

PROYECTO DE GRADO

Análisis y Procesamiento de Huellas Dactilares de Niños Pequeños

FACULTAD DE INGENIERÍA, UDELAR



Autores:

Noelia LENCINA
Victoria MADRID
Nicolás TINTE

Tutores:

Jorge CORRAL
Alicia FERNÁNDEZ
Javier PRECIOZZI

18 de agosto de 2020

Agradecimientos

Queremos agradecer a nuestras familias y amigos por el continuo apoyo, cariño y aliento. A los tutores y a Guillermo por la paciencia y ayuda durante todo el proyecto, y por tenernos en cuenta constantemente. Y finalmente, a la Dirección Nacional de Identificación Civil por siempre recibirnos con los brazos abiertos, especialmente a Pablo, quien estuvo muy interesado en nuestro proyecto de inicio a fin.

*Soy quien soy
No preciso identificación
Sé bien de donde vengo y donde voy
Porque soy lo que soy, y no quien quieras vos.*

EL CUARTETO DE NOS

Resumen

Este proyecto se desarrolla en el marco del convenio existente entre la Facultad de Ingeniería de la Universidad de la República y la Dirección Nacional de Identificación Civil (DNIC) de la República Oriental del Uruguay. El mismo se enfoca en proponer mejoras al sistema biométrico actual de identificación por huellas dactilares para el caso de niños pequeños, manteniendo el hardware existente.

Esta mejora de desempeño se enfrentó desde distintas áreas, cada una relacionada a distintos módulos que componen un sistema biométrico, cubriendo los aspectos que hacen a la calidad, escalado como mecanismo para abordar el problema de *ageing*, fusión y compresión de imágenes de huellas dactilares.

Durante todo el proyecto se trabajó en estrecha colaboración con el equipo de desarrollo de la DNIC, definiendo los requerimientos y las interfaces necesarias para que la integración de los módulos que se desarrollaran fuera lo más directa posible.

En cada una de las cuatro áreas se realizaron distintos experimentos para los cuales se desarrollaron *scripts* en Python y Matlab. Además, en aquellos casos donde se logró obtener resultados prometedores, se implementaron en .NET los módulos de software necesarios para agregar las mejoras encontradas.

En particular, en el área de escalado se construyó un módulo integrable con el cual es posible transformar las huellas de niños según la edad de su toma, a tamaño de adulto, mejorando así el mecanismo actual que hace posible utilizar el software disponible para adultos con niños. También, se realizó un estudio en el que se valida el crecimiento isotrópico de las huellas dactilares, proporcionando una forma alternativa de transformar una huella en otra de mayor edad sin necesidad de contar con datos previamente marcados por expertos, siendo esto un aporte al estado del arte. También, se obtuvo una forma alternativa de estimar el factor de interpolación por rango de edad.

Por otro lado, en el área de fusión biométrica no se construyó software específico pero sí se realizó una recomendación para mejorar el algoritmo utilizado actualmente, exponiendo los resultados que justifican este cambio.

Para el área de calidad también se implementó un módulo para proveer una forma sencilla de comenzar a usar el algoritmo NFIQ2 para la medición de calidad de huellas dactilares. Además, se hizo un estudio donde se pudiera comprender la relación entre los distintos algoritmos de calidad disponibles: NFIQ1, NFIQ2 y el algoritmo del software propietario utilizado por la DNIC.

Finalmente, en el área de compresión se realizó un estudio comparativo entre los algoritmos de compresión de imágenes de huellas dactilares JPEG 2000 y WSQ. De este se obtuvieron resultados promisorios para el caso de 5 años, habiendo indicios de

que JPEG 2000 puede repercutir en una mejora de desempeño del sistema para este rango de edad. Para edades más chicas no se logró llegar a un resultado concluyente.

Palabras clave: Biometría, Huellas Dactilares, Fusión Biométrica, NFIQ, Identificación de Niños, Tratamiento de Imágenes, Ageing, Sistema Biométrico.

Tabla de Contenidos

1. Introducción	1
1.1. Definición del Problema	1
1.2. Descripción del Proyecto	1
1.2.1. Objetivo	1
1.2.2. Alcance	2
1.2.3. Organización del Documento	3
2. Marco Teórico	4
2.1. Revisión de Antecedentes y Contexto del Proyecto	4
2.2. Biometría	5
2.2.1. Operación de un Sistema Biométrico	6
2.2.2. Verificación <i>versus</i> Identificación	8
2.2.3. Aplicaciones de Biometría	8
2.2.4. Huellas Dactilares	8
2.2.5. Fuentes de Error de un Sistema Biométrico	9
2.3. Calidad	10
2.4. Escalado	11
2.5. Fusión	11
2.6. Compresión	12
3. Protocolo Experimental	13
3.1. Datos Disponibles	13
3.1.1. Base de Datos A	13
3.1.2. Base de Datos B	13
3.1.3. Base de Datos C	14
3.1.4. Base de Datos D	14
3.2. Medidas de Evaluación de Desempeño	15
3.2.1. Cálculo de Genuinos e Impostores	16
4. Calidad	18
4.1. Marco Teórico	18
4.1.1. NFIQ1	18
4.1.2. NFIQ2	19
4.1.3. Algoritmo Provisto por el Software Propietario Utilizado por la DNIC	19
4.1.4. Relaciones entre NFIQ1 y NFIQ2	19
4.2. Protocolo Experimental	21
4.2.1. Conjunto de Datos	22
4.2.2. Relación entre NFIQ2 y NFIQ1	22
4.2.3. Relación entre NFIQ1 y NFIQ2	24

4.2.4.	Relación entre el Algoritmo del Software Propietario Utilizado por la DNIC y NFIQ1	26
4.2.5.	Relación entre NFIQ1 y el Algoritmo del Software Propietario Utilizado por la DNIC	27
4.3.	Análisis de Resultados	28
4.4.	Propuesta de Implementación	28
5.	Escalado	29
5.1.	Marco Teórico	29
5.2.	Protocolo Experimental	32
5.2.1.	Escalado Lineal	33
5.2.2.	Crecimiento de Huellas Dactilares	33
5.3.	Análisis de Resultados	37
5.3.1.	Escalado Lineal	37
5.3.2.	Crecimiento de Huellas Dactilares	38
5.4.	Propuesta de Implementación	43
6.	Fusión	45
6.1.	Marco Teórico	45
6.1.1.	Fusión a Nivel de <i>Matching Score</i>	45
6.1.2.	Técnica de Fusión de Huellas Dactilares en la DNIC	46
6.2.	Protocolo Experimental	47
6.3.	Análisis de Resultados	49
6.4.	Propuesta de Implementación	52
7.	Compresión	54
7.1.	Marco Teórico	54
7.2.	Protocolo Experimental	55
7.2.1.	Comparación de Calidad	56
7.2.2.	Comparación de Desempeño	56
7.2.3.	Factor Efectivo de Compresión	56
7.3.	Análisis de Resultados	57
7.4.	Propuesta de Implementación	60
8.	Implementación	61
8.1.	Arquitectura del Sistema de la DNIC	61
8.2.	DLL de Escalado Lineal	63
8.2.1.	Relevamiento de Requerimientos	64
8.2.2.	Operaciones Implementadas	64
8.2.3.	Verificación y Validación	65
8.3.	DLL de Calidad	65
8.3.1.	Relevamiento de Requerimientos	66
8.3.2.	Operaciones Implementadas	66
8.3.3.	Verificación y Validación	67
8.4.	<i>Scripts</i> para Análisis	67

9. Conclusiones y Trabajo Futuro	69
9.1. Conclusiones	69
9.2. Trabajo Futuro	70
Bibliografía	72
Anexos	75
A. Riesgos	76
B. Actividades de Vinculación	78
C. Documentación de la DLL Escalado Lineal	80
C.1. Requerimientos	80
C.2. Operaciones Provistas	80
C.2.1. EscalarAByteArray	80
C.2.2. EscalarABitmap	81
C.2.3. EscalarGuardarByteArray	81
C.2.4. EscalarGuardarBitmap	82
C.2.5. EscalarAImage	83
C.2.6. EscalarAImageConFactor	84
D. Documentación de la DLL de Calidad	85
D.1. Requerimientos	85
D.2. Operaciones Provistas	85
D.2.1. ModoSimple	85
D.2.2. ModoBatch	85
E. Herramientas Utilizadas	87
E.1. Python	87
E.2. Pandas	87
E.3. Seaborn	88
E.4. Numpy	88
E.5. OpenCV	88
E.6. .NET	89
E.7. C#	89
E.8. Matlab	90
E.9. MCC SDK	90
E.10. OpenJPEG	91
F. Descomposición de una Matriz Afín	92

1. Introducción

1.1. Definición del Problema

Si bien el reconocimiento de personas utilizando huellas dactilares está ampliamente desarrollado para adultos, no sucede lo mismo en el caso de los niños. El software con que se cuenta actualmente, tanto para medir la calidad de las huellas como para realizar la identificación y verificación de identidades, opera bajo el supuesto de que la huella de entrada es de un adulto.

El reconocimiento de niños es una problemática mundial que impacta en varios ámbitos, por ejemplo:

- Identificación desde edad temprana.
- Prevención de secuestros.
- Seguimiento de tratamientos médicos (por ejemplo, vacunas).
- Identificación de niños desaparecidos.

La situación actual en Uruguay, donde se toman las huellas dactilares a los niños desde que nacen, hace que muchas veces la identificación de niños deba ser manual (lo cual requiere de mayores tiempos de espera) y se necesite, en muchos casos, conocer quién es el progenitor del niño para identificarlo, dato que no siempre está disponible. Esta es una debilidad que es de interés disminuir para que la verificación e identificación automática sea cada vez más universal.

1.2. Descripción del Proyecto

1.2.1. Objetivo

El objetivo de este proyecto es lograr mejorar el sistema actual de la Dirección Nacional de Identificación Civil (DNIC de ahora en más) en el área de reconocimiento biométrico utilizando huellas dactilares, particularmente enfocándose en el desempeño del sistema para niños pequeños.

Estas mejoras han de realizarse de forma tal que la DNIC, una vez finalizado el proyecto, cuente con algoritmos, recomendaciones y/o módulos de software concretos que puedan ser evaluados, adaptados y eventualmente integrados a su sistema biométrico actual, además de los correspondientes informes presentando los resultados de los distintos análisis llevados a cabo.

1.2.2. Alcance

El alcance del proyecto está definido por los siguientes puntos:

- Estudio del estado del arte en el área de reconocimiento biométrico utilizando huellas dactilares, tanto para adultos como para niños.
- Estudio de métodos de procesamiento de imágenes, especialmente aplicados a huellas dactilares.
- Estudio de métricas de calidad para huellas dactilares y su relación, junto con estudio de factibilidad técnica de implantar las métricas del estado del arte en la DNIC. En caso de que la factibilidad técnica se confirme, implementación del software necesario para posible integración en dicha institución.
- Estudio de métodos de *ageing* de huellas, como se define en la sección 5.1, mediante escalado, junto con implementación de al menos un método para posible integración en la DNIC.
- Estudio de métodos de fusión biométrica, como se define en la sección 2.5, especialmente aplicado a huellas dactilares, evaluando la implementación de software para posible integración en la DNIC.
- Estudio de métodos de compresión para imágenes de huellas dactilares, junto con estudio de factibilidad técnica y operacional de utilizar un nuevo método de compresión más moderno al usado actualmente en la DNIC. En caso de comprobarse la factibilidad, implementación del software necesario para posible integración en la DNIC.

En general, se busca obtener un mejor desempeño en la identificación de niños pequeños mediante la mejora de los distintos módulos del sistema actual de la DNIC, sin realizar cambios en el hardware con el que cuentan actualmente (ver sección 8.1 para más información del sistema).

El estudio de los métodos de fusión permitiría utilizar más información a la hora de identificar niños, lo cual es relevante cuando la información disponible suele no ser de muy buena calidad, como sucede con las huellas de niños pequeños.

Por otro lado, la mejora en el escalado de las huellas ayudaría al desempeño del software biométrico actual, el cual está diseñado para trabajar con huellas de adulto.

El estudio de los métodos de compresión y de métricas de calidad, en cambio, ayudaría a la DNIC a actualizarse en el campo, introduciendo los últimos avances. Esto a su vez permitiría tener un mejor conocimiento del desempeño del sistema actual y adaptabilidad para el futuro. El uso de nuevos métodos de compresión es especialmente relevante si se quisiera, en un futuro, adquirir huellas con mayor resolución (1000 dpi o 2000 dpi).

1.2.3. Organización del Documento

En el capítulo 2 se introducen los antecedentes a este proyecto y, además, el marco teórico general necesario para comprender los conceptos básicos de biometría junto con una breve introducción, motivación e importancia de las áreas tratadas. En el capítulo 3 se presentan las bases de datos utilizadas para los diferentes experimentos llevados a cabo, así como las medidas de evaluación empleadas. En los capítulos del 4 al 7 se desarrolla el marco teórico específico a cada área (calidad, escalado, fusión y compresión, respectivamente), junto con los experimentos realizados y los resultados obtenidos. En el capítulo 8 se describe el software desarrollado, tanto para realizar los experimentos como los módulos entregados a la DNIC. Finalmente, el capítulo 9 contiene las conclusiones y trabajo futuro.

2. Marco Teórico

En este capítulo se presentan los antecedentes del proyecto, así como una visión general del estado del arte referente a biometría con especial foco en huellas dactilares. También se incluye una breve introducción a las cuatro áreas específicas tratadas: calidad, escalado, fusión y compresión.

2.1. Revisión de Antecedentes y Contexto del Proyecto

La Dirección Nacional de Identificación Civil [13], perteneciente al Ministerio del Interior, es la institución responsable de la identificación de las personas físicas que viven en la República Oriental del Uruguay (de ahora en más Uruguay). Esto se hace mediante la obtención de la cédula de identidad.

Desde el año 2000 rige la ley Nro. 17.243 [42]. El artículo 78 de la misma establece la obligatoriedad de la obtención de la cédula de identidad para toda persona mayor de cuarenta y cinco días de edad con residencia permanente en el país. Esto significa que, desde el año 2000, todos los niños nacidos en Uruguay deben ser enrolados en el Registro de Identificación Civil con su nombre completo, fecha y lugar de nacimiento, foto, número de identificación y huella dactilar.

El orden en que se obtiene la información necesaria para la obtención del documento es: foto de la persona, luego sus huellas, y por último la firma (en caso de ser posible). En el caso de la obtención por vez primera, se toma una huella dactilar de cada dedo de ambas manos de la persona y se guarda en la base de datos de la DNIC.

En caso de ser un bebé menor de cuarenta y cinco días de edad, también se obtiene la huella del dedo pulgar de su acompañante, debido a la dificultad que implica poder identificar a un niño pequeño mediante su huella dactilar de forma automática. Este problema se da en general con huellas de niños y es la motivación que impulsa este proyecto.

En la situación más común, cada individuo es enrolado en el sistema cuando tiene menos de cuarenta y cinco días de edad, registrando las diez huellas dactilares de los dedos de ambas manos. Después de 5 años, su cédula de identidad debería ser renovada y, junto con ella, el juego de huellas dactilares guardado en el sistema. Este nuevo conjunto de huellas dactilares es el que comúnmente se usará para identificar a dicho individuo durante toda su vida, en el caso de que esta toma sea de buena calidad. En caso de no serla, en cada renovación se intentará obtener una nueva muestra completa de calidad suficiente.

La identificación de las personas físicas mediante el documento descripto anteriormente y la confrontación dactiloscópica se lleva a cabo desde el año 1978, año en que fue creada la DNIC. Este proceso consiste en comparar dos o más muestras biométricas para determinar si pertenecen a la misma identidad o no. En la literatura lo más común es nombrar a este proceso por su anglicismo *matching*, por lo cual se le hará referencia de esta forma de aquí en más.

Esta realidad brinda una situación particularmente beneficiosa para la realización de estudios con huellas de niños, ya que se cuenta con una gran cantidad de datos para analizar en edades tempranas, hecho que no es común en el resto del mundo. Además, estos datos contienen la evolución de las huellas de una misma persona.

Es así que en el marco del proyecto de grado “*Identificación de niños utilizando huellas digitales*” [7], se realizó una evaluación de huellas dactilares de niños y se mostró que mediante una interpolación simple, estas huellas tenían una precisión aceptable al momento de realizar un *matching* automático. Dicho proyecto fue una primera instancia para mejorar el desempeño de los procesos de verificación e identificación para niños pequeños, utilizando datos brindados por la DNIC y trabajando con dicha institución como parte interesada.

2.2. Biometría

En esta sección se presentan conceptos generales sobre biometría y sistemas biométricos según se definen en el libro “*Handbook of Multibiometrics*” [4].

La biometría es la ciencia de establecer una identidad basándose en atributos físicos o de comportamiento de un individuo, ya sea la huella, cara, voz, firma, entre otros.

Un sistema de identificación es una componente crítica en una gran cantidad de aplicaciones que le brindan servicios solamente a usuarios inscriptos (por ejemplo, reserva de vuelos comerciales). Dado que estos sistemas son cada vez más comunes, la necesidad de un sistema de identificación robusto y de confianza es más grande cada día.

La funcionalidad principal de un sistema de identificación es la determinación y/o verificación de la identidad de un individuo (o la identidad que reclama). Tradicionalmente, los métodos utilizados por estos sistemas son basados en conocimiento (por ejemplo, contraseñas) o basados en *tokens* (por ejemplo, tarjetas de identificación). Estas formas de representación de identidad se pueden perder fácilmente, compartir, manipular o robar.

La biometría ofrece una solución natural y confiable a ciertos aspectos de la gestión de identidad utilizando esquemas semi o completamente automáticos para reconocer identidades, basados en las características físicas y/o de comportamiento inherentes al individuo, estableciendo la identidad de acuerdo a “quién eres” y no

a “qué tienes” (tarjeta de identificación) o “qué recuerdas” (contraseña). También, la biometría puede ser utilizada en el contexto de una autenticación en dos pasos.

El uso de la biometría ofrece ciertas ventajas, las cuales no se pueden proveer solamente con *tokens* y contraseñas, tales como:

- **Reconocimiento negativo:** proceso por el cual un sistema determina que un cierto individuo está en efecto en el sistema aunque el individuo quizás lo niegue.
- **No repudio:** es una forma de garantizar que el individuo que accedió al sistema no pueda luego negar haberlo hecho.

Los diferentes atributos físicos o de comportamiento que pueden ser utilizados por un sistema biométrico son conocidos como: rasgos o características; indicadores; e identificadores o modalidades. Algunos ejemplos de estos pueden ser:

- | | |
|--------------------------------|--------------------------|
| ■ Huellas dactilares. | ■ Modo de andar. |
| ■ Rostro. | ■ Impresión de la palma. |
| ■ Geometría de la mano o dedo. | ■ Patrón de la voz. |
| ■ Iris. | ■ Oreja. |
| ■ Retina. | ■ Vena de la mano. |
| ■ Firma. | ■ ADN. |

2.2.1. Operación de un Sistema Biométrico

Un sistema biométrico es esencialmente un sistema de reconocimiento de patrones que adquiere datos biométricos de un individuo, extrae un conjunto de características destacadas de los datos, lo compara con los conjuntos de características guardados en la base de datos y ejecuta una acción basada en el resultado de la comparación.

Estos sistemas están compuestos por cuatro grandes módulos principales: sensor, módulo de evaluación de calidad y extracción de características, módulo de *matching* y la base de datos. La figura 2.1 muestra la organización de uno de estos sistemas.

Sensor

Se necesita de un lector o escáner biométrico adecuado para adquirir los datos biométricos de un individuo. Define la interfaz humano-máquina y la información disponible para el sistema, por lo cual es fundamental para el rendimiento del mismo.

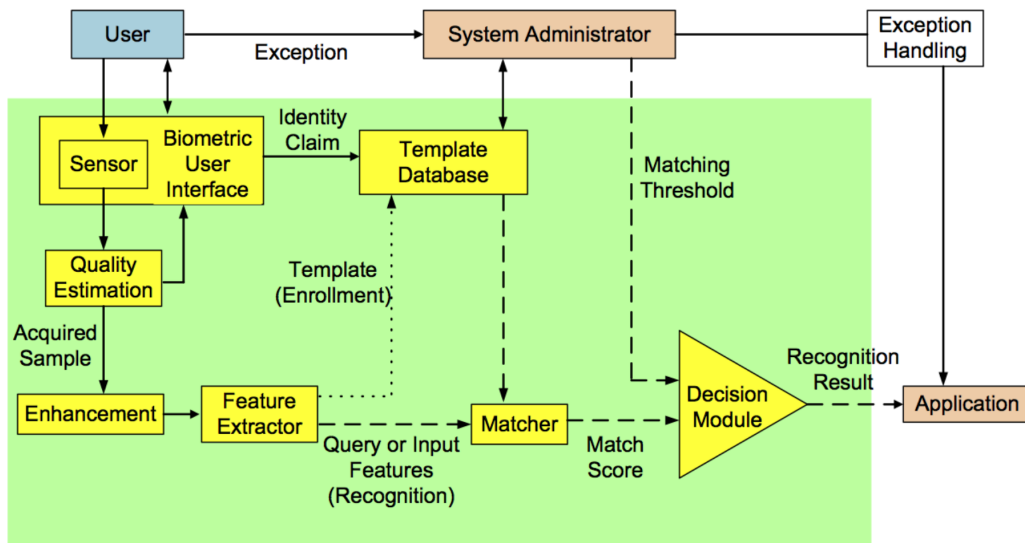


Figura 2.1: Organización de un sistema biométrico genérico. Imagen tomada del libro “*Encyclopedia of Biometrics*” [2].

Módulo de Evaluación de Calidad y Extracción de Características

Primero se evalúa la calidad de los datos biométricos adquiridos por el sensor para determinar su idoneidad para el procesamiento posterior. Típicamente, los datos adquiridos se someten a un algoritmo de mejora de señal para mejorar su calidad.

Luego, se procesa la información biométrica, se extrae un conjunto de características interesantes para representar el rasgo correspondiente, usualmente llamados *templates* y se guardan en la base de datos.

Módulo de *Matching*

Se comparan las características extraídas contra los *templates* guardados para generar el *matching score*. En este módulo también se debe tomar una decisión usando dichos *scores* para validar una identidad reclamada o proporcionar una clasificación de las identidades inscritas para identificar a un individuo.

Base de Datos

Este módulo actúa como repositorio de información biométrica. Durante el proceso de inscripción, las características extraídas de la muestra biométrica sin procesar se guardan en la base de datos con información biográfica (nombre, dirección, etc.) caracterizando al usuario.

2.2.2. Verificación *versus* Identificación

Dependiendo del contexto, un sistema biométrico puede operar en modo verificación o identificación.

Modo Verificación

Al operar en modo verificación, el sistema valida una identidad comparando la información biométrica capturada con su(s) *template(s)* biométricos guardados en la base de datos. El sistema hace una comparación uno a uno para determinar si es o no es. Este modo es usado típicamente para reconocimiento positivo y su objetivo principal es prevenir que múltiples personas usen la misma identidad.

Modo Identificación

El sistema reconoce un individuo buscando entre todos los *templates* de los usuarios en la base de datos para encontrar un *match*. El sistema hace una comparación uno a muchos.

Es un componente crítico en aplicaciones de reconocimiento negativo y su objetivo principal es prevenir que una persona use múltiples identidades, aunque también puede ser usado en reconocimiento positivo por conveniencia.

2.2.3. Aplicaciones de Biometría

La necesidad de autenticar a un usuario, es decir, establecer la identidad de un individuo con un alto grado de confianza es algo crítico en muchas aplicaciones. Por esto, la biometría juega un papel importante y está siendo cada vez más utilizada en una gran cantidad de aplicaciones, que se pueden agrupar en las siguientes categorías:

1. **Aplicaciones comerciales:** autenticación a una red, seguridad de información electrónica, *e-commerce*, cajeros automáticos, celulares, manejo de registros médicos, entre otras.
2. **Aplicaciones gubernamentales:** cédula de identidad, registro de reclusos, licencia de conducir, seguridad social, control fronterizo, pasaportes, entre otras.
3. **Aplicaciones forenses:** identificación de cuerpos, investigación criminal, determinación de paternidad, entre otros.

2.2.4. Huellas Dactilares

Los humanos han utilizado las huellas dactilares para identificación durante siglos debido a que la precisión en el *matching* de las mismas es alta.

Una huella dactilar, cuya formación se determina durante los primeros 7 meses de desarrollo fetal, es el patrón de crestas y valles en la superficie de la falange superior de los dedos.

Las crestas son las líneas en relieve que existen en ciertas zonas de la falange; usualmente son las visibles en negro en una huella impresa. Los valles, por el contrario, son las hendiduras presentes entre las crestas.

El patrón generado por las crestas y valles es único entre las personas. En este patrón, existen puntos donde las crestas terminan o se bifurcan llamados minucias. Usualmente, para comparar dos huellas dactilares, se utiliza un *matcher* de huellas dactilares que, internamente, extrae minucias de las huellas y las compara.

Hay aproximadamente ciento cincuenta tipos diferentes de minucias. En la figura 2.2 se ven algunas de las más comunes [8].



Figura 2.2: Ejemplos de tipos de minucias de una huella. Imagen tomada de “*Implementation of Minutiae Based Fingerprint Identification System Using Crossing Number Concept*” [8].

Hoy en día, el uso de las mismas en sistemas biométricos tiene un costo relativamente bajo y, como se mencionó anteriormente, la precisión es adecuada para el uso en un gran número de aplicaciones.

Aún así, existen ciertos problemas al utilizarlas. Por ejemplo, puede haber una pequeña fracción de la población cuyas huellas no sean adecuadas para identificación automática por factores genéticos, edad, razones ambientales o de ocupación, entre otros. También, cabe destacar que los sistemas biométricos a gran escala requieren de muchos recursos computacionales, especialmente en identificación.

2.2.5. Fuentes de Error de un Sistema Biométrico

Algunos de los desafíos comúnmente encontrados por los sistemas biométricos son:

- **Ruido en datos detectados:** por ejemplo una huella dactilar con una cicatriz. Puede generarse también por mal mantenimiento de los sensores

(por ejemplo, suciedad en sensores de huellas). Además, todo sensor electrónico tiene introduce ruido adicional debido a su funcionamiento.

- **Variaciones intra-clase:** son las variaciones observadas entre *templates* de una misma persona. Son causadas típicamente por un individuo que no interacciona correctamente con el sensor o por cambios en sus características biométricas con el paso del tiempo. Esto se puede solucionar guardando múltiples *templates* para una misma persona a lo largo del tiempo.
- **Similitudes entre-clase:** son las variaciones observadas entre *templates* de dos personas. Refiere a la superposición de espacios de características correspondientes a múltiples clases o individuos.
- **No universalidad:** es posible que el sistema biométrico no pueda adquirir datos biométricos significativos de un subconjunto de usuarios. Esto se da, por ejemplo, cuando a una persona le falta algún dedo.
- **Problemas de interoperabilidad:** la mayoría de los sistemas biométricos operan bajo el supuesto de que la información biométrica a comparar se obtiene usando el mismo sensor y por lo tanto lo restringen a la habilidad de comparar información biométrica originada por diferentes sensores.
- **Spoof attacks:** involucra la manipulación de los rasgos biométricos de alguien para evitar el reconocimiento o la creación de artefactos biométricos físicos para obtener la identidad de otra persona.
- **Otras vulnerabilidades:** vulnerabilidades generales en sistemas.

2.3. Calidad

La calidad de una muestra biométrica es una medida de su eficiencia al utilizarla con el objetivo de reconocer a un individuo. Esta puede ser influenciada por distintos factores. En el caso de huellas dactilares, factores ambientales como la luz o la humedad al momento de obtener la imagen la afectan directamente. También puede haber factores de comportamiento, como se suele dar en el caso de niños pequeños para los que se dificulta obtener una imagen de sus huellas de alta calidad debido al constante movimiento y la manipulación de sus manos pequeñas por parte de quien esté tomando sus huellas. Así como estos, existen otros factores como ser el sistema que se usa y las funcionalidades que brinda, la resolución del escáner (en caso de usar uno), entre otros.

La calidad de una muestra biométrica tiene un alto impacto en el desempeño de un sistema biométrico. Es por esto que es importante poder medirla y conocer los diferentes algoritmos existentes para hacerlo, para así poder realizar una decisión informada sobre cuál es el indicado para cada sistema y evolucionar junto con ellos.

2.4. Escalado

Cada vez es más común querer utilizar las huellas de niños pequeños en sistemas biométricos, pero no todos están preparados para el uso de las mismas.

En el caso particular de Uruguay, es de interés poder usar las huellas dactilares obtenidas cuando el individuo es niño, y emplearlas luego para realizar verificación 5 o 10 años más tarde. Por ejemplo, cuando se comparan las nuevas huellas obtenidas al momento de renovar la cédula de identidad con las huellas tomadas a los 5 años.

Una de las razones por la cual realizar esta comparación es un desafío es que la gran mayoría de los sistemas comerciales actuales están diseñados para trabajar con huellas de adultos. Por lo tanto, para poder realizar esto en dichos sistemas, es necesario realizar un pre-procesamiento a las huellas dactilares de niños, de forma tal que la imagen resultante sea compatible con el sistema.

2.5. Fusión

Como se explica en el libro “*Handbook of Multibiometrics*” [4], considerar los datos de solamente una característica biométrica en los sistemas biométricos tiene el problema de que el desempeño de los mismos se ve limitado por las cualidades intrínsecas de la característica siendo analizada. Por ejemplo, en un escenario en el que se quisiera enrolar a todas las personas del planeta en un solo sistema biométrico mundial que operara en modo de identificación, y se usara como modalidad de enrolamiento la cara del individuo, se tendría una cota superior de desempeño dada por la cantidad de gemelos. De forma similar, si se usara la voz, el desempeño estaría limitado por la cantidad de personas que no pueden hablar claramente. Esto lleva a concluir que los sistemas biométricos que emplean una sola característica están limitados por su propio diseño.

Una forma de mitigar estas limitaciones es usando más de una fuente de información y fusionándolas. Esto puede hacerse mediante el uso de más de una característica biométrica (cara y voz, por ejemplo) o usando más de una muestra de la misma característica (por ejemplo, múltiples imágenes de la cara o las huellas dactilares de más de un dedo).

Lo que logra la fusión biométrica es expandir el espacio que representa a los individuos, aumentando la cantidad de personas que pueden ser enroladas en el sistema. Además, este tipo de sistemas suele ser más confiable gracias a la disponibilidad de mayor cantidad de evidencia.

Los sistemas que funcionan fusionando varias fuentes de información biométrica para la toma de decisión, ya sea de la misma característica o no, se conocen como sistemas multibiométricos. Dentro de los sistemas multibiométricos existen otras varias categorías, o subtipos, dependiendo de la información que utilizan. Uno de estos subtipos es el llamado sistema multimodal, que considera la

información de más de una característica biométrica (por ejemplo: rostro y voz). Por otro lado, uno que considera la información de más de una muestra de una misma característica biométrica de un individuo para la toma de decisión, es denominado sistema multiinstancia o multiunidad. Con este tipo de sistemas (los multiinstancia o multiunidad) comúnmente no se necesita que se modifique el hardware utilizado, ni se requiere que se desarrolle nuevo software para la extracción de características o para el *matching*, por lo que suelen ser menos costosos de implementar.

Por último, es importante destacar que en casos donde la calidad de una muestra puede ser mala, tener la información de más de una muestra puede ayudar a mejorar la confiabilidad del sistema de forma considerable.

2.6. Compresión

Históricamente y a nivel mundial, las imágenes de huellas dactilares han sido intercambiadas y almacenadas con una resolución de 500 dpi (*dots per inch*, puntos por pulgada), lo cual equivale a 19,7 píxeles por milímetro [9]. Este es el caso de la DNIC, donde todos los escáneres disponibles actualmente para la captura de huellas dactilares son de esta resolución.

El uso de imágenes de 500 dpi hace que las huellas de niños pequeños, y especialmente aquellos menores de 1 año [33], sean de muy baja fidelidad resultando de poca utilidad para los procesos automáticos de identificación y verificación. A su vez, en el proceso de compresión también se pierde calidad en las huellas por el simple hecho de que se pierde información.

En el caso de huellas dactilares de 500 dpi, el algoritmo de compresión estándar utilizado es WSQ. Este algoritmo fue desarrollado específicamente para funcionar con esta resolución. En la actualidad existe otro algoritmo de compresión, llamado JPEG 2000, que ha sido desarrollado sin restricción de resolución y que se está usando para comprimir huellas obtenidas con resolución de 1000 dpi. Se profundizará sobre ambos algoritmos en el capítulo 7.

Al no tener restricción de resolución, JPEG 2000 podría ser usado para huellas de 500 dpi, y su comportamiento podría ser distinto al de WSQ. Esto podría resultar en un desempeño distinto para huellas de niños ya que se sabe que con WSQ no se logran desempeños muy buenos para este sector de la población. Por ende, es importante investigar si hay alguna opción a la hora de comprimir que se acople mejor a estas huellas, para así aumentar el desempeño de los sistemas y, también, para estar preparados en caso de disponer de escáneres de 1000 dpi o incluso 2000 dpi, donde el algoritmo de compresión utilizado para huellas de 500 dpi puede no ser óptimo.

3. Protocolo Experimental

Con el objetivo de brindar sugerencias para mejorar el sistema biométrico de huellas dactilares utilizado actualmente en la DNIC, así como también tener informes específicos de las distintas áreas de trabajo, se llevaron a cabo experimentos en cada una de las cuatro áreas cubiertas por el presente informe: calidad, escalado, fusión y compresión.

Utilizando varias bases de datos de huellas dactilares, se hicieron distintos protocolos experimentales cuyos resultados fueron analizados mediante el uso de medidas de evaluación acordes.

3.1. Datos Disponibles

Para realizar este proyecto se contó con más de una base de datos. Las mismas se utilizaron para llevar a cabo las pruebas durante desarrollo y determinar el desempeño de las mejoras propuestas.

3.1.1. Base de Datos A

Esta base de datos, utilizada en el artículo “*Fingerprint Biometrics From Newborn to Adult: A Study From a National Identity Database System*”[33], cuenta con las huellas de 9.785 individuos. Para cada individuo se cuenta, a su vez, con dos juegos de huellas: un juego obtenido exactamente a los 5 años de edad y otro obtenido a los 15 años aproximadamente teniendo así, en general, una diferencia de al menos 10 años entre tomas. Cabe destacar que todas las imágenes de esta base de datos están comprimidas con pérdida.

El proceso de renovación en la DNIC requiere solamente que se tomen las huellas del dedo pulgar y del índice de cada mano, por esto la mayoría de los pares de este conjunto de datos se armaron usando solo esos dedos. Con esto se obtienen aproximadamente unos 30.000 pares genuinos (29.867 precisamente). Además, muchas de las muestras en este conjunto de datos son basadas en papel, es decir tomadas originalmente como impresión de tinta sobre papel y luego escaneadas para su digitalización.

La tabla en la figura 3.1 muestra la distribución de edades al momento de adquisición de las muestras.

3.1.2. Base de Datos B

Se cuenta con un segundo conjunto de datos con 7.080 identidades de niños de hasta 6 años de edad, para las cuales se cuenta con al menos un par de muestras

Edad	Huellas
5	29.867
14	16
15	1.040
16	14.127
17	8.007
18	7.624
19	4.047
20	168

Figura 3.1: Distribución de cantidad de huellas por edad al momento de adquisición de las muestras de la base de datos A.

de un mismo dedo. Cabe destacar que todas las imágenes de esta base de datos también están comprimidas con pérdida.

3.1.3. Base de Datos C

Esta base de datos es simplemente la base de datos A y B mencionadas anteriormente en conjunto, obteniendo así un total de 150.927 imágenes de huellas dactilares.

3.1.4. Base de Datos D

Esta base de datos corresponde a la base de datos *301a* generada por el “*National Institute of Standards and Technology*” [29] (NIST de aquí en más). Es parte de una colección de datos biométricos recolectados durante un desafío denominado “*Nail to Nail Fingerprint Challenge Dry Run*” organizado por IARPA (*Intelligence Advanced Research Projects Activity*).

La misma cuenta con imágenes de los diez dedos de 24 identidades, todas en formato PNG y de 500 dpi. Además, todas las huellas pertenecen a adultos. Es importante el hecho de que las huellas nunca fueron comprimidas, lo cual es fundamental para los experimentos realizados utilizando esta base de datos.

Para cada identidad, se cuenta con catorce tomas de los diez dedos, donde las primeras dos tomas (identificadas como A y B) fueron obtenidas usando escáneres ya probados y en producción, y además fueron sacadas por profesionales del “*Federal Bureau of Investigation*” (FBI de ahora en más) con entrenamiento en captura de huellas. Las otras tomas fueron sacadas con escáneres experimentales y/o por personas no entrenadas para la captura de huellas.

3.2. Medidas de Evaluación de Desempeño

Para poder determinar si los algoritmos propuestos realmente son mejores a los existentes, es fundamental contar con alguna metodología que permita comparar el desempeño de los mismos.

En el caso de los sistemas biométricos, esta metodología se basa comúnmente en el uso de las siguientes medidas de desempeño:

- **True Acceptance Rate (TAR):** tasa que mide la proporción de veces que el sistema realiza una identificación correcta. Esta tasa se corresponde con la noción de verdadero positivo.
- **False Acceptance Rate (FAR):** tasa que mide la proporción de veces que el sistema identifica erróneamente a una persona. Se corresponde con la noción de falso positivo. En el área de biometría, el término *False Match Rate* (FMR) también es muy usado en la literatura para referirse a esta tasa.
- **False Rejection Rate (FRR):** tasa que mide la proporción de veces que el sistema rechaza una identificación erróneamente. Esta tasa se corresponde con la noción de falso negativo y también puede ser llamada como *False Non-Match Rate* (FNMR) en el área de biometría.

Estas tasas se calculan como:

$$FAR = \frac{\#s(q_i, q_j) > \delta \quad t.q. \quad id(q_i) \neq id(q_j)}{M} \quad (3.1)$$

$$FRR = \frac{\#s(q_i, q_j) < \delta \quad t.q. \quad id(q_i) = id(q_j)}{N} \quad (3.2)$$

$$TAR = 1 - FRR \quad (3.3)$$

donde q_i y q_j son dos muestras biométricas distintas; s es la función que calcula el *matching score* al confrontar ambas muestras; δ es el umbral de decisión para decidir si el *matching* se toma como positivo (el *matching score* es mayor al umbral) o como negativo (el *matching score* es menor al umbral), y varía entre 0 y el mayor de los *matching scores* obtenidos según s ; N es la cantidad de *matchings* genuinos para la base de datos considerada (en el caso de estar estudiando el desempeño por rango de edad, por ejemplo, se han de tomar los *matchings* genuinos para cada rango por separado); y M la cantidad de *matchings* impostores para la misma base de datos.

Genuinos refiere a *matchings* biométricos donde la muestra de consulta y la de referencia pertenecen a la misma identidad. Impostores, por otro lado, son aquellos en los que la muestra de consulta y la de referencia son de identidades distintas.

Por otra parte, estas medidas pueden graficarse de forma que se logre visualizar fácilmente el desempeño del sistema a medida que se varía el umbral δ . A esta gráfica se la denomina *Receiver Operating Characteristics (ROC)*, y se construye colocando en el eje de las abscisas el valor de FAR y en el de las ordenadas el de TAR para cada valor de δ considerado. Cuanto más cercano a 1 es el área que queda por debajo de la curva graficada, mejor es el desempeño del sistema para ese rango de δ .

En la figura 3.2 se puede ver un ejemplo de curva ROC para un buen desempeño (color anaranjado) y para un mal desempeño (color rojo).

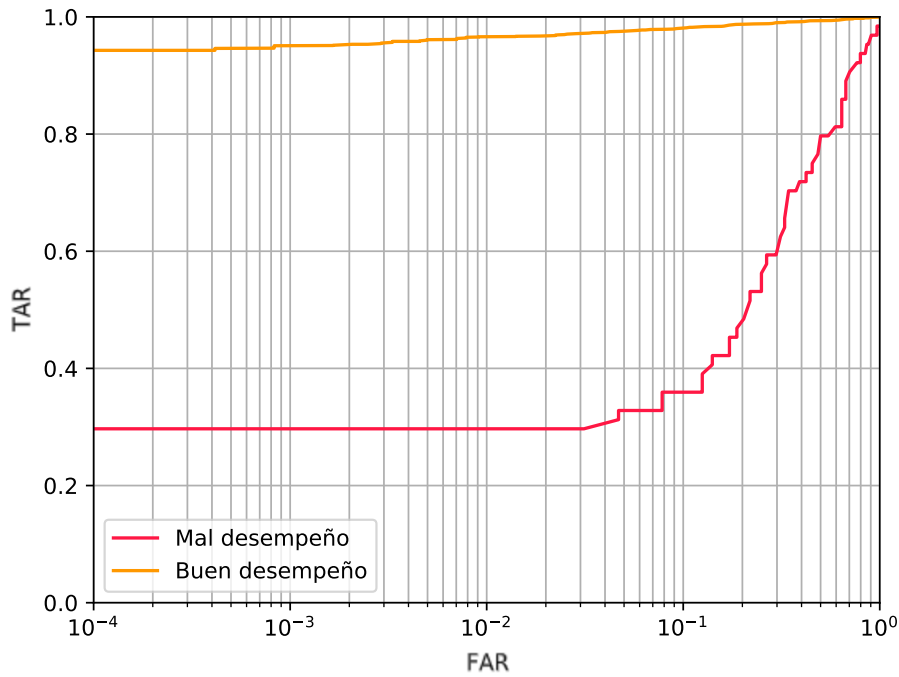


Figura 3.2: Ejemplos de curvas ROC.

3.2.1. Cálculo de Genuinos e Impostores

Para computar las medidas de evaluación de desempeño, es imprescindible tener los datos de *matchings* genuinos y *matchings* impostores. Estos datos a su vez deben, idealmente, ser fieles a los datos que se pueden encontrar en un sistema en producción.

Por esto, en este proyecto, los *matchings* genuinos se generaron tomando aquellas huellas para las que existiera una segunda toma de la misma identidad con la cual se pudiera confrontar. Los impostores, por otro lado, se generaron tomando, para cada huella a la que se le haya podido generar un genuino, una segunda huella del mismo dedo, de otra identidad al azar y de mayor edad. Esto genera la misma cantidad de genuinos que de impostores. Si bien sería posible generar una cantidad

mucho mayor de impostores, se optó por inicialmente trabajar con esa cantidad por temas puramente relacionados al desempeño del hardware disponible para realizar los análisis.

4. Calidad

En este capítulo se trata el área de calidad. Se presenta el marco teórico sobre el cual se basó el experimento realizado y luego se detalla el protocolo experimental utilizado, así como el análisis de resultados y la propuesta de implementación en base a ellos. Dicho experimento trabaja con tres medidas de calidad alternativas: NFIQ1, NFIQ2 y el algoritmo de calidad del software propietario disponible en la DNIC. Como resultado final, se logró construir funciones que mapean las clases de calidad de NFIQ1 con los valores de NFIQ2, y viceversa. Lo mismo se logró para NFIQ1 con el algoritmo propietario.

4.1. Marco Teórico

Hallar la calidad de una imagen de una huella dactilar es de vital importancia para los sistemas biométricos, en especial en el proceso de enrolamiento ya que en este se tiene que asegurar que la imagen provista tenga una calidad lo suficientemente buena.

La calidad define la precisión esperada al utilizar dicha imagen en una verificación de identidad (en términos de lograr obtener una baja tasa de error). La calidad reportada depende del algoritmo utilizado para extraer características de la huella.

Entre los diferentes algoritmos existentes, resaltan aquellos creados por el NIST: *NIST Fingerprint Image Quality* (NFIQ1) y su versión actualizada NFIQ2. En el marco de este trabajo también se utilizó el algoritmo provisto por software propietario utilizado por la DNIC.

4.1.1. NFIQ1

Creado por el NIST como parte del *NIST Biometric Image Software* en agosto del 2004, NFIQ1 es un algoritmo de código abierto para medir la calidad de imágenes de huellas dactilares de forma independiente del software de verificación utilizado.

Este algoritmo, utilizado hoy en día en la DNIC, clasifica la calidad en cinco clases, numeradas del 1 al 5, donde la clase 1 representa una “excelente” calidad y la clase 5 una calidad “muy mala”. A grandes rasgos, el algoritmo está basado en una red neuronal que intenta predecir la clase de la imagen a partir de once características extraídas de la misma. En el artículo “*Fingerprint Image Quality*” [15] se puede encontrar la explicación completa del mismo.

4.1.2. NFIQ2

Su desarrollo se inició en el 2011 por el NIST y surge de los avances tecnológicos en el área desde que su primera versión (NFIQ1) fuera publicada.

Las principales diferencias con su predecesor son que éste cuenta con un diseño modular, una mejora en la velocidad de ejecución, así como una mejor precisión para estimar la calidad. Sin embargo, la mayor diferencia es que este cuenta con un mayor rango de valores, es decir, tiene mayor sensibilidad. Este rango va de 0 a 100, siendo 0 una calidad “muy mala” y 100 una calidad “excelente”.

4.1.3. Algoritmo Provisto por el Software Propietario Utilizado por la DNIC

En la documentación proporcionada por el software propietario utilizado por la DNIC no se puede encontrar una explicación sobre dicho algoritmo. Solo se proporciona la interfaz del método que calcula la calidad utilizando el algoritmo, que retorna un valor entre 0 y 256, donde 0 representa imágenes con una calidad “muy mala” y 256 aquellas con una “excelente” calidad.

4.1.4. Relaciones entre NFIQ1 y NFIQ2

En el artículo “*Fingerprint Quality: Mapping NFIQ1 Classes and NFIQ2 Values*” [17] se realiza un estudio para encontrar la relación entre las clases de calidad en NFIQ1 y el rango de valores de calidad en NFIQ2. Este se realizó con una base de datos de huellas dactilares obtenidas con un escáner de 500 dpi, la cual cuenta con 265.321 identidades diferentes, que producen un total de 421.388 imágenes. Las mismas pueden ser divididas en tres grandes grupos, 0 a 17 años (niños), 18 a 25 años (adultos) y 65 a 98 años (adultos mayores).

En dicho estudio se plantea el problema de obtener la relación entre NFIQ2 y NFIQ1 como encontrar una función $f(NFIQ2) = NFIQ1$, el cual puede ser visto como un problema de clasificación enunciado de la forma: “dada una imagen que produce un valor $x \in [0; 100]$ de calidad NFIQ2, hallar cuál es la clase $c = \{1, 2, 3, 4, 5\}$ más probable de NFIQ1 que produciría dicha imagen”.

En la figura 4.1 se puede observar la gráfica que obtuvieron los autores al utilizar un clasificador bayesiano óptimo¹ para resolver el problema antes mencionado. De dicha gráfica, se logra definir la función f como:

¹Un clasificador bayesiano óptimo es aquel que elige la clase con mayor probabilidad de ocurrencia para una muestra dada.

$$f(NFIQ2) = NFIQ1 = \begin{cases} 5 & \text{si } 0 < NFIQ2 < 5 \\ 3 & \text{si } 6 < NFIQ2 < 38 \\ 1 & \text{si } 39 < NFIQ2 < 100 \end{cases}$$

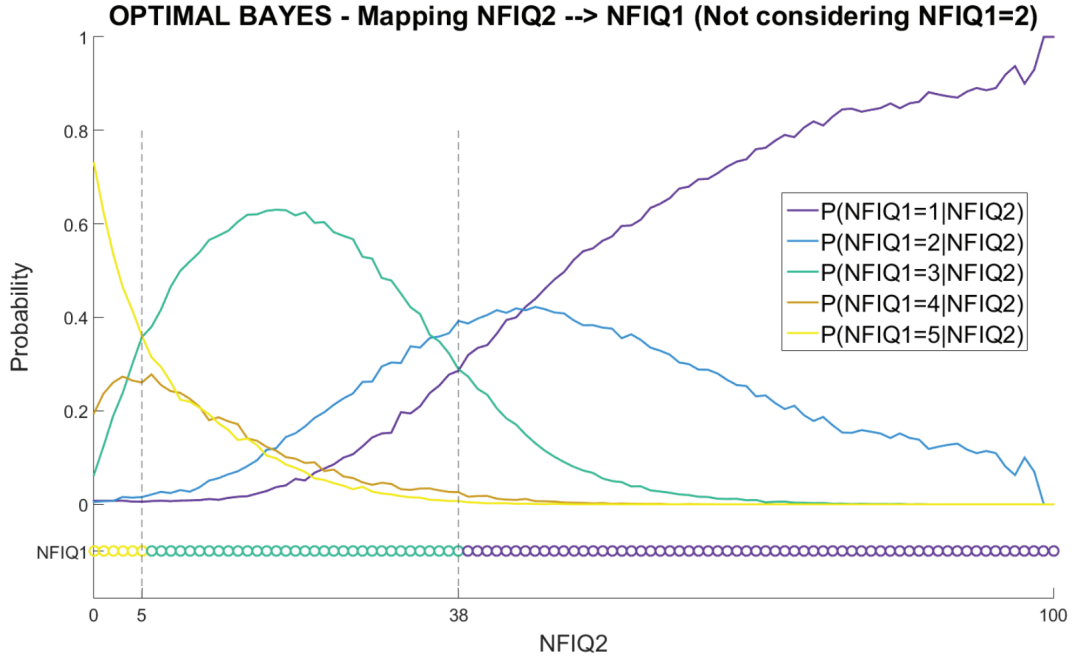


Figura 4.1: Relación entre NFIQ2 y NFIQ1 obtenida con un clasificador bayesiano óptimo. Imagen obtenida de “*Fingerprint Quality: Mapping NFIQ1 Classes and NFIQ2 Values*” [17].

Esto indica que, sin perder fiabilidad, se pueden agrupar las clases 4 y 5, las cuales representan las imágenes de muy mala calidad y, también así, se pueden agrupar las clases 1 y 2 que representan las imágenes de buena calidad.

De forma similar se plantea el problema para encontrar la relación entre NFIQ1 y NFIQ2: “dada una imagen que produce una clase c de NFIQ1, hallar cuál es el valor x más probable de NFIQ2 que produciría dicha imagen”.

En la figura 4.2 se puede observar la gráfica obtenida para esta relación, y de ella se desprende lo siguiente:

$$P(NFIQ2) = \begin{cases} N(\mu = 63, 1; \sigma = 14, 4) & \text{si } NFIQ1 = 1 \\ N(\mu = 52, 5; \sigma = 15, 7) & \text{si } NFIQ1 = 2 \\ N(\mu = 27, 0; \sigma = 14, 6) & \text{si } NFIQ1 = 3 \\ G(k = 0, 98; \theta = 13, 6) & \text{si } NFIQ1 = 4 \\ G(k = 0, 70; \theta = 9, 9) & \text{si } NFIQ1 = 5 \end{cases}$$

Donde N indica una distribución Normal ² y G una distribución Gamma ³.

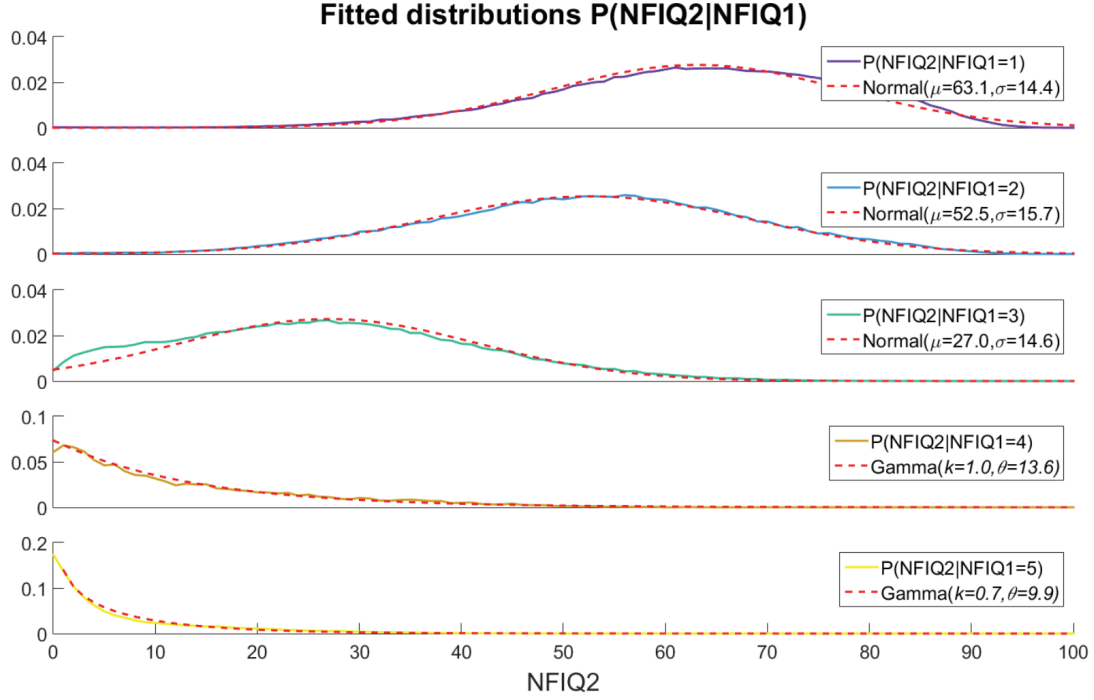


Figura 4.2: Distribuciones estimadas para $P(NFIQ2 = x|NFIQ1 = c)$ aproximadas por distribuciones conocidas. Imagen obtenida de “*Fingerprint Quality: Mapping NFIQ1 Classes and NFIQ2 Values*” [17].

4.2. Protocolo Experimental

Dado el interés de la DNIC de comenzar a utilizar un algoritmo con mayor sensibilidad y la voluntad de conocer como se vinculan los posibles algoritmos con el utilizado actualmente (NFIQ1), se realizó un experimento para obtener la relación entre estos de forma tal que sea fácil calcular la correspondencia entre ellos.

Para esto, se decidió reproducir el protocolo experimental utilizado en “*Fingerprint Quality: Mapping NFIQ1 Classes and NFIQ2 Values*” [17] sobre otro conjunto de datos, ampliando el estudio para obtener la relación entre NFIQ1 y el algoritmo del software propietario utilizado por la DNIC.

²La gráfica de su función de densidad tiene forma de campana y es simétrico respecto a un parámetro estadístico determinado. Dicha función de densidad queda definida como:

$$\Phi_{\mu,\sigma^2}(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}, x \in R$$

³Siendo k el parámetro que indica la forma de la distribución y θ el que indica la escala. La función de densidad se define como:

$$f(x; k; \theta) = \frac{x^{k-1} \cdot e^{-\frac{x}{\theta}}}{\theta^k \cdot \Gamma(k)}, \text{ para } x, k, \theta > 0 \text{ y donde } \Gamma \text{ es la función Gamma evaluada en } k.$$

4.2.1. Conjunto de Datos

Para realizar esta prueba se utilizaron las imágenes pertenecientes a la base de datos C (sección 3.1.3).

Para cada una de estas imágenes, en caso de ser necesario, se las escaló a tamaño de adulto de la misma forma que se explicará en la sección 5.2.1. Luego, se obtuvo su calidad con los diferentes algoritmos, obteniendo las distribuciones que se pueden observar en las figuras 4.3, 4.4 y 4.5. Las distribuciones para NFIQ2 y para el algoritmo del software propietario utilizado por la DNIC se presentan como una gráfica de barras aditiva, para poder observar la cantidad de muestras correspondientes a cada clase de NFIQ1 para los distintos valores.

Cabe mencionar que, de este conjunto, 144.046 imágenes son válidas para realizar este estudio, ya que para las restantes alguno de los algoritmos falla.

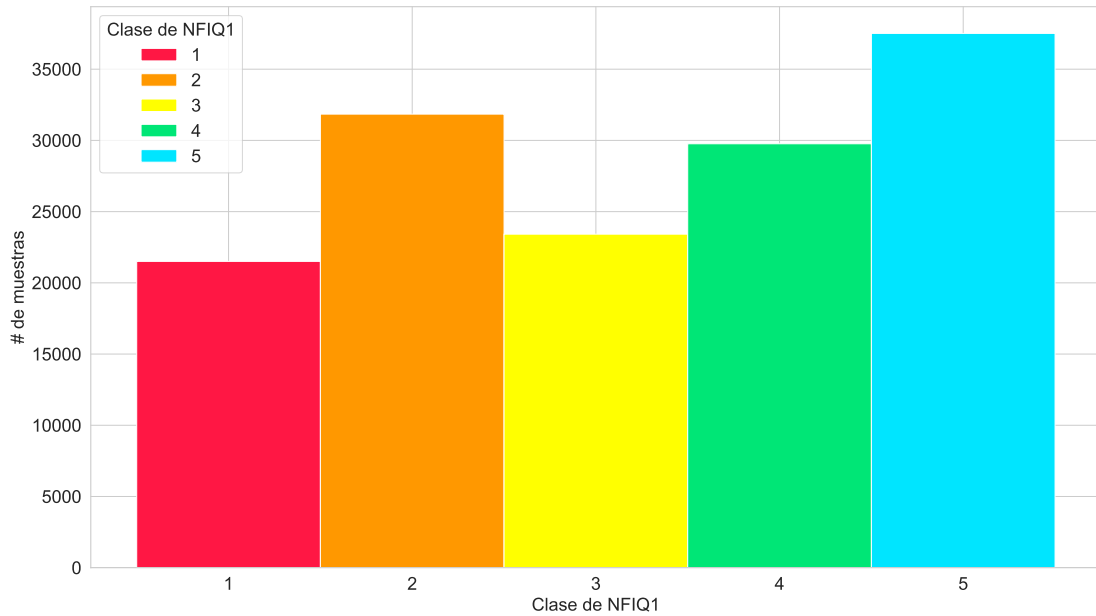


Figura 4.3: Distribución de las muestras para NFIQ1.

4.2.2. Relación entre NFIQ2 y NFIQ1

Como se mencionó anteriormente, en “*Fingerprint Quality: Mapping NFIQ1 Classes and NFIQ2 Values*” [17] se plantea este problema como un problema de clasificación enunciado como: “dada una imagen que produce un valor $x \in [0; 100]$ de calidad NFIQ2, hallar cuál es la clase $c = \{1, 2, 3, 4, 5\}$ más probable de NFIQ1 que produciría dicha imagen”.

Al tener ambas calidades para todas las imágenes, se optó por utilizar el Teorema de Bayes para resolver el problema antes mencionado. Para este

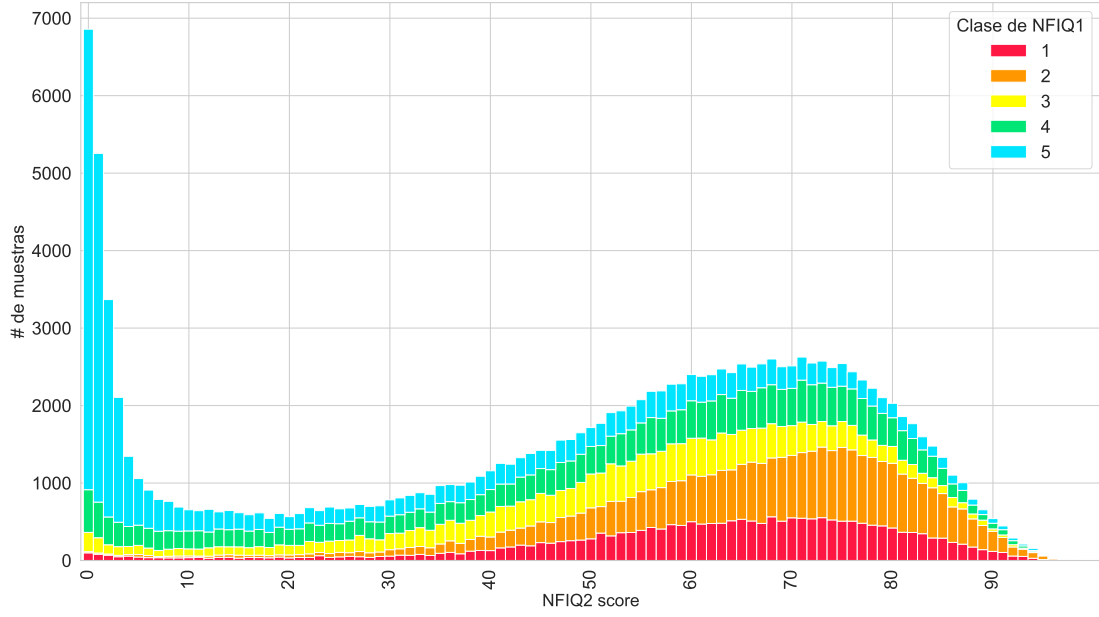


Figura 4.4: Distribución de las muestras para NFIQ2.

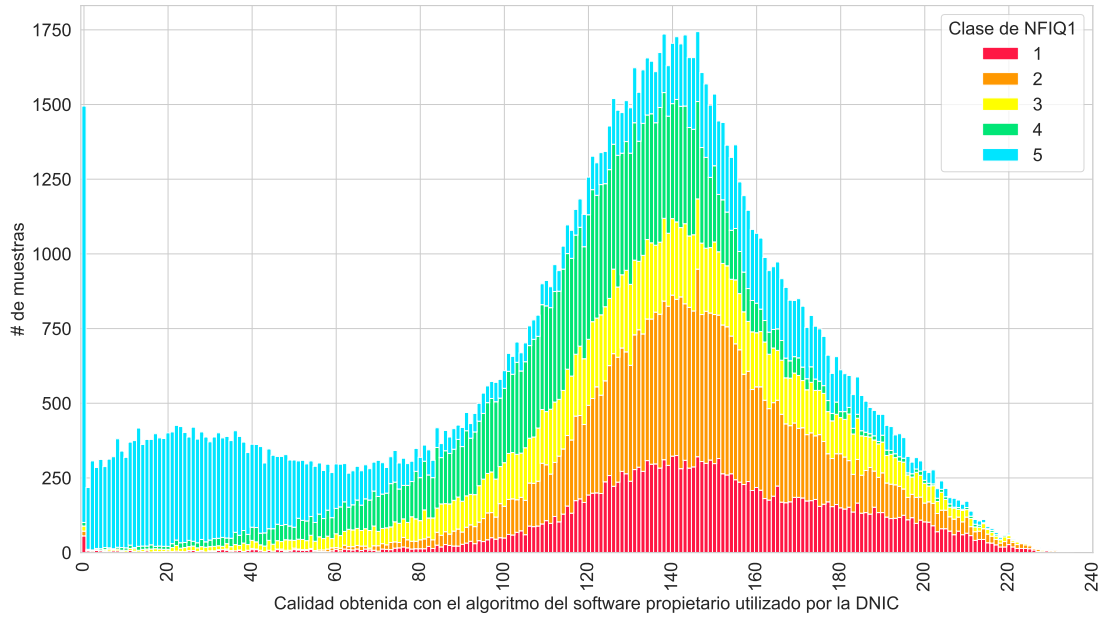


Figura 4.5: Distribución de las muestras para el algoritmo del software propietario utilizado por la DNIC.

problema, el teorema plantea la siguiente ecuación:

$$P(NFIQ1 = c | NFIQ2 = x) = \frac{P(NFIQ2 = x | NFIQ1 = c)P(NFIQ1 = c)}{P(NFIQ2 = x)}$$

En la figura 4.6 se puede observar la gráfica obtenida de aplicar dicha ecuación para todo $x \in [0; 100]$, es decir, se pueden ver las cinco probabilidades condicionales $P(NFIQ1 = c | NFIQ2)$ para cada valor de NFIQ2.

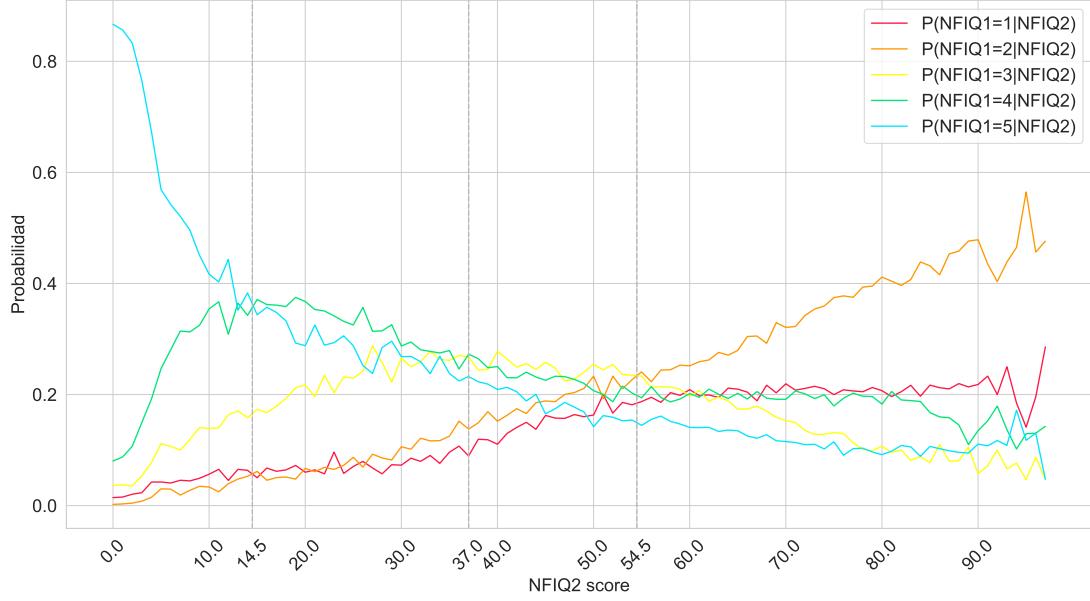


Figura 4.6: Probabilidades condicionales para $P(NFIQ1 = c|NFIQ2)$ para cada valor de NFIQ2.

De esto se puede obtener la siguiente función $f(NFIQ2) = NFIQ1$:

$$f(NFIQ2) = NFIQ1 = \begin{cases} 5 & \text{si } 0 \leq NFIQ2 < 14 \\ 4 & \text{si } 15 < NFIQ2 < 37 \\ 3 & \text{si } 38 < NFIQ2 < 54 \\ 2 & \text{si } 55 < NFIQ2 \leq 100 \end{cases}$$

En comparación a la función obtenida en el artículo, se puede observar que el rango de valores de NFIQ2 que da como resultado una imagen de mala calidad es mayor. Esto posiblemente se deba a la calidad de las imágenes de la base de datos utilizada, ya que la mayoría corresponden a niños en el caso de este estudio.

4.2.3. Relación entre NFIQ1 y NFIQ2

Dado que este problema se plantea de igual forma que el anterior, también se utilizó el Teorema de Bayes para resolverlo, en este caso con la siguiente ecuación:

$$P(NFIQ2 = x|NFIQ1 = c) = \frac{P(NFIQ1 = c|NFIQ2 = x)P(NFIQ2 = x)}{P(NFIQ1 = c)}$$

En la figura 4.7 se pueden observar las gráficas obtenidas al aplicar dicha ecuación para cada clase c de NFIQ1. En estas gráficas también se muestra con una línea punteada el máximo de cada una de ellas.

En una primera instancia, se intentó obtener una aproximación para estas gráficas por distribuciones conocidas, pero no se logró obtener aproximaciones lo

suficientemente buenas. Dado esto, se procedió a aproximar las mismas utilizando *Gaussian Mixture Models*⁴ (GMM de aquí en más), con lo que se logró una mejor aproximación.

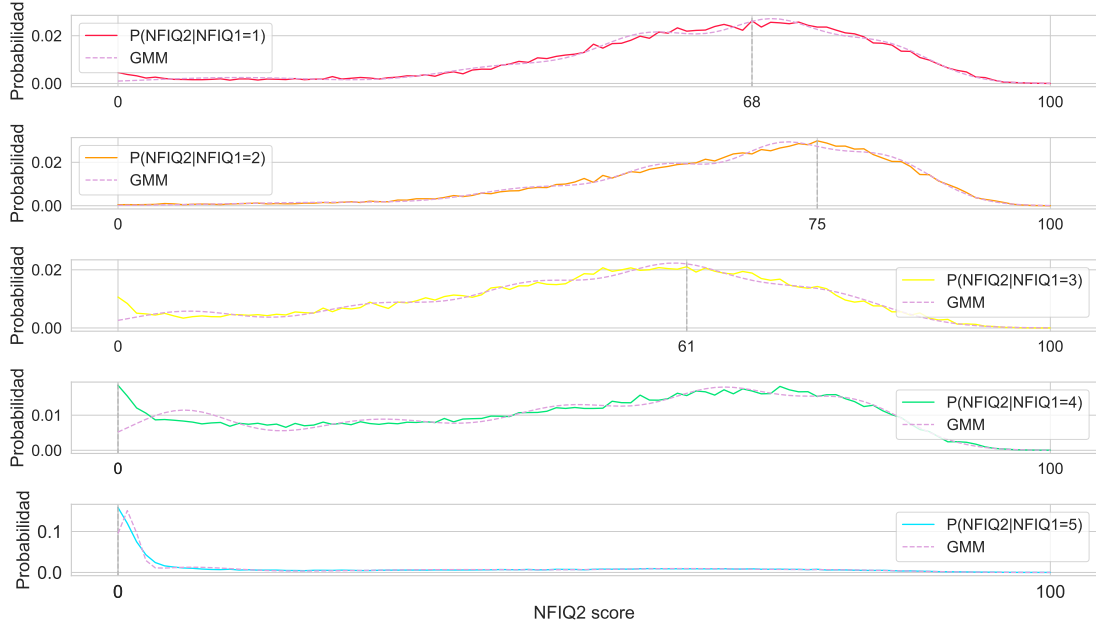


Figura 4.7: Probabilidades condicionales para $P(NFIQ2 = x|NFIQ1)$ para cada clase de NFIQ1.

De esto se obtiene:

$$P(NFIQ2) = \begin{cases} GMM(\mu = \{54, 9; 79, 7; 12, 0; 40, 7; 67, 3\}; \\ \sigma = \{29, 1; 37, 3; 82, 1; 54, 7; 26, 9\}) & si \ NFIQ1 = 1 \\ GMM(\mu = \{73, 6; 28, 1; 48, 9; 62, 8; 83, 2\}; \\ \sigma = \{16, 8; 169, 2; 51, 3; 26, 3; 24, 1\}) & si \ NFIQ1 = 2 \\ GMM(\mu = \{58, 3; 27, 1; 73, 7; 43, 5; 6, 9\}; \\ \sigma = \{39, 3; 46, 3; 59, 4; 38, 2; 31, 1\}) & si \ NFIQ1 = 3 \\ GMM(\mu = \{64, 6; 27, 9; 6, 8; 48, 5; 78, 5\}; \\ \sigma = \{38, 3; 66, 3; 30, 5; 48, 3; 37, 1\}) & si \ NFIQ1 = 4 \\ GMM(\mu = \{8, 0; 57, 7; 75, 5; 1, 0; 33, 7\}; \\ \sigma = \{24, 5; 70, 0; 66, 5; 1, 1; 111, 1\}) & si \ NFIQ1 = 5 \end{cases}$$

⁴Un *Gaussian Mixture Model* es un modelo probabilístico que asume que los datos son generados a partir de una combinación de un número finito de distribuciones Gaussianas con parámetros desconocidos.

4.2.4. Relación entre el Algoritmo del Software Propietario Utilizado por la DNIC y NFIQ1

Para obtener la relación entre el algoritmo del software propietario utilizado por la DNIC y NFIQ1 se siguieron los mismos pasos que en la sección 4.2.2, obteniendo así la gráfica de la figura 4.8, en la cual se pueden apreciar las cinco probabilidades condicionales $P(NFIQ1 = c|Vendor)$ para cada valor de *Vendor*, donde *Vendor* hace referencia al software propietario utilizado por la DNIC.

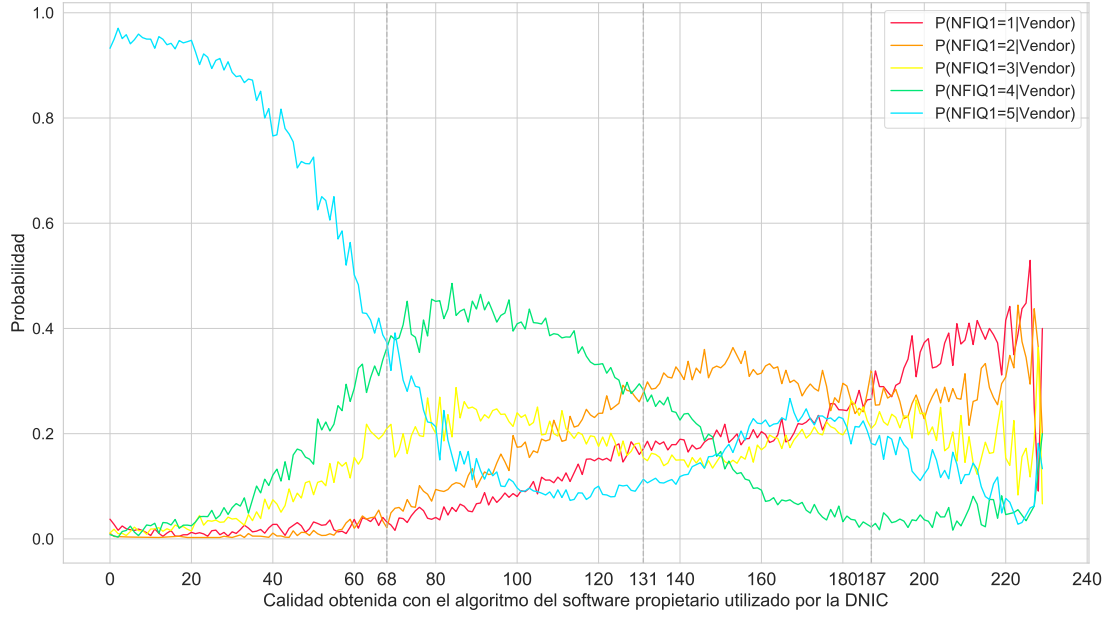


Figura 4.8: Probabilidades condicionales para $P(NFIQ1 = c|Vendor)$ para cada valor de *Vendor*.

De eso se puede obtener la siguiente función $f(Vendor) = NFIQ1$:

$$f(Vendor) = NFIQ1 = \begin{cases} 5 & \text{si } 0 \leq Vendor < 68 \\ 4 & \text{si } 69 < Vendor < 131 \\ 2 & \text{si } 132 < Vendor < 167 \\ 1 & \text{si } 168 < Vendor \leq 256 \end{cases}$$

De esta ecuación se obtiene como dato interesante que no hay imágenes que, dado su valor de *Vendor* se pueda decir que tienen clase 3 de NFIQ1. Esto quiere decir que dada la calidad de una imagen reportada por el software propietario utilizado por la DNIC, solo podemos decir si esta es de buena o mala calidad. Al menos en la base de datos utilizada, no se encontró un rango para aquellas imágenes con una calidad media.

4.2.5. Relación entre NFIQ1 y el Algoritmo del Software Propietario Utilizado por la DNIC

Para obtener la relación entre NFIQ1 y el algoritmo del software propietario utilizado por la DNIC se siguieron los mismos pasos que en la sección 4.2.3, obteniendo así la gráfica de la figura 4.9, en la cual, al igual que en la sección antes mencionada, se pueden apreciar las gráficas para cada clase c mostrando con una línea punteada el máximo de cada una de ellas.

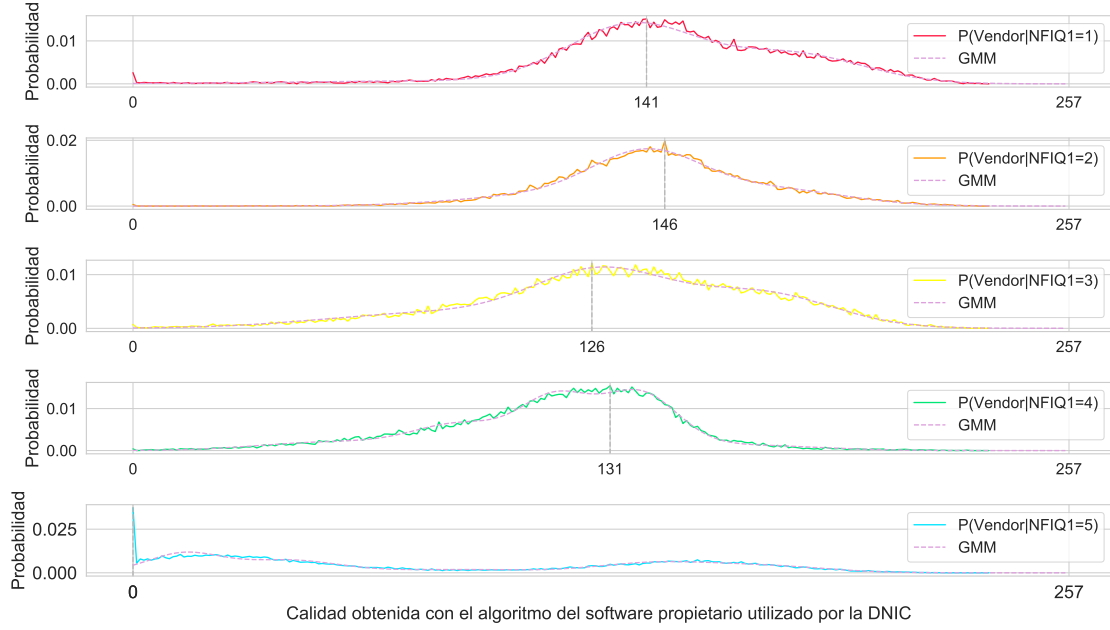


Figura 4.9: Probabilidades condicionales para $P(Vendor = x|NFIQ1)$ para cada clase de NFIQ1.

En este caso, también se aproximó las mismas utilizando un GMM, las cuales quedan definidas como:

$$P(NFIQ2) = \begin{cases} GMM(\mu = \{137, 1; 86, 9; 181, 2\}; \\ \sigma = \{314, 7; 1909, 2; 367, 0\}) & si \ NFIQ1 = 1 \\ GMM(\mu = \{175, 9; 141, 3; 116, 4\}; \\ \sigma = \{392, 9; 225, 4; 618, 2\}) & si \ NFIQ1 = 2 \\ GMM(\mu = \{170, 8; 126, 8; 78, 3\}; \\ \sigma = \{415, 9; 359, 6; 831, 0\}) & si \ NFIQ1 = 3 \\ GMM(\mu = \{112, 2; 136, 8; 84, 3; 160, 8; 50, 3\}; \\ \sigma = \{109, 2; 117, 9; 164, 3; 484, 0; 346, 6\}) & si \ NFIQ1 = 4 \\ GMM(\mu = \{13, 3; 140, 8; 82, 2; 171, 8; 40, 7\}; \\ \sigma = \{94, 7; 277, 3; 392, 7; 299, 0; 167, 1\}) & si \ NFIQ1 = 5 \end{cases}$$

4.3. Análisis de Resultados

En cuanto a la relación entre NFIQ2 y NFIQ1, se logró obtener una función confiable para aquellas imágenes cuya calidad está comprendida en los extremos, teniéndose menos seguridad en valores intermedios.

Esto quiere decir que, para valores bajos y altos de NFIQ2 se puede decir, con un muy alto grado de confiabilidad, que estos serán de clase 5 y 1 de NFIQ1 respectivamente.

No así para las imágenes con valores intermedios donde, si bien lo más probable es que estas sean de clase 3, esto no se puede asegurar con un grado de confiabilidad tan alto. Esto puede ser causa de la distribución de la base de datos utilizada, puesto que al observar la figura 4.3 se puede ver que la cantidad de muestras con clase 3 es menor a las demás.

Luego, se logró aproximar las gráficas obtenidas para la relación entre NFIQ1 y NFIQ2 (ver figura 4.7), cada una por una GMM. Esto provoca que la decisión de a qué valor de NFIQ2 corresponde una imagen dada su clase NFIQ1 sea simple y rápida.

Por otro lado, en cuanto a la relación entre el algoritmo de calidad propietario usado en la DNIC y NFIQ1, se obtuvo una función con alto grado de confiabilidad para todos los valores. Cabe destacar que no se obtuvo ningún rango que se corresponda con la clase 3 de NFIQ1, lo cual también puede ser causa de la distribución de la base de datos utilizada.

Finalmente, para la relación entre NFIQ1 y el algoritmo de calidad propietario también se logró aproximar las gráficas obtenidas (ver figura 4.9) utilizando GMM, obteniendo las ventajas antes mencionadas al tener las mismas.

4.4. Propuesta de Implementación

Se recomienda comenzar a utilizar NFIQ2 como algoritmo de calidad, dado que es la tendencia actual y es la versión mejorada del utilizado hoy en día en la DNIC.

Para ello, se implementó una DLL que provee acceso a dicho algoritmo y se puso disposición de la DNIC el estudio realizado para ayudar en la transición de un algoritmo a otro (de NFIQ1 a NFIQ2 en la caso de DNIC). En la sección 8.3 se encuentra una descripción más detallada de la misma.

De igual forma, también se recomienda realizar el protocolo experimental nuevamente, con una base de datos más variada y con mayor cantidad de imágenes (idealmente, entre medio millón y un millón en total, para que los problemas de clasificación queden bien definidos).

5. Escalado

A continuación se trata el tema de escalado. Se presenta el marco teórico sobre el que se basaron los experimentos realizados para esta área y se explican los mismos. Uno de los experimentos llevados a cabo se basa en reproducir los resultados del método de escalado lineal para llevar la huella a tamaño de adulto presentado en el marco teórico. En el segundo experimento se construyen y comparan varios modelos de crecimiento de una huella dactilar, basados en el estado del arte y con el agregado de que no necesitan de datos previamente marcados por expertos. Por último, se presenta el análisis de los resultados obtenidos en ambos experimentos, se comparan los distintos métodos y se propone la implementación de software que permita integrar al sistema actual el método de escalado lineal, ya que fue el que mejor desempeño obtuvo.

5.1. Marco Teórico

La gran mayoría de los algoritmos utilizados en biometría en el área de huellas dactilares son pensados y desarrollados para el procesamiento de huellas de adultos; es por esto que generalmente se presentan problemas de desempeño cuando se los usa para huellas dactilares de niños, debido al menor tamaño de la huella y la estructura de las crestas y valles en ella. Además, características particulares de la piel de los dedos de niños (grasitud y humedad) no permiten una correcta toma de la huella. Otro factor es el crecimiento más rápido (y por ende el desplazamiento de las minucias) del dedo en niños en comparación con adultos. A este efecto se lo conoce como *biometric ageing* y provoca una disminución en el desempeño de los sistemas de reconocimiento de huellas, especialmente cuando se trata de huellas de niños.

Una forma de mitigar este problema es escalar las imágenes de huellas de niños a tamaño de adulto utilizando métodos de interpolación de imágenes. La interpolación es una “operación que transforma una imagen discreta $u(i, j)$ (definida sobre una grilla) en una imagen $u(x, y)$ definida sobre un continuo” [12] y se puede hacer mediante el uso de distintos algoritmos. A modo de ejemplo, algunos algoritmos conocidos son: método del vecino más cercano, interpolación lineal, bilineal, bicúbica, Lanczos, entre otros.

En “*Fingerprint Growth Model for Mitigating the Ageing Effect on Children’s Fingerprints Matching*” [19] y “*Automatic fingerprint recognition: from children to elderly Ageing and age effects*” [5], se busca encontrar una mejora al problema de disminución de desempeño debido a *biometric ageing*, y se plantea además verificar que el mismo existe. Para esto, se utilizó una base de datos con huellas de personas en un rango de edad amplio (de 0 a 90 años aproximadamente), y se subdividió en

grupos de acuerdo a la edad en la que fue tomada la primera huella de la identidad (hay dos tomas de la misma huella por identidad). Luego, se calcularon los *matching scores* correspondientes utilizando un software de *matching* de huellas realizado por el NIST [28].

Los resultados obtenidos demostraron que las huellas de niños de entre 0 y 4 años de edad presentan un *matching score* promedio más bajo que las de niños de entre 5 y 12 años; así como también se observó que los resultados obtenidos para huellas tomadas entre los 13 y 17 años de edad y los obtenidos para huellas de adultos son muy similares. Esto confirma la presencia del llamado *age effect*. *Age effect* se le llama a la variación en el desempeño de un sistema biométrico observada entre diferentes grupos de usuarios según su edad.

Se comprobó también, que para todos los grupos de edad, cuanto mayor es el tiempo entre que se obtuvo cada toma, mayor es la pérdida de *matching score*.

En “*Fingerprint Growth Model for Mitigating the Ageing Effect on Children’s Fingerprints Matching*” [19] se propone abordar el problema descrito anteriormente mediante el desarrollo de un modelo de crecimiento de las huellas basado específicamente en las minucias y la distancia de ellas al centro de la huella. La hipótesis en la que se basa y que impulsa dicho artículo es la posibilidad de que la aplicación de un modelo de crecimiento a huellas de niños antes del proceso de *matching* permita mitigar la disminución de desempeño mencionada anteriormente en los sistemas de reconocimiento de huellas cuando la muestra de referencia y de ejemplo fueron tomadas en tiempos distantes.

Efectivamente se comprobó esa hipótesis luego de la realización de pruebas y verificación en una base de datos con más de 60.000 huellas de personas de entre 5 y 16 años de edad.

En dicho artículo, se llegó a la conclusión de que el crecimiento del dedo con el correr de los años (el cual se comprobó que es isotrópico en “*Modeling the Growth of Fingerprints Improves Matching for Adolescents*” [18]), genera un desplazamiento isotrópico de las minucias, lo que quiere decir que el desplazamiento de los puntos de minucias es independiente de la posición en la que se encuentre en la huella.

El modelo de crecimiento propuesto plantea, entonces, la existencia de constantes de crecimiento que dependen únicamente de la edad en la que la huella de referencia fue obtenida y de la diferencia en años entre dicha toma y la de la huella de prueba.

Durante el estudio realizado por los autores del artículo en cuestión, se llegó a ciertas funciones de crecimiento que se utilizaron para calcular los coeficientes de crecimiento para rangos de edad que van desde la niñez hasta la adolescencia.

Es importante destacar que ya existen tablas de crecimiento generales establecidas por la Organización Mundial de la Salud que se utilizan a nivel internacional desde los años setenta (aunque fueron actualizadas para que se

adecúen específicamente al crecimiento en la primera infancia entre los años 1997 y 2003) [38]. Pero dicho artículo se centra específicamente en el crecimiento de los dedos en niños.

Si bien los resultados reportados en el artículo son buenos, el método presenta la limitación de tener que marcar manualmente el centro de cada huella, lo que representa un problema práctico para su uso en producción.

Otro artículo que aborda el efecto de *biometric ageing* es “*Fingerprint Biometrics From Newborn to Adult: A Study From a National Identity Database System*” [33]. En este caso, el interés principal es poder usar huellas de niños para identificación automática, utilizando el hardware y software ya disponible. Las dos preguntas que guían el estudio realizado en dicho artículo son: “¿desde qué edad se pueden obtener huellas que sean lo suficientemente buenas como para comparación automática” y ¿cómo es el desempeño cuando se comparan huellas de niños con huellas de adultos?

Para responder estas preguntas, se hicieron estudios con una base de datos de más de 200.000 huellas, con 134.000 identidades aproximadamente.

Para el análisis, se decidió realizar un pre-procesamiento a las huellas de forma que se pudieran usar los sistemas comerciales existentes con ellas. Este pre-procesamiento consistió en dos etapas: segmentación e interpolación. Un detalle importante a destacar es que algunas de las huellas presentes en la base de datos utilizada fueron extraídas con tinta y papel y luego digitalizadas con un escáner; es por esto, principalmente, que se realizó la fase de segmentación, eliminando así posibles marcas presentes en las imágenes.

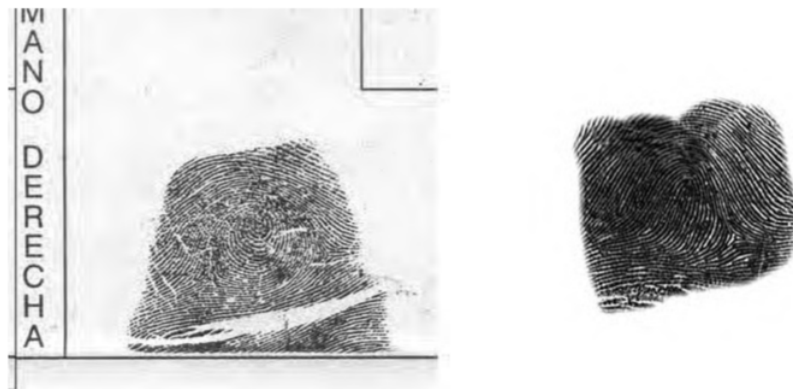


Figura 5.1: A la izquierda: huella obtenida en tinta y papel. A la derecha: huella obtenida mediante un escáner. Imagen tomada de “*Fingerprint Biometrics From Newborn to Adult: A Study From a National Identity Database System*” [33].

En este mismo artículo, se mencionan también otros dos estudios “*Biometrics for Child Vaccination and Welfare: Persistence of Fingerprint Recognition for Infants and Toddlers*” [21] y “*Fingerprint Recognition of Young Children*” [22], en los que se

utiliza interpolación lineal con un factor de interpolación fijo de 1,8 obtenido como resultado de comparar el tamaño y la distancia entre crestas en píxeles en imágenes de huellas de adultos y de niños obtenidas con un escáner de 500 dpi.

Sin embargo, en “*Fingerprint Biometrics From Newborn to Adult: A Study From a National Identity Database System*” [33], se decide obtener un factor de escala que dependa de la edad en la que se obtuvo la huella. Esto se hace mediante el análisis de huellas dactilares obtenidas a diferentes edades. En la tabla 5.2 se muestran los resultados obtenidos.

Edad	Factor de escalado
0 meses	1,52
1 mes	1,63
2 meses	1,65
3 meses	1,58
4 meses	-
5 meses	1,60
6-11 meses	1,54
1 año	1,49
2 años	1,47
3-4 años	-
5 años	1,32
6 años	1,29
7 años	1,26
8 años	1,24
9 años	1,22

Figura 5.2: Factores de escalado obtenidos de “*Fingerprint Biometrics From Newborn to Adult: A Study From a National Identity Database System*” [33].

Se decidió utilizar interpolación bicúbica debido a que a pesar de ser un método sencillo, se obtuvo buenos resultados en pruebas hechas con un conjunto reducido de imágenes.

Las comparaciones hechas con huellas sin pre-procesar muestran que los resultados obtenidos realmente mejoran de forma considerable la calidad de las huellas de niños de más de 1 año de edad. La calidad fue medida con NFIQ2 y particularmente para el rango de edad de 2 a 6 años, se obtuvo resultados comparables con los de un adulto, dando así una gran mejora en el desempeño del *matching* de huellas de niños con adultos.

5.2. Protocolo Experimental

Se analizaron dos métodos distintos para transformar las huellas a tamaño de adulto ya que esto es imprescindible para poder utilizar el software de *matching*

actual de forma confiable: escalado lineal y modelo de crecimiento de huellas dactilares.

5.2.1. Escalado Lineal

En “*Fingerprint Biometrics From Newborn to Adult: A Study From a National Identity Database System*” [33] se infirieron ciertos factores de escalado para imágenes de huellas dactilares por edad que, según los autores, mejoran el desempeño de los algoritmos de *matching* en el caso de los niños.

Con el objetivo de reproducir este resultado para poder determinar si realizar este tipo de procesamiento a las huellas podría ser de utilidad en el contexto de la DNIC, se decidió hacer un estudio del desempeño del software propietario de *matching* utilizado por la DNIC al aplicar el escalado.

Para la experimentación en este tema se utilizó la base de datos C (sección 3.1.3). Todas las imágenes en dicha base de datos fueron pre-procesadas de forma de tener las huellas correctamente segmentadas utilizando el segmentador desarrollado en el proyecto “*Identificación de niños utilizando huellas digitales*” [7] y provisto por el Ing. Guillermo Garella.

Luego, se generó una copia de los datos anteriormente mencionados en donde cada imagen fue escalada utilizando interpolación bicúbica, según el factor correspondiente a la edad en que fue extraída la huella. Los factores usados son los obtenidos en dicho artículo y se pueden ver en la tabla 5.2.

Para los casos en los que no se contaba con un factor de escalado, las imágenes se mantuvieron sin transformar.

Tanto para el conjunto de datos sin escalar como para el escalado, se generaron los datos de *matchings* de genuinos e impostores como se explica en la sección 3.2.1 del presente informe.

5.2.2. Crecimiento de Huellas Dactilares

En “*Fingerprint Growth Model for Mitigating the Ageing Effect on Children’s Fingerprints Matching*” [19] se propone modelar cómo es el crecimiento de las huellas dactilares mediante una transformación homográfica¹ M tal que dado un par de huellas $h0_i$ y $h1_i$ de una identidad i , obtenidas en distinto tiempo $T0$ y $T1$ con $T0 < T1$, se cumpla que $M(h0_i) \approx h1_i$.

Para lograr esto, los autores trabajaron con un conjunto de huellas para el que se tiene tanto el centro de las mismas así como 12 minucias marcadas para cada

¹Transformación perspectiva que determina una correspondencia entre dos figuras geométricas planas, de forma que a cada uno de los puntos y las rectas de una de ellas le corresponden, respectivamente, un punto y una recta de la otra.

huella. Estas marcas fueron realizadas por expertos.

En este proyecto se trabajó en replicar la construcción de esta transformación para poder comparar su desempeño con el método de escalado lineal para el caso de las huellas de niños y, además, se diseñó una solución en la cual no fuera necesario contar con huellas marcadas previamente.

Tomando como punto de partida la hipótesis de que el crecimiento de las huellas dactilares es isotrópico, se esperaría que una transformación perspectiva, como la presentada en el artículo antes mencionado, que incluya la información de rotación y traslación no obtenga mejor desempeño que un simple escalado lineal. Esto se debería a que la rotación y la traslación dependen de la captura de la huella y no tendrían que incidir en el crecimiento de la misma, generando sobreajuste a los datos incluir toda esta información. El sobreajuste probablemente lleve a un peor desempeño que en un modelo en donde se generaliza, como es el caso del escalado lineal.

El estudio se hizo primeramente sobre un conjunto de imágenes de prueba con el objetivo de lograr validar que fuera posible hallar la transformación. Este conjunto consistió de las tomas de treinta identidades con mejor *matching score* de la base de datos C (sección 3.1.3), con la única condición de que la primera toma para cada identidad fuese para una edad de 5 años o menor (y tuviese al menos dos tomas, para así poder hallar la transformación de una toma a la otra).

Posteriormente, se reprodujo el procedimiento sobre todas las identidades de la base de datos C (sección 3.1.3), para así tener un conjunto de imágenes mucho mayor sobre el cual obtener conclusiones.

Para ambos experimentos, se realizó un pre-procesamiento a las imágenes, el cual consistió en:

- Segmentar todas las tomas para dichas imágenes, utilizando el segmentador antes mencionado.
- En caso de ser necesario, aplicar un escalado lineal a las tomas segmentadas utilizando los factores de la tabla 5.2.

Una vez realizado el pre-procesamiento de las imágenes, es necesario obtener la correspondencia de minucias entre tomas. Para ello se decidió utilizar el software “Minutia Cylinder-Code SDK”(MCC SDK de aquí en más) [36][35][37][34], provisto por el *Biometric System Laboratory* de la Universidad de Bologna [43], el cual cuenta con funciones para obtener dicha correspondencia a partir de archivos de *template* en el formato estándar ISO 19794-2:2011 [20]. Para generar dicho *template* se utilizó el software propietario utilizado por la DNIC.

Se utilizaron las funcionalidades provistas por el MCC SDK para obtener las correspondencias de minucias entre imágenes de una misma identidad, mismo dedo, pero con distinta edad. En la figura 5.3 se puede observar un ejemplo de las

correspondencias retornadas por el MCC SDK.



Figura 5.3: Ejemplo de correspondencias entre minucias retornadas por MCC SDK.

Obtenidas las correspondencias entre minucias, se procedió a calcular la transformación. Para ello, primero se obtuvieron los puntos de las minucias en las imágenes sin escalar (es decir, se revirtió el escalado para dichos puntos) y se utilizaron estos puntos luego para hallar la transformación.

Para hallar la transformación se optó por utilizar la función “*findHomography*” de la librería OpenCV la cual recibe como parámetros los puntos origen, los puntos destino y el método a utilizar para computar la transformación, retornando una transformación perspectiva. Se eligió RANSAC² como el método a utilizar con un valor de 10 como el umbral para el error de re-proyección³. Dicha decisión fue tomada dada la habilidad de RANSAC de hacer estimaciones robustas de los parámetros del modelo, es decir, poder estimar dichos parámetros con un alto grado de precisión incluso cuando el conjunto de datos tiene un alto número de *outliers*⁴. En la figura 5.4 se muestra el resultado de aplicar la transformación a una tupla de imágenes.

Una vez obtenida la transformación, se calculan los factores de escalado en ambos ejes (s_x y s_y respectivamente), para así poder validar cómo es el crecimiento de las huellas. Para hallar estos factores, es necesario descomponer una transformación M de la forma $M = R \cdot S \cdot T$, donde R es la matriz de rotación, S la de escalado y T de la traslación. Dado que las transformaciones obtenidas anteriormente (utilizando la función *findHomography*) no son afines, sino que son perspectivas, no es posible realizar dicha descomposición.

Por esta razón, se buscó una forma de estimar transformaciones afines para

²RANSAC: técnica de remuestreo que genera soluciones candidatas utilizando el número mínimo de observaciones (puntos de datos) requerido para estimar los parámetros del modelo.

³El error de re-proyección es la distancia euclídea (en píxeles) entre el punto deseado y el obtenido.

⁴*Outliers*: dato cuyo error cae fuera del máximo provisto al algoritmo.

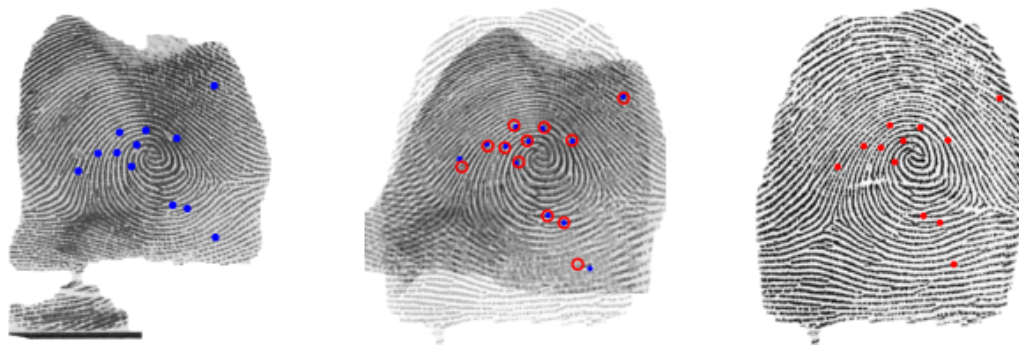


Figura 5.4: Resultado de aplicar la transformación perspectiva obtenida. Izquierda: minucias halladas en la imagen de la toma 1. Derecha: minucias halladas en la imagen de la toma 2. Medio: imagen de la toma 1 transformada para que sus minucias correspondan con la de la imagen de la toma 2.

así poder descomponerlas. Para esto, también se utilizó una función de la librería OpenCV, “*estimateAffinePartial2D*”, la cual tiene la misma interfaz que la función “*findHomography*” (es decir, recibe como parámetros los puntos origen, destino, método a utilizar y un valor para el umbral del error de re-proyección). El hecho de que ambas funciones utilicen el mismo método (RANSAC), con el mismo umbral, permite realizar una comparación cualitativa de los errores de re-proyección para evaluar la calidad de las transformaciones obtenidas.

En la figura 5.5 se puede observar le resultado de aplicar la transformación afín obtenida para una tupla de imágenes.

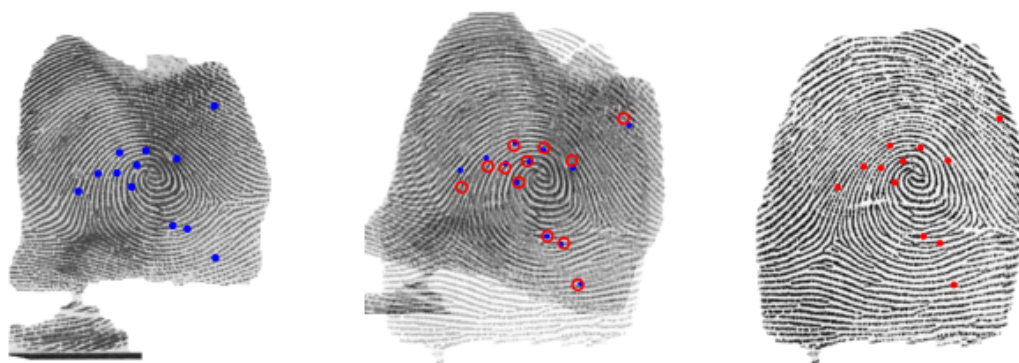


Figura 5.5: Resultado de aplicar la transformación afín obtenida. Izquierda: minucias halladas en la imagen de la toma 1. Derecha: minucias halladas en la imagen de la toma 2. Medio: imagen de la toma 1 transformada para que sus minucias correspondan con la de la imagen de la toma 2.

Una vez obtenidas las transformaciones afines, se procedió a descomponer las mismas en las tres matrices antes mencionadas. Para ello, se utilizó el sistema de

ecuaciones que surge de la siguiente igualdad (cálculos en el anexo F):

$$\begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s_x \cdot \cos(\theta) & -s_y \cdot \sin(\theta) & t_x \cdot s_x \cdot \cos(\theta) - t_y \cdot s_y \cdot \sin(\theta) \\ s_x \cdot \sin(\theta) & s_y \cdot \cos(\theta) & t_x \cdot s_x \cdot \sin(\theta) + t_y \cdot s_y \cdot \cos(\theta) \\ 0 & 0 & 1 \end{pmatrix}$$

donde: $\begin{cases} s_x : & \text{Factor de escalado en el eje } x \\ s_y : & \text{Factor de escalado en el eje } y \\ \theta : & \text{Ángulo de rotación} \\ t_x : & \text{Factor de traslación en el eje } x \\ t_y : & \text{Factor de traslación en el eje } y \end{cases}$

En resumen, el protocolo experimental contó con los siguientes pasos:

1. Pre-procesamiento de las imágenes.
2. Cálculo de la correspondencia entre minucias utilizando el software MCC SDK.
3. Hallar transformaciones perspectivas con la función *findHomography*.
4. Hallar transformaciones afines con la función *estimateAffinePartial2D*.
5. Hallar las componentes s_x y s_y de las transformaciones afines.

5.3. Análisis de Resultados

En esta sección se presenta el análisis de los resultados obtenidos para los dos experimentos realizados en el área de fusión.

5.3.1. Escalado Lineal

Para poder constatar si realizar un escalado lineal a las imágenes de huellas dactilares de niños mejora el desempeño del *matching*, se generaron las curvas ROC tanto sin aplicar el escalado como aplicándolo.

En la figura 5.6 se puede observar los resultados con las imágenes sin escalar y en la figura 5.7 al escalarlas.

En general, para todas las edades se logró mejorar el desempeño, lo cual coincide con lo reportado en “*Fingerprint Biometrics From Newborn to Adult: A Study From a National Identity Database System*” [33]. Además, al haberse realizado el estudio con el mismo software que utiliza la DNIC en su sistema, este resultado es relevante a la hora de contemplar cambios al sistema actual.

Es importante notar que, si bien para menores de 5 años el desempeño sigue teniendo mucho margen de mejora, se logró estar por arriba de 0.9 en TAR para huellas de 5 años. Las huellas de 5 años tienen gran valor para la DNIC hoy en día ya que, comúnmente, son las que quedan almacenadas en el sistema.

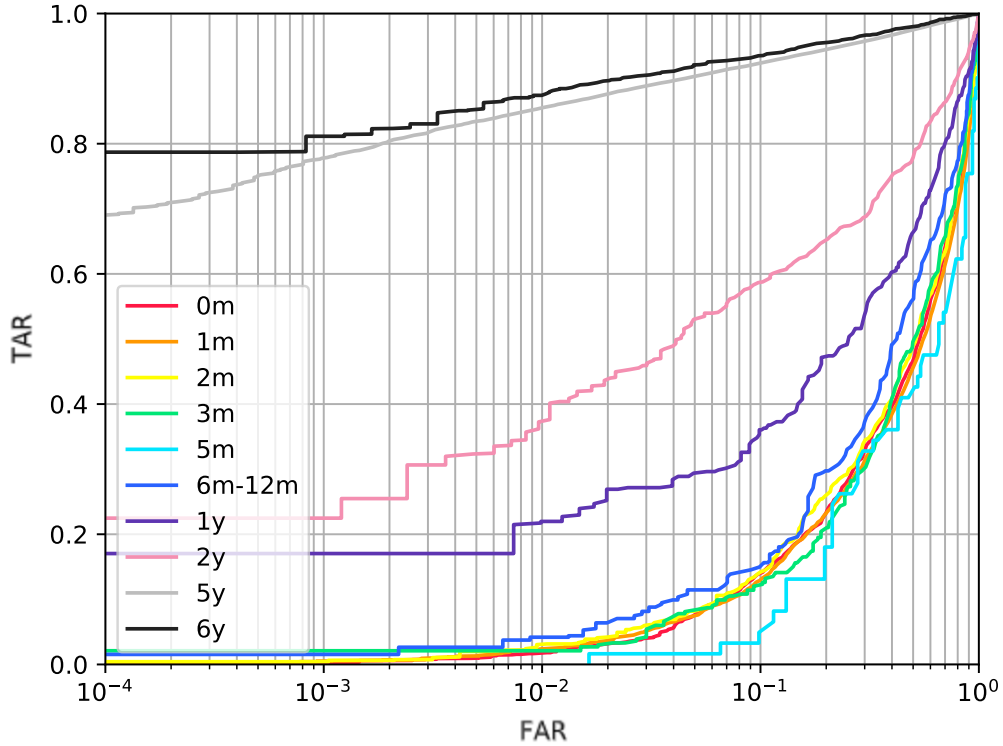


Figura 5.6: Curvas ROC para imágenes sin escalar por edad.

5.3.2. Crecimiento de Huellas Dactilares

Para el primer análisis se trabajó sobre el conjunto de imágenes de prueba y se tomaron como referencia los errores de re-proyección de cada uno de los puntos de minucia transformados (tanto utilizando las transformaciones perspectivas como las afines) y se tomó la media⁵ y mediana⁶ de los mismos para cada tupla de imágenes como punto de análisis.

Se pueden observar las gráficas obtenidas de dicho análisis sobre las transformaciones perspectivas en las figuras 5.8 y 5.9 y, de igual forma, se pueden observar las gráficas para las transformaciones afines en las figuras 5.10 y 5.11. En el eje de las abscisas se encuentra el *score* reportado por el MCC SDK, el cual toma valores entre 0 y 1, donde 0 corresponde a *templates* con similaridad muy baja y 1 a *templates* con similaridad muy alta; y en el de las ordenadas el error de re-proyección correspondiente a dos *templates*.

En las cuatro gráficas se puede ver que un alto porcentaje de las muestras, tanto en la media como en la mediana tienen un valor cercano o menor a 10 (el

⁵La media (promedio) de un conjunto de datos se encuentra al sumar todos los números en el conjunto de datos y luego dividir entre la cantidad de valores en el conjunto.

⁶La mediana es el valor medio cuando un conjunto de datos se ordena de menor a mayor.

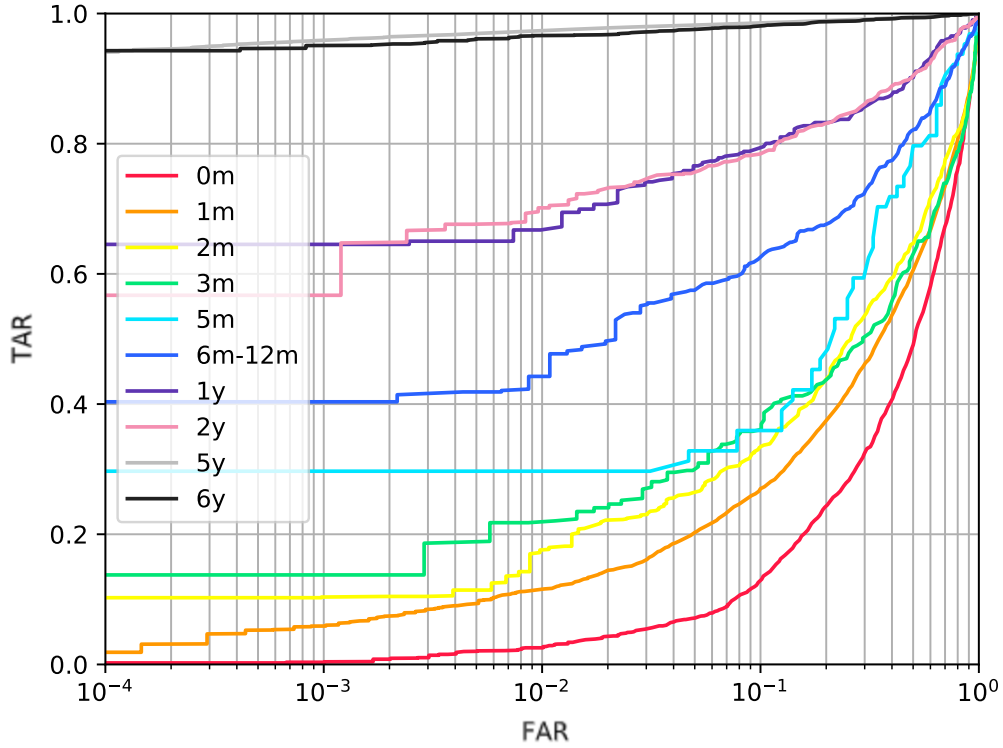


Figura 5.7: Curvas ROC para imágenes escaladas por edad.

umbral utilizado para el error de re-proyección). Esto indica (especialmente al ver la mediana) que en general, los puntos de minucia transformados están en un entorno muy cercano al deseado, lo que implica que, como se puede observar para el ejemplo de la figura 5.4, en general no son *outliers*. Es decir, que en general los puntos transformados están cercanos al punto destino objetivo, lo cual es un indicio de que la transformación hallada es de buena calidad. Cabe mencionar que, al comparar los errores de re-proyección obtenidos con los diferentes tipos de transformaciones, los errores de las afines son ligeramente mayores que los de las perspectivas, pero, de todas formas, estos son cercanos o menores al umbral antes mencionado.

Luego, en la figura 5.12 se puede observar los valores de s_x y s_y obtenidos a partir de las transformaciones afines y una línea punteada indicando el factor de escalado lineal para huellas de 5 años (1,32) según “*Fingerprint Biometrics From Newborn to Adult: A Study From a National Identity Database System*” [33]. En esta se pueden apreciar dos particularidades. Primero que, si bien están graficados ambos valores, solo se ven los puntos correspondientes a s_y , porque para todas las matrices de escalado se cumple que $s_x = s_y$. Esto quiere decir que el crecimiento de las huellas (al menos en el rango de 5 años a adulto) es isotrópico. Y segundo, se puede ver que los valores de s_x y s_y están un entorno cercano al factor de escalado lineal para huellas de cinco años.

Finalmente, para analizar la calidad de las transformaciones se calcularon las

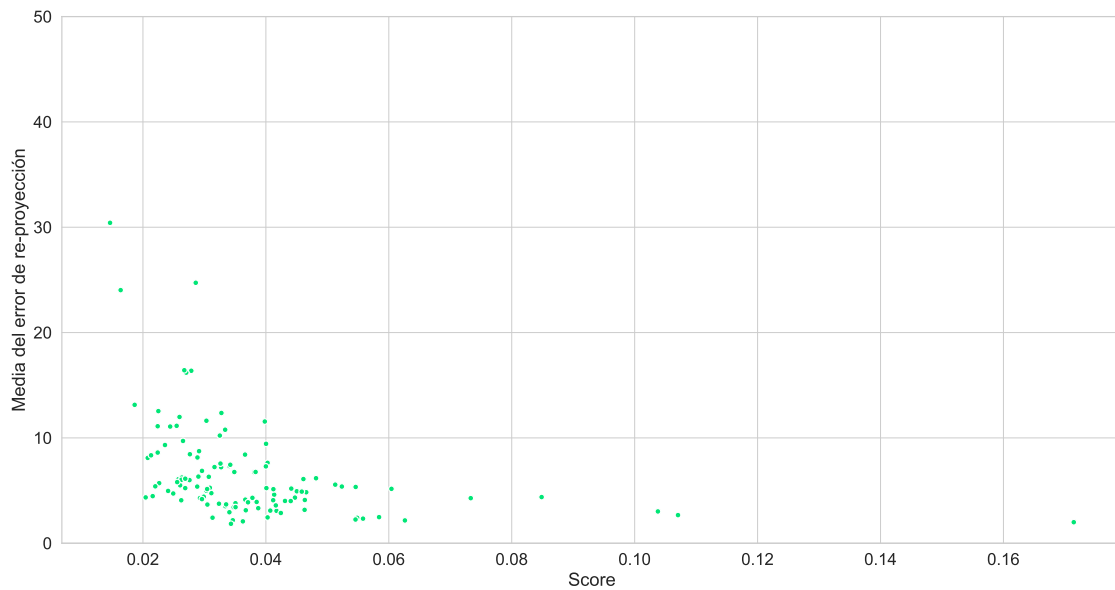


Figura 5.8: Media de los errores de re-proyección contra los *scores* (transformaciones perspectivas).

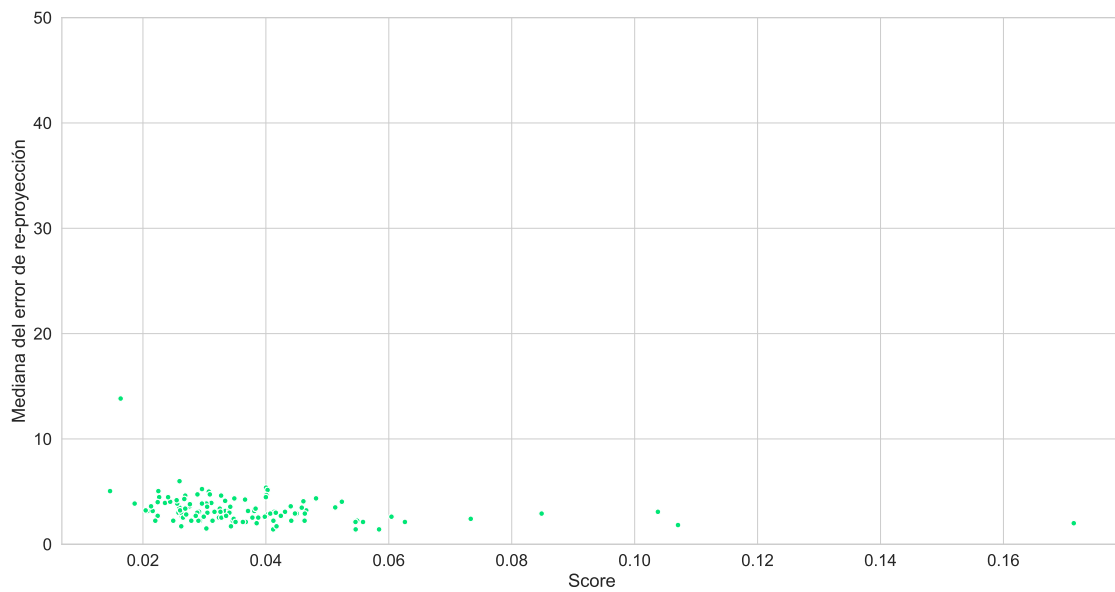


Figura 5.9: Mediana de los errores de re-proyección contra los *scores* (transformaciones perspectivas).

curvas ROC para cinco conjuntos de imágenes (con base el conjunto de imágenes de prueba):

- Imágenes originales, segmentadas sin escalar.
- Imágenes originales, segmentadas y escaladas con el escalado lineal.
- Imágenes originales, segmentadas y escaladas con la transformación perspectiva.

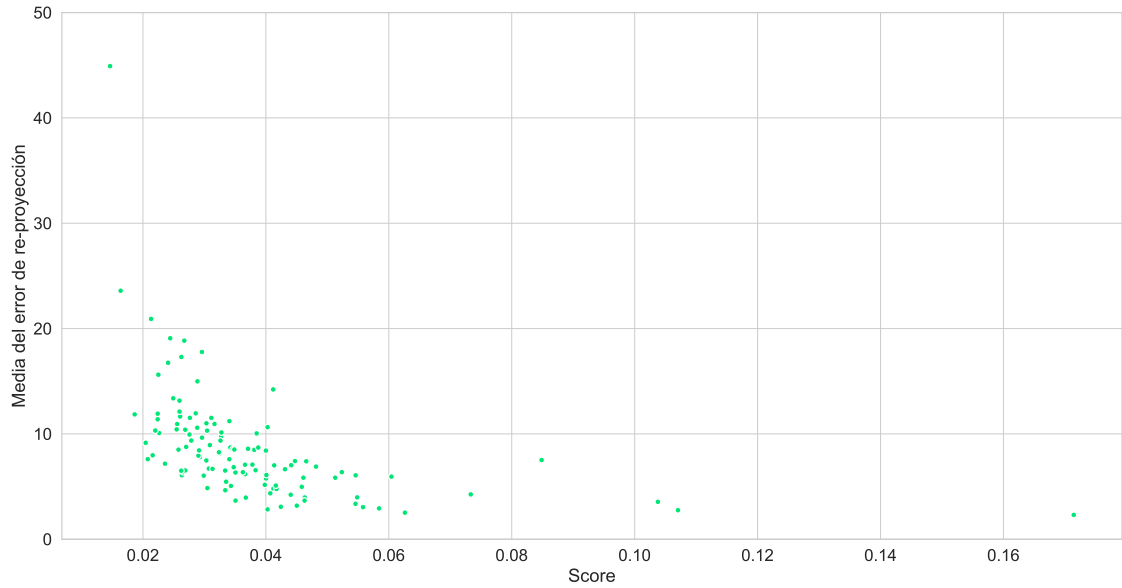


Figura 5.10: Media de los errores de re-proyección contra los *scores* (transformaciones afines).

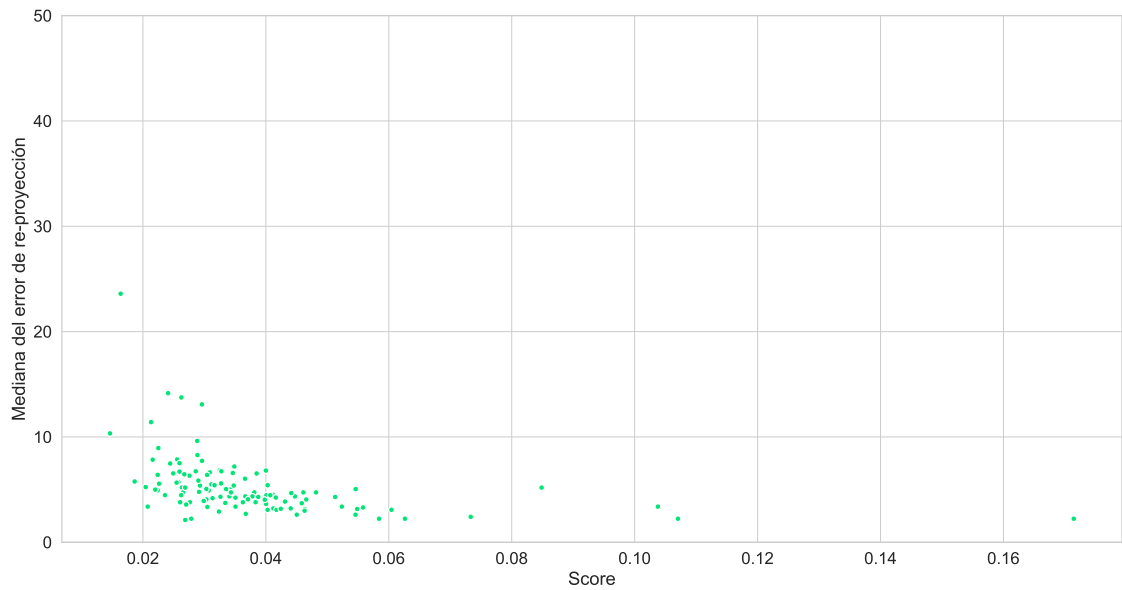


Figura 5.11: Mediana de los errores de re-proyección contra los *scores* (transformaciones afines).

- Imágenes originales, segmentadas y escaladas con la transformación afín.
- Imágenes originales, segmentadas y escaladas con la matriz de escalado obtenida a partir de la transformación afín.

En la figura 5.13 se pueden observar dichas curvas. Al analizar las mismas, se puede ver que todos los tipos de escalado mejoran el rendimiento, ya que el área debajo de la gráfica es claramente mayor para estos. Pero, entre estos el escalado lineal es el que parecería proveer mejor rendimiento, pero no es una diferencia tan

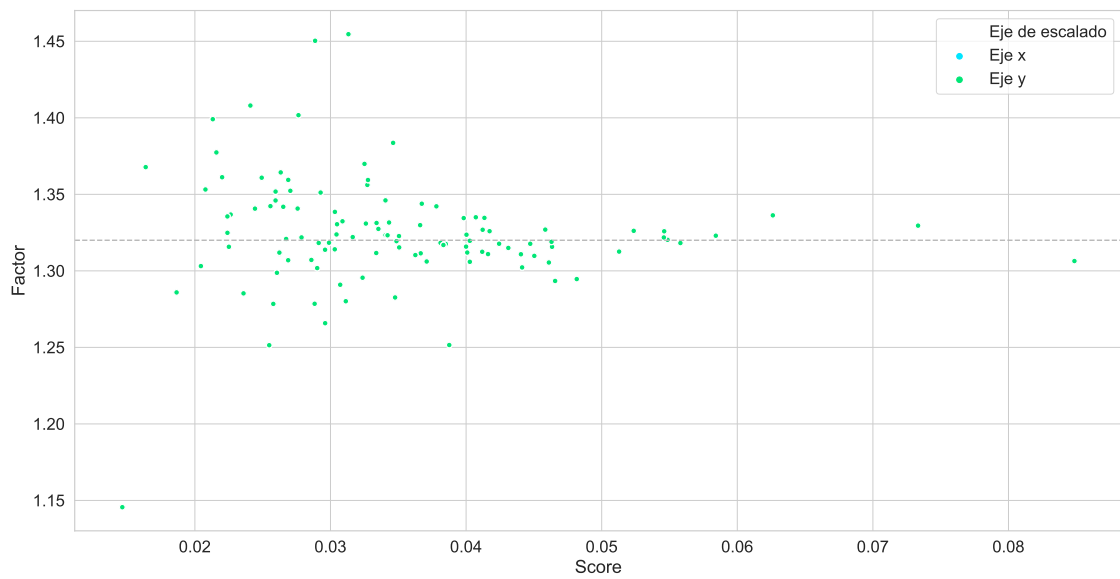


Figura 5.12: Factores de escalado obtenidos a partir de las transformaciones afines contra los *scores*.

clara como la antes mencionada (también, entre los restantes, el mejor parecería ser utilizar la matriz de escalado obtenida).

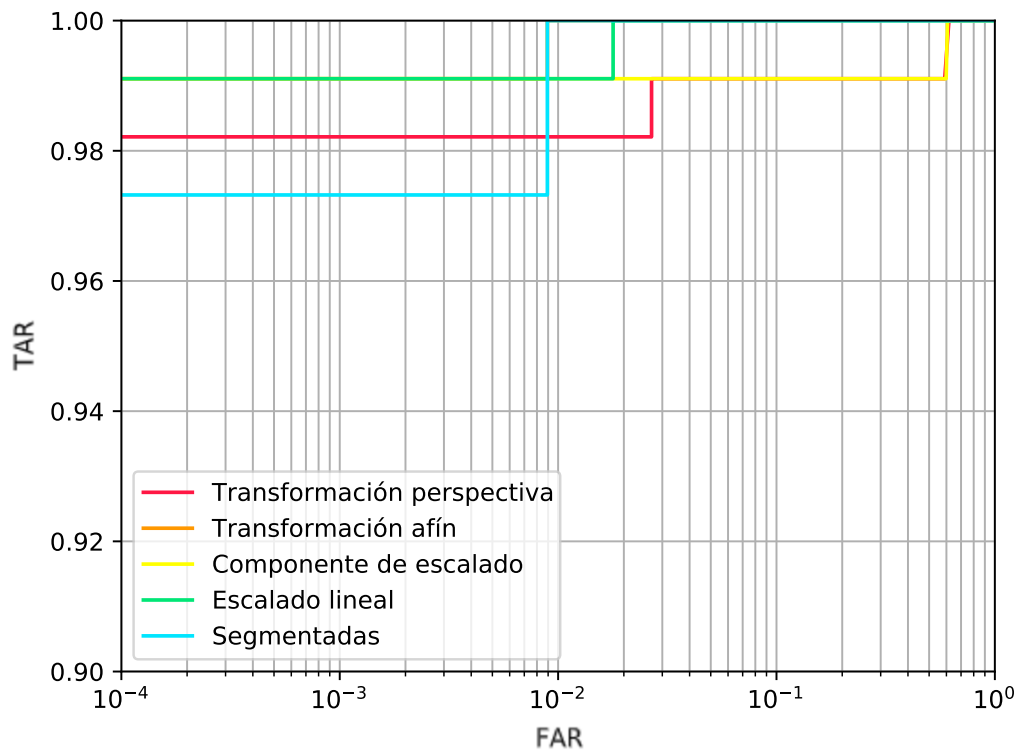


Figura 5.13: Curvas ROC para un mismo conjunto de imágenes escaladas con los diferentes métodos evaluados.

Dado que los resultados obtenidos con el conjunto de imágenes de prueba no son lo suficientemente convincentes en cuanto a qué escalado logra una mejora superior en rendimiento, se decidió a implementar el protocolo experimental sobre todas las identidades de la base de datos C (sección 3.1.3), para así tener un conjunto de imágenes mucho mayor sobre el cual obtener conclusiones.

Luego de realizar el pre-procesamiento a todas las identidades, y hallar para cada par de imágenes las diferentes transformaciones y factores de escalado, se calcularon las curvas ROC (en la figura 5.14 se pueden observar para aquellas tuplas cuya primer toma corresponde a una huella de 5 años) de la misma forma que para el conjunto de prueba.

De estas nuevas curvas se verifica que el escalado lineal provee un mejor rendimiento que los demás, y que, entre los restantes, el que provee mejor rendimiento es utilizar solo las componentes de escalado obtenidas. Esto es consistente con la hipótesis planteada anteriormente de que la transformación perspectiva sufre de sobreajuste, por lo que modelos más generales (como lo son la transformación afín, componente de escalado y escalado lineal) reportan mejor desempeño.

Concluyendo, en este análisis se logró verificar que las huellas tienen un crecimiento isotrópico, es decir, que crecen uniformemente en ambos ejes, dado que los valores de s_x y s_y son iguales para todas las descomposiciones halladas. Este resultado es consistente con el estado del arte. Además, se logró construir el modelo presentado en “*Fingerprint Growth Model for Mitigating the Ageing Effect on Children’s Fingerprints Matching*” [19] sin necesidad de contar con huellas marcadas anteriormente, pudiendo así aportar una mejora al proceso mencionado en dicho artículo. Finalmente, se verificó también que, si bien ambas transformaciones y utilizar solamente las componentes de escalado mejoran el rendimiento, el escalado lineal por rango de edad sigue siendo mejor en términos de desempeño, aún siendo el modelo más simple.

5.4. Propuesta de Implementación

Dados los resultados obtenidos en los experimentos realizados para esta área de trabajo, se recomienda realizar un escalado a las huellas dactilares de niños.

Se recomienda que el escalado a utilizar sea lineal y que el factor utilizado dependa de la edad del individuo al momento en el que se tomó la huella. Los factores de escalado recomendados se pueden observar en la tabla 5.2.

Para ello se implementó una Dynamic Link Library (DLL de aquí en más) que facilita la realización de dicho escalado. Esta DLL cuenta con varias operaciones, definidas en acuerdo con el Ing. Pablo Sanguinet (contacto de la DNIC), para así poder ser utilizada de diferentes maneras y en diferentes puntos del sistema actual. En la sección 8.2 se encuentra una descripción más detallada de la misma.

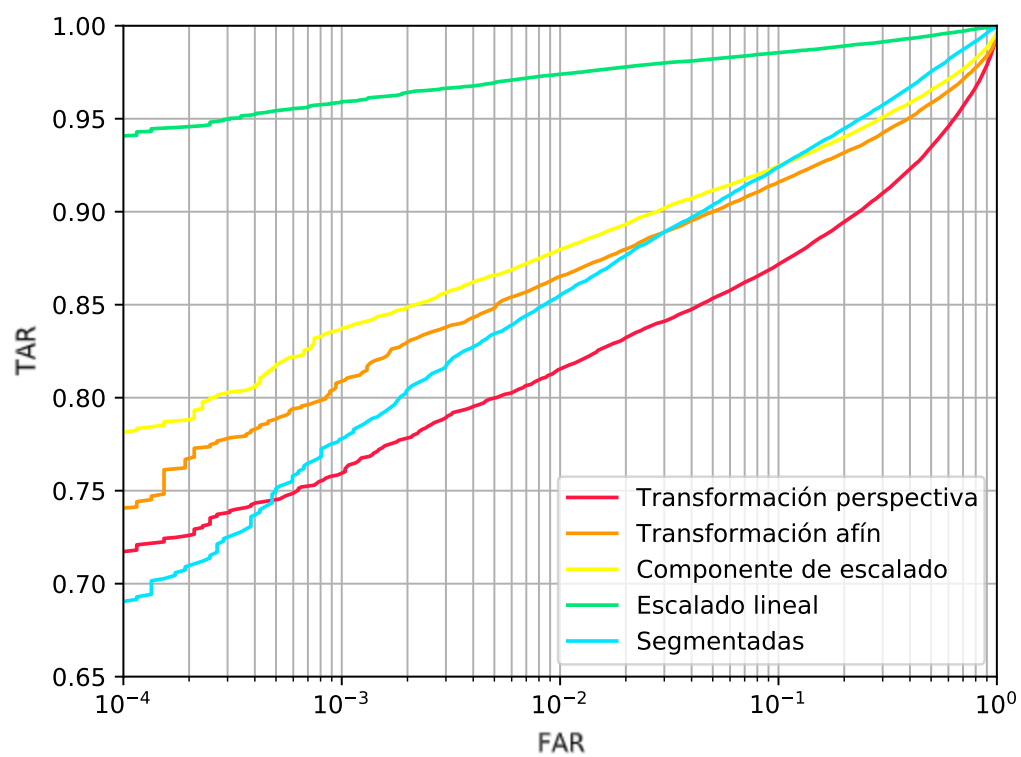


Figura 5.14: Curvas ROC para un mismo conjunto de imágenes cuya primer toma es de 5 años escaladas con los diferentes métodos evaluados.

6. Fusión

En este capítulo, se brinda información específica del estado del arte de fusión, así como también la realidad actual de la DNIC en este aspecto. Se presenta también el protocolo experimental llevado a cabo y los resultados obtenidos, que muestran que, con un leve cambio a lo que la DNIC utiliza actualmente, ya se logra un mejor desempeño. La propuesta de implementación se basa en este resultado.

6.1. Marco Teórico

En un sistema biométrico la fusión puede darse en distintos puntos del proceso, desde el momento en que se adquiere la muestra (muestra de consulta) hasta el momento en que se toma la decisión de si la identidad de la muestra coincide o no con la identidad representada por la información biométrica almacenada, también denominada *referencia*. A estos distintos puntos se los denomina niveles de fusión.

Los niveles de fusión se pueden catalogar en dos grupos: previa al *matching* y luego del *matching*.

- **Fusión previa al *matching*:** puede darse a nivel del sensor, combinando la información cruda obtenida directamente del mismo, o a nivel de característica en cuyo caso se realiza combinando diferentes conjuntos de características extraídos de múltiples fuentes biométricas.
- **Fusión luego del *matching*:** los esquemas para la integración de información después de la clasificación o *matching* pueden ser divididos en cuatro categorías:
 - Selección dinámica de clasificador.
 - Fusión a nivel de decisión.
 - Fusión a nivel de *ranking*.
 - Fusión a nivel de *matching score*.

6.1.1. Fusión a Nivel de *Matching Score*

Se denomina fusión a nivel de *matching score* a la consolidación de los *scores* de salida de diferentes *matchers* biométricos con el objetivo de llegar a una decisión final que tome en cuenta ambas fuentes.

Esto aplica también al caso en que un mismo *matcher* biométrico es utilizado, pero se obtienen los *scores* para más de una muestra. Por ejemplo, si se tiene el *score* de confrontar la huella de un dedo índice con la huella del dedo índice de una identidad almacenada, y también los *scores* con las huellas del dedo pulgar.

Al momento de hacer la fusión es importante analizar si los *scores* obtenidos son comparables de forma directa o si se necesita hacer algún tipo de transformación para que lo sean, o si conviene más trabajar con otra medida derivada de estos *scores* (como podrían ser probabilidades de alguna clase).

En el caso de uso en que los *scores* son obtenidos mediante el mismo *matcher* biométrico, es posible trabajar directamente con los resultados sin necesidad de hacer manipulaciones.

Existen varias técnicas para combinar los *scores* obtenidos, como tomar el máximo *score* y reportarlo como resultado de la fusión, tomar el mínimo, sumar todos los *scores* o hacer la productoria de los mismos [4].

También hay otros mecanismos más complejos como el uso de *likelihood ratio*, tal como lo presentan los autores en “*Likelihood Ratio-Based Biometric Score Fusion*” [27]. Este método se basa en el teorema de Neyman-Person, el cual permite fijar una tasa de falsas aceptaciones y definir el test óptimo para decidir si un vector de *matching scores* X corresponde a un usuario genuino o a un impostor. Se apoya en la estimación paramétrica (por ejemplo, un conjunto de funciones gaussianas) o no paramétrica (por ejemplo, utilizando KDE: *Kernel Density Estimation*) de densidades de genuinos y de impostores que es necesaria en el test.

Otra técnica elaborada es la usada en “*An A-Contrario Biometric Fusion Approach*” [11]. Los autores proponen extender el framework conocido como “a-contrario” para realizar fusión biométrica. Esta técnica es una aproximación estadística al problema, y anteriormente los mismos autores hicieron uso de la misma en “*A Statistical Approach to Reliability Estimation for Fingerprint Recognition*” [10] para estimar la confianza de la salida de un sistema biométrico para la identificación de individuos.

Una de las ventajas más importantes de la técnica a-contrario es que no se necesita la estimación de la distribución de los genuinos ya que trabaja solo con los impostores, por lo cual es más fácil, en muchos casos, lograr obtener una cantidad suficiente de datos como para aproximar las funciones de distribución. Otra ventaja es que se puede definir la información de cada individuo, minimizando las clasificaciones incorrectas debidas al problema del Zoológico de Doddington [14]. Este es un problema muy conocido en el área de biometría e implica que dada una característica biométrica particular, algunas personas son más difíciles de clasificar que otras, ya sea porque tienden a confundirse con otras identidades o porque es difícil asignarles una identidad.

6.1.2. Técnica de Fusión de Huellas Dactilares en la DNIC

En el caso particular de la DNIC, al momento de realizar la identificación de un individuo se utilizan técnicas de fusión de huellas dactilares solamente para el caso de mayores de 12 años. La fusión se realiza cuando no es posible identificar a la persona usando solamente la huella digital de un dedo.

Como la fusión biométrica en general ayuda a mejorar la confiabilidad del sistema, en los casos en los que no se usa, se exige que el *matching score* sea al menos el doble del umbral del sistema (3.400).

Si el *matching* se hace con dos o tres dedos, la técnica de fusión usada es hacer el promedio de los *scores* obtenidos, y comparar este valor contra el umbral.

Si en cambio el *matching* se hace con cuatro o más dedos, la técnica de fusión descarta los *scores* máximos y mínimos y luego realiza el promedio de los restantes. Finalmente, es este valor el que se compara contra el umbral.

En la figura 6.1 se presenta un diagrama de flujo con el algoritmo explicado anteriormente.

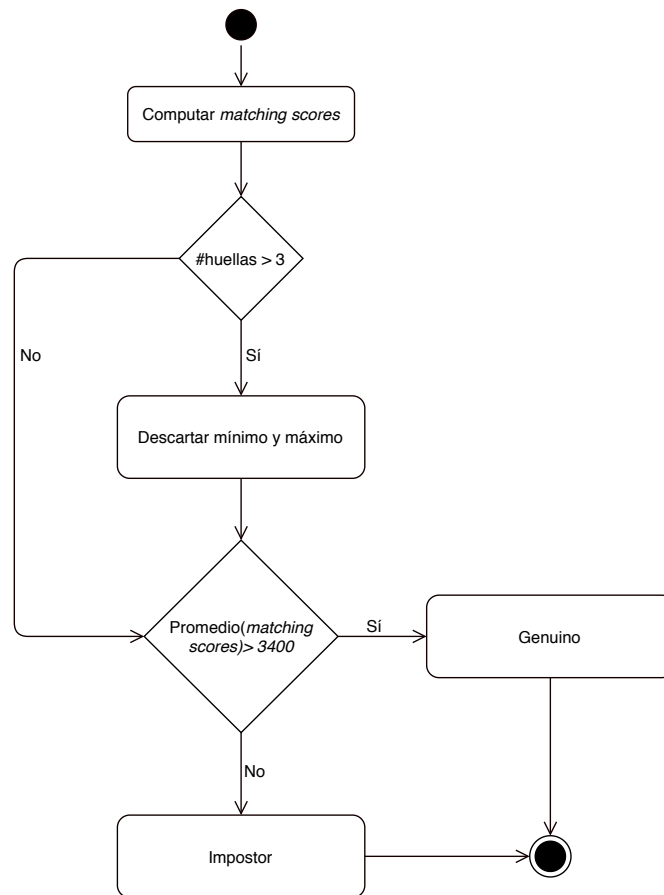


Figura 6.1: Diagrama de flujo del algoritmo de fusión utilizado en la DNIC.

6.2. Protocolo Experimental

Con el objetivo de determinar si un cambio en el método de fusión biométrica pudiera representar una mejora en el desempeño del sistema actual de la DNIC se evaluaron varias alternativas.

Para realizar este análisis, se comenzó seleccionando de forma aleatoria 500 identidades de la base de datos C (sección 3.1.3) que tuvieran, al menos, las huellas

1, 2, 6 y 7, que se corresponden con el pulgar e índice de la mano derecha e izquierda respectivamente. En la figura 6.2, se puede observar la distribución de edades de las muestras seleccionadas.

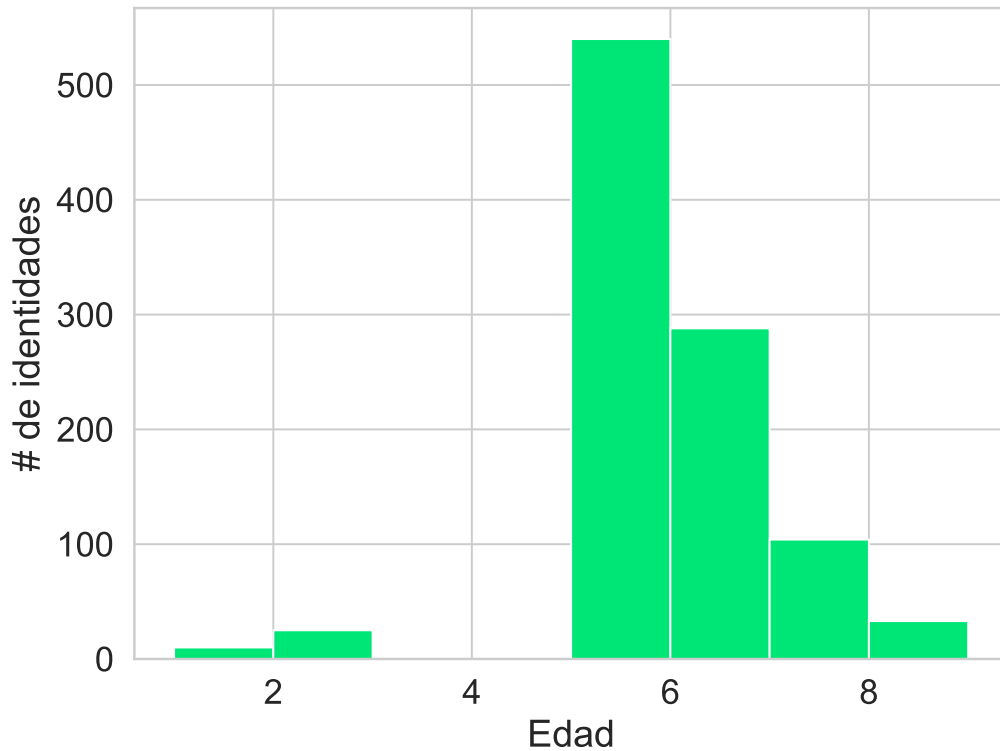


Figura 6.2: Distribución de edades en el subconjunto de 500 identidades utilizado.

Luego de obtener el conjunto de identidades con el que trabajar, se construyó, para cada dedo, una matriz de confusión con los resultados obtenidos de confrontar las huellas de las identidades utilizando el software de *matching* propietario usado en la DNIC. Estas matrices se utilizaron como fuente de datos para los distintos algoritmos de fusión que se probaron. Es así entonces que, para cada dedo, se obtuvieron 250.000 *matchings*. De estos, 249.500 corresponden a impostores y 500 a genuinos.

Para analizar realmente distintos algoritmos de fusión, se decidió comenzar implementando el algoritmo que la DNIC utiliza actualmente (detallado en sección 6.1.2). Esto se creyó necesario para tener como base los resultados de utilizar dicho algoritmo y usarlo como mínimo nivel aceptable.

El grupo docente proveyó de una implementación en Python de los algoritmos de fusión de suma, *likelihood ratio* y a-contrario. Esta implementación recibe las matrices de confusión de dos sistemas (por ejemplo huella 6 y huella 7) y retorna las curvas ROC generadas para cada uno de los algoritmos, así como también para cada huella por separado (es decir, el resultado que se obtendría sin aplicar fusión). Es importante destacar que, por como son los algoritmos de *likelihood ratio* y a-

contrario, se necesita primeramente entrenar el algoritmo y luego aplicarlo a un conjunto de prueba.

Dichos *scripts* se utilizaron, entonces, con la idea de comparar los tres algoritmos (suma, *likelihood ratio* y a-contrario) y evaluar cuál de estos brinda mejores resultados.

Como se detallará más adelante, de estos tres, el algoritmo de suma en general dio un resultado mejor o muy similar. Es por esto que, dado el alcance de este proyecto, se decidió implementar el algoritmo de suma de forma independiente para poder así compararlo con el de la DNIC y además utilizar 4 huellas, en lugar de 2.

Por último, se decidió probar un algoritmo más. Dicho algoritmo consiste en aplicar el mismo procedimiento que hace actualmente la DNIC a excepción de que también se utiliza el *matching score* más alto para el resultado final. Es decir, el algoritmo, para el caso de fusión con cuatro huellas, consiste en calcular los *matching scores* de cada par de dedos para ambas identidades y quitar del conjunto el resultado con valor más bajo. De esta forma, el *score* de fusión se obtiene realizando el promedio del conjunto de los tres *matching scores* restantes.

6.3. Análisis de Resultados

En la figura 6.3 se presentan los resultados obtenidos para el caso de la implementación del método de *likelihood ratio*, a-contrario y suma aplicados a las huellas 1 y 2 (dedos pulgar e índice derechos). Para los métodos de *likelihood ratio* (LR) y a-contrario (AC) se utilizó el modelo probabilístico *Gaussian Mixture Model* (GMM) para estimar la distribución de densidades de genuinos e impostores durante la fase de *test*.

En primer lugar, se puede observar que para cualquier método de fusión de los tres presentes en la gráfica, los resultados son muy superiores en comparación con utilizar una huella sola. Al comparar particularmente los tres algoritmos de fusión, para los datos utilizados se obtuvo un desempeño levemente mejor en el caso de la suma.

De esta forma, se procedió a realizar el experimento con los métodos de suma y de la DNIC utilizando las 500 identidades ya mencionadas y cuatro huellas para cada una. Las curvas ROC generadas en este caso se pueden observar en la figura 6.4.

Nuevamente, se destaca la notoria mejoría en el desempeño al utilizar fusión en comparación con no hacerlo. En el caso particular de estos dos métodos de fusión, la suma es la que tuvo mejores resultados.

Sin embargo, al agregar la curva ROC generada utilizando el algoritmo de fusión de la DNIC modificado (sin quitar del conjunto al *matching score* superior obtenido), se obtienen resultados incluso mejores (ver figuras 6.5 y 6.6)

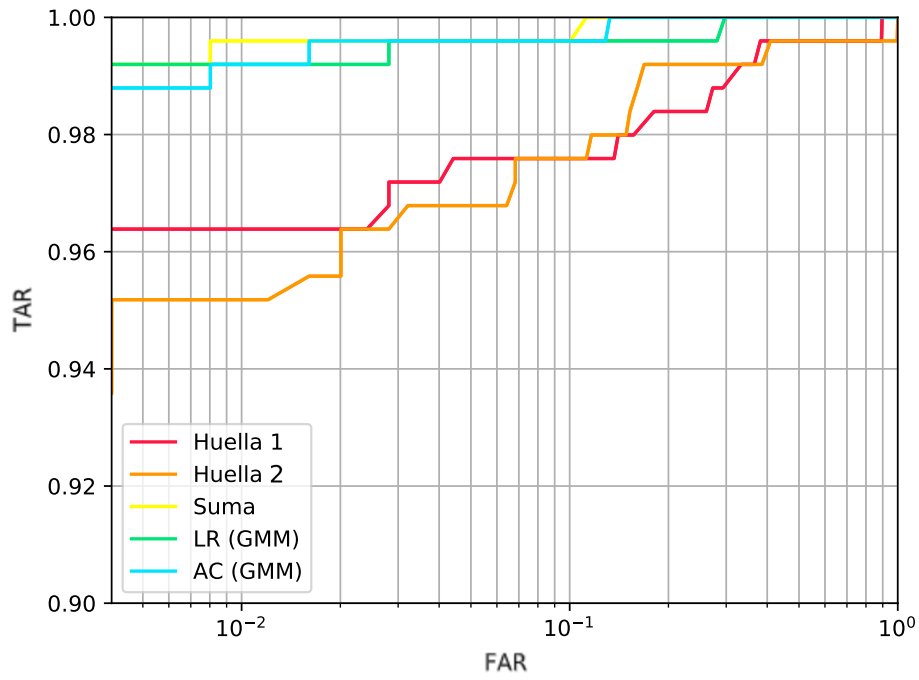


Figura 6.3: Curvas ROC obtenidas para dos huellas y métodos de fusión de suma, *likelihood ratio* (RT) y a-contrario (AC).

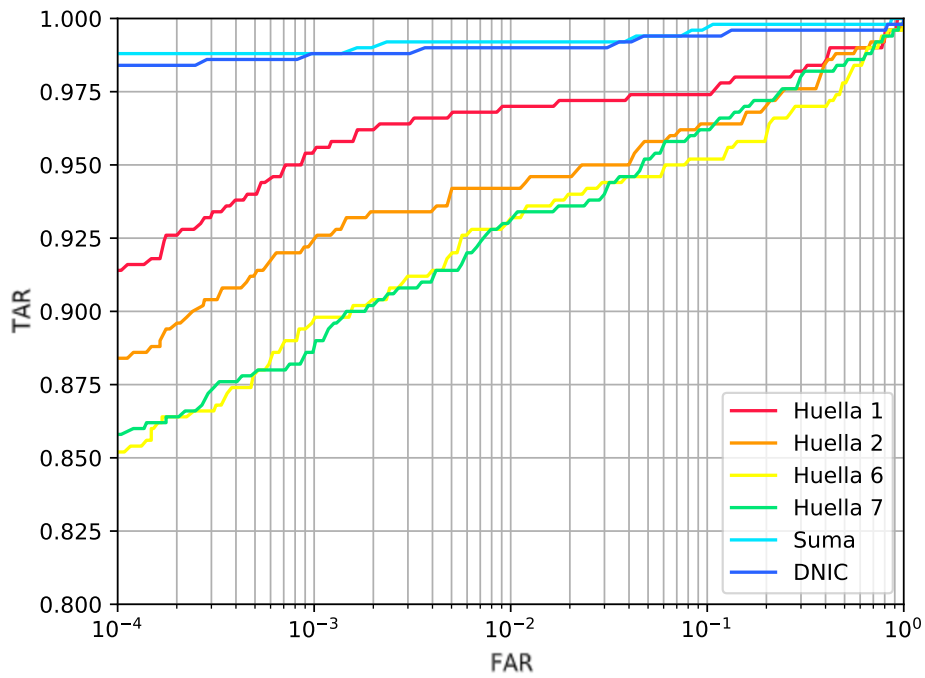


Figura 6.4: Curvas ROC obtenidas para cuatro huellas y métodos de fusión de suma y de la DNIC.

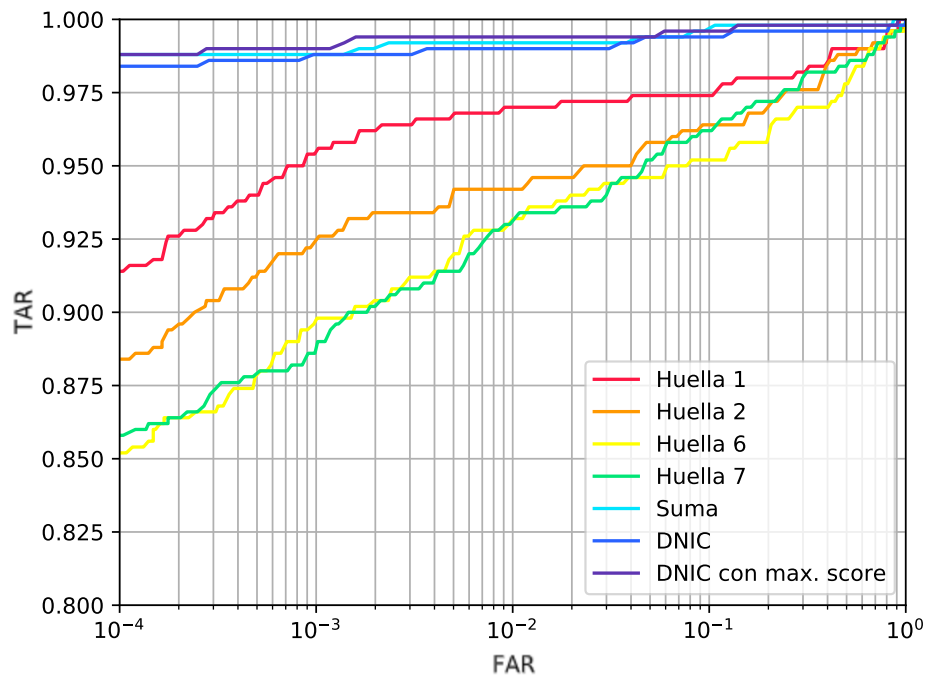


Figura 6.5: Curvas ROC obtenidas para cuatro huellas y métodos de fusión de suma, de la DNIC y DNIC con *max. score*.

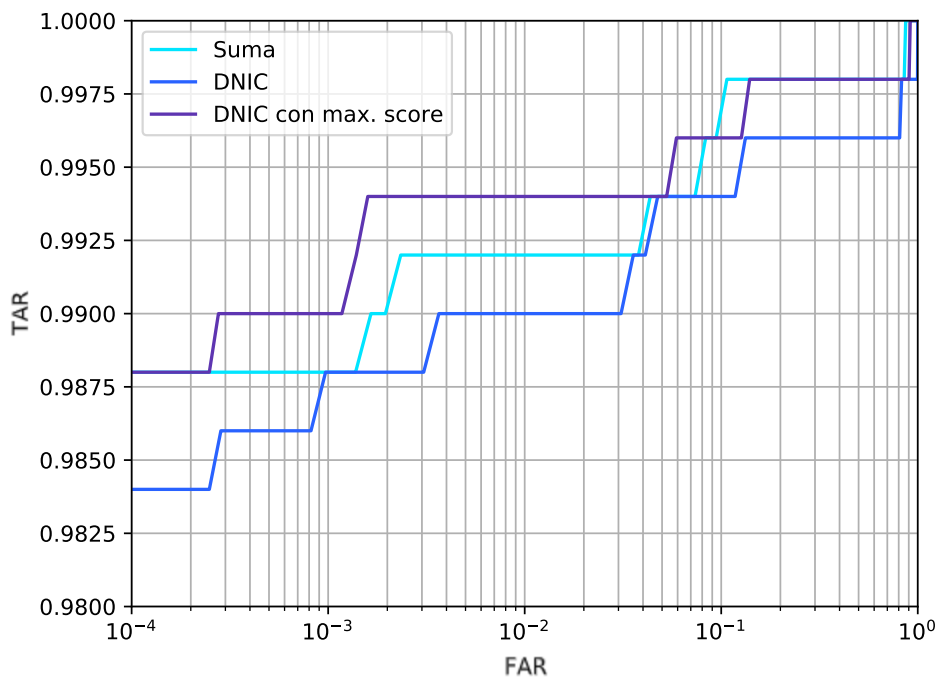


Figura 6.6: Curvas ROC obtenidas para los métodos de fusión de suma, de la DNIC y DNIC con *max. score*.

También se pueden comparar los TAR obtenidos para ciertos FAR específicos. Se reportan estos resultados en la tabla 6.7.

FAR	TAR DNIC	TAR suma	TAR DNIC con <i>max. score</i>
10e-4	0,984	0,9881	0,9881
10e-3	0,9881	0,9881	0,990
10e-2	0,988	0,990	0,994
0,03	0,990	0,992	0,994
0,07	0,994	0,994	0,996
0,12	0,994	0,998	0,996
1,0	1,0	1,0	1,0

Figura 6.7: FAR y TAR para los tres métodos de fusión.

Al evaluar los valores obtenidos, se observa que la mayor diferencia entre los tres métodos se ve, sobre todo, para los valores de FAR más bajos. A medida que el FAR se aproxima a 1, los valores de los métodos se acercan cada vez más; sin embargo, para todos los valores de FAR utilizados, el algoritmo de la DNIC toma valores iguales o menores que los obtenidos por el algoritmo de DNIC que incluye el *score* máximo, e incluso que los de suma.

Este resultado es interesante ya que se trata de una modificación mínima al algoritmo usado actualmente y que logra una mejora en el desempeño. Es relevante notar que si bien la mejora puede parecer pequeña, se está trabajando sobre un desempeño que ya era muy bueno, por lo que suele ser muy trabajoso lograr mejoras. En este caso, una pequeña modificación ya puede cumplir con eso.

6.4. Propuesta de Implementación

Dados los resultados obtenidos para esta área de trabajo, se propone considerar la posibilidad de realizar un leve cambio en el algoritmo de fusión utilizado actualmente por la DNIC.

Actualmente, el algoritmo para realizar fusión con cuatro o más huellas consiste en calcular el *matching score* entre cada par de huellas de los dedos correspondientes, descartar el mínimo y el máximo *score* y, finalmente, realizar el promedio entre los valores restantes.

El cambio propuesto es no descartar el máximo *score* calculado, ya que los resultados obtenidos muestran una mejora interesante al mantenerlo, en comparación con el algoritmo actual.

Como la propuesta sería una mínima alteración a lo ya utilizado, se decidió no re-implementar todo el algoritmo, sino que aportar el análisis correspondiente con la sugerencia.

De todas formas, es importante mencionar que también sería bueno poder realizar el mismo protocolo experimental hecho en esta sección con una base de datos con más de 500 identidades.

7. Compresión

En este capítulo se presenta el área de compresión, comenzando con el marco teórico sobre los dos algoritmos de compresión estudiados: WSQ y JPEG 2000, seguido por el protocolo experimental realizado, detallando cómo se compararon dichos algoritmos. Finalmente, se realiza un análisis de los resultados obtenidos al comparar ambos algoritmos, y se realiza la sugerencia de evaluar el uso de JPEG 2000 para el caso de huellas de 5 años de edad, ya que para este caso su desempeño parece ser mejor que el de WSQ (algoritmo utilizado en la actualidad).

7.1. Marco Teórico

En el artículo “*Compression Guidance for 1000 ppi Friction Ridge Imagery*” [9] se constata que, actualmente, el algoritmo estándar para la compresión con pérdida de imágenes de huellas dactilares es WSQ, por su nombre en inglés “*Wavelet Scalar Quantization*”. Su especificación provee lineamientos basados en estudios que establecen la cantidad aceptable de pérdida de la fidelidad debido a la compresión de forma tal que un codificador y decodificador de WSQ cumpla con la certificación del FBI. Justamente, WSQ fue desarrollado por el FBI en coordinación con el NIST en la década del 1990. WSQ fue específicamente creado para comprimir imágenes de huellas dactilares de 8 bits, en escala de grises y de **500 dpi** [44].

Por otro lado, los sistemas biométricos modernos están tendiendo, en los últimos tiempos, a aumentar la resolución de las imágenes de huellas dactilares, pasando de 500 dpi a al menos **1000 dpi**. Este aumento en la resolución provee de una mayor fidelidad de la imagen y mejor representación de las características conocidas como de “nivel 3”¹, lo cual hace que haya mayor probabilidad de que tanto sistemas automáticos como expertos dactiloscopos puedan decidir si una huella coincide o no con una identidad. Además, como se estudió en “*Fingerprint Biometrics From Newborn to Adult: A Study From a National Identity Database System*” [33], el uso de imágenes de 500 dpi hace que las huellas de niños pequeños y, especialmente aquellos menores de 1 año, sean de muy baja fidelidad resultando de poca utilidad para los procesos automáticos de identificación y verificación.

Un problema con la adopción de una mayor resolución para la captura y almacenamiento de las imágenes de huellas dactilares es que el algoritmo de

¹Las características de nivel 1 son el patrón y flujo general de las crestas papilares; las de nivel 2 incluyen detalles como los puntos de minucia; y las de nivel 3 alcanzan el detalle de poros, arrugas, forma de las líneas, crestas incipientes y otras características no incluidas en los primeros dos niveles.

compresión WSQ fue diseñado específicamente para ser usado en imágenes de 500 dpi como se mencionó anteriormente, por lo que se necesita un nuevo algoritmo de compresión para trabajar a mayor resolución. Los estándares *ANSI/NIST ITL* [3] e *ISO 19794* definen el uso de JPEG 2000 como algoritmo de compresión en lugar de WSQ. JPEG 2000 ofrece mayor flexibilidad que WSQ en el tipo de imágenes con las que opera así como en las formas que las mismas pueden ser comprimidas y codificadas, lo cual es ventajoso pero a su vez ofrece muchos grados de libertad que pueden, potencialmente, generar problemas de compatibilidad en los sistemas.

Como es de esperarse, los sistemas actuales que quieran empezar a hacer uso de esta ganancia en fidelidad han de pasar por un período de transición, donde imágenes adquiridas en 500 dpi deben ser interoperables con imágenes obtenidas a mayor resolución. En “*Compression Guidance for 1000 ppi Friction Ridge Imagery*” [9] se proveen los lineamientos para la compresión de imágenes de 1000 dpi así como un camino para la interoperabilidad entre imágenes de 500 dpi y de 1000 dpi.

Como lineamientos generales, este mismo artículo recomienda el uso de compresión con pérdida 10:1 para imágenes de huellas dactilares de 1000 dpi. Además, para su interoperabilidad con 500 dpi, también recomienda el uso de un filtro Gaussiano pasabajo² ($\sigma = 0,8475$ y $\tau = 4$) combinado con un submuestreo³ de 10:1, para convertir las imágenes de 1000 dpi a 500 dpi.

7.2. Protocolo Experimental

Para realizar el análisis de los distintos algoritmos para la compresión de imágenes de huellas dactilares, con el objetivo de evaluar si un cambio en el algoritmo de compresión puede significar una mejora en el desempeño en el sistema para el caso de niños, se optó por utilizar la base de datos D (sección 3.1.4). Esta base de datos cuenta con imágenes de huellas dactilares de adultos sin comprimir y para cada huella se cuenta con, al menos, dos tomas, por lo que permite realizar los estudios pertinentes de esta sección.

Por las características de los experimentos de interés, se optó por usar solamente los datos de las dos primeras tomas de forma tal de evitar contaminar los resultados con problemas específicos debido a las condiciones de captura de las otras tomas.

Con esto, la base de datos final utilizada para los experimentos cuenta con 24 identidades, con dos tomas de los diez dedos de cada una, totalizando 480 huellas.

²En procesamiento de señales, un filtro es un artefacto o proceso que elimina alguna componente no deseada de una señal. En el caso de un filtro Gaussiano, se utiliza una función Gaussiana para determinar qué componentes remover.

³Toma de muestras a una tasa o frecuencia constante.

7.2.1. Comparación de Calidad

Con el objetivo de evaluar el comportamiento de los algoritmos de compresión WSQ y JPEG 2000 en huellas de niños mediante la calidad de las huellas resultantes, se generó una base de datos, partiendo de la base de datos original, en donde se simula la huella de un niño a partir de la huella de adulto con la que se cuenta.

Para cada huella de adulto disponible, se la escaló a tamaño de niño usando inversamente los factores de la tabla 5.2. De esta forma, se obtuvieron huellas simuladas de niños para 1 año, 2 años y 5 años.

Posteriormente, se comprimieron estas huellas usando WSQ y JPEG 2000, ambos con un factor de compresión 12 (utilizado por la DNIC), para luego descomprimirlas para poder calcular la calidad con el algoritmo del software propietario utilizado en la DNIC.

7.2.2. Comparación de Desempeño

Para comparar los algoritmos de compresión a nivel de desempeño se generó una nueva base de datos partiendo de la original en la que, para cada una de las 24 identidades, se construyeron dos tomas de diez dedos cada una. Para la primera toma se tomaron las huellas del conjunto A de la base de datos y se las escaló a tamaño de niño como se hizo para comparar la calidad. Para la segunda toma se utilizó el conjunto B de la base de datos sin modificar. Con esto se intentó simular el caso más común en la realidad en el que hay diferencia de edad entre la primera toma y la segunda de una persona al momento de hacer el *matching*.

A las primeras tomas se las comprimió en WSQ y en JPEG 2000 con factor 12, mientras que a las segundas solo con WSQ por ser de adulto.

Luego, se utilizó el software de *matching* propietario usado por la DNIC y se hizo el estudio de desempeño mediante curvas ROC haciendo *matching* de todos contra todos, es decir, para cada huella de primera toma, se hizo el *matching* de ella contra todas las huellas del mismo dedo de todas las identidades en la segunda toma.

7.2.3. Factor Efectivo de Compresión

Como se explica en el artículo “*A Comparative Performance Analysis of JPEG 2000 vs. WSQ for Fingerprint Image Compression*” [16], WSQ tiende a comprimir con un factor efectivo mayor al factor real provisto como parámetro al algoritmo, y dicho factor no es predecible antes de correr el algoritmo. Por esto, se diseñó un tercer experimento en el que se hizo exactamente lo mismo que para los dos anteriores pero, al comprimir con JPEG 2000 se utilizó una aproximación del factor efectivo utilizado por WSQ.

La aproximación del factor efectivo se calculó como el promedio de los factores

efectivos aplicados a cada imagen escalada y comprimida con WSQ. El factor efectivo de una imagen se calculó como la razón del tamaño original de la imagen y el tamaño comprimido.

Esto dio como resultado que, para todas las edades simuladas, el factor efectivo con el que comprimió WSQ fue, aproximadamente, 26.

7.3. Análisis de Resultados

Al comparar la calidad de las huellas comprimidas con JPEG 2000 y con WSQ para las condiciones del experimento de comparación de calidad, se obtuvieron los resultados que se pueden ver en la tabla 7.1.

	5 años	2 años	1 año
Mejor WSQ	74	107	132
Mejor JPEG 2000	231	191	172
Iguales	175	182	176

Figura 7.1: Cantidad de veces con mejor calidad por algoritmo para el experimento de comparación de calidad.

En general, la calidad de JPEG 2000 en estos casos fue mejor o igual a la de WSQ, aunque se observa que a medida que se trabaja con edades más chicas, WSQ se acerca a JPEG 2000. De todas formas, es interesante ver que ya sea cuando uno u otro algoritmo da mejor calidad, la diferencia entre uno y otro es mínima, pudiéndose ver la distancia promedio entre los valores y la varianza en la tabla 7.2.

	5 años	2 años	1 año
Distancia promedio	0,952	0,977	0,919
Varianza	1,062	1,514	1,025

Figura 7.2: Promedios y varianzas de la distancia entre valores de calidad para el experimento de comparación de calidad.

En el caso de las condiciones del experimento utilizando el factor efectivo de compresión, los resultados se pueden ver en las tablas 7.3 y en 7.4.

	5 años	2 años	1 año
Mejor WSQ	151	257	266
Mejor JPEG 2000	173	77	79
Iguales	156	146	135

Figura 7.3: Cantidad de veces con mejor calidad por algoritmo con factor de compresión efectivo.

En este caso, es WSQ el que se comporta mejor o igual a JPEG 2000, lo cual sugiere que a igualdad de nivel de compresión efectiva, WSQ obtiene mejores niveles de calidad, aún para niños.

	5 años	2 años	1 año
Distancia promedio	1,114	1,393	1,469
Varianza	1,368	2,201	2,270

Figura 7.4: Promedios y varianzas de la distancia entre valores de calidad con factor de compresión efectivo.

En las figuras 7.5, 7.6 y 7.7 se pueden observar las curvas ROC correspondientes a las huellas sin comprimir (BMP), comprimidas con WSQ (WSQ), comprimidas con JPEG 2000 con factor de compresión 12 (JPEG 2000 factor 12) y comprimidas con JPEG 2000 usando factor de compresión 26 (JPEG 2000 factor 26), tanto para 5 años, 2 años y 1 año, respectivamente.

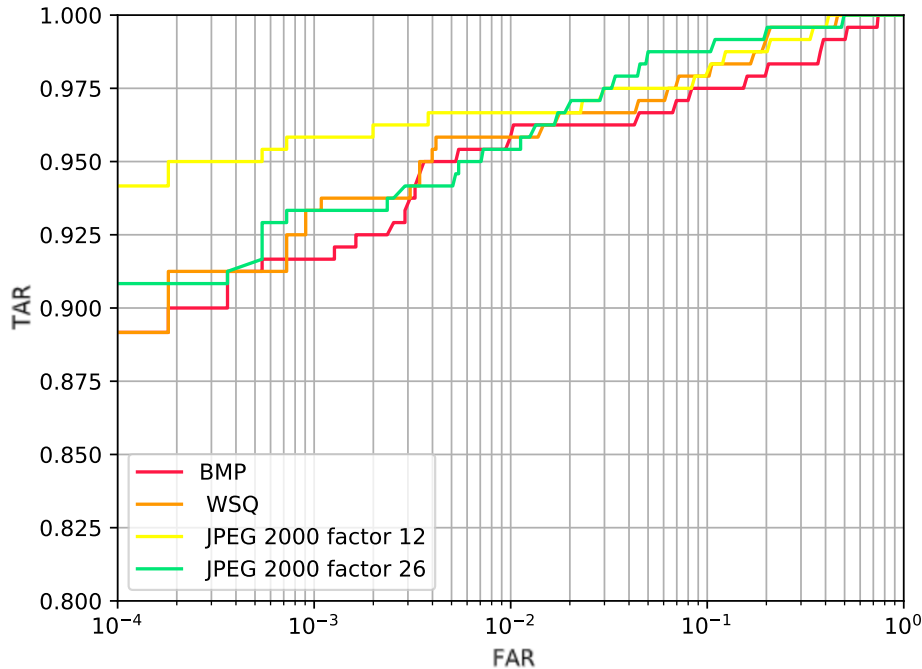


Figura 7.5: Curvas ROC para cada algoritmo de compresión con imágenes simuladas de 5 años.

Como era esperable según los resultados de calidad observados, JPEG 2000 tiende a tener un mejor desempeño que WSQ cuando se utiliza el mismo factor de compresión. En el caso de 1 año, esta tendencia no es clara pero puede deberse a que para tamaños de huellas tan chicos, tanto los compresores como el algoritmo de *matching* tengan problemas, como se menciona en el artículo “*Fingerprint Biometrics From Newborn to Adult: A Study From a National Identity Database System*” [33]. De todas formas, este mejor desempeño está sesgado por el hecho de que JPEG 2000 comprimió menos que WSQ con el mismo factor.

Cuando se utiliza el factor de compresión efectivo de WSQ con JPEG 2000, se puede observar que están muy cerca uno del otro en todos los casos, y no hay un

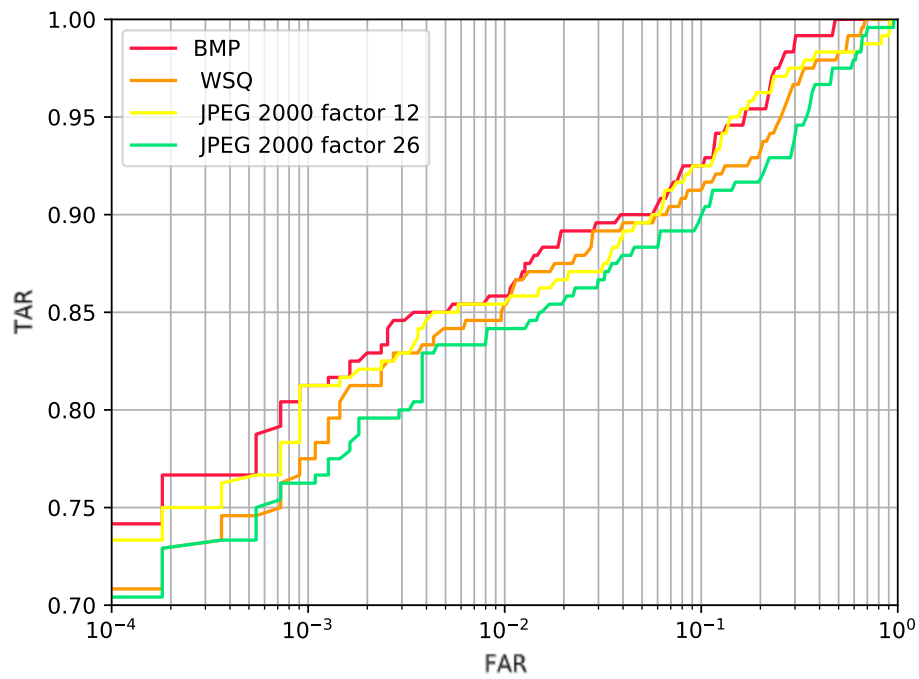


Figura 7.6: Curvas ROC para cada algoritmo de compresión con imágenes simuladas de 2 años.

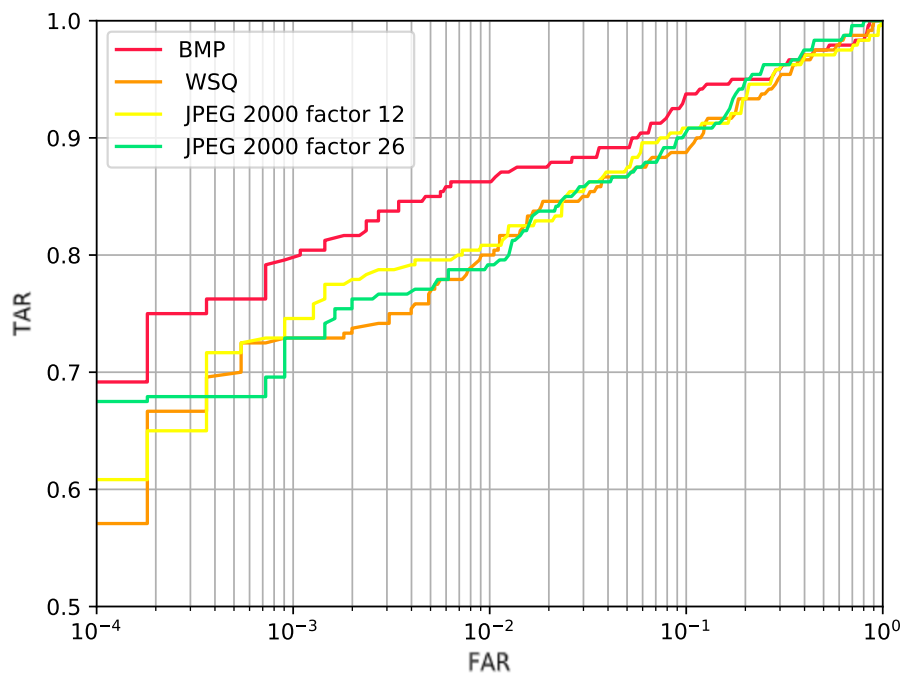


Figura 7.7: Curvas ROC para cada algoritmo de compresión con imágenes simuladas de 1 año.

algoritmo que esté por encima del otro siempre.

En el caso de 5 años, hay indicios de que JPEG 2000 se comportaría mejor que WSQ para valores de FAR cercanos a 10^{-4} . Este resultado es interesante ya que la gran mayoría de imágenes de huellas dactilares con las que cuenta la DNIC en su base de datos fueron tomadas a los 5 años de edad. Por esto, comprimir estas imágenes con JPEG 2000 en lugar de WSQ podría mejorar el desempeño para una porción significativa de los datos.

Para 1 y 2 años, los resultados no son claros ya que, por ejemplo, se esperaría que cuanto más se comprime, el desempeño fuera peor. Esto se da para el caso de 2 años, pero no para 1 año. Como ya se mencionó anteriormente esto se puede deber a que las huellas de 1 año de 500 dpi no tienen la suficiente calidad como para que los algoritmos funcionen correctamente.

7.4. Propuesta de Implementación

En este caso, no se tiene una propuesta concreta para implementar en el sistema actual de la DNIC. Esto se debe a que los resultados obtenidos con las distintas herramientas son muy similares. Lo que sí existe es una sugerencia de replicar los experimentos con datos reales de huellas dactilares de 5 años sin comprimir para validar que realmente JPEG 2000, utilizando el factor de compresión efectivo de WSQ, mejora el desempeño en este caso. De ser así, se recomienda comenzar a utilizar JPEG 2000 para esta edad.

Además, se recomienda fuertemente realizar todos los experimentos nuevamente, con una base de datos con imágenes sin comprimir que sean realmente de niños, es decir, no simulada. También sería ideal ver los resultados para estos experimentos utilizando bases de datos con imágenes de tanto 1000 dpi como 2000 dpi, ya que, en el contexto de este proyecto, no fue posible conseguir una base de datos de estas características.

8. Implementación

En este capítulo se describe la arquitectura actual del sistema biométrico que se encuentra en funcionamiento en la DNIC, y los diferentes componentes de software implementados para, potencialmente, ser integrados en el sistema de la DNIC. En particular, estos componentes incluyen pruebas unitarias, las cuales aportan documentación adicional en el caso que un desarrollador luego quiera modificar las mismas. A su vez, asegura que en el caso de modificaciones, las operaciones implementadas continúen funcionando de la forma esperada, logrando así mejor mantenibilidad. También, se describen los componentes de software implementados para los diferentes experimentos.

8.1. Arquitectura del Sistema de la DNIC

En esta sección se describe, a grandes rasgos, la arquitectura del sistema biométrico utilizado en la DNIC, incluyendo las DLL implementadas en el contexto de este proyecto. Conocer esta arquitectura fue necesario para poder implementar software que se pudiera integrar fácilmente.

La DNIC tiene, principalmente, dos sistemas biométricos de huellas dactilares: uno que se utiliza en las oficinas y otro para equipos móviles.

En la figura 8.1 se muestra un diagrama de la arquitectura del sistema de las oficinas de la DNIC. Primeramente, el módulo de captura de huellas es un archivo ejecutable de Windows (.exe) implementado con .NET que, luego de tomar las huellas, las guarda en el sistema de archivos de la computadora en la que se esté ejecutando. Este módulo se comunica con el sistema de trámites.

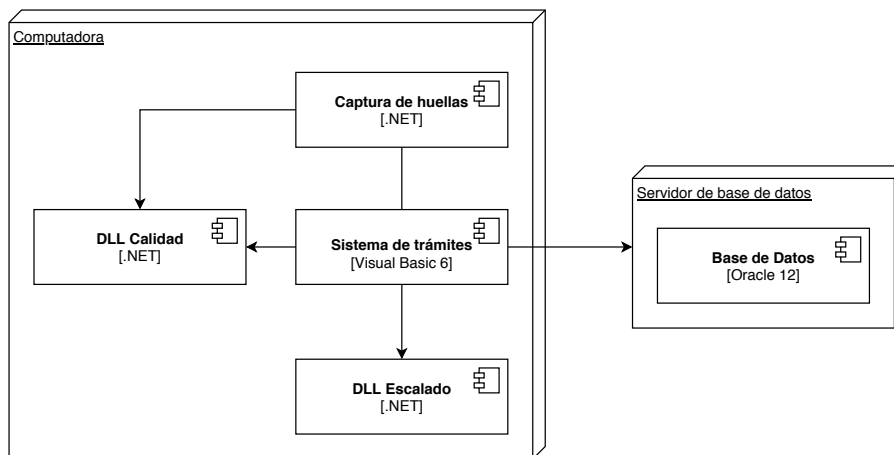


Figura 8.1: Arquitectura del sistema biométrico de las oficinas de la DNIC incluyendo las DLL implementadas.

El sistema de trámites es una aplicación monolítica¹, también construida como un archivo ejecutable (.exe), implementada en Visual Basic 6 que, entre otras cosas, es responsable de guardar todos los datos en la base de datos. En este caso, se utiliza el manejador de bases de datos Oracle, en su versión 12.

Dada esta arquitectura, las DLL implementadas en el contexto de este proyecto deben ser instaladas en las computadoras donde se ejecuta el sistema para así poder ser invocadas.

La DLL de escalado sería invocada por el sistema de trámites ya que este es el que tiene la fecha de nacimiento del tramitante y la fecha en la que fue tomada la huella. Estas fechas son necesarias para poder saber la edad de la persona al tomar la huella dactilar y así poder utilizar el factor de interpolación adecuado durante el escalado. Un ejemplo de uso de esta DLL podría ser en el proceso de filiación² y toma de huellas de menores.

La DLL de calidad, por otro lado, podría ser invocada tanto desde el sistema de trámites como del módulo de captura de huellas. También se podría utilizar tanto para huellas de niños como para huellas de adultos.

Respecto a los equipos móviles, estos son usados para casos en los que se debe tomar huellas a personas que, por alguna razón, no pueden asistir a las oficinas. Este es el caso, por ejemplo, de personas internadas en algún centro asistencial o privados de libertad.

Debido a las características de este sistema, se debe utilizar una arquitectura distinta a la utilizada en el sistema de las oficinas. En la figura 8.2, se presenta un diagrama de la misma.

En el caso del módulo de captura de huellas, no hay diferencias con el utilizado en oficinas. Sin embargo, la aplicación con la que este se comunica sí es muy diferente. Esta aplicación, desarrollada en .NET, está compuesta por tres capas: aplicación (utilizando WinForms³), negocio y acceso a datos.

La capa de acceso a datos utiliza el *framework* para ORM (*Object-Relational Mapping*: mapeo objeto-relacional) de .NET *Entity Framework* y se comunica con la base de datos, para la cual se usa el manejador SQL Server.

Finalmente, ambas DLL construidas serían invocadas por la aplicación .NET previamente descripta.

¹Una aplicación monolítica hace referencia a una aplicación en la que tanto la capa de interfaz de usuario y la de acceso a datos están combinadas en un mismo programa y sobre una misma plataforma.

²Proceso en el cual se llena una ficha con los datos personales de una persona y se reconoce la relación de parentesco entre padres e hijos.

³*Framework* de interfaz de usuario usado para construir aplicaciones de escritorio Windows.

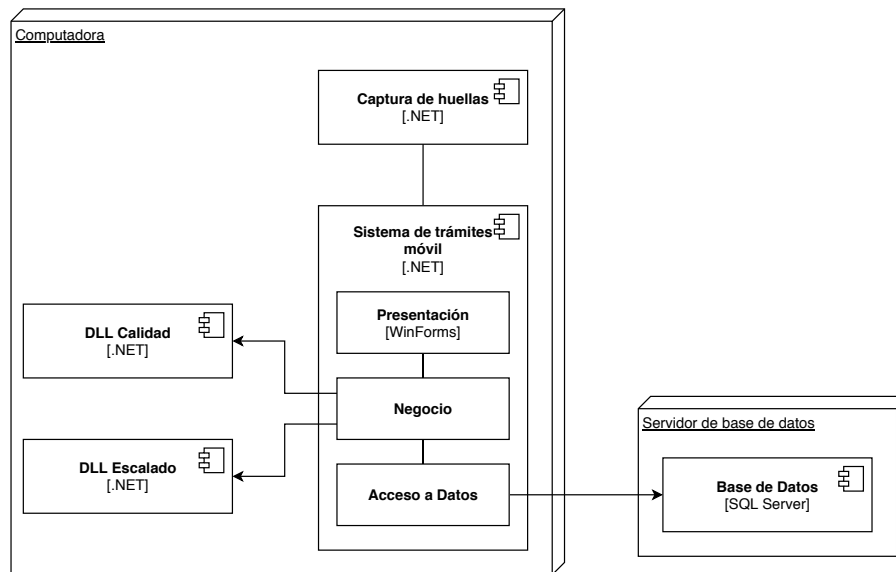


Figura 8.2: Arquitectura del sistema biométrico móvil de la DNIC.

8.2. DLL de Escalado Lineal

El objetivo de esta DLL es proveer una forma simple de escalar una imagen de una huella dactilar dada la edad del individuo en el momento que se tomó la misma. Esta edad se utiliza para determinar el factor de escalado descrito en la tabla 5.2, con la leve modificación de utilizar una simple interpolación lineal para definir el factor en valores no determinados, quedando definidos como en la tabla 8.3.

Edad	Factor de escalado
0-2 meses	1,65
3 meses	1,58
4 meses	1,59
5 meses	1,60
6-11 meses	1,54
1 año	1,49
2 años	1,47
3-4 años	1,40
5 años	1,32
6 años	1,29
7 años	1,26
8 años	1,24
9 años	1,22
10 años o más	1

Figura 8.3: Factores de escalado utilizados en la DLL de escalado lineal.

8.2.1. Relevamiento de Requerimientos

Para acordar las operaciones que fueran de utilidad para la DNIC se realizaron algunas reuniones de relevamiento de requerimientos. En ellas, se discutió en qué parte de su proceso se incluiría esta DLL, para así poder identificar los parámetros de entrada y las salidas esperadas.

De estas reuniones se recabó que las imágenes a escalar se deberían aceptar tanto como un *path* al archivo así como un objeto del tipo *Image*. De la misma forma, su correspondiente imagen escalada se debe poder tanto guardar como un archivo en un *path* dado, así como retornar la misma como un objeto del tipo *Image*.

Dado que el escalado depende de la edad al momento en que se tomó la imagen dada, se optó por recibir la fecha de nacimiento del individuo y la fecha en que se realizó la toma, para así poder calcular la edad. También, era de interés para la DNIC poder pasar como parámetro un factor de interpolación específico, para así poder probar y ver el resultado de diferentes factores.

Como dato no menor se obtuvo que la DLL debería ser implementada utilizando .NET en su versión 4.6.2, ya que esta es la versión que se utiliza en la DNIC.

8.2.2. Operaciones Implementadas

Luego de relevar los requerimientos, se definieron e implementaron las siguientes operaciones:

- **EscalarAByteArray**: permite, dada una imagen, escalarla utilizando el factor correspondiente a la edad del individuo cuando se tomó la huella y retornarla como *byte array*.
- **EscalarABitmap**: permite, dada una imagen, escalarla utilizando el factor correspondiente a la edad del individuo cuando se tomó la huella y retornarla como *Bitmap*.
- **EscalarGuardarByteArray**: permite, dada una imagen, escalarla utilizando el factor correspondiente a la edad del individuo cuando se tomó la huella y guardar la nueva imagen escalada como *byte array*.
- **EscalarGuardarBitmap**: permite, dada una imagen, escalarla utilizando el factor correspondiente a la edad del individuo cuando se tomó la huella y guardar la nueva imagen escalada como *bitmap*.
- **EscalarAImage**: permite, dada una imagen, escalarla utilizando el factor correspondiente a la edad del individuo cuando se tomó la huella y retornarla como *Image*.
- **EscalarAImageConFactor**: permite escalar una imagen dada con un factor dado y retornarla como *Image*.

Cabe destacar que, si bien en un principio se deseaba retornar las imágenes escaladas como un objeto *Image*, también se optó por tener operaciones que retornaran las mismas en *byte array* para proveer más opciones de uso y aumentar la flexibilidad de la DLL implementada.

8.2.3. Verificación y Validación

Para verificar que las operaciones funcionaran de la forma esperada, se implementaron pruebas unitarias utilizando el *framework* MSTest. Este provee funciones para confirmar que las funciones o métodos retornan el resultado esperado.

Para implementar dichas pruebas se utilizó el patrón *Arrange-Act-Assert* (AAA de aquí en más), el cual sugiere que las pruebas se deben dividir en tres secciones:

- **Arrange:** en este primer paso se prepara todo lo necesario para ejecutar la prueba. Por ejemplo, crear un objeto de la clase a probar.
- **Act:** aquí se invoca la función o método a probar utilizando lo creado en el paso anterior.
- **Assert:** en este último paso se verifica que el resultado obtenido en el anterior paso sea el esperado.

Un desafío particular fue encontrar la forma correcta de probar que las imágenes se estuvieran escalando correctamente. Para esto, se optó por comparar las dimensiones esperadas (en píxeles). Es decir, si la dimensión de la imagen original era de 500×500 píxeles y la edad del individuo al momento en que se tomó la imagen dada es de 5 años, se espera que la imagen resultante sea de 660×660 píxeles. Se espera esto ya que el factor de interpolación para 5 años es de 1,32, por lo cual se tiene $500 \cdot 1,32 = 660$ píxeles.

8.3. DLL de Calidad

El objetivo de esta DLL es permitir el uso del algoritmo de calidad NFIQ2 desde un programa ejecutando en Windows. Como NFIQ2 funciona solamente sobre los sistemas operativos CentOS, Ubuntu y MacOS, lograr esto no es trivial ya que, idealmente, es necesario poder compilar el código fuente de dicho algoritmo en Windows. Se evaluó la posibilidad de modificar el código fuente de NFIQ2 para que éste compile en Windows, pero luego de analizarlo, se decidió que el esfuerzo necesario para lograrlo sería demasiado alto y no sería factible a largo plazo. Debido a esto, se procedió a explorar otra solución.

Se optó por utilizar Docker como alternativa. Docker es un conjunto de productos gratuitos que, utilizando virtualización a nivel del sistema operativo, permite crear paquetes de software llamados contenedores. Estos encapsulan todo

lo necesario para ejecutar una aplicación sobre cualquier sistema operativo, sin importar el de la máquina donde se esté ejecutando.

Como Docker permite ejecutar aplicaciones sobre cualquier sistema operativo, se pudo crear un contenedor que utilizara CentOS, permitiendo instalar NFIQ2 en él. Con esto se puede invocar dicho contenedor y hacer uso de NFIQ2 desde un entorno Windows.

De esta forma, la arquitectura de esta DLL queda compuesta por dos componentes: la DLL en sí y el contenedor de Docker con NFIQ2 instalado (se puede observar un diagrama en la figura 8.4).

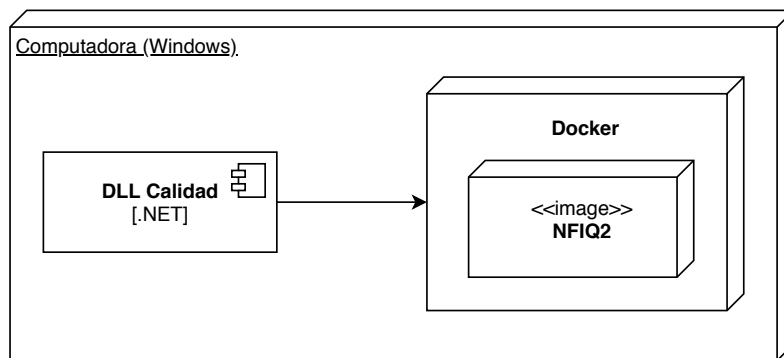


Figura 8.4: Arquitectura de la DLL de Calidad.

8.3.1. Relevamiento de Requerimientos

Hoy en día, en la DNIC se utiliza NFIQ1 para medir la calidad de las imágenes de huellas dactilares. Dado que este algoritmo tiene un rango de solamente cinco valores para evaluar la calidad, y que la actual tendencia es utilizar NFIQ2 por ser este una nueva versión, era de interés para la DNIC poder usar este último.

Para ello, se decidió proveer mediante esta DLL acceso a los dos métodos provistos por NFIQ2: modo simple y modo *batch*. Finalmente, al igual que la DLL anterior, esta también debería ser implementada en .NET en su versión 4.6.2.

8.3.2. Operaciones Implementadas

Las operaciones que se implementaron son las siguientes:

- **ModoSimple:** permite calcular la calidad de la imagen dada utilizando NFIQ2, retornando un entero entre 0 y 100, o 255 en caso que el algoritmo falle.
- **ModoBatch:** permite calcular la calidad de una lista de imágenes dadas utilizando NFIQ2, retornando una lista de enteros entre 0 y 100, 255 en caso que el algoritmo falle o -1 si la imagen no es válida.

Debido a que el contenedor de Docker necesita acceso a las imágenes de huellas dactilares a las cuales se les quiere calcular la calidad (esto se realiza vinculando una carpeta de la máquina al contenedor), se optó por generar una nueva carpeta temporal en cada ejecución con dicha imagen (o imágenes).

8.3.3. Verificación y Validación

Al igual que en la DLL de escalado lineal, se utilizó el *framework* MSTest y el patrón AAA para implementar pruebas unitarias.

Luego, para validar que la arquitectura de esta DLL es viable, se instaló Docker tanto en Windows 8 como en Windows 10, siendo este último el usado en la DNIC, logrando utilizar exitosamente el contenedor generado en ambas versiones del sistema operativo Windows.

Finalmente, también se creó un nuevo proyecto WinForms y se invocaron todas las operaciones desde el mismo, con el objetivo de tener una verificación adicional de que la DLL sea compatible con la arquitectura de la DNIC.

8.4. *Scripts* para Análisis

Para llevar a cabo las distintas pruebas y análisis en los diferentes tópicos en los que se trabajó, se realizaron *scripts* que utilizan la base de datos o archivos específicos de acuerdo a su función. En esta sección se describe brevemente la forma en la que el código implementado está organizado.

Se siguió una estructura modular de forma tal que se tuviera el código organizado y se pudieran reutilizar módulos o funciones específicas de ellos con tan solo importar los archivos necesarios entre los distintos *scripts*.

Dentro de la carpeta principal del proyecto se cuenta con una carpeta *scripts* donde se encuentra organizado todo el código que se utilizó para los análisis llevados a cabo en este proyecto. Dentro de esa carpeta hay otras, organizadas según utilidad, que funcionan como directorios de paquetes de Python⁴; para esto, cada directorio contiene un archivo `__init__.py`.

En la figura 8.5 se muestra la estructura de carpetas principales, y se puede observar que los módulos se dividieron, principalmente, de acuerdo a las cuatro áreas temáticas centrales que fueron abarcadas en el proyecto: compresión, fusión, escalado y calidad. Además, hay otros dos módulos principales: *Segmenter*, que contiene un segmentador de imágenes de huellas dactilares implementado en Matlab y brindado por el Ing. Guillermo Garella, y *Utils*, que abarca *scripts* y funciones genéricas que pueden ser utilizados en todas las áreas.

⁴Los paquetes, en Python, son simplemente *namespaces* que contienen múltiples paquetes y módulos.

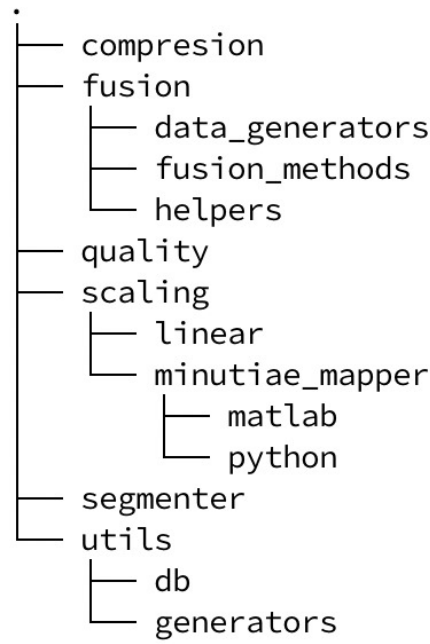


Figura 8.5: Estructura de directorios del código implementado.

Al funcionar como paquetes de Python, cada módulo se puede importar siguiendo exactamente su ubicación y nombre del archivo. Por ejemplo, el módulo *Vendor*, que se encuentra dentro del directorio *utils*, se puede referenciar como *scripts.utils.vendor*.

Al tener que implementar varios *scripts* y manipular los datos teniendo que repetir algunos experimentos varias veces, tener una estructura clara y ordenada fue de gran ayuda.

9. Conclusiones y Trabajo Futuro

La identificación automática de niños mediante sus huellas dactilares es una problemática presente a nivel mundial ya que, tanto el hardware como el software disponibles comercialmente suelen ser diseñados para el uso en adultos. En Uruguay, la DNIC enrola a los niños en su sistema biométrico de huellas dactilares desde su nacimiento, por lo que tratar esta problemática es de especial interés para poder obtener el mayor beneficio de los datos con los que se cuenta.

En el marco del convenio entre la Facultad de Ingeniería de la Universidad de la República y la DNIC se desarrolló el presente proyecto, en el cual se abordó la problemática antes mencionada desde distintas áreas. En particular, se trabajó en calidad, escalado, fusión y compresión de huellas dactilares. Además de realizar estudios de desempeño y proponer mejoras al sistema actual en dichas áreas, se construyeron módulos de software con los cuales se facilita la integración de algunas de las mejoras en el sistema biométrico de la DNIC.

En las siguientes secciones se describen las conclusiones obtenidas así como las oportunidades de trabajo futuro que se han detectado.

9.1. Conclusiones

En general, se logró un aumento en el desempeño del sistema bajo las condiciones de la DNIC. Esto implica que, manteniendo el hardware actual e integrando el software generado en este proyecto, se pueda disminuir la cantidad de casos en los que una identificación y verificación automática de niños falla, lo cual aumenta la confiabilidad en el sistema y agiliza los procesos que se llevan a cabo con el mismo al reducir la necesidad de intervención humana. Esta mejora se debe a los aportes alcanzados en los módulos de *Enhancement* y *Decision* (ver figura 2.1) con las sugerencias realizadas en las áreas de escalado y fusión respectivamente. A su vez, se pudieron reproducir resultados del estado del arte en una base de datos con imágenes de huellas dactilares de niños de la DNIC, consiguiendo también realizar aportes al estado del arte.

En el área de calidad, correspondiente al módulo de *Quality Estimation*, si bien no se propuso un cambio específico al sistema actual, sí se brindó mayor conocimiento sobre los distintos algoritmos disponibles y la relación entre los mismos. En particular, se encontraron funciones concretas para dichas relaciones. Además, se dejó disponible la última versión del algoritmo de medición de calidad de huellas dactilares implementado por el NIST (NFIQ2), mediante una DLL fácilmente integrable al sistema actual. Esto permite que la DNIC pueda tomar una decisión informada sobre qué algoritmo aplicar.

Con respecto a compresión, área que impacta sobre el módulo de *Template Database* y la conexión que éste tiene con el módulo *Matcher*, no se obtuvieron resultados decisivos que impliquen posibles mejoras para el sistema debido a la simulación de datos que se tuvo que hacer. De todas formas, con los datos simulados, se observa que JPEG 2000 puede representar una mejora en el desempeño para el caso de huellas de 5 años, por lo que esto podría ser un cambio interesante para la DNIC.

En el área de escalado, además de implementar un módulo con distintas operaciones para hacer escalado lineal con el factor de interpolación adecuado según la edad a la que se obtuvo la toma, se logró validar que el crecimiento de las huellas dactilares es isotrópico de una forma alternativa a la presentada en el estado del arte. Sumado a esto, se construyó una estimación de la transformación que lleva una huella a otra sin necesidad de marcar las huellas previamente, lo cual es un paso más allá en el trabajo realizado en el artículo de referencia. También, se obtuvo una forma alternativa de estimar el factor de interpolación por rango de edad que, en el caso de las huellas estudiadas, coincide con el valor que se presentó en el estado del arte, validando así una vez más este hallazgo.

Por último, respecto al área de fusión, área que impacta en el módulo de *Decision*, mediante un estudio sobre posibles métodos a utilizar, se encontró que con un sencillo cambio en el algoritmo actual utilizado por la DNIC, ya se logra mejorar el desempeño.

9.2. Trabajo Futuro

Para completar los estudios realizados en el presente trabajo, se identificaron diferentes aspectos en los que se puede continuar trabajando, tanto en las áreas específicas abordadas en el proyecto como en otros aspectos que hacen al sistema actual de la DNIC.

En calidad sería deseable reproducir el estudio realizado utilizando una base de datos de mayor tamaño. Idealmente, para que los problemas de clasificación queden bien definidos, se debería disponer de al menos medio millón de huellas. También, podría ser de interés para la DNIC contar con un mecanismo automático que permita, dado un valor para un algoritmo de calidad, retornar el valor más probable para otro.

Para los estudios realizados en el área de escalado, se podría contar con una base de datos preparada especialmente de forma tal que se puedan hallar los factores de interpolación desconocidos (ver tabla 5.2). Por otro lado, surgió de las reuniones con el Ing. Pablo Sanguinet, que es de interés para la DNIC contar con una operación que dada una imagen, calcule el factor de interpolación en base a la distancia entre las crestas y los valles de la misma. Esto quedó por fuera del alcance del proyecto, pero es un punto que se podría trabajar en el futuro.

En cuanto a fusión, se podría completar el estudio realizado observando la evolución del desempeño del sistema al agregar paulatinamente los datos de más dedos hasta completar, de ser posible, los diez. También sería interesante realizar el estudio con una mayor cantidad de identidades. Por último, queda abierta la posibilidad de utilizar métodos de fusión, particularmente el método a-contrario, para tratar el problema del zoológico de Doddington.

Para completar el estudio de compresión, sería ideal contar tanto con una base de datos de 2000 dpi sin comprimir con huellas de niños, así como una base de datos de las mismas características pero de 500 dpi. Esto permitiría evitar el uso de datos simulados, aumentando la fidelidad de los resultados.

Finalmente, podrían considerarse cambios en otros módulos del sistema. Por ejemplo, se podrían evaluar otros software de *matching* o cambiar los sensores actuales. En particular, para poder llegar a buenos niveles de desempeño para niños menores de 2 años, sería recomendable aumentar la resolución de los escáneres de captura de huellas a 1000 dpi o 2000 dpi. Con esto se podría conseguir tomas de mayor calidad de forma de que al llevarlas a tamaño de adulto, los algoritmos existentes sean capaces de identificar las minucias y así poder realizar el *matching*.

Bibliografía

- [1] .NET. <https://dotnet.microsoft.com/>. [Online; accedido 12-Mayo-2020]. 2020.
- [2] Anil K. Jain y Karthik Nandakumar. *Biometric System Design, Overview*. Springer, 2009. ISBN: 978-0-387-73003-5.
- [3] *ANSI/NIST-ITL Standard*. <https://www.nist.gov/programs-projects/ansinist-itl-standard>. [Online; accedido 1-Junio-2020]. 2019.
- [4] Anil K. Jain Arun A. Ross Karthik Nandakumar. *Handbook of Multibiometrics*. Springer, 2006. ISBN: 978-0-387-33123-2.
- [5] Laurent Beslay, Javier Galbally y Rudolf Haraksim. *Automatic fingerprint recognition: from children to elderly Ageing and age effects*. Inf. téc. Joint Research Centre, jun. de 2018. DOI: 10.2760/809183.
- [6] C#. <https://docs.microsoft.com/en-us/dotnet/csharp/>. [Online; accedido 23-Mayo-2020]. 2020.
- [7] Vanina Camacho, Guillermo Garella Guillermo y Francesco Franzoni. «Identificación de niños utilizando huellas digitales». En: 2017. DOI: <https://hdl.handle.net/20.500.12008/20141>.
- [8] Atul S. Chaudhari, G. K. Patnaik y Sandip Patil. «Implementation of Minutiae Based Fingerprint Identification System Using Crossing Number Concept». En: 2014.
- [9] *Compression Guidance for 1000 ppi Friction Ridge Imagery*. <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.500-289.pdf>. [Online; accedido 1-Junio-2020]. 2014.
- [10] Luis Di Martino y col. «A Statistical Approach to Reliability Estimation for Fingerprint Recognition.» En: *2016 International Conference of the Biometrics Special Interest Group (BIOSIG), Biometrics Special Interest Group (BIOSIG), 2016 International Conference of the* (2016), págs. 1-8. ISSN: 978-3-8857-9654-1.
- [11] Luis Di Martino y col. «An A-Contrario Biometric Fusion Approach». En: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Jun. de 2020.
- [12] *Diapositivas del curso Tratamiento de Imágenes - IIE*. <https://eva.fing.edu.uy/course/view.php?id=520>. 2019.
- [13] *Dirección Nacional de Identificación Civil*. <https://dnic.minterior.gub.uy/index.php>. [Online; accedido 14-Enero-2020]. 2020.
- [14] George R. Doddington y col. «SHEEP, GOATS, LAMBS and WOLVES: a statistical analysis of speaker performance in the NIST 1998 speaker recognition evaluation». En: *ICSLP*. 1998.
- [15] C. Watson E. Tabassi C. Wilson. «Fingerprint Image Quality». En: *NIST Internal Report 7151* (2004).

- [16] M.A. Figueroa-Villanueva, N.K. Ratha y R.M. Bolle. «A Comparative Performance Analysis of JPEG 2000 vs. WSQ for Fingerprint Image Compression». En: *Audio- and Video-Based Biometric Person Authentication* (jun. de 2003), págs. 385-392. DOI: 10.1007/3-540-44887-X_46.
- [17] Javier Galbally y col. *Fingerprint Quality: Mapping NFIQ1 Classes and NFIQ2 Values*. Jun. de 2019.
- [18] Carsten Gottschlich y col. «Modeling the Growth of Fingerprints Improves Matching for Adolescents». En: *IEEE Transactions on Information Forensics and Security* 6 (sep. de 2011), págs. 1165-1169. DOI: 10.1109/TIFS.2011.2143406.
- [19] Rudolf Haraksim, Javier Galbally y Laurent Beslay. «Fingerprint Growth Model for Mitigating the Ageing Effect on Children's Fingerprints Matching». En: *Pattern Recognition* 88 (dic. de 2018). DOI: 10.1016/j.patcog.2018.12.024.
- [20] *Information technology – Biometric data interchange formats – Part 2: Finger minutiae data*. Standard. International Organization for Standardization, 2011.
- [21] Anil Jain y col. «Biometrics for Child Vaccination and Welfare: Persistence of Fingerprint Recognition for Infants and Toddlers». En: (abr. de 2015).
- [22] Anil Jain y col. «Fingerprint Recognition of Young Children». En: *IEEE Transactions on Information Forensics and Security* PP (dic. de 2016). DOI: 10.1109/TIFS.2016.2639346.
- [23] *Mathworks*. <https://www.mathworks.com/>. [Online; accedido 23-Mayo-2020]. 2020.
- [24] *Matlab*. <https://www.mathworks.com/products/matlab.html>. [Online; accedido 12-Mayo-2020]. 2020.
- [25] *Microsoft*. <https://www.microsoft.com/>. [Online; accedido 23-Mayo-2020]. 2020.
- [26] *Minutae Cylinder-Code SDK - University of Bologna*. <https://biolab.csr.unibo.it/research.asp>. [Online; accedido 14-Mayo-2020]. 2020.
- [27] K. Nandakumar y col. «Likelihood Ratio-Based Biometric Score Fusion.» En: *IEEE Transactions on Pattern Analysis and Machine Intelligence, Pattern Analysis and Machine Intelligence, IEEE Transactions on, IEEE Trans. Pattern Anal. Mach. Intell* 30.2 (2008), págs. 342-347. ISSN: 0162-8828.
- [28] *NIST Biometric Image Software (NBIS)*. <https://www.nist.gov/services-resources/software/nist-biometric-image-software-nbis>. [Online; accedido 7-Agosto-2020].
- [29] *NIST Special Database 301 - Nail to Nail Fingerprint Challenge Dry Run*. <https://nvlpubs.nist.gov/nistpubs/TechnicalNotes/NIST.TN.2002.pdf>. [Online; accedido 9-Julio-2020]. 2018.
- [30] *Numpy*. <https://numpy.org/>. [Online; accedido 10-Mayo-2020]. 2020.

- [31] *OpenJPEG*. <https://github.com/uclouvain/openjpeg>. [Online; accedido 15-Julio-2020]. 2020.
- [32] *Pandas*. <https://pandas.pydata.org/>. [Online; accedido 9-Mayo-2020]. 2020.
- [33] J. Preciozzi y col. «Fingerprint Biometrics From Newborn to Adult: A Study From a National Identity Database System». En: *IEEE Transactions on Biometrics, Behavior, and Identity Science* 2.1 (2020). DOI: 10.1109/TBIOM.2019.2962188.
- [34] M. Ferrara R. Cappelli y D. Maltoni. «A Two-Factor Protection Scheme for MCC Fingerprint Templates». En: *International Conference of the Biometrics Special Interest Group (BIOSIG), Darmstadt, Germany* (2014).
- [35] M. Ferrara R. Cappelli y D. Maltoni. «Fingerprint Indexing based on Minutia Cylinder Code». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.5 (2011), págs. 1051-1057.
- [36] M. Ferrara R. Cappelli y D. Maltoni. «Minutia Cylinder-Code: a new representation and matching technique for fingerprint recognition». En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.12 (2010), págs. 2128-2141.
- [37] M. Ferrara R. Cappelli y D. Maltoni. «Noninvertible Minutia Cylinder-Code Representation». En: *IEEE Transactions on Information Forensics and Security* 7.6 (2012), págs. 1727-1737.
- [38] Organización Mundial de la Salud. *Patrones de crecimiento infantil*. <https://www.who.int/childgrowth/standards/es/>. [Online; accedido 23-Enero-2020].
- [39] *Seaborn*. <https://seaborn.pydata.org/>. [Online; accedido 9-Mayo-2020]. 2020.
- [40] *Seaborn*. <https://matplotlib.org/>. [Online; accedido 9-Mayo-2020]. 2020.
- [41] *Seaborn*. <https://opencv.org/>. [Online; accedido 9-Mayo-2020]. 2020.
- [42] Uruguay. Ley N°17243 Servicios públicos y privados, seguridad pública y condiciones en las que se desarrollan las actividades productivas (2000) Disponible en <https://www.impo.com.uy/bases/leyes/17243-2000>.
- [43] *Universidad de Bologna*. <https://www.unibo.it/en>. [Online; accedido 14-Abril-2020].
- [44] *WSQ* *Bibliography.*
<https://www.nist.gov/itl/iad/image-group/wsq-bibliography>. [Online; accedido 1-Junio-2020]. 2017.

Anexos

A. Riesgos

Al comienzo del proyecto se identificaron riesgos con el objetivo de controlarlos y poder contar con posibles formas de mitigarlos.

A cada riesgo se le asignó una probabilidad de ocurrencia subjetiva de acuerdo al conocimiento sobre el mismo, pudiendo ser ésta baja, media o alta. Se le asignó también el nivel de impacto que éste podría tener si ocurriera, pudiendo ser también bajo, medio o alto. Por último, se estableció la forma de monitorearlo y un plan de contingencia para el caso de que el riesgo suceda.

1. No tener el escáner de 2000 dpi en tiempo y forma.

Probabilidad de ocurrencia	Media.
Impacto	Alto.
Monitoreo	Enviar mails de consulta al contacto en la empresa proveedora del escáner.
Plan de contingencia	Cambiar algunos objetivos del proyecto de forma que el escáner no sea necesario, sin perder el objetivo principal y el beneficio e interés de la DNIC.

2. Perder contacto con Pablo o Gastón en la DNIC.

Probabilidad de ocurrencia	Baja.
Impacto	Alto.
Monitoreo	Enviar avances del proyecto y tener presentaciones del estado del mismo obteniendo retroalimentación de ellos.
Plan de contingencia	-

3. Tener una única licencia del software propietario de *matching* utilizado por la DNIC para trabajar.

Probabilidad de ocurrencia	Media.
Impacto	Medio.

Monitoreo	Enviar mail al contacto en la empresa del software propietario de <i>matching</i> utilizado por la DNIC para saber si es posible o no conseguir más licencias.
Plan de contingencia	El equipo se organizaría en turnos según prioridades para poder utilizar la única computadora que tiene dicho software funcionando.

4. Perder la única licencia del software propietario de *matching* utilizado por la DNIC que se tiene actualmente.

Probabilidad de ocurrencia	Bajo.
Impacto	Alto.
Monitoreo	-
Plan de contingencia	Enviar mail nuevamente al contacto en la empresa del software propietario de <i>matching</i> utilizado por la DNIC explicando la situación, y preguntar si hay posibilidad de obtener otra licencia del software. Este proceso podría llevar tiempo o incluso podría no conseguirse una nueva licencia; en ese caso se hablaría con la DNIC para evaluar la posibilidad de utilizar una computadora de su oficina con el software ya instalado.

5. Riesgos técnicos en general.

Probabilidad de ocurrencia	Media.
Impacto	Medio.
Monitoreo	Se estableció un período de tiempo estimado para evaluar la factibilidad técnica previo a cada actividad en la que se haya identificado algún riesgo de este tipo.
Plan de contingencia	Al haber identificado los riesgos y hacerse un análisis de la factibilidad técnica de ciertas actividades, ya se estableció la posibilidad de que el entregable de dichas actividades sea el análisis escrito con las conclusiones a las que se llegó.

B. Actividades de Vinculación

En cuanto a procesos, se realizaron reuniones con los tutores cada una o dos semanas (siempre que fuera posible y necesario), en las cuales se discutía el avance logrado, con breves presentaciones de ser de ayuda y se planificaba el avance para las siguientes semanas.

Para el relevamiento de requerimientos se tuvo una serie de reuniones en la Dirección Nacional de Identificación Civil. A continuación se detallan brevemente las mismas, incluyendo los participantes y objetivos.

La primera reunión fue realizada el 11 de octubre del 2019 y sus principales objetivos fueron interiorizar a la DNIC sobre el proyecto y comenzar a establecer pautas de trabajo entre las partes y los principales lineamientos. A la misma asistieron por parte de la DNIC el Dir. Gastón Arbenoiz y el Ing. Pablo Sanguinet, y por parte de la Facultad de Ingeniería asistieron los tutores Ing. Alicia Fernández, Ing. Javier Preciozzi y los estudiantes.

La siguiente reunión, el 21 de octubre, consistió de dos partes. En la primera participaron, por parte de la DNIC, la directora de dactiloscopia de la institución y el Ing. Pablo Sanguinet. En esta instancia se relevó información sobre el contexto de la DNIC en lo que respecta al uso de las huellas dactilares, haciendo foco especialmente en el tratamiento que se le hace a las huellas de niños y los problemas actuales con las mismas. En la segunda parte el equipo fue, junto al Ing. Pablo Sanguinet, al Hospital Pereira Rosell, donde la DNIC tiene puestos de filiación de recién nacidos. En esta visita se pudo observar el proceso de toma de huellas dactilares con recién nacidos, así como el proceso completo de filiación. También uno de los técnicos especializados en la toma de huellas hizo una demostración de la interfaz de usuario del sistema actual para la captura de huellas, de forma que el equipo pudiera familiarizarse con ella.

Se considera que la información obtenida en esta reunión fue realmente valiosa para ponerse en contexto real y entender, no solo los problemas técnicos que las huellas dactilares de niños tienen por sus características físicas, sino también por la forma y las condiciones en las que tienen que ser obtenidas.

En una tercera reunión, el 30 de octubre, se obtuvo información sobre cómo realiza la DNIC la verificación e identificación de huellas, tanto para las obtenidas en las oficinas como en los equipos móviles (de forma remota). Se concluyó que sería de interés estudiar un mejor criterio para la verificación/identificación, unificando también el criterio para la autenticación en ambos métodos de obtención de huellas.

Posteriormente, en una reunión realizada el 20 de noviembre se relevó información acerca de las diferentes tecnologías a utilizar, de forma tal que el producto final sea de utilidad para la DNIC. También se plantearon diferentes

preguntas acerca de sus procesos, tales como ¿cómo conectan los escáneres de obtención de huellas?, ¿en qué momento del flujo las imágenes son guardadas?, entre otras.

Luego, en la última reunión presencial del año 2019, el 15 de diciembre, se redefinieron algunos objetivos del proyecto debido a no poder contar con el escáner de 2000 dpi que originalmente formaría parte de las herramientas para este proyecto.

El 18 de diciembre se tuvo una videollamada con el Ing. Pablo Sanguinet para profundizar en los objetivos redefinidos en la reunión anterior. Se discutió sobre la factibilidad de conseguir huellas de menores de 5 años sin comprimir para poder hacer estudios sobre el impacto de distintos métodos de compresión en esta población, concluyendo que no sería posible obtener suficientes huellas como para realizar un estudio estadísticamente relevante pero que tal vez se podrían obtener algunas muestras con las que hacer una prueba de concepto.

También se habló sobre el interés de la DNIC de tener algún tipo de estudio que les ayudara a entender la relación entre las distintas medidas de calidad con las que cuentan, y sobre los entregables de software que se podrían generar. Con respecto a esto último, se acordó que los mismos fueran en forma de DLL para Windows que se pudiesen integrar con su sistema actual.

Durante el resto de diciembre, y en los siguientes meses de enero y febrero de 2020 no se tuvieron reuniones con la DNIC debido a los distintos períodos de licencia, pero se mantuvo comunicación mediante e-mail para mantener informada a la contraparte sobre el trabajo que se estuvo llevando a cabo.

El 6 de marzo se realizó una presentación en las oficinas de la DNIC, con el objetivo de mostrar los avances realizados y los resultados obtenidos hasta el momento. Además, esta presentación hizo posible que más integrantes del equipo de desarrollo de la DNIC conocieran el proyecto y se pudieran escuchar y resolver sus comentarios, preguntas o sugerencias. Por esto también se explicaron los objetivos del proyecto y los análisis a realizar en cada área de trabajo.

De parte de los autores del presente informe se evaluó dicha presentación como muy positiva, principalmente por haber permitido una muy buena interacción con los integrantes del equipo de desarrollo presentes que pudo ser tomada en cuenta luego.

Esta fue la última reunión presencial en la DNIC debido al contexto de pandemia que se inició a mediados de marzo del 2020. Sin embargo, se mantuvo comunicación vía e-mail, videollamada y Whatsapp siempre que fuera necesario.

C. Documentación de la DLL

Escalado Lineal

C.1. Requerimientos

- .NET Framework 4.6.1.
- OpenCVSharp 4.3.0.

C.2. Operaciones Provistas

Esta librería, cuyo nombre es “Escalado”, provee una única clase “Lineal” la cual cuenta con ciertas operaciones para permitir al usuario escalar una imagen dada, ya sea desde un archivo, o desde una estructura adecuada, y retorna una nueva imagen escalada, ya sea en una estructura adecuada o guardándola en disco.

El factor de interpolación a utilizar está dado por la edad del individuo en el momento en que se tomó la imagen a escalar o puede ser pasado como parámetro en una de las operaciones, con el propósito de dar más flexibilidad.

C.2.1. EscalarAByteArray

Este método permite, dada una imagen, escalarla utilizando el factor correspondiente a la edad del individuo cuando se tomó la huella y retornarla como *byte array*.

Firma

```
public byte[] EscalarAByteArray(  
    string srcPath ,  
    DateTime bornDate ,  
    DateTime shotDate  
)
```

Parámetros

- **srcPath**: path a la imagen a escalar (debe ser un archivo con extensión bmp).
- **bornDate**: fecha de nacimiento del individuo al que corresponde la huella.
- **shotDate**: fecha en que fue tomada la toma.

Ejemplo de uso

```
byte[] img = EscalarAByteArray(  
    "ejemplo.bmp",  
    DateTime.Now,  
    DateTime.Now.AddYears(5)  
);
```

C.2.2. EscalarABitmap

Este método permite, dada una imagen, escalarla utilizando el factor correspondiente a la edad del individuo cuando se tomó la huella y retornarla como *Bitmap*.

Firma

```
public Bitmap EscalarABitmap(  
    string srcPath,  
    DateTime bornDate,  
    DateTime shotDate  
)
```

Parámetros

- **srcPath**: path a la imagen a escalar (debe tener extensión bmp).
- **bornDate**: fecha de nacimiento del individuo al que corresponde la huella.
- **shotDate**: fecha en que fue tomada la toma.

Ejemplo de uso

```
Bitmap img = EscalarABitmap(  
    "ejemplo.bmp",  
    DateTime.Now,  
    DateTime.Now.AddYears(5)  
);
```

C.2.3. EscalarGuardarByteArray

Este método permite, dada una imagen, escalarla utilizando el factor correspondiente a la edad del individuo cuando se tomó la huella y guardar la nueva imagen escalada como *byte array*.

Firma

```
public void EscalarGuardarByteArray(  
    string srcPath ,  
    string dstPath ,  
    DateTime bornDate ,  
    DateTime shotDate  
)
```

Parámetros

- **srcPath**: path a la imagen a escalar (debe tener extensión bmp).
- **srcPath**: path donde guardar la imagen escalada.
- **bornDate**: fecha de nacimiento del individuo al que corresponde la huella.
- **shotDate**: fecha en que fue tomada la toma.

Ejemplo de uso

```
EscalarGuardarBitmap(  
    "ejemplo.bmp" ,  
    "ejemplo_escalado.txt" ,  
    DateTime.Now ,  
    DateTime.Now.AddYears(5)  
);
```

C.2.4. EscalarGuardarBitmap

Este método permite, dada una imagen, escalarla utilizando el factor correspondiente a la edad del individuo cuando se tomó la huella y guardar la nueva imagen escalada como *bitmap*.

Firma

```
public void EscalarGuardarBitmap(  
    string srcPath ,  
    string dstPath ,  
    DateTime bornDate ,  
    DateTime shotDate  
)
```

Parámetros

- **srcPath**: path a la imagen a escalar (debe tener extensión bmp).

- **srcPath**: path donde guardar la imagen escalada (debe tener extensión bmp).
- **bornDate**: fecha de nacimiento del individuo al que corresponde la huella.
- **shotDate**: fecha en que fue tomada la toma.

Ejemplo de uso

```
EscalarGuardarBitmap(
    "ejemplo.bmp",
    "ejemplo_escalado.bmp",
    DateTime.Now,
    DateTime.Now.AddYears(5)
);
```

C.2.5. EscalarAImage

Este método permite, dada una imagen, escalarla utilizando el factor correspondiente a la edad del individuo cuando se tomó la huella y retornarla como *Image*.

Firma

```
public Image EscalarAImage(
    Image img,
    DateTime bornDate,
    DateTime shotDate
)
```

Parámetros

- **img**: imagen a escalar.
- **bornDate**: fecha de nacimiento del individuo al que corresponde la huella.
- **shotDate**: fecha en que fue tomada la toma.

Ejemplo de uso

```
Image img = ...;
Image imgEscalada = EscalarAImage(
    img
    DateTime.Now,
    DateTime.Now.AddYears(5)
);
```

C.2.6. EscalarAImageConFactor

Este método permite escalar una imagen dada con un factor dado y la retorna como *Image*.

Firma

```
public Image EscalarAImageConFactor(  
    Image img,  
    double interpolationFactor  
)
```

Parámetros

- **img**: imagen a escalar.
- **interpolationFactor**: factor a ser utilizado para escalar la imagen.

Ejemplo de uso

```
Image img = ...;  
Image imgEscalada = EscalarAImageConFactor(img, 2);
```

Nota:

En el caso de que al ejecutar alguna de las operaciones provistas se lance la excepción que indica que falta la DLL “OpenCVSharpExtern”, se debe agregar como un elemento a la solución dicha DLL (la cual se provee junto con esta DLL).

D. Documentación de la DLL de Calidad

D.1. Requerimientos

- .NET Framework 4.6.1.
- Docker.
- Imagen de Docker con NFIQ2 instalado (se provee como parte de esta DLL el *Dockerfile* necesario para generar la misma).

D.2. Operaciones Provistas

Esta librería, cuyo nombre es “Calidad”, provee una única clase “NFIQ2” la cual cuenta con ciertas operaciones para permitir al usuario hallar la calidad de imágenes de huellas dactilares utilizando el algoritmo NFIQ2.

D.2.1. ModoSimple

Este método permite calcular la calidad con NFIQ2 de la imagen dada, retornando un entero entre 0 y 100, o 255 en caso que el algoritmo falle.

Firma

```
public int ModoSimple(string pathToImage)
```

Parámetros

- **pathToImage**: path a la imagen a la cual se le calculará la calidad (debe ser un archivo con extensión BMP). Este path puede ser tanto relativo como absoluto.

Ejemplo de uso

```
int quality = ModoSimple("./test.bmp");
```

D.2.2. ModoBatch

Este método permite calcular la calidad con NFIQ2 de una lista de imágenes dadas, retornando una lista de enteros entre 0 y 100, 255 en el caso que el algoritmo falle o -1 si la imagen no es válida.

Firma

```
public int [] ModoBatch(string [] pathToImages)
```

Parámetros

- **pathToImages**: lista con los paths a la imágenes a las cuales se les calculará la calidad (deben ser archivos con extensión BMP). De igual forma que en el método anterior, estos paths pueden ser tanto relativos como absolutos.

Ejemplo de uso

```
string [] images = new string [2];  
images [0] = "./test.bmp";  
images [1] = "./test_2.bmp";  
int [] qualities = ModoBatch(images);
```

Nota:

Por cuestiones de seguridad se optó por generar una nueva carpeta con la imagen (o imágenes) a la cual se le calculará la calidad. Dicha copia a una nueva carpeta es realizada automáticamente y, luego de la ejecución del método, es eliminada. Esta decisión se tomó ya que se necesita dar acceso a la carpeta que contiene la imagen de la huella a la imagen de Docker.

E. Herramientas Utilizadas

Para el desarrollo de este proyecto se utilizaron varias herramientas, tanto para el proceso como para la implementación propiamente dicha. En esta sección se presentan estas herramientas, así como también el análisis de por qué fueron necesarias o útiles.

E.1. Python

Es un lenguaje interpretado, es decir que no necesita compilación previo a ejecutarlo, dinámicamente tipado, de alto nivel y de propósito general¹

Otra de las características de Python es que es multiplataforma, lo que permitió que los distintos miembros del equipo pudieran implementar y ejecutar en su ambiente de desarrollo usual y, a su vez, poder ejecutar los distintos *scripts* que necesitaban acceso al software propietario de *matching* utilizado por la DNIC en la máquina específica con sistema operativo Windows que tenía dicho software instalado.

Estas características, sumadas al hecho de que es un lenguaje muy conocido y utilizado en las áreas de procesamiento de imágenes y *data science* por la cantidad de librerías para procesamiento de imágenes, análisis de datos, gráficos, etc. disponibles hicieron que Python sea el lenguaje elegido para la implementación de todas las pruebas que lo permitían y para el procesamiento y análisis de sus resultados.

E.2. Pandas

Pandas [32] es una librería *open source* (de código abierto, accesible al público) para análisis y manipulación de datos disponible en Python.

Su desarrollo comenzó en el 2008, y en 2009 fue liberado a la comunidad, lo cual es acorde con la visión del proyecto de que el software sea: accesible para todos, gratis para que los usuarios puedan usarlo y modificarlo, flexible, poderoso, fácil de usar y rápido.

Pandas ofrece estructuras particulares y operaciones para manipular tablas de datos (como archivos CSV por ejemplo). Particularmente, crea objetos de tipo *DataFrame* que permiten manipular gran cantidad de datos de forma rápida y eficiente, dando la posibilidad de generar y manejar subconjuntos de datos de forma fácil.

¹Diseñado para poder ser utilizado en una gran variedad de dominios.

Además, brinda la posibilidad de leer y escribir datos en disco en distintos formatos. Particularmente, en este proyecto se utilizó mucho el formato CSV, mencionado anteriormente.

La funcionalidad *group by* es una gran fortaleza de esta herramienta, permitiendo así dividir y combinar conjuntos de datos, así como también aplicar funciones a subconjuntos de ellos.

E.3. Seaborn

Relacionado también con el análisis de datos, Seaborn [39] es una librería de visualización para Python basada en Matplotlib [40], que permite crear gráficas estadísticas.

Las funciones de gráficas de Seaborn pueden utilizar objetos creados con Pandas y, para agrupar datos, utilizan internamente las operaciones de dicha herramienta.

E.4. Numpy

Numpy [30] es un paquete de computación científica disponible en Python. Entre sus funcionalidades, Numpy brinda un objeto de tipo *array* (arreglo) multidimensional y una gran variedad de funciones algebraicas y numéricas que pueden utilizar dichos arreglos.

La idea de Numpy es que sus arreglos sean hasta 50 veces más rápidos que las listas tradicionales de Python.

Si bien en un principio se pensó utilizar esta herramienta para analizar datos, manejar las estructuras de información que se necesitaban era más lento de lo deseado. Es por esto que se decidió usar Pandas, cuyo objetivo de por sí es el análisis de datos. Se sigue usando Numpy pero específicamente para operaciones matemáticas particulares.

E.5. OpenCV

OpenCV [41] es una librería *open source* para *computer vision*². Una de las características principales de esta herramienta es que es multiplataforma y está disponible para usarse con distintos lenguajes de programación: Python, C++, Java y Matlab. También existen *wrappers* en otros lenguajes como C#, Perl, Haskell y Ruby. Sin dudas, esta fue una de las razones por las que se decidió usar ya que,

²Disciplina científica interdisciplinaria que incluye métodos para adquirir, procesar, analizar y comprender las imágenes del mundo real con el fin de producir información numérica o simbólica para que puedan ser tratada por una computadora.

como se mencionó anteriormente, los *scripts* necesarios para las distintas pruebas fueron realizados con Python, pero los entregables para la DNIC con C# en .NET.

OpenCV dispone de más de 2.500 algoritmos, entre ellos una gran variedad de *computer vision* y *machine learning*. Es por esto que tiene una gran variedad de usos posibles, e incluso grandes compañías como son Google, Microsoft, Intel, IBM, Toyota, Honda, entre otras, la utilizan [41].

La primera versión (en alpha) de OpenCV fue liberada al público en la conferencia anual sobre *computer vision* y reconocimiento de patrones del IEEE (*Institute of Electrical and Electronics Engineers*) en el año 2000. Luego, del 2001 al 2005 se liberaron otras versiones todas en beta hasta que finalmente en el año 2006, se liberó la primer versión 1.0. Esto hace que, al año actual, OpenCV tenga ya 20 años de existencia.

E.6. .NET

.NET [1] es una plataforma de desarrollo creada por Microsoft. Es un conjunto de lenguajes y librerías que se pueden utilizar para desarrollar aplicaciones.

Los lenguajes que se pueden utilizar son C#, Visual Basic y F#, y algunas de las plataformas o implementaciones disponibles son .NET Core (que puede ser ejecutado tanto en Windows como Linux o macOS), .NET Framework (solo para Windows), Xamarin/Mono y .NET Standard. En el proyecto, se utilizó .NET Framework, por ser el que se utiliza en la DNIC.

La idea de .NET es que sea independiente del hardware. Es por esto que el software desarrollado con .NET se ejecuta en una máquina virtual denominada *Common Language Runtime* (CLR) que da ciertos servicios como manejo de memoria, de excepciones y seguridad.

El código escrito en cualquiera de los lenguajes que .NET soporta es compilado a un tipo de código intermedio llamado *CIL* (*Common Intermediate Language*); y luego es otro compilador en el CLR el que genera el código de máquina.

E.7. C#

C# [6] es un lenguaje de programación orientado a objetos desarrollado por Microsoft [25] en el año 2000 como parte de la iniciativa de .NET.

Dentro de algunas de las características de C#, se encuentran que es un lenguaje de propósito general, multiparadigma y, a diferencia de Python, es compilado y fuertemente tipado.

Otra característica particular es que si bien en un inicio solamente se podía utilizar en ambientes con sistema operativo Windows, actualmente .NET Core permite la portabilidad entre aplicaciones en distintos sistemas operativos.

C# y .NET son las herramientas utilizadas en la DNIC para implementar las distintas DLL que utilizan en sus sistemas. Es por esto que se utilizó también en el contexto de este proyecto, siendo uno de los requerimientos no funcionales acordados.

E.8. Matlab

Matlab [24] es un lenguaje y entorno de desarrollo para computación científica propietario creado y mantenido por MathWorks [23] desde el año 1983 y disponible para los sistemas operativos Windows, MacOS y GNU/Linux.

Dentro de las amplias funcionalidades de Matlab, se destacan el manejo de matrices y gráfico de datos y funciones. Es un lenguaje y entorno muy utilizado sobre todo en centros de investigación y desarrollo y en el ámbito académico.

El lenguaje en sí es interpretado, por lo que no necesita de una compilación previa. Con Matlab se puede, además, invocar a funciones o rutinas implementadas en C o Fortran, así como también a librerías escritas en lenguajes como Perl, Java o .NET.

Particularmente, la característica de poder usar fácilmente una DLL de .NET, es la que se utilizó para poder importar y llamar a funciones del software MCC SDK, en cuya documentación se usaba Matlab como ejemplo.

E.9. MCC SDK

El software Minutia Cylinder-Code SDK (MCC SDK) [26] es una librería DLL de .NET realizada por el Laboratorio de Sistema Biométrico de la Universidad de Bologna (Italia) que permite realizar verificación de huellas utilizando los algoritmos *Minutia Cylinder Code* [36][35][37][34] sin optimizar.

Si bien el SDK brinda varias funcionalidades, particularmente se utilizaron los métodos *SetMccEnrollParameters* y *SetMccMatchParameters* para establecer algunos parámetros necesarios; *CreateMccTemplateFromIsoTemplate* para crear un *MCC template* en base a otro *template* con minucias en formato ISO/IEC 19794-2 [20]; *MatchMccTemplates* que compara dos *MCC templates* y retorna un *score* en el rango [0; 1] para establecer el grado de similaridad entre ambos; y *SaveMccTemplateToTextFile* para guardar un *MCC Template* en formato .txt en un lugar determinado.

E.10. OpenJPEG

OpenJPEG [31] es un códec³ de JPEG 2000 de código libre escrito en C. Ha sido desarrollado para promover el uso de JPEG 2000 y desde abril del 2015 es oficialmente reconocido por *ISO/IEC* (*International Organization for Standardization e International Electrotechnical Commission* respectivamente) y *ITU-T* (*International Telecommunication Union*) como software de referencia para JPEG 2000.

OpenJPEG está publicado bajo la licencia *BSD 2-clause Simplified License* lo que permite a cualquier persona usar y modificar el código, aún para usos comerciales. La librería es mantenida por el *Image and Signal Processing Group* (*ISPGGroup*).

En el ámbito de este proyecto se utilizó su versión 2.3.1 para Windows, usando tanto su codificador como su decodificador.

³Programa o dispositivo hardware capaz de codificar o decodificar una señal o flujo de datos digitales.

F. Descomposición de una Matriz Afín

Dada una matriz afín M , ésta se puede descomponer en tres componentes: componente de rotación R , componente de escalado S y componente de traslación T . Dado esto, se puede obtener un sistema de ecuaciones determinado para hallar dichas componentes a partir de la matriz M .

A continuación se muestran los cálculos para obtener dicho sistema de ecuaciones:

$$\begin{aligned}
 M &= R \cdot S \cdot T = \\
 &\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} = \\
 &\begin{pmatrix} s_x \cdot \cos(\theta) & -s_y \cdot \sin(\theta) & 0 \\ s_x \cdot \sin(\theta) & s_y \cdot \cos(\theta) & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} = \\
 &\begin{pmatrix} s_x \cdot \cos(\theta) & -s_y \cdot \sin(\theta) & t_x \cdot s_x \cdot \cos(\theta) - t_y \cdot s_y \cdot \sin(\theta) \\ s_x \cdot \sin(\theta) & s_y \cdot \cos(\theta) & t_x \cdot s_x \cdot \sin(\theta) + t_y \cdot s_y \cdot \cos(\theta) \\ 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

Por consiguiente, se obtiene el siguiente sistema de ecuaciones:

$$\begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s_x \cdot \cos(\theta) & -s_y \cdot \sin(\theta) & t_x \cdot s_x \cdot \cos(\theta) - t_y \cdot s_y \cdot \sin(\theta) \\ s_x \cdot \sin(\theta) & s_y \cdot \cos(\theta) & t_x \cdot s_x \cdot \sin(\theta) + t_y \cdot s_y \cdot \cos(\theta) \\ 0 & 0 & 1 \end{pmatrix}$$

$$\text{donde: } \begin{cases} s_x : & \text{Factor de escalado en el eje } x \\ s_y : & \text{Factor de escalado en el eje } y \\ \theta : & \text{Ángulo de rotación} \\ t_x : & \text{Factor de traslación en el eje } x \\ t_y : & \text{Factor de traslación en el eje } y \end{cases}$$

