UNIVERSIDAD NACIONAL DE MOQUEGUA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

ALGORITMO HASH DE PERCEPCIÓN PARA DETECTAR IMÁGENES MIXTIFICADOS EN ENTORNO A LA PERICIA DE LA IMAGEN DIGITAL, PROVINCIA DE SURQUILLO, REGIÓN LIMA, AÑO 2020

TESIS

PRESENTADO POR:

REYNALDO HANCCO MOROCCO

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO DE SISTEMAS E INFORMÁTICA

MOQUEGUA – PERÚ

2023

Folio Nº 011

ACTA DE SUSTENTACIÓN

I. Modalidad:

Tesis : [X]
Trabajo de Suficiencia Profesional : [ ]
Otro: ..... : [ ]

II. Documento de aprobación para la sustentación:

Resolución de facultad Nº 217-2023-FACIA-UNAM

Fecha: 18 de diciembre de 2023

III. El jurado ha dictaminado la (el) Tesis, TITULADO:

"Algoritmo hash de percepción para detectar imágenes mixtificados en entorno a la pericia de la imagen digital, provincia de Surquillo, región Lima, año 2020"

Bachiller: Reynaldo Hancco Morocco

Asesor Principal: Alex Peter Zuñiga Incalla

Co-Asesor: Juan Carlos Clares Perca

IV. Se dictamina:

Como: Aprobado por Mayoría la sustentación de manera:

Presencial [X] Virtual [ ]

Con nota: Doce (12) por lo que; Sí [X] No [ ]

Opta el Título Profesional de INGENIERO DE SISTEMAS E INFORMÁTICA.

Siendo las 16:50 horas, del 21 de diciembre del año 2023 se procede con la firma en conformidad de lo dictaminado por el Jurado.

Presidente

Maribel Estela Coaquira Mamani
Primer Miembro

Salomón Rey Ramos Rivera
Segundo Miembro

Reynaldo Hancco Morocco
Bachiller

UNIVERSIDAD NACIONAL DE MOQUEGUA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

TÍTULO PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

ALGORITMO HASH DE PERCEPCIÓN PARA DETECTAR IMÁGENES MIXTIFICADOS EN ENTORNO A LA PERICIA DE LA IMAGEN DIGITAL, PROVINCIA DE SURQUILLO, REGIÓN LIMA, AÑO 2020

Tesis sustentada y aprobada el 21 de diciembre del 2023: estando el jurado calificador, asesor.

PRESIDENTE : _____
MSC. MARIO GAUNA CHINO.

1ER MIEMBRO : _____
Mg. MARIBEL ESTELA COAGUILA.

2DO MIEMBRO : _____
DR. SALOMÓN REY RAMOS RIVERA.

ASESOR : _____
M.Sc. ALEX PETER ZÚÑIGA INCALLA.

DEDICATORIA

Dedico este trabajo de investigación a mis padres y mis hermanos por motivarme para dar un paso en mi desarrollo profesional.

A mis padres, porque han sabido formarme con buenos sentimientos, hábitos y valores.

De igual manera, a mis amigos que me brindaron sus consejos y sus experiencias.

Reynaldo Hancco Morocco
Ilo, diciembre 2023

AGRADECIMIENTOS

Agradezco a Dios por protegerme y darme fuerzas para superar obstáculos.

A mi madre, que me ha enseñado a no rendirme y perseverar a través de sus sabios consejos.

A mi padre, por su apoyo incondicional por demostrarme la gran fe que tiene en mí.

Al M.Sc. Alex Peter Zúñiga Incalla, asesor de mi tesis, por su valiosa guía y asesoramiento a la realización de la misma.

Reynaldo Hancco Morocco
Ilo, diciembre 2023

Contenido

RESUMEN **10**

ABSTRACT **11**

INTRODUCCIÓN **12**

I. PLANTEAMIENTO DEL PROBLEMA **13**

1.1. Descripción de la realidad problemática 13

1.1.1. Problema general 16

1.1.2. Problemas específicos 16

1.2. Objetivos 16

1.2.1. Objetivo general 16

1.2.2. Objetivos específicos 16

1.3. Justificación e Importancia 17

1.4. Hipótesis 18

1.4.1. Hipótesis general 18

1.4.2. Hipótesis específicas 18

1.5. Variables 19

1.5.1. Operacionalización de variables 19

1.5.1.1. Variable Independiente 19

1.5.1.2. Variable Dependiente 20

II. MARCO TEÓRICO **21**

2.1. Antecedentes 21

2.2. Bases teóricas 23

2.2.1. Imagen digital 23

2.2.2. Representación de la imagen digital 25

2.2.3. Tipos de imágenes digitales 26

2.2.3.1. Imagen vectorial 26

2.2.3.2. Imagen de mapa de bits 27

2.2.4. Procesamiento de imágenes 27

2.2.5. Función hash.....30
2.2.6. Hash criptográfico 31
2.2.7. Hash basado en Contenido.....33
2.2.8. Algoritmo hash de Percepción.....36
2.2.8.1 Algoritmo de Hash promedio (a_hash) 38
2.2.8.2 Algoritmo de Hash diferencial (d_hash).....39
2.2.8.3 Algoritmo de Hash DCT (p_hash) 40
2.2.8.4 Algoritmo Hash basado en VGG – 16 (propuesta).....40
2.2.9. Normalización de imágenes.....41
2.2.9.1. VGG – 16. 43
2.2.10. Transfer learning 43
2.2.10.1. Distancia de Hamming. 45
2.2.10.2. Sensibilidad (Recall). 46
2.2.10.3. Precisión 47
2.2.10.4. F1-score 47
2.2.11. Peritaje informático.....47
2.2.11.1. Peritaje forense informático 47
2.2.11.2. Peritaje tecnológico de gestión,.....48
2.2.11.3. Peritaje tecnológico de mediación.....48
2.2.11.4. Tasación tecnológica.....48
2.3. Definición de términos.....50
III. METODOLOGÍA.....52
3.1. Tipo y nivel de investigación 52
3.1.1. Tipo de Investigación 52
3.1.2. Diseño de Investigación 52
3.1.3. Alcance de la Investigación 52
3.2. Ámbito temporal y espacial 53
3.2.1. Ámbito temporal 53

3.2.2. Ámbito espacial 53
3.3. Población y muestra 53
3.3.1. Población 53
3.3.2. Muestra 54
3.4. Técnicas e Instrumentos para la Recolección de Datos 54
3.4.1. Selección de Instrumentos 55
3.4.2. Validación y Confiabilidad del Instrumento 55
3.5. Procedimientos 55
3.5.1. Análisis de datos 56
3.5.2. Adquisición de datos 56
3.5.3. Procesamiento de imágenes 56
3.5.3.1. Desenfoque Gaussiano 57
3.5.3.2. Compresión JPEG 58
3.5.3.3. Rotación de imágenes 58
3.5.4. Implementación de los algoritmos de hash de percepción. 59
3.5.5. Descripción del Datasets 61
3.5.6. Extracción de características y generación de Hash 62
3.5.7. Evaluación de algoritmos hash (Análisis Comparativo) 63
3.5.7.1. Sensibilidad (Recall). 64
3.5.7.2. Umbral. 65
IV. RESULTADOS 66
V. DISCUSIÓN 74
VI. CONCLUSIONES 76
VII. RECOMENDACIONES 77
VIII. ANEXOS 87

Contenido de Ilustraciones

Ilustración 1: Imagen digital..... 24
Ilustración 2: Cuantización de la imagen digital..... 25
Ilustración 3: Imagen representada como función f(x,y)..... 26
Ilustración 4: Imagen vectorial o trazos de vectores..... 26
Ilustración 5: Imagen de mapa de bits o imagen raster..... 27
Ilustración 6: Etapas del procesamiento de la imagen digital..... 28
Ilustración 7: Función Hash..... 30
Ilustración 8: Detección de documentos adulterados comparado con los hashes 31
Ilustración 9: Función hash criptográfico, un solo bit cambia, el valor hash es totalmente diferente..... 33
Ilustración 10: Ejemplos de copias de imagen "Lena" con CPO..... 34
Ilustración 11: Marco de los algoritmos hash perceptuales..... 37
Ilustración 12: Entradas y salidas del algoritmo de normalización de imágenes 41
Ilustración 13: Arquitectura del modelo VGG-16..... 43
Ilustración 14: Proceso de aprendizaje Machine Learning y Transfer Learning 44
Ilustración 15: Distancia de Hamming de 4 dimensiones..... 46
Ilustración 16: Captura de la imagen por una cámara digital..... 56
Ilustración 17: Código del desenfoque de una imagen..... 57
Ilustración 18: Imagen original e imagen con desenfoque..... 57
Ilustración 19: Código de la compresión JPEG de una imagen..... 58
Ilustración 20: Imagen original e imagen con compresión 10..... 58
Ilustración 21: Código del ángulo de rotación de una imagen..... 59
Ilustración 22: Imagen original e imagen con giro 5°..... 59
Ilustración 23: Código de hash promedio..... 60
Ilustración 24: Código hash diferencial (hash)..... 60
Ilustración 25: Código hash DCT (p_hash)..... 60
Ilustración 26: Organización de carpetas con imágenes mixtificados y original 61
Ilustración 27: Generación de hash de imagen Original con algoritmo Hash DCT 61
Ilustración 28: Generación de hash de imagen mixtificada con algoritmo hash 62
Ilustración 29: Calculando Distancia de Hamming entre el hash de la imagen original y el hash de la imagen mixtificada..... 62
Ilustración 30: carpetas organizadas para los experimentos. 62

Ilustración 31: función para extraer características de imágenes y generación de hash. 63
Ilustración 32: generación de hashes con hash promedio 63
Ilustración 33: cálculo de distancia de Hamming entre el hash_original y hash_compresion_20..... 624
Ilustración 34: muestra los valores obtenidos para imágenes con desenfoque de evaluado con hash promedio..... 625
Ilustración 35: comparación de hashes producidos por los diferentes algoritmos de hash de percepción. 68
Ilustración 36: Distancia media de Hamming por tipos de mixtificación..... 70
Ilustración 37: Código para la extracción de características y generación de cadena de Hash..... 72
Ilustración 38: Tasa detección de imágenes mixtificadas por los diferentes algoritmos hash de percepción incluida la propuesta de hash basado en Vgg16.. 73

Contenido de Tablas

Tabla 1: Operacionalización de la variable independiente 19
Tabla 2: Operacionalización de la variable dependiente 20
Tabla 3: Configuración de parámetros para la mixtificación de imágenes..... 65
Tabla 4: Promedio de taza detección de imágenes mixtificadas por los diferentes algoritmos hash de percepción 66
Tabla 5: Promedio de taza detección por tipo de mixtificació..... 69
Tabla 6: Tabla de distancias mínimas de Hamming por cada tipo de mixtificación 70
Tabla 7: Tabla de distancias máximas de Hamming por cada tipo de mixtificación. 70
Tabla 8: Tabla de distancias media de Hamming por cada tipo de mixtificación 70
Tabla 9: Tasa detección de imágenes mixtificadas por los diferentes algoritmos hash de percepción incluida la propuesta de hash basado en Vgg16..... 71

RESUMEN

El aumento de la adulteración de las imágenes digitales por diferentes programas informáticos como Photoshop es un problema que afecta en estos tiempos a la sociedad, no obstante, existe una carencia de mecanismos para detectar estas alteraciones en las imágenes digitales. La presente investigación tiene como objetivo determinar el aporte del algoritmo hash de percepción en la detección de imágenes mixtificados en entorno a la pericia de la imagen digital. En la primera etapa se procedió a la obtención de 300 imágenes adquiridas de la Policía Nacional del Perú (PNP). En la segunda etapa se realizó el pre-procesamiento de imágenes con consiste en transformaciones geométricas, desenfoque de imágenes y compresión de imágenes, seguidamente extraemos las características de la imágenes con los diferentes algoritmos de hash de percepción implementados, luego comparamos la taza de detección de los diferentes algoritmos de hash de percepción, en esta investigación se propone el modelo VGG – 16 para extraer las características de la imagen para luego convertir en una cadena de hash con una función convertimos el vector de características de números flotantes en un hash binario; los números en coma flotante se convierten asignando 1 si el valor es mayor a la medida del vector y asignando 0 si el valor es menor o igual a la media. En la tercera etapa se utilizó las métricas para ver el aporte de los algoritmos hash percepción en la detección de imágenes mixtificados

ABSTRACT

The increase in adulteration of digital images by different computer programs such as Photoshop is a problem that affects society these days; however, there is a lack of mechanisms to detect these alterations in digital images. The objective of this research is to determine the contribution of the perception hash algorithm in the detection of mixed images in the context of digital image expertise. In the first stage, 300 images acquired from the National Police of Peru (PNP) were obtained. In the second stage, image pre-processing was carried out consisting of geometric transformations, image blurring and image compression, then we extracted the characteristics of the images with the different perception hash algorithms implemented, then we compared the detection rate of the different perception hashing algorithms, in this research the VGG – 16 model is proposed to extract the features of the image and then convert them into a hash chain with a function we convert the feature vector of floating numbers into a binary hash; floating point numbers are converted by assigning 1 if the value is greater than the vector measure and assigning 0 if the value is less than or equal to the mean. In the third stage, the metrics were used to see the contribution of the perception hash algorithms in the detection of mixed images.

INTRODUCCIÓN

El aumento de la adulteración de las imágenes digitales por diferentes programas informáticos como Photoshop es un problema que afecta en estos tiempos a la sociedad digital, no obstante, existe una carencia de mecanismos para detectar estas alteraciones en las imágenes digitales. Frente a este problema, los algoritmos de Hash de percepción se convierten en un aliado para detectar estas alteraciones. Esta investigación pretende mostrar las capacidades de los algoritmos de hash de percepción que aporten y permitan detectar imágenes mixtificados a favor de la pericia de la imagen digital mediante el algoritmo hash de percepción.

La investigación está dada bajo un enfoque cuantitativa de tipo aplicativa y experimental, para ello, se desarrolló la metodología hipotético-deductivo por basarse en la generación de hipótesis a partir de dos premisas, universal (leyes y teorías científicas), cuyas fases están orientadas a este tipo de estudio, y dentro de ellas, para la creación del algoritmo de hash de percepción. Se usa tres algoritmos de hash de percepción DCT, hash diferencial, hash basado en transformada discreta de coseno, hash promedio; para extraer características de la imagen original y su respectiva imagen mixtificada. Una vez obtenida las características de estas imágenes se convierten en vectores binarios, que serán nuestro hash de percepción, obtenemos el hash de percepción de las imágenes originales y de su correspondiente imagen mixtificada para luego sacar la Distancia de Hamming, seguidamente se examina utilizando métricas de evaluación como la sensibilidad la discriminabilidad, robustez de los hashes, interpretamos los resultados obtenidos y analizamos la capacidad del modelo.

El estudio se divide en capítulos y se encuentra organizado de la siguiente forma, Capítulo I Planteamiento del problema que corresponde a la descripción de la realidad problemática, objetivos, justificación, importancia, hipótesis y variables de la investigación; Capítulo II Marco teórico que corresponde a los antecedentes del estudio, bases teóricas de la investigación y la definición de términos; Capítulo III Metodología de la investigación que corresponde al tipo y nivel de la investigación, población y muestra, técnicas e instrumentos para la recolección de datos, análisis y procesamiento de los datos; Resultados, Discusión de los resultados, Conclusiones, Recomendaciones, Referencias bibliográficas y Anexos que respaldan la investigación.

I. PLANTEAMIENTO DEL PROBLEMA

1.1. Descripción de la realidad problemática

Actualmente vivimos en una sociedad en constante crecimiento, a medida que las áreas urbanas se expanden, también son acompañadas con múltiples factores negativos que están en contra de la ética y moral, uno de los temores que surge en este contexto tiene que ver con el aumento de la adulteración de imágenes digitales, esta adulteración es un fenómeno que ha ido aumentando constantemente intrínsecamente con la tecnología. La creciente Digitalización juega un papel crítico entorno de investigaciones criminales, legales y de diversas áreas que involucran imágenes, video y documentos digitales, es ante esta problemática que surgen nuevas tendencias para detectar este tipo de información.

Según (Nikoukhah et al., 2019) Menciona que hay un impacto de la manipulación de imágenes que repercute en la difusión de noticias falsas y este es cada vez más relevante, incluso algunas redes sociales permiten editar las imágenes y video. La manipulación de imágenes en JPEG también se da por la comprensión, esto permite que los archivos pierdan información y acompañado de herramientas de edición permiten realizar alteraciones en imágenes, sin embargo, también existen herramientas que pueden identificar estas alteraciones, pero tiene una capacidad limitada. En relación a la problemática para (Fernández et al., 2018) la manipulación de estas herramientas no requiere técnicas avanzadas, debido que el mismo avance tecnológico permite realizar todos los cambios, Esto presenta un desafío considerable para aquellos que se dedican a combatir este tipo de información.

El estudio (Bañuelos Capistrán, 2020) realizado en México, aborda las complejidades tecnológicas, culturales, legales y éticas relacionadas con el deepfake y menciona su alta preocupación por su rápido crecimiento y que quienes abordan este problema deben dar credibilidad frente a esta información manipulable, debido a su proliferación en una variedad de aplicaciones y son utilizados como entretenimiento, acoso, chantaje, propaganda y desinformación; incluyendo actrices reconocidas, quienes han sido víctimas de la creación de contenido pornográfico falso, lo cual genera cuestiones éticas y legales adicionales en torno a la privacidad y la difamación.

Según el estudio (Iakovidou et al., 2018) en Grecia, bajo el ámbito de imagen forense, hace mención que existen 2 categorías de falsificaciones; la utilización de firmas digitales y la manipulación en la imagen, ambos casos conlleva a la problemática generalizada en cuanto a la confiabilidad de las imágenes, afectando la percepción del público y en la toma de decisiones. La investigación muestra el desarrollado técnicas de detección de falsificaciones de imágenes, siendo una de las principales preocupaciones cómo identificar y verificar la autenticidad de las imágenes en situaciones en las que no se dispone de información previa o de confianza sobre su origen.

La investigación (Bruno & Luis, 2019) realizado en el contexto Peruano, expone cómo en el ámbito político se han registrado casos de manipulación fotográfica aplicados a diversos actores políticos con el fin de crear narrativas falsas de la realidad, que luego han sido refutadas. Estas prácticas tienen lugar en un contexto político donde los actores buscan desacreditar a sus oponentes a través de estrategias como el "terruqueo", la difamación de opositores y la alteración de las emociones de los protagonistas. En muchos de estos casos emplean programas como Photoshop para manipular imágenes y denigrar a opositores políticos vinculándolos con el violento pasado del país. El estudio resalta la necesidad de investigaciones futuras en el campo de la fotografía política y la manipulación de la comunicación, con la finalidad de salvaguardar la democracia y la libertad de pensamiento. Bajo este ámbito, la investigación (Martinez Garcia, 2017) observa que la manipulación de fotografías tiene un impacto significativo en la credibilidad de los medios de comunicación, la responsabilidad ética, la objetividad y el mensaje en el periodismo, bajo la percepción de los estudiantes de fotografía, teniendo una prueba estadística la cual respalda el nivel de significancia.

En el Perú, las tecnologías de la información son, sin lugar a dudas, una herramienta fundamental en el desarrollo de la sociedad actual, pero también es un arma potente para causar daños y en este sentido, podemos decir que, los delictivos informáticos es la parte negativa de este contexto actual, como la falsificación, alteración o manipulación de una imagen digital y otras anomalías que afectan la similitud de la originalidad y autenticidad, según la División de Estadísticas de la Dirección de Tecnología de la Información y Comunicaciones – Anuario Estadístico 2018 de la Policía Nacional del Perú existe 1816 actos de delito

informático, durante el periodo 2019 se registraron 2472 delitos informáticos; en el periodo 2020 se registraron 3973 delitos informáticos; correspondiendo al periodo 2021 fueron 7697 casos, sin embargo, según (*Boletín Estadístico Policial I Trimestre*, 2022) siendo su último reporte trimestral obtiene una cifra elevada correspondiente a 5090 casos delitos informativos, estos casos corresponden a falsificación de imágenes, falsificación de firmas, suplantación de identidad en redes sociales.

En el departamento de Lima, según reporte en los registros anuales de la Policía Nacional del Perú correspondientes a los años 2018, 2019, 2020 y 2021, se documentaron 868, 1161, 2097 y 3857, casos respectivamente de delitos informáticos según (*Anuario Estadístico Policial*, 2021) justificando el grado de preocupación en que las personas puede ser engañadas con frecuencia en su vida cotidiana, siendo insuficiente los esfuerzos de la autoridad fiscalizadora y de orden de identificar y detectar el acto ilícito

Frente a este flagelo, los algoritmos de hash de percepción y la inteligencia artificial se convierte en un aliado para reducir las negligencias. La inteligencia artificial tiene diversas aplicaciones, siendo una de ellas la detección de imágenes mixtificadas, tema de suma importancia y protagonista.

1.1.1. Problema general

¿Cómo el uso del algoritmo hash de percepción contribuye en la detección de imágenes mixtificadas, especialmente en contextos que involucran la pericia de la imagen digital?

1.1.2. Problemas específicos

PE$_1$: ¿De qué manera la sensibilidad del algoritmo hash de percepción aporta en la detección de manipulaciones ocultas o sutiles en imágenes mixtificadas, considerando el Recall?

PE$_2$: ¿Cómo la complejidad y variabilidad de las manipulaciones impactan la detección del algoritmo hash de percepción en la identificación de imágenes mixtificadas, centrándose en la discriminabilidad?

PE$_3$: ¿Es factible determinar un umbral de detección que garantice la robustez de los hashes generados por algoritmos de hash de percepción para mejorar la identificación de similitudes entre imágenes mixtificados?

1.2. Objetivos

1.2.1. Objetivo general

Evaluar de qué manera el algoritmo hash de percepción contribuye al mejoramiento en la detección de imágenes mixtificadas en el contexto de la pericia de la imagen digital.

1.2.2. Objetivos específicos

OE$_1$: Evaluar el aporte de la sensibilidad del algoritmo hash de percepción mediante el Recall para detectar manipulaciones sutiles u ocultas en imágenes mixtificadas.

OE$_2$: Determinar cómo la complejidad y la variabilidad de las manipulaciones afectan la capacidad de detección del algoritmo hash de percepción mediante la discriminabilidad para detectar imágenes mixtificadas.

OE$_3$: Determinar un umbral de detección que asegure la robustez de los hashes generados por algoritmos de hash de percepción, con el fin de mejorar la identificación precisa de similitudes entre imágenes manipuladas.

1.3. Justificación e Importancia

La investigación es crucial para abordar la necesidad de encontrar métodos efectivos que garanticen la autenticidad y credibilidad de las imágenes digitales, subrayando la importancia de la pericia digital en la autenticación del contenido visual en la era digital.

La presente investigación va destinada a mejorar la capacidad humana que mediante los algoritmos hash de percepción puedan detectar imágenes mixtificadas, falsificadas, adulteradas o manipuladas, la misma que beneficiaría a instituciones fiscalizadoras del estado y del orden, así como peritos o forenses informáticos.

El estudio (H. Wang & Wang, 2018) presenta una estrategia de autenticación pasiva que se basa en la utilización de algoritmos de hash perceptual y técnicas de agrupación de datos para identificar y ubicar áreas duplicadas en imágenes. La utilización de cadenas de hash perceptual para crear una representación en forma de vector de características de un bloque de imagen ha demostrado ser efectiva en la prevención de ataques convencionales, mientras que el algoritmo de agrupación de datos propuesto, reemplaza a los métodos tradicionales de ordenamiento lexicográfico. La evaluación de los resultados se basa en métricas como la precisión, la recuperación, el DAR y el FPR. Los resultados obtenidos en experimentos indican que el enfoque propuesto demuestra resistencia ante ciertos tipos de ataques específicos, como ajustes en el contraste, la luminosidad y la tonalidad de las imágenes, lo cual sostiene como un estudio base para investigaciones con ataques más violentos.

La investigación (Ding et al., 2023) presenta un enfoque innovador para abordar la autenticación de integridad en imágenes HRRS, enfocándose en el hash perceptual basado en transformer, debido que es método de extracción de características mejorada llamada Swin-Unet que permite identificar características relevantes en las imágenes para posteriormente generar las secuencias del hash por medio de un proceso de codificación de características que combina mecanismos de mapeo con análisis de componentes principales (PCA). Como resultado destacan las mejoras en robustez y detección de manipulaciones, sin embargo, mencionan que se debería de enfocar estudios para desafíos no abordados, como manipulaciones complejas como la rotación y el escalado.

Según la investigación (Du et al., 2020) realiza una revisión exhaustiva de la literatura reciente sobre el uso de funciones de hash perceptual en la autenticación de imágenes. Tiene como objetivo principal proporcionar un análisis completo de las técnicas más avanzadas explorando su estructura general de estas técnicas, su clasificación en la detección por medio de manipulaciones basadas en hash, el procesamiento discute la reducción de dimensiones y el ajuste de la iluminación para crear el código hash basado en características de la imagen. La investigación concluye resaltando la importancia en futuras investigaciones para áreas que se dedican a la exploración de información, autenticación de imágenes y la combinación efectiva de características de imágenes en el desarrollo de métodos de clasificación avanzados para tomar decisiones precisas.

En este contexto, la pericia informática desempeña un papel crucial como un valioso respaldo para la labor del investigador arbitral. Este proceso se lleva a cabo con total apego a los rigurosos estándares de cuidado, confidencialidad y precisión. El aspecto más destacado de la investigación es proporcionar un apropiado algoritmo de hash de percepción, para ser aplicado en la prevención y resolución de delitos informáticos.

Es importante La pericia informática es fundamental para la legalidad y jurisprudencia en la investigación arbitral. La confidencialidad y rigurosidad son esenciales para contextualizar las evidencias. La investigación científica de nuevos temas de inteligencia artificial es posible y se puede aplicar en la pericia de imágenes digitales, lo que ayuda a limitar los delitos informáticos mediante el uso de algoritmos hash de percepción.

1.4. Hipótesis

1.4.1. Hipótesis general

Se postula que el uso del algoritmo hash de percepción contribuye significativamente al mejoramiento de la detección de imágenes mixtificadas en el contexto de la pericia de la imagen digital.

1.4.2. Hipótesis específicas

HE$_1$: Se espera que la sensibilidad del algoritmo hash de percepción aporte significativamente su capacidad para detectar manipulaciones ocultas o sutiles, reflejándose en diferencias del Recall.

HE$_2$: La complejidad y variabilidad de las manipulaciones tendrán un impacto directo en la capacidad del algoritmo hash de percepción para detectar imágenes mixtificadas, evidenciándose cambios significativos en su capacidad discriminabilidad.

HE$_3$: Al determinar un umbral de detección óptimo para los hashes generados por algoritmos de hash de percepción garantizará la robustez y mejorará la identificación precisa de similitudes entre imágenes mixtificadas.

1.5. Variables

La identificación de las variables en la presente información se distingue en las siguientes características que es de interés de analizar el comportamiento de la investigación.

- Variable independiente (X):
  Algoritmo hash de percepción.

- Variable dependiente (Y):
  Detección de imágenes mixtificados en entorno a la pericia de la imagen digital.

1.5.1. Operacionalización de variables

1.5.1.1. Variable Independiente

La operacionalización de las variables del estudio se consigna de la siguiente manera.

Tabla 1: *Operacionalización de la variable independiente*

Variable independiente (X): **Algoritmo hash de percepción**
Definición conceptual: Como una forma alternativa de garantizar una protección eficiente de los derechos de autor de las imágenes, se ha propuesto el hash de imagen y, por lo tanto, calcular los valores de hash de estas características. El hash de la imagen depende del contenido de la imagen en sí (Prungsinchai, 2014).

Definición operacional: Para medir la presente variable se aplicará etapas como adquisición de datos, preprocesamiento, extracción de características y el reconocimiento para desarrollar la determinación el hash perceptual de la imagen. Además, la sensibilidad del algoritmo hash de percepción se evaluará mediante la medición de la distancia de Hamming entre los hashes generados, lo que permitirá cuantificar su capacidad para detectar y diferenciar pequeños cambios o variaciones en las imágenes analizadas.

Dimensiones	**Indicadores**	**Índice**
Capacidad de detección	Sensibilidad	Capacidad para detectar cambios mínimos.
	Discriminabilidad	Diferenciar dos hashes distintos
	Robustez	Similitud de hashes

Fuente. Elaboración propia

1.5.1.2.Variable Dependiente

Tabla 2: *Operacionalización de la variable dependiente*

Variable dependiente (Y): **Detección de imágenes mixtificados en entorno a la pericia de la imagen digital**
Definición conceptual: La autenticidad/identificación de una imagen se realiza mediante la comparación de los valores hash de los datos originales y los datos de la consulta utilizando funciones específicas (Prungsinchai, 2014).
Definición operacional: El análisis de los datos con respecto a la variable para la medición será correspondido mediante la taza de detección de imágenes alteradas.

Dimensiones	**Indicadores**	**Índice**
Cualidades	Originalidad	Elementos visuales poco conocidos
	Autenticidad	Fabricación completa de la imagen
	Integridad	imágenes sin alteraciones ni manipulaciones

Fuente. Elaboración propia

II. MARCO TEÓRICO

2.1. Antecedentes

(Samanta & Jain, 2021) se centra en la detección de alteraciones en imágenes digitales, tiene un enfoque comparativo de los algoritmos de hash perceptual, el estudio detalla dos algoritmos de referencia: phash y whash, que se basan en técnicas de transformación de imágenes como DCT y DWT para generar representaciones perceptuales de las mismas, mediante dos conjuntos de datos públicos, conocidos como CASIA2.0 y Photoshop Battle, estos algoritmos son sometidos a comparación, principalmente en cuanto a su capacidad para detectar cambios en el contenido de las imágenes frente a aquellas que mantienen su contenido original intacto. Los resultados del estudio revelan que el algoritmo SVD se desempeña mejor en la detección de manipulaciones que alteran el contenido de las imágenes, mientras que phash y whash obtienen mejores resultados cuando se trata de detectar alteraciones que conservan el contenido original.

(Hussain et al., 2022) realizaron la investigación titulada "An Efficient Supervised Deep Hashing Method for Image Retrieval" donde proponen un método de hashing profundo supervisado para la recuperación de imágenes. El objetivo principal del método es convertir los datos de entrada en un código binario de baja dimensión utilizando proyecciones no lineales. Se utiliza una red neuronal convolucional (CNN) para lograr un sistema de hashing de extremo a extremo. Además, se diseña una función de pérdida que busca mantener la similitud entre las imágenes y minimizar el error de cuantificación al proporcionar una distribución uniforme de los bits de hash. Una de las contribuciones principales del método propuesto es la capacidad de generar múltiples códigos binarios cortos utilizando proyecciones no lineales

(Sánchez Gálvez, 2021) realizó la investigación titulada "Análisis forense sobre imágenes digitales, un ejemplo de su aplicabilidad", donde manifiesta, que, la creciente incidencia de delitos informáticos y la manipulación de imágenes digitales en procesos judiciales ha generado una mayor demanda de técnicas y herramientas efectivas para el análisis forense de imágenes digitales. El análisis forense de imágenes digitales se refiere a un conjunto de técnicas utilizadas para detectar y analizar modificaciones o alteraciones realizadas en imágenes digitales. Estas técnicas son consideradas indiciarias, lo que significa que proporcionan

indicios de la existencia de posibles alteraciones en una imagen. El artículo menciona que una imagen digital es un arreglo de píxeles que contiene información sobre la intensidad de color en cada pixel. Además, se mencionan distintas técnicas para realizar modificaciones en imágenes digitales, desde ajustes de tonalidad hasta la creación de fotomontajes profesionales. A través del estudio de casos concretos, se examinarán las técnicas como el Análisis de metadatos que se recomienda para iniciar el análisis de imágenes digitales ya que los archivos de una imagen cuentan con información tales como dispositivo de captura de la imagen, fecha de creación, geolocalización, entre otros.

(X. Wang et al., 2021) realizaron la investigación titulada "Perceptual hash –based coarse – to – fine grained image tampering forensics method", quienes describen un método forense para detectar manipulaciones de imágenes, utilizando hashes perceptuales. El método se basa en calcular un hash perceptual de una imagen original y compararlo con el hash de una imagen sospechosa que se cree que ha sido manipulada. Si los hashes no coinciden, se utilizan técnicas de análisis de patrones para identificar la región de la imagen que ha sido alterada; luego, se utiliza una técnica de análisis de texturas para detectar las diferencias sutiles en la textura de la imagen manipulada en comparación con la imagen original. El método se probó en varios conjuntos de datos y se demostró que es efectivo para detectar manipulaciones de imágenes, incluso cuando se utilizan técnicas avanzadas de ocultamiento. La utilidad de la investigación proporciona en áreas como la seguridad informática y la forense digital; además, el uso de hash perceptuales en el método facilita una forma eficiente y escalable de comparar imágenes y detectar manipulaciones.

(López-Cantos & Gasteazi, 2019) realizaron la investigación titulada "Fake pictures. Falsificación de imágenes científicas y avances actuales en el análisis forense. Análisis de casos", donde manifiestan que, durante el proceso fotográfico se realizan intervenciones en la imagen original que dejan rastros, aunque los falsificadores puedan eliminar los metadatos. Un análisis pericial profundo puede determinar si una imagen fue capturada con una cámara específica y verificar si ha sido modificada. Los programas de manipulación informática, como Photoshop, ofrecen herramientas para realizar diversas operaciones en la imagen. Sin embargo,

los peritos informáticos pueden analizar el rastro de la manipulación y examinar los metadatos para verificar la autenticidad e integridad de la imagen.

(Sabahi et al., 2018) realizaron la investigación "Perceptual Image Hashing Using Random Forest for Content-based Image Retrieval" donde señalan que, el uso de conjuntos de datos de imágenes a gran escala se ha vuelto común y ha generado una demanda de operaciones de búsqueda de imágenes en diversas aplicaciones. Los métodos tradicionales de recuperación de imágenes basados en contenido son efectivos pero complejos y requieren muchos recursos. Para abordar este problema, se propone un marco basado en hashing de imágenes y bosque aleatorio que es rápido y de alto rendimiento. El marco propuesto utiliza técnicas de hashing de imágenes multiclave basadas en la transformada discreta del coseno (DCT) y la transformada discreta wavelet (DWT), junto con el bosque aleatorio basado en el árbol B+ normalizado (árbol NB+). Estas técnicas permiten reducir los vectores de alta dimensión a una dimensión única.

2.2. Bases teóricas

2.2.1. Imagen digital

Es una representación de una matriz bidimensional de una foto electrónica tomada de una escena o escaneada de documentos ya sean fotográficos, manuscritos, textos impresos e ilustraciones. Cuando se realiza una muestra de la imagen digital se confecciona un mapa de ella en forma de cuadricula o casillas de puntos donde cada casilla representa un pixel que es la representación más pequeña de un color o tonalidad. Los dígitos binarios (bits) para cada pixel son almacenados por una computadora en una secuencia, y con frecuencia se los reduce a una representación matemática (comprimida). Luego la computadora interpreta y lee los bits para producir una versión analógica para su visualización o impresión (Cornell, n.d.).

Ilustración 1: *Imagen digital*



66	255	58	0
66	67	108	90
86	97	38	70
56	56	28	120
56	57	58	70

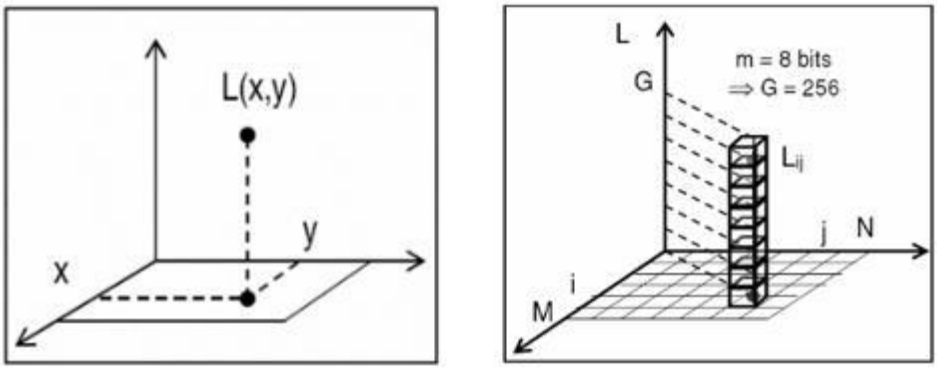
Fuente: (Grammatikopoulos et al., 2007)

Una imagen digital I se puede definir como una función f(x,y), siendo x e y las coordenadas especiales, y la amplitud de f en cualquier punto es la intensidad o nivel de gris de la imagen. Cuando f,x e y son todas cantidades discretas y finitas, entonces se trata de una imagen digital (Gonzáles, 2005). Es posible armar una matriz o arreglo de intensidades, de manera que:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(1,0) & \dots \\ f(0,1) & f(1,1) & \dots \\ \dots & \dots & \dots \end{bmatrix} \quad (1)$$

A cada elemento de esta matriz se lo conoce como pixel, y se suelen limitar los valores que almacenan a un entero sin signo de 8 bits, que da un rango de posibles valores de entre 0 y 2552. Los valores menores que 0, o mayores que 255, son "saturados" y no se puede representar información por debajo o por encima de ellos, un fenómeno conocido como clipping. Hay que tener en cuenta que las variables especiales x,y son valores discretos, usualmente valores enteros. La cuantización de la imagen asignará cada localización discreta (x,y) un valor entero 2b, con b siendo valores de: 2, 4, 8, 12, 16 o 32 bits por píxel. De esta forma la resolución radiométrica describirá el número de bits por píxeles en una imagen (Gonzáles Aguilera, 2005).

Ilustración 2: *Cuantización de la imagen digital*



Fuente: (Gonzáles Aguilera, 2005)

Un píxel es interpretado generalmente como una entidad cuadrada o rectangular a pesar de que el sensor que ha capturado la imagen tenga un campo de vista que no se corresponde con esas dimensiones. Sin embargo, una imagen digital es casi siempre presentada como un conjunto de píxeles homogéneos de igual tamaño. Una imagen en escala de grises tendrá una única matriz de intensidades, que indicarán el nivel de intensidad de la imagen en cada punto. Si se trata de una imagen a color, la representación más usual consiste en utilizar tres matrices de intensidad, correspondientes a colores rojo, verde y azul, o RGB. A cada una de estas matrices de intensidad se las suele llamar "canal". Otros espacios de color representan la imagen digital en función de otras características, por ejemplo, HSL tiene tres canales dedicados al tono (hue), saturación (saturation) y nivel de brillo (levels). Finalmente, también podemos definir el concepto de imagen digital (Gonzáles Aguilera, 2005) como el archivo informático resultante de una discretización de una imagen, natural o sintética, en elementos de imagen, llamados píxeles.

2.2.2. Representación de la imagen digital

Cuando se hace referencia a una imagen digital bidimensional, ésta es asociada a una función $f(x, y)$, donde el punto (x, y) representa las coordenadas espaciales y el valor de f en un punto cualquiera (x, y) es proporcional al nivel de brillo o color de la imagen en este punto (Young et al., 2007). Entonces una imagen digital es una función $f(x, y)$ discretizada tanto en las coordenadas espaciales como en el brillo.

Ilustración 3: *Imagen representada como función f(x,y)*



Fuente: Modificado según (Young et al., 2007)

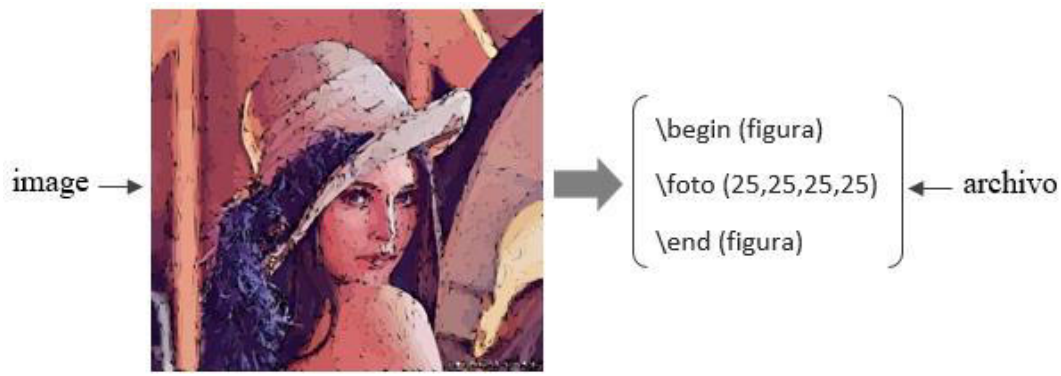
Las imágenes fijas o simplemente imágenes, se generan en las computadoras mediante dos formas: imagen vectorial o imagen de mapa de bits. Los sistemas adquieren los datos pixel a pixel, la imagen es guardada como una grilla de píxeles.

2.2.3. Tipos de imágenes digitales

2.2.3.1.Imagen vectorial

Se forman a partir de elementos topológicos, puntos, líneas y/o polígonos. En estas imágenes las capas generadas se rellenan de colores.

Ilustración 4: *Imagen vectorial o trazos de vectores*



Fuente: Adaptado de (Roque Arias, 2002)

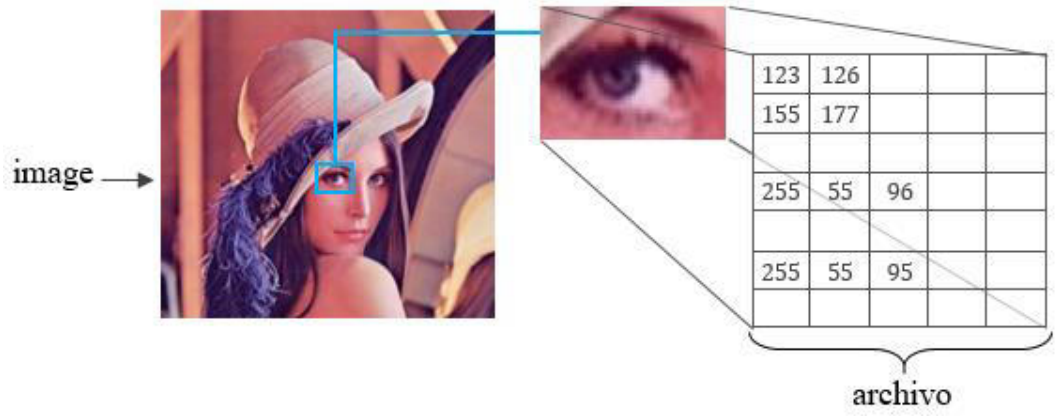
La imagen vectorial no tiene resolución ni número de colores, ampliando o reduciendo el módulo de los vectores (la longitud de las líneas), se obtienen imágenes mayores o menores sin que se produzca una pérdida de calidad (Roque Arias, 2002). La imagen vectorial conocido como trazos con vectores se emplean para describir objetos a través de líneas, cajas, círculos, polígonos y otras formas

gráficas que pueden ser expresadas matemáticamente haciendo uso de ángulos, coordenadas y distancias.

2.2.3.2.Imagen de mapa de bits

Son usadas para imágenes de foto realismo y cuando se desea elaborar trazos complejos que requieren en detalles finos.

Ilustración 5: *Imagen de mapa de bits o imagen raster*



Fuente: Adaptado de (Roque Arias, 2002)

Una imagen raster o imagen de mapa de bits o bitmap es una matriz de información simple, con filas y columnas que describe los puntos individuales, considerados como los elementos más pequeños dados por la resolución del monitor de la computación u otro dispositivo de salida gráfica, por lo tanto, la unidad fundamental de la imagen es llamada pixel.

Son ideales para representar imágenes que manejan una gama amplia de colores, demostrando una colección de pixeles que describen una imagen. En esencia, es una matriz de puntos bidimensionales que contiene la información necesaria para generar un mapa de bits

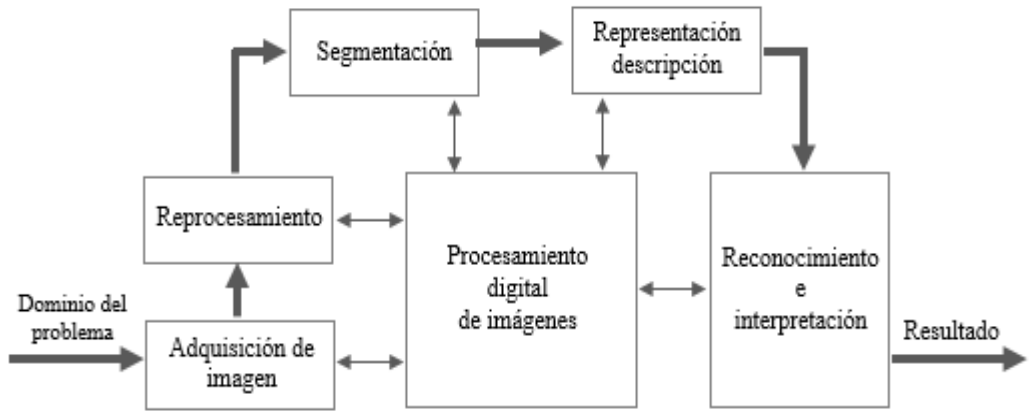
2.2.4. Procesamiento de imágenes

El procesamiento de imágenes es el término usado para denominar las operaciones desarrolladas sobre un set de datos de imagen para mejorarlas de alguna forma, para ayudar su interpretación o para extraer algún tipo de información útil de ella (Yuras, 1998). El uso de la computación en diferentes dominios ha permitido sofisticar la naturaleza del procesamiento de las imágenes, con el propósito de extraer al máximo, cualquier información disponible en las mismas que sea de utilidad como herramienta.

El estudio de técnicas avanzadas de procesamiento digital de imágenes aplicadas sobre cierta región de una imagen, que involucran el análisis de la información que poseen los puntos ubicados alrededor del que se desea estudiar, son comúnmente denominadas “Técnicas de procesamiento por Área o Región”. Las imágenes observadas en una escena digital de dos dimensiones (2D), se la presenta como un arreglo bidimensional, tal arreglo se denomina matriz imagen, es decir que, al obtener la matriz imagen, lo que realmente se hace es establecer una malla, donde cada punto del mismo forma un elemento de la imagen (Bravo, n.d.)

Las etapas del procesamiento de la imagen digital comprenden esencialmente en cinco fases, de las cuales, la adquisición, el reprocesado, la segmentación y la representación son aplicables a la comprensión de imágenes y son de vital importancia para un buen desempeño de un sistema que involucre el tratamiento digital de imágenes.

Ilustración 6: *Etapas del procesamiento de la imagen digital*



Fuente: Modificado (Gonzáles Aguilera, 2005)

En la adquisición de la imagen se hace referencia a la obtención de la imagen digital, para ello, es necesario un sensor de imágenes que provea la posibilidad de digitalizar la señal producida por el sensor. El sensor para adquirir la imagen digital es una cámara digital, monocromática o de color, un escáner que transforma una imagen analógica a su representación en imagen digital, una cámara de video. La naturaleza del sensor y la imagen viene determinada por el tipo de aplicación y esencialmente por el dominio del problema y los resultados que se espera obtener a través de la imagen (Molina et al., 1998).

La digitalización deja los datos con un formato adecuado para su manipulación por parte de la computadora. Éstos son ahora transformados con cualquier función matemática elegida. Cada sistema de procesamiento de imágenes tiene su propio software asociado, cada uno realizando una tarea diferente. La secuencia de tareas aplicadas a una imagen se denomina como procesamiento de imágenes, la cual varía dependiendo de las metas que se tengan en vista (Yuras, 1998).

Una vez obtenido la imagen digital, el siguiente paso es el reprocesado de la imagen. La función básica de este paso es la de mejorar la imagen de forma tal que las posibilidades de éxito sean altas en los procesos posteriores. Tales mejoras engloban entre otras posibles: mejorar el contraste de la imagen, eliminar segmentos o regiones que no aportan información al objeto de estudio, eliminar el ruido y realizar transformaciones a la imagen (Molina et al., 1998).

La segmentación, definida de manera general, divide una imagen de entrada en sus partes constituyente u objetos. En general, la segmentación es una de las tareas más difíciles en el procesamiento digital de imágenes. Por un lado, un procedimiento de segmentación tosco trae como consecuencia un largo proceso hasta la solución exitosa del problema. Por otro lado, los algoritmos de segmentación pobres o erróneos casi siempre garantizan un eventual fracaso. En términos del procesamiento digital de imágenes, el rol principal de la segmentación es el de dividir una imagen digital en sus partes constituyentes u objetos. La salida del proceso de segmentación generalmente son datos rústicos de pixeles, constituyendo ya sea el borde de una región o todos los puntos en la región misma (Calvet et al., 1993).

La elección de una representación es solo una parte de la solución para transformar los datos de pixel a una forma adecuada, para ser posteriormente tratada por la computadora. También es necesario que se especifique un método para describir los datos de forma que se resalten los rasgos o características de interés del objeto de estudio. La descripción consiste en extraer rasgos (alto, ancho, tamaño, formato) con alguna información cuantitativa de interés o que sea fundamental para diferenciar una clase de objetos de otra (Gonzáles Aguilera, 2005).

En esencia, el objetivo de la representación y descripción de imágenes o regiones de una imagen es la extracción de características o caracterización de un objeto para su posterior procesado y reconocimiento. La representación de cierta región de una imagen implica dos posibilidades; realizarlo en función de su frontera o borde o realizarlo en función de sus características internas (Molina et al., 1998).

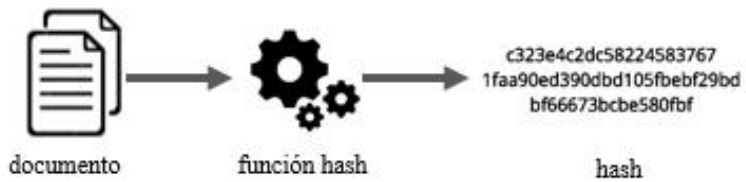
El reconocimiento es el proceso que asigna una identificación a un objeto basándose en la información proporcionada por sus descriptores. La interpretación involucra la asignación de significado a un conjunto de objetos reconocidos (Calvet et al., 1993).

2.2.5. Función hash.

La función hash o algoritmo hashing es un procedimiento bien definido o una función matemática que transforman un mensaje (documento) de entrada $\boldsymbol{m}$ en un digesto (de tamaño fijo) $\boldsymbol{d}$, tal que $\boldsymbol{d = f(m)}$, en otras palabras, convierte una gran cantidad de información a un dato de menor dimensión y se encuentro compuesto por una cadena binaria, que usualmente sirve para indexar, el valor que regresa la función hash se le conoce como valor hash, checksum o solamente hash.

El resultado de tal algoritmo se llama $\boldsymbol{hash}$, el cual, representa un identificador de dicho documento determinado según su contenido (Menezes et al., 1996). Para efecto, va a depender la función de hash utilizada y su objetivo; y variará la complejidad en el cálculo del digesto, y se tendrán distintas garantías sobre el resultado

Ilustración 7: *Función Hash*



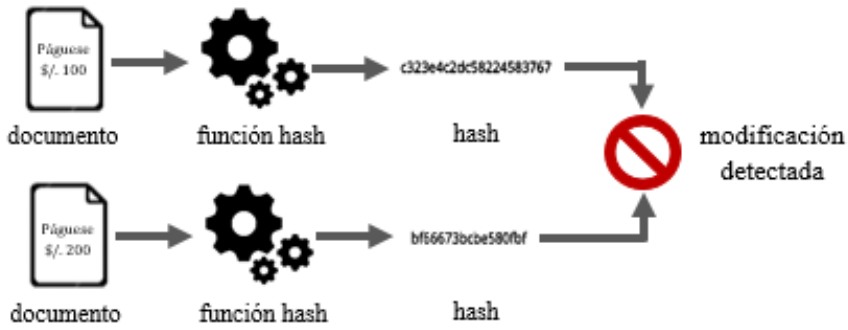
Fuente: (Menezes et al., 1996)

El uso de la función hash crean rápidamente identificadores de documentos, o también llamados imágenes. Uno de los usos más frecuentes de una función de hashing es el de detectar modificaciones en un documento. Cualquier alteración,

por más mínima que sea, un algoritmo hashing seguro debería dar un hash que sea totalmente independiente y sin correlación con el hash del documento original.

Este mecanismo permite detectar si un documento fue adulterado, asegurando la integridad del mensaje.

Ilustración 8: *Detección de documentos adulterados comparado con los hashes*



Fuente: (Menezes et al., 1996)

De acuerdo al objetivo, la funciones hash lo podemos clasificar de la siguiente manera:

- **Funciones de Índice para Tablas de Hashes**. Permiten ordenar información dentro de una estructura de datos de acceso aleatorio y, en base a una clave o identificador, rápidamente acceder a un contenido. Este tipo de funciones se utilizan para bases de datos, estructuras de dato de acceso rápido, o caches, entre otros.

- **Funciones de Checksum**. Que tienen la capacidad de detectar cambios en un archivo que ha viajado por un medio de transmisión, de manera que se pueda verificar su integridad. Dado que están orientadas a detectar errores de transmisión, su enfoque no es tan riguroso como los hashes criptográficos.

2.2.6. Hash criptográfico

Una función hash criptográfico H(X) mapea un objeto X (usualmente grande) a un valor hash (usualmente pequeño), esto permite comparar dos objetos grandes X y Y, sólo comparando sus valores hash H(X) y H(Y), una igualdad matemática de H(X) y H(Y) implica que X y Y sean iguales con poca probabilidad de error. Usando funciones hash criptográficas podemos tener un método eficiente

para saber si un objeto X este contenido en set grande Y=Y_i, en lugar de comparar X con todo el set Y es suficiente con comparar el set de valores hash {h_i=H(Y_i)} con el valor H(X), este método es más eficiente porque la cantidad de bits que se deben comparar es menor.

Las propiedades de las funciones hash criptográficas son:

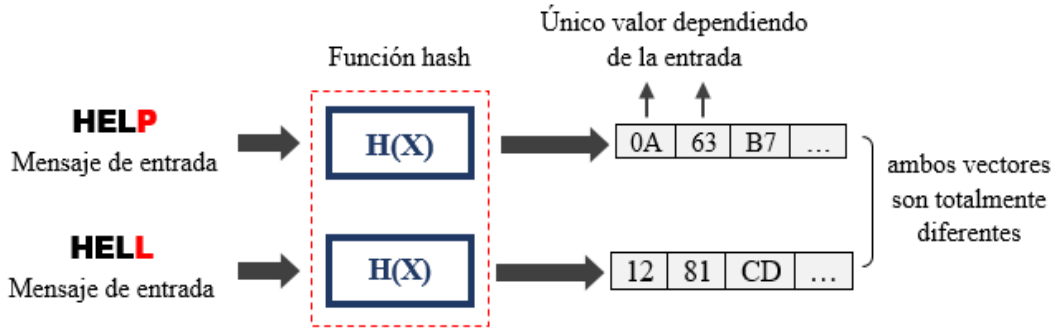
- Un solo sentido, dado H(X) debe ser difícil de encontrar el objeto original X, y dado X y H(X) es difícil encontrar un objeto X' donde X'≠X tal que H(X^')=H(X).
- Libre de colisiones, es difícil encontrar dos objetos X y Y tales que sus valores hash sean iguales (H(X)=H(Y)).
- Uso de llaves, sin el conocimiento de una llave K, es casi imposible determinar H K(X) de cualquier objeto X.

Aunque este tipo de esquemas son muy útiles debido a que el mensaje debe de estar intacto para poder generar el mismo valor hash. Sin embargo, para el problema de image hashing este método falla cuando se utilizan este tipo de funciones hash criptográficas.

Para contenido multimedia no estamos interesados en igualdades matemáticas sino en igualdad de contenido, ya que existen manipulaciones sin alteración de contenido como comprensión, realces, conversión analógica – digital, ataques geométricos, etc. Las funciones hash criptográficas no podrían procesar dichas manipulaciones como el archivo original.

Este problema podría resolverse si las funciones hash criptográficas tuvieran un comportamiento lineal por ejemplo si contenido perceptualmente similar generara al menos matemáticamente valores hash similares. Sin embargo, las funciones hash criptográficas tienen la propiedad contraria en el sentido en que son bit – sensitive, es decir, si un solo bit en el contenido cambia, el valor hash es totalmente diferente.

Ilustración 9: *Función hash criptográfico, un solo bit cambia, el valor hash es totalmente diferente*



Fuente: Elaboración propia.

Las funciones criptográficas de hash son funciones booleanas que dado un input variable producen un output de longitud fija. Su característica fundamental es que ante la mínima variación en el input el resultado del hash es completamente diferentes. Esto se conoce como "efecto avalancha". Se usan principalmente en el control de integridad y junto a una clave para encripción, para firma digital. Por esta observación podemos concluir que para la autenticación de contenido multimedia no podemos utilizar este tipo de funciones hash por lo que se busca funciones basadas en contenido.

2.2.7. Hash basado en Contenido

Reconocimiento de contenido usando características puede caracterizarse en dos grupos, métodos basados en características semánticas y no semánticas (Oostveen et al., 2002), las semánticas son por ejemplo extracción de límites o histograma de la imagen, que tienen un significado importante en la imagen, sin embargo, aquellas que no tiene un significado semántico son robustas a diferentes transformaciones no basadas en contenido. Ambos grupos pueden ser usados para establecer perceptualmente similitudes en imágenes digitales, cabe notar que las características a extraer para image hashing son un poco diferentes a las usadas en métodos de detección.

Para los métodos de detección las características deben facilitar las búsquedas de imágenes que de alguna manera parecen similares o de aquellas que poseen algún objeto similar, mientras que para image hashing se requiere identificar imágenes perceptualmente iguales, excepto por diferencia de calidad o algún defecto de procesamiento de señales. Construir una función hash basado en

contenido no es una tarea trivial, por ejemplo, los humanos no podemos diferenciar entre una imagen original o una versión comprimida o cambiando sus dimensiones en un monto pequeño. En general las funciones hash basadas en contenido tienen las siguientes características (Venkatesan et al., 2000):

- **Robustez**. Por robustez queremos decir que cuando la misma llave es usada, imágenes que perceptualmente son similares deben producir hashes similares.

$$(\boldsymbol{H(I,K)} \approx \boldsymbol{H(I',K)}) \geq \mathbf{1} - \boldsymbol{\theta_1}, \mathbf{0} \leq \boldsymbol{\theta_1} \leq \mathbf{1} \quad (2)$$

Para este caso, se requiere que cualquier par de imágenes perceptualmente similares/idénticas debe tener hashes similares; mientras que, para los datos de imágenes digitales, se introdujeron distorsiones insignificantes perceptualmente en las imágenes originales debido a los canales de comprensión o transmisión de ruido. Por lo tanto, es fundamental garantizar que las imágenes perceptivamente similares muestren hashes de imágenes similares, y los hashes de imágenes deben ser robustecidos a tales ataques de operaciones de procesamiento de contenido (CPO) para fines de identificación.

Ilustración 10: *Ejemplos de copias de imagen "Lena" con CPO*





(e) Ecualización de Histograma (f) Rotación (g) Traducción

Fuente: (Venkatesan et al., 2000)

Perceptivamente, estas imágenes son idénticas en el sistema visual humano (HVS). La robustez perceptiva del hashing de imágenes garantiza que estas imágenes tengan hashes muy similares, si el sistema es lo suficientemente robusto contra estos ataques.

- **Discriminabilidad**. La discriminabilidad del sistema de hash de imágenes garantiza que las imágenes perceptivamente distintas tengan hashes diferentes, demostrando la capacidad para diferenciar dos huellas distintas.

$$(H(I,K)\neq H(J,K))\geq 1-\theta,\ 0\leq\theta\leq 1 \qquad (3)$$

La discriminación significa que al tener I y J (información de entrada que se quiere proteger, verificar y/o autenticar) totalmente diferentes, sus H (información de salida basada en las propiedades de la entrada I y J) correspondientes deben ser a su vez totalmente diferentes (Neelima & Singh, 2014). Esto implica que la clasificación y el uso de funciones resumen puede estar directamente ligado al tipo de contenido sobre el cual se calculará la cadena de caracteres. Por ejemplo, si hablamos de un contenido tal como una señal de voz, es posible hablar de un método perceptual y con cierto grado de discriminación; en este caso, el interés principal de la función resumen debe radicar en evaluar la preservación del contenido, más no en la forma de presentación del mismo.

Esta práctica es comúnmente usada para la emisión de certificados, firmas digitales, generación y verificación de claves. Sus aplicaciones principales están orientadas a la protección y gestión de contenidos, debido a que permite distinguir algún tipo de modificación o manipulación de la información (Haitsma et al., 2001).

- **Imprevisibilidad**. Se logra a través de una llave para generar el valor hash, sin el conocimiento de esta llave el valor hash debe ser imposible de estimar o falsificar, donde fh(x) es la función de masa de probabilidad forh; con esta propiedad, los valores hash deben distribuirse aproximadamente por igual.

$$(H(I,K)); f_h\ (1)\approx f_h\ (0)\approx 0.5 \quad (4)$$

La seguridad es una preocupación importante para el cambio de imagen, incorporando algunas técnicas de pseudo-aleatorización en el proceso de generación de hash de imágenes para mejorar la seguridad de los hashes de imágenes mediante el uso de claves secretas.

- **Compacidad**. El tamaño de los hashes de la imagen debe ser mucho más pequeño que la imagen original.

$$Size(H(I,K) \ll Size(I)) \quad (5)$$

Los hashes de imágenes compactas deberían facilitar el proceso de búsqueda en una base de datos de hashes y deberían requerir menos espacios de almacenamiento (Ou & Rhee, 2010).

2.2.8. Algoritmo hash de Percepción

El hash de percepción son un tipo de firmas de imágenes. Usando una imagen como entrada, el algoritmo hash de percepción produce un hash de ***X*** bits como salida. Al buscar imágenes similares, hashes de una imagen de consulta se compara con una base de datos de hashes de imágenes. A diferencia de un hash criptográfico, donde pequeños cambios en la entrada conducen a grandes cambios en la salida, los hashes perceptuales son estrechamente relacionados entre sí siempre que las imágenes sean similares.

Una propiedad común de la percepción, el algoritmo hash es que las transformaciones de imagen, como la escala, la relación de aspecto o la modificación del color solo da como resultado cambios menores de hash (Krawetz, 2011). Esto hace que los hashes perceptivos sean ideales para las firmas utilizadas en sistemas de identificación de imagen y video. El hash perceptual se basa en características sensoriales de un archivo para realizar el cálculo. Para el caso de

archivos de imagen, los algoritmos de hashes de percepción se enfocan en los patrones visuales que se generan al renderizar una imagen.

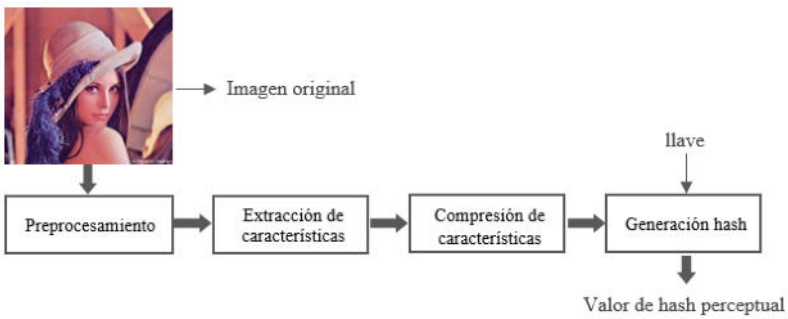
En su estudio de hashes perceptuales, Zauner define los siguientes conceptos de interés (Zauner, 2010):

- Modificación: es una operación sobre el objeto de medios que no afecta sus características.
- Manipulación: es una operación sobre el objeto de medios que si afecta sus características.

Los algoritmos de hashes de percepción deben ser resistentes a las modificaciones, y no confundir las manipulaciones sobre un contenido; es decir, si se compara la similitud de una imagen con otras versiones de la misma, que han sufrido modificaciones, debe identificarlas como la misma imagen, ya que no han cambiado sus características esenciales. Al contrario, si ha habido manipulación de la imagen, el hash de percepción debe identificarlas como imágenes distintas. La comparación entre dos hashes se realiza aplicando la distancia de Hamming (Hamming, 1950) entre ellos, y definiendo un umbral de similitud. Este umbral es dependiente del tipo de hash que se haya aplicado. Un umbral de 0 exige al algoritmo que el hash de las imágenes debe ser idéntico. Un umbral más alto, permitirá detectar modificaciones sobre una imagen, pero descartar manipulaciones.

La descripción general del algoritmo hash de percepción consiste en las siguientes etapas:

Ilustración 11: *Marco de los algoritmos hash perceptuales*



Fuente: (Hamming, 1950)

Preprocesamiento generalmente elimina la información redundante en una imagen, lo que hace que sea más fácil extraer características de la imagen más adelante; la extracción de características es extraer las características principales de la edad un método de extracción específico; la compresión de características viene ser la cuantificación de características es difuminar las características extraídas para mejorar la robustez; y la generación hash es hacer que las características cuantitativas sean más abstractas porque las características cuantitativas pueden llevar a que una gran cantidad de datos sea adecuada como secuencia de salida.

Para la autenticación de imágenes, un algoritmo hash de percepción debe poseer las siguientes propiedades:

- Compacidad. El valor de hash de la imagen debe ser lo más compacto posible, para que sea más fácil para transportar, almacenar y usar.
- Sensibilidad a la Manipulación. Las imágenes visualmente distintas deben tener valores hash significativamente diferentes.
- Robustez Perceptiva. La generación de hash debe ser insensible a cierto nivel de distorsión con respecto a la imagen de entrada.
- Seguridad. El cálculo del hash de imagen depende de una clave secreta, es decir, diferentes claves secretas pueden generar valores hash significativamente diferentes para la misma imagen.

Como parte y en cumplimiento de los objetivos del proyecto de tesis, se ha considerado para el estudio cuatro algoritmos de hashes de percepción de imagen: Algoritmo de average hash (a_hash), algoritmo de difference hash (d_hash), algoritmo de DCT – hash (p_hash) y el algoritmo hash basado en VGG – 16 propuesta en nuestra investigación.

2.2.8.1 Algoritmo de Hash promedio (a_hash)

Este hash de percepción es conceptualmente muy simple y fácil de calcular. Como modelo conceptual para entender otros hashes es interesante, pero no es tan robusto como las otras alternativas. El algoritmo para calcularlo es el siguiente:

- Convertir la imagen a escala de grises.

- Redimensionar la imagen a 8x8. Se llamará a esta imagen, I'.
- Calcular el valor promedio de los 64 pixeles, avg.
- Para cada pixel 〖p'〗_xy de I, el valor de hash H en la posición (x,y) será h_xy=1 si p_xy>avg, o h_xy=0 en caso contrario.

De esta manera, el hash resultante describe la estructura macro de la imagen, cualquier alteración sutil, o que afecte detalles finos, no generará cambios lo suficientemente grandes en el pixel p_xy afectado para impactar en el valor h_xy final. Si en cambio fueran alteraciones mayores, entonces se modificará tanto el valor de p_xy, como el promedio de la imagen. Un cambio de esa magnitud afectaría globalmente al hash H, modificando varios de sus elementos h_xy.

2.2.8.2 Algoritmo de Hash diferencial (d_hash)

Este hash se basa en los gradientes horizontales de la imagen. Su cálculo es relativamente simple, pero el concepto en el que se basa es un mejor descriptor de la estructura macroscópica de una imagen que el utilizado por el average hash. El algoritmo para calcular este hash es el siguiente:

- Convertir la imagen a escala de grises.
- Redimensionar la imagen a 9x8. Se llamará a esta imagen, I'.
- Se toman dos subregiones de I' de tamaño 8x8, α y β, siendo α la subregión de más a la izquierda de I', y β la subregión de más a la derecha.
- El hash final H se construye como un arreglo de 8x8, donde el valor de hash en la posición (x,y) se construye de la siguiente manera: h_xy=1 si α_xy<β_xy, h_xy=0 en caso contrario.

Esta construcción aproxima el gradiente horizontal de la imagen escalada I'. Esto permite que el hash se construya en base a los bordes de imagen, que son más resistentes a las alteraciones que resultan problemáticas para el average hash.

2.2.8.3 Algoritmo de Hash DCT (p_hash)

Este hash se basa en coeficientes DCT calculados sobre la imagen. Si bien el cálculo es más exigente que las alternativas vistas hasta ahora, la implementación del algoritmo lograda no presenta grandes diferencias de velocidad con respecto a los anteriores. El algoritmo para calcular este hash es el siguiente:

- Convertir la imagen a escala de grises.
- Redimensionar la imagen a 32x32. Se llamará a esta imagen, I'.
- Aplicar la transformada DCT sobre I'.
- De DCT (I^'), quedarse con el segmento superior izquierdo de tamaño 8x8. Se llamará a esta imagen I".
- Calcular la mediana de los valores en I^",avg.
- Para cada pixel 〖p"〗_xy de I", el valor de hash final H en la posición (x,y) se construye de la siguiente manera: h_xy=1 si 〖p"〗_(xy)>avg, 〖h〗_xy=0 en caso contrario.

2.2.8.4 Algoritmo Hash basado en VGG – 16 (propuesta)

Este hash utiliza una red neuronal convolucional para extraer características visuales. VGG – 16 es una red neuronal profunda entrenada en grandes conjuntos de datos de imágenes y es conocida por su capacidad para extraer características visuales significativas de las imágenes:

- Importar las bibliotecas necesarias en Python, como keras y otras bibliotecas relevantes para el procesamiento de imágenes y aprendizaje profundo.
- Crear un modelo secuencial en keras, que permitirá agregar capas de forma secuencial.
- Inicializar el modelo VGG – 16 pre – entrenado utilizando la arquitectura y los pesos aprendidos del modelo VGG – 16, que se encuentra disponible en keras o en otros recursos.

- Pasar o transferir los datos de entrada a través de la capa densa del modelo VGG – 16 para obtener la representación de características o características visuales de las imágenes.

- Binarizar las características obtenidas.

2.2.9. Normalización de imágenes

En el procesamiento de imágenes, la normalización es un proceso que cambia el rango de valores de intensidad de píxeles. Las aplicaciones incluyen imágenes con poco contraste debido al deslumbramiento, por ejemplo, la normalización a veces se denomina estiramiento de contraste o estiramiento de histograma. En campos generales del procesamiento de datos como el procesamiento de señales digitales se denomina expansión de rango dinámico (Gonzáles & Woods, 2008).

La normalización automática en el software de procesamiento de imágenes normalmente se normaliza al rango dinámico completo del sistema numérico especificado en el formato de archivo de imagen.

- Algoritmo de Normalización. Como se muestra en la Ilustración 12, tenemos las entradas (a), (b) o (c) y la salida Ilustración (d) del algoritmo de normalización de imágenes cuando (a), (b) o (c) es la entrada.

Ilustración 12: *Entradas y salidas del algoritmo de normalización de imágenes*



Fuente: (Ping Dong & Yongyi, 2012)

(Ping Dong & Yongyi, 2012) propuso un algoritmo para el procedimiento de normalización que consiste en los siguientes pasos:

- Centramos la imagen $\boldsymbol{f(x,y)}$, esto lo logramos haciendo la matriz $\boldsymbol{A} = \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{pmatrix}$ y el vector $\boldsymbol{d} = (\boldsymbol{d_1}\ \boldsymbol{d_2})^{\boldsymbol{T}}$ en la ecuación (28) con:

$$\boldsymbol{d_1} = \frac{\boldsymbol{m_{1,0}}}{\boldsymbol{m_{0,0}}}, \qquad \boldsymbol{d_2} = \frac{\boldsymbol{m_{0,1}}}{\boldsymbol{m_{0,0}}} \tag{6}$$

  Donde $\boldsymbol{m_{10}}, \boldsymbol{m_{01}}$ y $\boldsymbol{m_{00}}$ son los momentos de $\boldsymbol{f(x,y)}$ definidos en la ecuación (25), con este paso hacemos que sea invariante a traslación. El resultado de este paso lo denotaremos como $\boldsymbol{f_1(x,y)}$.

- Aplicamos shearing en la dirección $\boldsymbol{x}$ a $\boldsymbol{f_1(x,y)}$ con la siguiente matriz $\boldsymbol{A} = \begin{pmatrix} \mathbf{1} & \boldsymbol{\beta} \\ \mathbf{0} & \mathbf{1} \end{pmatrix}$ y el vector $\boldsymbol{d} = \mathbf{0}$ y denotamos la salida como $\boldsymbol{f_2(x,y)}$. Para determinar beta usamos:

$$\boldsymbol{\mu_{30} + 3\beta\mu_{21} + 3\beta^2\mu_{12} + \beta^3\mu_{03} = 0} \tag{7}$$

  De esta ecuación podemos tener hasta tres raíces, si una de ellas es real y las otras complejas, utilizamos la raíz real. Si dos o las tres raíces son reales utilizamos la media de las raíces reales. Si todas las raíces son complejas $\boldsymbol{\beta = 0}$. En casos muy raros $\boldsymbol{\beta}$ puede tener un número infinito de soluciones, si todos los momentos son cero, esto pasa cuando la imagen es rotacionalmente simétrica (Shen, 1997).

- Aplicamos shearing en la dirección $\boldsymbol{y}$ a $\boldsymbol{f_2(x,y)}$ con la siguiente matriz $\boldsymbol{A} = \begin{pmatrix} \mathbf{1} & \mathbf{0} \\ \boldsymbol{\lambda} & \mathbf{1} \end{pmatrix}$ y el vector $\boldsymbol{d} = \mathbf{0}$ y denotamos la salida como $\boldsymbol{f_3(x,y)}$. $\boldsymbol{\lambda}$ es definida con la ecuación:

$$\boldsymbol{\mu_{11} = \lambda\mu_{20} + \mu_{11}} \tag{8}$$

  y hacemos $\boldsymbol{\mu_{11} = 0}$ obtenemos:

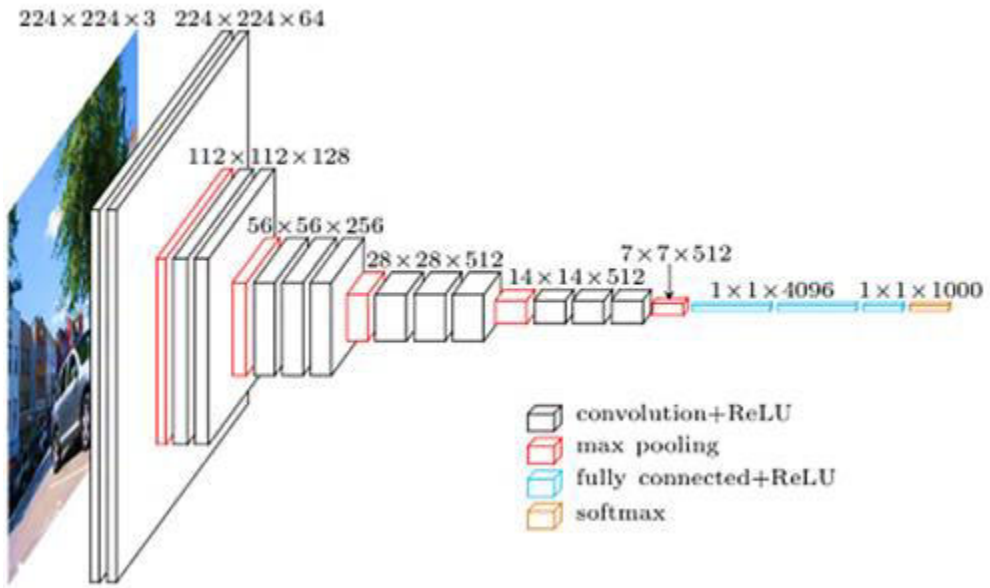
$$\boldsymbol{\lambda} = \frac{\boldsymbol{\mu_{11}}}{\boldsymbol{\mu_{20}}} \tag{9}$$

- Finalmente, se le aplica escalamiento a $\boldsymbol{f_3(x,y)}$ en las dimensiones deseadas para tener la imagen normalizada $\boldsymbol{g(x,y)}$.

2.2.9.1. VGG – 16.

Es considerada como una de las mejores arquitecturas de modelos para la visión artificial. Lo diferente de VGG – 16 es el uso de capas de convolución de filtros, capas completamente conectadas y finalmente una capa de salida con softmax como función de activación (Simonyan & Zisserman, 2015).

Ilustración 13: *Arquitectura del modelo VGG – 16*



Fuente: (Simonyan & Zisserman, 2015)

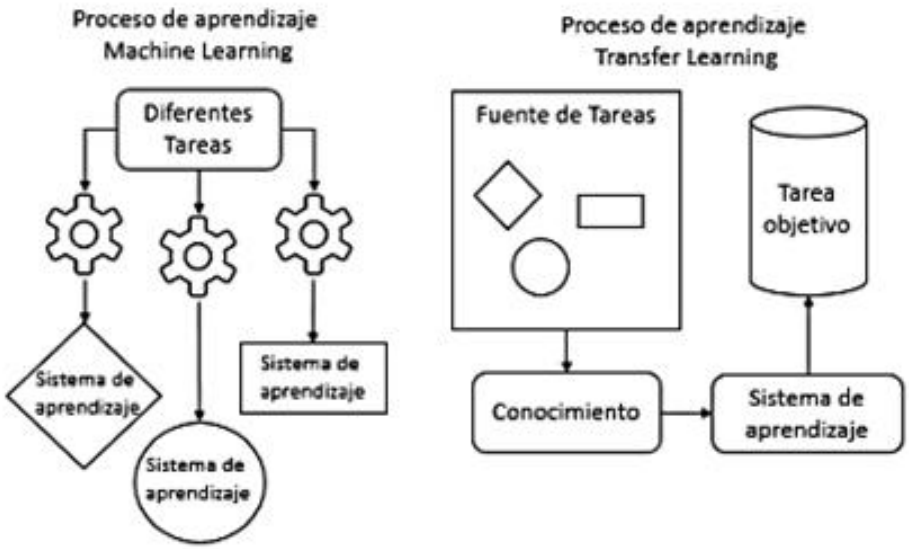
2.2.10. Transfer learning

Tradicionalmente, los algoritmos de aprendizaje son diseñados para afrontar los problemas de manera independiente. Dependiendo de los requerimientos del caso y de la información disponible, un algoritmo es aplicado para entrenar el modelo para dicha tarea. Sin embargo, el proceso denominado "Transfer Learning" lleva el aprendizaje un paso más allá y "lo acerca" a la manera en que los humanos utilizan la información a través de las tareas. En términos simples, el proceso consiste en reutilizar un modelo existente y ya entrenado para una tarea relacionada, extendiéndose o adaptándola para una nueva tarea (Sarkar et al., 2018).

Por su parte, (Olivas et al., 2009) lo define como "la mejora del aprendizaje en una nueva tarea mediante la transferencia de conocimientos de una tarea relacionada que ya se ha aprendido", por lo que podemos inferir que de esta manera se obtiene un rápido progreso y un alto desempeño. Adicionalmente, esta técnica

es ampliamente utilizada por su enfoque diverso y (Ou & Rhee, 2010) lo define que el Transfer Learning es “la idea de romper el paradigma de resolver un problema aislado y utilizar el conocimiento adquirido para enfocarlo en una tarea similar”. Para simplificar la comparación, en la Ilustración 18, se puede apreciar la diferencia entre el Machine Learning “tradicional” y el Transfer Learning.

Ilustración 14: *Proceso de aprendizaje Machine Learning y Transfer Learning*



Fuente: (Hidalgo Barrientos et al., 2021)v

Si el conjunto de datos original con el que se entrenó la red neuronal es suficientemente grande y general, entonces la jerarquía espacial de las características (features) aprendidas por la red permite al modelo entrenado actuar como un modelo genérico del mundo visual y, por lo tanto, sus características pueden resultar útiles para muchos problemas diferentes de visión por computadora, a pesar de que estos nuevos problemas pueden involucrar clases completamente diferentes a las de la tarea original (Torres, 2020).

Hay dos formas de utilizar una red entrenada previamente, las cuales presentamos a continuación:

- **Extracción de Características (Feature Extractiony)**. Es un proceso de reducción de la dimensionalidad, por lo cual, un conjunto inicial de datos en bruto se reduce a grupos más manejables para su procesamiento. Una característica de estos grandes conjuntos de datos es un gran número de variables que requieren muchos recursos

informáticos para su procesamiento. La extracción de características es el nombre de los métodos que seleccionan y/o combinan variables en características, reduciendo efectivamente la cantidad de datos que deben procesarse, sin dejar de describir con precisión y por completo el conjunto de datos original.

El proceso de extracción de características es útil cuando se necesita reducir el número de recursos necesarios para el procesamiento sin perder información importante o relevante. La extracción de características también puede reducir la cantidad de datos redundantes para un análisis determinado. Asimismo, la reducción de los datos y los esfuerzos de la máquina en la construcción de combinaciones variables (características) facilitan la velocidad de aprendizaje y los pasos de generalización en el proceso de aprendizaje de la máquina (Torres, 2020)

- **Sintonía Fina (Fine – Tuning).** Es un enfoque para transferir el aprendizaje en el que adaptamos el dominio bajo el que entrenamos el modelo. Dicho con otras palabras, entrenamos el mismo modelo con distintos conjuntos, los cuales pueden tener distintas distribuciones de clases o incluso incluyen nuevas clases (Torres, 2020).

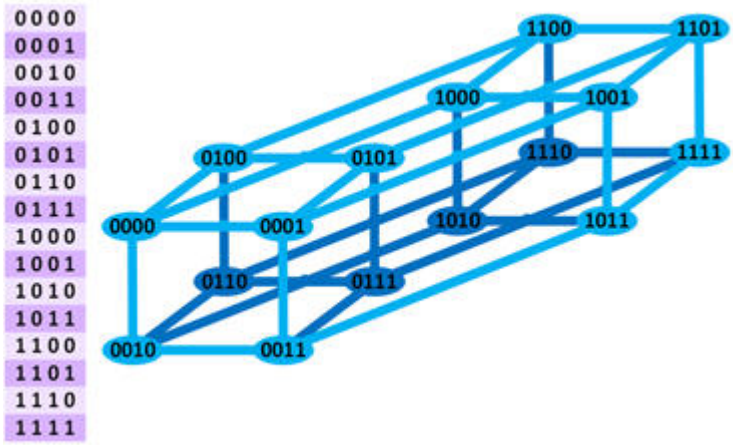
2.2.10.1. Distancia de Hamming.

Se utiliza típicamente con vectores booleanos o de cadenas, identificando los puntos en los que los vectores no coinciden. La distancia de Hamming es una herramienta de ordenamiento que calcula la diferencia entre los extremos de los intervalos. Así, en este método no se diferencia entre un exceso o un defecto respecto al ideal, por lo que se evalúan ambos de forma equivalente (Montero Morales, 2008)

El algoritmo de la distancia de Hamming calcula una puntuación de coincidencia para dos cadenas de datos calculando el número de posiciones en las que los caracteres difieren de una cadena de datos a otra. En el caso de que las cadenas tengan una longitud diferente, cada carácter adicional de la cadena más larga se cuenta como una diferencia de una cadena a otra (Informática Guía de transformaciones de Developer, 2019).

Si tenemos un bit, por ejemplo, el 0 y se cambia a 1, la distancia de Hamming es de 1; la misma distancia si se cambia de 1 a 0.

Ilustración 15: *Distancia de Hamming de 4 dimensiones*



Fuente: (Hidalgo Barrientos et al., 2021)

Las propiedades de detección y corrección de errores de un código dependen de su distancia de Hamming. Esta representa la cantidad de símbolos en los que una cadena de datos se diferencia de otra cadena y se simboliza con la letra "d". Para determinar la cantidad de bits diferentes, en el caso de una cadena de bits diferentes, en el caso de una cadena binaria, basta aplicar una operación OR exclusivo a las dos cadenas y contar la cantidad de bits 1 en el resultado; por ejemplo:

$$\begin{array}{l} 10001001 \\ 10110001 \\ 00111000 \end{array} \qquad d = 3, N = 8$$

En el caso anterior se muestra que la distancia de Hamming es d = 3 y la longitud de la cadena de datos es N = 8.

2.2.10.2. Sensibilidad (Recall).

la sensibilidad mide la proporción de casos positivos reales que son correctamente identificados por el modelo o la prueba. Se calcula como la proporción de verdaderos positivos (TP) dividida entre la suma de verdaderos positivos (TP) más falsos negativos (FN).

$$Sensibilidad = \frac{TP}{TP + FN}$$

TP: Verdadero Positivos

FN: Falsos Negativos

2.2.10.3. Precisión

la precisión mide la proporción de imágenes identificadas como mixtificadas que realmente lo son, es decir, mide la exactitud de las predicciones positivas realizadas por el algoritmo.

$$Precisión = \frac{TP}{TP + FP}$$

TP: Verdadero Positivo

FP: Falsos Positivos

2.2.10.4. F1-score

El F1-score, también conocido como puntaje F1, es una medida de la precisión de un modelo de clasificación. Se calcula a partir de la precisión y el recall del modelo y proporciona una medida más equilibrada entre ambas métricas.

$$\boldsymbol{F1 = 2 * \frac{Precision * Recall}{Precision + Recall}}$$

2.2.11. Peritaje informático

Consiste en la aplicación de técnicas científicas y analíticas especializadas a una infraestructura tecnológica que permite identificar, preservar, analizar y presentar datos que sean válidos dentro de un proceso legal (Santos Tello, 2013).

Es posible distinguir diferentes o distintos tipos de peritaje informático en base a la función llevada a cabo en ello en nuestra investigación (Ribalda, 2012).

2.2.11.1. Peritaje forense informático

La función es obtener evidencias de la información digital encontrada en los dispositivos informáticos y confeccionar los medios de prueba correspondiente, tanto si se presentan ante el juez o no. Este tipo de peritaje informático que es el objeto principal de esta investigación.

2.2.11.2. Peritaje tecnológico de gestión,

Tiene la función es la obtención de evidencias relacionadas con el cumplimiento de las responsabilidades contractuales asumidas por las partes en cuanto a niveles de calidad o niveles de servicio. Su ámbito de actuación es por ello, la gestión y explotación de proyectos, de servicios o de colaboraciones empresariales, la consultoría y la auditoría informática, etc.

2.2.11.3. Peritaje tecnológico de mediación

Su función es la resolución de modo amistoso entre dos o más partes, de controversias relacionadas con las Tecnologías de la Información y de las Comunicaciones (TIC's) u otras tecnologías.

2.2.11.4. Tasación tecnológica

Tiene la función que es la valoración de bienes tangibles o intangibles de tipo informático en base a normativas y procedimientos establecidos.

- El alcance de las pericias informáticas va mucho más allá de los delitos informáticos exclusivamente, es decir, no siempre que la informática forma parte de un asunto judicial es con motivo de un delito. La informática puede ser protagonista tanto cuando es utilizada como medio para cometer un delito (por ejemplo, una estafa); como cuando es el objeto propio del delito (por ejemplo, la compra de software ilegal); cuando tiene lugar en el conflicto de forma colateral, (por ejemplo, los incumplimientos de contratos de programación y desarrollo). La Pericia Informática, se presenta como una ciencia auxiliar para el mundo legal, brindando apoyo técnico al juez, que permita iluminar sobre los oscuros puntos, que por su especialidad no alcanza a interpretar.

  Para este caso de la investigación, se ha considerado el acto ilícito la falsificación, adulteración o mixtificación de la imagen digital, donde se define que la falsificación es un acto consistente en la creación o modificación de ciertos documentos e imágenes, efectos, productos (bienes o servicios), con el fin de hacerlos parecer como verdaderos, o para alterar o simular la verdad (Mundo Contact, 2010).

- La Prueba Digital. Se tiende como prueba digital o evidencia digital toda información digital empleada por las partes afirmar la realidad de un hecho durante el proceso judicial. Hay que tener en cuenta, que la información que se genera y transmite en el ámbito digital lo hace de dos formas; una que no es inteligible para las personas (la forma y escritura en la que se crea, viaja y almacena la información) y otra que es una representación visual que el ser humano puede entender e interpretar. Esta distinción es importante, porque en muchas ocasiones, la validez de la prueba digital se encuentra en poder acreditar su autenticidad y eso depende de los elementos no visibles que contiene y que requieren de otros medios para poder constatarlos (Datos, 2021)

  Los problemas asociados a la prueba digital, aunque poco a poco va aumentando la jurisprudencia sobre la prueba digital y su admisión como medio de prueba en los procedimientos judiciales, lo cierto es que la validez y la eficacia procesal de la prueba electrónica lleva asociados una serie de problemas sobre la garantía de su carga probatoria y, en muchas ocasiones, pueden hacer que la prueba digital sea desestimada incluso antes de llegar al juicio oral (Datos, 2021)

- **Documentabilidad e Intangibilidad**. Por las propias características que se mencionó sobre la prueba digital, transcribirla a formato papel puede hacer que pierda toda su validez, porque, salvo que quien lo haga tengan los conocimientos técnicos necesarios, se perderá información valiosa de la prueba (como metadatos, encabezados, fechas, etc.).

- **Integridad**. La prueba digital puede ser manipulada y modificada, pensemos en esas conversaciones de WhatsApp falsificadas o en esos mensajes de Twitter supuestamente expresados por una persona concreta, pero que en realidad son falsos. Actualmente, no es muy difícil encontrar tutorial en internet que enseñen a falsificar este tipo de pruebas, por lo que demostrar la integridad de la prueba, que esta no ha sido modificada de ninguna manera, es fundamental para que dicha prueba sea admitida finalmente.

- **Autenticidad**. Las pruebas digitales también presentan problemas que pueden llevar a dudar de su autenticidad. Ya no solo hablamos de modificar una prueba, sino de fabricarla completamente; y el ámbito digital permite ese tipo de creaciones.

- **Otros inconvenientes para dotarlas de valor probatorio**. Las pruebas digitales tampoco pueden ser verificadas por un fedatario público, es decir, por un notario, puesto que lo que este estaría verificando sería que entre la copia de la prueba y el contenido de la misma en su soporte digital no hay diferencias, pero no en sí su autenticidad o falta de manipulación, pues el notario no lleva a cabo el procedimiento necesario para comprobarlo ni tiene los conocimientos técnicos para hacerlo.

2.3. Definición de términos

- **Algoritmo:** Un conjunto ordenado de pasos que toman como entrada ciertos datos y producen como salida un resultado útil (T. H. et al., 2009).

- **Algoritmo Hash**: Un algoritmo hash es un algoritmo que toma un mensaje de cualquier longitud como entrada y devuelve un valor hash fijo de longitud (Stallings, 2017) (p. 36).

- **Algoritmo de Partición Aleatoria**: Proceso para dividir un conjunto de datos en un conjunto de entrenamiento y un conjunto de prueba de forma aleatoria (Gareth et al., 2017) (p. 43).

- **Autenticación de Imágenes**: Proceso de verificar la integridad y autenticidad de una imagen mediante la comparación con una versión original o la utilización de técnicas de análisis de imágenes (Mushtaq & Mir, 2014).

- **Digitalización**: Proceso de convertir una imagen analógica en una imagen digital mediante la asignación de un valor numérico a cada punto de la imagen (Jain, 1989) (p. 3).

- **Función Hash**: Algoritmo matemático que toma una entrada de longitud variable y devuelve un valor numérico fijo de longitud, conocido como hash o resumen (Stallings, 2017) (p. 175).
- **Hash**: Valor numérico fijo de longitud generado por una función hash, utilizando para verificar la integridad de los datos y para crear estructuras de datos eficientes como tablas hash (Stallings, 2017) (p. 175).
- **Imagen Digital**: Representación numérica de una imagen, donde cada punto de la imagen se representa mediante uno o más valores numéricos que representan el nivel de brillo y color en ese punto (Gonzalez & Woods, 2008) (p. 1).
- **Imagen Hash**: Valor numérico generado por una función hash aplicada a una imagen digital, utilizado para verificar la integridad de la imagen y para encontrar similitudes entre imágenes (Lukas et al., 2006).
- **Integridad**: Es la propiedad de un sistema o conjunto de datos de mantener su coherencia y estabilidad a lo largo del tiempo, sin sufrir cambios no autorizados o accidentales (Pfleeger, 2015) (p. 541).
- **Metadato**: Información que describe o proporciona información sobre otros datos. Pueden incluir información como el nombre del archivo, la fecha de creación, el autor, el formato del archivo, entre otros (Lagoze et al., 2002, p. 1).
- **Normalización de Imágenes**: Proceso de ajustar el contraste y el brillo de una imagen para que cumpla con ciertos estándares o para mejorar su visualización (Gonzalez & Woods, 2008) (p. 76).
- **Pixel**: Unidad básica de información en una imagen digital, es un punto de color en una imagen y su valor de color se almacena en una matriz de puntos (Gonzalez & Woods, 2008) (p. 76).
- **Transformada de Affine**: Combinación lineal de traslación, rotación, escala y distorsión (Gonzalez & Woods, 2008) (p. 555).

III. METODOLOGÍA

3.1. Tipo y nivel de investigación

3.1.1. Tipo de Investigación

Para (Arias & Covinos, 2021) la investigación de tipo Aplicada, consiste en transformar el conocimiento existente para resolver problemas existentes, el resultado de la investigación muestra una contribución en la comunidad científica enriqueciendo la aplicación práctica de teorías en un campo determinado por el investigador.

La Siguiente investigación es de tipo Aplicada, porque se centra en la aplicación práctica del algoritmo hash de percepción para abordar un problema específico como la detección de imágenes mixtificadas en el entorno de la pericia de la imagen digital, siendo un problema práctico con implicaciones en el campo de la pericia de imágenes digitales.

3.1.2. Diseño de Investigación

Consiste en la aplicación de técnicas científicas y analíticas especializadas a una infraestructura tecnológica que permite identificar, preservar, analizar y presentar datos que sean válidos dentro de un proceso legal (Santos Tello, 2013).

Los diseños cuasiexperimentales implican la manipulación intencional de al menos una variable independiente para examinar su impacto en una o más variables dependientes. Además, estos diseños utilizan grupos preexistentes que no son asignados aleatoriamente, sino que permanecen intactos antes del experimento.

G1 X1 O1

G2 X2 O2

G3 X3 O3

(Hernandez-Sampieri et al., 2014)

3.1.3. Alcance de la Investigación

La investigación tiene un alcance descriptivo. Para el estudio descriptivo se centra en las propiedades intrínsecas del algoritmo hash de percepción, por el motivo de que se especificarán características propias del algoritmo para originar

perfiles del objeto de estudio, llegando a responder el comportamiento de las variables (Hernandez-Sampieri et al., 2014).

3.2. Ámbito temporal y espacial

3.2.1. Ámbito temporal

El ámbito temporal para la investigación se desarrollará en un periodo de 11 meses, contemplo inicialmente desde la notificación de resolución del proyecto de investigación y en caso de ser financiada, desde la recepción del primer requerimiento.

Los datos, el desarrollo y los resultados de la investigación se efectuó en un espacio de 11 meses.

- Fecha de inicio: 01.01.2021
- Fecha de término: 30.11.2021

3.2.2. Ámbito espacial

Para la presente investigación se hace referencia al lugar o espacio geográfico en el que se realizará el estudio. La investigación se efectuó en el siguiente espacio geográfico:

- País: Perú
- Institución: Policía Nacional del Perú
- Área específica: Área Criminalista
- Departamento: Lima
- Distrito: Surquillo

3.3. Población y muestra

3.3.1. Población

Para (Arias & Covinos, 2021) define como la población o en términos más precisos población objetivo, es un conjunto finito o infinito de elementos con características comunes para los cuales serán extensivas las conclusiones de la investigación. Esta queda delimitada por el problema y por los objetivos del estudio.

La población objetivo de esta investigación se compone de un total de 4,800 imágenes. De estas, 300 imágenes fueron obtenidas directamente de redes sociales públicas de la (Dirección de Criminalística PNP). Las restantes 4,500 imágenes fueron generadas a partir de las originales mediante un proceso de mixtificación, que incluyó técnicas como desenfoque gaussiano, compresión JPEG y rotación. Este enfoque permite comparar los hashes de percepción entre las imágenes originales y las mixtificadas, contribuyendo así al análisis de la percepción visual en el contexto de la investigación.

3.3.2. Muestra

Según (Tamayo, 2004) define la muestra como el conjunto de operaciones que se realizan para estudiar la distribución de determinados caracteres en totalidad de una población universo o colectivo, participando de la observación de una fracción de la población considerada.

En esta investigación, se utilizará toda la población para maximizar la precisión y la fiabilidad de los resultados. Al estudiar todas las imágenes disponibles en lugar de seleccionar una muestra aleatoria, se eliminan posibles sesgos de muestra y se garantiza que las conclusiones estén respaldadas por un conjunto completo de datos. Esto proporcionará una comprensión más completa y precisa, ya que se analizan todos los elementos relevantes sin dejar ningún segmento de la población sin examinar.

3.4. Técnicas e Instrumentos para la Recolección de Datos

De acuerdo a Orellana & Huamán (1999) manifiesta que las técnicas de investigación son procedimientos operativos o mecanismos que permiten recoger información necesaria de la muestra determinada.

- Recolección de imágenes: Esta técnica implica la obtención de imágenes de redes sociales públicas para su posterior análisis. imágenes obtenidas de fanpage publica “Dirección de Criminalística PNP, Surquillo”
- Procesamiento de imágenes: Se utiliza para realizar el tratamiento de las imágenes descargadas, incluyendo la extracción de características visuales y la generación de los hashes de percepción.

- Medición con distancia de Hamming: Una técnica para calcular la similitud entre los hashes de percepción de diferentes imágenes.

3.4.1. Selección de Instrumentos

- Software de descarga de imágenes
- Herramienta de procesamiento de imágenes (opencv - python)
- Algoritmos de hash de percepción hash diferencial (d_hash), el algoritmo hash DCT (p_hash), el algoritmo hash promedio (a_hash).
- Algoritmo de distancia de Hamming

3.4.2. Validación y Confiabilidad del Instrumento

Los instrumentos de medición para la detección de imágenes mixtificados están validados mediante las manipulaciones que conservan el contenido y configuración de parámetros de imágenes de ataque para el análisis y comparación de robustez, que se encuentran consideradas en la investigación "Perceptual hash – based coarse – to – fine grained image tampering forensics method" (X. Wang et al., 2021).

Tabla 3: *Configuración de parámetros para la mixtificación de imágenes.*

Tipo de alteración	Configuración de parámetros				
desenfoque gaussiano	2	4	6	8	10
Comprensión JPEG	20	40	60	80	90
Imagen con rotación (ángulo)	5	10	15	20	25

Fuente: Modificado por (X. Wang et al., 2021)

3.5. Procedimientos

Los procedimientos a seguir para el análisis de los hashes están alineados al procesamiento de imágenes. Dado que el análisis de hashes implica la manipulación y comparación de valores hash generados a partir de imágenes, se requiere un procesamiento de imágenes para llevar a cabo estas operaciones de manera efectiva.

- Adquisición de datos
- procesamiento de imágenes (Mixtificación)

- implementación de los algoritmos de hash de percepción
- extracción de características
- Cálculo de la distancia de Hamming
- Evaluación de algoritmos hash en la detección de imágenes Mistificados. (Análisis Comparativo)

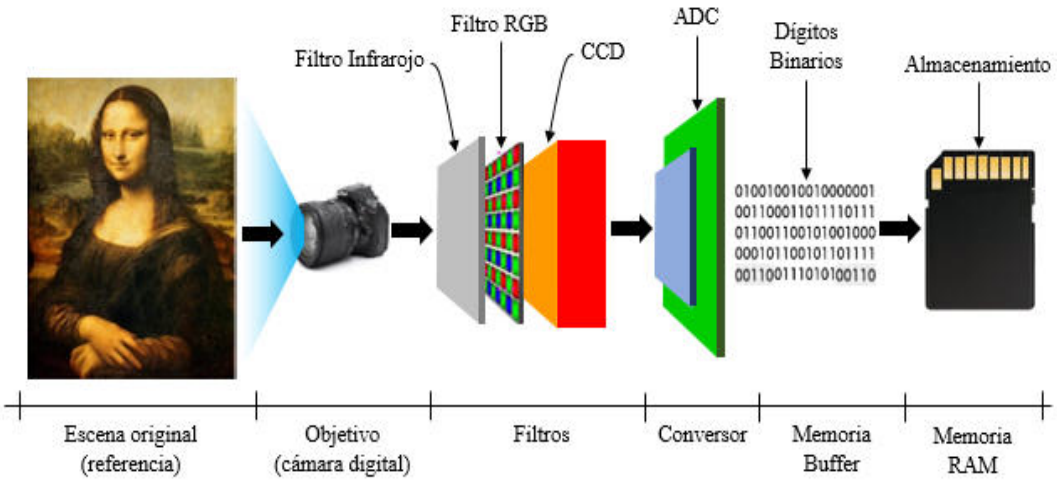
3.5.1. Análisis de datos

En este apartado se describirá para cubrir la metodología, origen de la data, asimismo se describe la adquisición de la imagen, procesamiento de imágenes (desenfoque gaussiano, compresión de imagen, rotación de imagen), extracción de características, calcula de la distancia de Hamming.

3.5.2. Adquisición de datos

Cuando hablamos de la adquisición de datos, nos referimos al proceso de realizar capturas de las imágenes y registrarlas de alguna manera para analizarlo antes de realizar un tratamiento digital de la imagen, esto implica, que el método que se utilizó va a depender de la naturaleza del medio tecnológico.

Ilustración 16: *Captura de la imagen por una cámara digital*



Fuente: Elaboración propia

3.5.3. Procesamiento de imágenes

Conjunto de operaciones y actividades realizadas sobre datos para transformarlos, organizarlos, analizarlos y obtener información útil y significativa. Este proceso implica la manipulación de datos de entrada mediante diferentes técnicas, algoritmos y herramientas. Como desenfoque Gaussiano, compresión JPEG, rotación.

3.5.3.1.Desenfoque Gaussiano

Esta técnica funciona mediante el uso de una función gaussiana y un kernel que es una matriz rectangular de números que sigue una distribución gaussiana. Cada píxel de la imagen se procesa de forma independiente, y su valor se modifica en función de su propio valor y su entorno, utilizando el kernel para generar un nuevo valor de píxel de salida basado en un promedio ponderado de los píxeles vecinos según la distribución gaussiana, tal como se muestra en la Ilustración 17 el código de la técnica del desenfoque.

Ilustración 17: *Código del desenfoque de una imagen*

```
def f_desenfoque(path, t_kernel, str_name):

    contenido = os.listdir(path)

    for i in contenido:
        img = cv2.imread(path + '/' + i)
        # generando el kernel
        kernel_motion_blur = np.zeros((t_kernel, t_kernel))
        kernel_motion_blur[int((t_kernel-1)/2), :] = np.ones(t_kernel)
        kernel_motion_blur = kernel_motion_blur / t_kernel

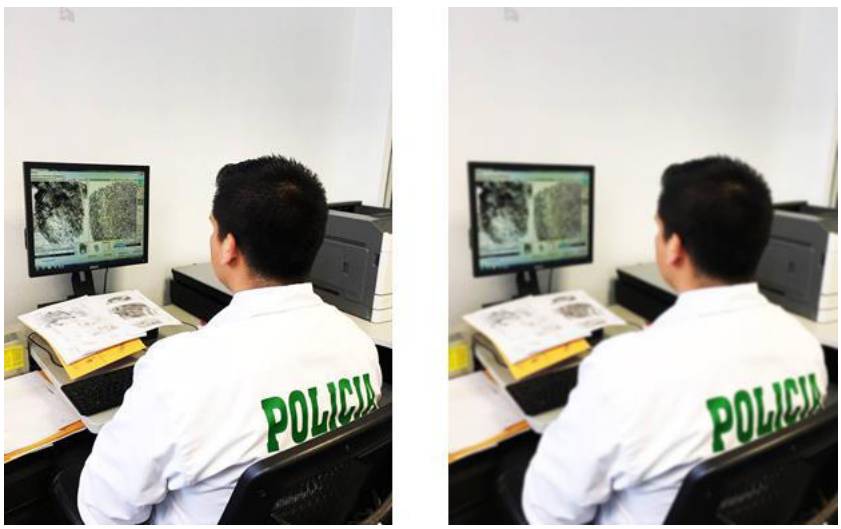
        # aplicando el kernel a la imagen de entrada
        salida = cv2.filter2D(img, -1, kernel_motion_blur)

        cv2.imwrite(os.path.join(path, '{}_{}.jpg'.format(i[:-4], str_name)), salida)
```

Fuente: Elaboración propia

En la Ilustración 18 se observa el resultado obtenido por la técnica del desenfoque gaussiano de una imagen.

Ilustración 18: *Imagen original e imagen con desenfoque*



Fuente: Elaboración propia

3.5.3.2. Compresión JPEG

Esta técnica se utiliza ampliamente con la generación de miniaturas. Cuando almacenamos una imagen en formato JPEG podemos establecer la calidad en un valor que varía de 0 a 100, tal como se muestra en la Ilustración 19 el código de la técnica de la compresión de la imagen JPEG.

Ilustración 19: *Código de la compresión JPEG de una imagen*

```
def f_compresion(path, path1, factor_compresion):

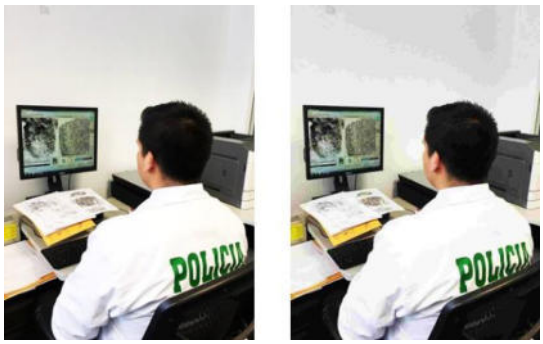
    contenido = os.listdir(path)

    for i in contenido:
        img = cv2.imread(path + '/' + i)
        # compresion
        cv2.imwrite(os.path.join(path1, '{}'.format(i)),img,[cv2.IMWRITE_JPEG_QUALITY, factor_compresion])
```

Fuente: Elaboración propia

En la Ilustración 20 se observa el resultado obtenido por la técnica de la compresión JPEG de una imagen.

Ilustración 20: *Imagen original e imagen con compresión 10*



Fuente: Elaboración propia

3.5.3.3. Rotación de imágenes

Es una variable de las distorsiones geométricas que consiste en girar la imagen en un ángulo determinado, tal como se muestra en la Ilustración 21 el código de la técnica de la imagen con rotación.

Ilustración 21: *Código del ángulo de rotación de una imagen*

```python
def f_girador(path, path_s, angulo):

    contenido = os.listdir(path)

    for i in contenido:
        img = cv2.imread(path + '/' + i)
        # generando l variables
        ancho = img.shape[1]
        alto = img.shape[0]  # filas

        # rotacion
        M =cv2.getRotationMatrix2D((ancho//2, alto//2), angulo, 1)
        img_rt = cv2.warpAffine(img, M, (ancho, alto))

        cv2.imwrite(os.path.join(path_s, '{}.jpg'.format(i[:-4])), img_rt)
```

Fuente: Elaboración propia

En la Ilustración 22 se observa el resultado obtenido por la técnica de la imagen con rotación; el ángulo es un factor para resaltar un aspecto en la imagen.

Ilustración 22: *Imagen original e imagen con giro 5°*



Fuente: Elaboración propia

3.5.4. Implementación de los algoritmos de hash de percepción.

Desarrollar y configurar los algoritmos de hash de percepción para la detección de imágenes mixtificadas,

Se desarrolla los siguientes algoritmos de hash de percepción: Algoritmo de Hash promedio (a_hash), Algoritmo de Hash diferencial (d_hash), Algoritmo de DCT – Hash (p_hash)

Ilustración 23: *Código de hash promedio.*

```python
def average_hash(image_path):
    # Cargar la imagen usando cv2
    image = cv2.imread(image_path)

    # Redimensionar la imagen a un tamaño fijo
    resized_image = cv2.resize(image, (16, 16))

    # Convertir la imagen a escala de grises
    gray_image = cv2.cvtColor(resized_image, cv2.COLOR_BGR2GRAY)

    # Calcular el valor promedio de los píxeles de la imagen
    average_value = gray_image.mean()

    # Generar el hash utilizando los píxeles mayores o menores que el valor promedio
    hash_value = []
    for i in range(16):
        for j in range(16):
            if gray_image[i, j] > average_value:
                hash_value.append(1)
            else:
                hash_value.append(0)
    array_np = np.array(hash_value)
    return array_np.T
```

Fuente: Elaboración propia

Ilustración 24: *Código de hash diferencial (d_hash).*

```python
def dhash_cv2(path, hash_size = 16):
    # leemos la imagen
    img = cv2.imread(path)
    # Convertir a escala de grises
    imz = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    #cv2.cvtColor()
    # escalamos
    px = cv2.resize(imz, (hash_size+1,hash_size), interpolation=cv2.INTER_AREA)

    # restamos los dos bloques
    diff = (px[:,:-1] > px[:,1:])

    ##########################################

    # Convierte la lista en un array de numpy
    arr = np.array(diff)

    # Convierte los valores booleanos en enteros (0 o 1)
    arr_int = np.vectorize(int)(arr)
    flatten_arr = arr_int.flatten()
    return flatten_arr
```

Fuente: Elaboración propia

Ilustración 25: *Código hash DCT (p_hash).*

```python
#image_names = sorted(glob.glob(r'C:\Users\MSI-NB\Desktop\models\Empalme_2x2\*.jpg'))

def dct_hash(image_path, hahs_size = 16):
    # leer todos los nombres de las imágenes en el directorio de la base de datos

    # para cada imagen calcular el valor PHash
    for name in image_names:
        img = cv2.imread(image_path)
        # cambiar el tamaño de la imagen y convertirla a escala de grises
        img = cv2.resize(img, (64, 64))
        img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        img = np.array(img, dtype = np.float32)
        # calcular dct de la imagen
        dct = cv2.dct(img)
        # para reducir la longitud del hash, tome solo 8*8 bloques arriba a la izquierda
        # ya que este bloque tiene más información que el resto
        dct_block = dct[: hahs_size, : hahs_size]
        # calcular la media del bloque dct excluyendo el primer término, es decir, dct(0, 0)
        dct_average = (dct_block.mean() * dct_block.size - dct_block[0, 0]) / (dct_block.size - 1)
        # convertir bloque dct a valores binarios basados en dct_average
        dct_block[dct_block < dct_average] = 0.0
        dct_block[dct_block != 0] = 1.0

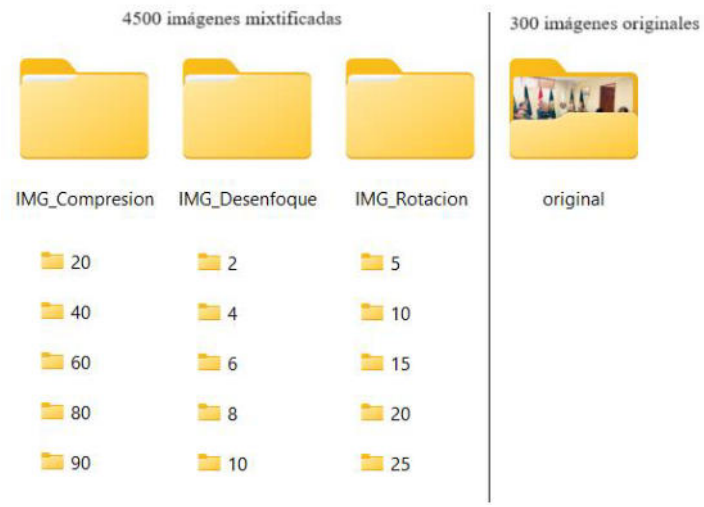
    return dct_block #dct.flatten()
```

Fuente: Elaboración propia

3.5.5. Descripción del Datasets

El datasets contiene 300 imágenes originales, y 4500 imágenes mixtificadas con las técnicas y algoritmos de desenfoque Gaussiano, compresión JPEG, rotación. De acuerdo con la configuración de parámetros para la Mixtificación de imágenes que conservan el contenido, en la siguiente ilustración se muestra en el siguiente ordenamiento de las carpetas con las imágenes mixtificadas.

Ilustración 26: *organización de carpetas con imágenes mixtificadas y original.*



Fuente: Elaboración propia

3.5.6. Extracción de características y generación de Hash

Se emplea los diferentes algoritmos hash de percepción como el algoritmo hash diferencial (d_hash), algoritmo hash promedio (a_hash), algoritmo hash DCT (p_hash) y algoritmo hash basado en VGG – 16 para extraer las características y convertir en un vector hash.

***Ilustración 187**: generación de hash de imagen Original con algoritmo Hash DCT.*

```
In [19]:
path = r'C:\Users\MSI-NB\Desktop\Fotos/42213363_2717336931629317_2456791430127943680_n.jpg'

dct_hash_0 = hash_DCT(path, 16)
dct_hash_0

Out[19]: array([1., 0., 0., 0., 1., 0., 0., 1., 0., 1., 1., 0., 1., 0., 1., 1., 1.,
                1., 1., 0., 1., 0., 1., 1., 0., 1., 0., 0., 0., 0., 1., 1., 0.,
                1., 1., 1., 0., 0., 1., 1., 0., 0., 1., 0., 1., 1., 0., 0., 1., 0.,
                0., 0., 1., 1., 0., 1., 0., 0., 1., 0., 0., 1., 0., 0., 1., 1., 1.,
                1., 0., 0., 0., 1., 0., 1., 1., 0., 1., 1., 1., 1., 0., 0., 1., 1.,
                1., 0., 0., 1., 1., 1., 0., 0., 1., 0., 0., 1., 1., 0., 0., 0., 0.,
                1., 1., 0., 0., 0., 1., 1., 1., 0., 0., 0., 1., 1., 0., 1., 1., 0.,
                1., 1., 1., 0., 0., 1., 0., 1., 0., 0., 0., 1., 1., 0., 1., 0., 0.,
                0., 0., 1., 1., 0., 0., 1., 1., 0., 0., 0., 0., 1., 0., 0., 0., 1.,
                1., 0., 0., 0., 1., 1., 1., 0., 1., 0., 0., 1., 0., 1., 0., 0., 0.,
                0., 0., 0., 1., 0., 0., 1., 0., 0., 0., 0., 1., 1., 1., 0., 0., 0.,
                1., 1., 0., 0., 0., 1., 1., 1., 1., 1., 0., 1., 1., 0., 0., 1., 0.,
                0., 0., 1., 1., 0., 1., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0.,
                1., 0., 1., 1., 0., 0., 0., 1., 1., 0., 0., 0., 1., 0., 0., 1., 1.,
                1., 0., 0., 1., 0., 0., 1., 0., 1., 1., 0., 0., 1., 0., 0., 0., 0.,
                1.], dtype=float32)
```

Fuente: Elaboración propia

***Ilustración 28:** Generación de hash de la imagen Mixtificada con algoritmo hash DCT.*

```
In [13]:
path = r'C:\Users\MSI-NB\Desktop\models\JUNIO/con desenfoque.jpg'
dct_hash = hash_DCT(path, 16)
dct_hash

Out[13]: array([1., 1., 0., 0., 1., 1., 0., 1., 1., 1., 1., 0., 1., 0., 1., 1., 1.,
                0., 1., 0., 1., 0., 0., 0., 0., 1., 1., 1., 1., 0., 1., 0., 1., 1.,
                0., 0., 0., 0., 0., 1., 1., 0., 0., 0., 0., 0., 0., 0., 1., 0., 1.,
                1., 0., 0., 1., 0., 0., 0., 1., 1., 0., 1., 0., 0., 0., 1., 0., 0.,
                0., 0., 1., 0., 1., 0., 1., 0., 1., 0., 1., 0., 0., 1., 0., 0., 1.,
                1., 0., 1., 0., 1., 0., 0., 1., 1., 1., 1., 1., 0., 1., 0., 0., 1.,
                0., 0., 1., 1., 0., 1., 1., 0., 0., 1., 1., 1., 1., 1., 0., 0., 1.,
                1., 0., 0., 1., 0., 0., 1., 1., 0., 0., 1., 1., 1., 1., 1., 1., 0.,
                0., 0., 1., 1., 1., 0., 0., 0., 1., 1., 0., 0., 0., 0., 0., 1., 1.,
                1., 0., 1., 0., 1., 1., 0., 1., 1., 1., 1., 0., 0., 1., 0., 1., 1.,
                0., 0., 1., 1., 1., 0., 0., 0., 1., 0., 1., 1., 1., 0., 0., 0., 1.,
                1., 0., 0., 0., 0., 0., 0., 0., 1., 0., 0., 1., 1., 1., 0., 0.,
                1., 1., 1., 1., 1., 1., 0., 1., 0., 0., 1., 0., 0., 1., 0., 1.,
                0., 1., 0., 1., 0., 0., 1., 0., 1., 1., 0., 0., 0., 1., 0., 0., 0.,
                0., 1., 1., 1., 0., 1., 0., 0., 1., 1., 0., 0., 0., 1., 0., 1., 0.,
                0.], dtype=float32)

In [14]:
dct_hash.shape

Out[14]: (256,)
```

Fuente: Elaboración propia

Ilustración 29: **Calculando Distancia de Hamming entre el hash de la imagen original y el hash de la imagen mixtificado**

```
In [22]:  1  # Calcular la Distancia de Hamming
          2  hamming(dct_hash_0, dct_hash)

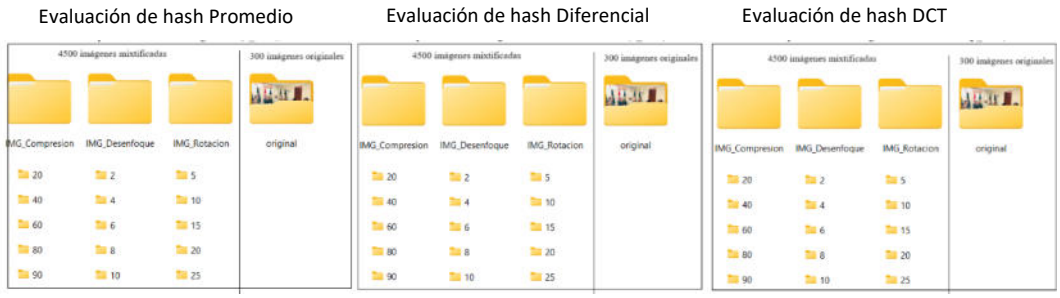
Out[22]: 0.48828125
```

Fuente: Elaboración propia

3.5.7. Evaluación de algoritmos hash (Análisis Comparativo)

La evaluación de algoritmos hash de percepción consiste en la extracción de características, generación de hashes, comparación de hashes y cálculo de la distancia de Hamming de las imágenes originales y su respectiva imagen mixtificadas, se realizan de acuerdo a la siguiente ilustración

Ilustración 30: *Carpetas organizadas para la evaluación de los algoritmos hash*



Fuente: Elaboración propia

Extracción de características. Para ello se programa un extractor de características por tipo algoritmo de hash de percepción.

Ilustración 31: *función para extraer características de imágenes y generación de hash.*

```
In [13]:  1  # funcion de extracion de carateristicas y hash
          2  def extractor_y_hash(path, nombre):
          3
          4      contenido = os.listdir(path)
          5      total = []
          6      nombres = []
          7      lista1 = []
          8      for i in contenido:
          9          nombres.append(i)
         10          hash_a = average_hash(''.join([path, '\{}'.format(i)]))
         11
         12          total.append(hash_a)
         13          data = {'Nombres': nombres}
         14          dfa = pd.DataFrame(data)
         15          dfa[nombre] = total
         16
         17      return dfa
```

Fuente: Elaboración propia

Ilustración 32: *generación de hashes con hash promedio.*

```
In [19]:
path = r'C:\Users\MSI-NB\Desktop\Fotos\original'
inicio = time.time()
hashes_originales = extractor_y_hash(path, 'hash_original')
fin = time.time()
print("tiempo de ejecucion es: ", fin - inicio)

# Especifica el nombre del archivo de Excel
nombre_archivo = 'hashes_originales_promedio.xlsx'

# Guarda el DataFrame en el archivo de Excel
hashes_originales.to_excel(nombre_archivo, index=False)

hashes_originales
```

```
tiempo de ejecucion es:  11.60541033744812
```

Out[19]:

	Nombres	hash_original
0	42213363_2717336931629317_2456791430127943680_...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
1	42835837_2732603083436035_6470595336409710592_...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
2	42855214_2732318000131210_4156190769452941312_...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
3	42861645_2732317956797881_2652767549439082496_...	[0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
4	42878879_2732317930131217_456084101875630080_n...	[1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, ...
5	42886923_2732318030131207_1654044342945316864_...	[1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, ...
6	42896591_2732472833449060_645551949868957696_n...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, ...
7	42903831_2732318083464535_415617989459574784_n...	[1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
8	42920479_2732603016769375_2610007460630495232_...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
9	42928838_2732318150131195_6380614229665251328_...	[1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...

Fuente: Elaboración propia

Ilustración 33: *cálculo de distancias de Hamming entre el hash_original y hash_compresion_20.*

```
In [35]:
path1 = r'C:\Users\MSI-NB\Desktop\Fotos\IMG_Compresion\20'
dataframe_distancia, tiempo, tasa_de_deteccion = distancia_hamming(hashes_originales, path1, 'hash_compresion_20')
tiempo, tasa_de_deteccion
```

```
Out[35]: (6.231516122817993, 275)
```

```
In [22]:
dataframe_distancia
```

Out[22]:

	Nombres	hash_original	hash_compresion_20	distancia_hamming
0	42213363_2717336931629317_2456791430127943680_...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	0.011719
1	42835837_2732603083436035_6470595336409710592_...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	0.000000
2	42855214_2732318000131210_4156190769452941312_...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	0.003906
3	42861645_2732317956797881_2652767549439082496_...	[0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	[0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...	0.011719
4	42878879_2732317930131217_456084101875630080_n...	[1, 1, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, ...	[1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 0, ...	0.019531
5	42886923_2732318030131207_1654044342945316864_...	[1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, ...	[1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, ...	0.007812

Fuente: Elaboración propia

3.5.7.1.Sensibilidad (Recall).

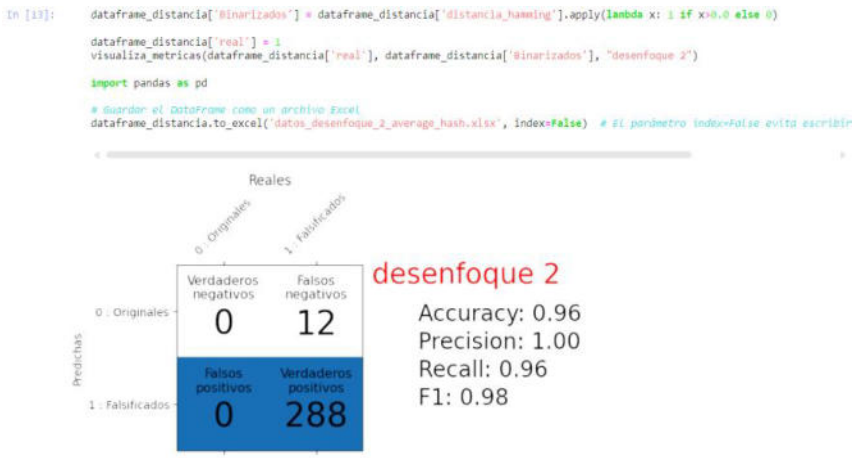
La sensibilidad mide la proporción de casos positivos reales que son correctamente identificados por el modelo o la prueba. Se calcula como la proporción de verdaderos positivos (TP) dividida entre la suma de verdaderos positivos (TP) más falsos negativos (FN).

$$Sensibilidad = \frac{TP}{TP + FN}$$

TP: Verdadero Positivos

FN: Falsos Negativos

Ilustración 34: *muestra los valores obtenidos para imágenes con desenfoque de evaluado con hash promedio.*



Elaboración propia.

La obtención de las métricas se realizará por cada tipo de alteración y su respectiva configuración de parámetros.

3.5.7.2. Umbral.

Para determinar el umbra calculamos la distancia media de las distancias de Hamming por tipo de mixtificación. Este umbral determinara si dos imágenes son similares o no. Por ejemplo, si la distancia de Hamming entre dos hashes es menor que la media de las distancias de Hamming para un tipo específico de manipulación.

IV. RESULTADOS

Tras la recogida de información por medio de la metodología empleada y en base al análisis y el procesamiento de datos se realizó implementación de la presentación de los resultados, que se encuentra en base a los algoritmos de hash de percepción para fundamentar y corroborar las hipótesis planteadas en la presente investigación científica.

Los datos procesados explicarán los resultados obtenidos para comparar y evaluar en forma crítica con datos derivados por otros investigadores; desde la perspectiva del investigador se debe tomar en cuenta los trabajos científicos si corresponde al estudio. Para este contexto, no es más que describir en primera instancia la hipótesis general que exige argumentar como proposición lo que establece la relación entre los hechos, brindando una posible solución del problema detectado.

- Hipótesis General: Se postula que el uso del algoritmo hash de percepción contribuye significativamente al mejoramiento de la detección de imágenes mixtificadas en el contexto de la pericia de la imagen digital.

  Para determinar el aporte en la detección de imágenes mixtificadas, de los algoritmos de hash de percepción se procede a la validación, con la taza de detección de imágenes mixtificados, a continuación de presenta la tabla de resultados.

Tabla 3: *Se puede apreciar la Taza detección de imágenes mixtificadas por los diferentes algoritmos hash de percepción donde resalta el algoritmo hash DCT con los mejores resultados.*

	Desenfoque Gaussiano					Compresión JPEG					Imagen con rotación				
	2	4	6	8	10	20	40	60	80	90	5	10	15	20	25
Algoritmo Hash Promedio	0,96	0,98	0,98	0,99	1	0,96	0,87	0,82	0,68	0,47	1	1	1	1	1
Algoritmo Hash Diferencial	0,32	0,55	0,73	0,81	0,87	0,75	0,64	0,39	0,26	0,16	1	1	1	1	1
Algoritmo Hash DCT	0,97	1	1	1	1	0,96	0,88	0,84	0,68	0,53	1	1	1	1	1

Fuente: Elaboración propia

Se observa en la tabla, una mayor tasa de detección del algoritmo hash DCT frente a sus semejantes, lo que evidencia el aporte en la detección de imágenes mixtificados.

- Hipótesis especifica 1: Se espera que la sensibilidad del algoritmo hash de percepción afecte significativamente su capacidad para detectar manipulaciones ocultas o sutiles, reflejándose en diferencias del Recall.

  Para determinar la sensibilidad (la tasa de detección de imágenes mixtificadas en esta investigación), de los algoritmos de hash de percepción se procede a promediar la taza de detección de imágenes mixtificados por cada tipo de alteración y su respectiva configuración de parámetros. Para determinar el algoritmo con más alta taza de detección.

Tabla 4: *Promedio de taza detección de imágenes mixtificadas por los diferentes algoritmos hash de percepción*

	Desenfoque Gaussiano	Compresión JPEG	Imagen con rotación
Algoritmo Hash Promedio	98,20%	76,00%	100,00%
Algoritmo Hash Diferencial	65,60%	44,00%	100,00%
Algoritmo Hash DCT	99,40%	77,80%	100,00%

Fuente: Elaboración propia

Se observa que el algoritmo Hash DCT es mas sensible ya que obtuvo mayor porcentaje en detección de imágenes mixtificadas por cada tipo de alteración.

Ilustración 35: *Comparación de hashes producidos por los diferentes algoritmos de hash de percepción.*

	Nombres	hash_original	hash_compresion_20	distancia_hamming	Original	Mixtificados
HASH DCT	image_100.jpg	1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 0. 0. 1. 1. 1. 0. 1. 1. 1. 0. 1. 1. 1. 1. 0. 0. 0. 1. 1. 0. 1. 0. 1. 1. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0. 1. 1. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0. 0. 0. 0. 1. 1. 1. 0. 1. 1. 1. 1. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0. 1. 0. 1. 1. 1. 0. 0. 1. 1. 0. 1. 0. 1. 0. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 0. 1. 0. 1. 1. 1. 1. 1. 1. 1. 0. 0. 0. 1. 0. 0. 1. 0. 1. 1. 1. 1. 1. 1. 0. 1. 0. 1. 0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 1. 1. 1. 1. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 0. 1. 1. 1. 1. 0. 0. 0. 0. 0. 1. 0. 1. 1. 1. 1. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 1. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 1. 0. 1. image_100.jpg	0. 1. 1. 1. 1. 1. 1. 0. 0. 1. 0. 1. 0. 0. 0. 0.	1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 0. 0. 1. 1. 1. 0. 1. 1. 1. 0. 1. 0. 0. 0. 0. 0. 1. 1. 0. 1. 0. 1. 1. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 1. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0. 1. 1. 1. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 1. 1. 0. 1. 1. 0. 1. 0. 0. 0. 0. 0. 1. 1. 1. 0. 1. 1. 1. 1. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0. 1. 0. 1. 1. 1. 0. 0. 1. 1. 0. 1. 0. 1. 0. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 0. 1. 0. 1. 1. 1. 1. 1. 1. 1. 0. 0. 0. 1. 0. 0. 1. 0. 1. 1. 1. 1. 1. 1. 0. 1. 0. 1. 0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 1. 1. 1. 1. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 0. 1. 1. 1. 1. 0. 0. 0. 0. 0. 1. 0. 1. 1. 1. 1. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 1. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 1. 1. 1. 1. 1. 1. 0. 0. 1. 0. 1. 0. 0. 0. 0.]	0,03125	1	1
HASH PROMEDIO	image_100.jpg	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0 1 0 0 1 1 1 1 0 1 1 0 0 0 0 0 0 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0]	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0 1 0 0 1 1 1 1 0 1 1 0 0 0 0 0 0 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0]	0,00390625	1	1
HASH DIFERENCIAL	image_100.jpg	[0 1 0 1 1 1 0 0 1 0 1 0 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1 0 0 0 1 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 0 0 1 1 1 0 1 0 0 1 1 0 0 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 0 1 0 0 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 1 0 0 0 1 1 1 1 0 0 1 1 1 0 0 1 1 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1 0 0 0 0 1 0 1 1 1 0 1 1 1 1 0 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 0 1 0 0 0 0 1 0]	[0 1 0 1 1 1 0 0 1 0 1 0 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1 0 0 0 1 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 0 0 1 1 1 0 1 0 0 1 1 0 0 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 0 1 0 0 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 1 0 0 0 1 1 1 1 0 0 1 1 1 0 0 1 1 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1 0 0 0 0 1 0 1 1 1 0 1 1 1 1 0 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 0 1 0 0 0 0 1 0]	0	1	0

	Nombres	hash_original	hash_rotacion_5	distancia_hamming	Original	Mixtificados
HASH DCT	image_100.jpg	[1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 0. 0. 1. 1. 1. 0. 1. 1. 1. 0. 1. 1. 1. 1. 0. 0. 0. 1. 1. 0. 1. 0. 1. 1. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0. 1. 1. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0. 0. 0. 0. 1. 1. 1. 0. 1. 1. 1. 1. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0. 1. 0. 1. 1. 1. 0. 0. 1. 1. 0. 1. 0. 1. 0. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 0. 1. 0. 1. 1. 1. 1. 1. 1. 1. 0. 0. 0. 1. 0. 0. 1. 0. 1. 1. 1. 1. 1. 1. 0. 1. 0. 1. 0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 1. 1. 1. 1. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 0. 1. 1. 1. 1. 0. 0. 0. 0. 0. 1. 0. 1. 1. 1. 1. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 1. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 1. 1. 1. 1. 1. 1. 0. 0. 1. 0. 1. 0. 0. 0. 0.]	[1. 1. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 0. 1. 1. 1. 1. 1. 1. 0. 1. 1. 1. 0. 1. 0. 1. 0. 0. 0. 0. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 0. 1. 0. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 1. 1. 1. 1. 1. 1. 1. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0. 0. 1. 0. 0. 1. 0. 1. 1. 1. 0. 0. 1. 0. 0. 0. 1. 0. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 1. 0. 0. 0. 1. 1. 1. 1. 0. 0. 0. 1. 0. 1. 1. 1. 1. 1. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 1. 1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 1. 0. 0. 0. 1. 1. 1. 1. 0. 0. 0. 1. 0. 1. 1. 1. 1. 1. 0. 0. 0. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 1. 1. 1. 1. 1. 0. 0. 0. 1. 0. 1. 0. 1. 1. 0. 0. 1. 0. 0. 0. 1. 0. 1. 0. 1. 0. 0. 0. 0.]	0,32421875	1	1
HASH PROMEDIO	image_100.jpg	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 0 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 1 0 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0 1 0 0 1 1 1 1 0 1 1 0 0 0 0 0 0 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 1 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0]	[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 1 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0 1 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0]	0,12890625	1	1
HASH DIFERENCIAL	image_100.jpg	[0 1 0 1 1 1 0 0 1 0 1 0 1 0 0 0 0 1 1 1 1 1 1 0 0 0 0 1 0 0 0 1 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1 1 1 0 0 1 1 1 0 1 0 0 1 1 0 0 0 1 1 1 0 1 0 1 0 1 1 1 1 1 1 1 0 1 0 0 0 0 0 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 1 0 0 0 1 1 1 1 0 0 1 1 1 0 0 1 1 0 1 1 1 1 1 1 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 0 1 1 0 0 1 0 0 0 0 1 0 1 1 1 0 1 1 1 1 0 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 0 1 1 1 1 0 0 1 0 0 0 0 1 0]	[0 0 1 0 1 0 0 0 0 1 0 1 0 0 1 1 1 1 1 1 1 1 0 0 1 0 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 1 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 1 1 1 0 1 1 1 1 1 1 1 1 0 1 0 0 1 0 1 1 1 1 1 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 0 0 1 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 1 0 0 0 1 1 1 1 1 0 1 1 1 0 0 1 1 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 1 0 0 1 1 0 0 1 1 0 1 1 0 1 1 0 0 0 0 1 0 1 1 1 0 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 0 0 1 1 1 1 0 1 0 1 0]	0,26953125	1	1

Se observa en la ilustración, los hashes producidos por el algoritmo hash DCT en imagen rotación, permiten la discriminación por que tienen más caracteres diferentes entre el hash original y el hash mixtificado, y se puede reflejar en la distancia de Hamming de 0.32421875. a comparación de su semejante es mayor la distancia de Hamming.

Para la robustez se observa la similitud de los hashes generados por algoritmos hash de percepción que no haya mucha variación, también la robustez se mide cuando la distancia de Hamming no supera el umbral determinado investigación se determinó un umbral de 0,032977431 como se demostrara más adelante.

- Hipótesis especifica 2: La complejidad y variabilidad de las manipulaciones tendrán un impacto directo en la capacidad del algoritmo hash de percepción para detectar imágenes mixtificadas, evidenciándose cambios significativos en su capacidad discriminabilidad.

Para ello se promedia la taza de detección por tipo de alteración, ya que las imágenes mixtificadas con las diferentes técnicas como, Desenfoque Gaussiano, Compresión JPEG, Rotación de imágenes, tienen su grado de complejidad.

Tabla 5: *Promedio de taza detección por tipo de mixtificación.*

	Desenfoque Gaussiano	Compresión JPEG	Imagen con rotación
Algoritmo Hash Promedio	98,20%	76,00%	100,00%
Algoritmo Hash Diferencial	65,60%	44,00%	100,00%
Algoritmo Hash DCT	99,40%	77,80%	100,00%
promedio	87,73%	65,93%	100,00%

Fuente: Elaboración propia

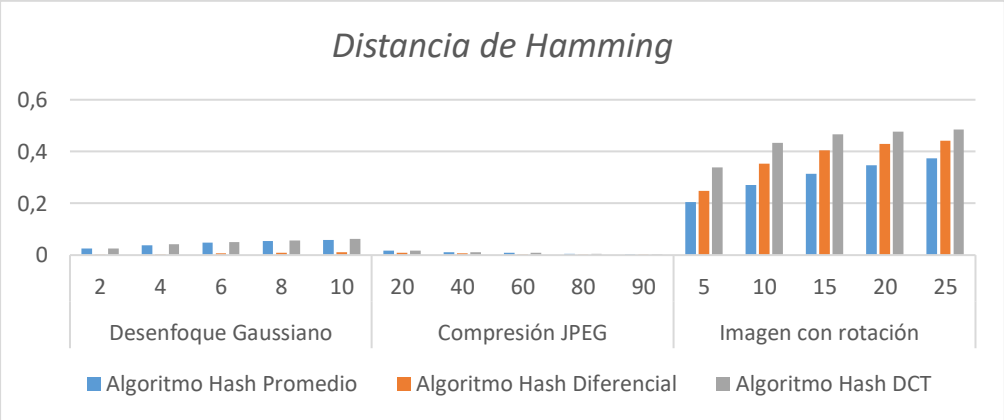
Se observa que los tipos de mixtificación con mayor tasa de detección son ‘imagen con rotación’, seguidamente el desenfoque Gaussiano y por ultimo Comprensión JPEG, indicándonos que las imágenes mixtificadas con las técnicas de compresión JPEG, no son detectados con fácilmente con los diferentes algoritmos de hash de percepción, por lo que concluimos que los

hashes generados para la compresión JPEG, tiene más colisión porque los hashes no son discriminables quiere decir que una imagen y su respectiva imagen mixtificada utilizando el algoritmo JPEG con un nivel de calidad del 90% dará como resultado la misma cadena de Hash.

- Hipótesis especifica 3: Al determinar un umbral de detección óptimo para los hashes generados por algoritmos de hash de percepción garantizará la robustez y mejorará la identificación precisa de similitudes entre imágenes mixtificadas.

  Para ello se grafica las distancias de Hamming

Ilustración 36: *Distancias media de Hamming por tipos de mixtificación.*



Fuente: Elaboración propia

Se puede observar la variación de la distancia de Hamming por tipo de mixtificación. Algo en común que comparten la mixtificación de 'desenfoque Gaussiano' y 'Compresión JPEG' son que su distancia de Hamming es menor 0.1.

Tabla 6: *Tabla de distancias mínimas de Hamming por cada tipo de mixtificación*

	Desenfoque Gaussiano					Compresión JPEG					Imagen con rotación				
	2	4	6	8	10	20	40	60	80	90	5	10	15	20	25
A. Hash Promedio	0,004	0,0039	0,0039	0,0039	0,0039	0,0039	0,0039	0,0039	0,0039	0,0039	0,0195	0,0352	0,0508	0,043	0,0703
A. Hash Diferencial	0,004	0,0039	0,0039	0,0039	0,0039	0,0039	0,0039	0,0039	0,0039	0,0039	0,0508	0,0898	0,1172	0,1328	0,1523
A.Hash DCT	0,004	0,0039	0,0039	0,0039	0,0039	0,0039	0,0039	0,0039	0,0039	0,0039	0,1406	0,2383	0,293	0,3125	0,3281

Fuente: Elaboración propia

Tabla 7: *Tabla de distancias máximas de Hamming por cada tipo de mixtificación*

	Desenfoque Gaussiano					Compresión JPEG					Imagen con rotación				
	2	4	6	8	10	20	40	60	80	90	5	10	15	20	25
A. Hash Promedio	0,14	0,19	0,2	0,22	0,23	0,09	0,06	0,04	0,03	0,02	0,46	0,45	0,51	0,54	0,07
A. Hash Diferencial	0,02	0,03	0.0546875	0,04	0,05	0,08	0,05	0,05	0,04	0,04	0,48	0,51	0,56	0,57	0,56
A. Hash DCT	0,11	0,15	0,15	0,17	0,19	0,07	0,05	0,05	0,03	0,03	0,54	0,54	0,57	0,59	0,57

Fuente: Elaboración propia

Se observa el valor de mínimo de la distancia de haming es igual: 0,00390625 el cual se usó como el umbral mínimo para diferenciar las cadenas de hash. De igual manera el máximo valor de distancia de Hamming es 0.59 que proviene de 'Imagen con rotación' para tener un umbral de generalización sacamos la media de las distancias de Hamming por tipo de mixtificación

Tabla 8: *Tabla de distancias media de Hamming por cada tipo de mixtificación*

	Desenfoque Gaussiano	Compresión JPEG
media	0,032977431	0,007491319

Fuente: Elaboración propia

En esta investigación se considera como referencia la media de desenfoque gaussiano como nuestro umbral 0.032977431.

Propuesta de Hash basado en VGG16 modelo pre entrenado

En esta investigación se propone un algoritmo basado en el modelo pre entrenado VGG16 la cual se usa para extraer las características de la imagen digital, para luego generar la cadena hash.

Ilustración 37: *Código para la extracción de características y generación de cadena de Hash.*

```python
model = VGG16(weights='imagenet', include_top=False)
#model.summary()
def hash_vgg16(path, hahs_size = 256):
    layer = tf.keras.layers.GlobalAveragePooling2D()

    img = keras.utils.load_img(path, target_size=(256, 256))

    img_tensor = keras.utils.img_to_array(img)
    img_tensor = np.expand_dims(img_tensor, axis=0)

    img_data = preprocess_input(img_tensor)

    vgg16_feature = model.predict(img_data)

    ##plt.matshow(vgg16_feature[0, :, :, 4], cmap='viridis')
    H0_1 = layer(vgg16_feature)
    H0_1 = np.array(H0_1).squeeze()

    hash_average = H0_1.mean()
    # Crear un nuevo array con 0 y 1 según la relación con la media
    array_resultado = np.where(H0_1 < hash_average, 0, 1)

    return array_resultado[:hahs_size]
```

Fuente: Elaboración propia

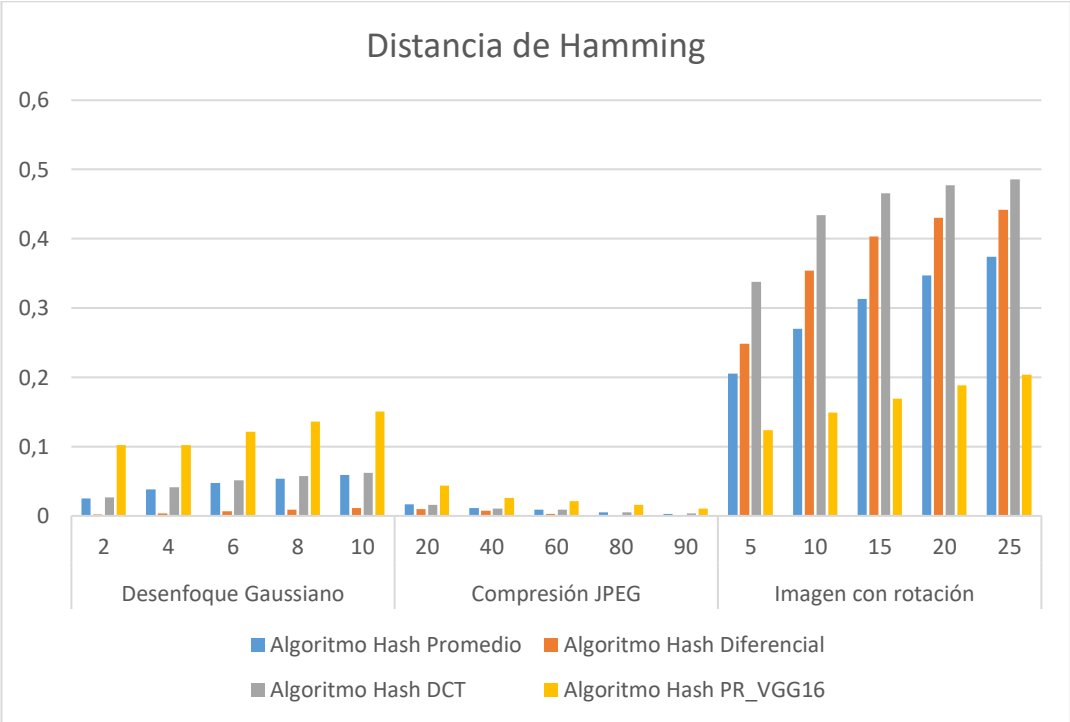
Tabla 9: *Tasa detección de imágenes mixtificadas por los diferentes algoritmos hash de percepción incluida la propuesta de hash basado en Vgg16.*

	Desenfoque Gaussiano	Compresión JPEG	Imagen con rotación
Algoritmo Hash Promedio	98,20%	76,00%	100,00%
Algoritmo Hash Diferencial	65,60%	44,00%	100,00%
Algoritmo Hash DCT	99,40%	77,80%	100,00%
Algoritmo Hash PR_VGG16	100,00%	95,20%	100,00%

Fuente: Elaboración propia

Se observa que la tasa de detección de la propuesta es superior frente a los demás.

Ilustración 38: *Tasa detección de imágenes mixtificadas por los diferentes algoritmos hash de percepción incluida la propuesta de hash basado en Vgg16*



Fuente: Elaboración propia

Se observa las barras del algoritmo propuesto no superan el 0.2 en la escala de distancia de Hamming. Indicador clave de sensibilidad. Lo cual indica su baja sensibilidad con las imágenes con rotación.

V. DISCUSIÓN

En la evaluación del aporte de los algoritmos Hash de percepción en la detección de imágenes mixtificadas se utilizó las medidas como la Taza de Verdaderos Positivos (TPR) según los resultados de la taza de detección se evidencia que el algoritmo de hash de percepción obtiene DCT 89 % de detección de imágenes mixtificadas, seguidamente para el algoritmo hash promedio con 88 % de taza de detección y finalmente hash diferencial con 79%. Estos resultados son menores frente a los resultados de Wang (X. Wang et al., 2021) que logra altas tasas de detección llegando al 100% en la mixtificación de tipo compresión de imagen, Wang es esta investigación propone una combinación de diferentes algoritmos de percepción para la extracción de características como Wavelet Transform (SWT), SIFT, que mejora la robustez de los hashes.

Frente a ello en esta investigación la propuesta de extraer las características de la imagen digital, con Vgg16, dio mejores resultados como lo demuestran la tabla 10, los promedios de tasa de detección son superior frente a los algoritmos de estudio. Otro punto a tomar en cuenta es que el hash basado en Vgg16 es menos sensibles que el hash DCT porque el promedio de las distancias de Hamming apenas llega a 0.2 a comparación del promedio de las distancias de Hamming de los hashes del algoritmo hash DCT. Lo cual demuestra una baja sensibilidad ante las imágenes con rotación.

Ling Du en su investigación En la investigación "Perceptual hashing for image authentication: A surveyse" habla de las funciones hash criptográficas tradicionales y resalta que son muy sensibles a los datos de entrada, donde el cambio de 1 bit puede afectar la salida, lo que genera un efecto de avalancha que da como resultado un hash completamente diferente. Por otro lado afirma que las funciones hash de percepción no son muy sensibles a pequeños cambios en los datos de entrada, pero son susceptibles a las diferencias en las características de percepción del objeto comparado, lo que da una importancia a la sensibilidad que deben tener los algoritmos hash de percepción, en nuestra investigación se demuestra esta sensibilidad en la ilustración 35, donde se observa claramente que los

valores de la distancia de Hamming del algoritmo hash percepción DCT, se incrementan en la mixtificación de rotación de imagen y no ocurre tanto con nuestra propuesta de hash basado en Vgg16 como se observa en la ilustración 37.

Din (Ding et al., 2023) en su investigación, mencionan que se debería de enfocar estudios para desafíos no abordados, como manipulaciones complejas como la rotación y el escalado. en nuestra investigación se demostró que los algoritmos de hash de percepción son muy sensibles a las imágenes con rotación es por ello que crece la distancia de Haming. Por tal razón Wang (X. Wang et al., 2021) busca extraer característica únicos invariables con SIFT de igual manera Din busca características relevantes en las imágenes, que no varían ante la rotación de imagen, del mismo modo en esta investigación se plantea extraer características únicas de la imagen con Vgg16, que no son afectados por la rotación de imagen y diferentes tipos de mixtificación.

Swamy, M. N. S. (2018). Propone encontrar un umbral para mantener en equilibrio satisfactorio entre la robustez y discriminación. En la investigación se pudo observar en las imágenes con rotación, el aumento de la distancia de Hamming lo que indica que las cadenas de hash son muy diferentes cuando se les compara lo que demuestra la baja robustez, aparte de ello la distancias de Hamming de imágenes con rotación superan el umbral propuesto 0.033 en esta investigación para mantener el equilibrio de robustez. Mientras que las imágenes con desenfoque y compresión, mantienen bajo su distancia de Hamming a nivel del umbral, lo que asegura la robustez de los hashes. La propuesta de Swamy es acertado.

VI. CONCLUSIONES

Los algoritmos de hash de percepción analizados, particularmente el basado en DCT, el hash promedio y el hash diferencial, demuestran la capacidad para detectar imágenes mixtificadas, aunque en niveles de eficacia variables. Es importante destacar que la propuesta de extraer características de imágenes digitales utilizando el modelo pre entrenado Vgg16 y posteriormente binarizar para obtener hashes ha sobrepasado significativamente el rendimiento de los algoritmos de hash de percepción evaluados. Los resultados demuestran que esta técnica ha logrado una mayor efectividad en la detección de imágenes mixtificadas.

La sensibilidad de los algoritmos hash de percepción aportan en la detección de imágenes mixtificadas como se pudo demostrar en la tabla 4 con el algoritmo hash DCT con la taza de detección más alta frentes a los demás algoritmos, demostrando que es más sensible frente a los algoritmos evaludos.

La complejidad de las manipulaciones en las imágenes impacta en la detección como se demostró la técnica de mixtificación de tipo compresión JPEG que es difícil de detectar para los algoritmos de hash de percepción, donde se obtuvo más colisión de hash.

En la investigación se determino el umbral con la media de las distancias de Hamming de Desenfoque Gaussiano.

La propuesta de extraer características robustas e invariables ante estas manipulaciones, como la presentada mediante Vgg16 en este estudio, demuestra ser una vía prometedora para mejorar la capacidad de detección de imágenes mixtificadas en entornos de pericia digital. Por otro lado, en el caso de las imágenes con desenfoque y compresión, se mantuvo una distancia de Hamming por debajo del umbral propuesto, lo que confirma la robustez de los hashes en estos escenarios específicos. Este hallazgo respalda la validez de la propuesta de & Swamy, Los resultados confirman que la propuesta de establecer umbrales para mantener un equilibrio entre robustez y discriminación es pertinente, resaltando la necesidad de adaptar estos algoritmos para ser más efectivos en entornos donde las manipulaciones son más complejas o variables.

VII. RECOMENDACIONES

Considerar la implementación de técnicas avanzadas de extracción de características, como Vgg16, para mejorar la precisión y robustez en la detección de imágenes mixtificadas en entornos de pericia digital.

Evaluar y explorar el potencial de nuevos enfoques de algoritmos de hash de percepción que puedan superar la sensibilidad y mejorar la capacidad de detección de manipulaciones visuales, especialmente en situaciones complejas como la rotación de imágenes.

Investigar la combinación de diferentes algoritmos de percepción o la incorporación de características específicas que sean invariables ante manipulaciones, con el objetivo de aumentar la eficacia en la detección de mixtificaciones en diversos contextos y tipos de imágenes.

Continuar el estudio y desarrollo de estrategias que busquen mantener un equilibrio entre la robustez y la discriminación en los algoritmos de hash de percepción, considerando umbrales adecuados para diferentes tipos de manipulaciones visuales.

REFERENCIAS

Anuario Estadístico Policial. (2021). https://web.policia.gob.pe/anuario_estadistico/anuario_policial.html

Arias, J., & Covinos, M. (2021). Diseño y metodología de la investigación. In *Enfoques Consulting EIRL*. https://repositorio.concytec.gob.pe/handle/20.500.12390/2260

Bañuelos Capistrán, J. (2020). Deepfake: la imagen en tiempos de la posverdad. *Revista Panamericana de Comunicación*, *1*, 51–61.

Boletín Estadístico Policial I Trimestre. (2022). https://web.policia.gob.pe/anuario_estadistico/anuario_policial.html

Bravo, A. J. (n.d.). *Procesamiento Digital de Imágenes*. Universidad de los Andes.

Bruno, C., & Luis, J. (2019). *La manipulación de la fotografía utilizada con fines políticos en el Perú* [Universidad Peruana de Ciencias Aplicadas]. https://repositorioacademico.upc.edu.pe/handle/10757/657039

Calvet, S., Lejsek, H., & Terán, J. (1993). *Análisis y comprensión de técnicas para la comprensión de imágenes*. Universidad Mayor de San Andres.

Chen, H., Zhou, H., Zhang, J., Chen, D., Zhang, W., Chen, K., Hua, G., & Yu, N. (2021). Perceptual Hashing of Deep Convolutional Neural Networks for Model Copy Detection. *ACM Transactions on Multimedia Computing, Communications, and Applications*, *19*(3), 1–20. https://doi.org/10.1145/3572777

Cornell. (n.d.). *Tutorial de Digitalización de Imágenes*. https://www.academia.edu/30220448/Tutorial_digitalizacion

Cosenza, D. N., Korhonen, L., Maltamo, M., Packalen, P., Strunk, J. L., Næsset,

E., Gobakken, T., Soares, P., & Tomé, M. (2021). Comparison of linear regression, k-nearest neighbour and random forest methods in airborne laser-scanning-based prediction of growing stock. *Forestry*, *94*(2), 311–323. https://doi.org/10.1093/forestry/cpaa034

Datos, A. L. P. (2021). *El uso y validez de la prueba digital en el procedimiento judicial*. https://ayudaleyprotecciondatos.es/2021/07/22/prueba-digital/

Deng, X., Liu, Q., Deng, Y., & Mahadevan, S. (2016). An improved method to construct basic probability assignment based on the confusion matrix for classification problem. *Information Sciences*, *340–341*, 250–261. https://doi.org/10.1016/j.ins.2016.01.033

Ding, K., Chen, S., Zeng, Y., Wang, Y., & Yan, X. (2023). Transformer-Based Subject-Sensitive Hashing for Integrity Authentication of High-Resolution Remote Sensing (HRRS) Images. *Applied Sciences (Switzerland)*, *13*(3). https://doi.org/10.3390/app13031815

Du, L., Ho, A. T. S., & Cong, R. (2020). Perceptual hashing for image authentication: A survey. *Signal Processing: Image Communication*, *81*, 1. https://doi.org/10.1016/j.image.2019.115713

Elgendy, M. (2020). *Deep Learning for Vision Systems*. https://www.manning.com/books/deep-learning-for-vision-systems

Fernández, E. G., Orozco, A. L. S., Villalba, L. J. G., & Hernandez-Castro, J. (2018). Digital image tamper detection technique based on spectrum analysis of CFA artifacts. *Sensors*, *18*(9), 1–18. https://doi.org/10.3390/s18092804

Gareth, J., Witten, D., Hastie, T., & Tibshirani, R. (2017). An introduction to statical learning. In *Handbook of Quantile Regression*. https://doi.org/10.1201/9781315120256

Giménez Pérez, J. F. (2022). *Detección de imágenes alteradas mediante el uso de un hash perceptual basado en CNNs*.

Gómez-Ros, A., Tabik, S. ;, Luengo, J. ;, Herrera, F. ;, Shihavuddin, A. S. M., & Krawczyk, B. ; (2019). Redes Neuronales Convolucionales para Una Clasificacion Precisa de Imagenes de Corales. *Citation*, 1171–1176.

Gonzáles Aguilera, D. (2005). Preprocesamiento de imágenes. In *Visión por computador, fundamentos y métodos*. Universidad de Salamanca. https://gredos.usal.es/handle/10366/83443

Gonzalez, R. C., & Woods, R. E. (2008). *Digital Image Processing* (3th Edition (Ed.)).

Grammatikopoulos, G., Hatzigaidas, A., Papastergiou, A., Lazaridis, P., Zaharis, Z., Kampitaki, D., & Tryfon, G. (2007). Simple Matlab tool for automated malignant melanoma diagnosis. *WSEAS Transactions on Information Science and Applications*, *4*(3), 460–465.

Haitsma, J., Kalker, T., & Oostveen, J. (2001). Robust Audio Hashing for Content Identification. *Content Based Multimedia Indexing 2001, Brescia, Italy, September 2001*, *4*(Wy 82).

Hamming, R. W. (1950). Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, *29*(2), 147–160. https://doi.org/10.1002/j.1538-7305.1950.tb00463.x

Hernandez-Sampieri, R., Fenández-Collado, C., & Baptista-Lucio, M. del P. (2014). Metodología de la Investigación. In *Antimicrobial agents and*

chemotherapy (Vol. 58, Issue 12). https://www.uca.ac.cr/wpcontent/uploads/2017/10/Investigacion.pdf%0Ahttp://www.ncbi.nlm.nih.gov/pubmed/25246403%0Ahttp://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4249520

Hernández Sampieri, R., Fernández Collado, C., & Baptista Lucio, M. del P. (2014). *Metodología de la Investigación* (6ta edició).

Hidalgo Barrientos, M. F., Hayes Ortiz, B. I., Delgadillo Vera, I., & Goyo Escalona, M. (2021). Deep learning aplicado para la detección de hemorragias y tumores cerebrales. *AtoZ: Novas Práticas Em Informação e Conhecimento*, *10*(3), 1. https://doi.org/10.5380/atoz.v10i3.81284

Hussain, A., Li, H. C., Ali, M., Wali, S., Hussain, M., & Rehman, A. (2022). An Efficient Supervised Deep Hashing Method for Image Retrieval. *Entropy*, *24*(10). https://doi.org/10.3390/e24101425

Iakovidou, C., Zampoglou, M., Papadopoulos, S., & Kompatsiaris, Y. (2018). Content-aware detection of JPEG grid inconsistencies for intuitive image forensics. *Journal of Visual Communication and Image Representation*, *54*, 155–170. https://doi.org/10.1016/j.jvcir.2018.05.011

Jain, A. K. (1989). *Fundamentals of Digital Image Processing.* (pp. 1–565).

Krawetz, N. (2011). *The Hacker Factor Blog*. A Healthy Part of This Nutritious Breakfast. Looks Like It. https://www.hackerfactor.com/blog/index.php?/archives/432-Looks-Like-It.html

López-Cantos, F., & Gasteazi, A. M. (2019). Fake Pictures. Falsification of Scientific Images and Current Advances in Forensic Analysis. Case Analysis. *Perspectivas de La Comunicación*, *12*, 209–226. www.perspectivasdelacomunicacion.cl

Lukas, J., Fridrich, J., & M, G. (2006). *Digital image hash functions: A survey.* Journal of Electronic Imaging.

Luna Gonzales, J. (2018). *Tipos de aprendizaje automático*. https://medium.com/soldai/tipos-de-aprendizaje-automático-6413e3c615e2

Martinez Garcia, Ka. G. (2017). *Manipulación fotográfica y credibilidad del medio periodístico a través de la fotografía publicada en el diario Los Angeles Times referente a la segunda Guerra del Golfo, año 2003* [Universidad San Martin de Porres]. https://repositorio.usmp.edu.pe/handle/20.500.12727/3312

Menezes, A., Oorschot, P., & VanstoneS. (1996). *Handbook of Applied Cryptography*. Mathematics and Its Applications. https://cacr.uwaterloo.ca/hac

Mohajon, J. (2020). *Confusion Matrix for Your Multi-Class Machine Learning Model*. https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machinelearning-model-ff9aa3bf7826

Molina, M. C., Prause, G. P. M., Radeva, P., & Sonka, M. (1998). *3-.D Catheter Path Reconstruction from Biplane Angiograms* (K. M. Hanson (Ed.); pp. 504–512). https://doi.org/10.1117/12.310929

Montero Morales, J. A. (2008). *Hacia una metodologia docente basada en el aprendizaje activo del estudiante presencial de ingeniería, compatible con las exigencias del EEES*. https://www.tdx.cat/handle/10803/9135

Mushtaq, S., & Mir, A. H. (2014). Digital Image Forgeries and Passive Image Authentication Techniques: A Survey. *International Journal of Advanced Science and Technology*, *73*, 15–32. https://doi.org/10.14257/ijast.2014.73.02

Nikoukhah, T., Anger, J., Ehret, T., Colom, M., Morel, J. M., & Grompone von Gioi, R. (2019). JPEG grid detection based on the number of DCT zeros and its application to automatic and localized forgery detection. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, *2019-June*, 110–118.

Olivas, E. S., Soria Olivas, E., Martín Guerrero, J. D., Martinez-Sober, M., Magdalena-Benedito, J. R., & Serrano López, A. J. (2009). *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*. https://www.igi-global.com/book/handbook-research-machine-learning-applications/486

Oostveen, J., Kalker, T., & Haitsma, J. (2002). *Feature Extraction and a Database Strategy for Video Fingerprinting*. 117–128.

Ou, Y., & Rhee, K. H. (2010). A Survey on Image Hashing for Image Authentication. *IEICE Transactions on Information and Systems*, *E93-D*(5), 1020–1030. https://doi.org/10.1587/transinf.E93.D.1020

Pfleeger, C. (2015). *Security in Computing* (5th ed.). https://www.amazon.com/Security-Computing-5th-CharlesPfleeger/dp/0134085043

Ping Dong, J., & Yongyi, Y. (2012). Digital Watermarking Robust to Geometric Distortions. In *Communications in Computer and Information Science: Vol. 342 CCIS* (Issue 12, pp. 55–62). https://doi.org/10.1007/978-3-642-35270-6_8

Ribalda, L. (2012). *Peritaje Informático y Tecnológico.* Un Enfoque Teórico-Práctico. Perito IT. https://peritoit.com

Roque Arias, G. (2002). *IMAGENES DE PIXELES (BITMAPS) E IMAGENES DE VECTORES.* http://www0.unsl.edu.ar/~tecno/multimedia/IMAG/2_1.html

Sabahi, F., Ahmad, M. O., & Swamy, M. N. S. (2018). Perceptual Image Hashing Using Random Forest for Content-based Image Retrieval. *2018 16th IEEE International New Circuits and Systems Conference, NEWCAS 2018*, 348–351. https://doi.org/10.1109/NEWCAS.2018.8585506

Samanta, P., & Jain, S. (2021). Analysis of Perceptual Hashing Algorithms in Image Manipulation Detection. *Procedia Computer Science*, *185*(June), 203–212. https://doi.org/10.1016/j.procs.2021.05.021

Sánchez Gálvez, F. E. (2021). *Análisis forense sobre imágenes digitales, un ejemplo de su aplicabilidad*. *2*, 1–10. https://dialogoforense.inacif.gob.gt/index.php/dialogoforense/article/view/47/42

Santos Tello, J. D. (2013). *Procedimientos en la investigacion, recoleccion y manejo de la evidencia digital en la escena del crimen*. 103. http://biblio3.url.edu.gt/Tesario/2013/07/03/Santos-Jorge.pdf

Sarkar, D., Bali, R., & Sharma, T. (2018). *Practical Machine Learning with Python A Problem-Solver's Guide to Building Real-World Intelligent Systems*. https://link.springer.com/book/10.1007/978-1-4842-3207-1

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.

Stallings, W. (2017). *Cryptography and Network Secuirty*.

Sultana, N. N., Mandal, B., & Puhan, N. B. (2018). Deep residual network with regularised fisher framework for detection of melanoma. *IET Computer Vision*, *12*(8), 1096–1104. https://doi.org/10.1049/iet-cvi.2018.5238

T. H., C., Ch. E, L., R. L, R., & C, S. (2009). *Introduction to algorithms*.

Tamayo, M. (2004). El proceso de la investigación científica. In *Tabasco, Mexico* (p. 164). https://books.google.com.ec/books?hl=es&lr=&id=BhymmEqkkJwC&oi=fnd&pg=PA13&dq=nivels+y+tipos+de+investigación+científica&ots=TrdH9o25oH&sig=LuUy8OrcuTqx2cp7NTQENjLPSmM#v=onepage&q=nivels y tipos de investigación científica&f=false%0Ahttps://books.google.c

Torres, J. (2020). *Python Deep Learning. Introducción Práctica con Keras y TensorFlow 2*. https://www.amazon.com/-/es/Jordi-Torres/dp/8426728286

Venkatesan, R., Koon, S. M., Jakubowski, M. H., & Moulin, P. (2000). Robust image hashing. *IEEE International Conference on Image Processing*, *3*, 1–3. https://doi.org/10.4018/978-1-4666-5888-2.ch592

Wang, H., & Wang, H. (2018). Perceptual Hashing-Based Image Copy-Move Forgery Detection. *Security and Communication Networks*, *2018*. https://doi.org/10.1155/2018/6853696

Wang, X., Zhang, Q., Jiang, C., & Xue, J. (2021). Perceptual hash-based coarse-to-fine grained image tampering forensics method. *Journal of Visual Communication and Image Representation*, *78*(March 2020), 103124. https://doi.org/10.1016/j.jvcir.2021.103124

Wassem, R., & Zenghui, W. (2018). *Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review*. *2733*(March), 2709–2733. https://doi.org/10.1162/NECO

Young, I., Gerbrands, J., & van Vliet, L. (2007). *Fundamentals of Image Processing*. 1–85. https://doi.org/10.1201/9781420046090-c13

Yuras, G. (1998). *Introducción a la percepción remota del océano*. http://www.profc.udec.cl/~gabriel/tutoriales/curso/cap06- imagenes digitales.PDF

VIII. ANEXOS

Anexo 1: Código del hash de propuesta.

```
from keras.models import load_model
from keras.applications import VGG16

from keras.preprocessing import image
import numpy as np
import os

from scipy.spatial import distance
```

```
def funcion_hash(llave1, path, n_capas=14, capa_elegida=12, llave2=122):
    """
    Calcula la función hash de una imagen.
    """
    img = keras.utils.load_img(path, target_size=(150, 150))
    #img = keras.utils.load_img(path, target_size=(224, 224))
    img_tensor = keras.utils.img_to_array(img)
    img_tensor = np.expand_dims(img_tensor, axis=0)

    capas_salida = [layer.output for layer in llave1.layers[:n_capas]]
    modelo_activado = models.Model(inputs=llave1.input, outputs=capas_salida)

    activacion = modelo_activado.predict(img_tensor)

    capa_activo = activacion[capa_elegida]
    n_map = capa_activo[0, :, :, llave2]

    dct_block = n_map
    dct_average = dct_block.mean()
    dct_block[dct_block < dct_average] = 0.0
    dct_block[dct_block != 0] = 1.0

    cadena = hash_array_to_hash_hex(dct_block.flatten())
    #cadena = zigzag_dct_hash(dct_block, 32)

    return cadena[2:], dct_block.flatten()
```

```
def hash_array_to_hash_hex(hash_array):
    """
    Convierte una matriz hash de 0 o 1 a una cadena hash en hexadecimal.
    """
    hash_array = np.array(hash_array, dtype=np.uint8)
    hash_str = ''.join(str(i) for i in 1 * hash_array.flatten())
    return hex(int(hash_str, 2))
```

```
ort tkinter as tk
ort tkinter.filedialog as filedialog
ort shutil
ort os
ort numpy as np

ve1 = VGG16(weights='imagenet', include_top=False)

t = tk.Tk()

ult_label = tk.Label(root, text="")
ult_label.pack()


ified_hash_ = tk.Label(root, text="")
ified_hash_.pack()

ginal_hash_ = tk.Label(root, text="")
ginal_hash_.pack()


 select_original_image():
 original_image_path = filedialog.askopenfilename(title="Seleccionar imagen original", filetypes=[("Archivos de imagen",
                                                                                                    "*.jpg;*.jpeg;*.png")])
 if original_image_path:
     shutil.copy(original_image_path, r"C:\Users\MSI-NB\Desktop\Original/")

 select_modified_image():
 modified_image_path = filedialog.askopenfilename(title="Seleccionar imagen modificada", filetypes=[("Archivos de imagen",
                                                                                                     "*.jpg;*.jpeg;*.png")])
 if modified_image_path:
     shutil.copy(modified_image_path, r"C:\Users\MSI-NB\Desktop\Mixti/")
```

```
def compare_images():

    ruta_carpeta = path1
    extensiones_imagen = ['.jpg', '.jpeg', '.png', '.bmp']  # Lista de extensiones de imagen

    # Obtener una lista de todos los archivos en la carpeta
    archivos_en_carpeta = os.listdir(ruta_carpeta)

    # Filtrar la lista para obtener solo los archivos de imagen
    imagenes_en_carpeta = [archivo for archivo in archivos_en_carpeta if os.path.splitext(archivo)[1] in extensiones_imagen]

    # Imprimir la ruta completa de cada imagen
    for imagen in imagenes_en_carpeta:
        ruta_imagen = os.path.join(ruta_carpeta, imagen)


    #funcion_hash(llave1, path, n_capas=14, capa_elegida=12, llave2=122):
    original_hash, original_array = funcion_hash(llave1, ruta_imagen)

    original_hash_.config(text="Hash de imagen Original     : {} ".format(original_hash))

    ruta_carpeta = path
    extensiones_imagen = ['.jpg', '.jpeg', '.png', '.bmp']  # Lista de extensiones de imagen

    # Obtener una lista de todos los archivos en la carpeta
    archivos_en_carpeta = os.listdir(ruta_carpeta)

    # Filtrar la lista para obtener solo los archivos de imagen
    imagenes_en_carpeta = [archivo for archivo in archivos_en_carpeta if os.path.splitext(archivo)[1] in extensiones_imagen]
```

```
    # Imprimir la ruta completa de cada imagen
    for imagen in imagenes_en_carpeta:
        ruta_imagen = os.path.join(ruta_carpeta, imagen)


    modified_hash, modified_array = funcion_hash(llave1, ruta_imagen)
    modified_hash_.config(text="Hash de imagen modificado: {} ".format(modified_hash))
    if np.array_equal(original_array, modified_array):

        #print("Las imágenes son iguales. Distancia de Hamming:", 0)
        result_label.config(text="Las imágenes son iguales. Distancia de Hamming: 0")
    else:

        distancia = distance.hamming(original_array,modified_array)

        result_label.config(text="Las imágenes son diferentes. Distancia de Hamming: {}".format(distancia))

original_button = tk.Button(root, text="Seleccionar imagen original", command=select_original_image)
original_button.pack()

modified_button = tk.Button(root, text="Seleccionar imagen Mixtificada", command=select_modified_image)
modified_button.pack()

compare_button = tk.Button(root, text="Comparar imágenes", command=compare_images)
compare_button.pack()

root.mainloop()
```

Anexo 2: Matriz de Consistencia

Título del Proyecto: Algoritmo hash de percepción para detectar imágenes mixtificados en entorno a la pericia de la imagen digital, Provincia de Surquillo, Región Lima, Año 2020.

Objeto de Estudio: Procesamiento de imágenes - Seguridad de la información

PROBLEMA	OBJETIVOS	HIPÓTESIS	VARIABLES / INDICADORES
PROBLEMA GENERAL: ¿Cómo el uso del algoritmo hash de percepción contribuye en la detección de imágenes mixtificadas, especialmente en contextos que involucran la pericia de la imagen digital?	**OBJETIVO GENERAL:** Evaluar de qué manera el algoritmo hash de percepción contribuye al mejoramiento en la detección de imágenes mixtificadas en el contexto de la pericia de la imagen digital.	**HIPÓTESIS GENERAL:** Se postula que el uso del algoritmo hash de percepción contribuye significativamente al mejoramiento de la detección de imágenes mixtificadas en el contexto de la pericia de la imagen digital.	**VARIABLE INDEPENDIENTE** Algoritmo Hash de Percepción **INDICADORES:** - Sensibilidad - discriminabilidad - Robustez
PROBLEMAS ESPECÍFICOS: 1. ¿De qué manera la sensibilidad del algoritmo hash de percepción aporta en la detección de manipulaciones ocultas o sutiles en imágenes mixtificadas, considerando el Recall? 2. ¿Cómo la complejidad y variabilidad de las manipulaciones impactan la detección del algoritmo hash de percepción en la identificación de imágenes mixtificadas, centrándose en la discriminabilidad? 3. ¿Es factible determinar un umbral de detección que garantice la robustez de los hashes generados por algoritmos de hash de percepción para mejorar la identificación de similitudes entre imágenes mixtificadas?	**OBJETIVOS ESPECÍFICOS:** . 1. Evaluar el aporte de la sensibilidad del algoritmo hash de percepción mediante el Recall para detectar manipulaciones sutiles u ocultas en imágenes mixtificadas. 2. Determinar cómo la complejidad y la variabilidad de las manipulaciones afectan la capacidad de detección del algoritmo hash de percepción mediante la discriminabilidad para detectar imágenes mixtificadas. 3. Determinar un umbral de detección que asegure la robustez de los hashes generados por algoritmos de hash de percepción, con el fin de mejorar la identificación precisa de similitudes entre imágenes mixtificadas.	**HIPÓTESIS ESPECÍFICAS:** 1. Se espera que la sensibilidad del algoritmo hash de percepción aporte significativamente su capacidad para detectar manipulaciones ocultas o sutiles, reflejándose en diferencias del Recall. 2. La complejidad y variabilidad de las manipulaciones tendrán un impacto directo en la capacidad del algoritmo hash de percepción para detectar imágenes mixtificadas, evidenciándose cambios significativos en su capacidad discriminabilidad. 3. Al determinar un umbral de detección óptimo para los hashes generados por algoritmos de hash de percepción garantizará la robustez y mejorará la identificación precisa de similitudes entre imágenes mixtificadas.	**VARIABLE DEPENDIENTE:** Detección de Imágenes Mixtificadas **INDICADORES:** - Originalidad - Integridad - Autenticidad

FORMATO DE AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL UNAM

Formulario de autorización para la publicación electrónica en la página web del Repositorio Institucional de la Universidad Nacional de Moquegua, de conformidad con la Ley N° 30035 Ley que regula el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto y Resolución de Comisión Organizadora N° 041-2022-UNAM que aprueba el Reglamento que regula el Repositorio Institucional Digital de la UNAM.

1. DATOS DEL AUTOR

Nombres y Apellidos	: Reynaldo Hancco Morocco
DNI/Carnet de Extranjería/Pasaporte N°	: 44536582
Correo electrónico	: adeszion@gmail.com.
Teléfono/Celular	: 900915835

2. GRADO, TITULO O ESPECIALIZACIÓN (Marque con una X)

Bachiller	Título Profesional	Magister	Doctor	Segunda Especialidad
	X			

3. TIPO DE PUBLICACIÓN (Marque con una X)

a) Articulo ( )
b) Tesis de pregrado (**X**)
c) Tesis de Maestría ( )
d) Tesis de Doctorado ( )
e) Monografía ( )
f) Reporte de caso científico o hallazgos ( )
g) Libro ( )
h) Parte de un libro ( )
i) Revisión ( )
j) Conferencia ( )
k) Presentación ( )
l) Artículo preliminar ( )
m) Articulo antes de ser publicado ( )
n) Artículo de prensa ( )
o) Documentos internos ( )
p) Imagen ( )
q) Documentos técnicos ( )
r) Ponencias presentadas en eventos científicos (Jornada, Congreso, Simposio Conferencias, etc.) ( )
s) Informes de suficiencia profesional ( )
t) Patentes producidas por académicos/as ( )

4. TIPO DE TRABAJO DE INVESTIGACIÓN (Marque con una X)

a) Tesis (**X**)
b) Trabajo de Investigación ( )
c) Trabajo de suficiencia profesional ( )
d) Trabajo académico ( )

5. DATOS DE LA PUBLICACION

a) Título : Algoritmo Hash de Percepción Para Detectar Imágenes Mixtificados en Entorno a la Pericia de la Imagen Digital, Provincia de Surquillo Región Lima, año 2020.

b) Campo del conocimiento OCDE (Indicar el código de clasificación OCDE de la obra : https://catalogos.concytec.gob.pe/vocabulario/ocde_ford.html) : Ingeniería de sistemas y comunicaciones
c) Fecha de publicación: 21 de diciembre 2023

6. DATOS DEL ASESOR:

a) Nombres y Apellidos : Alex Peter Zúñiga Incalla
b) DNI/Pasaporte/Cédula : 04641157
c) ORCID : 0000-0003-4734-2262

FIRMA : .....

DOCUMENTO DE IDENTIDAD : 44536582
NOMBRES Y APELLIDOS : Reynaldo Hancco Morocco

LICENCIAS

Licencia Estándar

Bajo los siguientes términos, autorizo el depósito de mi trabajo de investigación en el Repositorio Institucional Digital de la UNAM.

Con la autorización de depósito da mi trabajo de investigación, otorgo a la Universidad Nacional de Moquegua, una Licencia No Exclusiva para reproducir. distribuir, comunicar al público, transformar (únicamente mediante su traducción a otros idiomas) y poner a disposición del público mi trabajo de investigación (incluido el resumen), en formato físico o digital, en cualquier medio, conocido o por conocerse, a través de los diversos servicios provistos por la Universidad, creados o por crearse, tales como el Repositorio Institucional Digital de la UNAM, Colección de Tesis, entre otros, en el Perú y en el extranjero, por el tiempo y veces que considere necesarias, y libre de remuneraciones.

En virtud de dicha Licencia, la UNIVERSIDAD NACIONAL DE MOQUEGUA, podrá reproducir mi trabajo de investigación en cualquier tipo de soporte y en más de un ejemplar, sin modificar su contenido, solo con los propósitos de seguridad, respaldo y preservación.

Declaro que el trabajo de investigación es una creación de mi autoría y exclusiva titularidad, o coautoría con titularidad compartida, y me encuentro facultado a conceder la presente licencia y, asimismo, garantizo que dicho trabajo de investigación no infringe derechos de autor de terceras personas.

La UNIVERSIDAD NACIONAL DE MOQUEGUA consignara el nombre del/los autor/es del trabajo de investigación, y no le hará ninguna modificación más de la permitida en la presente licencia.

Autorizo el depósito (marque con una X).

(x) **SI autorizo que se deposite inmediatamente.**

FIRMA : .....

DOCUMENTO DE IDENTIDAD : 44536582
NOMBRES Y APELLIDOS : Reynaldo Hancco Morocco
FECHA : 28 de mayo 2024



ANEXO VIII

DECLARACIÓN JURADA DE ORIGINALIDAD DE TRABAJO DE INVESTIGACIÓN / TESIS / TRABAJO DE SUFICIENCIA PROFESIONAL

Yo, Reynaldo Hurtado Moscoso,
bachiller de la Escuela Profesional de:
Ing. de Sistemas e Informática,
de la Universidad Nacional de Moquegua, identificado(a) con DNI N°....., autor(a) del Trabajo de Investigación / Tesis / Trabajo de suficiencia profesional denominado:
Algoritmo Hash de Percepción para detectar Imagenes Manipuladas en Entorno a la percepción de la Imagen Digital, provincia Sanchilla Región Lima, año 2020

Declaro bajo juramento que:

1. El trabajo de investigación es auténtico, siendo resultado de mi trabajo personal, que no se ha copiado, que no se ha utilizado ideas, formulaciones, citas integrales e ilustraciones diversas, sacadas de cualquier tesis, obra, artículo, memoria, etc., (en versión digital o impresa), sin mencionar de forma clara y exacta su origen o autor, tanto en el cuerpo del texto, figuras, cuadros, tablas u otros que tengan derechos de autor. Por tanto, el trabajo de investigación no ha sido plagiado total ni parcialmente.
2. El trabajo de investigación no ha sido autoplagiada, es decir, no ha sido publicada ni presentada anteriormente para obtener algún grado académico previo o título profesional.
3. Los datos presentados en los resultados son reales, no han sido falseados, ni duplicados, ni copiados y por tanto los resultados que se presentan en el trabajo de investigación constituirán aporte a la realidad investigada.

Soy consciente de que, el hecho de no respetar los derechos de autor y hacer plagio, es objeto de sanciones universitarias y/o legales, por lo que asumo cualquier responsabilidad que pudiera derivarse de irregularidades en el trabajo de investigación, así como de los derechos sobre la obra presentada.

Moquegua, 28 de mayo 2024



Huella digital

Nombres y Apellidos: Reynaldo Hurtado Moscoso
DNI: 44536582

CERTIFICADO DE ANÁLISIS
magister

Tesis V12- Reynaldo Hancco

13% Textos sospechosos

- 9% Similitudes
  < 1% similitudes entre comillas
- 5% Idioma no reconocido
- 0% Textos potencialmente generados por la IA

Nombre del documento: Tesis V12- Reynaldo Hancco.docx
ID del documento: 245ff49fb5850f904dc7ee33e6c7dc257f2bd955
Tamaño del documento original: 3,74 MB

Depositante: DIEGO ISMAEL MAMANI PADILLA
Fecha de depósito: 20/12/2023
Tipo de carga: interface
fecha de fin de análisis: 20/12/2023

Número de palabras: 22.176
Número de caracteres: 137.296

Ubicación de las similitudes en el documento:



Fuentes principales detectadas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	**ayudaleyprotecciondatos.es** | El uso y validez de la prueba digital en el procedimi... https://ayudaleyprotecciondatos.es/2021/07/22/prueba-digital/#:~:text=Se entiende por prueba digit... 1 fuente similar	2%		Palabras idénticas: 2% (436 palabras)
2	**revistas.ufpr.br** | Deep learning aplicado para la detección de hemorragias y tumo... https://revistas.ufpr.br/atoz/article/download/81284/45215	1%		Palabras idénticas: 1% (225 palabras)
3	**www.scielo.org.co** | Perceptual Digest Function for Verifying Integrity in Audio For... http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S1794-91652017000100167&lng=en&tln... 1 fuente similar	< 1%		Palabras idénticas: < 1% (146 palabras)
4	**unam.edu.pe** https://unam.edu.pe/wp-content/uploads/2022/01/Registro-Proy-Invest-Estudiantes-2020.pdf	< 1%		Palabras idénticas: < 1% (91 palabras)
5	**Documento de otro usuario** #a1ab0f El documento proviene de otro grupo 1 fuente similar	< 1%		Palabras idénticas: < 1% (72 palabras)

Fuentes con similitudes fortuitas

Nº	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	**reunir.unir.net** | Herramientas de análisis forense y su aplicabilidad en la investig... https://reunir.unir.net//bitstream/123456789/2828/1/arnedo blanco.pdf	< 1%		Palabras idénticas: < 1% (40 palabras)
2	**sisbib.unmsm.edu.pe** https://sisbib.unmsm.edu.pe/bibvirtual/publicaciones/risi/2009_n2/v6n2/a02v6n2.pdf	< 1%		Palabras idénticas: < 1% (38 palabras)
3	**hdl.handle.net** | Estimación de la incertidumbre en redes neuronales profundas http://hdl.handle.net/20.500.12590/16647	< 1%		Palabras idénticas: < 1% (29 palabras)
4	**scielo.conicyt.cl** | FAKE PICTURES. FALSIFICACIÓN DE IMÁGENES CIENTÍFICAS Y AVA... https://scielo.conicyt.cl/pdf/perspectcomun/v12n1/0718-4867-perspectcomun-12-01-209.pdf	< 1%		Palabras idénticas: < 1% (29 palabras)
5	**doi.org** | Applied Sciences | Free Full-Text | Transformer-Based Subject-Sensitive ... https://doi.org/10.3390/app13031815	< 1%		Palabras idénticas: < 1% (24 palabras)

Fuentes mencionadas (sin similitudes detectadas)

Estas fuentes han sido citadas en el documento sin encontrar similitudes.

1 https://web.policia.gob.pe/anuario_estadistico/anuario_policial.html
2 https://repositorio.concytec.gob.pe/handle/20.500.12390/2260
3 https://www.academia.edu/30220448/Tutorial_digitalizacion
4 https://doi.org/10.1093/forestry/cpaa034
5 https://ayudaleyprotecciondatos.es/2021/07/22/prueba-digital/

DECENIO DE LA IGUALDAD DE OPORTUNIDADES PARA MUJERES Y HOMBRES
"AÑO DEL BICENTENARIO, DE LA CONSOLIDACIÓN DE NUESTRA INDEPENDENCIA, Y DE LA CONMEMORACIÓN DE LAS HEROICAS BATALLAS DE JUNÍN Y AYACUCHO"

RESOLUCIÓN DE COMISIÓN ORGANIZADORA Nº 391-2024-UNAM/CO

Moquegua, 12 de junio de 2024

LA COMISIÓN ORGANIZADORA DE LA UNIVERSIDAD NACIONAL DE MOQUEGUA

VISTOS:

El Acuerdo de Sesión Ordinaria de Comisión Organizadora de fecha 12 de junio de 2024; PROVEÍDO N° 9805-2024 de fecha 11 de junio de 2024 emitido por la Presidencia de la Comisión Organizadora; INFORME N° 333-2024-UNAM-CO-P/SG de fecha de recepción 12 de junio de 2024 emitido por Secretaria General; CARTA Nº 069-2024-GYT/UNAM-PCAS de fecha 05 de junio de 2024 emitido por el Especialista de Grados y Títulos; Sobre el otorgamiento del Título Profesional de Ingeniero de Sistemas e Informática y demás documentos que escoltan el expediente administrativo, y;

CONSIDERANDO:

Que, la Constitución Política del Perú en su artículo 18º prevé que cada Universidad es autónoma en su régimen normativo, de gobierno, académico, administrativo y económico, y el artículo 8º de la Ley Nº 30220, Ley Universitaria, establece que la referida autonomía inherente a las Universidades, se ejerce de conformidad con la Constitución y las Leyes de la República e implica las organizaciones de su sistema académico, económico y administrativo, así como la administración de sus bienes y rentas, elaborar su presupuesto y aplicar sus fondos con la responsabilidad que impone la Ley;

Que, de conformidad con la Ley N° 28520 de fecha 23/05/2005, se creó la Universidad Nacional de Moquegua, cuyo artículo 1º dispone: "créase la Universidad Nacional de Moquegua como persona jurídica de derecho público interno, con sede en la ciudad de Moquegua, Provincia de Mariscal Nieto, Departamento de Moquegua". Asimismo, a través de la Resolución del Consejo Directivo N° 103-2017-SUNEDU/CD, el Concejo Directivo de la SUNEDU, le otorgó la Licencia Institucional a la Universidad Nacional de Moquegua; la misma que fue modificada a través de la Resolución del Consejo Directivo N° 135-2022-SUNEDU/CD;

Que, el artículo 29° de la Ley Universitaria, Ley N° 30220, establece que, aprobada la ley de creación de una universidad pública, el Ministerio de Educación (MINEDU), constituye una Comisión Organizadora integrada por tres (3) académicos de reconocido prestigio, que cumplan los mismos requisitos para ser Rector (…). Esta Comisión tiene a su cargo la aprobación del estatuto, reglamentos y documentos de gestión académica y administrativa de la universidad, formulados en los instrumentos de planeamiento, así como su conducción y dirección hasta que se constituyan los órganos de gobierno que, de acuerdo a la presente Ley, le correspondan";

Que, mediante Resolución Viceministerial Nº 011-2024-MINEDU de fecha 31/01/2024, se reconformó la Comisión Organizadora de la Universidad Nacional de Moquegua, integrada por los académicos: Dr. Jorge Lázaro Franco Medina, como Presidente; Dr. Edwin Molina Porcel, como Vicepresidente Académico y Dr. Avid Roman Gonzalez, como Vicepresidente de Investigación;

Que, el artículo 58° de la Ley Universitaria, Ley Nº 30220, concordado con el artículo 14º del Estatuto de la Universidad Nacional de Moquegua, aprobado con Resolución de Comisión Organizadora Nº 578-2021-UNAM y sus modificatorias, establecen que el **Consejo Universitario es el máximo órgano de gestión, dirección y ejecución académica y administrativa de la universidad.** En el caso de la Universidad Nacional de Moquegua, dada su condición y naturaleza, corresponde ejercer dicha atribución

Domicilio Fiscal: Prolongación Av., Ancash S/N - Moquegua
RUC: 20449347448 - Teléfono: (053) 670168
Página Web: www.unam.edu.pe

///... Resolución de Comisión Organizadora Nº 391-2024-UNAM/CO

a la Comisión Organizadora quien hace las veces de Consejo Universitario, disposición relacionada con la Resolución Viceministerial N° 244-2021-MINEDU y sus modificatorias;

Que, el artículo 44° de la Ley Nº 30220, Ley Universitaria, señala que las universidades otorgan los grados académicos de Bachiller, Maestro, Doctor y los Títulos Profesionales que correspondan, a nombre de la Nación (...);

Que, el numeral 45.2 del artículo 45° de la Ley Nº 30220, Ley Universitaria, precisa **los requisitos mínimos para la obtención del título profesional, como son: el grado de Bachiller y la aprobación de una tesis** o trabajo de suficiencia profesional. Asimismo, establece que el título profesional solo se puede obtener en la universidad en la cual se haya obtenido el grado de bachiller;

Que, mediante Resolución de Comisión Organizadora Nº 1153-2021-UNAM, se aprobó el Reglamento de Grados y Títulos de la Universidad Nacional de Moquegua, el mismo que fue modificado a través de las Resoluciones de Comisión Organizadora N° 410; 453 y 890-2022 y N° 880-2023-UNAM el cual establece los requisitos y norma los procedimientos para la obtención de los grados académicos y títulos profesionales conferidos por esta Casa Superior de Estudios;

Que, mediante la CARTA Nº 069-2024-GYT/UNAM-PCAS, el Especialista del Área de Grados y Títulos, remite a Secretaria General, la relación de un (1) bachiller de la Escuela Profesional de Ingeniería de Sistemas e Informática, quien se encuentra apto para la expedición del Título Profesional de Ingeniero de Sistemas e Informática, al haber cumplido satisfactoriamente con los requisitos legales y reglamentarios[1], exigidos en el Reglamento de Grados y Títulos de la Universidad Nacional de Moquegua, conforme lo señala la Dirección de la Escuela Profesional de Ingeniería de Sistemas e Informática, razón por la cual a petición de la Presidencia de la Comisión Organizadora, la Secretaría General incluye el expediente como punto de agenda para ser tratado en Sesión de Comisión Organizadora de fecha 12 de junio de 2024;

---

[1] Artículo 122°. - Requisitos para la obtención del Título Profesional
Para obtener el Título Profesional el interesado deberá presentar ante su Escuela Profesional un expediente con la siguiente documentación:

a. Solicitud dirigida al Director de la Escuela Profesional o quien haga sus veces, indicando la fecha y numero de comprobante de pago.
b. Dos (2) fotografías de frente, tamaño pasaporte a color con fondo blanco (con traje formal); para garantizar la seguridad ciudadana.
c. Dos ejemplares de la Tesis, del Trabajo de Suficiencia Profesional o Informe del Examen Profesional en físico, uno para la Biblioteca y otro para la Escuela Profesional, y un CD con la versión digital.
d. Formulario de autorización del graduado para la publicación de la Tesis, Trabajo de Suficiencia Profesional o Informe de Examen Profesional en el repositorio.
e. Declaración Jurada de originalidad de la Tesis, Trabajo de Suficiencia Profesional o Informe de Examen Profesional.

La Escuela Profesional es responsable de completar el expediente del bachiller en un plazo máximo de 3 días hábiles y derivar a la Oficina de Secretaría General, adjuntando los siguientes documentos:

a. Copia del acta de sustentación indicando la modalidad de obtención del título profesional.
b. Constancia de "no adeudo" de Bienes a la Universidad, bienes a los Laboratorios Básicos y de Especialidad, material bibliográfico; dichos documentos deberán ser refrendados por los responsables a cargo.

Domicilio Fiscal: Prolongación Av., Ancash S/N - Moquegua
RUC: 20449347448 - Teléfono: (053) 670168
Página Web: www.unam.edu.pe

DECENIO DE LA IGUALDAD DE OPORTUNIDADES PARA MUJERES Y HOMBRES
"AÑO DEL BICENTENARIO, DE LA CONSOLIDACIÓN DE NUESTRA INDEPENDENCIA, Y DE LA CONMEMORACIÓN DE LAS HEROICAS BATALLAS DE JUNÍN Y AYACUCHO"

Que, se ha establecido como atribución de la Comisión Organizadora de la UNAM, la facultad de conferir los grados académicos y los títulos profesionales (...), por lo que en mérito al pedido del Secretario General y debatido el caso en Sesión Ordinaria de fecha 12 de junio de 2024, se acordó por unanimidad **conferir a nombre de la Nación el Título Profesional de Ingeniero de Sistemas e Informática por la modalidad de la Sustentación de Tesis, a un (1) Bachiller de la Escuela Profesional de Ingeniería de Sistemas e Informática de la Universidad Nacional de Moquegua**;

Estando en uso de las facultades y atribuciones conferidas en el artículo 18° de la Constitución Política del Perú; la Ley N° 30220, Ley Universitaria; el Documento Normativo denominado: "Disposiciones para la Constitución y Funcionamiento de las Comisiones Organizadoras de las Universidades Públicas en proceso de Constitución", aprobado mediante la Resolución Viceministerial N° 244-2021-MINEDU y sus modificatorias, y en la Resolución Viceministerial N° 011-2024-MINEDU, así como en el Estatuto de la Universidad Nacional de Moquegua y demás normas concordantes;

SE RESUELVE:

ARTÍCULO PRIMERO: CONFERIR, a nombre de la Nación el **TÍTULO PROFESIONAL DE INGENIERO DE SISTEMAS E INFORMÁTICA**, por la modalidad de la Sustentación de Tesis, al Bachiller de la Escuela Profesional de Ingeniería de Sistemas e Informática de la Facultad de Ingeniería y Arquitectura de la Universidad Nacional de Moquegua, que se consigna en el listado siguiente:

1. **REYNALDO HANCCO MOROCCO**

TESIS : "ALGORITMO HASH DE PERCEPCIÓN PARA DETECTAR IMÁGENES MIXIFICADOS EN ENTORNO A LA PERICIA DE LA IMAGEN DIGITAL, PROVINCIA DE SURQUILO REGIÓN DE LIMA, AÑO 2020", cuyo Asesor fue el M.Sc. Alex Peter Zúñiga Incalla.

ARTÍCULO SEGUNDO: AUTORIZAR, a la Secretaría General de la Universidad Nacional de Moquegua, expedir y registrar los Diplomas correspondientes para su otorgamiento.

ARTÍCULO TERCERO: ENCARGAR al Área de Grados y Títulos de la Universidad Nacional de Moquegua, para que realice las acciones pertinentes conducentes, relacionadas con la inscripción de los grados y títulos en el Registro de Grados y Títulos de la Superintendencia Nacional de Educación Superior Universitaria - SUNEDU.

ARTÍCULO CUARTO: NOTIFICAR, la presente Resolución a todas las instancias y dependencias administrativas y académicas de la Universidad Nacional de Moquegua, para su conocimiento, cumplimiento y demás fines pertinentes.

ARTÍCULO QUINTO: ENCARGAR, a la Oficina de Tecnología de la Información, la publicación de la presente Resolución en el portal de transparencia de la Universidad Nacional de Moquegua.

REGÍSTRESE, COMUNÍQUESE, CÚMPLASE Y ARCHÍVESE.

DR. JORGE LAZARO FRANCO MEDINA
Presidente de la Comisión Organizadora
Universidad Nacional de Moquegua

Mg. KEVIN YUVER CHIHUAN QUISPE
Secretario General
Universidad Nacional de Moquegua

Domicilio Fiscal: Prolongación Av., Ancash S/N - Moquegua
RUC: 20449347448 - Teléfono: (053) 670168
Página Web: www.unam.edu.pe