers and Communications (ISCC)*, pages 000059–000066, July 2012.

[60] Ya Su. Robust video face recognition under pose variation. *Neural Processing Letters*, pages 1–15, 2017.

[61] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[62] F Tournus, N Blanc, A Tamion, M Hillenkamp, and V Dupuis. Synthesis and magnetic properties of size-selected copt nanoparticles. *Journal of Magnetism and Magnetic Materials*, 323(14):1868–1872, 2011.

[63] Matthew A Turk and Alex P Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991.Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 586–591. IEEE, 1991.

[64] Universal. Aumenta casos de robo de identidad: Fepade. http://www.eluniversal.com.mx/articulo/nacion/politica/2017/08/2/aumentan-casos-de-robo-de-identidad-fepade, 2017. [Web; accedido el 09-11-2017].

[65] Paul Viola and Michael J Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

[66] Laurenz Wiskott, Norbert Krüger, N Kuiger, and Christoph Von Der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on pattern analysis and machine intelligence*, 19(7):775–779, 1997.

[67] John Wright, Allen Y Yang, Arvind Ganesh, S Shankar Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE transactions on pattern analysis and machine intelligence*, 31(2):210–227, 2009.

[68] Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1):22–32, 2014.

[69] Wenyi Zhao, Rama Chellappa, P Jonathon Phillips, and Azriel Rosenfeld. Face recognition: A literature survey. *ACM computing surveys (CSUR)*, 35(4):399–458, 2003.