



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO**

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

**Evaluación de algoritmos para la detección de huellas
dactilares alteradas**

**PARA OPTAR EL TÍTULO PROFESIONAL
DE INGENIERO DE SISTEMAS**

Autor

**Bach. Diaz Carrasco, Natividad Alejandro
ORCID: <https://orcid.org/0000-0002-5775-0203>**

Asesor

**Mg. Celis Bravo Percy Javier
ORCID: <https://orcid.org/0000-0002-8390-0295>**

**Línea de Investigación
Infraestructura, Tecnología y Medio Ambiente**

**Pimentel – Perú
2023**

**EVALUACIÓN DE ALGORITMOS PARA LA DETECCIÓN DE HUELLAS
DACTILARES ALTERADAS**

Aprobación del jurado

**Mg. BRAVO RUIZ JAIME ARTURO.
Presidente del Jurado de Tesis**

**Mg. MEJIA CABRERA HEBER IVAN.
Secretario del Jurado de Tesis**

**Mg. BANCES SAAVEDRA DAVID ENRIQUE.
Vocal del Jurado de Tesis**



Universidad
Señor de Sipán

DECLARACIÓN JURADA DE ORIGINALIDAD

Quien suscribe la **DECLARACIÓN JURADA**, soy del Programa de Estudios de INGENIERÍA DE SISTEMAS de la Universidad Señor de Sipán S.A.C, declaro bajo juramento que soy autor del trabajo titulado:

EVALUACIÓN DE ALGORITMOS PARA LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS.

El texto de mi trabajo de investigación responde y respeta lo indicado en el Código de Ética del Comité Institucional de Ética en Investigación de la Universidad Señor de Sipán (CIEI USS) conforme a los principios y lineamientos detallados en dicho documento, en relación a las citas y referencias bibliográficas, respetando al derecho de propiedad intelectual, por lo cual informo que la investigación cumple con ser inédito, original y autentico.

En virtud de lo antes mencionado, firman:

DIAZ CARRASCO, NATIVIDAD ALEJANDRO	DNI: 40807498	
---------------------------------------	---------------	---

Pimentel, 3 de Junio de 2023.

Dedicatoria

Esta investigación lo dedico a mi familia, quienes me apoyaron en todo momento y fueron la fuente de energía que necesitaba para culminar el presente.

Agradecimientos

Agradezco principalmente a Dios, quien ha hecho posible que yo pueda terminar con esta profesión, proveyéndome en todo momento de salud, paciencia y sabiduría necesaria, para poder afrontar los retos que he tenido durante la vida.

Agradezco a los docentes de la Universidad Señor de Sipán, quienes con su conocimiento y experiencia me guiaron para poder culminar con la redacción de la presente Tesis.

Índice

Dedicatoria	iv
Agradecimientos	v
Índice de tablas, figuras y fórmulas (de ser necesario)	vii
Resumen	xi
Abstract	xii
I. INTRODUCCIÓN	13
1.1. Realidad problemática.	13
1.2. Formulación del problema.....	24
1.3. Hipótesis.....	24
1.4. Objetivos.....	24
1.5. Teorías relacionadas al tema.....	25
II. MATERIALES Y MÉTODO	43
2.1. Tipo y Diseño de Investigación	43
2.2. Variables, Operacionalización.....	43
2.3. Población de estudio, muestra, muestreo y criterios de selección.....	45
2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad..	46
2.5. Procedimiento de análisis de datos.....	47
2.6. Criterios éticos	50
III. RESULTADOS Y DISCUSIÓN.....	51
3.1. Resultados.....	51
3.2. Discusión	57
3.3. Aporte de la investigación	59
IV. CONCLUSIONES Y RECOMENDACIONES	112
4.1. Conclusiones	112
4.2. Recomendaciones	113
REFERENCIAS	114
ANEXOS	119

Índice de tablas, figuras y fórmulas (de ser necesario)

Índice de tablas:

Tabla I. Operacionalización de la variable Independiente y Dependiente.....	44
Tabla II. Técnicas e instrumentos de recolección de datos, validez y confiabilidad	46
Tabla III. Criterios Éticos	50
Tabla IV. Resultados-Base de datos utilizadas en la investigación	51
Tabla V. Cuadro comparativo de algoritmos de clasificación automática para la detección de huellas dactilares alteradas del resultado de investigaciones.....	66
Tabla VI. Cuadro comparativo del array unidimensional, de los algoritmos de extracción de minucias y del campo de orientación.....	105
Tabla VII. Resumen de las matrices de confusión, de los algoritmos EM y CO.....	109
Tabla VIII. Resultados de Exactitud, Exhaustividad y Precisión de las pruebas realizadas a los algoritmos EM y CO.....	110
Tabla IX Resultados de APCER y NPCER de las pruebas realizadas a los algoritmos EM y CO para la detección de huellas dactilares alteras	110
Tabla X. Resultados del tiempo de procesamiento promedio de los algoritmos EM y CO	111

Índice de Figuras:

Fig. 1. Esquema de la Realidad Problemática.....	24
Fig. 2. Ejemplos de rasgos biométricos que se pueden utilizar para la autenticación de un individuo	26
Fig. 3. Imágenes de Huellas Dactilares.....	28
Fig. 4. Patrones de Huellas Dactilares.....	29
Fig. 5. Minucias de Huellas Dactilares.....	30
Fig. 6. Adquisición de Huellas Dactilares	31
Fig. 7. Ejemplo de mejora de huella dactilar.....	32
Fig. 8. Obliteración.....	35
Fig. 9. Distorsión.....	36
Fig. 10. Imitación.....	37

Fig. 11. Comparación del nivel de predicción de la Exactitud para los algoritmos en los diferentes tipos de alteración.....	52
Fig. 12. Comparación del nivel de predicción de la Exhaustividad para los algoritmos en los diferentes tipos de alteración.....	53
Fig. 13. Comparación del nivel de predicción de la Precisión para los algoritmos en los diferentes tipos de alteración.....	54
Fig. 14. Comparación del tiempo de procesamiento en segundos, de los algoritmos en los diferentes tipos de alteración.....	55
Fig. 15. Comparación del nivel de predicción de error de clasificación APCER, para los algoritmos en los diferentes tipos de alteración	56
Fig. 16. Comparación del nivel de predicción de la detección de error de clasificación NPCER, para los algoritmos en los diferentes tipos de alteración	57
Fig. 17. Método de desarrollo de objetivos específicos	59
Fig. 18. Imagen del Lector óptico “U.are.U 4500”, marca de DigitalPersona, utilizado para el escaneo de huellas dactilares	60
Fig. 19. Orden para la toma de huellas dactilares	61
Fig. 20. Redimensionamiento de huella dactilar	61
Fig. 21. Estructura de la Base de datos construida	65
Fig. 22. Secuencia de Implementación de algoritmos de clasificación automática	68
Fig. 23. Imágenes de huellas dactilares	69
Fig. 24. Representación de una matriz de huella dactilar en un gráfico 3D	70
Fig. 25. Resultados de la Implementación del algoritmo Escala de Gris.....	71
Fig. 26. Representación gráfica de la diferencia de la intensidad de una huella dactilar en escala de grises con una normalizada.....	72
Fig. 27. Resultados de la Implementación del algoritmo para la normalización de Huellas Dactilares	74
Fig. 28. Resultados de la Implementación del algoritmo para el Histograma de ecualización de Huellas Dactilares	77
Fig. 29. Resultados de la Implementación del algoritmo para el análisis del campo de orientación de Huellas Dactilares	85
Fig. 30. Resultados de la Implementación del algoritmo para la binarización de Huellas Dactilares	87
Fig. 31. Representación de la matriz de un hueco y una punta en una imagen binarizada.	89
Fig. 32. Resultados de la Implementación del algoritmo para el adelgazamiento de Huellas	

Dactilares	97
Fig. 33. Representación de la matriz de un fin de cresta y una bifurcación en una imagen adelgazada.....	98
Fig. 34. Resultados de la Implementación del algoritmo para extraer minucias de las Huellas Dactilares	102
Fig. 35. Resultados de la Implementación del algoritmo del análisis de minucias de Huellas Dactilares.	104

Índice de Fórmulas:

Fórmula 1. Algoritmos de clasificación - Normalización.....	38
Fórmula 2. Exactitud	47
Fórmula 3. Exhaustividad.....	47
Fórmula 4. Precisión	48
Fórmula 5. Tiempo de procesamiento.....	48
Fórmula 6. NPCER	49
Fórmula 7. APCER	49
Fórmula 8. Escala de grises.....	71
Fórmula 9. Normalización	72
Fórmula 10. Media	73
Fórmula 11. Varianza.....	73
Fórmula 12. Función de probabilidad basado en el histograma	75
Fórmula 13. Función de densidad acumulativa	75
Fórmula 14. Cálculo del pixel de la imagen ecualizada	75
Fórmula 15. máscara sobel X	78
Fórmula 16. máscara sobel Y	78
Fórmula 17. Gradiente X	78
Fórmula 18. Gradiente Y	78
Fórmula 19. Campo de orientación X	79
Fórmula 20. Campo de orientación Y	79
Fórmula 21. Campo de orientación Local	79

Fórmula 22. Componente campo de vectorial X	79
Fórmula 23. Componente campo de vectorial Y	79
Fórmula 24. Campo de aproximación local	80
Fórmula 25. Mapa de error	80
Fórmula 26. Binarización	86

Resumen

El objetivo del presente trabajo es evaluar algoritmos de clasificación automática para la detección de huellas dactilares alteradas basado en la extracción de minucias (EM) y campo de orientación (CO), teniendo como primero paso del método, construir la base de datos con imágenes de huellas dactilares; segundo paso, la seleccionar dos algoritmos de clasificación automática para la detección de huellas dactilares alteradas; tercer paso, implementar algoritmos de clasificación automática que detectan huellas dactilares alteradas; y el último paso, evaluar los resultados. De las pruebas realizadas se obtuvo que los mejores rendimientos para el algoritmo de clasificación automática basado en la extracción de minucias son: exactitud 83%, exhaustividad 94%, precisión 78%, APCER 28% y NPCER 6% en la alteración tipo obliteración; en el tiempo de procesamiento de 0.564s, en la alteración tipo rotación central; y los mejores resultados para el algoritmo basado en el campo de orientación son: exactitud 77%, exhaustividad 64%, precisión 87%, APCER 10% y tiempo de procesamiento de 0.230s, en la alteración tipo obliteración y NPCER 34% en la alteración tipo corte Z. Se concluye ambos algoritmos tienen mejores rendimientos en la alteración tipo obliteración y bajo rendimiento en las alteraciones tipo corte Z y rotación central. También se concluye que, el método propuesto es eficiente para detectar huellas dactilares alteradas y también para la detección de ataques de presentación biométrica, obteniendo tasas bajas en NPCER y APCER, implicando una menor inspección manual de huellas dactilares, por parte de expertos en huellas dactilares alteradas.

Palabras Clave: biometría, huella dactilar, alteración, extracción de minucias, campo de orientación.

Abstract

The objective of this work is evaluate algorithms for detection altered fingerprints based on minutiae extraction (ME) and field of orientation (FO), having as the first step of the method, building the database with fingerprint images; second step, the selection of two algorithms for the detection of altered fingerprints; third step, the implementation of algorithms for altered fingerprints; and the last step, evaluating results. Tests performed it obtained best performances for the algorithm based on minutiae extraction are: accuracy 83%, completeness 94%, precision 78%, APCER 28% and NPCER 6% in the obliteration type alteration; in the processing time of 0.564 s, in the central rotation type alteration; and the best results for the orientation field based algorithm are: accuracy 77%, completeness 64%, precision 87%, APCER 10% and processing time of 0.230 s, in the obliteration type alteration and NPCER of 34% in the Z-slice type alteration. It is concluded that both algorithms have better performance in the obliteration type alteration and poor performance in the Z-slice and central rotation type alterations. It is also concluded, proposed method detecting altered fingerprints and also for detecting biometric presentation attacks, obtaining low rates in NPCER and APCER, involving less manual inspection of fingerprints, by experts in altered fingerprints.

Keywords: biometry, fingerprint, alteration, minutiae extraction, orientation field.

I. INTRODUCCIÓN

1.1. Realidad problemática.

Las huellas digitales alteradas: Un desafío en las entidades encargadas de aplicar la Ley de Identificación.

Examinadores de huellas digitales del FBI (Federal Bureau of Investigation), se han encontrado con situaciones en las que los delincuentes, incluidos los que están en el país ilegalmente, han alterado deliberadamente sus huellas dactilares ellos mismos o con la ayuda de profesionales de la medicina. Ellos falsamente creen que hacerlo sería evitar que los funcionarios policiales descubrieran su verdadera identidad. En el 2014 el FBI llevó a cabo un estudio de registros de huellas digitales alteradas almacenadas en su IAFIS (Sistema para Identificación Huellas Dactilares Automatizado e Integrado), que se convirtió en el sistema de próxima generación de identificación (NGI) el 7 de septiembre de 2014. Los examinadores de huellas digitales aislaron 412 registros de huellas dactilares del IAFIS que mantenían indicadores de alteración deliberada de la huella digital. Una revisión de los registros de los patrones y tendencias que los funcionarios de Massachusetts tuvieron el mayor número de registros con personas que habían modificado las huellas dactilares, seguido por Nueva York, Texas, California y Arizona. [1]

Freddy Dávila fue detenido el 9 de junio de 2011, por la protección aduanera y fronteriza en una declaración emitida el día 29, mientras intentaba abordar un vuelo de Boston a Santo Domingo en República Dominicana. Fue detenido y cuestionado por los oficiales de CBP que encontraron que sus huellas dactilares habían sido alteradas intencionalmente para evitar la identificación de aplicación de la ley. Dávila fue detenido por los funcionarios CBP, dijo la Agencia. [2]

Dentro del procedimiento de medicina legal, cuando ingresan los NN (Cadáveres), se toma las huellas dactilares de los 10 dedos, luego estos son comparados con la base de datos de la RENIEC, casi siempre llegan a la identificación de los NN, no llegan a la identificación solo en los casos cuando las huellas dactilares han sido alteradas [3]. Como se podrá observar en este caso ya existe una necesidad para tener sistemas de detección de huellas dactilares alteradas.

La Policía capturo a William Calderón (30), alumno cursaba ciclo quinto en Universidad Nacional Mayor de San Marcos, quien se clonaba huellas dactilares, las cuales serían utilizadas posteriormente por postulantes a dicha universidad. Los detectives encontraron que la imitación de las huellas dactilares había sido hecho con material utilizado por odontólogos de color de la piel [4].

Las fuerzas de seguridad han estado usando la técnica de identificación de huellas dactilares para el reconocimiento biométrico. Pero debido a la utilización con éxito de este sistema, ha habido actividades extremas de eludir este sistema. Una de esas actividades es alterar la huella dactilar. Este ha sido un centro primordial de preocupación para muchos debido a la alteración de huellas digitales es mucho más fácil que las alteraciones de otro rasgo biométrico. [5]

El reconocimiento de huellas dactilares ha sido utilizado con éxito en la fuerza pública para identificar intrusos o sospechosos durante casi 100 años. El éxito del sistema de reconocimiento de huellas dactilares en individuos que se identificaron con precisión ha permitido que algunas personas empelar medidas extremas con el fin de evadir el sistema de identificación. El objetivo principal de la alteración de huellas digitales es escapar de identificación mediante técnicas diferentes de raspar ni se desgasta por el roce del flujo de huellas digitales cresta y la quema de los dedos y la realización de la cirugía plástica. El uso

de huellas digitales alteradas es una máscara de identidad para dar una serie ataque contra un sistema biométrico de control de fronteras, ya que, se anula el propósito mismo por el cual el sistema fue diseñado. Con el fin de detectar ataques basados en huellas dactilares falsas, muchos software y hardware se ha propuesto. Sin embargo, los problemas de la huella digital alterada no se han estudiado en la literatura y no existen técnicas disponibles para detectar las huellas digitales alteradas. [2]

Los trabajos previos para la solución del problema de detección de huellas dactilares alteradas son los siguientes:

YANG, LI, & ZHANG [6], investigación “Altered Fingerprint Detection Based on Orientation Field”, en China, proponen un método para detectar huellas dactilares alteradas basado en el campo de orientación, para Identificar a los delincuentes que modifican las crestas de sus huellas dactilares para no ser identificados por los AFIS. Para esto, se propone un método de detección de huellas dactilares alteradas basado en la discontinuidad del campo de orientación y la discontinuidad del mapa de diferencia de campo de orientación, y se propone un operador para extraer esta discontinuidad. Primero, calcula el campo de orientación de la huella dactilar y su mapa de diferencia, y se mejora el mapa de diferencia. Luego, el operador propuesto extrae las características discontinuas del campo de orientación y el mapa de diferencia mejorado. Finalmente, en función de las características discontinuas, las huellas dactilares alteradas y las huellas dactilares reales se clasifican usando máquinas de vectores de soporte (SVM). El resultado, es que, cuando la FAR es del 10%, el TDR de detección de huellas dactilares deformadas aumenta en más del 10%. Los resultados experimentales muestran que el método propuesto detecta imágenes de huellas dactilares modificadas.

Shehu, James, & Palade [7], investigación “Detecting an Alteration in Biometric

Fingerprint Databases”, en Japón, investigan posibles técnicas forenses que podrían utilizarse para la detección de manipulación de plantillas de huellas dactilares biométricas almacenadas. Asegurar la integridad de las plantillas biométricas de huellas dactilares en las dataset de huellas dactilares es de suma importancia. Para esta tarea, utilizaron patrones locales binarios y un kernel para el analizar los principales componentes, la técnica de Extracción de Minucias, con un enfoque de clasificación Support Vector Machine (SVM). Las plantillas originales y manipuladas se utilizan para entrenar al clasificador SVM. Las funciones de huellas dactilares se pueden clasificar en funciones de tres niveles. En primer Nivel, capturan detalles microscópicos de huellas dactilares. Para el Nivel 2, se extraen minucias de huellas dactilares, que incluyen bifurcaciones de crestas y terminaciones. Para el nivel 3, se utilizan los atributos dimensionales. Los resultados que se encontraron fue que la precisión fue del 86,15% y el Recall fue del 96,5%. El método propuesto detecta la manipulación de imágenes de huellas dactilares con una precisión del 90,5%.

Vinoth & Saravanakumar [8], en su investigación denominada “Region based Minutiae Mass Measure for Efficient Finger Print Forgery Detection in Health Care System”, en la india, las organizaciones médicas mantienen registros de diferentes pacientes datos confidenciales que deben protegerse del acceso ilegal. Las huellas dactilares se han utilizado como clave, hay usuarios mal intencionados que intentan ingresar al sistema y robar información. Con respecto al problema de la detección de falsificaciones, se presenta un enfoque eficiente de medición masiva de minucias basadas en la región (RMMM) para respaldar la seguridad de los sistemas de atención médica. El método primero mejora la imagen de la huella dactilar de entrada aplicando un filtro Gabor para eliminar el ruido. Luego, el ruido se elimina mejorando la calidad de la imagen, con la nitidez de los bordes de las crestas presentes en la imagen. Luego, la imagen se ha dividido en varias regiones y, para cada imagen de sección, el método extrae varias características. Usando las características extraídas, el método estima el valor de masa de Minucia para cada imagen seccional. Lo mismo se ha realizado en la imagen de

prueba de entrada y, en función del valor de masa de las minucias, se ha detectado la impresión falsificada. Como resultado de la aplicación del algoritmo RMMM, se obtuvo una exactitud del 95% y un tiempo de ejecución de 12 segundos. El método ha producido resultados eficientes en la detección de huellas dactilares falsificadas y mejora la precisión de la clasificación.

Gottschlich, Tamsy, & Huckemannz [9]. en su investigación denominada "Perfect Fingerprint Orientation Fields by Locally Adaptive Global Models", en Alemania, debido a la diferente resolución y tamaño de las imágenes de huellas dactilares, la extracción de características automática es mucho más difícil y lenta, en esta investigación los autores presentan un enfoque novedoso para modelar globalmente un campo de orientación (OF), combinado con métodos adaptativos localmente. Como resultados se obtuvo que, para 20 puntos de anclaje y con un tiempo de ejecución es de 130.5, 180 y 230 segundos. Se concluye que este modelo se adapta perfectamente al 'verdadero OF'.

Singh, Ayub, & Jp [10], en su investigación denominada "Minutiae Extraction Based Matching Between Normal And Altered Fingerprint", en la India, debido al uso de huellas dactilares falsas en los sistemas de seguridad y de control fronterizo. Este documento se basa en experimentos, y crea una base de datos de huellas dactilares humanas en vivo y descubre el puntaje de coincidencia de la misma persona, entre el mismo dedo en función de la extracción de minucias, es decir, calcula las minucias del dedo normal y el mismo dedo alterado, encuentra el porcentaje de coincidencia. El resultado basado en el experimento mostró un 18% promedio de dedos coincidentes, que es una comparación que significa que el mismo dedo es normal y alterado. y el 45% más alto y el 10% más bajo. La conclusión importante es que la puntuación promedio de coincidencia del mismo dedo es del 18%.

Patil & Phulpagar [11], en su investigación denominada "Efficient Altered Fingerprint

Analysis and Rectification of Distorted Fingerprint”, en la India, el problema de la clasificación de casos falsos habla ampliamente de la baja resolución de imágenes, cuando la identificación del usuario pueda autenticar a un usuario errado. La detección de distorsión se puede categorizar en un problema de clasificación de dos clases que se puede resolver usando el algoritmo de extracción de minucias y campo de orientación y un clasificador LibSVM. La rectificación de huellas dactilares distorsionadas (o equivalentemente estimación de campo de distorsión) se considera un problema de regresión, donde la entrada es una huella dactilar distorsionada y la salida es el campo de distorsión. Para tales problemas, es imprescindible la detección y rectificación de huellas dactilares distorsionadas. Se obtuvieron resultados, donde el tiempo de ejecución del sistema para la detección de distorsión propuesto es de 7 seg, y del sistema base 16.5 seg; y el sistema de rectificación de distorsión propuesto 41seg y el sistema base 131 seg. El resultado experimental muestra que se mejora la detección de distorsión y el procesamiento de rectificación.

Haraksim, y otros [12], en su investigación denominada “Altered fingerprint detection – algorithm performance evaluation”, en Chipre, debido al uso de huellas dactilares alteradas en las entidades que se encargan de implantar la ley (y, en particular, a las entidades de control fronterizo), se presenta un estudio comparativo sobre el desempeño de algoritmos de detección de huellas dactilares alterados. Se han evaluado diferentes algoritmos de diferentes instituciones en dos conjuntos de datos diferentes. Ambos conjuntos de datos presentan alteraciones reales en los dedos y la verdad fundamental con respecto a la alteración se conoce a priori, ya que, en algunos casos, también estaban disponibles las huellas dactilares pre-alteradas correspondientes. Los algoritmos utilizados son: La aproximación del campo de orientación (OFA), el análisis de orientación de minucias (MOA), nuevamente el análisis de distribución de minucias (MDA) y el análisis de densidad de puntos singulares (SPD). Como resultado de los diferentes algoritmos de detección de huellas dactilares alteradas evaluados, 10% ERR , NPCER de 14.41% para OFA-MOA y APCER de 27.62% para OFA-MOA. En

ambos conjuntos de datos, todos los algoritmos generaron una ERR de entre el 13 % y el 30 %. El rendimiento obtenido en ambos conjuntos de datos producidos por algoritmos de referencia de última generación o personalizados es superior al 10 % de EER notificado en estudios anteriores.

Papi, Ferrara, Maltoni, & Anthonioz [13], en su investigación denominada “On the generation of synthetic Fingerprint alterations”, en Alemania, proponen algunas técnicas para generar huellas dactilares sintéticas alteradas y demostrar la utilidad de los conjuntos de datos generados para desarrollar, ajustar y evaluar algoritmos que detectan la huella dactilar alterada. Debido a la falta de bases de datos públicas de huellas dactilares alteradas, la herramienta de generación propuesta (y disponible gratuitamente) puede ser un instrumento válido para impulsar la investigación sobre estos desafiantes problemas. Se enfocó en las tres más comunes alteraciones encontradas en situaciones reales, obliteración, central rotación y Corte Z. Se obtuvo como resultado ERR 10.3% y FMR 19.6%. Se concluye que las alteraciones generadas parecen poco realistas al ojo humano, demostramos su utilidad en una aplicación práctica.

Mini & Anoop [14], en su investigación denominada “Two stage framework for altered fingerprint matching”, en la India, presenta un método de dos etapas para la coincidencia de huellas dactilares alteradas. Debido a los errores en los sistemas al comparar huellas dactilares alteradas. En la primera etapa, el campo de orientación de la cresta global aproximado de la huella dactilar alterada se compara con el campo de orientación de su pareja inalterada. Si esta coincidencia tiene éxito, las huellas digitales pasan a la segunda etapa. La segunda etapa comienza con la selección de la región inalterada del FP alterado y la misma región de las parejas inalteradas. El emparejamiento en esta etapa se realiza mediante la extracción de la textura de la cresta y la frecuencia de la cresta de la región de interés seleccionada. La distancia euclidiana se utiliza en ambas etapas para calcular la puntuación de coincidencia.

Obtuvo como resultados: Tasa Genuino Aceptación (GAR) y la Tasa Falsa Aceptación (FAR) para el limite seleccionado (70FP). GAR da la coincidencia con éxito de FP alterado en 90% y FAR da la tasa de falsa coincidencia 3.33%. Asimismo, concluye que los resultados muestran que este método es adecuado para encontrar FP alteradas.

Mini & Anoop [15], en su investigación denominada "Altered Fingerprint Matching Using Ridge Texture and Frequency in the Unaltered Region", en la india, presenta un método para la coincidencia de huellas dactilares alterada utilizando textura y frecuencia de cresta en la región inalterada. La alteración de huellas dactilares es el proceso de cambiar la estructura de la cresta espaciada regularmente por medios mecánicos o químicos para ocultar la identidad del AFIS. La puntuación coincidente obtenida tanto para la textura como para la frecuencia de las crestas se fusiona para obtener la puntuación final. Los resultados de una coincidencia de uno a muchos entre las huellas dactilares alteradas sintéticamente y sus compañeros inalterados muestran que este método es adecuado para la coincidencia de huellas dactilares alteradas. Obtuvo como resultados: Tasa Genuino Aceptación (GAR) y la Tasa Falsa Aceptación (FAR) para el limite seleccionado (70FP). GAR da la coincidencia con éxito de FP alterado en 84.28% y FAR da la tasa de falsa coincidencia 12.85%, concluye Los resultados muestran que este método es adecuado para encontrar FP alterada.

Gottschlich, Mikaelyan, Olsen, Bigun, & Busch [16] en su investigación denominada "Improving Fingerprint Alteration Detection", en Croacia, La alteración de huellas dactilares es un tipo de ataque en el que el atacante se esfuerza por evitar la identificación. p. Ej. en el control de fronteras o en investigaciones forenses. La detección de alteraciones de huellas dactilares tiene como objetivo descubrir automáticamente la ocurrencia de tales ataques clasificando las imágenes de huellas dactilares como "normales" o "alteradas". Todas las imágenes de huellas dactilares se han procesado utilizando el método de paso de banda direccional factorizado (FDB). Primero, la región de interés (ROI) se estimó mediante el

método FDB y, a continuación, las imágenes se ajustaron automáticamente eliminando todas las filas y todas las columnas que contienen solo píxeles de fondo. Se ha realizado una inspección visual para garantizar que el preprocesamiento automático mediante el método FDB conduzca a ROI adecuados para todas las imágenes. Este procedimiento se ha aplicado a varios algoritmos incluyendo el MDA y OFA. Los resultados obtenidos para el algoritmo MDA es, para NPCER 1% el APCER 45% y para el algoritmo OFA es, para NPCER 1% el APCER 20%. En resumen, se propuso cuatro nuevas características para la detección de alteraciones de huellas dactilares. Dos de estas se encuentran entre las tres características con mejor rendimiento en el punto de referencia considerado con un NPCER del 1%, lo que es de especial relevancia para aplicaciones prácticas en el control fronterizo.

Nagthane & Fadewar [17], en su investigación denominada "Matching of Altered Fingerprint using Minutiae Technique", en la India, debido a que los delincuentes están eludiendo su identificación del sistema de reconocimiento de huellas dactilares al alterar sus dedos. En esta investigación se analiza los tipos de huellas digitales alteradas y correspondencia de la huella digital alterada, utilizando técnica de correspondencia de minucias. Los resultados que se obtuvieron fueron el 79.5% de Coincidencias de 40 Huellas Dactilares Alterada. Se concluye La huella digital alterada de las existentes técnicas que sólo funciona cuando la zona alterada es lo suficientemente grande y (detecta la huella digital se altera o no). Si el área alterada es pequeña, entonces no se considera como la huella digital alterado. El sistema propuesto supera este problema ya que coincide a pesar de que el área alterada es pequeña. También reconoce la cantidad de área alterada.

Mohan [18], en su investigación denominada "Embedded Fingerprint Matching Mechanism in HIS for Health Care Service", en la India, en este trabajo se analiza el problema de la alteración de huellas dactilares y la amenaza de huellas digitales alteradas. Debido a que el NFIQ carece de la capacidad para distinguir las huellas dactilares alteradas y naturales.

se desarrolló un algoritmo de huellas dactilares alteradas que detectar automáticamente sobre la base de las características del campo de orientación y extracción de minucias de la huella digital. Obtuvo como resultados: Tiempos por Pasos del Algoritmo Extracción de minucias : 5s; Formación de trío : 8s; Comprobación de Coincidencia : 1.25s, concluye Este algoritmo satisface los 2 requisitos para algoritmo de detección de alteración: a) tiempo de funcionamiento rápido y, b) la facilidad de integración en AFIS / NGI.

Chen, Chang, Huang, He, Wang & Huang [19], en su investigación denominada “SVM-Based Synthetic Fingerprint Discrimination Algorithm and Quantitative Optimization Strategy”, en la china, en este trabajo se propone un algoritmo eficaz para la discriminación de huellas digitales sintético y una estrategia de optimización cuantitativa, Obtuvo como resultados: El método se puede lograr una precisión de reconocimiento superior a 98% para los dos tipos de huellas dactilares sintéticos separado, así como para los casos mixtos., concluye Este método supera el dilema de la selección de parámetros basados en la experiencia y también puede guiar a problemas de optimización de computación inteligentes en otros campos.

Latha & Manikandan [20], en su investigación denominada “Critical Analysis and Detection of Altered Fingerprints”, en la india, en este trabajo se propone que la detección de todos los tipos de huellas dactilares y huellas dactilares falsas alteradas sin salir de cualquier otra consideración. Se centra en optimizar el algoritmo NFIQ modificado mediante la implementación de Neuro Difuso basado en imágenes de huellas dactilares., Obtuvo como resultados: Exactitud : Difuso ≤ 1000 , Red Neuronal ≤ 2000 y Neuro Difuso ≤ 3000 . Ratio de falsa alarma 4% y Ratio de detección verdadera 70%, concluye Algunos de los tipos de huellas dactilares alterados no pueden ser identificados por el NFIQ. Para superar el problema, el algoritmo optimiza la técnica neuro difusa.

Yoon, Feng, & Jain [21], en su investigación denominada “Altered Fingerprints:

Analysis and Detection”, en Korea del Sur, en esta investigación se proponen desarrollar una técnica para detectar automáticamente huellas dactilares alteradas, ya que la alteración de huellas dactilares es una seria amenaza para AFIS, ya que revoca una de las premisas fundamentales de que la huella dactilar es persistente durante toda la vida. La técnica para detectar automáticamente huellas dactilares alteradas fue basada en el análisis del campo de orientación y la distribución de minucias. Los resultados obtenidos con una tasa de falsos positivos del 2,1 %, donde las huellas dactilares naturales en NIST SD14 con el valor NFIQ de 5 se determinan como huellas dactilares alteradas, el algoritmo propuesto alcanza una tasa de verdaderos positivos del 70,2 %.

Como podemos observar la detección de las huellas digitales alteradas son un desafío para la Aplicación de la Ley de Identificación, en todos los países del mundo. En cuanto a los países desarrollados ellos ya están implementando sistemas para la DETECCIÓN DE HUELLAS DACTILARES ALTERADAS; pero países como el Perú (Instituciones Gubernamentales y Privadas), aun no se encuentra información que se esté implementando la tecnología propia para, la DETECCION DE HUELLAS DACTILARES ALTERADAS, para poder hacer frente a esta problemática, la misma que se muestra en la Fig. 1. Por tal motivo, en esta investigación, se hizo un esfuerzo por EVALUAR ALGORITMOS DE CLASIFICACION AUTOMATICA PARA LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS.

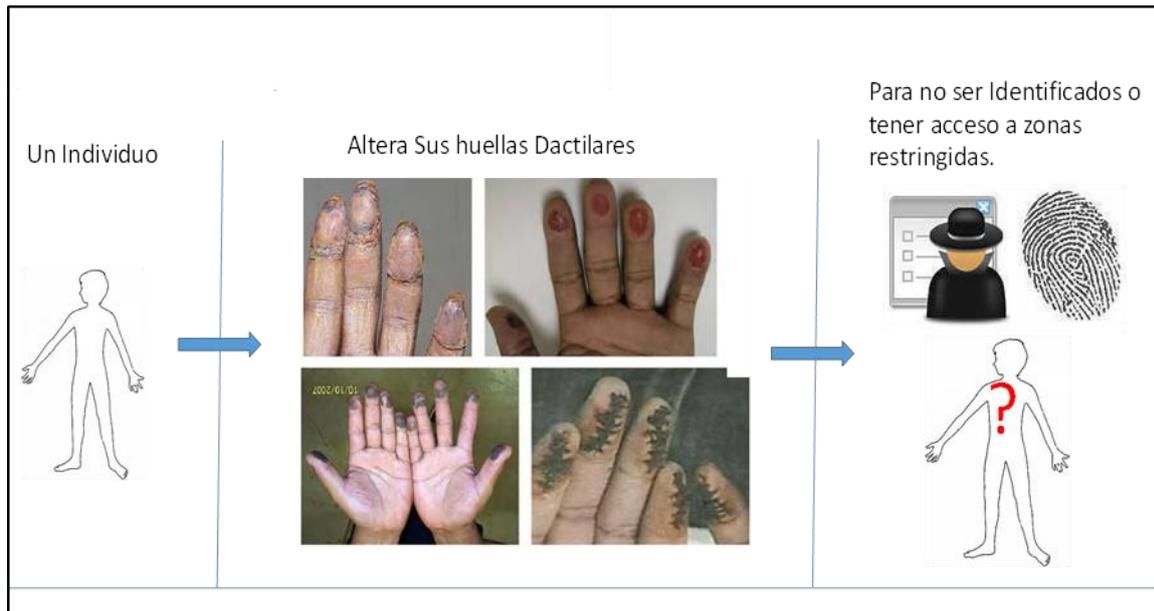


Fig. 1. Esquema de la Realidad Problemática. Fuente : *Elaboración Propia, las imágenes de huellas dactilares alteradas fueron obtenidas de [22], [5].*

1.2. Formulación del problema

¿Cómo detectar una huella dactilar alterada en la autenticación biométrica?

1.3. Hipótesis

Mediante el uso de algoritmos de clasificación automática se podrá detectar las huellas dactilares alteradas.

1.4. Objetivos

Objetivo general

Evaluar dos algoritmos de clasificación automática para la detección de huellas dactilares alteradas.

Objetivos específicos

- A. Construir base de datos con Imágenes de huellas dactilares.

- B. Seleccionar dos algoritmos de clasificación automática para la detección de huellas dactilares alteradas.
- C. Implementar algoritmos de clasificación automática seleccionados para la detección de huellas dactilares alteradas.
- D. Evaluar resultados de la detección de huellas dactilares alteradas.

1.5. Teorías relacionadas al tema

(1) Biometría

Biometría es una ciencia que permite identificar una persona de acuerdo a sus características físicas, químicas o en el comportamiento del individuo. Ejemplos: compartir recursos de ordenador en red, acceso a instalaciones nucleares, realizar transacciones financieras. Servicios basados en web (Banca por Internet) , han subrayado la necesidad que estos cuenten con sistemas de control de la identidad confiable. La biometría puede utilizar las características de un individuo (corporales o conductuales), para verificar su identidad. Estos sistemas biométricos pueden tener algunas limitaciones, pero a su vez tienen la ventaja, de que no puede ser fácilmente robados, refuerzan la seguridad, y ayudan al usuario al no utilizar contraseñas para acceder a los sistemas [23]. en la Fig. 2, se muestra ejemplos de rasgos biométricos que se pueden utilizar para la autenticación de un individuo.

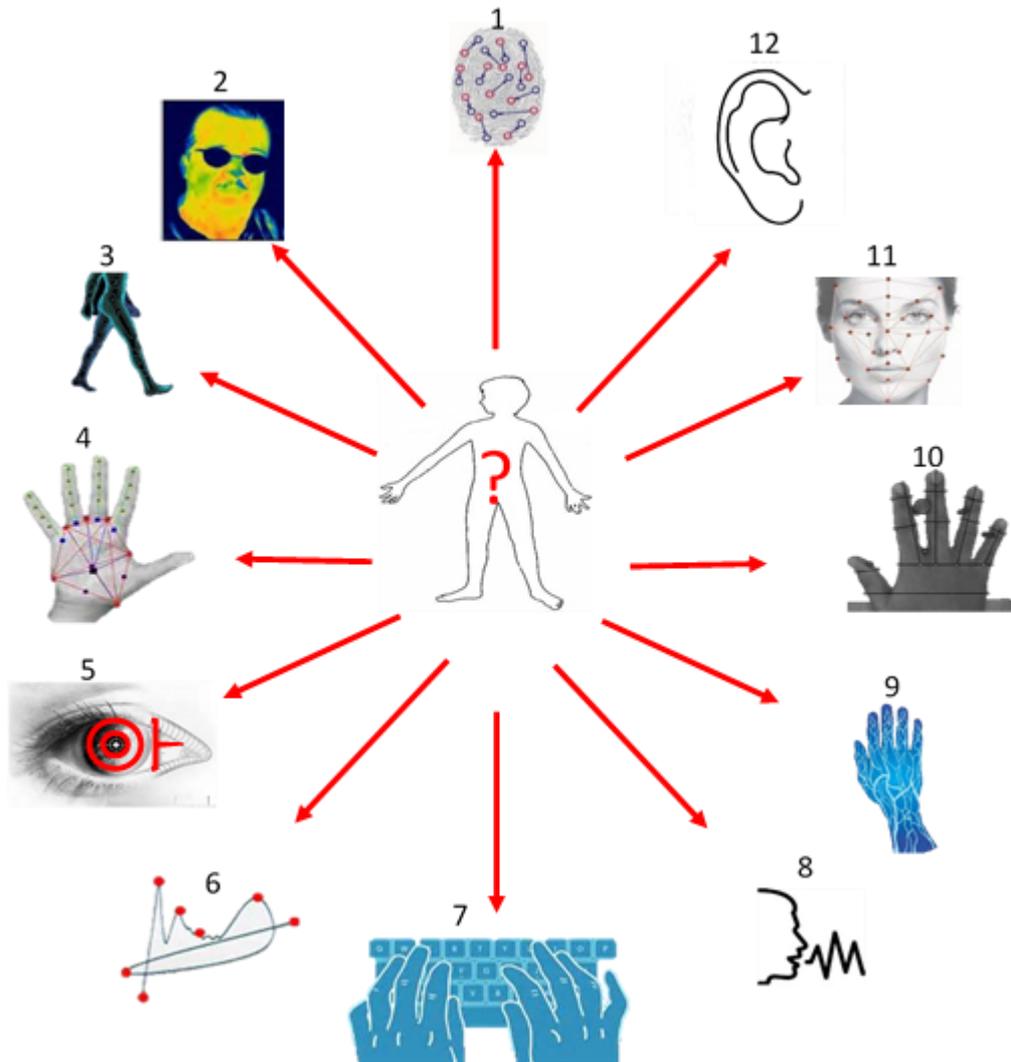


Fig. 2. Ejemplos de rasgos biométricos que se pueden utilizar para la autenticación de un individuo. 1) Dactilar, 2) Termo grama facial, 3) Caminar, 4) Palmar, 5) Iris, 6) Firma, 7) Patrón de Pulsación de teclas, 8) voz, 9) Patrón de las Venas, 10) Geometría de la mano, 11) Rostro, 12) Oreja. Fuente : Elaboración Propia.

(2) Modos de operación (Verificación Vs Identificación)

Un sistema biométrico se puede usar para la verificación o para la identificación. **En la verificación**, se valida la identificación del individuo mediante la comparación de sus características capturadas con su propia plantilla biométrica

almacenada en un dataset. El propósito es que una misma identificación no la utilicen varios individuos. **En la identificación**, el sistema reconoce un individuo buscando en la base de datos en varios individuos. El propósito es identificar a un individuo (falla cuando el individuo no se encuentra en la dataset) [23].

(3) **Aplicaciones de la Biometría.**

El establecimiento de la identidad de una persona con alta confianza se está volviendo crítica en un número de aplicaciones en nuestra sociedad muy interconectada. Preguntas como "¿Es ella realmente quien dice ser?", "¿Es esta persona autorizada para utilizar este servicio?" O "¿Está en la lista publicada por el gobierno?" Son rutinariamente se plantea en una variedad de escenarios que van desde la emisión una licencia de conducir a la obtención de entrada en un país. La necesidad de técnicas de autenticación de usuario fiables ha aumentado como consecuencia de las crecientes preocupaciones acerca de la seguridad, y los rápidos avances en la creación de redes, la comunicación y la movilidad. Por lo tanto, la biometría se está incorporando cada vez más en varias aplicaciones diferentes. Estas aplicaciones se pueden clasificar en tres grupos principales [23].

- a) Las aplicaciones comerciales: seguridad, comercio por Internet, cajeros automáticos o tarjetas de crédito, telefonía móvil, etc.
- b) Aplicaciones en el gobierno: identidad, seguridad, control de las fronteras, etc.
- c) Aplicaciones forenses: identificación cadáveres, etc.

(4) **Características de la Biometría**

Existen siete características de la biometría [23].

- a) **Universalidad:** Todo individuo que accede a la aplicación debe tener el rasgo.
- b) **Singularidad:** El rasgo utilizados deben ser diferentes a otros individuos.
- c) **Permanencia:** El rasgo utilizado debe ser invariante.

- d) **Cuantificación:** Los rasgos adquiridos deben contener características procesables.
- e) **Rendimiento:** La precisión debe cumplir con los límites establecidos por la aplicación.
- f) **Aceptabilidad:** Los individuos deben aceptar el uso de su rasgo biométrico.
- g) **Elusión:** El rasgo del individuo puede ser alterada o imitada.

(5) Huellas Dactilares.

Las huellas dactilares es la impresión de la epidermis (Crestas), de la yema de un dedo humano o la palma de la mano. Una huella dactilar se compone de un conjunto de líneas llamadas líneas de cresta, que fluyen principalmente en paralelo, formando un patrón llamado patrón de cresta. Las huellas dactilares pueden estar impresas con tinta u obtenidas con un lector de huellas dactilares [24], como se muestran en Fig. 3.

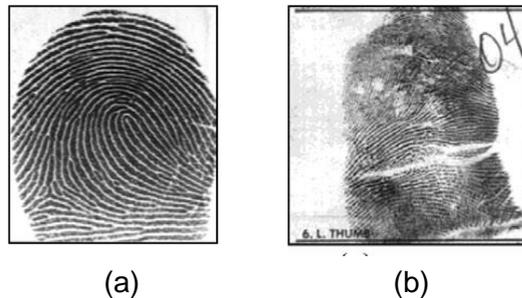


Fig. 3. Imágenes de Huellas Dactilares. (a) Obtenidas con lector de huellas dactilares; (b) Impresión fuera de línea entintado;
Fuente: [22]

(6) Características de Huellas Dactilares

En las huellas dactilares lo podemos agrupar en dos: Patrones de Huellas dactilares y Minucias.

a) Patrones de Huellas.

Hay principalmente 3 tipos de patrones de huellas dactilares, como bucle, espiral o arcos [25], los cuales se muestran en la Fig. 4.

Bucles: los bucles comienzan en un lado y salen por el otro lado del dedo. El bucle cubital y el bucle radial son dos tipos de bucles: la pendiente de los bucles cubitales es hacia el dedo meñique, mientras que la pendiente de los bucles radiales es hacia el pulgar.

Espirales: Los espirales forman un patrón circular o en espiral.

Arcos: Los arcos son como montañas estrechas que se inclinan hacia arriba y luego hacia abajo.

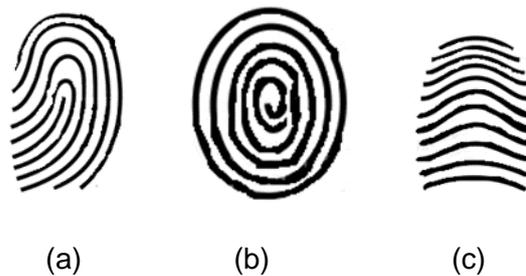


Fig. 4. Patrones de Huellas Dactilares. (a) Patrón tipo Bucle; (b) Patrón tipo espiral; (c) Patrón tipo arco; Fuente: Elaboración propia.

b) Minucias.

Las minucias son características importantes de una huella dactilar, y al usar esto podemos comparar una huella dactilar con otra, se pueden, las que se muestran en la Fig. 5. Las distribuciones minuciosas de las huellas dactilares alteradas son diferentes de las huellas dactilares naturales [24]:

- Final: Es el final de una cresta.
- Bifurcación: Una cresta que se divide en dos crestas.

- Cresta independiente: Cresta que recorre una distancia corta.
- Isla: Pequeña cresta que no está conectada a todas las demás crestas.
- Laguna: una sola cresta que se bifurca y se reúne poco después para continuar como una sola cresta.
- Aguijón: una bifurcación con una cresta corta y una más larga.
- Puente: una cresta corta que corre entre dos crestas paralelas.



Fig. 5. Minucias de Huellas Dactilares. Fuente: Elaboración propia.

(7) Algoritmos para Detección de Huellas Digitales.

Los algoritmos son un conjunto de órdenes para realizar una tarea. Los diseñadores de algoritmos para identificar e investigar los siguientes cinco problemas principales en el diseño de sistemas de procesamiento de huellas digitales automatizadas: Adquisición de huella digital, mejora de la imagen, función de extracción (por ejemplo, las minucias), a Comparación, y la indexación / recuperación [22].

a) Adquisición de la Imagen.

Las huellas dactilares se pueden adquirir mediante la aplicación de tinta sobre un dedo, a esto se llama "entintado". Cuando la impresión de un dedo esta sobre una superficie donde el dedo a tocado, se llama "latentes" y son capturadas electrónica y manualmente mediante fotografía o cinta adhesiva. A las imágenes capturadas con los métodos anteriormente mencionados se les llama imágenes "off-line". Los dispositivos que capturan las huellas dactilares en vivo ejecutan algoritmos y casi siempre conectados a un ordenador [22] .

Las imágenes de huellas dactilares se muestran en la Fig. 6.

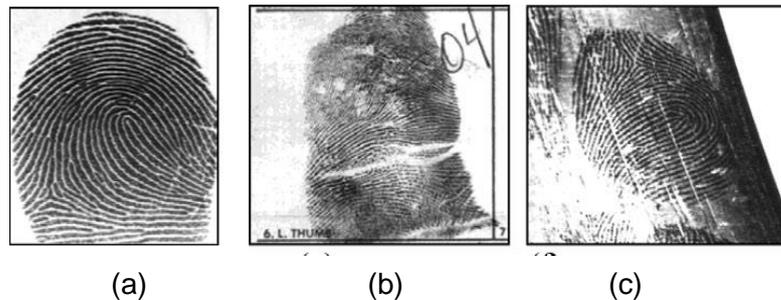


Fig. 6. Adquisición de Huellas Dactilares. (a) Una huella dactilar capturada en vivo; (b) Una impresión fuera de línea entintado; (c) Una huella dactilar latente. Fuente: [22]

b) Mejoramiento de la imagen.

Las imágenes adquiridas por los distintos métodos mencionados anteriormente, tienen ruido, para reducir este ruido, se aplican algoritmos de mejora, como se muestra en la Fig. 7. Estas pueden ser de dos tipos [22]:

- (a) Mejora de huellas dactilares latentes, en el cual el especialista forense ejecuta algoritmos como: región de interés, cambiar tamaño de imagen, cambiar color, rotar imagen, etc...

(b) Mejora de Imágenes automáticamente, las aplicaciones utilizan algoritmos automatizados para mejorar huellas dactilares: región de interés, cambiar tamaño de imagen, cambiar color, rotar imagen, etc...

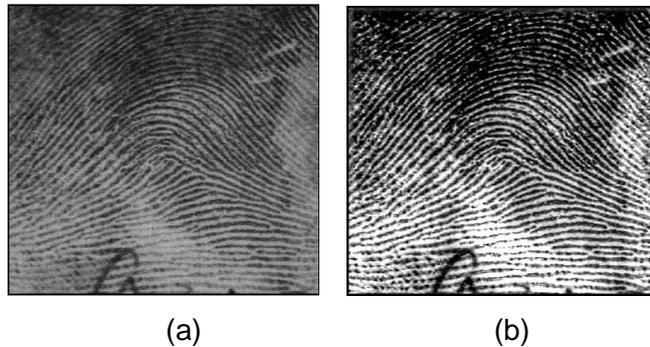


Fig. 7. Ejemplo de mejora de huella dactilar. (a) huella dactilar sin mejora; (b) Huella dactilar mejorada; Fuente: [22]

c) Extracción de características.

Singularidades locales de la huella digital, comúnmente conocidos como puntos minucias, siendo los extremos de las crestas y las bifurcaciones los más importantes, porque a partir de estas se componen las demás características. Sin embargo, algoritmos más automáticos de extracción de minucias de huellas digitales sólo consideran finales de crestas y bifurcaciones. Estos algoritmos son los mismos métodos desarrollados por expertos forenses para imitar la ubicación de las minucias.

Comúnmente el algoritmo que extrae minucias utiliza el adelgazamiento de la imagen, con la cual localizan a la minucia en una coordenada x,y [22].

d) Comparación.

Comparación de huellas dactilares se puede definir como el ejercicio de la

búsqueda de la similitud o diferencia. Mientras más sea la puntuación de coincidencia, lo más probable es que las huellas digitales se acoplan (y, a la inversa, cuanto más baja es la puntuación, menos probabilidades hay una coincidencia) [22].

e) Indexación.

Los algoritmos automáticos de comparación de huellas dactilares gastan mucho tiempo, por lo que resulta deseable (aunque no es necesario), utilizar algoritmos automáticos de indexación y recuperación de huellas digitales. Esta indexación es realizada manualmente por expertos, en gabinetes de archivos de huellas dactilares como se define por un sistema de clasificación la huella digital particular [22].

(8) Vulnerabilidad de los AFIS.

Aunque la estructura de los patrones de huellas digitales puede ser aprovechada, en cierta medida, con el fin de luchar contra los intentos de alteración, los sistemas de identificación de huellas dactilares en funcionamiento son de hecho vulnerables a este tipo de ataques. Es muy difícil para el estado del arte de las AFIS (sistemas automáticos de identificación dactilar), identificar las huellas dactilares alteradas de manera significativa. No es necesario alterar toda la región de la piel de fricción en la mano humana, ya que sólo una parte del patrón de crestas de fricción se utiliza en los sistemas de identificación más prácticos. Dependiendo del nivel de seguridad y la aplicación prevista, las áreas de crestas de fricción que se registran y se comparan por sistemas de identificación pueden variar de toda la mano de un solo dedo. Asimismo, muchos sistemas de huellas dactilares no forenses utilizan imágenes de huellas dactilares de fricción (o planos) en lugar de imágenes

laminadas. También, no es necesario alterar completamente la entrada de huellas digitales a los sistemas automatizados, ya que su precisión de identificación está limitada por la calidad de imagen, los requisitos de rendimiento y la cantidad de imágenes. Aunque la precisión de los sistemas automatizados en identificar las huellas digitales de baja calidad puede ser mejorada significativamente con la ayuda de operadores humanos (como se observa en la práctica de identificación latente), ningún software específico todavía está disponible para reconstruir el patrón original de huellas dactilares alteradas [26].

(9) Huellas Dactilares Alteradas.

Ahora están surgiendo muchos problemas en la seguridad de los sistemas de huella dactilar. Los delincuentes esconden su identificación de sistemas mediante diversos métodos como huellas digitales falsas, alteración de huellas dactilares, etc. El objetivo principal de alteración de huellas dactilares es evadir la identificación mediante técnicas como la abrasión, corte, quema los dedos, realizando cirugía plástica, etc. La alteración de las huellas digitales, son diferentes de las huellas dactilares falsas. Huellas digitales falsas son facsímil de dedos reales, estos se utilizan para adoptar la identidad de otra persona. Sin embargo, los dedos alterados son los dedos reales que se utilizan para máscara [17].

(10) Tipos de Huellas Dactilares Alteradas.

De acuerdo con los cambios realizados en los modelos de crestas, huellas dactilares alteraciones pueden ser categorizados en tres grupos y son las siguientes:

- a) La obliteración
- b) Distorsión
- c) La imitación

a) Obliteración.

Los patrones de crestas de la yema de los dedos pueden ser borrados por abrasión, cortar, quemar, la aplicación de productos químicos fuertes, y el trasplante de piel suave. Otros factores tales como la enfermedad de la piel (por ejemplo, lepra) y efectos secundarios de un fármaco contra el cáncer también puede destruir las huellas dactilares. La estructura de las crestas de fricción es apenas visible dentro de la región borrada como se puede ver en la Fig. 8. El tipo obliteración es la forma más popular de alteración. Esto puede ser porque la obliteración, es mucho más sencillo de realizar que la distorsión / imitación, que requiere un procedimiento quirúrgico. Las huellas dactilares obliteradas pueden evadir el software de control de calidad de huellas digitales. Si el área de dedo afectado es pequeña, los softwares de evaluación de calidad de huellas digitales existentes no pueden detectar una huella digital alterada, pero el AFIS es probable que coincida con éxito la huella digital dañada con la huella digital original acoplada. Pero, si el área alterada es suficientemente grande, el software de control de calidad de huellas dactilares puede detectar fácilmente los daños. Para identificar a los individuos con huellas dactilares severamente obliteradas, puede ser necesario tratar estas huellas digitales como imágenes latentes, realizar la búsqueda utilizando el AFIS características marcadas de forma manual [21].



Fig. 8. Obliteración. Fuente: [5]

b) Distorsión.

Los patrones de crestas de la yema de los dedos se pueden convertir en modelos de crestas no naturales, mediante la eliminación de porciones de piel de la yema del dedo, como se muestra en la Fig. 9. Las huellas dactilares distorsionadas tienen modelos de crestas inusuales que no se encuentran en huellas dactilares naturales. Estas anomalías incluyen la distribución espacial anormal de puntos singulares o cambios bruscos en el campo de la orientación a lo largo de las cicatrices. Tenga en cuenta que la discontinuidad del campo de orientación en huellas dactilares naturales por lo general se observó sólo en los puntos singulares. Las huellas dactilares distorsionadas también pueden pasar con éxito la prueba de calidad de huellas dactilares, ya que su estructura de reborde local sigue siendo similar a huellas dactilares naturales, mientras que su patrón de cresta global es anormal. Por ejemplo, una huella digital distorsionada como resultado de intercambio de parches para la piel en el mismo dedo, retiene la misma propiedad cresta (por ejemplo, la frecuencia de cresta y la anchura) sobre toda el área de huella digital. Las huellas digitales alteradas por corte "Z" son de especial interés, ya que conservan su estructura original de la cresta, lo que permite la reconstrucción de la huella dactilar original antes de la alteración [21].



Fig. 9. Distorsión. Fuente: [5]

c) Imitación.

Los patrones de crestas de la yema de los dedos aún pueden conservar patrón de huella, como después de un procedimiento elaborado de alteración de huellas dactilares: 1) una parte de la piel se elimina y la piel restante se tira y se cose juntos, 2) La piel, crestas de fricción entre otras partes del cuerpo se usan para rellenar la parte extraída de la yema del dedo para conciliar con la estructura restante de la cresta, o 3) el trasplante de toda la punta del dedo, rotación circular de la yema del dedo. Simplemente intercambiando la piel de los dedos entre las manos izquierda y derecha se ha evadido con éxito el AFIS. Huellas dactilares imitadas no pueden pasar solamente con éxito el software de evaluación de la calidad de huellas digitales, sino que también puede confundir a los examinadores humanos [21].



Fig. 10. Imitación. Fuente: [5]

(11) Algoritmos de clasificación automática para Detección de Huellas Digitales Alteradas.

CAMPO DE ORIENTACIÓN.

El campo orientación describe el flujo de crestas de las huellas dactilares y se define como la orientación cresta local en el intervalo $(0, \pi)$. Las huellas

dactilares de buena calidad tienen un campo de orientación lisa excepto cerca de los puntos singulares (por ejemplo, el núcleo y delta). Basándose en este hecho, muchos de los modelos de campo de orientación se han desarrollado mediante la combinación del modelo de campo de orientación global para el campo de flujo continuo de la huella digital con el modelo de campo orientación local alrededor de los puntos singulares. El modelo de campo de orientación global representa cualquiera de las huellas digitales de tipo arco, que no tienen ninguna singularidad, o el campo de la orientación general de canto, excepto singularidad en las huellas dactilares. Si el modelo de campo de orientación global solo se utiliza para la aproximación, la diferencia observada y el modelo ideal será distinto de cero solamente alrededor de los puntos singulares. Por otro lado, para las huellas digitales ofuscado, se observa que el error de ajuste de modelo en la región alterada. Por lo tanto, usamos la diferencia entre el campo de orientación observado extraído de la imagen de la huella y el campo de orientación aproximada por el modelo como un vector de características para la clasificación de una huella digital como huella digital natural o de uno alterado. Los principales pasos del algoritmo propuesto se describen a continuación [21]:

- (a) **Normalización.** Con la normalización se busca que todos los pixeles de la huella dactilar se encuentren entre los valores determinados por un mismo factor, facilitando el procesamiento de la huella dactilar [27].

$$N(x, y) = \begin{cases} M_0 + \sqrt{\frac{V_0 \cdot [I(x, y) - M]^2}{V}} & , \text{si } I(x, y) > M \\ M_0 - \sqrt{\frac{V_0 \cdot [I(x, y) - M]^2}{V}} & , \text{otro caso} \end{cases}$$

donde un valor supuesto de M_0 se establece para la Media deseada y un valor supuesto de V_0 se establece para la varianza deseada.

- (b) **Estimación del Campo de orientación.** $\theta(x, y)$, se calcula utilizando el método basado en gradiente. El campo de la orientación inicial es suavizada con un promedio de filtro $M \times M$, seguido de un promedio de las orientaciones en bloques de píxeles $N \times N$. Una máscara de primer plano se obtiene midiendo el rango dinámico de valores de gris de la imagen de la huella en bloques locales y proceso morfológico para llenar agujeros y la eliminación de bloques aislados [21].
- (c) **Aproximación del Campo de Orientación.** El campo de orientación, $\theta(x, y)$, se aproxima mediante un modelo polinomial para obtener $\theta(x, y)$ [21].
- (d) **Extracción de características.** El mapa de error, $\varepsilon(x, y)$, se calcula como la diferencia absoluta entre $\theta(x, y)$ y $\theta(x, y)$ y se utiliza para construir el vector de características [21].

EXTRACCION DE MINUCIAS.

Una minucia de la huella digital indica las características de la cresta, como extremo de una cresta o bifurcación de cresta. La mayoría de aplicaciones que reconocen huellas dactilares usan minucias para la búsqueda. Además de la anomalía observada en el campo de la orientación, también señaló que la distribución minucias de huellas digitales alteradas a menudo difiere de las huellas digitales naturales [21].

Una de las principales características que utiliza AFIS para comparar huellas dactilares son las minucias. Las minucias se encuentran en los extremos de las crestas o en la bifurcación de las crestas.

El análisis se basa en la observación de que la distribución minuciosa de

las huellas dactilares alteradas a menudo difiere de la de las huellas dactilares naturales. El análisis construye un mapa de densidad de las minucias. Los pasos para el análisis de minucias son los siguientes:

1. Como primer paso binarizamos la imagen. La binarización se refiere al proceso de conversión de escala de grises a imágenes binarias. La binarización mejora el contraste entre las crestas y los valles y facilita además los procesos de extraer minucias. En una imagen binaria cada valor de píxel es 0 o 1 (255). Los píxeles negros que representan las crestas y los píxeles blancos que representan los valles [28].
2. Como segundo paso adelgazamos la imagen. En el adelgazamiento de la imagen de huellas dactilares, se toma la imagen binarizada y se va erosionando hasta que queda a la anchura de un píxel, del cual se van a extraer las minucias [29].
3. Como tercer paso extraemos las minucias de la imagen adelgazada. Cabe mencionar que las minucias que se extraerán son los fines de cresta y las bifurcaciones. Para analizar el píxel si se trata de una minucia creamos una matriz de 3x3, tomando como centro el píxel analizado.
4. Como cuarto paso creamos un mapa de densidades con las minucias extraídas. Con este mapa ubicaremos las zonas en donde existe mayor concentración de minucias, por lo que en una huella dactilar alterada encontraremos un mayor número de minucias que en una huella dactilar no alterada.
5. El paso siguiente es guardar la matriz calculada como una imagen para que sea procesada en el clasificador SVM.

(12) Métricas y Evaluación del desempeño.

El rendimiento de diferentes algoritmos se evalúa por su capacidad para

detectar y clasificar correctamente una huella dactilar alterada mediante su precisión, exhaustividad y exactitud. De acuerdo al ISO /IEC 30107-3 “Detección de ataques de presentación biométrica - Parte 3: Pruebas e informes”, Por analogía con el rendimiento biométrico, las métricas especificadas para la detección de alteraciones vienen dadas por el falso positivo, "tasa de error de clasificación de la presentación normal" (NPCER), que se define como proporción de presentaciones normales clasificadas incorrectamente como presentaciones de ataque, y por otro lado por el falso negativo, "tasa de error de clasificación de la presentación de ataque" (APCER), que se define como proporción de presentaciones de ataque clasificadas incorrectamente como presentaciones normales [12], [16], [30].

(13) MICROSOFT VISUAL BASIC.

Es un lenguaje de programación de computadoras. Visual Basic además de ser referido como un lenguaje de programación, a menudo también se lo conoce como una herramienta para producir programas de aplicación basados en Windows [31]. Su propietario es Microsoft y quien lo desarrollo fue Alan Cooper, la última versión VB6 salió de soporte en el 2008, pero aun es compatible con sistemas operativos de Windows de 32bits y 64bits(WOW). En este proyecto se utiliza para tener acceso al SDK y por ende a los eventos del lector óptico “U.are.U 4500”

(14) PYTHON.

Python es un poderoso lenguaje de programación, que fue escrito por un programador llamado Guido van Rossum. Donde la reducción de la línea de lectura de código es un aspecto importante. Los intérpretes de Python están disponibles para muchos sistemas operativos [32]. Python es de código abierto y su última versión a Junio 2022, es la 3.10.5. en este proyecto se utiliza las siguientes librerías o módulos:

- Librería OpenCV-4.5.4.60:, para tener acceso las imágenes mediante su función `imread(rutaimg)`.
- Módulo OS: Con el cual se tiene acceso a las funciones del Sistema Operativo, que mediante su función `os.path.join`, se accede a las imágenes individuales.
- Módulo Math: Para tener acceso a las funciones matemáticas.
- Módulo Time: Para tener acceso a las funciones del tiempo, con su función `time()`, al segundo actual de ejecución, tomando como base el 01/01/1970(UTC 00:00:00) cuando `gmtime(X)`, $X=0$.
- Librería numba-0.54.1: Compilador inteligente para acelerar código de Python. El decorador básico `@numba.jit` para acelerar el código numérico de las funciones y `numba.jit(int16[:,:](int16[:,:]))`, cuando deseamos indicar el tipo de variable de ingreso y salida de las funciones.
- Librería scikit-learn-0.24.2: Para el entrenamiento y pruebas de las imágenes.
- Librería numpy-1.20.3: Para crear las matrices de utilizadas.
- Librería matplotlib-3.4.3: para visualizar los gráficos de las huellas dactilares procesadas y la matriz de confusión.

(15) SVM (MÁQUINA VECTORIAL DE SOPORTE).

Conjuntos de técnicas relacionadas para el aprendizaje supervisado, usándose en la clasificación y la regresión. Una propiedad especial de SVM es que minimiza simultáneamente el error de clasificación empírica y maximiza el margen geométrico. Hay muchas funciones del kernel en SVM, para propósitos generales, hay algunas funciones del kernel típicas como son Lineal, Polinomial, RBF y Sigmoide [33].

II. MATERIALES Y MÉTODO

2.1. Tipo y Diseño de Investigación

2.1.1. Tipo de investigación:

El desarrollo del presente fue cuantitativo de tipo básica, en donde se desarrolló de manera satisfactoria el objetivo general planteado, y con la recolección de datos en un contexto científico, se pudo cuantificar y comprobar la hipótesis.

2.1.2. Diseño de Investigación.

Fue Cuasi-Experimental con post prueba y aplicándose a tres tipos de huellas dactilares alteradas más comunes (obliteración, corte z y rotación central), y a huellas inalteradas. Se analizó las características de las imágenes de huellas dactilares en la cual se manipuló la variable dependiente con respecto a la variable independiente y observándose sus resultados.

2.2. Variables, Operacionalización

2.2.1. Variables.

Variable Independiente.

“Algoritmos de clasificación automática”

Variable Dependiente.

“Detección huellas dactilares alteradas”

2.2.2. Operacionalización.

Tabla I
Operacionalización de la variable Independiente y Dependiente.

Variable de estudio	Definición conceptual	Definición operacional	Dimensión	Indicadores	Ítems	Instrumento	Valores finales	Tipo de variable	Escala de medición
Independiente “Algoritmos”	Conjunto de instrucciones para realizar una tarea, de forma automática	Porcentaje de aciertos.	Algoritmos de clasificación automática	Exactitud	$E = \frac{VP + VN}{VP + FP + VN + FN}$	Ficha registro de procesamiento y Matriz de confusión	Decimal Porcentaje	numérica	$\geq 90\%$ [7]
				Exhaustividad	$R = \frac{VP}{VP + FN}$		Decimal Porcentaje	numérica	$\geq 96\%$ [7]
				Precisión	$P = \frac{VP}{VP + FP}$		Decimal Porcentaje	numérica	$\geq 86.5\%$ [7]
Dependiente “Detección huellas dactilares”	Características que se encuentran en las yemas de los dedos	Tiempo y Porcentaje de errores	Detección huellas dactilares alteradas	Tiempo	$T = TF - TI$		Decimal Segundos	numérica	≤ 7 segundos [11]
				APCER	$APCER = \frac{FP}{FP + VN}$		Decimal Porcentaje	numérica	$\leq 14.41\%$ [12]
				NPCER	$NPCER = \frac{FN}{VP + FN}$		Decimal Porcentaje	numérica	$\leq 27.62\%$ [12]

Fuente: Elaboración propia.

Descripción de las variables utilizadas en las fórmulas del indicador:

VP = número de imágenes de huellas dactilares alterada correctamente identificadas;

FN = número de imágenes de huellas dactilares identificadas erróneamente como sin alterar;

FP = número de imágenes de huellas dactilares identificadas erróneamente como alteradas;

VN = número de imágenes de huellas dactilares sin manipular correctamente identificadas

TF = Representa el tiempo en que culmina el procesamiento

TI = Representa el inicio del procesamiento.

2.3. Población de estudio, muestra, muestreo y criterios de selección

2.3.1. Población.

Está determinada por una cantidad de 04 algoritmos de clasificación automática para la detección de huellas dactilares alteradas, los mismos que se muestran a continuación:

- Algoritmos Basados extracción de minucias [7], [8], [10], [11] , [12] [13], [16] , [17], [18], [21].
- Algoritmos Basados en el Campo de orientación [6],[9], [11], [12],. [14], [16] , [21].
- Algoritmos Basados en la Frecuencia de Cresta [14] , [15] , [19].
- Algoritmos Basados en Neuro Fuzzy [20].

2.3.2. Muestra.

Se estableció que la muestra va ser con la cantidad determinada de 02 Algoritmos de clasificación automática para la Detección de Huellas Dactilares Alteradas.

Los algoritmos seleccionados según el muestreo y los criterios de selección, son el algoritmo basado en el campo de orientación y el basado en la extracción de minucias.

2.3.3. Muestreo.

El muestreo fue no probabilístico y por conveniencia, seleccionado los dos algoritmos con mejor rendimiento en las investigaciones de los últimos 5 años, como se muestran en el cuadro comparativo de algoritmos para la detección de huellas dactilares alteradas Tabla V, del numeral 3.3 Aporte

práctico.

2.3.4. Criterios de selección.

Se delimito la población de acuerdo a las siguientes características que permitieron excluir a las investigaciones que no la cumplan:

- Mejor rendimiento en los últimos 5 años.
- Que el tema de la investigación contenga la cadena: ("Algorithm" OR "Algorithms" OR "method" OR "model") AND ("Alter" OR "Altered" OR "Alterations") AND ("Fingerprint") AND ("Detection" OR "Recognition" OR "Analysis" OR "Verification" OR "Matching").
- Que sea investigación pública de acceso libre.
- Repositorios científicos como IEEE Explore, Science Direct, Scopus, Acm y en las revistas científicas reconocidas.

2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad

Tabla II

Técnicas e instrumentos de recolección de datos, validez y confiabilidad.

Técnica	USO	Instrumento de recolección de datos, validez y confiabilidad.
Análisis de resultados.	Con la matriz de fusión se ha calculado los indicadores (Exactitud, Exhaustividad, Precisión, APCER, NPCER), obtenido el desempeño de los algoritmos. Para generar la matriz de confusión se utilizó las imágenes procesadas de los algoritmos.	Matriz de confusión
Fichaje	Mediante la técnica de fichaje se registró el tiempo de procesamiento de los algoritmos de	Ficha registro de procesamiento

detección de huellas dactilares alteradas, el cual permitió medir los tiempos y cantidades de datos que se procesan.

Fuente: Elaboración propia.

2.5. Procedimiento de análisis de datos

Los datos obtenidos se organizaron en registros de procesamiento y matriz de confusión para cada uno de los indicadores y algoritmos seleccionados la descripción de los indicadores es el siguiente:

2.5.1. Exactitud.

Acerca la predicción al valor exacto. Para las métricas se valor deseado debe ser 100% y para esta investigación el valor esperado es 90% [7].

$$E = \frac{VP + VN}{VP + FP + VN + FN}$$

VP = Verdaderos Positivos, VN = Verdaderos Negativos, FP = Falsos Positivos y FN = Falsos Negativos; en la matriz de confusión. [7]

2.5.2. Exhaustividad.

Muestra la tasa de verdaderos positivos detectados correctamente para cada uno de los algoritmos. También es conocida como sensibilidad o tasa de verdaderos positivos. Para las métricas se valor deseado debe ser 100% y para esta investigación el valor esperado es 96% [7].

$$R = \frac{VP}{VP + FN}$$

VP = Verdaderos Positivos y FN = Falsos Negativos; en la matriz de

confusión [7].

2.5.3. Precisión.

Muestra la tasa de aciertos para cada uno de los algoritmos. Esta métrica es ideal para la clasificación, pues me muestra cuantos aciertos ha tenido el algoritmo. También es conocida como valor predictivo positivo. Para las métricas se valor deseado debe ser 100% y para esta investigación el valor esperado es 86.5% [7].

$$P = \frac{VP}{VP + FP}$$

VP = Verdaderos Positivos y FP = Falsos Positivos; en la matriz de confusión [7].

2.5.4. Tiempo de procesamiento.

Calcula el tiempo que demora el procesamiento de los datos, para cada uno de los algoritmos. Para las métricas, esta investigación el valor esperado es 7 segundos [11].

$$T = TF - TI$$

Donde, TF representa el tiempo en que culmina el procesamiento y TI representa el inicio del procesamiento.

2.5.5. NPCER.

Tasa de imágenes falsamente aceptadas como alteradas. También es la Tasa de error de clasificación de presentación normal (NPCER). Para las métricas se valor deseado debe ser 0% y para esta investigación el valor

esperado es 14.41% [12].

$$NPCER = \frac{FN}{VP + FN}$$

Donde, FN representa los Falsos Negativos y VP representa los Verdaderos positivos, de la matriz de confusión. [30]

2.5.6. APCER.

Tasa de imágenes alteradas falsamente rechazadas. También es la Tasa de error de clasificación de presentación de ataques (APCER). Para las métricas se valor deseado debe ser 0% y para esta investigación el valor esperado es 27.62% [12].

$$APCER = \frac{FP}{FP + VN}$$

Donde, FP representa los Falsos Positivos y VN representa los Verdaderos Negativos, de la matriz de confusión. [30].

2.6. Criterios éticos

Tabla III

Criterios Éticos.

Criterios	Características éticas del criterio
Medio ambiente	En el desarrollo del presente proyecto se cuidó el medio ambiente, al no utilizar papel o tinta, todo se manejó de manera electrónica.
Confidencialidad	Se protegió la identidad de los sujetos que portaron sus huellas dactilares.
Objetividad	El estado actual del problema, se analizó basándose en el análisis técnico.
Originalidad	Citándose las investigaciones que fueron fuente, que aportaron a la presente investigación, demostrándose que no existe plagio intelectual.

Fuente: Elaboración propia.

III. RESULTADOS Y DISCUSIÓN

3.1. Resultados

En el entrenamiento y pruebas de los algoritmos seleccionados se ha utilizado dos bases de datos. Una conteniendo huellas dactilares sin alterar (imágenes de 100 dedos de 10 individuos) y la otra con imágenes de las alteraciones más comunes (obliteración, corte Z y rotación central), creadas sintéticamente (300 huellas dactilares), como podemos observar en la Tabla IV.

Tabla IV

Base de datos utilizadas en la investigación.

Base de datos	Real	Obliterada	Corte Z	Rotación Central	Total
Sin Alterar	100				100
Alterada		100	100	100	300
Total	100	100	100	100	400

Nota: La información mostrada en la tabla se obtiene de la base de datos física de la investigación. Fuente: Elaboración Propia.

Los algoritmos fueron implementados utilizando Python y los resultados fueron evaluados utilizando SVM en Python. Los algoritmos evaluados fueron el análisis de extracción de minucias (EM) y Campo de Orientación (CO). Para el entrenamiento se utilizó el 50% y pruebas 50% de las dos bases de datos (alterada y sin alterar, para cada uno del tipo de alteración).

El rendimiento de los algoritmos se evaluó por su capacidad para detectar y clasificar correctamente una huella dactilar alterada mediante su precisión, exhaustividad y exactitud,

los mismos que fueron calculados a partir de los valores de sus respectivas matrices de confusión. A continuación, se muestra los resultados de los indicadores anteriormente mencionados:

La Exactitud acerca la predicción al valor exacto del total de huellas dactilares detectadas alteradas y sin alterar. En la Fig. 11, se puede observar que mejor exactitud tiene el algoritmo de Extracción de Minucias (EM), con un porcentaje de 83% en la detección de las huellas dactilares alteradas tipo obliteradas, seguido del algoritmo del campo de orientación (CO), con un 77% en las huellas dactilares alteradas tipo obliteradas. También se puede observar que con la alteración tipo obliteración ambos algoritmos tienen mejor rendimiento (EM=83% y CO=77%) y con el que tienen menor rendimiento es el tipo de alteración Corte Z (EM=54% y CO=52%).

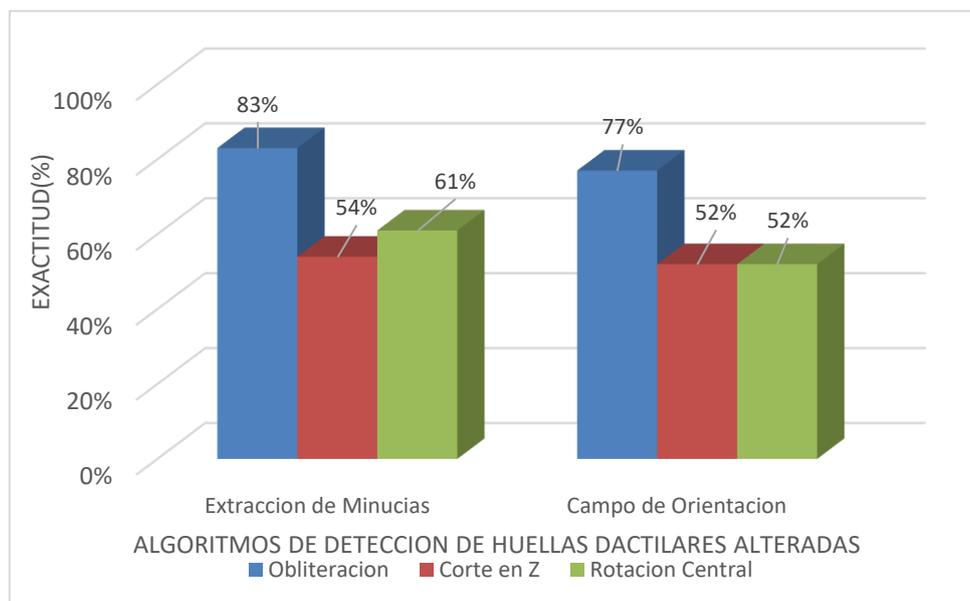


Fig. 11. Comparación del nivel de predicción de la Exactitud para los algoritmos en los diferentes tipos de alteración. Fuente: Elaboración propia.

La exhaustividad muestra la tasa de verdaderos positivos detectados correctamente para cada uno de los algoritmos, pues muestra cuantas huellas dactilares alteradas han sido detectadas. También es conocida como sensibilidad o tasa de verdaderos positivos. En la

Fig. 12, se puede observar que mejor exhaustividad tiene el algoritmo de Extracción de Minucias (EM), con un porcentaje de 94% en la detección de las huellas dactilares alteradas tipo obliteradas, seguido del algoritmo del campo de orientación (CO), con un 66% en las huellas dactilares alteradas tipo Corte Z. También se puede observar que, con la alteración tipo Rotación Central, ambos algoritmos tienen menor rendimiento (EM=64% y CO=52%).

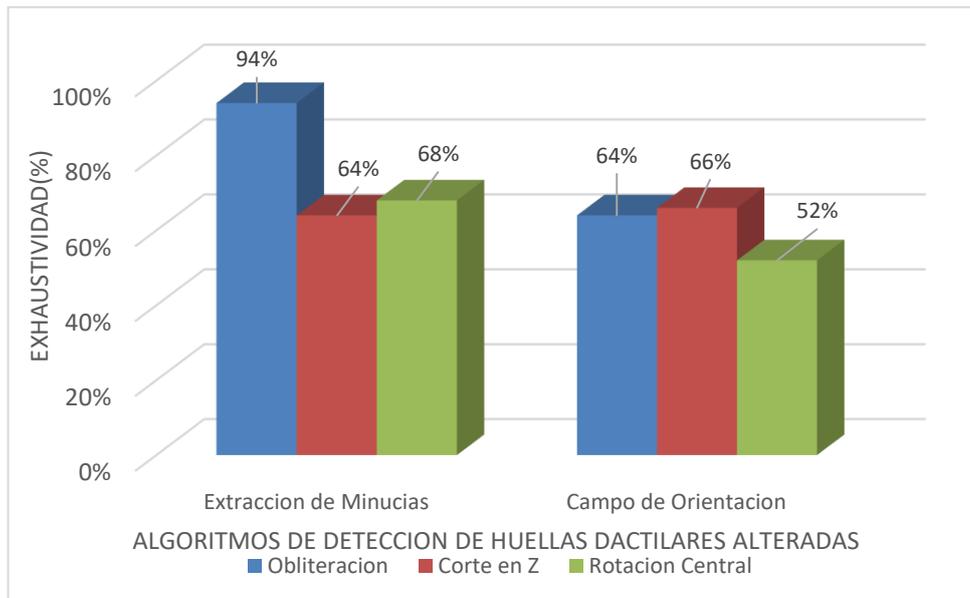


Fig. 12. Comparación del nivel de predicción de la Exhaustividad para los algoritmos en los diferentes tipos de alteración. Fuente: Elaboración propia.

La Precisión muestra la tasa de aciertos para cada uno de los algoritmos. Esta métrica es ideal para la clasificación, pues muestra cuantas huellas dactilares alteradas han sido detectadas. También es conocida como valor predictivo positivo. En la Fig. 13, se puede observar que mejor precisión tiene el algoritmo de campo de orientación (CO), con un porcentaje de 87% en la detección de las huellas dactilares alteradas tipo obliteradas, seguido del algoritmo del Extracción de Minucias (EM), con un 78% en las huellas dactilares alteradas tipo obliteración. También se puede observar que con la alteración tipo obliteración ambos algoritmos tienen mejor rendimiento (EM=78% y CO=87%) y con el que tienen menor rendimiento es el tipo de alteración Corte Z (EM=54% y CO=52%).

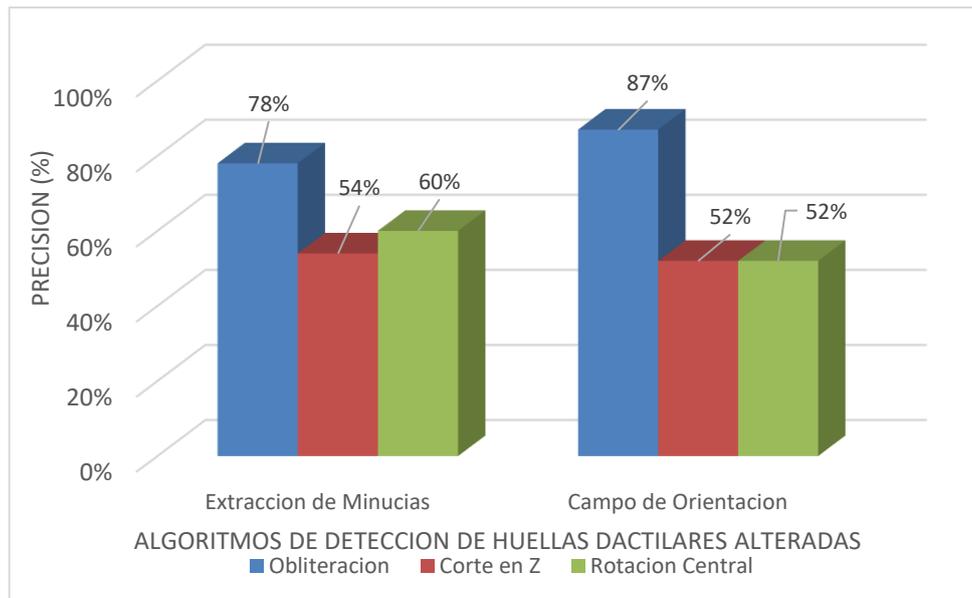


Fig. 13. Comparación del nivel de predicción de la Precisión para los algoritmos en los diferentes tipos de alteración. Fuente: Elaboración propia.

Otra forma de medir el rendimiento de algoritmos es el tiempo de procesamiento. En la Fig. 14, podemos observar la comparación de los promedios del tiempo de procesamiento de ambos algoritmos por cada uno de los tipos de alteración, en donde el algoritmo de campo orientación (CO), tiene el mejor tiempo de procesamiento en el tipo de alteración tipo Corte Z con 0.230 seg. y el peor tiempo de procesamiento en el tipo de alteración rotación central con 0.247 seg. y el algoritmo de extracción de minucias (EM), tiene el mejor tiempo de procesamiento en el tipo de alteración tipo rotación central con 0.564 seg. y el peor tiempo de procesamiento en el tipo de alteración corte Z con 0.606 seg.

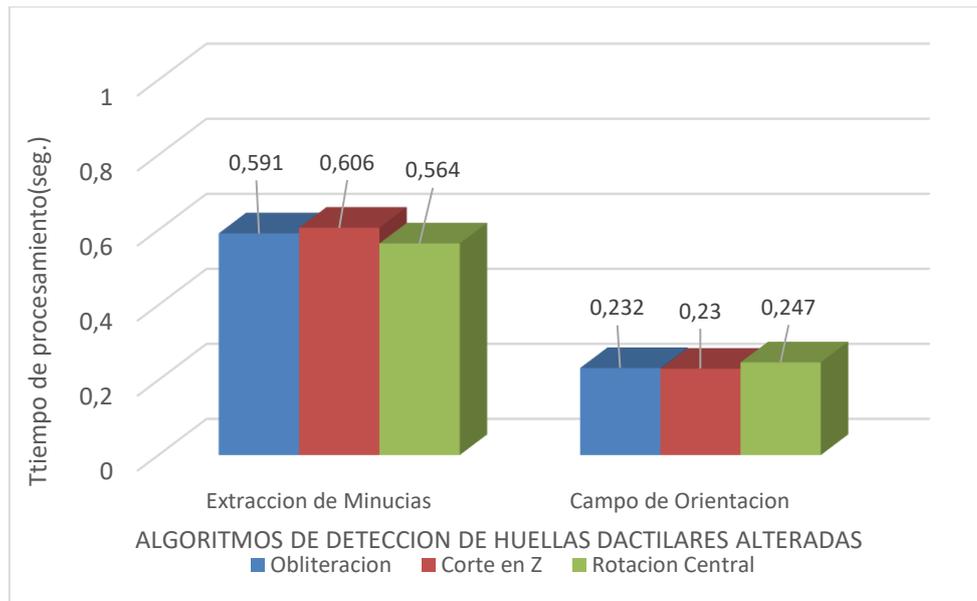


Fig. 14. Comparación del tiempo de procesamiento en segundos, de los algoritmos en los diferentes tipos de alteración. Fuente: Elaboración propia.

La precisión para Detección huellas dactilares alteradas de los algoritmos, se informan en términos de porcentaje de errores para clasificar la presentación normal (NPCER), por el porcentaje en errores al clasificar la presentación de ataques, que se define como “proporción de ataques de presentación clasificados incorrectamente como presentaciones normales” (APCER) [12].

La métrica APCER, Tasa de imágenes alteradas falsamente rechazadas. También es la Tasa de error de clasificación de presentación de ataques (APCER). Esta métrica muestra cuantas huellas dactilares alteradas han sido detectadas como sin alterar. En la Fig. 15, se puede observar que la mejor detección lo tiene el algoritmo de campo de orientación (CO), con un porcentaje de 10% en la detección de las huellas dactilares alteradas tipo obliteradas, seguido del algoritmo del Extracción de Minucias (EM), con un 28% en las huellas dactilares alteradas tipo obliteración. También se puede observar que con la alteración tipo obliteración ambos algoritmos tienen mejor rendimiento (EM=28% y CO=10%) y con el que tienen peor rendimiento es el tipo de alteración Corte Z (EM=56% y CO=62%).

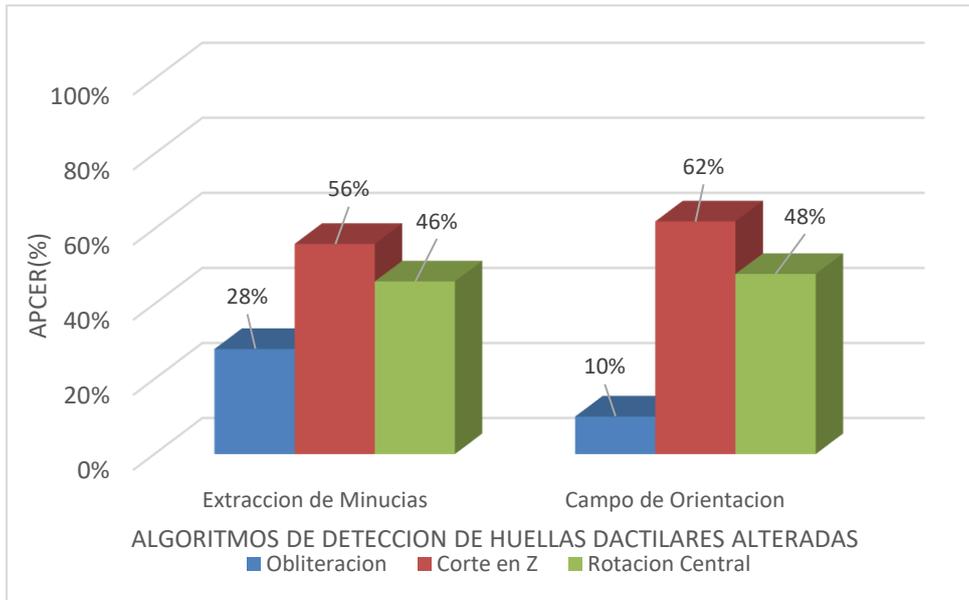


Fig. 15. Comparación del nivel de predicción de error de clasificación APCER, para los algoritmos en los diferentes tipos de alteración. Fuente: Elaboración propia.

La métrica NPCER, Tasa de imágenes falsamente aceptadas como alteradas. También es la Tasa de error de clasificación de presentación normal (NPCER). Las tasas bajas del NPCER es un valor deseable para los algoritmos de detección de huellas dactilares alteradas, porque mientras más alta es la tasa, implicaría mayor inspección manual de huellas dactilares, por parte de expertos; lo que sería más costoso y más lento tránsito en los controles fronterizos. En la Fig. 16, se puede observar que la mejor detección tiene el algoritmo de Extracción de Minucias (EM), con un porcentaje de 6% en la detección de las huellas dactilares alteradas tipo obliteradas, seguido del algoritmo del campo de orientación (CO), con un 34% en las huellas dactilares alteradas tipo Corte Z. También se puede observar que, con la alteración tipo Rotación Central, ambos algoritmos tienen peor rendimiento (EM=36% y CO=48%).

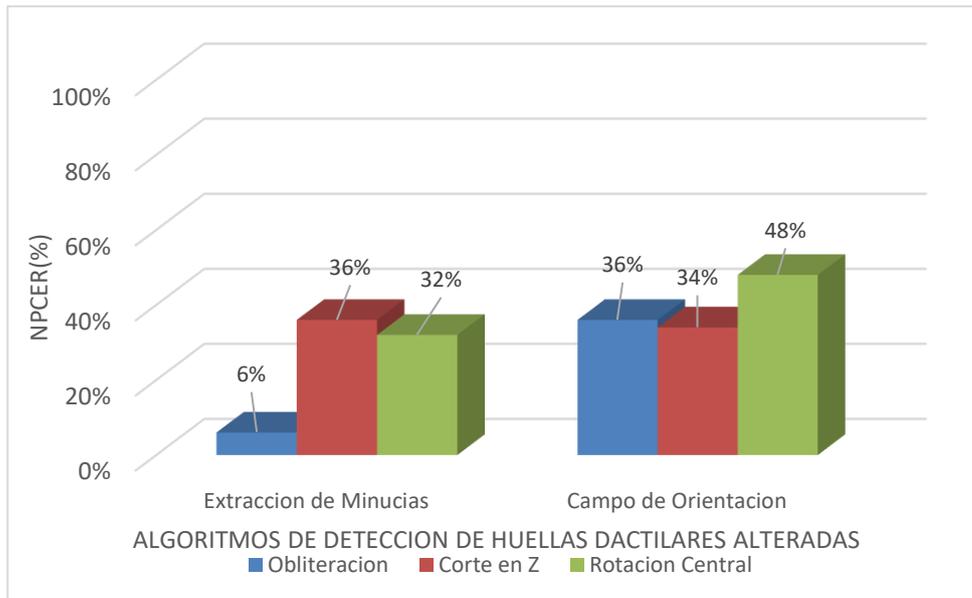


Fig. 16. Comparación del nivel de predicción de la detección de error de clasificación NPCER, para los algoritmos en los diferentes tipos de alteración. Fuente: Creación propia.

3.2. Discusión

De las pruebas realizadas a los algoritmos de clasificación automática seleccionados en esta investigación, nos permite observar que los mejores rendimientos para el algoritmo basado en la extracción de minucias son: exactitud de 83%, exhaustividad de 94%, precisión de 78%, APCER de 28% y NPCER de 6% en la alteración tipo obliteración; en el tiempo de procesamiento de 0.564 seg, en la alteración tipo rotación central; y los mejores resultados para el algoritmo basado en el campo de orientación son: exactitud de 77%, precisión de 87% y APCER de 10% en la alteración tipo obliteración y exhaustividad de 66%, tiempo de procesamiento de 0.230 seg y NPCER de 34% en la alteración tipo corte Z. De lo anteriormente mencionado se desprende que el algoritmo basado en el campo de orientación tiene mejores rendimientos para clasificar huellas dactilares y tener la tasa más baja, de error de clasificación de APCER (imágenes alteradas falsamente rechazadas) y el algoritmo basado en la extracción de minucias tiene mejores rendimientos para predecir si una huella dactilar es alterada o no alterada y también tener la tasa de error más baja, el error de clasificación NPCER (imágenes falsamente aceptadas como alteradas). También se puede

mencionar que ambos algoritmos tienen mejores resultados para detectar la alteración tipo obliteración.

También, permiten comparar con otras investigaciones para los mismos algoritmos, como es el caso de [7], en su investigación utilizando los algoritmos de extracción de minucias y el campo de orientación obtuvo resultados similares como son precisión de 86.15% , exhaustividad de 96.5% y Exactitud de 90.5% y el método propuesto en esta investigación se obtuvo resultados similares de precisión de 87% , exhaustividad de 94% y Exactitud de 83%; siendo la diferencia más resaltante, que Shehu, James, & Palade, incluyeron después del mejoramiento de la imagen y antes de la extracción de minucias el operador LBP (patrones binarios locales) y un kernel de Análisis de Componentes Principales (PCA) para hacer una preclasificación.

Con respecto al tiempo de procesamiento en la investigación de Patil & Phulpagar [11], utilizando los algoritmos de extracción de minucias y el campo de orientación obtuvo 7 segundos, en la detección de la alteración y el método propuesto en esta investigación se obtuvo mejores resultados de 0.230 segundos; siendo la diferencia más resaltante, que Patil & Phulpagar, utilizaron para el procesamiento imágenes de mayor tamaño 640x480 pixeles; mientras que en el método propuesto las imágenes fueron 250x250 pixeles, el cual representa el área de la imagen de la huella dactilar de mayor información (crestas y valles).

En la investigación de Haraksim, y otros [12], utilizando los algoritmos de extracción de minucias y el campo de orientación (MOA-OFA), obtuvo resultados de APCER de 27.62% y NPCER de 14.41% y el método propuesto en esta investigación obtuvo mejores resultados de APCER =10% y NPCER = 6%; siendo la diferencia más resaltante para obtener mejores rendimientos que [12], es que, él no utilizó algoritmos para el mejoramiento de imagen.

Las pruebas realizadas a los algoritmos seleccionados en esta investigación, nos permite observar que, el método propuesto, es eficiente para detectar huellas dactilares alteradas y también para la detección de ataques de presentación biométrica, obteniendo tasas bajas en NPCER y APCER (6% y 10% respectivamente), implicando una menor inspección manual de huellas dactilares, por parte de expertos en los controles fronterizos.

3.3. Aporte de la investigación

Para el desarrollo de los objetivos específicos se ha seguido un método, es cual es mostrado en la Fig. 17.

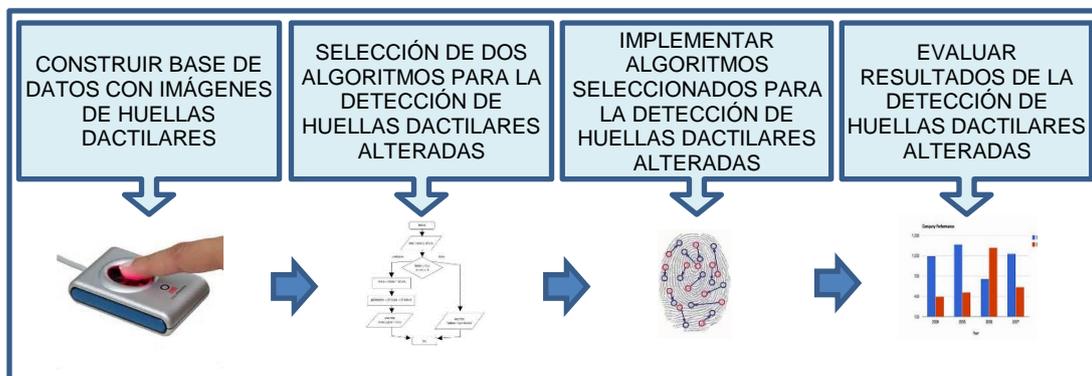


Fig. 17. Método de desarrollo de objetivos específicos. Fuente: Elaboración propia.

A. CONSTRUIR BASE DE DATOS CON IMÁGENES DE HUELLAS DACTILARES.

En esta investigación se utilizaron 02 bases de datos, una con imágenes de huella dactilar sin alterar y otra con imágenes de huellas dactilares alteradas.

Para construir la base de datos de Imágenes sin alterar, se escaneó los dedos de las manos (yemas de los 10 dedos), de 10 individuos voluntarios, del Distrito de Chongoyape, Provincia de Chiclayo, Departamento de Lambayeque; haciendo un total de 100 imágenes de huellas dactilares. Para el escaneo de las huellas dactilares se utilizó el lector óptico “U.are.U 4500”, marca de DigitalPersona, el que se muestra en la Fig. 18, siendo sus especificaciones las siguiente:

- Resolución de captura : 512dpi (Píxeles promedio)
- Dimensiones de captura : 14.6mm x 18.1 mm (ancho x largo);
- Escala de grises : 8bits (256 nivel grises);
- Dimensiones del equipo : 65mmx36mmx15.56mm.
- U.S.B : 1.0/1.1/2.0
- Voltaje : 5.0V
- Norma :WHQL/MIC/CE/FCC_Class_B/USB /ICES
- Peso : 105gr.

Para escanear huellas dactilares se tuvieron las siguientes consideraciones [22], como se muestra en la Fig. 19:

- Verificar que los dedos están limpios y secos.
- Se comienza por el dedo pulgar (mano derecha), se termina por el dedo meñique (mano izquierda).
- Se colocar la yema del ultimo falange del dedo en el área de escaneo.
- Verificar que se hayan tomado las huellas dactilares de los 10 dedos.



Fig. 18. Imagen del Lector óptico “U.are.U 4500”, marca de DigitalPersona, utilizado para el escaneo de huellas dactilares. Fuente: Elaboración Propia

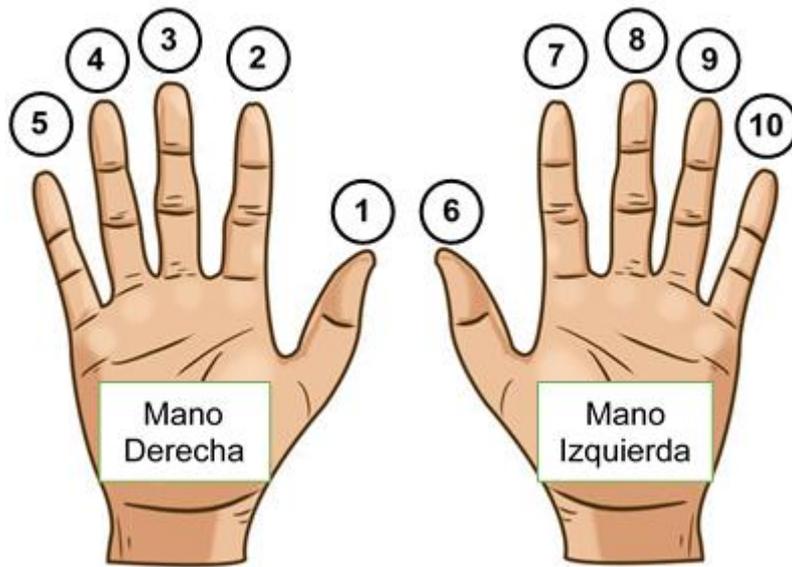


Fig. 19. Orden para la toma de huellas dactilares. Fuente: Elaboración Propia

Para la captura se utilizó la aplicación del dispositivo “U.are.U 4500” y agregué un módulo para redimensionar la huella dactilar capturada.

La imagen capturada por el escáner de 500x550 píxeles, es redimensionada a 250x250 píxeles, porque la imagen capturada por el escáner, tiene áreas que no serán procesadas; porque tienen mucho ruido o no tienen la información a procesar (crestas y valles de la huella dactilar), como se podemos ver en la Fig. 20.

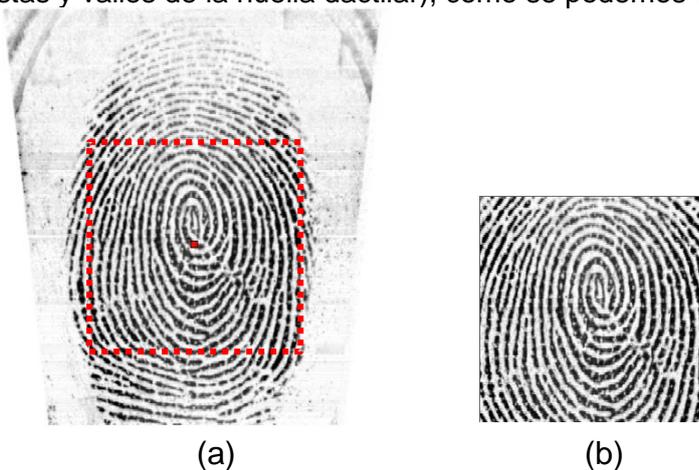


Fig. 20. Redimensionamiento de huella dactilar, a) Imagen capturada por el escáner (500x550 píxeles), b) imagen redimensionada (250x250 píxeles). Fuente: Elaboración Propia

Para redimensionar la imagen capturada por el escáner (500x550 pixeles), primero almacenamos en una matriz la imagen capturada por el escáner, luego calculamos el centro de mayor información (crestas y valles de la huella dactilar), mediante un promedio ponderado por cada coordenada X y Y. El promedio ponderado devuelve las coordenadas donde se centra la mayor cantidad de pixeles que representan las crestas (centro de carga). Con el centro de carga ubicado, creamos otra matriz de 250x250, el cual tiene como centro, el centro de carga calculado. Luego esta matriz es guardada como una imagen de mapa de bits.

El siguiente es el extracto de código en vb6 para redimensionar la imagen:

```
For py = 1 To M
  For px = 1 To N
    promcolor=CByte((CInt(pxls(pxR, px, py))+pxls(pxG, px, py)+pxls(pxB,
px, py)) \ 3)
    vlong = 255 - CLng(promcolor)
    sX = sX + vlong * px
    sY = sY + vlong * py
    sXY = sXY + vlong
  Next px
Next py
nX = CInt(sX / sXY)
nY = CInt(sY / sXY)

For py = yini To yfin
  For px = xini To xfin
    promcolor = CByte((CInt(pxls(pxR, px, py)) + pxls(pxG, px, py) + _
pxls(pxB, px, py)) \ 3)
```

```
'Debug.Print px - xini, py - yini

mxEnrollGray(px - xini + 1, py - yini + 1) = promcolor

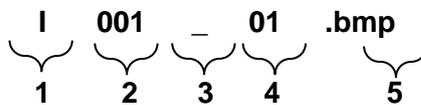
Next px
Next py
```

Para guardar las imágenes de huellas dactilares se ha considerado la siguiente nomenclatura:

Nombre y extensión del archivo:

I001_01.bmp

Nomenclatura:



En donde:

- 1) La letra "I" que representa la imagen.
- 2) Tres dígitos enteros que representan al individuo del 001-010.
- 3) Separador.
- 4) Dos dígitos enteros que representan al dedo.
- 5) La extensión del archivo tipo mapa de bits.

Las huellas dactilares alteradas se generaron sintéticamente a partir de la base de datos sin alterar, mediante la herramienta STRANGE de Papi, Ferrara, Maltoni, & Anthonioz [13]. Esta herramienta genera las tres alteraciones más frecuentes como son la obliteración, la rotación central (para imitación) y el corte en Z (para distorsión).

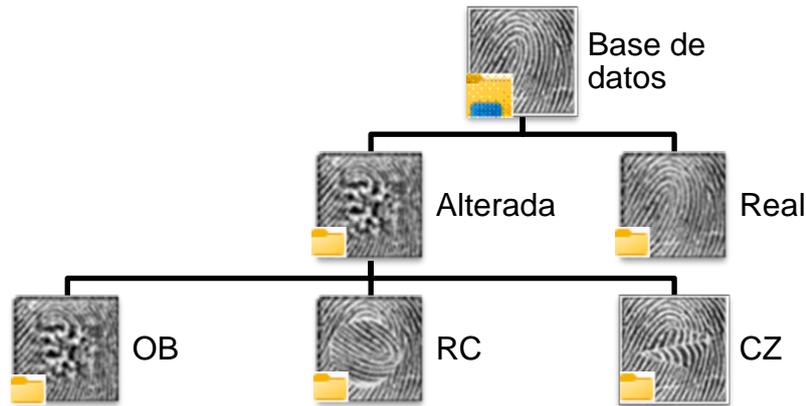


Fig. 21. Estructura de la Base de datos construida.

Fuente: Elaboración Propia.

B. SELECCIONAR DOS ALGORITMOS DE CLASIFICACION AUTOMATICA PARA LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS.

Existen varios algoritmos de clasificación automática para detectar huellas dactilares alteradas, para poder seleccionar los algoritmos que se desarrollaron en este trabajo, se realizó una búsqueda de investigaciones publicadas de acceso libre, en repositorios científicos como IEEE Explore, Science Direct, Scopus, Acm y en las revistas científicas ijarcse, iosrjournals, bonfring, ictactjournals, irjet. Se aplicó la siguiente cadena de búsqueda:

("Algorithm" OR "Algorithms" OR "method" OR "model") AND ("Alter" OR "Altered" OR "Alterations") AND ("Fingerprint") AND ("Detection" OR "Recognition" OR "Analysis" OR "Verification" OR "Matching")

Con la cadena de búsqueda se obtuvo investigaciones con las cuales se realizó un cuadro comparativo de algoritmos usados para la detección de huellas dactilares alteradas, el cual se observa en la Tabla V. Siendo seleccionados los dos

algoritmos con mejor rendimiento en las investigaciones de los últimos 5 años: El algoritmo basado en el campo de orientación y el basado en la extracción de minucias.

Tabla V

Cuadro comparativo de algoritmos de clasificación automática para la detección de huellas dactilares alteradas del resultado de investigaciones

Investigación	Algoritmo	Rendimiento
Yang, Li, & Zhang[6]	Campo de orientación	FAR es del 10%
Shehu, James & Palade [7]	Extracción de minucias y Campo de orientación	precisión = 86.15% , recall = 96.5%. Exactitud =90.5%.
Gottschlich, Tamsy, & Huckemannz [9]	Campo de orientación	Tiempo de Ejecución 130.5, 180.0 y 230.0 segundos
Vinoth & Saravanakumar [8]	Extracción de minucias	Exactitud 94% Times 12seg
Singh, Ayub, & Jp [10]	Extracción de minucias	Puntuación promedio de coincidencia= 18%.
Patil & Phulpagar [11]	Extracción de minucias y Campo de Orientación	Tiempo de Ejecución = 7 segundos
Haraksim, y otros [12]	Extracción de minucias y Campo de Orientación	10% ERR , NPCER de 14.41% para OFA-MOA Y APCER de 27.62% para OFA-MOA).
Papi, Ferrara, Maltoni, & Anthonioz [13]	Extracción de minucias	EER 10.3% FMR 19.6%

Investigación	Algoritmo	Rendimiento
Mini & Anoop [14]	Campo de Orientación y Frecuencia de Cresta.	GAR = 90% FAR =3.33%
Mini & Anoop [15]	Frecuencia de Cresta.	GAR 84.28 % FAR in 12.85 %
Gottschlich, Mikaelyan, Olsen, Bigun, & Busch [16]	Extracción de minucias y Campo de Orientación	MDA es, para NPCER 1% el APCER 45%. OFA es, para NPCER 1% el APCER 20%.
Nagthane & Fadewar [17]	Extracción de minucias	promedio de coincidencia= 79.5 %, FP alteradas.
Mohan [18]	Extracción de minucias	Extracción de minucias : 5s; Formación de trío : 8s
Chen, Chang, Huang, He, Wang & Huang [19]	Frecuencia de Cresta.	Exactitud 98%
Latha & Manikandan [20]	Neuro Fuzzy	Ratio de falsa alarma=4% Ratio de detección verdadera=70%
Yoon, Feng, & Jain [21]	Extracción de minucias y Campo de Orientación	TFP= 2,1 % y TVP= 70,2%.

Fuente: Elaboración Propia.

C. IMPLEMENTAR ALGORITMOS DE CLASIFICACION AUTOMATICA SELECCIONADOS PARA LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS.

En la Fig. 22, se muestra la secuencia para la implementación de los algoritmos:

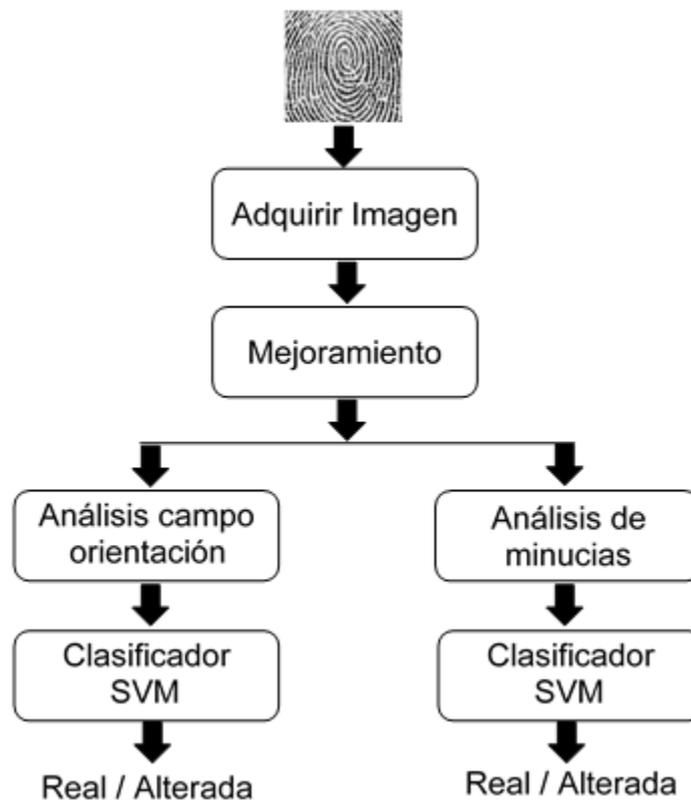


Fig. 22. Secuencia de Implementación de algoritmos de clasificación automática. Fuente: Elaboración Propia

ADQUIRIR IMAGEN DE HUELLA DACTILAR.

Las Base de datos utilizadas fueron: una de Imágenes de huellas dactilares sin alterar, se escaneó los dedos de las manos (yemas de los 10 dedos), de 10 individuos voluntarios; haciendo un total de 100 imágenes. Para el escaneo se utilizó el dispositivo "U.are.U 4500", marca DigitalPersona. Las imágenes alteradas fueron creadas sintéticamente para los tipo de alteraciones más comunes (Obliteración,

Corte Z y rotación central), 100 imágenes por cada tipo, haciendo un total de 300 imágenes alteradas. En la Fig. 23 se muestran una imagen para la huella dactilar real y una alterada.



Fig. 23. Imágenes de huellas dactilares: (a) Imagen Real y (b) Imagen Alterada.

Fuente: Elaboración Propia.

La adquisición de la huella dactilar consistió en almacenar la imagen de mapa de bits en una matriz, como se muestra en la Fig. 24, donde para cada pixel el formato de almacenamiento es $\text{pixel}(\text{Color}, X, Y) = \text{Intensidad}$.

El fragmento de código en Python para adquirir la imagen y asignarla a una matriz, es el siguiente:

```
import cv2
himg=cv2.imread(rutaimg)
```

En donde “rutaimg”, es la variable en donde se encuentra almacenada la imagen y “cv2.imread”, es la función imread de la librería OpenCV, el cual por defecto obtiene la matriz de la imagen a todo color; “himg” es la matriz en donde se almacena la imagen, que luego será utilizado por la función de escala de grises. Para que la función “imread” sea reconocida, se importa la librería OpenCV (cv2).

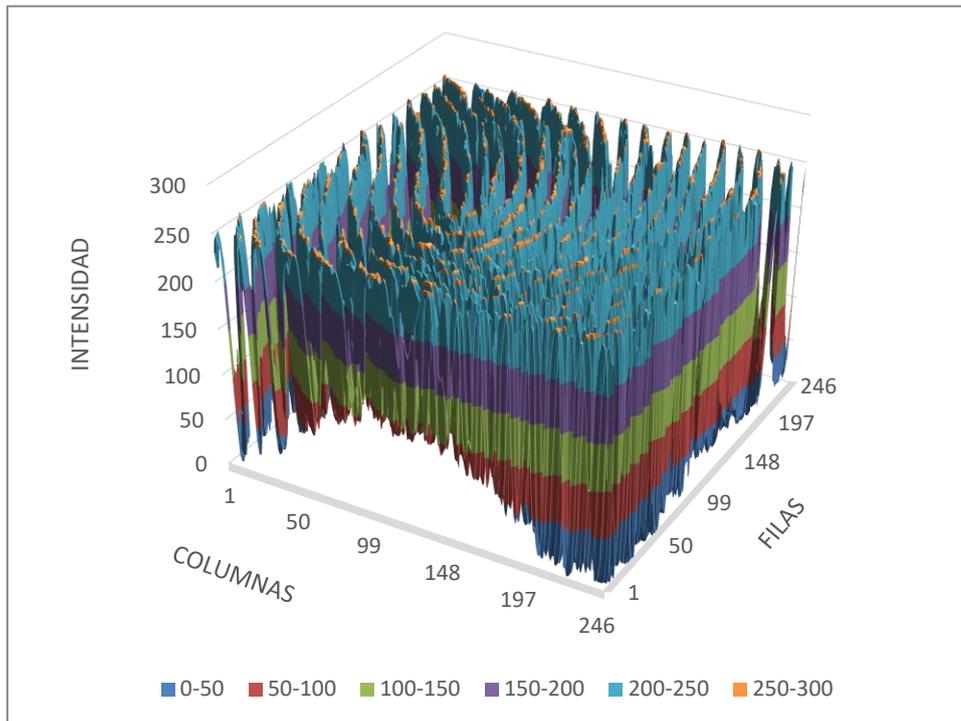


Fig. 24. Representación de una matriz de huella dactilar en un gráfico 3D. Fuente: Elaboración Propia.

MEJORAMIENTO DE IMAGEN.

El objetivo es que la imagen procesada sea mejor que la imagen original y que los algoritmos siguientes obtengan un mejor resultado.

a) Cambiar Imagen de la huella en escala de grises.

El objetivo de convertir a grises es porque en los siguientes pasos se utiliza técnicas de procesamiento de imágenes que tienen como entrada una imagen en escala a grises, ya que solo se utilizaría un solo color (promedio de rojo, verde y azul), para representar cada pixel de las imágenes, disminuyendo el tamaño de la matriz. En la conversión de escala de grises, se utilizó la Media como método, consistiendo en, que para cada pixel de la imagen se encontrará el promedio de la suma sus tres colores. [34].

Como la matriz que almacenara los valores es de tipo byte, en la cual no

permite decimales, se utilizó la siguiente función.

$$Gris = (R + G + B) \setminus 3$$

Donde R, G y B representan al color rojo, color verde y color azul del pixel, respectivamente, cuando una imagen está representada en el modelo RGB.

El fragmento de código en Python para convertir en escala de grises es siguiente:

```
for py in range(M):
    for px in range(N):
        R = int(str(matGray[px][py][0]))
        G = int(str(matGray[px][py][1]))
        B = int(str(matGray[px][py][2]))
        mZeros[px][py]= (R+G+B)//3
return mZeros
```

Donde M y N son el tamaño de matGray (columnas y filas respectivamente); matGray, es la matriz de entrada que contiene la información de la imagen y mZeros, es la matriz de salida con las intensidades convertidas a escala de grises.

En la Fig. 25 se observa el resultado al implementar el algoritmo para convertir a Escala de Grises.



Fig. 25. Resultados de la Implementación del algoritmo Escala de Gris: a) Imagen adquirida, b) Imagen escala de gris. Fuente: Elaboración Propia.

b) Normalización.

Con la normalización modificamos la escala de grises de las crestas y valles de la huella dactilar en escala de grises. De esta manera, normalizamos todos los pixeles suavizándolos y ubicándolos en un promedio deseado, como podemos ver la representación gráfica de la Fig. 26, facilitando el siguiente paso, que es el procesado de la imagen con el algoritmo de ecualización.

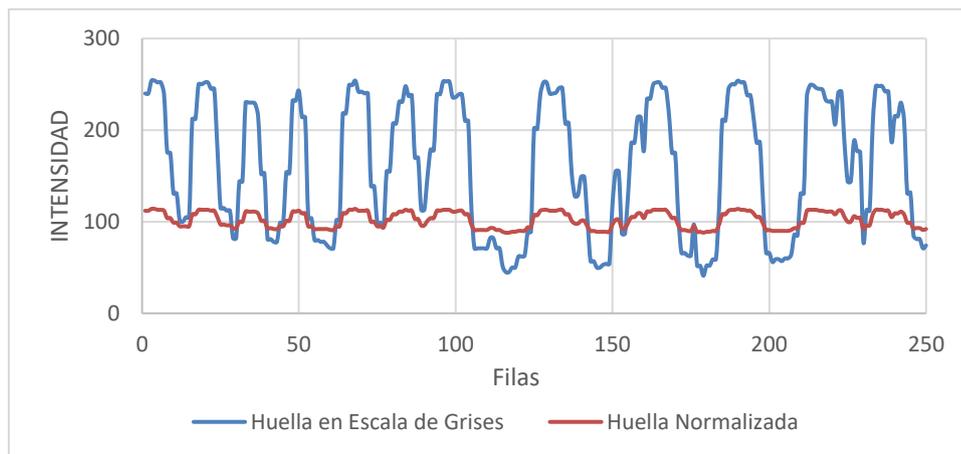


Fig. 26. Representación gráfica de la diferencia de la intensidad de una huella dactilar en escala de grises con una normalizada. Fuente: Elaboración Propia.

Teniendo en cuenta la media y la varianza del pixel Local en la imagen. Para un pixel $I(x, y)$, de la imagen se compara con la media de la imagen, el resultado de la comparación produce un valor normalizado de nivel de gris $N(x, y)$ se define por la fórmula [27]:

$$N(x, y) = \begin{cases} M_0 + \sqrt{\frac{V_0 \cdot [I(x, y) - M]^2}{V}} & , \text{si } I(x, y) > M \\ M_0 - \sqrt{\frac{V_0 \cdot [I(x, y) - M]^2}{V}} & , \text{otro caso} \end{cases}$$

donde un valor supuesto de M_0 se establece para la media deseada y un

valor supuesto de V_0 se establece para la varianza deseada. Para los cuales $M_0=100$ y $V_0=100$

El elemento $I(i,j)$ representa la intensidad de un píxel en la fila i y columna j . La media, $M(I)$, y la varianza, $V(I)$, de la imagen se definen de la siguiente manera:

$$M(I) = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N I(i,j)$$

$$V(I) = \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N [I(i,j) - M(I)]^2$$

El fragmento de código en Python para la normalización utilizado es el siguiente:

```
for py in range(M):
    for px in range(N):
        Intensidad = matGray[px][py]
        sum1 = sum1 + Intensidad

Media = abs(sum1 / NM)

for py in range(M):
    for px in range(N):
        Intensidad = matGray[px][py]
        SUM2 = SUM2 + (Intensidad - Media) ** 2

Varianza = abs(SUM2 / NM)

for py in range(M):
    for px in range(N):
```

```

Intensidad = matGray[px][py]

if Intensidad > Media:
    NewInten = int(vnMed + math.sqrt((vnMed * (Intensidad - Media)
** 2) / Varianza))
else:
    NewInten = int(vnMed - math.sqrt((vnMed * (Intensidad - Media)
** 2) / Varianza))

if NewInten > 255: NewInten = 255
if NewInten < 0 : NewInten = 0

mZeros[px][py] = NewInten

return mZeros

```

Donde M y N son el tamaño de matGray (columnas y filas respectivamente); matGray, es la matriz de entrada que contiene la información de la imagen; vnMed=100, representa la media y varianza deseada y mZeros, es la matriz de salida con las intensidades normalizadas.

En la Fig. 27, se observa el resultado al implementar del algoritmo para la normalización de huellas dactilares.

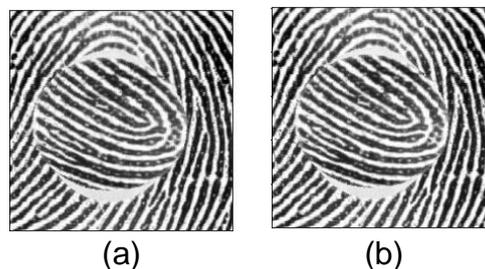


Fig. 27. (a) Imagen en escala de grises, (b) Imagen Normalizada. Fuente: Creación Propia.

c) Ecuación.

El objetivo del histograma de ecualización, es extender el contraste de una imagen dada de manera uniforme en toda la gama dinámica disponible, en este caso entre 0 y 255. Suponemos que tenemos una imagen (x) y que el rango dinámico de la intensidad (rk), varía de 0 (negro) a 255 (blanco). El objetivo de aplicarle en las huellas dactilares es aumentar los niveles de blanco uniformemente y eliminar los pixeles con pocos niveles de grises, de esta forma quedaría más visibles y fácil de trabajar las crestas. Se puede aproximar usando la probabilidad basado en el histograma p (rk) como sigue [35]:

$$probabilidad(x) = p(rk) = \frac{total\ intensidad\ rk}{total\ en\ imagen\ x}$$

A partir de esta probabilidad, entonces podemos obtener la función de densidad acumulativa (cdf) de la siguiente manera:

$$cdf = \sum_{k=0}^{k-1} p(k)$$

Donde p (k) es la probabilidad de pixel de k densidad. Para obtener el valor del píxel, sk tienen que ser multiplicado por 255 y luego se redondea al entero más cercano:

$$sk = \sum_{k=0}^{255} Entero(cdf(k) * 255)$$

El método utilizado, preserva la imagen fuente:

$$m = rk + sk$$

rk = es el pixel de la imagen fuente

sk = es el pixel de la imagen ecualizada

m = es el nuevo valor del pixel

El fragmento de código en Python para el histograma de ecualización utilizado es el siguiente:

```
NM=N*M

#Calculo Histograma
for py in range(M):
    for px in range(N):
        Intensidad = Mtrx[px][py]
        ImgHist[Intensidad-1] = ImgHist[Intensidad-1]+1

#Calculo la Probabilidad
for p in range(1,256):
    k=p-1
    probabilidad[k] = ImgHist[k] / NM
    if k > 0 :
        #Calculo la Acumulada
        acumulada[k] = acumulada[k - 1]+ probabilidad[k]
    else:
        acumulada[k] = probabilidad[k]

#Calculo la ecualización de los pixeles
ecualizador[k] = int(acumulada[k] * 255)
```

```
# se crea la matriz con el nuevo valor del pixel
for py in range( M):
    for px in range( N):
        Intensidad = Mtrx[px][py]
        NewInten = Intensidad + ecualizador[Intensidad-1]
        if NewInten >= 255 : NewInten = 255
        if NewInten < 0 : NewInten = 0
        mZeros[px][py] = NewInten
return mZeros
```

Donde M y N son el tamaño de Mtrx (columnas y filas respectivamente); Mtrx, es la matriz de entrada que contiene la información de la imagen; ImgHist, es el array con los datos del histograma calculado y mZeros, es la matriz de salida con las intensidades ecualizadas.

En la Fig. 28 se muestra el resultado de implementar el algoritmo para el histograma de ecualización de huellas dactilares.



Fig. 28. Resultados de la Implementación del algoritmo para el Histograma de ecualización de Huellas Dactilares. (a) Imagen normalizada, (b) Imagen ecualizada. Fuente: Elaboración Propia.

ANALISIS DEL CAMPO DE ORIENTACIÓN.

En la gran parte de topologías de huellas dactilares, el campo de orientación es bastante suave y uniforme, mientras que, en algunas áreas, la orientación

aparece de una manera discontinua [36], cuando hay esta discontinuidad el campo de orientación cambia sus valores bruscamente indicando una alteración o un punto singular.

En el análisis del campo de orientación de una imagen ecualizada, se sigue los siguientes pasos:

1. Primero calculamos las gradientes, para saber las zonas en la que están cambiando las crestas de la huella dactilar tanto en la dirección "X" y en la "Y", como se muestran en las imágenes (b) y (c) de la Fig. 29. Para calcular las gradientes $G_x(i, j)$ y $G_y(i, j)$ a cada pixel (i,j) , el gradiente operador es un simple Sobel 3x3.

Donde la máscara sobel X es:

$$S_x = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Y, en donde la máscara sobel Y es:

$$S_y = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}$$

En el cual se utilizó la siguiente formula:

$$G_x(i, j) = \sum_{u=i-1}^{i+1} \sum_{v=j-1}^{j+1} pixel(u, v) S_x(u, v)$$

$$G_y(i, j) = \sum_{u=i-1}^{i+1} \sum_{v=j-1}^{j+1} pixel(u, v) S_y(u, v)$$

2. Para saber los cambios brusco en las crestas de la huella dactilar, como se muestran en la imagen (d) de la Fig. 29, estimamos la orientación del pixel con respecto a un bloque de 16x16, centrado en el pixel (i , j) usando la siguiente ecuación:

$$\theta_x(i,j) = \sum_{u=i-\frac{b}{2}}^{i+\frac{b}{2}} \sum_{v=j-\frac{b}{2}}^{j+\frac{b}{2}} 2G_x(u,v)G_y(u,v)$$

$$\theta_y(i,j) = \sum_{u=i-\frac{b}{2}}^{i+\frac{b}{2}} \sum_{v=j-\frac{b}{2}}^{j+\frac{b}{2}} [G_x^2(u,v) - G_y^2(u,v)]$$

$$\theta(x,y) = \frac{\pi}{2} + (0.5 * \text{Atan2}(\theta_x(i,j), \theta_y(i,j)))$$

Donde $\theta(x, y)$ es la orientación cresta local en el bloque tomando como en pixel (i, j).

3. Dado que la orientación local de cresta varía lentamente en una vecindad local, donde no aparecen puntos singulares, la imagen necesita ser convertido en un campo de vector continuo, como se muestran en la imagen (e) de la Fig. 29, para calcular la aproximación del campo de orientación, que se define como sigue:

$$\Phi_x(x,y) = \cos(2\theta(x,y))$$

$$\Phi_y(x,y) = \sin(2\theta(x,y))$$

Donde Φ_x e Φ_y son los componentes x e y del campo vectorial.

4. Con los componentes Φ_x e Φ_y se calcula la aproximación del campo de orientación Local de cresta en (x,y) usando:

$$\Phi'(x,y) = \frac{1}{2} \text{Atan}\left(\frac{\Phi_x(x,y)}{\Phi_y(x,y)}\right)$$

5. El mapa de error muestra qué tan precisa es la aproximación a la estimación. Los cambios bruscos y las discontinuidades en el flujo de la cresta darán como resultado valores altos en el mapa de errores, como se muestran en la imagen (f) de la Fig. 29.

$$\varepsilon(x,y) = \min(|\theta(x,y) - \Phi'(x,y)|, \pi - |\theta(x,y) - \Phi'(x,y)|) / (\pi * 1/2)$$

6. El paso siguiente es guardar la matriz calculada para que sea procesada en el clasificador SVM.

El fragmento de código en Python para el análisis del campo de orientación usada es el siguiente:

```
#Cálculo la Gradiente Sobel
for py in range(M):
    for px in range( N):
        gs,mGSx, mGSy = functX.Gradiente_Sobel(Mtrx, px, py)
        ImgGx[px][py]= mGSx
        ImgGy[px][py]= mGSy

# Estimo la Orientación Local
for py in range(M):
    for px in range(N):
```

```

LO = functX.Local_Orientacion2(px, py, 16, ImgGx, ImgGy) #16

Angx[px][py] = math.cos(2 * LO) #Angulo px

Angy[px][py] = math.sin(2 * LO) #Angulo py

AngO2[px][py] = LO

#Aproximación Campo de Orientación

for py in range(M):

    for px in range(N):

        Ox = Angx[px][py]

        Oy = Angy[px][py]

        if Oy == 0 : Oy = 1

        AngO[px][py] = math.atan(Ox / Oy) * 0.5

'Mapa de Error

#Mapa de Error

mie=0.00

for py in range(M):

    for px in range(N):

        mie = functX.MinE(abs(AngO2[px][py] - AngO[px][py]), functX.PI -
abs(AngO2[px][py] - AngO[px][py])) / (functX.PI / 2)

        mZeros[px][py] = mie

return mZeros

# Funciones de Usuario Utilizadas.

@numba.jit

def Gradiente_Sobel(Mtrx, px, py):

    #Creo las mascara Sobel px

    SobelX[0][0] =-1

```

```

SobelX[0][1]= 0
SobelX[0][2]= 1
SobelX[1][0]=-2
SobelX[1][1]= 0
SobelX[1][2]= 2
SobelX[2][0]= -1
SobelX[2][1]= 0
SobelX[2][2]= 1

#Creo las mascara Sobel py

SobelY[0][0]= -1
SobelY[0][1]= -2
SobelY[0][2]= -1
SobelY[1][0]= 0
SobelY[1][1]= 0
SobelY[1][2]= 0
SobelY[2][0]= 1
SobelY[2][1]= 2
SobelY[2][2]= 1

    #Calculo Sobel px - py

sum1 = 0
SUM2 = 0
col = 0

for vX in range(xini-1 , xfin):

    fil = 0

    for vy in range(yini-1 , yfin):

        Intensidad = Mtrx[vX][vy]

        sx=SobelX[fil][col]

```

```

    sy=SobelY[fil][col]

    sum1 = sum1 + sx* Intensidad

    SUM2 = SUM2 + sy* Intensidad

    #print(vX,vy,fil,col,sx,sy)

    fil = fil + 1

    col = col + 1

    GS = (sum1 ** 2 + SUM2 ** 2)**(1/2)

    return GS,sum1,SUM2

```

```
@numba.jit
```

```
def Local_Orientacion2(px, py, L, MGx, MGy):
```

```

    #Vector Gxx

    sum = 0

    for vX in range( xini-1 , xfin):

        for vy in range( yini-1 , yfin):

            gSX = MGx[vX][vy]

            gSY = MGy[vX][vy]

            vSX = vSX + (2 * gSX * gSY)

            vSY = vSY + (gSX ** 2 - gSY ** 2)

        if vSX == 0 :vSX = 1

    return PI / 2 + (0.5 * Atan2(vSX, vSY)) #Gxx

```

```
@numba.jit
```

```
def Atan2(px , py ) :
```

```

    if px > 0 :

        return math.atan(py / px)

    elif py >= 0 and px < 0 :

```

```

return math.atan(py / px) + PI
elif py < 0 and px < 0 :
    return math.atan(py / px) - PI
elif py > 0 and px == 0 :
    return PI / 2
elif py < 0 and px == 0 :
    return -PI / 2
else:
    return 0

```

Donde M y N son el tamaño de Mtrx (columnas y filas respectivamente); Mtrx, es la matriz de entrada que contiene la información de la imagen; `functX.Gradient_Sobel(Mtrx, px, py)`, calcula la gradiente sobel, en función a la intensidad de la coordenadas px,py, de la matriz Mtrx; `functX.Local_Orientacion2(px, py, 16, ImgGx, ImgGy)`, calcula el campo de orientación, en función a las gradientes de la coordenadas px,py, de las matrices `ImgGx` y `ImgGy`; y `mZeros`; `@numba.jit`, es un decorador de numba para acelerar el código de las funciones; `mZeros`, es la matriz de salida con el mapa de error calculado.

En la Fig. 29 se muestra el resultado de implementar el algoritmo para el análisis del campo de orientación de huellas dactilares.



(a)



(b)



(c)

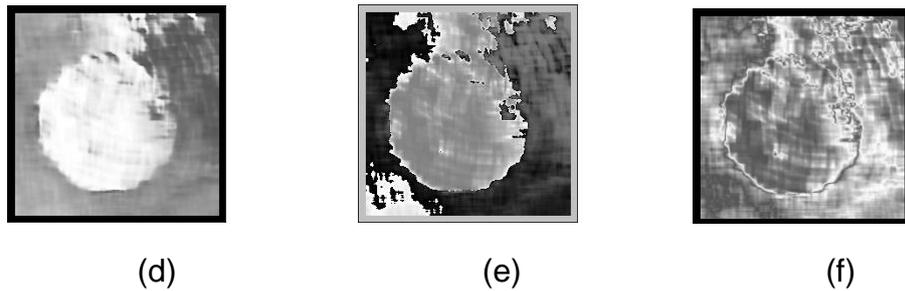


Fig. 29. Resultados de la Implementación del algoritmo para el análisis del campo de orientación de Huellas Dactilares. (a) Imagen ecualizada, (b) Imagen con gradiente X, (c) Imagen con gradiente Y, (d) Imagen con estimación de la orientación, (e) Imagen con aproximación del campo de orientación, (f) Imagen del mapa del Campo de orientación. Fuente: Elaboración Propia.

ANALISIS DE MINUCIAS.

Una de las principales características de las huellas dactilares son las minucias. En el caso de huellas dactilares alteradas hay ciertas minucias aumentan al no tener la continuidad de una huella dactilar sin alterar, estas minucias son las que se encuentran en los extremos de las crestas o en la bifurcación.

Este análisis está basado en la cuantificación de minucias de las huellas dactilares alteradas, que a menudo son más que en las huellas dactilares sin alterar. El análisis finaliza construyendo un mapa de densidad de las minucias. Los pasos para la extracción y análisis de minucias son los siguientes:

1. Como primer paso binarizamos la imagen. La binarización se refiere al proceso de convertir la imagen ecualizada a imágenes binarias. Esta binarización mejora el contraste entre las crestas y los valles; y facilita además el proceso de extracción de minucias. En una imagen binaria cada valor de píxel es 0 o 1 (255). El color negro(0), son las crestas y el color blanco(1), son los valles, como se muestra en la Fig. 30. [28]

Para binarizar la imagen analizaremos cada pixel. Para cada pixel se calculó la media con respecto a sus vecinos en una ventana de 15x15 en donde el pixel $p(x,y)$ es el centro. La media calculada es el umbral, en donde, si el pixel $p(x,y)$ es mayor o igual a la media, entonces $p(x,y)$ toma el valor de 1 (255), sino toma el de 0.

$$b(x,y) = \begin{cases} 1, & p(x,y) \geq M \\ 0, & p(x,y) < M \end{cases}$$

$$M = \frac{1}{w \times w} \sum_{i=x-w/2}^{x+w/2} \sum_{j=y-w/2}^{y+w/2} p(i,j)$$

El fragmento de código en Python para la binarización de una huella dactilar es el siguiente:

```

for py in range(M):
    for px in range(N):
        Intensidad = functX.MediaW(Mtrx, px, py, 15)
        if Mtrx[px][py] >= Intensidad :
            NewInten = 255
        else:
            NewInten = 0
        if NewInten > 0 :
            mZeros[px][py] = 0
        else:
            mZeros[px][py] = 1

return mZeros

```

```

'Calculo la media
@numba.jit
def MediaW(Mtrx, px, py, Limite):
    for vX in range( xini-1 , xfin):
        for vy in range( yini-1 , yfin):
            Intensidad2 = Mtrx[vX][vy]
            pSum = pSum + Intensidad2
            pCnt = pCnt + 1
    Media =pSum / pCnt
    return Media

```

Donde M y N son el tamaño de Mtrx (columnas y filas respectivamente); Mtrx, es la matriz de entrada que contiene la información de la imagen; `functX.MediaW(Mtrx, px, py, 15)`, calcula el valor promedio, en función a la intensidad de la coordenadas px,py, de la matriz Mtrx; `@numba.jit`, es un decorador de numba para acelerar el código de las funciones; `mZeros`, es la matriz de salida con las intensidades binarizadas.

En la Fig. 30 se muestra el resultado de implementar el algoritmo para la binarización de huellas dactilares.

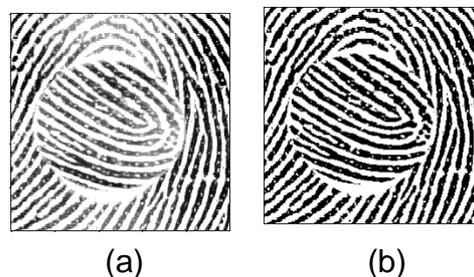


Fig. 30. Resultados de la Implementación del algoritmo para la binarización de Huellas Dactilares. (a) Imagen ecualizada, (b) Imagen binarizada. Fuente: Elaboración Propia.

2. Como segundo paso adelgazamos la imagen. En el adelgazamiento de la imagen de huellas dactilares, se toma la imagen binarizada y se va erosionando hasta que queda a la anchura de un pixel, del cual se van a extraer las minucias [29].

Para adelgazar la imagen binarizada, las crestas tienen el valor de 1 y los valles 0. En la imagen se debe eliminar las puntas y rellenar los huecos para que no creen falsas minucias, para ello se crea una matriz de 3x3, tomando como centro el pixel analizado.

$$Matriz = \begin{pmatrix} 9 & 2 & 3 \\ 8 & 1 & 4 \\ 7 & 6 & 5 \end{pmatrix}$$

Para eliminar las puntas, pixeles con valor 1 rodeados con valor 0, como se ve en la imagen (a) de la Fig. 31, realizamos una iteración como se ve en el siguiente algoritmo:

1. para todos los pixeles que tiene valor 1 (cresta),
2. donde sumamos los vecinos del pixel 1 (del 2 al 9),
3. si la suma es menor o igual a 3.
4. entonces preguntamos cuantos cambios de 0 a 1, o de 1 a 0, existen entre los pixeles 2 y 9,
5. se toma el mayor;
6. si el número de cambios es menor o igual a 1,
7. entonces procedemos a agregar 0 como valor del pixel.

La iteración termina cuando no existan pixeles para borrar, entonces iniciamos el llenado de huecos.

Para llenar los huecos, pixeles con valor 0 rodeados de valor 1, como se

ve en la imagen (b) de la Fig. 31, realizamos una iteración como se ve en el siguiente algoritmo:

1. para todos los pixeles que tiene valor 0 (valle),
2. donde sumamos los vecinos del pixel 1 (del 2 al 9),
3. si la suma es mayor que 5.
4. Entonces preguntamos cuantos cambios de 0 a 1, o de 1 a 0 existen entre los pixeles 2 y 9,
5. se toma el mayor;
6. si el número de cambios es menor o igual a 1,
7. entonces procedemos a agregar 1 como valor al pixel.

Después del análisis aún existe huecos para ello agregamos dos condiciones más,

8. preguntamos si la suma de los pixeles 2,6,4,8 son mayores a 2, o si la suma de los pixeles 3,7,5,9 también es mayor a 2,
9. en cualquier de los dos casos se agrega 1 como valor del pixel.

La iteración termina cuando no existan pixeles que agregar, entonces iniciamos el adelgazamiento de la imagen.

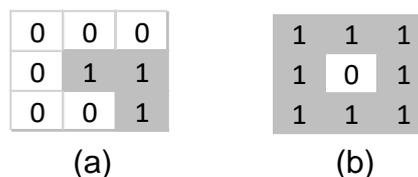


Fig. 31. Representación de la matriz de un hueco y una punta en una imagen binarizada. (a) Punta, (b) Hueco. Fuente: Elaboración Propia.

Para adelgazar la imagen, en donde mediante una iteración se erosiona el borde de la cresta de la imagen, quedando en el centro de esta un fino hilo con pixeles de valor 1.

Realizamos una iteración para todos los pixeles que tiene valor 1 (cresta). Para el análisis de cada uno de estos pixeles realizamos una matriz de 5x5, tomando como centro el pixel analizado.

$$Matriz = \begin{pmatrix} 24 & 25 & 10 & 11 & 12 \\ 23 & 9 & 2 & 3 & 13 \\ 22 & 8 & 1 & 4 & 14 \\ 21 & 7 & 6 & 5 & 15 \\ 20 & 19 & 18 & 17 & 16 \end{pmatrix}$$

Cada pixel se analiza y se marca con valor de 1 en una matriz auxiliar para luego ser eliminado,

1. para todos los pixeles que tiene valor 1 (cresta).
2. Si el número de cambios (de 0 a 1 y 1 a 0), entre sus vecinos (del 2 al 9), es
 - 1.
3. Si la suma de sus vecinos (del 2 al 9), es 0.
4. se marca con valor de 1
5. Si la suma de sus vecinos (del 2 al 9), es 3,
6. se verifica que la suma de los pixeles vecinos 23, 9, 2, 3 y 13 sea mayor a 3, o que la suma de los pixeles vecinos 21,7,6,5 y 15 sea mayor a 3, o que la suma de los pixeles vecinos 11, 3, 4, 5 y 17 sea mayor a 3, o que la suma de los pixeles vecinos 25, 9, 8, 7 y 17 sea mayor a 3.
7. se marca con valor de 1
8. Si la suma de sus vecinos es igual a 4,
9. se marca con valor de 1
10. Si la suma de sus vecinos es igual a 5, se verifica que la suma de los pixeles vecinos 3,4,5,21,22 y 23 no sea 0, o que la suma de los pixeles vecinos 5,6,7,25,10 y 11 no sea 0.
11. se marca con valor de 1
12. Los pixeles marcados en la matriz auxiliar se eliminan de la matriz

binarizada,

13. luego se reinicia la matriz auxiliar.

La iteración termina cuando no existan pixeles marcados.

El fragmento de código en Python para el adelgazamiento es el siguiente:

```
Copialmg = Mtrx.copy()

#Abrir

cont = N

while cont > 0:

    cont = 0

    for px in range( N-1):

        for py in range( M-1):

            if Copialmg[px][py] == 1 :

                cont = cont + functX.Puntas(Copialmg, px, py)

#Cerrar

cont = N

while cont > 0:

    cont = 0

    for px in range( N - 1):

        for py in range( M - 1):

            if Copialmg[px][py] == 0 :

                cont = cont + functX.RellenarHuecos(Copialmg, px, py)

cont = N

while cont > 0:

    #Marcar

    cont = 0

    MarcarImg= np.zeros((N,M),dtype=np.int16)
```

```
for px in range(2, N-2):
    for py in range(2, M-2):
        if Copialmg[px][py] == 1 :

            p = functX.ThinningAlg(Copialmg, px, py)

            MarcarImg[px][py] = p

            cont = cont + p

#Eliminar Marcados

cont = 0

for px in range( N):
    for py in range( M):
        if MarcarImg[px][py] == 1 :
            Copialmg[px][py] = 0
            cont = cont + 1

return Copialmg

#Funciones creadas por el usuario

@numba.jit

def RellenarHuecos(Mtrx, px, py):

    RH = 0

    #lleno Las variables

    p=[]

    p.append(0)

    p.append(Mtrx[px][py])

    p.append(Mtrx[px][py-1])

    p.append(Mtrx[px + 1][py - 1])

    p.append(Mtrx[px + 1][py])

    p.append(Mtrx[px + 1][ py + 1])
```

```

p.append(Mtrx[px][py + 1])

p.append(Mtrx[px - 1][py + 1])

p.append(Mtrx[px - 1][py])

p.append(Mtrx[px - 1][py - 1])

if BP(p) > 5 :

    if AP(p) <= 1 : #Un solo Grupo

        Mtrx[px][py] = 1

        RH = 1

if p[2] + p[6] + p[4] + p[8] > 2:

    Mtrx[px][py] = 1

    RH = 1

elif p[3] + p[7] + p[5] + p[9] > 2 :

    Mtrx[px][py] = 1

    RH = 1

return RH

```

@numba.jit

```

def Puntas(Mtrx, px, py):

    #lleno Las variables

    #On Error Resume Next

    p=[]

    p.append(0)

    p.append(Mtrx[px][py])

    p.append(Mtrx[px][py-1])

    p.append(Mtrx[px + 1][py - 1])

    p.append(Mtrx[px + 1][py])

```

```

p.append(Mtrx[px + 1][ py + 1])
p.append(Mtrx[px][py + 1])
p.append(Mtrx[px - 1] [py + 1])
p.append(Mtrx[px - 1][ py])
p.append(Mtrx[px - 1][ py - 1])
RP=0
if BP(p) <= 3 :
    if AP(p) <= 1 :
        Mtrx[px][py] = 0
        RP= 1
        return RP

```

@numba.jit

```

def ThinningAlg(Mtrx, px, py):
    p=[]
    p.append(0)
    p.append(Mtrx[px][py])
    p.append(Mtrx[px][py-1])
    p.append(Mtrx[px + 1][py - 1])
    p.append(Mtrx[px + 1][py])
    p.append(Mtrx[px + 1][ py + 1])
    p.append(Mtrx[px][py + 1])
    p.append(Mtrx[px - 1] [py + 1])
    p.append(Mtrx[px - 1][ py])
    p.append(Mtrx[px - 1][ py - 1])
    #Matriz Auxiliar

```

```

p.append(Mtrx[px][ py - 2])
p.append(Mtrx[px + 1][py - 2])
p.append(Mtrx[px + 2][py - 2])
p.append(Mtrx[px + 2][py - 1])
p.append(Mtrx[px + 2][ py])
p.append(Mtrx[px + 2][py + 1])
p.append(Mtrx[px + 2][py + 2])
p.append(Mtrx[px + 1][ py + 2])
p.append(Mtrx[px][ py + 2])
p.append(Mtrx[px - 1][ py + 2])
p.append(Mtrx[px - 2][ py + 2])
p.append(Mtrx[px - 2][ py - 1])
p.append(Mtrx[px - 2][ py])
p.append(Mtrx[px - 2][py - 1])
p.append(Mtrx[px - 2][py - 2])
p.append(Mtrx[px - 1][py - 2])

TP=0

if AP(p) == 1 :
    if BP(p) == 0 :
        TP = 1
    elif BP(p) == 3 :
        if p[23] + p[9] + p[2] + p[3] + p[13] > 3 : TP= 1
        if p[21] + p[7] + p[6] + p[5] + p[15] > 3 : TP= 1
        if p[11] + p[3] + p[4] + p[5] + p[17] > 3 : TP= 1
        if p[25] + p[9] + p[8] + p[7] + p[19] > 3 : TP= 1
    elif BP(p) == 4 :
        TP = 1

```

```

elif BP(p) == 5 :
    if (p[3] + p[4] + p[5] + p[21] + p[22] + p[23] == 0) or (p[5] + p[6] + p[7]
+ p[25] + p[10] + p[11] == 0) :
        #No imprime
        h=0
    else:
        TP= 1
        return TP

```

```
@numba.jit
```

```

def BP(vA) :
    s=0
    for i in range ( 2 , 10):
        s = s + vA[i]
    return s

```

```
@numba.jit
```

```

def AP(vA) :
    s1=0
    s2=0
    pa = vA[2]
    for i in range( 3 , 10):
        if vA[i] == 1 :
            if pa == 0 : s1 = s1 + 1
        pa = vA[i]
    pa = vA[2]
    for i in range( 3 , 10):

```

```

if vA[i] == 0 :
    if pa == 1 : s2 = s2 + 1
    pa = vA[i]
if s1>s2:
    return s1
else:
    return s2

```

Donde M y N son el tamaño de Mtrx (columnas y filas respectivamente);

Copialmg es una copia de la matriz Mtrx, que es la matriz de entrada que contiene la información de la imagen; functX.Puntas(Copialmg, px, py), elimina las puntas o islas agregando valor 0, en función a la intensidad de la coordenadas px,py con valor 1, de la matriz Copialmg; functX.RellenarHuecos(Copialmg, px, py), agrega valor 1 en los huecos con valor 0, en función a la intensidad de la coordenadas px,py, de la matriz Copialmg; functX.ThinningAlg(Copialmg, px, py) , erosiona , en función a la intensidad de la coordenadas px,py, de la matriz Copialmg ;@numba.jit, es un decorador de numba para acelerar el código de las funciones; Copialmg, es la matriz de salida con la imagen adelgazada.

En la Fig. 32 se muestra el resultado de implementar el algoritmo para el adelgazamiento de huellas dactilares.

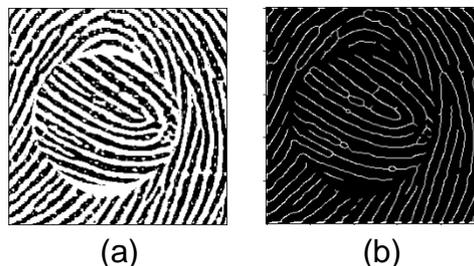


Fig. 32. Resultados de la Implementación del algoritmo para el adelgazamiento de Huellas Dactilares. (a) binarizada, (b) Adelgazada. Fuente: Elaboración Propia.

3. Como tercer paso extraemos las minucias de la imagen adelgazada. Cabe mencionar que las minucias que se extraerán son los fines de cresta y las bifurcaciones, como se muestra en la Fig. 33.

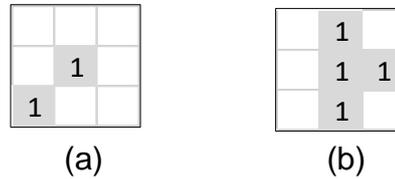


Fig. 33. Representación de la matriz de un fin de cresta y una bifurcación en una imagen adelgazada. (a) Fin de cresta, (b) bifurcación. Fuente: Elaboración Propia.

Para analizar el pixel si se trata de una minucia creamos una matriz de 3x3, tomando como centro el pixel analizado.

$$Matriz = \begin{pmatrix} 9 & 2 & 3 \\ 8 & 1 & 4 \\ 7 & 6 & 5 \end{pmatrix}$$

Para extraer las minucias realizamos una iteración como muestra, el siguiente algoritmo:

1. para todos los pixeles que tiene valor 1 (cresta adelgazada),
2. sumamos los vecinos del pixel 1 (del 2 al 9).
3. Si la suma es 1, entonces
4. agregamos la minucia (fin de cresta).
5. Si el número de cambios de entre sus vecinos (del 2 al 9), de 0 a 1 y 1 a 0 es mayor o igual a tres,
6. agregamos minucias (bifurcación).

Como lo que se desea es identificar si la huella dactilar es alterada, eliminaremos minucias que están en los extremos de la imagen para disminuir la cantidad de minucias no alteradas.

Para quitar las minucias del borde de la imagen se realiza dos iteraciones verticales en la imagen adelgazada, la primera identifica todos los primeros pixeles con valor 1 que se encuentran de arriba hacia abajo y la segunda identifica todos los primeros pixeles que se encuentran de abajo hacia arriba. Los pixeles identificados se agregan a una matriz de bordes.

Los pixeles con valor 1 de la matriz de minucias son comparados con los pixeles de la matriz de bordes, si ambos tienen el valor de 1 se elimina la minucia.

El fragmento de código en Python para la extracción de minucias es el siguiente:

```
mxMascara=functX.Mascara(Mtrx)
for py in range( 2 , M - 1):
    for px in range ( 2 , N - 1):
        Intensidad = Mtrx[px][py]
        if Intensidad == 1 :
            vm=[]
            vm.append(0)
            vm.append(0)
            vm.append(Mtrx[px][ py - 1])
            vm.append(Mtrx[px + 1][ py - 1])
            vm.append(Mtrx[px + 1][ py])
            vm.append(Mtrx[px + 1][ py + 1])
            vm.append(Mtrx[px][ py + 1])
            vm.append(Mtrx[px - 1][py + 1])
            vm.append(Mtrx[px - 1][ py])
            vm.append(Mtrx[px - 1][py - 1])
```

```

        if functX.BP(vm) == 1 : #fN
            NewInten = 1
        elif functX.AP(vm) >= 3 :
            NewInten = 1
        else:
            NewInten = 0
    else:
        NewInten = 0
    if NewInten > 1 : NewInten = 0
    if NewInten == 1 :
        if mxMascara[px][py] == 1 :
            NewInten = 0
        else:
            NewInten = 1
    Matrix2[px][py] = NewInten

```

```

#procedimiento mascara mtrx

```

```

@numba.jit(int16[:,:](int16[:,:]))

```

```

def Mascara(Mtrx):

```

```

    M= Mtrx.shape[0]

```

```

    N= Mtrx.shape[1]

```

```

    mZeros= np.zeros((N,M),dtype=np.int16)

```

```

    #Calculo el Promedio

```

```

    for px in range(M):

```

```

        #i = 0

```

```

for py in range(N):
    Intensidad = Mtrx[px][py]
    if Intensidad == 1 :
        mZeros[px][py] = 1
        break

for px in range(M):
    #i = 0
    for py in range( N-1,-1,-1): #To 1 Step -1
        Intensidad = Mtrx[px][py]
        if Intensidad == 1 :
            mZeros[px][py] = 1
            break

for px in range(M):
    for py in range(N):
        if px <= 1 or py<=1 : mZeros[px][py] = 1
        if px >= M-2 or py>=M-2 : mZeros[px][py] = 1

return mZeros

```

Donde M y N son el tamaño de Mtrx (columnas y filas respectivamente); Mtrx, es la matriz de entrada que contiene la información de la imagen; functX.Mascara(Mtrx), marca las minucias que se encuentran en los bordes de la matriz, en función a la intensidad de la matriz Mtrx; @numba.jit, es un decorador de numba para acelerar el código de las funciones.

En la Fig. 34 se muestra el resultado de implementar el algoritmo para extraer minucias de las imágenes.

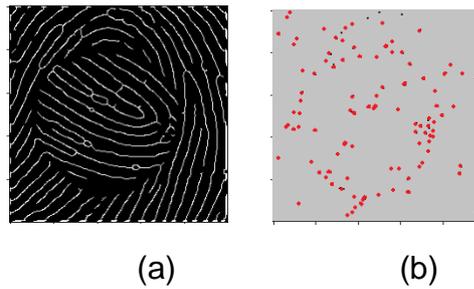


Fig. 34. Resultados de la Implementación del algoritmo para extraer minucias de las Huellas Dactilares. (a) Imagen a escala de grises, (b) Imagen identificando la ubicación de la minucia. Fuente: Elaboración Propia.

4. Como cuarto paso creamos un mapa de densidades con las minucias extraídas. Con este mapa ubicaremos las zonas en donde existe mayor concentración de minucias, por lo que en una huella dactilar alteradas encontraremos un mayor número de minucias que en una huella dactilar no alterada.

Para encontrar el mapa de densidades, se analizan todos los pixeles de la imagen. Se crea una matriz de 25x25, tomando como centro el pixel analizado. Se suman todos los pixeles de esta matriz, incluyendo el pixel analizado y esta suma se agrega como valor en la matriz del mapa de densidad en las coordenadas del pixel analizado.

El fragmento de código en Python para el mapa de densidades es el siguiente:

```
for py in range( M):
    for px in range( N):
        Intensidad = functX.MapaDensidad(Matrix2, px, py, 25)
#25M
```

```

        if Intensidad <=0: #0M
            mZeros[px][py] = 0 #If(Intensidad < 255, Intensidad, 255)
        else:
            mZeros[px][py] = Intensidad
    return mZeros

#Funciones realizadas por el usuario
@numba.jit
def MapaDensidad(Mtrx, px, py, Limite):
    pSum = 0
    for vX in range( xini-1 , xfin):
        for vy in range( yini-1 , yfin):
            Intensidad2 = Mtrx[vX][vy]
            pSum = pSum + Intensidad2
    return pSum

```

Donde M y N son el tamaño de Matrix2 (columnas y filas respectivamente); Matrix2, es la matriz que contiene la extracción de minucias; functX.MapaDensidad(Matrix2, px, py, 25), encuentra la densidad de minucias en una ventana de 25x25, en función a la intensidad de la coordenadas px,py, de la matriz Matrix2; @numba.jit, es un decorador de numba para acelerar el código de las funciones; mZeros, es la matriz de salida con la en mapa de densidades.

En la Fig. 35 se muestra el resultado de implementar el algoritmo para la creación de mapa de densidades de minucias.

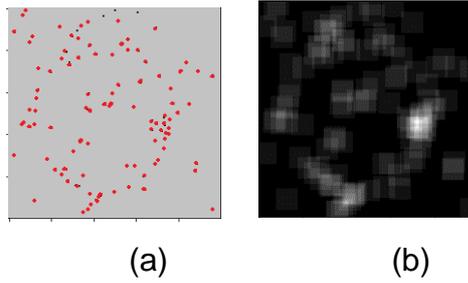


Fig. 35. Resultados de la Implementación del algoritmo del análisis de minucias de Huellas Dactilares. (a)Imagen con ubicación de minucias, (b) Imagen del mapa de densidades. Fuente: Elaboración Propia.

5. El paso siguiente es guardar la matriz calculada para que sea procesada en el clasificador SVM.

Con las dos matrices calculadas en el algoritmo de extracción de minucias (250x250) y el algoritmo del campo de orientación (250x250), cada una se convierte en un array unidimensional (62500 ítems), el primer array con los valores unitarios que representan la cantidad de minucias que fueron sumadas en un determinado pixel; y el segundo array con valores decimales que representan el vector calculado en un determinado pixel.

En la Tabla VI, muestra el cuadro comparativo de un extracto del array unidimensional del algoritmo de extracción de minucias y el array unidimensional del algoritmo del campo de orientación, de una imagen de huella dactilar con la alteración tipo obliteración; en donde se observa que cuando no hay minucias o las minucias son pocas (1), el vector del campo de orientación no sufre cambios; pero cuando existen varias minucias, el vector del campo de orientación varía en cada elemento.

Tabla VI

Cuadro comparativo del array unidimensional, de los algoritmos de extracción de minucias y del campo de orientación.

Índice	Array unidimensional del algoritmo de extracción de minucias	Array unidimensional del algoritmo Campo de Orientación
1	1	0.25
2	1	0.25
3	1	0.25
4	1	0.25
5	1	0.25
6	1	0.25
7	1	0.25
8	1	0.25
9	1	0.25
10	1	0.25
...
34575	16	0.581619
34576	17	0.595399
34577	17	0.634283
34578	17	0.659168
34579	17	0.657366
34580	17	0.649402
34581	16	0.672759
34582	15	0.633673
34583	15	0.588675
34584	15	0.542358
...
62491	0	0.25
62492	0	0.25
62493	0	0.25
62494	0	0.25
62495	0	0.25
62496	0	0.25
62497	0	0.25
62498	0	0.25
62499	0	0.25
62500	0	0.25

Nota: En la segunda columna es el extracto del array del algoritmo de extracción de minucias y en la tercera columna es el extracto del array del algoritmo del campo de orientación. Los valores mostrados corresponden a 30 elementos de los 62500 de cada matriz(10 primeros, 10 intermedios y 10 últimos). Fuente: Elaboración Propia.

D. EVALUAR RESULTADOS DE LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS

En la evaluación de los algoritmos de clasificación automática para detección de huellas dactilares alteradas, se utilizó un 50% de imágenes para entrenamiento y 50% para prueba, para cada grupo de imágenes (reales, obliteradas, corte Z y rotación central) de los dos algoritmos implementados. Teniendo un total de 800 imágenes procesadas, 400 procesadas por el algoritmo del campo de orientación y 400 por el algoritmo de extracción de minucias.

El código en Python, para dividir las imágenes en entrenamiento y prueba es el siguiente:

```
import numpy as np

import HuellaDactilar02 as hda

import matplotlib.pyplot as plt

from sklearn.metrics import accuracy_score

from sklearn.metrics import confusion_matrix

from sklearn.metrics import classification_report

from sklearn import svm

from sklearn.metrics import plot_confusion_matrix

# Declaro e inicializo las variables con las rutas de las bases de datos

dir_re='D:\JANO\dataset\Original\Real'

dir_cz='D:\JANO\dataset\Original\Alterada\CZ'

dir_rc='D:\JANO\dataset\Original\Alterada\RC'
```

```

dir_ob='D:\JANO\dataset\Original\Alterada\OB'

# Declaro e inicializo las variables con la combinación a clasificar
m_ob_re = [dir_ob,dir_re]
em_ob_re=['Obliteracion','Real']

m_cz_re = [dir_cz,dir_re]
em_cz_re=['CorteZ','Real']

m_rc_re = [dir_rc,dir_re]
em_rc_re=['RotaCentro','Real',]

# Declaro e inicializo las variables a evaluar
Categorias=m_rc_re
Etiquetas= em_rc_re
EtiquetasX=Etiquetas

#Declaro las variables en las que se almacena la evaluación
DataSet=[]
xntrain=[]
yntrain=[]
xntest=[]
yntest=[]

# porcentaje que se utiliza como entrenamiento.
train=0.5

# Codigo para dividir las imágenes en entrenamiento y prueba
for idx in range(len(Categorias)):

```

```

ruta=Categorias[idx]
etiqueta=Etiquetas[idx]
idn=0
itrain=int(len(os.listdir(ruta))*train)
for img in os.listdir(ruta):
    rutaimg=os.path.join(ruta,img)
    himg = hda.procesarHuella(rutaimg, 1)
    # 0-Extraccion de Minucias / 1-Campo de Orientacion
    huella=np.array(himg).flatten()

    idn=idn+1
    if idn<=itrain:
        xntrain.append(huella)
        yntrain.append(etiqueta)
    else:
        xntest.append(huella)
        yntest.append(etiqueta)

```

Para la clasificación SVM se utilizó sklearn, con el kernel RBF (Función Básica Radial) [7].

El código en Python, para el entrenamiento y predicción es el siguiente:

```

xn_predict = []
clasf = svm.SVC(kernel='rbf')
clasf.fit(xntrain, yntrain)
xn_predict = clasf.predict(xntest)
accuracy = accuracy_score(yntest, xn_predict)

```

El código en Python, para extraer la información de predicción es el siguiente:

```
print("Predicción para:")
print(EtiquetasX)

con_matrix=confusion_matrix(yntest,x_predict,EtiquetasX)
plot_confusion_matrix(clasf, xntest, yntest,EtiquetasX)
plt.show()
```

Las matrices de confusión, que fueron el resultado de la implementación la clasificación SVM para los diferentes grupos de imágenes (reales, obliteradas, corte Z y rotación central), de los dos algoritmos de clasificación automática implementados están en Anexo 03.

En la Tabla VII podemos observar el resumen de las de la matriz de confusión de cada uno de los algoritmos para cada tipo de alteración.

Tabla VII

Resumen de las matrices de confusión, de los algoritmos EM y CO.

Tipo de Alteración	A. Extracción de Minucias (EM)				A. Campo de Orientación (CO)			
	VP	FN	FP	VN	VP	FN	FP	VN
	Obliteración	47	3	14	36	32	18	5
Corte en Z	32	18	28	22	33	17	31	19
Rotación Central	34	16	23	27	26	24	24	26

Nota: Las cantidades son un resumen de la matriz de confusión de cada uno de los algoritmos para cada tipo de alteración. Fuente: Elaboración Propia.

En la Tabla VIII, podemos observar los porcentajes para los indicadores de exactitud, exhaustividad y precisión, que fueron calculados a partir del resumen de la matriz de confusión de cada uno de los algoritmos para cada tipo de alteración.

Tabla VIII

Resultados de Exactitud, Exhaustividad y Precisión de las pruebas realizadas a los algoritmos EM y CO.

Tipo de Alteración	Exactitud		Exhaustividad		Precisión	
	EM	CO	EM	CO	EM	CO
Obliteración	83%	77%	94%	64%	78%	87%
Corte en Z	54%	52%	64%	66%	54%	52%
Rotación Central	61%	52%	68%	52%	60%	52%

Nota: Los rendimientos que se observan, fueron calculados a partir de la matriz de confusión de cada uno de los algoritmos para cada tipo de alteración. Fuente: Elaboración Propia.

En la Tabla IX podemos observar los porcentajes para los indicadores de porcentaje de falsos positivos (TFP) y porcentaje de falsos negativos (TFN), que fueron calculados a partir del resumen de la matriz de confusión de cada uno de los algoritmos para cada tipo de alteración.

Tabla IX

Resultados de APCER y NPCER de las pruebas realizadas a los algoritmos EM y CO para la detección de huellas dactilares alteras.

Tipo de Alteración	APCER		NPCER	
	EM	CO	EM	CO
Obliteración	28%	10%	6%	36%
Corte en Z	56%	62%	36%	34%
Rotación Central	46%	48%	32%	48%

Nota: Estos rendimientos fueron calculados a partir de la matriz de

confusión de cada uno de los algoritmos para cada tipo de alteración. Fuente: Elaboración Propia.

También se calculó el tiempo de procesamiento de cada una de las imágenes a partir del registro del tiempo de procesamiento, que se encuentra en Anexo 3. En la Tabla X, se observa el tiempo procesamiento promedio de cada uno de los algoritmos para cada tipo de alteración.

Tabla X
Resultados del tiempo de procesamiento promedio de los algoritmos EM y CO.

Tipo de Alteración	Tiempo de Procesamiento (Seg.)	
	EM	CO
Obliteración	0.591	0.232
Corte Z	0.606	0.230
Rotación Central	0.564	0.247

Nota: El tiempo promedio que se muestra en la tabla fue calculado a partir del registro de tiempo de procesamiento de cada uno de los algoritmos para cada tipo de alteración. Fuente: Elaboración Propia.

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

Se concluye:

Que, para la construcción de la base de datos de huellas sin alterar, se escaneó los 10 dedos de las manos, de 10 individuos voluntarios, del Distrito de Chongoyape, Provincia de Chiclayo, Departamento de Lambayeque; haciendo un total de 100 imágenes de huellas dactilares huellas sin alterar y se redimensionaron calculando el centro ponderado con mayor escala de grises, el cual permitió centrarnos en la zona con mayor información de la huella.

Que, durante la búsqueda para la selección de algoritmos de clasificación automática de huellas dactilares alteradas en los repositorios científicos, se encontraron que algunas investigaciones no son públicas, dificultando ampliar el número de algoritmos. Siendo los algoritmos de clasificación automática seleccionados, los basados en extracción de minucias y los basados en el campo de orientación.

Que, en la implementación de los algoritmos de clasificación automática, en la etapa de mejoramiento de la imagen, no fue necesario calcular la región de interés (ROI), porque al aumentar los niveles de blancos con el algoritmo de ecualización los pixeles con bajos niveles de grises se eliminan, además que al no calcular el ROI disminuye el tiempo de procesamiento.

Que, de la evaluación realizada se obtuvo que, el algoritmo de clasificación automática basado en el campo de orientación tiene mejores rendimientos para clasificar huellas dactilares, con una precisión del 87%, con un tiempo de procesamiento de 0.230 seg. y tiene la tasa más baja de error de clasificación de APCER con un 10% (huellas dactilares alteradas falsamente rechazadas).

Que, de la evaluación realizada se obtuvo que, el algoritmo de clasificación automática

basado en la extracción de minucias tiene mejores rendimientos para predecir si una huella dactilar es alterada o no alterada, teniendo una tasa de exhaustividad del 94%, con un tiempo de procesamiento de 0.564 seg. y también tiene la tasa más baja el error de clasificación de NPCER con un 6% (huellas dactilares falsamente aceptadas como alteradas).

Que, de la evaluación realizada se obtuvo que, el algoritmo basado en el campo de orientación y el algoritmo basado en la extracción de minucias, tienen mejor rendimiento para detectar la alteración tipo obliteración y poco rendimiento para las alteraciones tipo Corte Z y Rotación Central.

Que, el método propuesto es eficiente para detectar huellas dactilares alteradas y también para la detección de ataques de presentación biométrica, obteniendo tasas bajas en NPCER y APCER (6% y 10% respectivamente), implicando una menor inspección manual de huellas dactilares, por parte de expertos en huellas dactilares alteradas.

4.2. Recomendaciones

Se recomienda que, para obtener la misma calidad de imágenes sin alterar, debe utilizar el lector óptico "U.are.U 4500", marca de DigitalPersona.

Se recomienda que, para obtener la misma calidad de imágenes alteradas, debe utilizar la herramienta STRANGE de Papi, Ferrara, Maltoni, & Anthonioz [13].

Se recomienda que, para obtener los mismos resultados en el tiempo de procesamiento debe utilizar una computadora con un procesador Intel (R) Core (TM)/i5 – 10300 H/CPU@2.50GHz/2.50GHz con 16 GB de R.A.M y sistema operativo Windows 11.

Se recomienda que, para obtener los mismos resultados, los algoritmos y la clasificación en SVM se debe implementar en Python 3.9.

REFERENCIAS

- [1] C. J. I. S. D. FBI's, «Altered Fingerprints: A Challenge to Law Enforcement Identification Efforts,» FBI, 05 05 2015. [En línea]. Available: <https://leb.fbi.gov/spotlights/forensic-spotlight-altered-fingerprints-a-challenge-to-law-enforcement-identification-efforts>. [Último acceso: 15 06 2016].
- [2] Velayutham y Vijayalakshmi, «Automatic Detection of Altered Fingerprints,» *International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)*, vol. 1, nº 5, pp. 31-41, 2012.
- [3] mpfn, «Instituto de Medicina Legal y Ciencias Forenses,» Ministerio Público - Fiscalía de la Nación, 15 03 2016. [En línea]. Available: <https://www.mpfn.gob.pe/iml/?K=460&id=3752>. [Último acceso: 24 09 2021].
- [4] Peru21, «Peru 21,» Peru 21, 01 04 2015. [En línea]. Available: <http://peru21.pe/actualidad/policia-capturo-estudiante-que-clonaba-huellas-digitales-2215640>. [Último acceso: 15 06 2016].
- [5] P. K. Patil, S. S. Patil y N. P. Jagtap, «Study of Altered Fingerprint Detection Using Artificial Neural Network,» *International Journal of Science and Research (IJSR)*, vol. 4, nº 1, pp. 2471-2473, 2013.
- [6] Y. Gaoqiang, L. Sheng y Z. Xinpeng, «Altered Fingerprint Detection Based on Orientation Field,» *JOURNAL OF APPLIED SCIENCES—Electronics and Information Engineering*, vol. 37, nº 6, pp. 775-782, 2019.
- [7] Y. I. Shehu, A. James y a. V. Palade, «Detecting an Alteration in Biometric Fingerprint Databases,» *ICDSP 2018: Proceedings of the 2nd International Conference on Digital Signal Processing*, vol. 1, nº 1, pp. 6-11, 2018.

- [8] A. Vinoth y S. Saravanakumar, «Region based Minutiae Mass Measure for Efficient Finger Print Forgery Detection in Health Care System,» *International Journal of Recent Technology and Engineering (JRTE)*, vol. 7, nº 4, pp. 9-14, 2018.
- [9] C. Gottschlich, B. Tamsy y S. Huckemannz., «Perfect Fingerprint Orientation Fields,» *Computer Vision and Pattern Recognition (cs.CV) IET Biometrics*, vol. 6, no.3, vol. 1, nº 1, pp. 183-190, 2018.
- [10] S. P. Singh, S. Ayub y S. Jp, «Minuatiae Extraction Based Matching Between Normal And Altered Fingerprint,» *Journal of Industrial Pollution Control*, vol. 1, nº 1, pp. 1560-1565, 2017.
- [11] M. V. R. Patil y P. (. B. D. Phulpagar, «Efficient Altered Fingerprint Analysis and Rectification of Distorted Fingerprint,» *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, nº 7, pp. 2055-2059, 2017.
- [12] R. Haraksim, A. Anthonioz, C. Champod, M. Olsen, J. Ellingsgaard y C. Busch, «Altered fingerprint detection – algorithm performance evaluation,» *2016 4th International Conference on Biometrics and Forensics (IWBF)*, vol. 1, nº 1, pp. 1-6, 2016.
- [13] S. Papi, M. Ferrara, D. Maltoni y A. Anthonioz, «On the generation of synthetic Fingerprint alterations,» *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*, vol. 1, nº 1, pp. 51-62, 2016.
- [14] T. Anoop y M. Mini, «Two stage framework for altered fingerprint matching,» *ICTACT Journal on Image and Video Processing*, vol. 5, nº 4, pp. 1024-1029, 2015.
- [15] T. Anoop y M. Mini, «Altered Fingerprint Matching Using Ridge Texture and Frequency in the Unaltered Region,» *Bonfring International Journal of Advances in Image Processing 5(2)*, vol. 5, nº 2, pp. 6-9, 2015.

- [16] C. Gottschlich, A. Mikaelyan, M. A. Olsen, J. Bigun y C. Busch, «Improving Fingerprint Alteration Detection,» *2015 9th International Symposium on Image and Signal Processing and Analysis (ISPA)*, vol. 1, nº 1, pp. 83-86, 2015.
- [17] R. K. Nagthane y H. S. Fadewar, «Matching of Altered Fingerprint using Minutiae Technique,» *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, nº 7, pp. 649-654, 2015.
- [18] G. Mohan, «Embedded Fingerprint Matching Mechanism in HIS for Health Care Service,» *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 6, nº 5, pp. 56-64, 2014.
- [19] S. Chen, S. Chang, Q. Huang, H. W. Jin He y Q. Huang, «SVM-Based Synthetic Fingerprint Discrimination Algorithm and Quantitative Optimization Strategy,» *PLoS ONE* 9(10): e1111099. doi:10.1371/journal.pone.01111099, vol. 9, nº 10, pp. 1-9, 2014.
- [20] D. K. Latha y C. Manikandan, «Critical Analysis and Detection of Altered Fingerprints,» *International Conference on Innovations in Engineering and Technology (ICIET'14)*, vol. 3, nº 3, pp. 2077-2080, 2014.
- [21] S. Yoon, J. Feng y A. K. Jain, «Altered Fingerprints: Analysis and Detection,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, nº 3, pp. 451-464, 2012.
- [22] N. I. o. J. NIJ, *Fingerprint Sourcebook*, Washington, DC - U.S.: U.S. Department of Justice - Office of Justice Programs, 2011.
- [23] Jain, Flynn y Ross, *Handbook of Biometrics*, USA: Springer, 2008.
- [24] U. S. Kumar, «An Integrated Automated Altered Fingerprint Identification System,» *International Journal of Research in Computer and Communication Technology*, vol. 1, nº 1, p. Volumen 3, 2014.

- [25] R. Suryawanshi, T. Kamble, N. Joshi, A. Nikam y A. Khedkar, «Analysis and Detection of Altered Human Fingerprints using Minutiae Technique,» *International Journal of Emerging Technology and Advanced Engineering*, vol. 4, nº 10, pp. 346-350, 2014.
- [26] R. Kumar, J. P. Singh y G. Srivastava, «A Survey Paper on Altered Fingerprint Identification & Classification,» *International Journal of Electronics Communication and Computer Engineering*, vol. 3, nº 5, pp. 1212-1216, 2012.
- [27] I. G. Babatunde, A. O. Charles, A. B. Kayode y O. Olatubosun, «Fingerprint Image Enhancement: Segmentation to Thinning,» *International Journal of Advanced Computer Science and Applications*, vol. 3, nº 1, pp. 15-24, 2012.
- [28] M. Saveski, «Development of an Automated Fingerprint Verification System,» Staffordshire University, UK, 2010.
- [29] D. V. S. Gayathri S, «An Improved Fast Thinning Algorithm for Fingerprint Image,» *International Journal of Engineering Science and Innovative Technology (IJESIT)*, vol. 2, nº 1, pp. 264-270, 2013.
- [30] S. Khade, S. Gite, S. D. Thepade y B. P. y. A. Alamri, «Detection of Iris Presentation Attacks Using Hybridization of Discrete Cosine Transform and Haar Transform With Machine Learning Classifiers and Ensembles,» *IEEE Access*, vol. 9, nº 1, pp. 169231-169249, 2021.
- [31] O. Dermawan, Defriyanto, Junaidah, S. M. Ayu y B. y. Meisuri, «Microsoft Visual Basic 6.0 Software to Develop Mathematics Teaching Materials,» *IOP Conf. Series: Journal of Physics: Conf. Series 1155 (2019) 012090*, vol. 1, nº 1, pp. 1-16, 2019.
- [32] A. Sharma, F. Khan y D. S. y. D. S. Gupta, «Python: The Programming Language of Future,» *INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN TECHNOLOGY*, vol. 6, nº 12, pp. 115-118, 2020.

- [33] H. Byun y S.-W. Lee, «Applications of Support Vector Machines for Pattern Recognition: A Survey,» *Pattern Recognition with Support Vector Machines*, vol. 2388, nº 1, pp. 213-236, 2002.
- [34] C. Kanan y G. W. Cottrell, «Color-to-Grayscale: Does the Method Matter in Image,» *PLoS ONE* 7(1): e29740., vol. 1, nº 1, pp. 1-7, 2012.
- [35] Z. A. Kadhum, «Equalize The Histogram Equalization for Image enhancement,» *Journal of Kufa for Mathematics and Computer*, vol. 1, nº 1, pp. 14-21, 2012.
- [36] L. Hong, Y. Wan y a. A. Jain, «Fingerprint Image Enhancement: Algorithm and Performance Evaluation,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, nº 8, pp. 777-789, 1998.

ANEXOS

Anexo 1. Resolución ampliación de vigencia del proyecto de investigación.

Anexo 2. Matriz de revisión de artículos.

Anexo 3. Registro de tiempo de procesamiento y Matriz de confusión de la implementación de algoritmos.

Anexo 4. Consentimiento Informado.

Anexo 01

Resolución ampliación de vigencia del proyecto de investigación.

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N°1000-A-2021/FIAU-USS

Pimentel, 11 de noviembre de 2021

VISTO:

El oficio N° 0359-2021/FIAU-IS-USS de fecha 14 de octubre de 2021, de la Dirección de Escuela profesional de INGENIERÍA DE SISTEMAS con el que remite el Acta de reunión N°0610-2021 del Comité de investigación de la referida Escuela profesional, acerca de la Tesis presentada por estudiantes del Programa de estudios de INGENIERÍA DE SISTEMAS, y;

CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: "Los temas de trabajo de investigación, trabajo académico y *tesis* son aprobados por el Comité de Investigación y derivados a la facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El periodo de vigencia de los mismos será de dos años, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma.

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24° señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; es individual o en pares para obtener un título profesional. Asimismo, en su artículo 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C."

Que, mediante documentos de vistos, el Comité de investigación de la referida Escuela profesional acordó aprobar la ampliación de la vigencia de las tesis que se detallan en el Acta de reunión N°0610-2021, a cargo de estudiantes del Programa de estudios INGENIERÍA DE SISTEMAS, **hasta el 6 de octubre de 2023.**

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

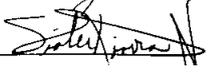
SE RESUELVE:

ARTÍCULO ÚNICO: AMPLIAR VIGENCIA, de la Tesis a cargo de los estudiantes del Programa de estudios de **INGENIERÍA DE SISTEMAS** que se detallan en el anexo de la presente Resolución, hasta el 6 de octubre de 2023.

REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE



Mg. Victor Alexei Tuesta Monteza
Decano (e) / Facultad De Ingeniería,
Arquitectura Y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.



MBA María Noelia Sialer Rivera
Secretaría Académica / Facultad de Ingeniería,
Arquitectura y Urbanismo
UNIVERSIDAD SEÑOR DE SIPÁN SAC.

Cc: Interesado, Archivo

FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO
RESOLUCIÓN N°1000-A-2021/FIAU-USS

Pimentel, 11 de noviembre de 2021

ANEXO

N°	APELLIDOS Y NOMBRES	TEMA DE TESIS	FECHA RESOLUCIÓN DE APROBACIÓN/MODIFICACIÓN TEMA DE TESIS/AMPLIACIÓN DE VIGENCIA
1	DIAZ CARRASCO NATIVIDAD ALEJANDRO	EVALUACIÓN DE ALGORITMOS PARA LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS	13-05-2016
2	MENDOZA LINARES JERSSON GERMAN	METODOLOGÍA DE CONVERSIÓN DE APLICACIONES MONOLÍTICAS A MICROSERVICIOS DESPLEGABLE EN LA NUBE PARA PEQUEÑAS EMPRESAS	22-07-2019
3	GONZALEZ FLORES PAUL GUSTAVO	ANÁLISIS COMPARATIVO DE ALGORITMOS DE CLASIFICACIÓN PARA DIAGNOSTICAR TIPOS DE LEUCEMIA INFANTIL	17-11-2020
4	SOPLOPUCO MONJA BRAYAN ALONSO	COMPARACIÓN DE TÉCNICAS CONSTRUCCIÓN DE PROTOTIPOS Y REVISIÓN DE REQUERIMIENTOS PARA REALIZAR UNA CORRECTA VALIDACIÓN DE REQUERIMIENTOS DE SOFTWARE	Año 2018
5	MONTENEGRO GUERRERO VICTOR AGUSTIN	ANALISIS COMPARATIVO DE ALGORITMOS DE MACHINE LEARNING PARA DETECCION DE MALWARE EN APLICACIONES ANDROID	Año 2019

Anexo 02.

Matriz de revisión de Artículos (Papers)						
N°	Origen (País)	Año de publicación	Autores	Título original del artículo	Conferencia/revista/journal/Magazine donde se publicó del artículo	Base de datos donde se encontró el artículo (IEEEExplore, ACM Library, EBSCO, Scopus, ScienceDirect,
1	China	2019	YANG Gaoqiang, LI Sheng, ZHANG Xinpeng	Altered Fingerprint Detection Based on Orientation Field	JOURNAL OF APPLIED SCIENCES—Electronics and Information Engineering Vol37, Nro 6	SCOPUS / JAS.SHU
2	Tokyo, Japan	2018	Yahaya Isah Shehu, Anne James and Vasile Palade	Detecting an Alteration in Biometric Fingerprint Databases	ICDSP 2018: Proceedings of the 2nd International Conference on Digital Signal Processing	SCOPUS / ACM
3	India	2018	A. Vinoth, S. Saravanakumar	Region based Minutiae Mass Measure for Efficient Finger Print Forgery Detection in Health Care System	International Journal of Recent Technology and Engineering	SCOPUS / IJRTE
4	Alemania	2018	Carsten Gottschlich, Benjamin Tamsy, and Stephan Huckemann	Perfect Fingerprint Orientation Fields by Locally Adaptive Global Models	Computer Vision and Pattern Recognition (cs.CV) IET Biometrics, vol. 6, no.3, pp. 183-190, May 2017	SCOPUS / arxiv
5	India	2017	Sharad Pratap Singh; Shahanaz Ayub; Saini Jp	Minutiae Extraction Based Matching Between Normal And Altered Fingerprint	Journal of Industrial Pollution Control	SCOPUS / icontrolpollution
6	India	2017	Ms. Vidya Ramesh Patil; Prof. (Dr). B. D. Phulpagar	Efficient Altered Fingerprint Analysis and Rectification of Distorted Fingerprint	International Research Journal of Engineering and Technology (IRJET)	IRJET
7	Limassol, Cyprus	2016	Rudolf Haraksim, Alexandre Anthonioz, Christophe Champod, Martin Olsen, John Ellingsgaard, Christoph Busch	Altered fingerprint detection – algorithm performance evaluation	2016 4th International Conference on Biometrics and Forensics (IWBF)	SCOPUS / IEEE
8	Alemania	2016	Serena Papi; Matteo Ferrara; Davide Maltoni; Alexandre Anthonioz	On the generation of synthetic Fingerprint alterations	2016 International Conference of the Biometrics Special Interest Group (BIOSIG)	SCOPUS/IEEE
9	India	2015	T.R. Anoop and M.G. Mini	Two Stage Framework For Altered Fingerprint Matching	ICTACT Journal on Image and Video Processing	ictactjournals
10	India	2015	T.R. Anoop and M.G. Mini	Altered Fingerprint Matching Using Ridge Texture and Frequency in the Unaltered Region	Bonfring International Journal of Advances in Image Processing 5(2):06-09	bonfring
11	Zagreb, Croatia	2015	Carsten Gottschlich, Anna Mikaelyan, y Martin Aastrup Olsen, z Josef Bigun, y and Christoph Busch	Improving Fingerprint Alteration Detection	2015 9th International Symposium on Image and Signal Processing and Analysis (ISPA)	SCOPUS / IEEE
12	India	2015	R.K. Nagthane*, H. S. Fadewar	Matching of Altered Fingerprint using Minutiae Technique	International Journal of Advanced Research in Computer Science and Software Engineering	ijarcsse
13	India	2014	Gayathri Mohan	Embedded Fingerprint Matching Mechanism in HIS for Health Care Service	IOSR Journal of Computer Engineering (IOSR-JCE)	iosrjournals
14	China	2014	Suhang Chen, Sheng Chang*, Qijun Huang, Jin He, Hao Wang, Qiangui Huang	SVM-Based Synthetic Fingerprint Discrimination Algorithm and Quantitative Optimization Strategy	PLoS ONE 9(10): e111099. doi:10.1371/journal.pone.0111099	SCOPUS / PLoS ONE 9(10)
15	India	2014	Dr. K. Latha, C. Manikandan	Critical Analysis and Detection of Altered Fingerprints	K.L.N. College of Engineering, Madurai, Tamil Nadu, India / International Conference on Innovations in Engineering and Technology (ICIET'14)	SCOPUS / Research and Review International Journals
16	Korea del Sur	2012	Soweon Yoon; Jianjiang Feng; Anil K. Jain	Altered Fingerprints: Analysis and Detection	IEEE Transactions on Pattern Analysis and Machine Intelligence (Volume: 34, Issue: 3, March 2012)	SCOPUS / IEEE

Anexo 3.

Registro de tiempo de procesamiento y Matriz de confusión de la implementación de algoritmos.

REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DE EXTRACCION DE MINUCIAS EN IMÁGENES DE HUELLAS DACTILARES OBLITERADAS

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
001	I001_01_Obl.BMP	1657556629.69	1657556632.85	3155.85	3.156
002	I001_02_Obl.BMP	1657556632.85	1657556633.37	518.89	0.519
003	I001_03_Obl.BMP	1657556633.37	1657556633.88	517.68	0.518
004	I001_04_Obl.BMP	1657556633.88	1657556634.46	579.18	0.579
005	I001_05_Obl.BMP	1657556634.46	1657556635.23	766.68	0.767
006	I001_06_Obl.BMP	1657556635.23	1657556635.76	531.13	0.531
007	I001_07_Obl.BMP	1657556635.76	1657556636.23	469.84	0.470
008	I001_08_Obl.BMP	1657556636.23	1657556636.68	453.01	0.453
009	I001_09_Obl.BMP	1657556636.68	1657556637.33	641.73	0.642
010	I001_10_Obl.BMP	1657556637.33	1657556639.06	1736.08	1.736
011	I002_01_Obl.BMP	1657556639.06	1657556639.60	533.98	0.534
012	I002_02_Obl.BMP	1657556639.60	1657556640.05	453.02	0.453
013	I002_03_Obl.BMP	1657556640.05	1657556640.54	490.69	0.491
014	I002_04_Obl.BMP	1657556640.54	1657556641.04	499.88	0.500
015	I002_05_Obl.BMP	1657556641.04	1657556641.48	438.60	0.439
016	I002_06_Obl.BMP	1657556641.48	1657556641.96	484.26	0.484
017	I002_07_Obl.BMP	1657556641.96	1657556642.42	458.80	0.459
018	I002_08_Obl.BMP	1657556642.42	1657556642.92	499.88	0.500
019	I002_09_Obl.BMP	1657556642.92	1657556643.42	503.29	0.503
020	I002_10_Obl.BMP	1657556643.42	1657556643.99	565.68	0.566
021	I003_01_Obl.BMP	1657556644.01	1657556644.60	594.74	0.595
022	I003_02_Obl.BMP	1657556644.60	1657556645.27	669.87	0.670
023	I003_03_Obl.BMP	1657556645.27	1657556646.18	908.09	0.908
024	I003_04_Obl.BMP	1657556646.18	1657556647.30	1119.10	1.119
025	I003_05_Obl.BMP	1657556647.30	1657556648.10	797.89	0.798
026	I003_06_Obl.BMP	1657556648.10	1657556648.57	472.94	0.473
027	I003_07_Obl.BMP	1657556648.58	1657556649.08	500.96	0.501
028	I003_08_Obl.BMP	1657556649.08	1657556649.63	546.81	0.547
029	I003_09_Obl.BMP	1657556649.63	1657556650.40	766.64	0.767
030	I003_10_Obl.BMP	1657556650.40	1657556651.04	640.47	0.640

**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DE EXTRACCION DE
MINUCIAS EN IMÁGENES DE HUELLAS DACTILARES OBLITERADAS**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
031	I004_01_Obl.BMP	1657556651.04	1657556651.49	454.21	0.454
032	I004_02_Obl.BMP	1657556651.49	1657556652.04	546.78	0.547
033	I004_03_Obl.BMP	1657556652.06	1657556652.68	625.97	0.626
034	I004_04_Obl.BMP	1657556652.68	1657556653.29	612.56	0.613
035	I004_05_Obl.BMP	1657556653.29	1657556653.83	538.18	0.538
036	I004_06_Obl.BMP	1657556653.83	1657556654.29	454.14	0.454
037	I004_07_Obl.BMP	1657556654.29	1657556654.85	563.55	0.564
038	I004_08_Obl.BMP	1657556654.85	1657556655.47	616.85	0.617
039	I004_09_Obl.BMP	1657556655.47	1657556655.94	468.61	0.469
040	I004_10_Obl.BMP	1657556655.94	1657556656.53	593.61	0.594
041	I005_01_Obl.BMP	1657556656.53	1657556657.00	468.67	0.469
042	I005_02_Obl.BMP	1657556657.00	1657556657.58	577.96	0.578
043	I005_03_Obl.BMP	1657556657.58	1657556658.25	671.74	0.672
044	I005_04_Obl.BMP	1657556658.25	1657556659.26	1015.36	1.015
045	I005_05_Obl.BMP	1657556659.26	1657556659.84	578.02	0.578
046	I005_06_Obl.BMP	1657556659.84	1657556660.26	421.78	0.422
047	I005_07_Obl.BMP	1657556660.26	1657556660.79	531.13	0.531
048	I005_08_Obl.BMP	1657556660.79	1657556661.57	781.06	0.781
049	I005_09_Obl.BMP	1657556661.57	1657556662.61	1031.71	1.032
050	I005_10_Obl.BMP	1657556662.61	1657556663.29	687.34	0.687
051	I006_01_Obl.BMP	1657556663.29	1657556663.90	609.26	0.609
052	I006_02_Obl.BMP	1657556663.90	1657556664.39	484.27	0.484
053	I006_03_Obl.BMP	1657556664.39	1657556664.89	499.88	0.500
054	I006_04_Obl.BMP	1657556664.89	1657556665.39	499.85	0.500
055	I006_05_Obl.BMP	1657556665.39	1657556665.92	531.12	0.531
056	I006_06_Obl.BMP	1657556665.92	1657556666.43	515.54	0.516
057	I006_07_Obl.BMP	1657556666.43	1657556667.26	827.93	0.828
058	I006_08_Obl.BMP	1657556667.26	1657556667.78	515.50	0.515
059	I006_09_Obl.BMP	1657556667.78	1657556668.25	468.64	0.469
060	I006_10_Obl.BMP	1657556668.26	1657556669.20	937.31	0.937
061	I007_01_Obl.BMP	1657556669.20	1657556669.67	468.64	0.469
062	I007_02_Obl.BMP	1657556669.67	1657556670.17	499.89	0.500
063	I007_03_Obl.BMP	1657556670.17	1657556670.89	718.55	0.719
064	I007_04_Obl.BMP	1657556670.89	1657556671.37	485.14	0.485

**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DE EXTRACCION DE
MINUCIAS EN IMÁGENES DE HUELLAS DACTILARES OBLITERADAS**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
065	I007_05_Obl.BMP	1657556671.37	1657556671.87	499.88	0.500
066	I007_06_Obl.BMP	1657556671.87	1657556672.35	484.26	0.484
067	I007_07_Obl.BMP	1657556672.35	1657556672.81	453.05	0.453
068	I007_08_Obl.BMP	1657556672.81	1657556673.28	468.64	0.469
069	I007_09_Obl.BMP	1657556673.28	1657556673.78	499.88	0.500
070	I007_10_Obl.BMP	1657556673.78	1657556674.35	577.97	0.578
071	I008_01_Obl.BMP	1657556674.35	1657556674.84	484.29	0.484
072	I008_02_Obl.BMP	1657556674.84	1657556675.37	531.09	0.531
073	I008_03_Obl.BMP	1657556675.37	1657556675.90	531.15	0.531
074	I008_04_Obl.BMP	1657556675.90	1657556676.37	468.62	0.469
075	I008_05_Obl.BMP	1657556676.39	1657556676.82	437.40	0.437
076	I008_06_Obl.BMP	1657556676.82	1657556677.23	406.16	0.406
077	I008_07_Obl.BMP	1657556677.23	1657556677.70	468.64	0.469
078	I008_08_Obl.BMP	1657556677.70	1657556678.14	437.99	0.438
079	I008_09_Obl.BMP	1657556678.14	1657556678.67	531.12	0.531
080	I008_10_Obl.BMP	1657556678.67	1657556679.20	531.13	0.531
081	I009_01_Obl.BMP	1657556679.20	1657556679.73	531.10	0.531
082	I009_02_Obl.BMP	1657556679.73	1657556680.20	468.66	0.469
083	I009_03_Obl.BMP	1657556680.20	1657556680.68	484.27	0.484
084	I009_04_Obl.BMP	1657556680.68	1657556681.15	468.63	0.469
085	I009_05_Obl.BMP	1657556681.15	1657556681.65	499.89	0.500
086	I009_06_Obl.BMP	1657556681.65	1657556682.15	499.89	0.500
087	I009_07_Obl.BMP	1657556682.15	1657556682.62	468.63	0.469
088	I009_08_Obl.BMP	1657556682.62	1657556683.06	437.41	0.437
089	I009_09_Obl.BMP	1657556683.06	1657556683.68	624.84	0.625
090	I009_10_Obl.BMP	1657556683.68	1657556684.21	531.13	0.531
091	I010_01_Obl.BMP	1657556684.21	1657556684.66	453.02	0.453
092	I010_02_Obl.BMP	1657556684.66	1657556685.06	390.53	0.391
093	I010_03_Obl.BMP	1657556685.06	1657556685.48	421.75	0.422
094	I010_04_Obl.BMP	1657556685.48	1657556686.01	531.15	0.531
095	I010_05_Obl.BMP	1657556686.01	1657556686.56	546.75	0.547
096	I010_06_Obl.BMP	1657556686.56	1657556687.07	515.73	0.516
097	I010_07_Obl.BMP	1657556687.07	1657556687.48	407.59	0.408
098	I010_08_Obl.BMP	1657556687.48	1657556687.99	515.48	0.515

**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DE EXTRACCION DE
MINUCIAS EN IMÁGENES DE HUELLAS DACTILARES OBLITERADAS**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
099	I010_09_Obl.BMP	1657556687.99	1657556688.45	453.01	0.453
100	I010_10_Obl.BMP	1657556688.45	1657556688.90	453.05	0.453
TIEMPO PROMEDIO (SEG.)					0.591

**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DE EXTRACCION DE
MINUCIAS EN IMÁGENES DE HUELLAS DACTILARES EN CORTE Z**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
001	I001_01_Zcut.BMP	1,657,556,911.16	1,657,556,914.41	3248.54	3.249
002	I001_02_Zcut.BMP	1,657,556,914.41	1,657,556,914.94	531.13	0.531
003	I001_03_Zcut.BMP	1,657,556,914.96	1,657,556,915.51	556.43	0.556
004	I001_04_Zcut.BMP	1,657,556,915.51	1,657,556,916.03	515.53	0.516
005	I001_05_Zcut.BMP	1,657,556,916.03	1,657,556,916.75	719.67	0.720
006	I001_06_Zcut.BMP	1,657,556,916.75	1,657,556,917.31	563.51	0.564
007	I001_07_Zcut.BMP	1,657,556,917.31	1,657,556,917.89	578.51	0.579
008	I001_08_Zcut.BMP	1,657,556,917.89	1,657,556,918.35	456.19	0.456
009	I001_09_Zcut.BMP	1,657,556,918.35	1,657,556,919.00	658.21	0.658
010	I001_10_Zcut.BMP	1,657,556,919.00	1,657,556,919.74	735.31	0.735
011	I002_01_Zcut.BMP	1,657,556,919.74	1,657,556,920.18	438.65	0.439
012	I002_02_Zcut.BMP	1,657,556,920.18	1,657,556,920.62	437.36	0.437
013	I002_03_Zcut.BMP	1,657,556,920.62	1,657,556,921.08	469.76	0.470
014	I002_04_Zcut.BMP	1,657,556,921.08	1,657,556,921.54	453.03	0.453
015	I002_05_Zcut.BMP	1,657,556,921.54	1,657,556,922.10	563.49	0.563
016	I002_06_Zcut.BMP	1,657,556,922.10	1,657,556,922.56	460.68	0.461
017	I002_07_Zcut.BMP	1,657,556,922.56	1,657,556,923.14	582.82	0.583
018	I002_08_Zcut.BMP	1,657,556,923.14	1,657,556,923.63	488.65	0.489
019	I002_09_Zcut.BMP	1,657,556,923.63	1,657,556,924.13	501.08	0.501
020	I002_10_Zcut.BMP	1,657,556,924.13	1,657,556,924.68	547.95	0.548
021	I003_01_Zcut.BMP	1,657,556,924.68	1,657,556,925.17	485.47	0.485
022	I003_02_Zcut.BMP	1,657,556,925.17	1,657,556,925.84	671.72	0.672
023	I003_03_Zcut.BMP	1,657,556,925.84	1,657,556,926.65	813.41	0.813
024	I003_04_Zcut.BMP	1,657,556,926.65	1,657,556,927.90	1251.10	1.251
025	I003_05_Zcut.BMP	1,657,556,927.90	1,657,556,928.49	588.28	0.588
026	I003_06_Zcut.BMP	1,657,556,928.49	1,657,556,929.03	534.18	0.534
027	I003_07_Zcut.BMP	1,657,556,929.03	1,657,556,929.54	516.67	0.517
028	I003_08_Zcut.BMP	1,657,556,929.54	1,657,556,930.09	549.08	0.549

**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DE EXTRACCION DE
MINUCIAS EN IMÁGENES DE HUELLAS DACTILARES EN CORTE Z**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
029	I003_09_Zcut.BMP	1,657,556,930.09	1,657,556,930.86	765.66	0.766
030	I003_10_Zcut.BMP	1,657,556,930.86	1,657,556,931.45	594.80	0.595
031	I004_01_Zcut.BMP	1,657,556,931.45	1,657,556,931.98	531.12	0.531
032	I004_02_Zcut.BMP	1,657,556,931.98	1,657,556,932.47	483.51	0.484
033	I004_03_Zcut.BMP	1,657,556,932.47	1,657,556,933.17	704.04	0.704
034	I004_04_Zcut.BMP	1,657,556,933.17	1,657,556,934.05	880.22	0.880
035	I004_05_Zcut.BMP	1,657,556,934.05	1,657,556,934.69	641.69	0.642
036	I004_06_Zcut.BMP	1,657,556,934.69	1,657,556,935.13	438.59	0.439
037	I004_07_Zcut.BMP	1,657,556,935.15	1,657,556,935.71	562.98	0.563
038	I004_08_Zcut.BMP	1,657,556,935.71	1,657,556,936.23	516.74	0.517
039	I004_09_Zcut.BMP	1,657,556,936.23	1,657,556,936.73	499.88	0.500
040	I004_10_Zcut.BMP	1,657,556,936.73	1,657,556,937.28	548.24	0.548
041	I005_01_Zcut.BMP	1,657,556,937.28	1,657,556,937.78	499.88	0.500
042	I005_02_Zcut.BMP	1,657,556,937.79	1,657,556,938.39	596.86	0.597
043	I005_03_Zcut.BMP	1,657,556,938.39	1,657,556,939.05	664.30	0.664
044	I005_04_Zcut.BMP	1,657,556,939.05	1,657,556,939.79	735.39	0.735
045	I005_05_Zcut.BMP	1,657,556,939.79	1,657,556,940.41	626.09	0.626
046	I005_06_Zcut.BMP	1,657,556,940.41	1,657,556,941.09	674.12	0.674
047	I005_07_Zcut.BMP	1,657,556,941.09	1,657,556,941.57	484.29	0.484
048	I005_08_Zcut.BMP	1,657,556,941.59	1,657,556,942.42	829.15	0.829
049	I005_09_Zcut.BMP	1,657,556,942.42	1,657,556,943.27	849.64	0.850
050	I005_10_Zcut.BMP	1,657,556,943.27	1,657,556,943.82	552.79	0.553
051	I006_01_Zcut.BMP	1,657,556,943.82	1,657,556,944.58	756.25	0.756
052	I006_02_Zcut.BMP	1,657,556,944.58	1,657,556,945.39	813.40	0.813
053	I006_03_Zcut.BMP	1,657,556,945.39	1,657,556,945.86	468.63	0.469
054	I006_04_Zcut.BMP	1,657,556,945.86	1,657,556,946.34	485.49	0.485
055	I006_05_Zcut.BMP	1,657,556,946.34	1,657,556,946.92	577.99	0.578
056	I006_06_Zcut.BMP	1,657,556,946.92	1,657,556,947.47	544.00	0.544
057	I006_07_Zcut.BMP	1,657,556,947.47	1,657,556,948.14	672.87	0.673
058	I006_08_Zcut.BMP	1,657,556,948.14	1,657,556,948.68	543.73	0.544
059	I006_09_Zcut.BMP	1,657,556,948.68	1,657,556,949.12	438.55	0.439
060	I006_10_Zcut.BMP	1,657,556,949.14	1,657,556,950.06	922.39	0.922
061	I007_01_Zcut.BMP	1,657,556,950.06	1,657,556,950.65	594.75	0.595
062	I007_02_Zcut.BMP	1,657,556,950.65	1,657,556,951.20	547.81	0.548
063	I007_03_Zcut.BMP	1,657,556,951.20	1,657,556,951.72	521.64	0.522
064	I007_04_Zcut.BMP	1,657,556,951.72	1,657,556,952.20	480.70	0.481
065	I007_05_Zcut.BMP	1,657,556,952.20	1,657,556,952.61	411.47	0.411
066	I007_06_Zcut.BMP	1,657,556,952.61	1,657,556,953.02	401.89	0.402

**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DE EXTRACCION DE
MINUCIAS EN IMÁGENES DE HUELLAS DACTILARES EN CORTE Z**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
067	I007_07_Zcut.BMP	1,657,556,953.02	1,657,556,953.52	501.75	0.502
068	I007_08_Zcut.BMP	1,657,556,953.52	1,657,556,954.04	517.61	0.518
069	I007_09_Zcut.BMP	1,657,556,954.04	1,657,556,954.54	501.08	0.501
070	I007_10_Zcut.BMP	1,657,556,954.54	1,657,556,955.04	499.88	0.500
071	I008_01_Zcut.BMP	1,657,556,955.04	1,657,556,955.48	445.17	0.445
072	I008_02_Zcut.BMP	1,657,556,955.48	1,657,556,955.95	468.63	0.469
073	I008_03_Zcut.BMP	1,657,556,955.95	1,657,556,956.52	563.59	0.564
074	I008_04_Zcut.BMP	1,657,556,956.52	1,657,556,957.05	533.06	0.533
075	I008_05_Zcut.BMP	1,657,556,957.05	1,657,556,957.45	403.31	0.403
076	I008_06_Zcut.BMP	1,657,556,957.45	1,657,556,957.92	468.60	0.469
077	I008_07_Zcut.BMP	1,657,556,957.92	1,657,556,958.55	630.32	0.630
078	I008_08_Zcut.BMP	1,657,556,958.55	1,657,556,959.40	848.65	0.849
079	I008_09_Zcut.BMP	1,657,556,959.40	1,657,556,959.97	570.94	0.571
080	I008_10_Zcut.BMP	1,657,556,959.97	1,657,556,960.63	658.73	0.659
081	I009_01_Zcut.BMP	1,657,556,960.63	1,657,556,961.43	797.83	0.798
082	I009_02_Zcut.BMP	1,657,556,961.43	1,657,556,961.94	515.48	0.515
083	I009_03_Zcut.BMP	1,657,556,961.94	1,657,556,962.59	650.87	0.651
084	I009_04_Zcut.BMP	1,657,556,962.59	1,657,556,963.22	622.37	0.622
085	I009_05_Zcut.BMP	1,657,556,963.22	1,657,556,963.77	556.41	0.556
086	I009_06_Zcut.BMP	1,657,556,963.77	1,657,556,964.38	610.35	0.610
087	I009_07_Zcut.BMP	1,657,556,964.38	1,657,556,964.99	610.63	0.611
088	I009_08_Zcut.BMP	1,657,556,964.99	1,657,556,965.51	521.18	0.521
089	I009_09_Zcut.BMP	1,657,556,965.52	1,657,556,966.07	556.92	0.557
090	I009_10_Zcut.BMP	1,657,556,966.07	1,657,556,966.66	582.86	0.583
091	I010_01_Zcut.BMP	1,657,556,966.66	1,657,556,967.09	438.56	0.439
092	I010_02_Zcut.BMP	1,657,556,967.09	1,657,556,967.70	610.45	0.610
093	I010_03_Zcut.BMP	1,657,556,967.70	1,657,556,968.14	439.75	0.440
094	I010_04_Zcut.BMP	1,657,556,968.14	1,657,556,968.62	472.97	0.473
095	I010_05_Zcut.BMP	1,657,556,968.62	1,657,556,969.17	547.98	0.548
096	I010_06_Zcut.BMP	1,657,556,969.17	1,657,556,969.60	438.61	0.439
097	I010_07_Zcut.BMP	1,657,556,969.60	1,657,556,970.29	688.51	0.689
098	I010_08_Zcut.BMP	1,657,556,970.29	1,657,556,970.78	484.29	0.484
099	I010_09_Zcut.BMP	1,657,556,970.78	1,657,556,971.20	422.97	0.423
100	I010_10_Zcut.BMP	1,657,556,971.20	1,657,556,971.81	609.23	0.609
TIEMPO PROMEDIO (SEG.)					0.606

**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DE EXTRACCION DE
MINUCIAS EN IMÁGENES DE HUELLAS DACTILARES EN ROTACION CENTRAL**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
001	I001_01_CR.BMP	1657557277.81	1657557281.08	3262.52	3.263
002	I001_02_CR.BMP	1657557281.08	1657557281.61	532.27	0.532
003	I001_03_CR.BMP	1657557281.61	1657557282.16	547.87	0.548
004	I001_04_CR.BMP	1657557282.16	1657557282.67	515.50	0.516
005	I001_05_CR.BMP	1657557282.67	1657557283.32	646.41	0.646
006	I001_06_CR.BMP	1657557283.32	1657557283.81	491.37	0.491
007	I001_07_CR.BMP	1657557283.81	1657557284.33	516.62	0.517
008	I001_08_CR.BMP	1657557284.33	1657557284.87	547.81	0.548
009	I001_09_CR.BMP	1657557284.87	1657557285.48	610.35	0.610
010	I001_10_CR.BMP	1657557285.48	1657557286.20	720.22	0.720
011	I002_01_CR.BMP	1657557286.20	1657557286.72	515.50	0.516
012	I002_02_CR.BMP	1657557286.72	1657557287.17	454.14	0.454
013	I002_03_CR.BMP	1657557287.17	1657557287.60	421.81	0.422
014	I002_04_CR.BMP	1657557287.61	1657557288.05	437.38	0.437
015	I002_05_CR.BMP	1657557288.05	1657557288.54	490.14	0.490
016	I002_06_CR.BMP	1657557288.54	1657557288.99	453.02	0.453
017	I002_07_CR.BMP	1657557288.99	1657557289.51	516.60	0.517
018	I002_08_CR.BMP	1657557289.51	1657557289.96	454.13	0.454
019	I002_09_CR.BMP	1657557289.96	1657557290.62	657.22	0.657
020	I002_10_CR.BMP	1657557290.62	1657557291.08	461.77	0.462
021	I003_01_CR.BMP	1657557291.08	1657557291.63	546.76	0.547
022	I003_02_CR.BMP	1657557291.63	1657557292.27	641.59	0.642
023	I003_03_CR.BMP	1657557292.29	1657557293.11	828.03	0.828
024	I003_04_CR.BMP	1657557293.11	1657557293.84	724.50	0.724
025	I003_05_CR.BMP	1657557293.85	1657557294.45	591.45	0.591
026	I003_06_CR.BMP	1657557294.45	1657557295.07	623.93	0.624
027	I003_07_CR.BMP	1657557295.07	1657557295.62	547.85	0.548
028	I003_08_CR.BMP	1657557295.62	1657557296.24	625.95	0.626
029	I003_09_CR.BMP	1657557296.24	1657557296.95	702.96	0.703
030	I003_10_CR.BMP	1657557296.95	1657557297.52	568.69	0.569
031	I004_01_CR.BMP	1657557297.52	1657557298.00	488.05	0.488
032	I004_02_CR.BMP	1657557298.00	1657557298.57	567.73	0.568
033	I004_03_CR.BMP	1657557298.57	1657557299.23	658.40	0.658
034	I004_04_CR.BMP	1657557299.23	1657557299.88	645.94	0.646
035	I004_05_CR.BMP	1657557299.88	1657557300.61	735.35	0.735
036	I004_06_CR.BMP	1657557300.61	1657557301.21	595.30	0.595
037	I004_07_CR.BMP	1657557301.21	1657557301.77	559.81	0.560
038	I004_08_CR.BMP	1657557301.77	1657557302.39	626.01	0.626

**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DE EXTRACCION DE
MINUCIAS EN IMÁGENES DE HUELLAS DACTILARES EN ROTACION CENTRAL**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
039	I004_09_CR.BMP	1657557302.39	1657557302.96	563.93	0.564
040	I004_10_CR.BMP	1657557302.96	1657557303.57	614.94	0.615
041	I005_01_CR.BMP	1657557303.57	1657557304.04	468.64	0.469
042	I005_02_CR.BMP	1657557304.04	1657557304.72	675.54	0.676
043	I005_03_CR.BMP	1657557304.72	1657557305.39	672.94	0.673
044	I005_04_CR.BMP	1657557305.39	1657557306.05	656.06	0.656
045	I005_05_CR.BMP	1657557306.05	1657557306.70	657.21	0.657
046	I005_06_CR.BMP	1657557306.70	1657557307.20	499.98	0.500
047	I005_07_CR.BMP	1657557307.20	1657557307.63	423.06	0.423
048	I005_08_CR.BMP	1657557307.64	1657557308.24	594.72	0.595
049	I005_09_CR.BMP	1657557308.24	1657557308.95	708.84	0.709
050	I005_10_CR.BMP	1657557308.95	1657557309.43	485.40	0.485
051	I006_01_CR.BMP	1657557309.43	1657557309.93	499.86	0.500
052	I006_02_CR.BMP	1657557309.93	1657557310.46	532.24	0.532
053	I006_03_CR.BMP	1657557310.46	1657557310.93	468.64	0.469
054	I006_04_CR.BMP	1657557310.93	1657557311.40	469.71	0.470
055	I006_05_CR.BMP	1657557311.40	1657557311.92	515.51	0.516
056	I006_06_CR.BMP	1657557311.92	1657557312.38	462.44	0.462
057	I006_07_CR.BMP	1657557312.38	1657557313.13	750.95	0.751
058	I006_08_CR.BMP	1657557313.13	1657557313.62	489.37	0.489
059	I006_09_CR.BMP	1657557313.62	1657557314.07	453.01	0.453
060	I006_10_CR.BMP	1657557314.07	1657557315.04	969.64	0.970
061	I007_01_CR.BMP	1657557315.04	1657557315.45	407.26	0.407
062	I007_02_CR.BMP	1657557315.45	1657557316.01	562.37	0.562
063	I007_03_CR.BMP	1657557316.01	1657557316.52	508.26	0.508
064	I007_04_CR.BMP	1657557316.52	1657557317.02	499.88	0.500
065	I007_05_CR.BMP	1657557317.02	1657557317.52	502.20	0.502
066	I007_06_CR.BMP	1657557317.52	1657557317.91	390.53	0.391
067	I007_07_CR.BMP	1657557317.91	1657557318.46	550.41	0.550
068	I007_08_CR.BMP	1657557318.46	1657557318.93	470.45	0.470
069	I007_09_CR.BMP	1657557318.93	1657557319.50	563.45	0.563
070	I007_10_CR.BMP	1657557319.50	1657557320.00	501.02	0.501
071	I008_01_CR.BMP	1657557320.00	1657557320.42	422.88	0.423
072	I008_02_CR.BMP	1657557320.42	1657557320.84	421.78	0.422
073	I008_03_CR.BMP	1657557320.84	1657557321.31	469.78	0.470
074	I008_04_CR.BMP	1657557321.31	1657557321.80	484.26	0.484
075	I008_05_CR.BMP	1657557321.80	1657557322.28	485.37	0.485
076	I008_06_CR.BMP	1657557322.28	1657557322.67	390.53	0.391

**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DE EXTRACCION DE
MINUCIAS EN IMÁGENES DE HUELLAS DACTILARES EN ROTACION CENTRAL**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
077	I008_07_CR.BMP	1657557322.67	1657557323.13	454.13	0.454
078	I008_08_CR.BMP	1657557323.13	1657557323.62	490.08	0.490
079	I008_09_CR.BMP	1657557323.62	1657557324.09	469.72	0.470
080	I008_10_CR.BMP	1657557324.09	1657557324.59	499.92	0.500
081	I009_01_CR.BMP	1657557324.59	1657557325.18	594.73	0.595
082	I009_02_CR.BMP	1657557325.18	1657557325.70	515.51	0.516
083	I009_03_CR.BMP	1657557325.70	1657557326.17	469.75	0.470
084	I009_04_CR.BMP	1657557326.17	1657557326.62	453.02	0.453
085	I009_05_CR.BMP	1657557326.62	1657557327.06	442.60	0.443
086	I009_06_CR.BMP	1657557327.06	1657557327.59	524.39	0.524
087	I009_07_CR.BMP	1657557327.59	1657557328.09	500.94	0.501
088	I009_08_CR.BMP	1657557328.09	1657557328.59	505.79	0.506
089	I009_09_CR.BMP	1657557328.59	1657557329.08	484.26	0.484
090	I009_10_CR.BMP	1657557329.08	1657557329.64	564.66	0.565
091	I010_01_CR.BMP	1657557329.64	1657557330.02	376.43	0.376
092	I010_02_CR.BMP	1657557330.02	1657557330.55	532.25	0.532
093	I010_03_CR.BMP	1657557330.55	1657557331.00	453.02	0.453
094	I010_04_CR.BMP	1657557331.00	1657557331.49	485.47	0.485
095	I010_05_CR.BMP	1657557331.49	1657557332.04	546.75	0.547
096	I010_06_CR.BMP	1657557332.04	1657557332.44	407.22	0.407
097	I010_07_CR.BMP	1657557332.44	1657557332.88	437.43	0.437
098	I010_08_CR.BMP	1657557332.88	1657557333.32	438.88	0.439
099	I010_09_CR.BMP	1657557333.32	1657557333.76	440.35	0.440
100	I010_10_CR.BMP	1657557333.76	1657557334.32	563.51	0.564
TIEMPO PROMEDIO (SEG.)					0.564

**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DEL CAMPO DE
ORIENTACION EN IMÁGENES DE HUELLAS DACTILARES OBLITERADAS**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
001	I001_01_Obl.BMP	1657556171.53	1657556174.52	2990.85	2.991
002	I001_02_Obl.BMP	1657556174.56	1657556174.76	202.98	0.203
003	I001_03_Obl.BMP	1657556174.77	1657556174.98	203.07	0.203

**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DEL CAMPO DE
ORIENTACION EN IMÁGENES DE HUELLAS DACTILARES OBLITERADAS**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
004	I001_04_Obl.BMP	1657556175.01	1657556175.21	203.98	0.204
005	I001_05_Obl.BMP	1657556175.24	1657556175.46	218.71	0.219
006	I001_06_Obl.BMP	1657556175.48	1657556175.68	203.07	0.203
007	I001_07_Obl.BMP	1657556175.68	1657556175.88	203.09	0.203
008	I001_08_Obl.BMP	1657556176.02	1657556176.24	219.76	0.220
009	I001_09_Obl.BMP	1657556176.26	1657556176.47	207.82	0.208
010	I001_10_Obl.BMP	1657556176.55	1657556176.75	199.76	0.200
011	I002_01_Obl.BMP	1657556176.76	1657556176.96	203.11	0.203
012	I002_02_Obl.BMP	1657556176.98	1657556177.17	188.69	0.189
013	I002_03_Obl.BMP	1657556177.22	1657556177.42	203.07	0.203
014	I002_04_Obl.BMP	1657556177.42	1657556177.62	203.05	0.203
015	I002_05_Obl.BMP	1657556177.65	1657556177.84	187.46	0.187
016	I002_06_Obl.BMP	1657556177.86	1657556178.04	187.46	0.187
017	I002_07_Obl.BMP	1657556178.06	1657556178.25	188.59	0.189
018	I002_08_Obl.BMP	1657556178.26	1657556178.47	205.69	0.206
019	I002_09_Obl.BMP	1657556178.48	1657556178.69	207.30	0.207
020	I002_10_Obl.BMP	1657556178.69	1657556178.88	188.68	0.189
021	I003_01_Obl.BMP	1657556178.90	1657556179.08	188.72	0.189
022	I003_02_Obl.BMP	1657556179.10	1657556179.29	187.49	0.187
023	I003_03_Obl.BMP	1657556179.32	1657556179.52	203.08	0.203
024	I003_04_Obl.BMP	1657556179.52	1657556179.73	204.14	0.204
025	I003_05_Obl.BMP	1657556179.73	1657556179.93	203.04	0.203
026	I003_06_Obl.BMP	1657556179.94	1657556180.15	204.28	0.204
027	I003_07_Obl.BMP	1657556180.15	1657556180.34	187.46	0.187
028	I003_08_Obl.BMP	1657556180.38	1657556180.57	187.46	0.187
029	I003_09_Obl.BMP	1657556180.59	1657556180.79	203.02	0.203
030	I003_10_Obl.BMP	1657556180.79	1657556180.99	203.08	0.203
031	I004_01_Obl.BMP	1657556180.99	1657556181.20	204.35	0.204
032	I004_02_Obl.BMP	1657556181.23	1657556181.42	187.46	0.187
033	I004_03_Obl.BMP	1657556181.46	1657556181.68	220.76	0.221
034	I004_04_Obl.BMP	1657556181.72	1657556181.90	187.88	0.188
035	I004_05_Obl.BMP	1657556181.93	1657556182.12	189.17	0.189
036	I004_06_Obl.BMP	1657556182.15	1657556182.34	187.40	0.187
037	I004_07_Obl.BMP	1657556182.36	1657556182.56	203.04	0.203
038	I004_08_Obl.BMP	1657556182.59	1657556182.80	204.33	0.204
039	I004_09_Obl.BMP	1657556182.80	1657556183.00	203.07	0.203
040	I004_10_Obl.BMP	1657556183.03	1657556183.24	204.29	0.204

**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DEL CAMPO DE
ORIENTACION EN IMÁGENES DE HUELLAS DACTILARES OBLITERADAS**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
041	I005_01_Obl.BMP	1657556183.25	1657556183.46	205.29	0.205
042	I005_02_Obl.BMP	1657556183.49	1657556183.69	205.24	0.205
043	I005_03_Obl.BMP	1657556183.71	1657556183.91	203.08	0.203
044	I005_04_Obl.BMP	1657556183.94	1657556184.15	204.21	0.204
045	I005_05_Obl.BMP	1657556184.15	1657556184.35	203.07	0.203
046	I005_06_Obl.BMP	1657556184.40	1657556184.60	203.07	0.203
047	I005_07_Obl.BMP	1657556184.62	1657556184.82	203.06	0.203
048	I005_08_Obl.BMP	1657556184.82	1657556185.02	203.09	0.203
049	I005_09_Obl.BMP	1657556185.02	1657556185.23	204.32	0.204
050	I005_10_Obl.BMP	1657556185.24	1657556185.44	203.03	0.203
051	I006_01_Obl.BMP	1657556185.46	1657556185.66	203.07	0.203
052	I006_02_Obl.BMP	1657556185.68	1657556185.90	218.69	0.219
053	I006_03_Obl.BMP	1657556185.99	1657556186.18	188.66	0.189
054	I006_04_Obl.BMP	1657556186.21	1657556186.40	187.45	0.187
055	I006_05_Obl.BMP	1657556186.41	1657556186.60	187.49	0.187
056	I006_06_Obl.BMP	1657556186.65	1657556186.84	192.26	0.192
057	I006_07_Obl.BMP	1657556186.86	1657556187.04	187.46	0.187
058	I006_08_Obl.BMP	1657556187.08	1657556187.28	204.25	0.204
059	I006_09_Obl.BMP	1657556187.31	1657556187.51	203.05	0.203
060	I006_10_Obl.BMP	1657556187.53	1657556187.73	203.08	0.203
061	I007_01_Obl.BMP	1657556187.75	1657556187.95	203.07	0.203
062	I007_02_Obl.BMP	1657556187.95	1657556188.16	204.26	0.204
063	I007_03_Obl.BMP	1657556188.20	1657556188.42	220.87	0.221
064	I007_04_Obl.BMP	1657556188.44	1657556188.64	205.14	0.205
065	I007_05_Obl.BMP	1657556188.64	1657556188.85	203.11	0.203
066	I007_06_Obl.BMP	1657556188.85	1657556189.05	203.07	0.203
067	I007_07_Obl.BMP	1657556189.07	1657556189.27	204.24	0.204
068	I007_08_Obl.BMP	1657556189.27	1657556189.47	203.08	0.203
069	I007_09_Obl.BMP	1657556189.50	1657556189.71	203.10	0.203
070	I007_10_Obl.BMP	1657556189.71	1657556189.91	203.08	0.203
071	I008_01_Obl.BMP	1657556189.91	1657556190.10	188.65	0.189
072	I008_02_Obl.BMP	1657556190.15	1657556190.35	203.07	0.203
073	I008_03_Obl.BMP	1657556190.40	1657556190.61	209.77	0.210
074	I008_04_Obl.BMP	1657556190.67	1657556190.89	218.70	0.219
075	I008_05_Obl.BMP	1657556190.89	1657556191.09	205.49	0.205
076	I008_06_Obl.BMP	1657556191.09	1657556191.29	202.06	0.202
077	I008_07_Obl.BMP	1657556191.31	1657556191.53	218.69	0.219

**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DEL CAMPO DE
ORIENTACION EN IMÁGENES DE HUELLAS DACTILARES OBLITERADAS**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
078	I008_08_Obl.BMP	1657556191.53	1657556191.76	234.32	0.234
079	I008_09_Obl.BMP	1657556191.76	1657556191.97	203.05	0.203
080	I008_10_Obl.BMP	1657556192.00	1657556192.24	237.88	0.238
081	I009_01_Obl.BMP	1657556192.25	1657556192.52	270.34	0.270
082	I009_02_Obl.BMP	1657556192.54	1657556192.74	203.08	0.203
083	I009_03_Obl.BMP	1657556192.74	1657556192.99	249.94	0.250
084	I009_04_Obl.BMP	1657556193.01	1657556193.22	217.43	0.217
085	I009_05_Obl.BMP	1657556193.22	1657556193.43	205.25	0.205
086	I009_06_Obl.BMP	1657556193.43	1657556193.63	205.20	0.205
087	I009_07_Obl.BMP	1657556193.66	1657556193.88	218.73	0.219
088	I009_08_Obl.BMP	1657556193.88	1657556194.09	204.26	0.204
089	I009_09_Obl.BMP	1657556194.09	1657556194.28	187.48	0.187
090	I009_10_Obl.BMP	1657556194.29	1657556194.49	203.07	0.203
091	I010_01_Obl.BMP	1657556194.49	1657556194.70	203.08	0.203
092	I010_02_Obl.BMP	1657556194.70	1657556194.95	249.94	0.250
093	I010_03_Obl.BMP	1657556194.95	1657556195.15	204.26	0.204
094	I010_04_Obl.BMP	1657556195.15	1657556195.35	203.08	0.203
095	I010_05_Obl.BMP	1657556195.35	1657556195.56	203.08	0.203
096	I010_06_Obl.BMP	1657556195.56	1657556195.76	203.08	0.203
097	I010_07_Obl.BMP	1657556195.82	1657556196.04	218.54	0.219
098	I010_08_Obl.BMP	1657556196.06	1657556196.25	188.53	0.189
099	I010_09_Obl.BMP	1657556196.29	1657556196.50	203.05	0.203
100	I010_10_Obl.BMP	1657556196.51	1657556196.72	206.22	0.206
TIEMPO PROMEDIO (SEG.)					0.232

**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DEL CAMPO DE
ORIENTACION EN IMÁGENES DE HUELLAS DACTILARES CON CORTE Z**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
001	I001_01_Zcut.BMP	1657555903.05	1657555906.09	3042.69	3.043
002	I001_02_Zcut.BMP	1657555906.11	1657555906.31	202.99	0.203
003	I001_03_Zcut.BMP	1657555906.32	1657555906.53	203.07	0.203
004	I001_04_Zcut.BMP	1657555906.56	1657555906.76	203.02	0.203
005	I001_05_Zcut.BMP	1657555906.78	1657555906.98	203.04	0.203

**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DEL CAMPO DE
ORIENTACION EN IMÁGENES DE HUELLAS DACTILARES CON CORTE Z**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
006	I001_06_Zcut.BMP	1657555907.03	1657555907.23	204.16	0.204
007	I001_07_Zcut.BMP	1657555907.26	1657555907.47	203.07	0.203
008	I001_08_Zcut.BMP	1657555907.48	1657555907.68	203.07	0.203
009	I001_09_Zcut.BMP	1657555907.72	1657555907.92	203.07	0.203
010	I001_10_Zcut.BMP	1657555907.93	1657555908.12	188.66	0.189
011	I002_01_Zcut.BMP	1657555908.15	1657555908.35	195.46	0.195
012	I002_02_Zcut.BMP	1657555908.38	1657555908.57	189.65	0.190
013	I002_03_Zcut.BMP	1657555908.59	1657555908.79	203.01	0.203
014	I002_04_Zcut.BMP	1657555908.84	1657555909.03	188.92	0.189
015	I002_05_Zcut.BMP	1657555909.30	1657555909.50	202.19	0.202
016	I002_06_Zcut.BMP	1657555909.52	1657555909.70	187.46	0.187
017	I002_07_Zcut.BMP	1657555909.73	1657555909.92	194.44	0.194
018	I002_08_Zcut.BMP	1657555909.96	1657555910.16	204.17	0.204
019	I002_09_Zcut.BMP	1657555910.19	1657555910.39	202.99	0.203
020	I002_10_Zcut.BMP	1657555910.41	1657555910.61	203.01	0.203
021	I003_01_Zcut.BMP	1657555910.61	1657555910.82	203.08	0.203
022	I003_02_Zcut.BMP	1657555910.86	1657555911.07	203.07	0.203
023	I003_03_Zcut.BMP	1657555911.10	1657555911.29	187.48	0.187
024	I003_04_Zcut.BMP	1657555911.33	1657555911.54	211.91	0.212
025	I003_05_Zcut.BMP	1657555911.58	1657555911.78	203.00	0.203
026	I003_06_Zcut.BMP	1657555911.78	1657555911.98	203.07	0.203
027	I003_07_Zcut.BMP	1657555912.01	1657555912.22	205.58	0.206
028	I003_08_Zcut.BMP	1657555912.35	1657555912.55	202.96	0.203
029	I003_09_Zcut.BMP	1657555912.60	1657555912.80	203.02	0.203
030	I003_10_Zcut.BMP	1657555912.82	1657555913.02	203.07	0.203
031	I004_01_Zcut.BMP	1657555913.04	1657555913.23	188.58	0.189
032	I004_02_Zcut.BMP	1657555913.24	1657555913.45	206.21	0.206
033	I004_03_Zcut.BMP	1657555913.48	1657555913.70	220.85	0.221
034	I004_04_Zcut.BMP	1657555913.75	1657555913.94	187.45	0.187
035	I004_05_Zcut.BMP	1657555913.98	1657555914.19	204.24	0.204
036	I004_06_Zcut.BMP	1657555914.22	1657555914.42	203.07	0.203
037	I004_07_Zcut.BMP	1657555914.44	1657555914.64	203.07	0.203
038	I004_08_Zcut.BMP	1657555914.69	1657555914.89	203.06	0.203
039	I004_09_Zcut.BMP	1657555914.90	1657555915.11	204.14	0.204
040	I004_10_Zcut.BMP	1657555915.14	1657555915.34	203.04	0.203
041	I005_01_Zcut.BMP	1657555915.36	1657555915.55	187.49	0.187
042	I005_02_Zcut.BMP	1657555915.56	1657555915.75	187.43	0.187

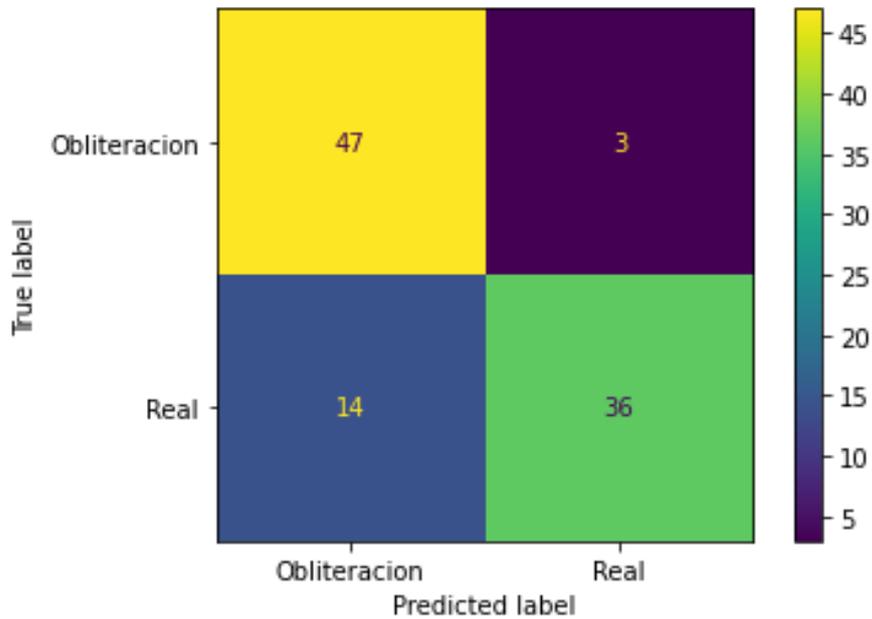
**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DEL CAMPO DE
ORIENTACION EN IMÁGENES DE HUELLAS DACTILARES CON CORTE Z**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
043	I005_03_Zcut.BMP	1657555915.80	1657555916.00	203.02	0.203
044	I005_04_Zcut.BMP	1657555916.02	1657555916.22	204.26	0.204
045	I005_05_Zcut.BMP	1657555916.23	1657555916.44	200.29	0.200
046	I005_06_Zcut.BMP	1657555916.47	1657555916.66	194.78	0.195
047	I005_07_Zcut.BMP	1657555916.69	1657555916.90	203.03	0.203
048	I005_08_Zcut.BMP	1657555916.91	1657555917.10	188.66	0.189
049	I005_09_Zcut.BMP	1657555917.13	1657555917.33	203.02	0.203
050	I005_10_Zcut.BMP	1657555917.37	1657555917.57	203.08	0.203
051	I006_01_Zcut.BMP	1657555917.60	1657555917.80	203.02	0.203
052	I006_02_Zcut.BMP	1657555917.83	1657555918.04	203.04	0.203
053	I006_03_Zcut.BMP	1657555918.05	1657555918.26	204.27	0.204
054	I006_04_Zcut.BMP	1657555918.26	1657555918.46	205.25	0.205
055	I006_05_Zcut.BMP	1657555918.49	1657555918.73	239.94	0.240
056	I006_06_Zcut.BMP	1657555918.75	1657555918.95	203.11	0.203
057	I006_07_Zcut.BMP	1657555918.95	1657555919.16	203.84	0.204
058	I006_08_Zcut.BMP	1657555919.17	1657555919.39	220.20	0.220
059	I006_09_Zcut.BMP	1657555919.42	1657555919.63	203.10	0.203
060	I006_10_Zcut.BMP	1657555919.72	1657555919.92	203.06	0.203
061	I007_01_Zcut.BMP	1657555919.95	1657555920.16	202.76	0.203
062	I007_02_Zcut.BMP	1657555920.17	1657555920.39	219.10	0.219
063	I007_03_Zcut.BMP	1657555920.42	1657555920.63	203.08	0.203
064	I007_04_Zcut.BMP	1657555920.66	1657555920.84	187.40	0.187
065	I007_05_Zcut.BMP	1657555920.88	1657555921.08	203.03	0.203
066	I007_06_Zcut.BMP	1657555921.10	1657555921.28	187.49	0.187
067	I007_07_Zcut.BMP	1657555921.33	1657555921.53	198.73	0.199
068	I007_08_Zcut.BMP	1657555921.54	1657555921.75	202.99	0.203
069	I007_09_Zcut.BMP	1657555921.76	1657555921.95	187.36	0.187
070	I007_10_Zcut.BMP	1657555921.98	1657555922.17	188.65	0.189
071	I008_01_Zcut.BMP	1657555922.19	1657555922.39	203.01	0.203
072	I008_02_Zcut.BMP	1657555922.42	1657555922.62	202.91	0.203
073	I008_03_Zcut.BMP	1657555922.62	1657555922.83	203.11	0.203
074	I008_04_Zcut.BMP	1657555922.84	1657555923.05	203.01	0.203
075	I008_05_Zcut.BMP	1657555923.06	1657555923.25	188.58	0.189
076	I008_06_Zcut.BMP	1657555923.30	1657555923.50	203.02	0.203
077	I008_07_Zcut.BMP	1657555923.55	1657555923.74	187.40	0.187
078	I008_08_Zcut.BMP	1657555923.79	1657555923.99	203.01	0.203
079	I008_09_Zcut.BMP	1657555924.00	1657555924.21	204.15	0.204

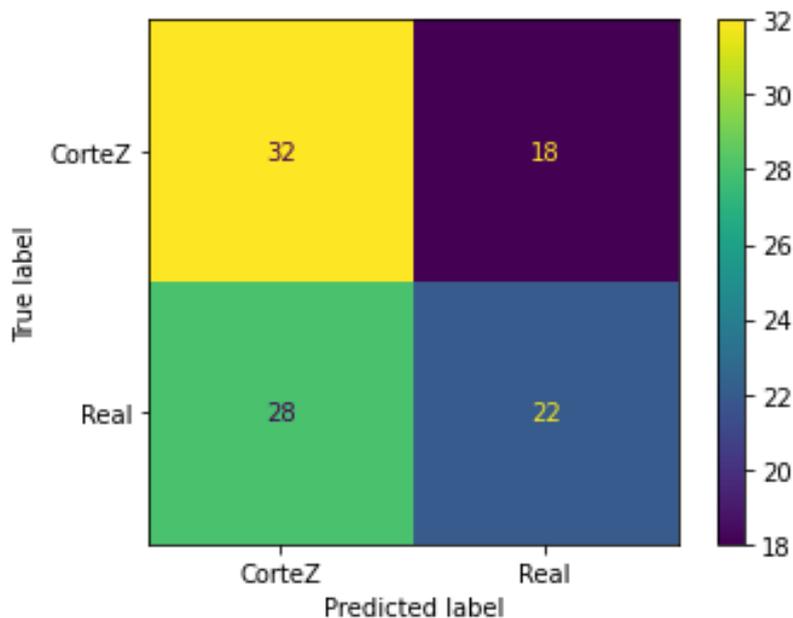
**REGISTRO DE TIEMPO DE PROCESAMIENTO DEL ALGORITMO DEL CAMPO DE
ORIENTACION EN IMÁGENES DE HUELLAS DACTILARES CON CORTE Z**

N°	Imagen	Tiempo Inicio (milisegundos)	Tiempo Fin (milisegundos)	Milisegundos	Segundos
080	I008_10_Zcut.BMP	1657555924.22	1657555924.43	203.02	0.203
081	I009_01_Zcut.BMP	1657555924.47	1657555924.68	203.03	0.203
082	I009_02_Zcut.BMP	1657555924.68	1657555924.88	203.07	0.203
083	I009_03_Zcut.BMP	1657555924.93	1657555925.13	204.15	0.204
084	I009_04_Zcut.BMP	1657555925.15	1657555925.33	187.37	0.187
085	I009_05_Zcut.BMP	1657555925.37	1657555925.57	203.08	0.203
086	I009_06_Zcut.BMP	1657555925.58	1657555925.79	203.66	0.204
087	I009_07_Zcut.BMP	1657555925.80	1657555925.99	187.37	0.187
088	I009_08_Zcut.BMP	1657555926.02	1657555926.23	204.25	0.204
089	I009_09_Zcut.BMP	1657555926.24	1657555926.44	193.47	0.193
090	I009_10_Zcut.BMP	1657555926.45	1657555926.65	203.11	0.203
091	I010_01_Zcut.BMP	1657555926.69	1657555926.89	203.06	0.203
092	I010_02_Zcut.BMP	1657555926.90	1657555927.11	204.23	0.204
093	I010_03_Zcut.BMP	1657555927.11	1657555927.31	203.08	0.203
094	I010_04_Zcut.BMP	1657555927.34	1657555927.55	203.07	0.203
095	I010_05_Zcut.BMP	1657555927.56	1657555927.75	187.46	0.187
096	I010_06_Zcut.BMP	1657555927.81	1657555928.05	234.29	0.234
097	I010_07_Zcut.BMP	1657555928.05	1657555928.23	188.59	0.189
098	I010_08_Zcut.BMP	1657555928.27	1657555928.47	206.19	0.206
099	I010_09_Zcut.BMP	1657555928.54	1657555928.74	203.11	0.203
100	I010_10_Zcut.BMP	1657555928.77	1657555928.97	202.99	0.203
TIEMPO PROMEDIO (SEG.)					0.230

Matrix de confusión del algoritmo Extracción de Minucias, cuando se compara una huella dactilar Real con una Alterada Obliterada.

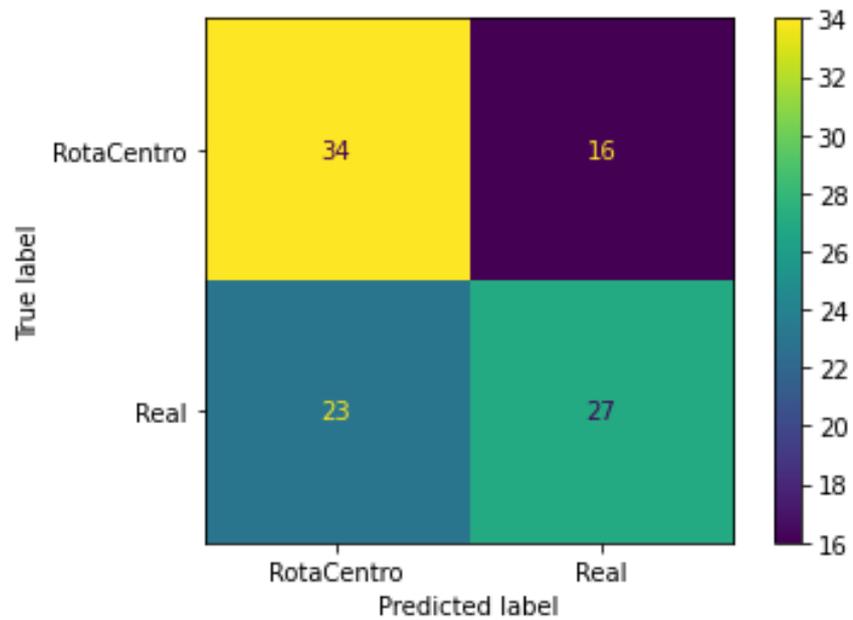


Matrix de confusión del algoritmo Extracción de Minucias, cuando se compara una huella dactilar Real con una Alterada en Corte Z.

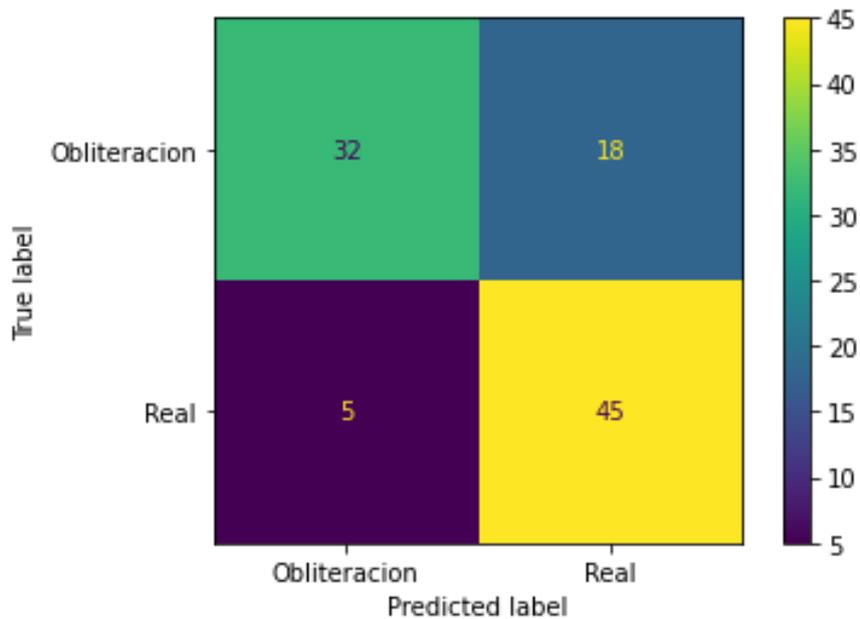


Matrix de confusión del algoritmo Extracción de Minucias, cuando se

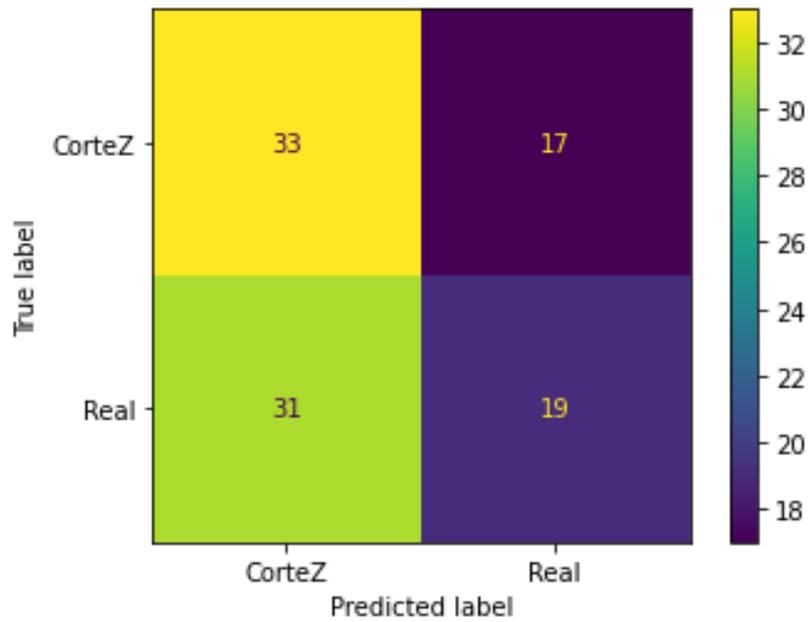
compara una huella dactilar Real con una Alterada en Rotación Central.



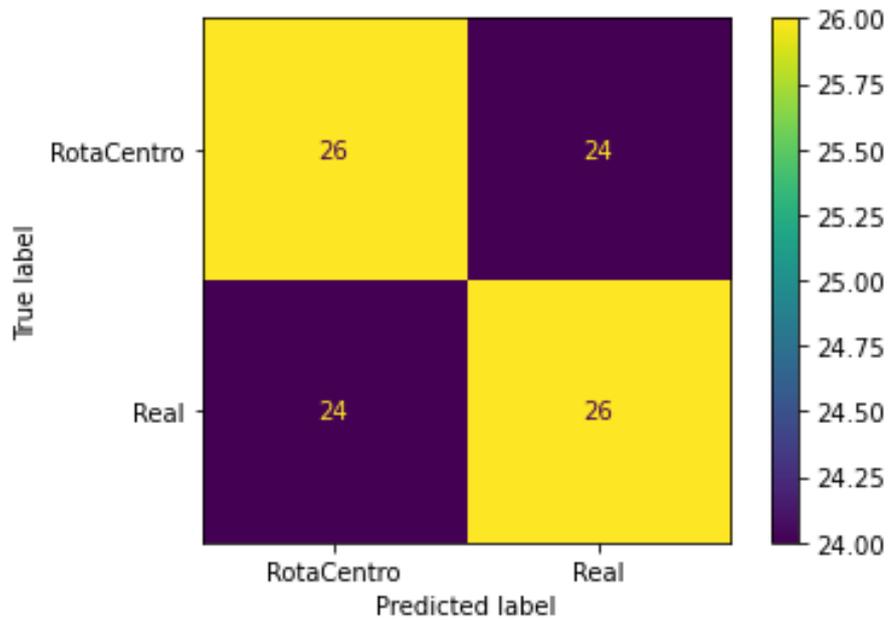
Matrix de confusión del algoritmo Campo de Orientación, cuando se compara una huella dactilar Real con una Alterada Obliterada.



Matrix de confusión del algoritmo Campo de Orientación, cuando se compara una huella dactilar Real con una Alterada en Corte Z.



Matrix de confusión del algoritmo Campo de Orientación, cuando se compara una huella dactilar Real con una Alterada en Rotación Central.



Anexo 4.
Consentimiento informado.

DECLARACIÓN DE CONSENTIMIENTO INFORMADO

Fecha: 21-12-2021

Yo, Dante Omar Davila Salazar, identificado(a) con DNI N° 47226711, con residencia en el Distrito CHONGUYALE, Provincia CHICLAYO, Departamento LAMBAYEQUE, por medio del presente documento doy consentimiento al Alumno en ingeniería de sistemas de la Universidad Señor de Sipán, Natividad Alejandro Díaz Carrasco, al uso de las huellas dactilares de los dedos de mis manos (10 dedos), exclusivamente para fines académicos, en la investigación titulada: "EVALUACIÓN DE ALGORITMOS PARA LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS".

Se me ha informado acerca de las características del estudio y he recibido garantía de que, los datos personales que se revelen al encargado de realizar la investigación, serán estrictamente confidenciales, además de que los procedimientos a evaluar no son perjudiciales a la integridad de mi persona.



Firma

Nombre: Dante Omar Davila Salazar

DNI : 47226711

DECLARACIÓN DE CONSENTIMIENTO INFORMADO

Fecha: 21-12-2021

Yo, Francisca Díaz Carrasco, identificado(a) con DNI N° 16511659, con residencia en el Distrito Chongoyape, Provincia Chiclayo, Departamento Lambayeque, por medio del presente documento doy consentimiento al Alumno en ingeniería de sistemas de la Universidad Señor de Sipán, Natividad Alejandro Díaz Carrasco, al uso de las huellas dactilares de los dedos de mis manos (10 dedos), exclusivamente para fines académicos, en la investigación titulada: "EVALUACIÓN DE ALGORITMOS PARA LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS".

Se me ha informado acerca de las características del estudio y he recibido garantía de que, los datos personales que se revelen al encargado de realizar la investigación, serán estrictamente confidenciales, además de que los procedimientos a evaluar no son perjudiciales a la integridad de mi persona.


Firma

Nombre: Francisca Díaz Carrasco

DNI : 16511659

DECLARACIÓN DE CONSENTIMIENTO INFORMADO

Fecha: 21-12-2021

Yo, Brando Jefferson Vera Díaz, identificado(a) con DNI N° 73574675, con residencia en el Distrito Lhanguyape, Provincia Chiclayo, Departamento Lambayeque, por medio del presente documento doy consentimiento al Alumno en ingeniería de sistemas de la Universidad Señor de Sipán, Natividad Alejandro Díaz Carrasco, al uso de las huellas dactilares de los dedos de mis manos (10 dedos), exclusivamente para fines académicos, en la investigación titulada: "EVALUACIÓN DE ALGORITMOS PARA LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS".

Se me ha informado acerca de las características del estudio y he recibido garantía de que, los datos personales que se revelen al encargado de realizar la investigación, serán estrictamente confidenciales, además de que los procedimientos a evaluar no son perjudiciales a la integridad de mi persona.



Firma

Nombre: Brando Jefferson Vera Díaz
DNI : 73574675

DECLARACIÓN DE CONSENTIMIENTO INFORMADO

Fecha: 21-12-2021

Yo, VICTOR ALFONSO QUINTANA DIAZ, identificado(a) con DNI N° 45604091, con residencia en el Distrito CHONGUYOPE, Provincia CHICLAYO, Departamento LOMBAYEQUE, por medio del presente documento doy consentimiento al Alumno en ingeniería de sistemas de la Universidad Señor de Sipán, Natividad Alejandro Díaz Carrasco, al uso de las huellas dactilares de los dedos de mis manos (10 dedos), exclusivamente para fines académicos, en la investigación titulada: "EVALUACIÓN DE ALGORITMOS PARA LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS".

Se me ha informado acerca de las características del estudio y he recibido garantía de que, los datos personales que se revelen al encargado de realizar la investigación, serán estrictamente confidenciales, además de que los procedimientos a evaluar no son perjudiciales a la integridad de mi persona.



Firma

Nombre: VICTOR ALFONSO QUINTANA DIAZ

DNI : 45604091

DECLARACIÓN DE CONSENTIMIENTO INFORMADO

Fecha: 21-12-2021

Yo, Florita del Milagro Antón Díaz, identificado(a) con DNI N° 70510542, con residencia en el Distrito Chongoyape, Provincia Chiclayo, Departamento Lambayeque, por medio del presente documento doy consentimiento al Alumno en ingeniería de sistemas de la Universidad Señor de Sipán, Natividad Alejandro Díaz Carrasco, al uso de las huellas dactilares de los dedos de mis manos (10 dedos), exclusivamente para fines académicos, en la investigación titulada: "EVALUACIÓN DE ALGORITMOS PARA LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS".

Se me ha informado acerca de las características del estudio y he recibido garantía de que, los datos personales que se revelen al encargado de realizar la investigación, serán estrictamente confidenciales, además de que los procedimientos a evaluar no son perjudiciales a la integridad de mi persona.



Firma

Nombre: Florita del Milagro Antón Díaz

DNI : 70510542.

DECLARACIÓN DE CONSENTIMIENTO INFORMADO

Fecha: 21-12-2021

Yo, Rosa Elvira Díaz Carrasco, identificado(a) con DNI N° 16711216, con residencia en el Distrito Chongoyape, Provincia Chiclayo, Departamento Lambayeque, por medio del presente documento doy consentimiento al Alumno en ingeniería de sistemas de la Universidad Señor de Sipán, Natividad Alejandro Díaz Carrasco, al uso de las huellas dactilares de los dedos de mis manos (10 dedos), exclusivamente para fines académicos, en la investigación titulada: "EVALUACIÓN DE ALGORITMOS PARA LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS".

Se me ha informado acerca de las características del estudio y he recibido garantía de que, los datos personales que se revelen al encargado de realizar la investigación, serán estrictamente confidenciales, además de que los procedimientos a evaluar no son perjudiciales a la integridad de mi persona.


Firma

Nombre: Rosa Elvira Díaz Carrasco

DNI : 16711216

DECLARACIÓN DE CONSENTIMIENTO INFORMADO

Fecha: 21-12-2021

Yo, Amalia Díaz Carrasco, identificado(a) con DNI N° 16619333, con residencia en el Distrito Chongoyape, Provincia Chiclayo, Departamento Lambayeque, por medio del presente documento doy consentimiento al Alumno en ingeniería de sistemas de la Universidad Señor de Sipán, Natividad Alejandro Díaz Carrasco, al uso de las huellas dactilares de los dedos de mis manos (10 dedos), exclusivamente para fines académicos, en la investigación titulada: "EVALUACIÓN DE ALGORITMOS PARA LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS".

Se me ha informado acerca de las características del estudio y he recibido garantía de que, los datos personales que se revelen al encargado de realizar la investigación, serán estrictamente confidenciales, además de que los procedimientos a evaluar no son perjudiciales a la integridad de mi persona.


Firma

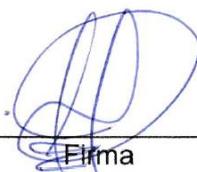
Nombre: Amalia Díaz Carrasco
DNI : 16619333

DECLARACIÓN DE CONSENTIMIENTO INFORMADO

Fecha: 21-12-2021

Yo, María Elena Díaz Carrasco, identificado(a) con DNI N° 16663475, con residencia en el Distrito Chongoyape, Provincia Chilayo, Departamento Kambayegu, por medio del presente documento doy consentimiento al Alumno en ingeniería de sistemas de la Universidad Señor de Sipán, Natividad Alejandro Díaz Carrasco, al uso de las huellas dactilares de los dedos de mis manos (10 dedos), exclusivamente para fines académicos, en la investigación titulada: "EVALUACIÓN DE ALGORITMOS PARA LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS".

Se me ha informado acerca de las características del estudio y he recibido garantía de que, los datos personales que se revelen al encargado de realizar la investigación, serán estrictamente confidenciales, además de que los procedimientos a evaluar no son perjudiciales a la integridad de mi persona.



Firma

Nombre: María Elena Díaz Carrasco
DNI : 16663475

DECLARACIÓN DE CONSENTIMIENTO INFORMADO

Fecha: 21-12-2021

Yo, Natividad Alejandro Díaz Carrasco, identificado(a) con DNI N° 40807498, con residencia en el Distrito Chongoyape, Provincia Chiclayo, Departamento Lambayeque, por medio del presente documento doy consentimiento al Alumno en ingeniería de sistemas de la Universidad Señor de Sipán, Natividad Alejandro Díaz Carrasco, al uso de las huellas dactilares de los dedos de mis manos (10 dedos), exclusivamente para fines académicos, en la investigación titulada: "EVALUACIÓN DE ALGORITMOS PARA LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS".

Se me ha informado acerca de las características del estudio y he recibido garantía de que, los datos personales que se revelen al encargado de realizar la investigación, serán estrictamente confidenciales, además de que los procedimientos a evaluar no son perjudiciales a la integridad de mi persona.



Firma

Nombre: Natividad Alejandro Díaz Carrasco

DNI : 40807498

DECLARACIÓN DE CONSENTIMIENTO INFORMADO

Fecha: 21-12-2021

Yo, Rocio del Pilar Mendo Maldonado identificado(a) con DNI N° 40346150, con residencia en el Distrito Chongoyape, Provincia Chiclayo, Departamento Lambayegve, por medio del presente documento doy consentimiento al Alumno en ingeniería de sistemas de la Universidad Señor de Sipán, Natividad Alejandro Díaz Carrasco, al uso de las huellas dactilares de los dedos de mis manos (10 dedos), exclusivamente para fines académicos, en la investigación titulada: "EVALUACIÓN DE ALGORITMOS PARA LA DETECCIÓN DE HUELLAS DACTILARES ALTERADAS".

Se me ha informado acerca de las características del estudio y he recibido garantía de que, los datos personales que se revelen al encargado de realizar la investigación, serán estrictamente confidenciales, además de que los procedimientos a evaluar no son perjudiciales a la integridad de mi persona.


Firma

Nombre: Rocio del Pilar Mendo Maldonado

DNI : 40346150