



**Universidad
Zaragoza**

Trabajo Fin de Grado

Evaluación de técnicas forenses en la detección del
origen de imágenes digitales: impacto del
postprocesado móvil y redes sociales

Evaluation of forensic techniques in detecting the
origin of digital images: impact of mobile
post-processing and social networks

Autor

Daniel Simón Gayán

Directores

Ricardo Julio Rodríguez Fernández
José Roldán Gómez

Grado en Ingeniería Informática

ESCUELA DE INGENIERÍA Y ARQUITECTURA
2026

Agradecimientos

Quiero expresar mi agradecimiento a mis tutores, Ricardo y José, por su atención constante, su paciencia y por guiarme en cada paso de este trabajo. A mis padres y a mi hermana, por estar siempre a mi lado, aguantarme en los momentos más complicados y apoyarme en todo desde el primer día. Y por supuesto, a mi pareja Alba y a mis amigos, por las risas, la desconexión y por hacer que los años de carrera hayan sido una experiencia muchísimo más amena. Gracias a todos.

Evaluación de técnicas forenses en la detección del origen de imágenes digitales: impacto del postprocesado móvil y redes sociales

RESUMEN

La informática forense es la disciplina científica que se encarga de la adquisición, preservación, análisis y presentación de evidencias digitales con validez legal. En los últimos años, esta ciencia se ha visto comprometida debido al auge de los *smartphones* modernos y las redes sociales, los cuales han planteado desafíos sin precedentes. El procesamiento computacional nativo que aplican dichos *smartphones* y los algoritmos de compresión de las aplicaciones difuminan las huellas invisibles que los dispositivos dejan en las imágenes, comprometiendo la fiabilidad de los métodos de identificación de origen de las imágenes. Este Trabajo Fin de Grado realiza un análisis comparativo de la robustez de las dos metodologías predominantes: el método clásico basado en el ruido físico del sensor y el enfoque basado en aprendizaje profundo. Para ello, se ha desarrollado un flujo modular unificado en Python y se ha evaluado un banco de pruebas expuesto a la degradación por transferencia en WhatsApp e Instagram, así como a filtros lineales y espaciales. Este trabajo proporciona una visión crítica sobre los límites operativos de la forense digital actual, fundamentando la necesidad de actualizar las herramientas forenses ante las constantes amenazas que surgen día a día.

Evaluation of forensic techniques in detecting the origin of digital images: impact of mobile post-processing and social networks

ABSTRACT

Digital forensics is the scientific discipline responsible for the acquisition, preservation, analysis, and presentation of legally valid digital evidence. In recent years, this science has been compromised due to the rise of modern smartphones and social networks, which have posed unprecedented challenges. The native computational processing applied by these smartphones, combined with the compression algorithms used by applications, blurs the invisible fingerprints that devices leave on images, thereby compromising the reliability of image source identification methods. This Bachelor's Thesis performs a comparative analysis of the robustness of the two predominant methodologies: the classical method based on physical sensor noise and the deep learning-based approach. To this end, a unified pipeline has been developed in Python, and an exhaustive benchmark has been evaluated under transfer degradation via WhatsApp and Instagram, as well as linear and spatial filters. This work provides a critical insight into the operational limits of current digital forensics, establishing the need to update forensic tools against the constant threats that emerge day by day.

Índice general

1. Introducción	1
1.1. Contexto y motivación	1
1.2. Objetivos	2
1.3. Estructura del documento	2
2. Conceptos previos y marco teórico	3
2.1. Análisis forense de imágenes digitales	3
2.2. Métodos de identificación de huellas digitales	4
3. Estado del arte	6
3.1. Los inicios de la identificación forense	6
3.2. El cambio de paradigma: aprendizaje profundo y redes neuronales	7
3.3. El panorama actual y retos del análisis forense digital	8
4. Metodología y entorno de trabajo	11
4.1. Metodología de evaluación	11
4.2. Conjuntos de datos utilizados	12
4.2.1. Análisis de bases de datos públicas y sus limitaciones	12
4.2.2. Elaboración de un conjunto de datos propio	12
4.3. Herramientas y tecnologías	13
4.4. Entorno de hardware	13
5. Diseño e implementación del banco de pruebas	14
5.1. Arquitectura del sistema	14
5.2. Extracción y análisis de huellas forenses	16
5.2.1. Aislamiento del ruido de patrón fijo	16
5.2.2. Inferencia del residuo algorítmico	17
5.3. Métricas de decisión y evaluación	18
5.3.1. Métricas de decisión forense	18
5.3.2. Métricas de evaluación de rendimiento	19
6. Experimentos y resultados	22
6.1. Análisis preliminar y comportamiento base	22
6.2. Impacto del postprocesado	23
6.2.1. Análisis y rendimiento del método PRNU	23
6.2.2. Análisis y rendimiento del método Noiseprint	24
6.3. Impacto de redes sociales	25
6.3.1. Impacto de WhatsApp	25
6.3.2. Impacto de Instagram	26

6.4. Impacto de los filtros	27
6.4.1. Filtro lineal	28
6.4.2. Filtro espacial	31
6.5. Comparativa de robustez	32
7. Conclusiones y trabajo futuro	34
Bibliografía	36
A. Planificación temporal y diagrama de Gantt	38

Capítulo 1

Introducción

1.1. Contexto y motivación

En la actualidad, el uso de dispositivos móviles como cámara fotográfica principal se ha universalizado. Por ello, la captura y distribución de contenido multimedia mediante dispositivos móviles se realiza de manera más rápida y sencilla. Sin embargo, esta accesibilidad tecnológica también ha facilitado la alteración, edición y falsificación de imágenes digitales [1].

Debido a estas amenazas, la informática forense es fundamental para verificar la autenticidad y el origen de las fotografías. Tradicionalmente, la identificación del dispositivo de captura se ha basado en la extracción de imperfecciones físicas del hardware. En este ámbito, el ruido de patrón fijo del sensor, conocido como ruido de no uniformidad de respuesta (PRNU, de sus siglas en inglés, *Photo Response Non Uniformity*), se ha consolidado como la huella digital por excelencia para relacionar una imagen con su cámara de origen [2].

No obstante, la evolución tecnológica plantea nuevos y complejos desafíos para estos métodos clásicos. Por un lado, actualmente se incorporan algoritmos avanzados de postprocesamiento en los dispositivos móviles, los cuales alteran la señal original del sensor antes de guardar la imagen. Por otro lado, no solo se realizan y se guardan las imágenes, sino que se suelen compartir a través de redes sociales como WhatsApp o Instagram, que realizan procesos de compresión y redimensionamientos espaciales para conseguir optimizar el almacenamiento y ancho de banda [3].

Estas transformaciones, tanto en el origen como durante la transmisión, amenazan la precisión de los métodos forenses. Como respuesta a esta problemática, recientemente se han propuesto alternativas basadas en Inteligencia Artificial (IA) como Noiseprint [4], diseñadas para extraer y clasificar las huellas dejadas por el propio software del dispositivo.

Por tanto, surge la necesidad de evaluar la robustez y rendimiento de los distintos enfoques forenses actuales frente a escenarios reales de uso. Por ejemplo, analizar cómo reaccionan las técnicas basadas en hardware frente a las basadas en IA al enfrentarse a las degradaciones provocadas por el postprocesado móvil y las aplicaciones de terceros.

1.2. Objetivos

El objetivo principal de este Trabajo Fin de Grado (TFG) es evaluar el rendimiento y la robustez de las técnicas forenses actuales (PRNU y Noiseprint) en la detección del origen de imágenes digitales, analizando específicamente el impacto que genera el postprocesado de los dispositivos móviles modernos y la posible compresión de las redes sociales.

Para lograr este propósito, el objetivo principal se divide en los siguientes subobjetivos:

- SO1** Implementar un entorno de evaluación unificado: Desarrollar un *pipeline* que integre tanto los métodos clásicos basados en defectos físicos como los enfoques modernos basados en IA.
- SO2** Construir un banco de pruebas: Configurar un conjunto de datos compuesto por fotografías capturadas con *smartphones* actuales, sometiénolas a los procesos de compresión habituales de plataformas como WhatsApp o Instagram.
- SO3** Realizar una evaluación comparativa: Enfrentar de manera empírica ambos métodos forenses bajo los mismos escenarios de degradación, midiendo su precisión y tasa de acierto.
- SO4** Analizar las vulnerabilidades actuales: Identificar las limitaciones de las herramientas existentes frente a la fotografía computacional moderna, aportando evidencia sobre los nuevos retos de la disciplina forense.

1.3. Estructura del documento

Este documento consta de 7 capítulos y un anexo. El Capítulo 2 explica el contexto necesario para que el lector comprenda el resto del documento, introduciendo el análisis forense de imágenes y los métodos de identificación de huellas digitales. El Capítulo 3 es una descripción de otros trabajos relacionados, resaltando las diferencias y similitudes con este. El Capítulo 4 explica la metodología seguida y el entorno de trabajo, mostrando las herramientas que se han utilizado y las características de los conjuntos de datos usados a lo largo del proyecto. El Capítulo 5 describe el diseño e implementación del banco de pruebas, explicando la arquitectura del sistema, la configuración del entorno y las métricas de evaluación seleccionadas. El Capítulo 6 expone las pruebas realizadas, evaluando el impacto del postprocesado móvil, la compresión y las aplicaciones de terceros, junto con un análisis de los resultados obtenidos. El Capítulo 7 presenta las conclusiones y plantea posibles líneas de trabajo futuro. Por último, en el Apéndice A se detallan las horas invertidas en el proyecto y su distribución en tareas y en el tiempo.

Capítulo 2

Conceptos previos y marco teórico

En este capítulo se presentan los conceptos fundamentales para entender el trabajo realizado. En primer lugar, se comienza explicando qué es una imagen digital y posteriormente las fases que intervienen en la formación de esta. Por último, se detallan dos de los métodos que existen para identificar su origen, siendo uno basado en el hardware y el otro basado en el software.

2.1. Análisis forense de imágenes digitales

Para comprender cómo funciona el mundo de la identificación de cámara de origen se debe analizar previamente la naturaleza de una fotografía digital. Una imagen no es más que una representación bidimensional discreta, formada por una matriz matemática de píxeles. El proceso mediante el cual una escena física se transforma en esta estructura de datos se conoce como cadena de adquisición de la cámara. En cada una de las fases de esta cadena, el hardware y el software del dispositivo alteran la señal original y dejan patrones imperceptibles a simple vista que, posteriormente, serán los que se detecten a través del análisis forense [5].

Como se aprecia en la Figura 2.1, la primera fase del proceso comienza cuando la luz de la escena atraviesa el sistema de lentes del dispositivo. Dado que los sensores son daltónicos por naturaleza y solo miden la intensidad luminosa, la luz pasa primero por un Filtro de Arreglo de Color (CFA, de sus siglas en inglés, *Color Filter Array*). El patrón más extendido es el filtro de Bayer, el cual permite que cada píxel capture únicamente un color primario (rojo, verde o azul). A continuación, los fotones filtrados impactan sobre el sensor de imagen, que son habitualmente de tecnología de Dispositivo de Carga Acoplada (CCD, de sus siglas en inglés, *Charge-Coupled Device*) o de Semiconductor Complementario de Óxido Metálico (CMOS, de sus siglas en inglés, *Complementary Metal-Oxide-Semiconductor*), donde se genera la señal digital inicial [6]. Es precisamente en esta etapa física donde se suman diversas fuentes de ruido inherentes al hardware, como la corriente oscura, el ruido de disparo, el ruido de lectura y, fundamentalmente, PRNU, conformando así la imagen cruda o conocida en inglés como imagen *raw*.

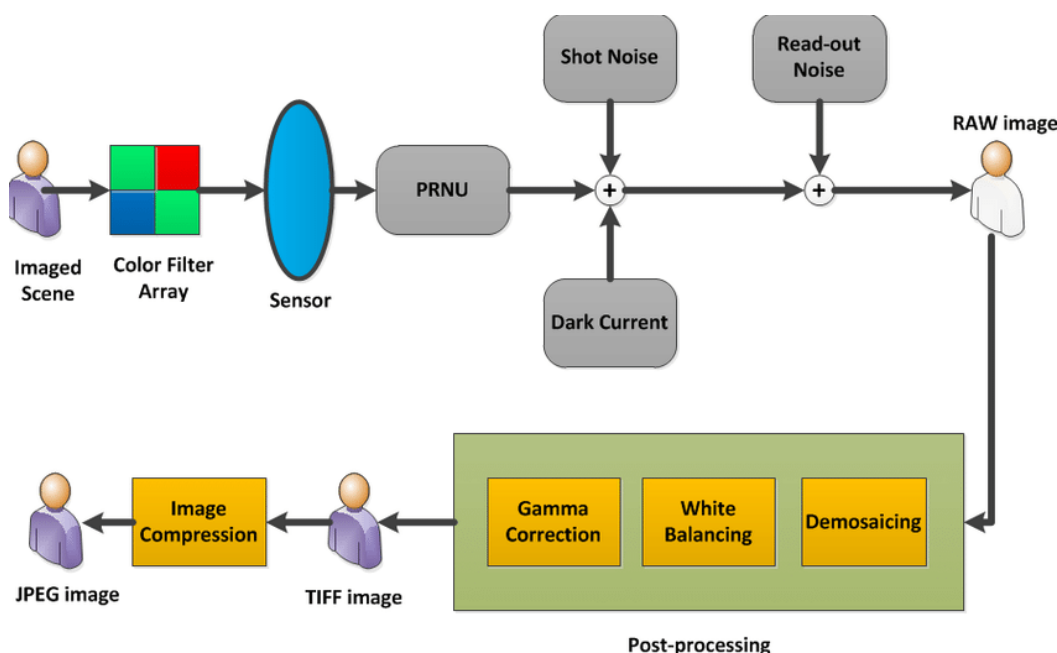


Figura 2.1: Ilustración de la cadena de adquisición típica dentro de una cámara digital y sus fuentes de ruido. Fuente: [7].

A continuación, esta señal *raw* se transfiere al Procesador de Señal de Imagen (ISP, de sus siglas en inglés, *Image Signal Processor*) para la etapa de postprocesado. Aquí se ejecuta un algoritmo de interpolación espacial conocido como *demosaiicing*, cuyo objetivo es estimar los dos colores faltantes para cada píxel basándose en sus valores vecinos, ya que cada píxel solo ha capturado un único color primario. Tras esta reconstrucción del color, el procesador aplica una serie de correcciones automáticas, como el balance de blancos y la corrección gamma, generando una imagen digital sin comprimir que se suele denominar Formato de Archivo de Imagen Etiquetada (TIFF, de sus siglas en inglés, *Tagged Image File Format*) [5]. Adicionalmente, el ISP también aplica algoritmos propietarios que dejan dependencias espaciales únicas en la imagen.

Finalmente, esta imagen procesada se somete a un algoritmo de compresión con pérdidas, siendo el estándar JPEG [8] el formato mayoritario, y se almacena en la memoria del dispositivo. Como resultado de todo este flujo, la fotografía final no es un reflejo exacto de la luz entrante, sino el producto de múltiples operaciones físicas y algorítmicas. La ciencia forense de imágenes explora las regularidades estadísticas y las huellas ocultas que estas operaciones graban en la matriz de píxeles para verificar su integridad o descubrir su origen [6].

2.2. Métodos de identificación de huellas digitales

Una vez explicado el proceso de formación de una imagen, es evidente que la señal final contiene huellas para identificar el dispositivo de origen [6]. En el mundo forense, existen principalmente dos enfoques para detectar estas huellas. Por un lado, destacan los métodos físicos basados en el hardware. Por otro lado, emergen los enfoques modernos basados en el software y la IA. Ambos se describen a continuación en más detalle.

El enfoque físico: PRNU

La identificación clásica del dispositivo de captura de la imagen se explica con el análisis de los defectos físicos del hardware. Dentro de estas imperfecciones, destaca especialmente el ruido conocido como PRNU [6]. Su origen radica en las anomalías del silicio generadas durante la fabricación de los sensores CCD o CMOS [6]. Como consecuencia de estas variaciones microscópicas, la respuesta ante la luz es ligeramente distinta entre los fotodiodos del componente.

Matemáticamente hablando, la huella PRNU se trata como un factor de ruido multiplicativo presente en las altas frecuencias de la imagen [6]. Además, esta señal residual no suele cambiar si la imagen se conserva a lo largo de su ciclo de vida bajo condiciones normales [6]. Por lo tanto, este patrón se puede definir como una huella digital del propio sensor que permite diferenciar dispositivos de forma inequívoca.

El proceso forense para aislar esta huella digital exige separar la información visual de la escena fotografiada del ruido inherente. Para lograrlo, se aplica un filtro de suavizado espacial sobre la imagen. A continuación, se calcula la diferencia entre la fotografía original y la versión filtrada para extraer un mapa de ruido residual. Finalmente, una vez se tiene el mapa residual, se evalúa la similitud matemática entre dicho residuo y un patrón de referencia del dispositivo sospechoso previamente calculado. Para saber si la imagen pertenece al dispositivo, la similitud matemática debe superar un límite prefijado o ser el máximo entre varios patrones de referencia [6].

El enfoque de software e IA: Noiseprint

PRNU se ha consolidado como el estándar en la identificación de origen. No obstante, se vuelve vulnerable a alteraciones de procesos habituales en el entorno móvil, ya que procesos como la compresión JPEG y el reescalado destruyen la sincronización de la señal [7]. Estas alteraciones reducen considerablemente la eficacia de los métodos forenses tradicionales.

Como respuesta a estas limitaciones recientes, ha surgido una nueva generación de herramientas forenses basadas en el aprendizaje profundo. Entre estas alternativas destaca Noiseprint [4]. A diferencia de PRNU, Noiseprint extrae las huellas generadas por el software de la cámara.

Noiseprint utiliza Redes Neuronales Convolucionales (CNN) en una arquitectura siamesa [4]. Este modelo se entrena para extraer un mapa de residuos donde se eliminan las características del contenido visual original. De este modo, se resalta exclusivamente la huella del modelo de cámara. Al basarse en patrones algorítmicos complejos extraídos mediante IA, este método presenta una mayor resiliencia ante manipulaciones post-captura.

Capítulo 3

Estado del arte

Este capítulo revisa los trabajos de investigación más relevantes en el campo de la identificación del origen de imágenes digitales a lo largo de los años. Se explica la transición desde los métodos basados en los defectos del hardware hasta las soluciones más actuales basadas en el aprendizaje profundo. Asimismo, se resalta el impacto de las redes sociales y las imágenes generadas por IA en el ámbito de la informática forense.

3.1. Los inicios de la identificación forense

La identificación del origen del contenido multimedia comenzó a llamar la atención con el auge de las cámaras de vídeo. Los trabajos iniciales se centraron en identificar imperfecciones intrínsecas en los componentes físicos de captura.

En 1999, se planteó un método pionero basado en la extracción de ruidos de patrón fijo presentes en los sensores CCD de las cámaras de vídeo. Se identificó que las irregularidades en la corriente oscura de estos componentes creaban marcas únicas en las grabaciones. Este planteamiento permitió encontrar una conexión directa entre un vídeo y un dispositivo físico concreto simplemente mediante el análisis del ruido de píxeles defectuosos [9].

Posteriormente, la investigación se expandió a los artefactos generados por la lógica de procesamiento interna de los dispositivos. En 2005, se presentó una técnica basada en las huellas dejadas por la interpolación del filtro de color. Se demostró que los algoritmos de *demosaiçing* que usan los diferentes fabricantes introducen patrones estadísticos específicos en la matriz de píxeles. Por lo tanto, estos patrones permiten identificar el modelo de cámara a partir de las dependencias espaciales presentes en la imagen procesada [10].

En 2006, el foco de estudio se extendió hacia los componentes ópticos del sistema de captura. Se propuso el uso de la distorsión radial de la lente como una firma biométrica que pueda identificar la cámara de origen [11]. Se encontró que cada lente introduce distorsiones geométricas únicas en la imagen debido a imperfecciones microscópicas en su fabricación. Esto se debe a que cada unidad de lente es única. Este método permitió distinguir incluso entre dispositivos del mismo modelo al estimar

parámetros de distorsión radial a partir de las imágenes [11].

Durante estos años, también se analizó el impacto del procesamiento de software y la compresión a la hora de identificar las fuentes de las imágenes. Se introdujo el análisis de las tablas de cuantización JPEG, que demostró que cada cámara, ya sea de una marca o modelo específico, utiliza tablas de cuantización propias para encontrar el equilibrio perfecto entre la calidad de la imagen y el espacio de almacenamiento [12]. Unos años después, la robustez de este método se validó mediante el estudio de extensas bases de datos, confirmando que es posible identificar la marca y el modelo con una alta precisión a partir de estas trazas de compresión [13].

Finalmente, el campo forense se consolidó con la caracterización de PRNU. En 2008, se presentó un marco formal para su uso como una huella digital de alta fiabilidad. Se comprobó que este ruido, de naturaleza multiplicativa y estable en el tiempo, permite identificar un sensor individual con una precisión muy buena en entornos controlados. Este hallazgo convirtió al PRNU en el estándar de referencia para la identificación de dispositivos y la verificación de integridad de imágenes durante la siguiente década [2].

3.2. El cambio de paradigma: aprendizaje profundo y redes neuronales

A pesar de la alta fiabilidad de PRNU en entornos controlados, este método, como se ha mencionado anteriormente, presenta debilidades que le hacen vulnerable a manipulaciones espaciales comunes, por ejemplo la compresión o el reescalado de imagen. Esta limitación impulsó a la comunidad científica a investigar otros métodos o enfoques, encontrando soluciones basándose en la IA.

En 2016, se propuso uno de los primeros enfoques basados en CNN para la identificación del modelo de cámara [14]. Inicialmente, los filtros convolucionales tendían a eliminar el ruido natural de la imagen. Para solucionar este obstáculo, se implementó una etapa de preprocesamiento mediante un filtro de paso alto antes de introducir la imagen a la red neuronal [14]. Este enfoque permitió que el modelo aprendiera a clasificar basándose exclusivamente en los residuos de la imagen.

Poco después, se descubrió que las CNN tenían una capacidad de análisis superior a los clasificadores estadísticos tradicionales. En 2017, se consiguió que un algoritmo fuera capaz de aprender características discriminativas directamente a partir de pequeñas partes de una imagen [15]. Este método superó ampliamente el rendimiento de los algoritmos del estado del arte, demostrando que los modelos basados en datos son muy efectivos para identificar fuentes [15].

Otro avance significativo en esta área fue la presentación de Noiseprint, un método basado en una arquitectura neuronal siamesa. Al entrenar el modelo siamés para suprimir el contenido visual de la escena original, la CNN puede aislar de forma exclusiva estas trazas del procesamiento. Además, se demostró empíricamente que estas huellas de software son más resistentes frente a compresiones agresivas [4]. De este modo,

el aprendizaje profundo abrió una nueva vía de investigación mucho más robusta y adaptada a los escenarios forenses modernos.

3.3. El panorama actual y retos del análisis forense digital

En la actualidad, la forma en la que se adquieren los contenidos multimedia ha migrado de las cámaras tradicionales hacia los dispositivos móviles. Los *smartphones* hoy en día tienen técnicas de fotografía computacional muy avanzadas e integran sistemas de imagen cada vez más complejos. A través de sus ISP, estos dispositivos aplican múltiples filtros y ajustes automáticos en tiempo real. Investigaciones recientes confirman que estas operaciones alteran la señal original, lo que provoca que las huellas clásicas, como PRNU, sean inestables y sufran cambios en comparación con las cámaras fotográficas convencionales [16].

A este desafío de hardware y software interno se suma el impacto de las plataformas de comunicación que se utilizan constantemente. A día de hoy, la mayoría de las imágenes se comparten a través de redes sociales y aplicaciones de mensajería, como WhatsApp, Facebook e Instagram. Estas plataformas aplican algoritmos de compresión y eliminan los metadatos de los archivos para optimizar el tráfico de red. Algunos estudios recientes han demostrado que esta degradación destruye gran parte de la información necesaria para la identificación de la fuente [17]. Para contrarrestar este efecto, la comunidad científica ha comenzado a entrenar arquitecturas profundas, como el modelo *Inception V3*, orientadas específicamente a la identificación ciega de fuentes en estos entornos hostiles [17].

Paralelamente, la informática forense se enfrenta al reciente auge de las imágenes generadas por IA. Diversos estudios actuales evalúan si los patrones de ruido sintético introducidos por los modelos de IA interfieren con las huellas de los dispositivos físicos reales [18]. Esta nueva rama de investigación muestra que el análisis del ruido residual sigue siendo una herramienta fundamental, ya sea para identificar el teléfono de origen o para descartar que la fotografía haya sido creada por un algoritmo de IA.

Al igual que los estudios analizados, este trabajo utiliza el ruido residual como huella identificativa, apoyándose en la formulación matemática original de PRNU y en la red neuronal de Noiseprint. Sin embargo, la principal diferencia de este trabajo frente a los mencionados anteriormente es su entorno de validación. Mientras que los trabajos previos suelen evaluar estos métodos de forma aislada y sobre bases de datos grandes y de alta calidad, este TFG plantea una evaluación en un escenario real. En lugar de diseñar un nuevo algoritmo, se ha desarrollado un clasificador propio que actúa como un entorno de pruebas común. Esto permite enfrentar cara a cara un método clásico (PRNU) y uno moderno (Noiseprint) bajo las mismas condiciones adversas contemporáneas: imágenes tomadas con *smartphones* actuales y degradadas por la compresión de las redes sociales. De esta manera, se realiza una comparativa directa y actualizada que los estudios originales no pudieron contemplar en su momento.

A modo de síntesis, la Tabla 3.1 resume la evolución cronológica de los principales trabajos analizados en este capítulo, categorizados según su enfoque tecnológico, y sitúa la contribución del presente TFG frente a la literatura previa.

Tabla 3.1: Resumen de la evolución del estado del arte y posicionamiento de nuestra propuesta.

Categoría	Autores (año)	Contribución principal	Entorno de prueba / Limitaciones
Métodos clásicos: Hardware y procesamiento interno	Kurosawa (1999)	Extracción de ruido de patrón fijo en sensores CCD.	Entornos muy controlados (grabaciones de vídeo).
	Bayram (2005)	Clasificación de dispositivos basada en las huellas del CFA y el <i>demosaiçing</i> .	Limitado a imágenes no muy comprimidas.
	Choi (2006)	Uso de la distorsión radial intrínseca de la lente como firma biométrica de la cámara.	Requiere imágenes con referencias geométricas para medir cómo se curva la lente.
	Farid (2006, 2008)	Balística digital orientada al software mediante el análisis de tablas de cuantización JPEG.	Vulnerable a recompresión externa y modificaciones de metadatos.
	Chen et al. (2008)	PRNU: Consolidación del estándar físico de referencia basado en la huella del sensor.	Vulnerable a alteraciones espaciales severas (reescalado, compresión).
Métodos basados en IA y software (ISP)	Tuama (2016)	CNN con filtros de paso alto para aislar los residuos de la imagen.	Dificultad inicial de las redes para no suprimir el ruido natural.
	Bondi (2017)	Modelo basado en datos para la clasificación mediante fragmentos.	Principalmente evaluado sobre bases de datos de imágenes prístinas.
	Cozzolino (2020)	Noiseprint: Huella de software extraída mediante arquitectura neuronal siamesa.	Ligeramente dependiente al conjunto de datos de entrenamiento.
Nuestra propuesta	Este trabajo (2026)	Evaluación unificada (PRNU clásico frente a Noiseprint moderno).	Escenario real: <i>smartphones</i> actuales y compresión de redes sociales.

Capítulo 4

Metodología y entorno de trabajo

En este capítulo se describe detalladamente el marco experimental diseñado para llevar a cabo la evaluación de las técnicas forenses presentadas en los capítulos anteriores. En primer lugar, se define el enfoque metodológico que guía los experimentos y establece las bases de la comparativa. Después, se explican las bases de datos de imágenes seleccionadas para la construcción del banco de pruebas, asegurando la presencia de imágenes capturadas con *smartphones*. Finalmente, se introducen las herramientas de software utilizadas para el desarrollo del clasificador y los recursos de hardware que se utilizan para la ejecución de los modelos de IA.

4.1. Metodología de evaluación

La metodología aplicada en este TFG se fundamenta en un enfoque empírico y comparativo. Tal y como se ha justificado en los objetivos y análisis del estado del arte, el propósito de este proyecto es la evaluación de los dos paradigmas forenses predominantes, PRNU y Noiseprint, a escenarios reales y actuales. Para asegurar que la comparación sea justa y precisa, se ha creado un plan de evaluación paso a paso:

- **Fase de control (análisis nativo):** En esta etapa inicial, se busca evaluar los clasificadores sobre las imágenes nativas extraídas directamente de los dispositivos, sin alteraciones externas. El propósito de esta fase es demostrar empíricamente cómo el postprocesado nativo de los *smartphones* actuales, la llamada fotografía computacional, ya es un obstáculo para las técnicas forenses, incluso antes de aplicar cualquier compresión externa.
- **Fase de estrés (redes sociales):** Una vez establecida la línea base del comportamiento nativo, los modelos se someten a un entorno de pruebas más complejo que simula la vida de una fotografía digital en su día a día. Para ello, las imágenes se procesan a través de aplicaciones de mensajería y redes sociales, introduciendo la compresión característica de plataformas como WhatsApp o Instagram.

Este diseño metodológico incremental permite cuantificar de manera empírica no solo la tasa de acierto en general, sino también separar qué porcentaje del fracaso de un método forense se debe al propio teléfono móvil y qué porcentaje es atribuible a la compresión que aplican las redes sociales.

4.2. Conjuntos de datos utilizados

Dado que el objetivo de este trabajo es comparar el rendimiento de los algoritmos frente a la fotografía móvil contemporánea, la selección del banco de pruebas requiere un análisis previo de las opciones públicas que hay disponibles hoy en día.

4.2.1. Análisis de bases de datos públicas y sus limitaciones

Durante la fase de investigación, se analizan las bases de datos forenses más reconocidas en la literatura académica:

- **Dresden Image Database [19]:** Estándar histórico por excelencia en la validación de PRNU. Su principal limitación para este proyecto es la ausencia de *smartphones* ya que está formada mayoritariamente por fotografías de cámaras digitales compactas y réflex de principios de la década de 2010.
- **SOCRatES (SOurce Camera REcognition on Smartphones) [20]:** Esta base de datos se centra en teléfonos móviles, aunque los dispositivos que incluye son ya muy antiguos. Las imágenes de esta base de datos no muestran la complejidad de la fotografía computacional que tienen los *smartphones* actuales.
- **VISION Dataset [21]:** Esta base de datos está diseñada para tratar de luchar precisamente contra los problemas que empiezan a surgir en imágenes capturadas con *smartphones*, que muchas veces son compartidas a través de plataformas como WhatsApp, YouTube o Facebook. Sin embargo, al haber sido publicada en 2017, los dispositivos incluidos (como el Apple iPhone 6 o el Samsung Galaxy S5) no tienen motores de procesamiento computacional tan agresivos como los dispositivos de hoy en día.

Ante la falta de representatividad de estas opciones tradicionales, se decide utilizar el conjunto de datos **Multi-Camid**, alojado por el *Center for Statistics and Applications in Forensic Evidence* (CSAFE)¹. Esta base de datos destaca por incluir una variedad de *smartphones* mucho más modernos, adecuándose mejor a los requisitos del trabajo. Su única limitación notable es que el volumen es limitado, ya que cuenta con un conjunto reducido de 100 fotografías por dispositivo. No obstante, al tener en cuenta los requisitos del proyecto, se concluye que la actualidad tecnológica de los sensores tiene más peso que esta limitación.

4.2.2. Elaboración de un conjunto de datos propio

Para añadir más realismo a los experimentos, se toma la decisión de construir un pequeño conjunto de datos propio de apoyo. Se construye capturando imágenes directamente desde dos *smartphones* modernos que pertenecen al autor: un Apple iPhone 14 Pro y un Apple iPhone 15. Ambos integran motores de procesamiento nativo (como el *Photonic Engine*, *Deep Fusion* y *Smart HDR*). La combinación del conjunto de datos público de CSAFE junto con las capturas de estos dispositivos Apple permite someter tanto a PRNU como a Noiseprint a los escenarios más exigentes y realistas posibles dentro del panorama tecnológico actual. Por cuestiones de privacidad, este conjunto de datos no es público, pero está disponible bajo petición.

¹Disponible en: <https://data.csafe.iastate.edu/Multi-Camid/>

4.3. Herramientas y tecnologías

Para llevar a cabo el desarrollo del banco de pruebas y la ejecución de los experimentos se utiliza un conjunto de herramientas de software de código abierto. Concretamente:

- **Python:** Se usa como lenguaje de programación principal para la construcción de todo el *pipeline* experimental, desde la carga de imágenes hasta el cálculo de métricas.
- **Anaconda:** La gestión de entornos virtuales se realiza mediante Anaconda. Esta herramienta es necesaria ya que el modelo de IA evaluado (Noiseprint) requiere versiones antiguas de bibliotecas, permitiendo así aislar estas dependencias sin generar conflictos en el sistema central.
- **TensorFlow:** Se emplea este *framework* de aprendizaje profundo para la ejecución de Noiseprint. Debido a los requisitos de la arquitectura original propuesta por los autores, es necesario utilizar una versión específica antigua de esta biblioteca, gestionada a través del entorno virtual mencionado anteriormente.
- **OpenCV y bibliotecas auxiliares:** Para el tratamiento digital de las imágenes como recortes, por ejemplo, se utiliza principalmente OpenCV, apoyado en otras bibliotecas estándar del ecosistema de Python como NumPy.
- **Implementación PRNU:** La extracción de la huella del sensor se resuelve utilizando una implementación de código abierto en Python². Este código se revisa e integra en el clasificador principal para unificar toda la evaluación bajo un mismo lenguaje.

4.4. Entorno de hardware

La utilización de modelos de IA o el procesamiento de imágenes requieren bastantes recursos computacionales. En la fase inicial del proyecto, se plantea la posibilidad de utilizar plataformas como MATLAB o servidores proporcionados por la universidad. Sin embargo, dado que las pruebas de concepto se desarrollan correctamente en un entorno local, se toma la decisión de mantener la ejecución de todos los experimentos en el ordenador personal del autor. Esta decisión evita posibles problemas como límites de uso o desconexiones imprevistas que puedan suceder en otras plataformas, y permite más control sobre el almacenamiento y la manipulación del conjunto de datos privado.

Las especificaciones técnicas del dispositivo utilizado para los experimentos son:

- **Procesador (CPU):** Intel Core i7-12650H
- **Memoria RAM:** 16 GB DDR5
- **Tarjeta Gráfica (GPU):** NVIDIA GeForce RTX 3060 Laptop

²Implementación PRNU en Python: <https://github.com/polimi-ispl/prnu-python>

Capítulo 5

Diseño e implementación del banco de pruebas

En este capítulo se detalla el desarrollo técnico del marco experimental. Se describe la estructura interna del sistema, desde el inicio con la arquitectura general hasta la explicación de los algoritmos de extracción de huellas. Finalmente, se definen las métricas estadísticas que se utilizan para medir y comparar el rendimiento de ambos métodos forenses bajo los distintos experimentos realizados.

5.1. Arquitectura del sistema

El sistema se divide principalmente en las siguientes fases, cuyo flujo de ejecución se muestra de manera global en la Figura 5.1:

- **Fase 0 - Preprocesado:** Es una fase opcional. Dado que antes de iniciar algunos experimentos se requiere de cierto preprocesamiento, este módulo prepara los datos antes de comenzar con las fases posteriores. Es una fase que filtra las imágenes según sus metadatos para asegurar que las imágenes son de la cámara principal del dispositivo, descartando imágenes no válidas. Además, permite generar los escenarios de estrés mediante la aplicación automática de filtros o transformaciones (como la aplicación de los filtros de belleza) sobre los conjuntos de imágenes.
- **Fase 1 - Extracción de huellas individuales:** En este bloque, el sistema procesa de forma masiva las imágenes de los distintos *smartphones*. Dependiendo de la técnica seleccionada (PRNU o Noiseprint), extrae el residuo de ruido correspondiente de cada imagen y recorta la zona central exacta (1024x1024 píxeles) para estandarizar su formato. Estas huellas individuales se almacenan como matrices de datos.
- **Fase 2 - Generación de huellas maestras (entrenamiento):** Este módulo es el encargado de generar la huella maestra de referencia para cada modelo de teléfono. El sistema agrupa todas las huellas individuales extraídas en la fase 1 y las combina matemáticamente para generar una única huella maestra, limpia y representativa de ese dispositivo en específico.

- **Fases 3 y 4 - Verificación y evaluación global:** Son las fases del sistema que se encargan de la clasificación de imágenes. Permiten analizar tanto una imagen individual (fase 3) como conjuntos de datos de prueba enteros (fase 4). El programa extrae la huella de cada imagen de test al momento y la compara contra la base de datos de huellas maestras para identificar su procedencia. Durante las evaluaciones, el sistema automatiza el cálculo de resultados y exporta análisis estadísticos visuales.

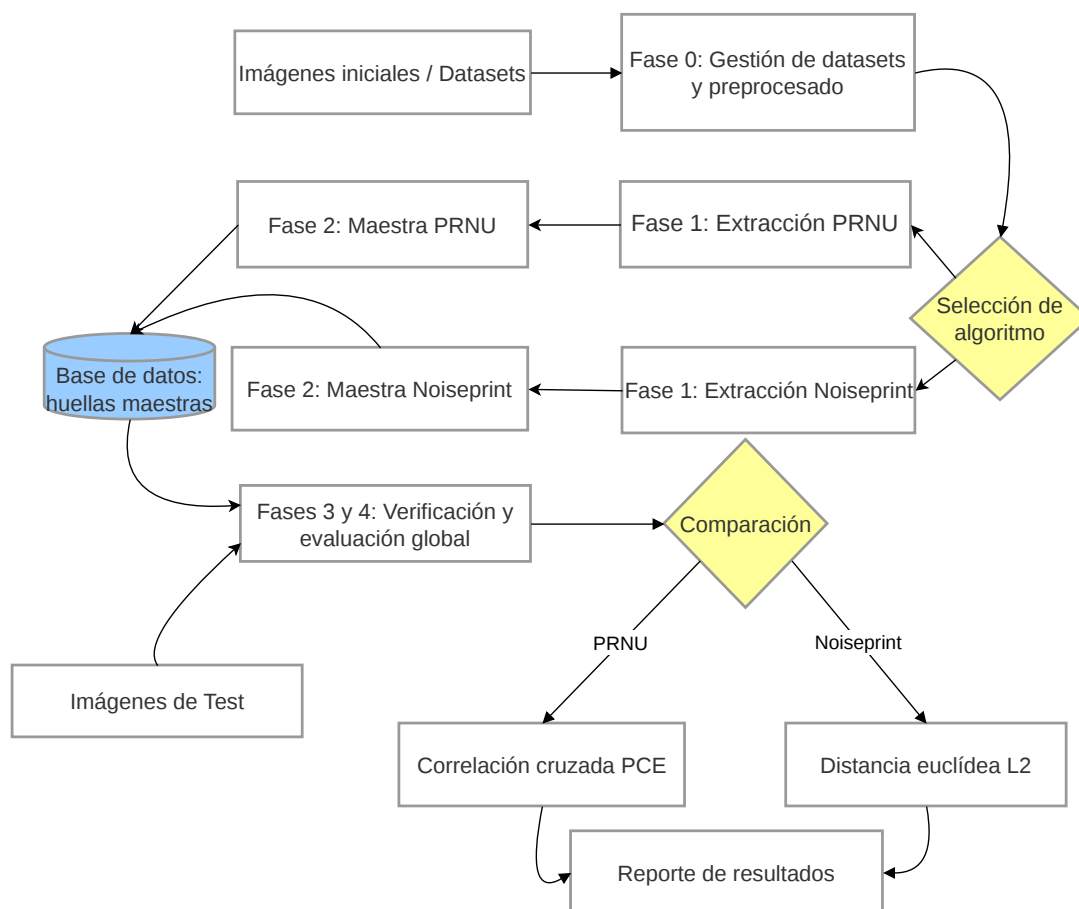


Figura 5.1: Diagrama de flujo que representa las opciones de ejecución del software forense desarrollado en este TFG.

El código fuente completo del software interactivo desarrollado, los scripts de automatización de las pruebas de estrés y los módulos de análisis estadístico se encuentran publicados bajo un repositorio de acceso público en GitHub con licencia GNU/GPLv3¹.

¹<https://github.com/Daniel870984/AnalisisImagenes.git>

5.2. Extracción y análisis de huellas forenses

Una vez definida la arquitectura general del sistema, en esta sección se detallan las funciones y procesos que permiten la identificación de los dispositivos. Recuérdese que aunque ambos métodos tienen como objetivo vincular una imagen con su cámara de origen, lo hacen enfocando el problema desde ángulos opuestos: el procesamiento de señales físicas y la visión artificial.

5.2.1. Aislamiento del ruido de patrón fijo

PRNU imita el comportamiento de una huella dactilar física, pero trasladado al sensor fotográfico. Cuando una cámara captura una foto, la luz ideal de la escena se contamina de forma inevitable por los pequeños defectos microscópicos generados durante la fabricación de la matriz de silicio. Matemáticamente, este fenómeno se describe simplificando la relación entre la luz real que entra al sensor, $I^{(0)}$, el ruido de patrón fijo único del dispositivo, K , y el resto de ruidos aleatorios del entorno Θ . El factor K multiplica porque depende de la luz: cuantos más fotones entran, más se nota la diferencia de sensibilidad del píxel. En cambio, Θ es aditivo porque engloba ruidos aleatorios que se suman al final, haya luz o no [2]:

$$I = I^{(0)} + I^{(0)}K + \Theta$$

En la Figura 5.2 se puede observar el resultado de aislar la huella K y eliminar el dibujo de la foto (como árboles o caras). El software ejecuta una cadena de limpieza de cuatro pasos sobre cada imagen de forma individual:

1. **Extracción del residuo base:** El primer paso consiste en separar la textura de la imagen del ruido de fondo. Para ello, la imagen se descompone en varios niveles usando la Transformada de Onda Discreta [2]. Un filtro adaptativo analiza estas frecuencias para suavizar los bordes marcados de los objetos y, al reconstruir la imagen a la inversa, devuelve un ruido residual limpio de contenido visual.
2. **Transformación a luminancia:** Para ahorrar recursos de procesamiento y unificar los datos, el ruido obtenido en los tres canales de color se fusiona para trabajar sobre una única capa en escala de grises.
3. **Supresión de artefactos espaciales (Zero-Mean):** El residuo obtenido todavía arrastra algunos artefactos, entre ellos el patrón lineal provocado por la compresión JPEG que no es exclusivo de la cámara de origen, sino del formato de archivo. El modelo los elimina de raíz calculando y restando la media matemática de cada fila y cada columna, dejando la matriz de ruido en un valor neutro.
4. **Filtrado en el dominio frecuencial (Wiener DFT):** Como último paso, el ruido se lleva al campo de las frecuencias mediante la Transformada de Fourier. Aquí se aplica un filtro de Wiener global para eliminar por completo cualquier patrón repetitivo que compartan los teléfonos de la misma marca debido al procesamiento interno del color, el denominado *demosaiçing*. Al volver a transformar la imagen al estado normal, lo que queda es la huella digital del sensor totalmente pura.

Para fabricar la huella maestra de un dispositivo, el sistema extrae estas huellas limpias de varias fotografías y aplica por última vez de nuevo Zero-Mean y Wiener DFT, ya que al combinar las huellas podrían amplificarse los artefactos no deseados. Finalmente, en la fase de test, el éxito del emparejamiento se mide con la métrica *Peak to Correlation Energy* (PCE), que se explicará a continuación.

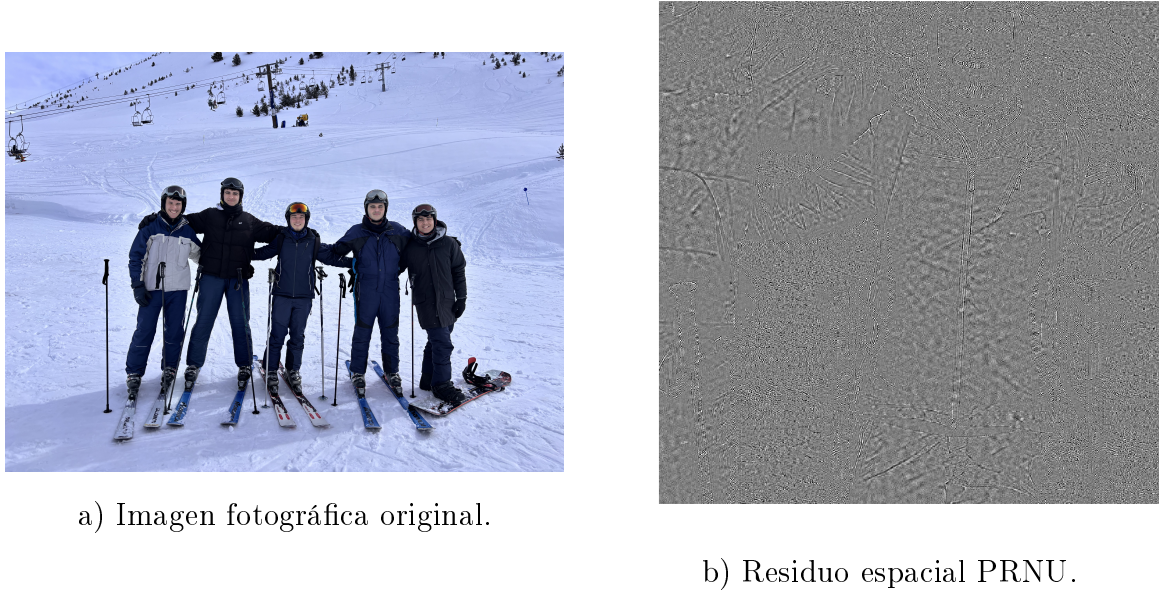


Figura 5.2: Comparativa entre una fotografía original y el residuo del recorte 1024x1024 extraído mediante las técnicas explicadas. Se puede ver cómo en el residuo aún quedan trazas semánticas (siluetas) de la imagen original.

5.2.2. Inferencia del residuo algorítmico

En la Figura 5.3 se puede observar el resultado del proceso para extraer este residuo. Está optimizado para no saturar el ordenador y se ejecuta de manera secuencial a través de los siguientes pasos:

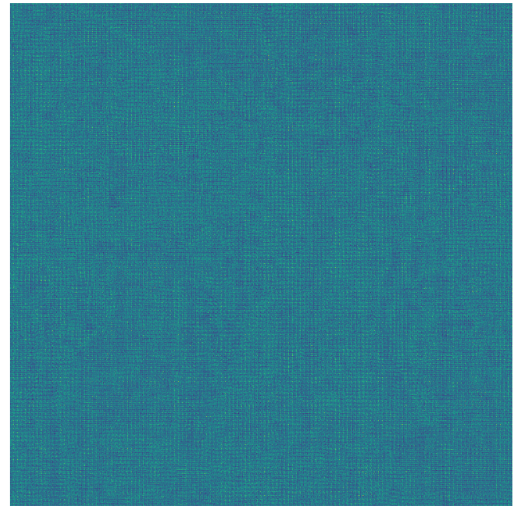
1. **Preprocesamiento y enrutamiento dinámico:** Al igual que se hace con el método clásico, la imagen de entrada se transforma primero a una única capa en escala de grises. Justo después, el sistema examina el archivo para estimar su Factor de Calidad JPEG (QF, de sus siglas en inglés, *Quality Factor*). Este paso es muy importante ya que Noiseprint no utiliza una IA genérica, el software usa este dato para elegir y utilizar la red neuronal experta que mejor sabe leer las fotos con ese QF.
2. **Inferencia por ventanas solapadas:** La extracción del ruido se consigue gracias a la CNN. Como las fotos actuales tienen una resolución enorme y esta red no tiene capas de reducción espacial, procesar la imagen entera de golpe requiere mucha potencia computacional. Para evitarlo, el modelo trocea la imagen en bloques pequeños, los analiza uno a uno de forma secuencial y luego vuelve a construir los mapas de ruido solapando los bordes para que no se noten los cortes previos.

3. **Estandarización espacial:** Una vez que la IA devuelve el mapa de ruido completo, el sistema se queda con un recorte del bloque central exacto de 1024x1024 píxeles. Esta matriz limpia se empaqueta y se almacena en formato binario comprimido, guardando al mismo tiempo la huella extraída y el QF que se utilizó para conseguirla.

Para construir la huella maestra bajo este enfoque, el sistema recolecta los patrones de varias fotos de un mismo dispositivo y calcula su promedio aritmético directo. La gran ventaja es que la propia CNN ya se encarga de desintegrar el paisaje y los objetos de la imagen, entregando características completamente limpias. Finalmente, la fase de comparación calcula la distancia euclídea entre el mapa de ruido de la foto de prueba y la huella maestra de referencia, que se explica con más detalle a continuación.



a) Imagen fotográfica original.



b) Mapa de características algorítmico.

Figura 5.3: Extracción de la huella Noiseprint mediante la CNN siamesa. El modelo elimina el contenido semántico de la escena, aislando el patrón de artefactos del ISP.

5.3. Métricas de decisión y evaluación

Para garantizar la validez científica del banco de pruebas, es necesario establecer un marco de evaluación sólido. Por un lado, están las métricas de decisión que son utilizadas por ambos algoritmos de manera interna para emparejar a una imagen con la huella maestra de un dispositivo. Por otro lado, están las métricas de evaluación de rendimiento que son utilizadas para medir la eficacia global del sistema sobre los conjuntos de datos.

5.3.1. Métricas de decisión forense

Cuando el sistema extrae el residuo de una imagen de prueba X y lo enfrenta a la huella maestra de un dispositivo Y , necesita una función de similitud matemática para tomar una decisión. Dependiendo del algoritmo utilizado, se utilizan dos métricas distintas:

Correlación cruzada y PCE (en PRNU)

Para evaluar la similitud entre las señales, el sistema desplaza espacialmente la huella de test sobre la huella maestra buscando el punto exacto de alineamiento. Este proceso genera una matriz de correlación cruzada bidimensional donde, si la imagen pertenece al dispositivo, se diferencia un pico máximo de coincidencia frente a un fondo de caos. A partir de esta matriz se calcula la métrica PCE, una relación señal-ruido que mide la energía de dicho pico respecto al resto del plano. Para garantizar la fiabilidad del cálculo, el algoritmo identifica el pico y anula matemáticamente a sus píxeles vecinos dentro de un radio determinado. De este modo, se evita que los valores altos de la zona ya alineada inflen artificialmente la media del ruido de fondo. Un valor de PCE alto indica que existe un pico muy nítido sin interferencias, lo que confirma el emparejamiento de la huella con el dispositivo. En cambio, un valor cercano a cero significa que el pico se disuelve por completo en el ruido de fondo, indicando que la foto no pertenece a esa cámara.

Distancia euclídea (en Noiseprint)

Para el método basado en aprendizaje profundo, las huellas se interpretan como puntos en un espacio dimensional de características. La similitud entre dos huellas se determina calculando la distancia euclídea (norma L_2) entre la matriz de la imagen de prueba y la huella maestra:

$$d(X, Y) = \|X - Y\|_2 = \sqrt{\sum_i \sum_j (X_{i,j} - Y_{i,j})^2}$$

En este caso, la métrica funciona al revés que el PCE: el clasificador asigna la imagen al dispositivo que minimice la distancia entre la huella de la imagen y las huellas maestras de cada dispositivo.

5.3.2. Métricas de evaluación de rendimiento

Para cuantificar el éxito del software al clasificar conjuntos de datos enteros bajo distintos escenarios de estrés (como la compresión de WhatsApp o filtros fotográficos), el banco de pruebas genera automáticamente reportes estadísticos basados en tres herramientas de evaluación global:

Matriz de confusión

Permite observar el desempeño del algoritmo frente a múltiples clases. Las filas representan la verdadera clase (en este caso, el dispositivo real de donde proviene la foto), mientras que las columnas representan la predicción del modelo. Los aciertos se agrupan en la diagonal principal, permitiendo identificar rápidamente si un dispositivo concreto está siendo bien clasificado por el modelo.

Exactitud (*Accuracy*)

Es la proporción de predicciones correctas sobre el total de casos evaluados, siendo TP las imágenes que pertenecen al dispositivo objetivo y son correctamente clasificadas con su origen, TN las imágenes que no pertenecen al dispositivo objetivo y que

se clasifican como de otros dispositivos, FP las imágenes que son clasificadas con el dispositivo objetivo cuando no lo son y FN las imágenes que no son clasificadas con el dispositivo objetivo cuando realmente sí lo son:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Aunque es una métrica intuitiva, puede resultar engañosa si el conjunto de datos está desbalanceado.

Coeficiente de correlación de Matthews (MCC)

Se utiliza también el coeficiente de Matthews para tratar de resolver las limitaciones de la exactitud. Esta métrica tiene en cuenta todas las celdas de la matriz de confusión y produce un valor alto solo si la predicción obtuvo buenos resultados en todas las categorías, siendo más robusta con distribuciones desbalanceadas. Se define como:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

El valor del MCC pertenece al intervalo $[-1,1]$, donde 1 representa una predicción perfecta, 0 dice que el modelo no es mejor que una predicción aleatoria, y -1 indica un desacuerdo total entre la predicción y la realidad.

Métricas de apoyo por clase

Además de las métricas globales de rendimiento, el reporte estadístico generado por el banco de pruebas también calcula una serie de métricas específicas para cada dispositivo evaluado. Las métricas de apoyo calculadas para cada clase son las siguientes:

Precisión (*Precision*): Evalúa la fiabilidad del sistema cuando predice que una imagen pertenece a un dispositivo concreto. Matemáticamente, es la proporción de identificaciones positivas correctas frente al total de identificaciones positivas (tanto correctas como incorrectas) realizadas por el sistema para esa clase:

$$Precision = \frac{TP}{TP + FP}$$

Exhaustividad (*Recall* o sensibilidad): Mide la capacidad del sistema para encontrar y clasificar correctamente todas las imágenes reales de un dispositivo específico. Es la proporción de verdaderos positivos frente al total de positivos reales:

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

Puntuación F1 (*F1-Score*): Normalmente en los sistemas de clasificación existe un compromiso entre maximizar la precisión (minimizar falsos positivos) y maximizar la exhaustividad (minimizar falsos negativos). La puntuación F1 proporciona una métrica de consenso y se calcula mediante la media armónica de ambas métricas. De esta manera, se penaliza a los valores extremos. Se define concretamente:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Soporte (*Support*): Esta métrica no evalúa en sí el algoritmo, sino que proporciona el contexto para que se pueda interpretar las métricas anteriores. Es el número absoluto de ocurrencias reales de una clase específica dentro del conjunto de datos de prueba evaluado. Un valor de *Accuracy* o precisión del 100 % no tiene valor estadístico si el soporte es insuficiente (por ejemplo, evaluar una única imagen).

Capítulo 6

Experimentos y resultados

En este capítulo se detalla el diseño de los experimentos realizados y se analizan los resultados obtenidos tras someter ambos modelos forenses a diferentes pruebas de estrés. Antes de evaluar la robustez de los métodos ante plataformas de mensajería y edición que degradan la señal, se analiza el comportamiento de ambos enfoques en condiciones más estables para identificar una línea base de rendimiento.

6.1. Análisis preliminar y comportamiento base

Para cada uno de los dispositivos analizados en este proyecto se recopila un volumen de 100 fotografías, ya sea procedente del conjunto de datos de CSAFE¹ o recopiladas a mano por el autor. Este conjunto se divide en dos bloques de 50 imágenes cada uno: el primer bloque se utiliza exclusivamente para el cálculo y la estimación de la huella maestra de la cámara, mientras que las 50 imágenes restantes se reservan como imágenes de test individuales para evaluar el porcentaje de acierto y la fiabilidad de las métricas de decisión bajo los diferentes escenarios de degradación. Cabe destacar que existen excepciones con conjuntos de 49 imágenes en lugar de 50, debido a que algunas de ellas están corruptas.

Con esta estructura de datos validada, se lleva a cabo una fase de experimentación preliminar cuyo objetivo es comprender el comportamiento nativo de ambos métodos:

1. **Fase inicial de pruebas e incorporación progresiva de dispositivos:** Las primeras pruebas del sistema se realizan utilizando datos entre dispositivos con arquitecturas de procesamiento distintas, utilizando imágenes de un teléfono iPhone 14 frente a un teléfono Samsung S21. Una vez se valida la correcta extracción y el aislamiento de los residuos en este escenario inicial con tan solo dos dispositivos, se procede a incorporar de forma progresiva nuevos dispositivos al banco de pruebas, entre ellos un iPhone 15 y un Samsung Note 10. Esta expansión progresiva permite evaluar cómo escalan las métricas de correlación a medida que aumenta la complejidad y el volumen del ecosistema de dispositivos analizados.
2. **Evaluación con imágenes de control y adaptación metodológica:** A continuación, se ejecuta una prueba utilizando imágenes homogéneas de control (*flatfields*), que son imágenes de superficies uniformes. En estas condiciones de

¹Disponible en: <https://data.csafe.iastate.edu/Multi-Camid/>

laboratorio, no hay contenido semántico que pueda contaminar el residuo, lo que permite que la matriz de correlación de PRNU devuelva un pico de coincidencia muy limpio y sin interferencias. Hay que mencionar una adaptación en el TFG respecto a la literatura clásica: aunque los autores originales recomiendan el uso de *flatfields* para el cálculo de la huella maestra, al tener que incorporar dispositivos de uso propio y simular un escenario forense más realista, se decide generar las huellas utilizando fotografías naturales de escenas cotidianas (como un edificio, una mesa o un árbol), evaluando así la robustez del sistema bajo condiciones de adquisición reales.

6.2. Impacto del postprocesado

Una vez establecida la metodología de análisis preliminar, el primer escenario de estrés mide el impacto del postprocesado nativo de los dispositivos sobre las imágenes extraídas directamente del almacenamiento interno, sin aplicar ninguna degradación o compresión externa adicional. Este experimento actúa como grupo de control del proyecto y su objetivo es cuantificar cuánta señal de ruido se pierde debido únicamente a los algoritmos de optimización automáticos que integran los fabricantes en el software de la cámara. En la Figura 6.1 se detallan las matrices de confusión y en la Tabla 6.1 se detallan las métricas obtenidas.

6.2.1. Análisis y rendimiento del método PRNU

En primer lugar, se analiza el comportamiento del enfoque clásico basado en la extracción del ruido digital del sensor.

- **Ecosistema Apple (iPhone 14 y iPhone 15):** El sistema demuestra un buen rendimiento al analizar el iPhone 14, logrando una clasificación prácticamente perfecta; sin embargo, al enfrentarse al iPhone 15 baja por completo. Una hipótesis muy probable para justificar este hecho es la evolución del hardware de captura. Mientras que el iPhone 14 incorpora el procesamiento Smart HDR 4, el iPhone 15 integra la versión posterior, Smart HDR 5. Este cambio en la generación podría implicar algoritmos de fotografía computacional más complejos, lo que dificultaría la consistencia de los residuos de ruido analizados en este modelo más reciente.
- **Ecosistema Samsung (Note 10 y S21):** En los Samsung se aprecian dos comportamientos opuestos que validan la hipótesis del proyecto. Por un lado, el Samsung Note 10 logra mantener una alta tasa de identificación, teniendo en cuenta que se toma como acierto la clasificación de una imagen del Note 10 si se empareja con la huella maestra natural del Note 10 o con la *flatfield* ya que son el mismo dispositivo. Por otro lado, PRNU es incapaz de emparejar una imagen del S21 correctamente, clasificándolas aleatoriamente. Esto demuestra la vulnerabilidad del método convencional ante los estándares modernos del software fotográfico. Cabe destacar que este dispositivo implementa tecnologías como *pixel binning*, que consiste en agrupar conjuntos de 4 o hasta 9 píxeles adyacentes del sensor de la cámara para que actúen como un solo píxel más grande.

6.2.2. Análisis y rendimiento del método Noiseprint

En segundo lugar, se evalúa el rendimiento del algoritmo basado en aprendizaje profundo para la extracción de firmas de ruido de la cámara. A partir de estos experimentos se extraen estas conclusiones:

- **Ecosistema Apple (iPhone 14 y iPhone 15):** El rendimiento roza la perfección. El modelo logra 50 aciertos para el iPhone 14 con un $F1$ -score de 0.99. Por su parte, iPhone 15 se clasifica correctamente en 45 de las 50 muestras logrando un $F1$ -score de 0.96.
- **Ecosistema Samsung (Note 10 y S21):** Se observa un comportamiento muy favorable, aunque con ligeros solapamientos propios de la marca. El Note 10 acierta 33, sumando las 24 que clasifica correctamente más las 9 que clasifica como *flatfield*, y desvía 17 hacia el S21. Esto puede ser debido a que al ser móviles de la misma marca comparten software y patrones de compresión. Para el Samsung S21 se consigue clasificar correctamente 40 de 49 imágenes, logrando un $F1$ -score de 0.75.

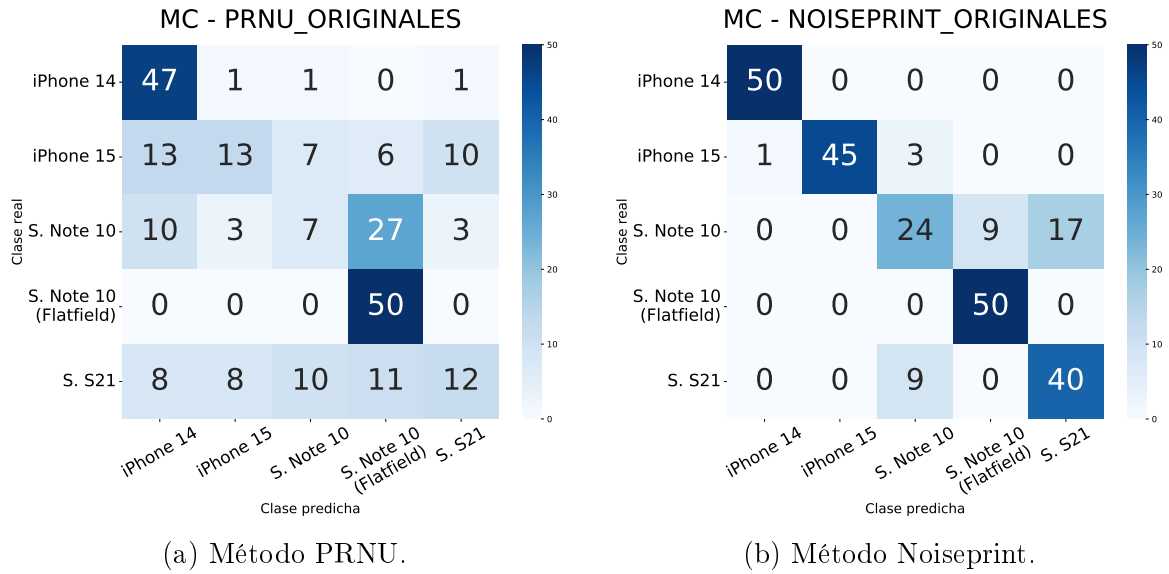


Figura 6.1: Matrices de confusión ante el escenario de postprocesado nativo.

Tabla 6.1: Métricas de rendimiento comparativas (PRNU / Noiseprint) para el escenario de postprocesado nativo.

Clase dispositivo	Precision	Recall	F1-Score	FP	FN	Support
iPhone 14	0.60 / 0.98	0.94 / 1.00	0.73 / 0.99	31 / 1	3 / 0	50 / 50
iPhone 15	0.52 / 1.00	0.27 / 0.92	0.35 / 0.96	12 / 0	36 / 4	49 / 49
S. Note 10	0.28 / 0.67	0.14 / 0.48	0.19 / 0.56	18 / 12	43 / 26	50 / 50
S. Note 10 (Flatfield)	0.53 / 0.85	1.00 / 1.00	0.69 / 0.92	44 / 9	0 / 0	50 / 50
S. S21	0.46 / 0.70	0.24 / 0.82	0.32 / 0.75	14 / 17	37 / 9	49 / 49
Accuracy global	0.5202 / 0.8427					
MCC	0.4191 / 0.8064					

6.3. Impacto de redes sociales

Este escenario experimental evalúa la robustez de los métodos forenses al someterlos a las plataformas más comunes en el entorno digital actual: las aplicaciones de mensajería y las redes sociales [3]. Como se ha descrito anteriormente, dichas plataformas alteran los archivos mediante algoritmos de compresión con pérdida y procesos de reescalado geométrico para optimizar el almacenamiento y el ancho de banda.

Para analizar este impacto, el experimento se divide en dos subescenarios independientes: En el caso de WhatsApp, se evalúa el impacto del canal de distribución y su compresión asociada sobre imágenes capturadas previamente de forma nativa. Por otro lado, en el subescenario de Instagram, el enfoque se centra en el efecto de la adquisición directa a través de la propia interfaz de la aplicación.

6.3.1. Impacto de WhatsApp

El objetivo es comprobar si los algoritmos son capaces de recuperar los residuos restantes tras los procesos que aplica WhatsApp al compartir una imagen por un chat.

A continuación, se presentan los resultados obtenidos tras cruzar estas muestras degradadas frente a las huellas maestras de referencia. Se detallan las métricas en la Tabla 6.2, mientras que las matrices de confusión se muestran en la Figura 6.2.

Tabla 6.2: Métricas de rendimiento comparativas (PRNU / Noiseprint) para el escenario de la compresión de WhatsApp.

Clase dispositivo	Precision	Recall	F1-Score	FP	FN	Support
iPhone 14	0.27 / 0.00	0.24 / 0.00	0.26 / 0.00	32 / 0	38 / 50	50 / 50
iPhone 15	0.16 / 0.00	0.20 / 0.00	0.18 / 0.00	51 / 0	39 / 49	49 / 49
S. Note 10	0.15 / 0.22	0.14 / 0.40	0.15 / 0.29	39 / 69	43 / 30	50 / 50
S. Note 10 (Flatfield)	0.33 / 0.00	0.38 / 0.00	0.36 / 0.00	38 / 1	31 / 50	50 / 50
S. S21	0.15 / 0.28	0.12 / 0.92	0.13 / 0.43	34 / 113	43 / 4	49 / 49
Accuracy global	0.2177 / 0.2621					
MCC	0.0223 / 0.1033					

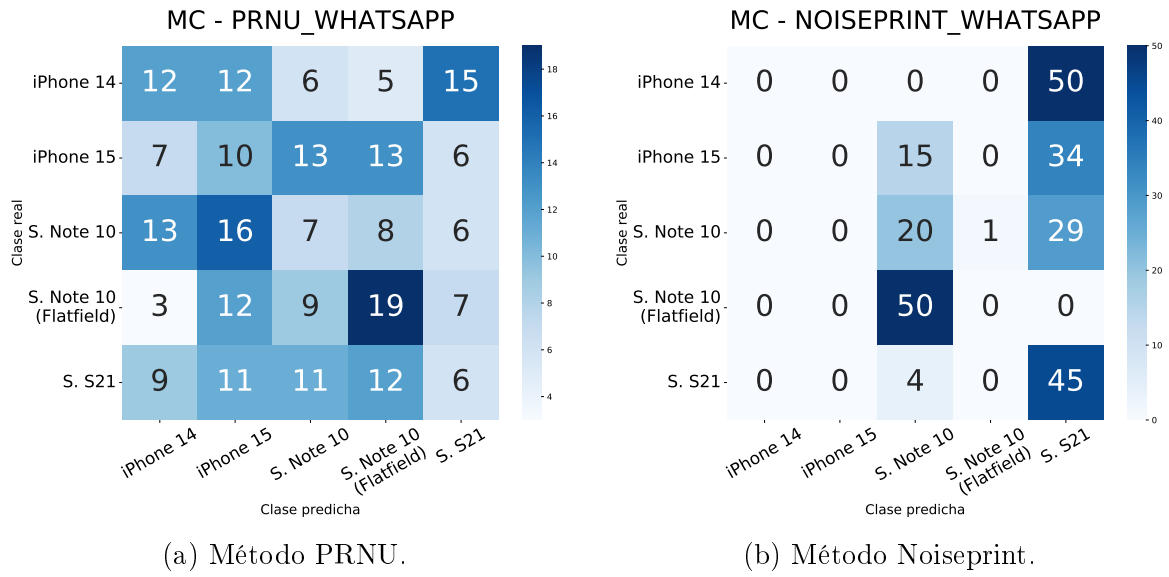


Figura 6.2: Matrices de confusión ante la compresión de WhatsApp.

Análisis y rendimiento del método PRNU

Se evidencia el colapso absoluto del método clásico ante este escenario. El algoritmo muestra un comportamiento aleatorio, registrando valores de PCE residuales y tasas de acierto prácticamente nulas en todo el ecosistema de dispositivos. Esto se debe a que la cuantización del formato JPEG y el reescalado que aplica WhatsApp eliminan por completo los componentes de alta frecuencia espacial donde se encuentra el ruido físico del sensor, borrando así la huella digital de la cámara.

Análisis y rendimiento del método Noiseprint

En el caso de Noiseprint, los resultados muestran una dinámica relacionada con la propia naturaleza de su arquitectura basada en aprendizaje profundo. Aunque el algoritmo parece que logra clasificar correctamente dos de los dispositivos, el análisis detallado revela que este comportamiento no responde a una identificación correcta del sensor individual, sino a las huellas que introduce el software. Al estar entrenada para reconocer firmas de procesado, la red neuronal interpreta que el patrón regular de compresión que utiliza WhatsApp guarda una similitud estadística mucho mayor con los algoritmos de postprocesado de Samsung que con la filosofía de renderizado de Apple, sesgando así las predicciones hacia esos dos dispositivos.

6.3.2. Impacto de Instagram

En este caso, las fotografías son tomadas directamente desde la cámara integrada dentro de la aplicación de Instagram y, posteriormente, almacenadas en el dispositivo. Dado que es necesario disponer del dispositivo en el momento para guardar la imagen, este test se ha realizado exclusivamente con los *smartphones* del autor: un iPhone 14 y un iPhone 15. Los resultados expuestos en la Figura 6.3 y la Tabla 6.3 confirman un comportamiento deficiente en ambos enfoques, el cual es muy similar al colapso observado previamente en el test de WhatsApp.

Análisis y rendimiento del método PRNU

El algoritmo clásico vuelve a operar en niveles de aleatoriedad estadística, registrando un *Accuracy* global de 0.54 y un coeficiente MCC residual de 0.0810. Del mismo modo que en el test anterior, se destruyen todos los componentes de alta frecuencia que caracterizan al sensor de la cámara.

Análisis y rendimiento del método Noiseprint

Al enfoque basado en aprendizaje profundo le ocurre lo mismo que en el test de WhatsApp. Debido a su naturaleza, trata de encontrar una firma de software y por casualidad se parece más a la del iPhone 14 que a la del iPhone 15. Esto provoca que Noiseprint clasifique automáticamente toda la muestra a un único patrón de procesado. Se puede observar claramente con el *Accuracy* global de 0.49 y un coeficiente MCC incluso de -0.0586.

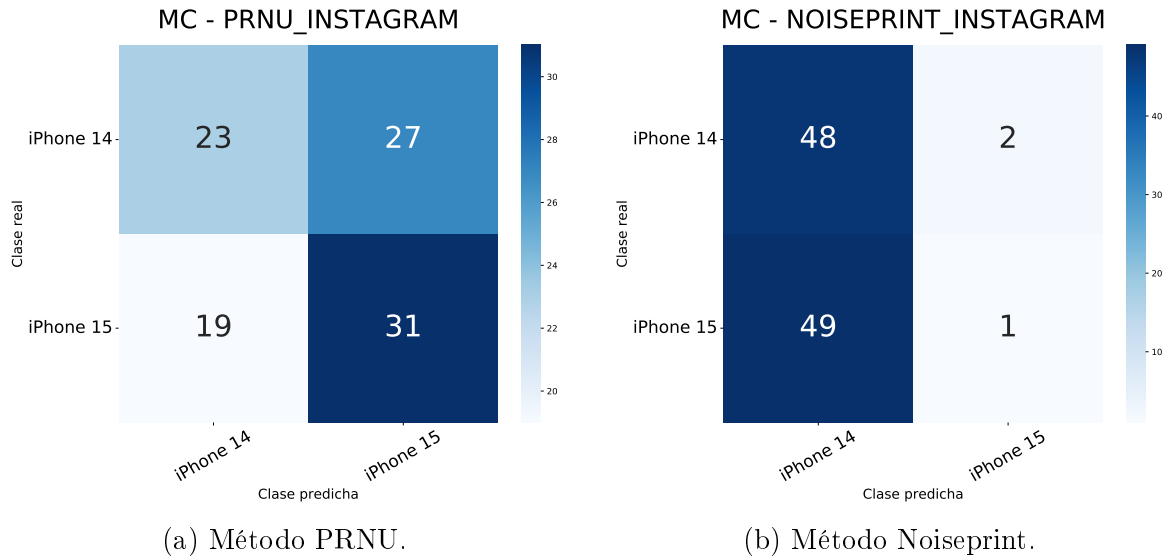


Figura 6.3: Matrices de confusión ante la compresión de Instagram.

Tabla 6.3: Métricas de rendimiento comparativas (PRNU / Noiseprint) para el escenario de la compresión de Instagram.

Clase dispositivo	Precision	Recall	F1-Score	FP	FN	Support
iPhone 14	0.55 / 0.49	0.46 / 0.96	0.50 / 0.65	19 / 49	27 / 2	50 / 50
iPhone 15	0.53 / 0.33	0.62 / 0.02	0.57 / 0.04	27 / 2	19 / 49	50 / 50
Accuracy global	0.5400 / 0.4900					
MCC	0.0810 / -0.0586					

6.4. Impacto de los filtros

Tras analizar el comportamiento de los algoritmos ante los anteriores experimentos, estas últimas pruebas se basan en la manipulación directa de la imagen. El objetivo

es evaluar la robustez de ambas metodologías forenses frente a procesos de edición fotográfica. Para ello, se diseñan dos escenarios de estrés basados en naturalezas matemáticas opuestas: una transformación lineal puntual y un operador no lineal espacial, en el cual los valores de sus píxeles dependen también de sus vecinos, a diferencia del filtro lineal, en el que el valor de cada píxel depende exclusivamente del valor de ese mismo píxel.

A continuación, se detallan los resultados y el análisis correspondiente a cada uno de estos filtros de edición.

6.4.1. Filtro lineal

Este escenario de edición evalúa la tolerancia y los límites de supervivencia de los algoritmos forenses ante modificaciones puntuales que alteran el rango dinámico global de la imagen sin distorsionar la geometría de la escena. El filtro implementado aplica una transformación matemática lineal sobre cada píxel de forma independiente, respondiendo a la ecuación $y = \max(0, \min(255, \alpha \cdot (x - 128) + 128 + \beta))$, lo que garantiza que los píxeles con valor por encima de 128 se vuelvan más claros mientras que los píxeles con valor por debajo de 128 se vuelvan más oscuros. Con el propósito de poner al límite los algoritmos se diseñan dos pruebas de estrés con niveles de agresividad distintos: un primer nivel de estrés leve ($\alpha = 1.2, \beta = 30$) y un segundo nivel de estrés severo ($\alpha = 1.4, \beta = 80$). Para entender estos valores, se puede observar el resultado de las imágenes en la Figura 6.4.



Figura 6.4: Muestras comparativas del impacto del brillo y contraste en los distintos escenarios.

Análisis y rendimiento del método PRNU:

Se demuestra una resistencia ante ambos escenarios de estrés. Tal y como se observa en la Figura 6.5 se puede ver que, con el estrés leve, casi no ha cambiado su comportamiento respecto a las imágenes originales de la Figura 6.1. Al forzar al método PRNU al máximo nivel de estrés, se demuestra que sigue siendo bastante resistente ya que solo falla al clasificar unas pocas imágenes más. Las métricas se detallan en la Tabla 6.4.

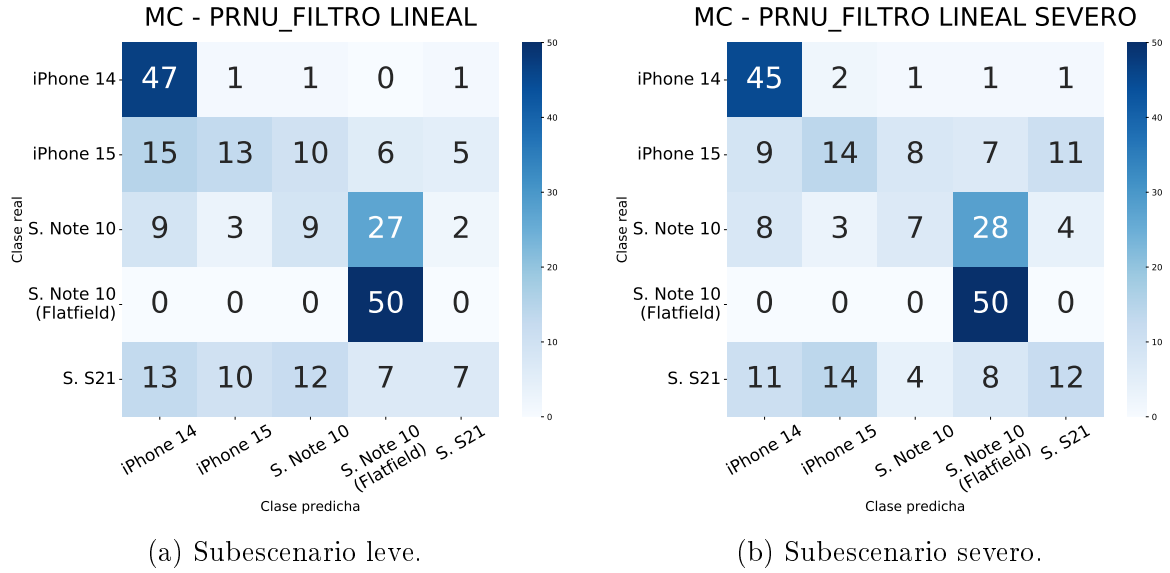


Figura 6.5: Matrices de confusión del método PRNU ante el escenario del filtro lineal.

Tabla 6.4: Métricas de rendimiento comparativas del método PRNU según la intensidad del filtro lineal aplicado (filtro leve / filtro severo).

Clase dispositivo	Precision	Recall	F1-Score	FP	FN	Support
iPhone 14	0.56 / 0.62	0.94 / 0.90	0.70 / 0.73	37 / 28	3 / 5	50 / 50
iPhone 15	0.48 / 0.42	0.27 / 0.29	0.34 / 0.34	14 / 19	36 / 35	49 / 49
S. Note 10	0.28 / 0.35	0.18 / 0.14	0.22 / 0.20	23 / 13	41 / 43	50 / 50
S. Note 10 (Flatfield)	0.56 / 0.53	1.00 / 1.00	0.71 / 0.69	40 / 44	0 / 0	50 / 50
S. S21	0.47 / 0.43	0.14 / 0.24	0.22 / 0.31	8 / 16	42 / 37	49 / 49
Accuracy global	0.5081 / 0.5161					
MCC	0.4043 / 0.4121					

Análisis y rendimiento del método Noiseprint:

El enfoque basado en aprendizaje profundo demuestra una resiliencia inicial que se degrada conforme se enfrenta a un mayor nivel de estrés: En la Figura 6.6 se pueden observar las matrices de confusión de ambos niveles de estrés. En la Tabla 6.5 se observa como el algoritmo alcanza un *Accuracy* global de 0.6895 y un MCC de 0.6392, manteniendo una clasificación excelente en la diagonal principal con plena fiabilidad en los iPhones (F1-Scores de 0.97 y 0.94) y en el Samsung S21 (F1-Score de 0.82). Sin embargo, al elevar el filtro al máximo nivel, el modelo alcanza su límite operativo y la exactitud global baja hasta 0.5565 (MCC de 0.4741). Curiosamente, se equivoca al clasificar las imágenes *flatfields*, esto se puede deber a que al modificar el brillo y contraste de imágenes tan planas, el modelo las confunde con la huella más natural del Samsung Note 10.

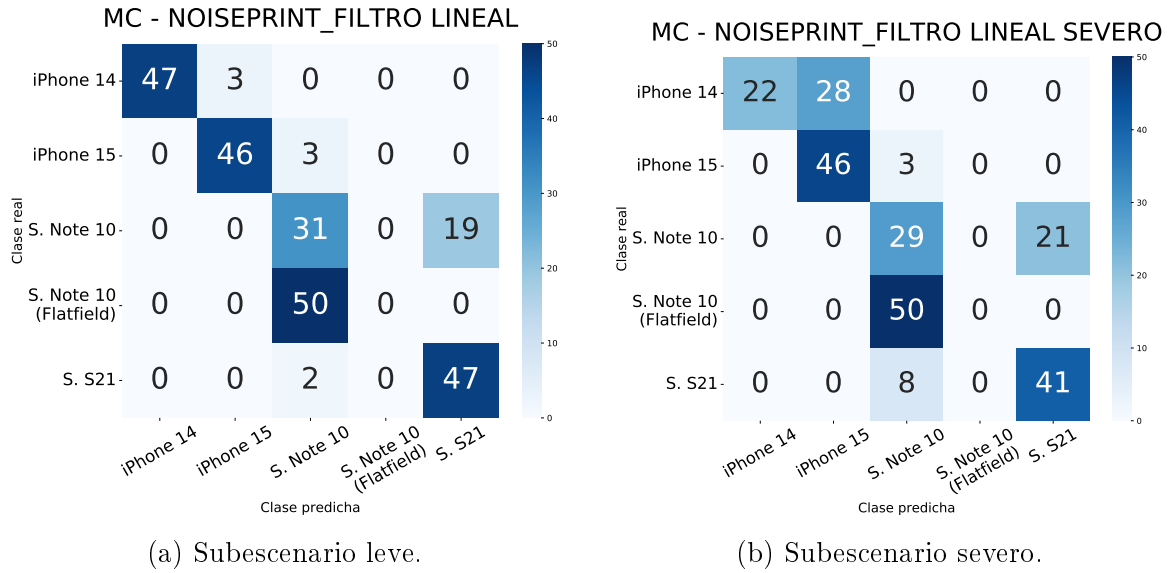


Figura 6.6: Matrices de confusión del método Noiseprint ante el escenario del filtro lineal.

Tabla 6.5: Métricas de rendimiento comparativas del método Noiseprint según la intensidad del filtro lineal aplicado (filtro leve / filtro severo).

Clase dispositivo	Precision	Recall	F1-Score	FP	FN	Support
iPhone 14	1.00 / 1.00	0.94 / 0.44	0.97 / 0.61	0 / 0	3 / 28	50 / 50
iPhone 15	0.94 / 0.62	0.94 / 0.94	0.94 / 0.75	3 / 28	3 / 3	49 / 49
S. Note 10	0.36 / 0.32	0.62 / 0.58	0.46 / 0.41	55 / 61	19 / 21	50 / 50
S. Note 10 (Flatfield)	0.00 / 0.00	0.00 / 0.00	0.00 / 0.00	0 / 0	50 / 50	50 / 50
S. S21	0.71 / 0.66	0.96 / 0.84	0.82 / 0.74	19 / 21	2 / 8	49 / 49
Accuracy global	0.6895 / 0.5565					
MCC	0.6392 / 0.4741					

6.4.2. Filtro espacial

El segundo escenario de edición simula los efectos comúnmente llamados “filtros de belleza”. A diferencia del operador lineal anterior, este filtro se basa en un procesamiento espacial no lineal donde el valor final de cada píxel depende del contenido de su entorno local. Para su implementación, se combina un filtro bilateral encargado de suavizar las regiones similares respetando las líneas de contorno nítidas para que la imagen no parezca simplemente borrosa, junto con un algoritmo de ecualización adaptativa del contraste limitado por histograma (CLAHE) para que las zonas suavizadas queden de forma natural. Este experimento busca cuantificar el límite de supervivencia de ambas tecnologías ante estas modificaciones espaciales.

A continuación, se expone en la Figura 6.7 las matrices de confusión y se detallan las métricas resultantes en la Tabla 6.6 tras someter el banco de imágenes al proceso de suavizado adaptativo y retoque local.

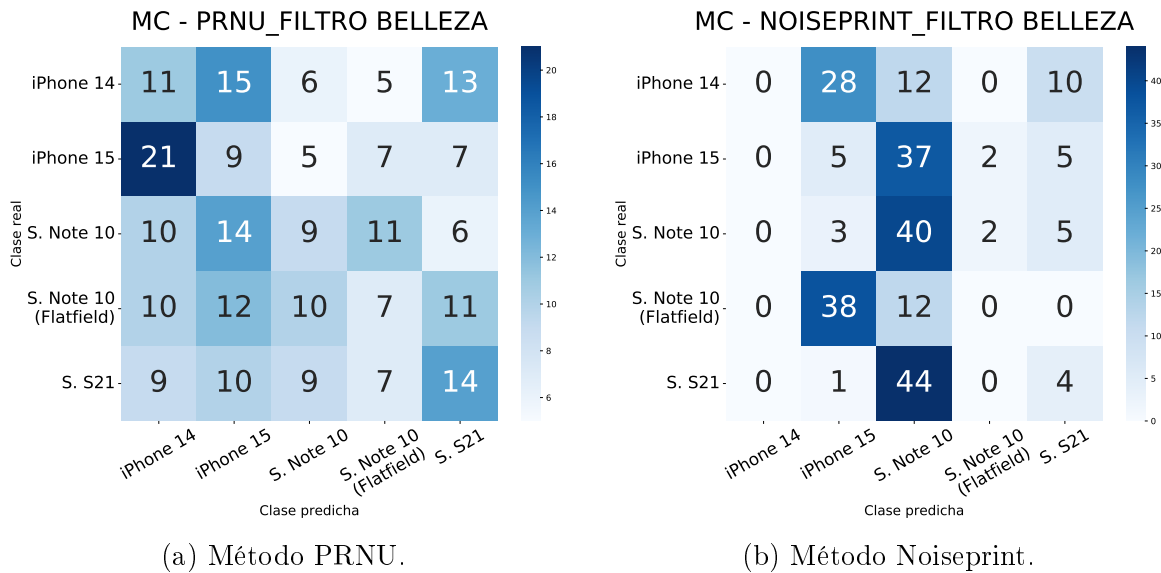


Figura 6.7: Matrices de confusión ante la compresión de Instagram.

Tabla 6.6: Métricas de rendimiento comparativas (PRNU / Noiseprint) para el escenario del filtro de belleza.

Clase dispositivo	Precision	Recall	F1-Score	FP	FN	Support
iPhone 14	0.18 / 0.00	0.22 / 0.00	0.20 / 0.00	50 / 0	39 / 50	50 / 50
iPhone 15	0.15 / 0.07	0.18 / 0.10	0.17 / 0.08	51 / 70	40 / 44	49 / 49
S. Note 10	0.23 / 0.28	0.18 / 0.80	0.20 / 0.41	30 / 105	41 / 10	50 / 50
S. Note 10 (Flatfield)	0.19 / 0.00	0.14 / 0.00	0.16 / 0.00	30 / 4	43 / 50	50 / 50
S. S21	0.27 / 0.17	0.29 / 0.08	0.28 / 0.11	37 / 20	35 / 45	49 / 49
Accuracy global	0.2016 / 0.1976					
MCC	0.0023 / -0.0036					

Análisis y rendimiento del método PRNU

Debido a que el filtro bilateral actúa como un operador que promedia los píxeles en regiones homogéneas, la señal microscópica de alta frecuencia del sensor queda eliminada. Al destruirse el residuo físico, las métricas de correlación caen significativamente, el algoritmo colapsa por completo de nuevo y opera bajo niveles de aleatoriedad estadística pura, registrando un *Accuracy* global crítico de 0.2016 y un MCC de 0.0023.

Análisis y rendimiento del método Noiseprint

Se puede observar que se registra un problema similar, con un *Accuracy* de 0.1976 y un MCC de -0.0036. Se vuelve a mostrar el mismo patrón que en los experimentos anteriores: se decanta por uno o dos dispositivos debido a su naturaleza de funcionamiento (en este caso, el Samsung Note 10 y el iPhone 15).

6.5. Comparativa de robustez

Esta sección sintetiza el rendimiento global de los escenarios de estrés analizados.

- **Resiliencia ante el postprocesado nativo:** El escenario base expone la primera gran diferencia. Mientras que el método Noiseprint demuestra solidez en la identificación (alcanzando un *Accuracy* global del 0.8427 y un MCC de 0.8064), el algoritmo PRNU muestra una degradación parcial crítica (0.5202 de exactitud y un MCC de 0.4191). Estos resultados sugieren que las técnicas de fotografía computacional implementadas en los *smartphones* modernos, tales como el *Pixel Binning* o el *Smart HDR*, introducen transformaciones no lineales que enmascaran las imperfecciones físicas de alta frecuencia del sensor analizadas por PRNU. Por el contrario, los enfoques basados en IA muestran una mayor resiliencia ante estos procesados al extraer características locales ligadas al ISP.
- **Vulnerabilidad ante compresión en redes sociales:** Al introducir el factor de las plataformas de terceros (WhatsApp e Instagram), se evidencia un colapso completo en ambos algoritmos, confirmando que las redes sociales son el entorno más hostil para la informática forense de imágenes actual. Sin embargo, los motivos del fallo son de naturalezas opuestas. Por un lado, PRNU sufre una degradación total por eliminación de señal debido a la cuantización JPEG agresiva. Por otro lado, el método Noiseprint experimenta una degradación por sesgo algorítmico al destruirse gran parte del ruido de software original. En este caso, la red es engañada por los artefactos de compresión estandarizados introducidos por las propias aplicaciones (el servidor de WhatsApp o la interfaz de captura de Instagram), perdiendo así, la capacidad de discriminación entre modelos.
- **Sensibilidad frente a la manipulación directa (filtros):**
 - La evaluación de transformaciones lineales muestra que mientras la IA sufre una caída debido a la alteración de las texturas de software, el método clásico tiene un mejor rendimiento en términos de consistencia. Al elevar el filtro al rango agresivo, Noiseprint colapsa y confunde los modelos de una

misma marca con más frecuencia que en condiciones estables. Por el contrario, PRNU demuestra un rendimiento casi idéntico en ambos escenarios de estrés. Esto prueba que la huella física y geométrica del silicio permanece invariable ante modificaciones lumínicas puntuales frente a la volatilidad del aprendizaje profundo ante este tipo de alteraciones.

- Las transformaciones espaciales no lineales (filtro de belleza) provocan un desajuste irreversible al combinar el planchado de texturas mediante operadores bilaterales con la inyección de patrones artificiales mediante CLAHE, anulando la eficacia de ambos algoritmos por completo.

Capítulo 7

Conclusiones y trabajo futuro

El propósito fundamental de este TFG ha sido evaluar la robustez de los dos tipos de metodologías forenses predominantes (PRNU y Noiseprint) en la detección del origen de imágenes frente al ecosistema computacional y social de hoy en día. Para ello, se ha desarrollado un flujo modular e interactivo en Python que unifica los flujos de extracción físicos y algorítmicos. Asimismo, se ha construido un conjunto de datos combinando la base de datos pública CSAFE con capturas nativas de dispositivos propios del autor, simulando el ciclo de vida real de las imágenes mediante su exposición a WhatsApp, Instagram y diversos filtros de edición. La posterior evaluación ha permitido delimitar e identificar los fallos de ambos métodos, aportando evidencia sobre nuevos retos de la disciplina forense.

Partiendo de las limitaciones y los comportamientos observados a lo largo de este proyecto, se plantean diversas vías de investigación para la continuidad y expansión de este trabajo. En primer lugar, visto que los clasificadores colapsan cuando las imágenes sufren compresiones o ediciones, se propone evaluar la generación de huellas maestras expuestas a esos mismos escenarios de estrés. Esto implica calcular los patrones de referencia de los dispositivos utilizando fotografías previamente degradadas; es decir, analizar de forma empírica si el sistema recupera su precisión y es capaz de discriminar, por ejemplo, un iPhone 14 de un iPhone 15 cuando ambos se evalúan bajo el mismo dominio de compresión de WhatsApp o bajo el mismo tipo de filtro. Por otra parte, conviene extender la evaluación y el volumen del conjunto de datos explorando el impacto de una variedad más amplia de filtros así como analizar plataformas de mensajería alternativas como Telegram, Signal o las nuevas modalidades de transferencia en alta definición de las aplicaciones tradicionales. Por último, con el auge que está teniendo la IA generativa, otra vía sería añadir módulos de clasificación ciega para detectar imágenes creadas de forma sintética, ya sea con modelos de difusión, redes GAN (de sus siglas en inglés, *Generative Adversarial Networks*) o VAE (de sus siglas en inglés, *Variational AutoEncoders*). La idea aquí sería estudiar tanto la detección de imágenes generadas completamente desde cero, como el impacto en escenarios híbridos donde una IA modifica una fotografía real, analizando si los patrones de ruido que introducen estos motores generativos acaban interfiriendo o anulando los residuos de las cámaras físicas.

Declaración de uso responsable de IA

El autor declara que se ha utilizado la herramienta de IA generativa Gemini única y exclusivamente como apoyo para la optimización de la redacción, la depuración de código y la consulta de documentación técnica asociada de las distintas bibliotecas utilizadas. Toda información o fragmento de código sugerido por la IA ha sido revisado, verificado y validado por el autor antes de su inclusión. El uso de esta tecnología se ha limitado a tareas de asistencia complementaria, por lo que el desarrollo intelectual del proyecto, el análisis de las métricas, las conclusiones obtenidas y la responsabilidad íntegra sobre todo el contenido de la memoria recaen de forma exclusiva sobre el autor del trabajo.

Bibliografía

- [1] Mauricio Delbracio, Damien Kelly, Michael S. Brown, and Peyman Milanfar. «Mobile Computational Photography: A Tour». In: *Annual Review of Vision Science* 7.1 (2021), pp. 571–604.
- [2] Mo Chen, Jessica Fridrich, Miroslav Goljan, and Jan Lukas. «Determining Image Origin and Integrity Using Sensor Noise». In: *IEEE Transactions on Information Forensics and Security* 3.1 (2008), pp. 74–90.
- [3] Roberto Caldelli, Rudy Becarelli, and Irene Amerini. «Image origin classification based on social network provenance». In: *IEEE Transactions on Information Forensics and Security* 12.6 (2017), pp. 1299–1308.
- [4] Davide Cozzolino and Luisa Verdoliva. «Noiseprint: A CNN-Based Camera Model Fingerprint». In: *IEEE Transactions on Information Forensics and Security* 15 (2020), pp. 144–159.
- [5] Anderson Rocha, Walter Scheirer, Terrance Boult, and Siome Goldenstein. «Vision of the unseen: Current trends and challenges in digital image and video forensics». In: *ACM Computing Surveys (CSUR)* 43.4 (2011), pp. 1–42.
- [6] Alessandro Piva. «An overview on image forensics». In: *International Scholarly Research Notices* 2013.1 (2013), p. 496701.
- [7] Tong Qiao, Florent Retraint, Rémi Cogranne, and Thanh Thai. «Individual camera device identification from JPEG images». In: *Signal Processing: Image Communication* 52 (Mar. 2017), pp. 74–86.
- [8] Gregory K. Wallace. «Overview of the JPEG (ISO/CCITT) still image compression standard». In: *Image Processing Algorithms and Techniques*. Ed. by Robert J. Moorhead II and Keith S. Pennington. Vol. 1244. International Society for Optics and Photonics. SPIE, 1990, pp. 220–233.
- [9] K. Kurosawa, K. Kuroki, and N. Saitoh. «CCD fingerprint method-identification of a video camera from videotaped images». In: *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*. Vol. 3. 1999, 537–540 vol.3.
- [10] S. Bayram, H. Sencar, N. Memon, and I. Avcibas. «Source camera identification based on CFA interpolation». In: *IEEE International Conference on Image Processing 2005*. Vol. 3. 2005, pp. III–69.
- [11] Kai San Choi, Edmund Y. Lam, and Kenneth K. Y. Wong. «Automatic source camera identification using the intrinsic lens radial distortion». In: *Opt. Express* 14.24 (Nov. 2006), pp. 11551–11565.
- [12] Hany Farid. *Digital image ballistics from JPEG quantization*. Technical Report TR2006-583. Dartmouth College, 2006.

- [13] Hany Farid. *Digital image ballistics from JPEG quantization: A followup study*. Technical Report TR2008-638. Dartmouth College, 2008.
- [14] Amel Tuama, Frédéric Comby, and Marc Chaumont. «Camera model identification with the use of deep convolutional neural networks». In: *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*. 2016, pp. 1–6.
- [15] Luca Bondi, Luca Baroffio, David Güera, Paolo Bestagini, Edward J. Delp, and Stefano Tubaro. «First Steps Toward Camera Model Identification With Convolutional Neural Networks». In: *IEEE Signal Processing Letters* 24.3 (2017), pp. 259–263.
- [16] Fu-Yuan Liang, Shu-Hui Gao, and Liang-Ju Xu. «Research on smartphone image source identification based on PRNU features collected multivariate sampling strategy». In: *Forensic Science International: Digital Investigation* 54 (2025), p. 301991. ISSN: 2666-2817.
- [17] Dipan Banerjee and Ruchira Naskar. «Deep Learning based Blind Source Identification of whatsapp and Facebook Images». In: *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*. 2021, pp. 306–311.
- [18] Samantha Klier and Harald Baier. «Boon or Bane: Source Camera Identification meets AI-generated images». In: *Forensic Science International: Digital Investigation* 56 (2026). DFRWS EU 2026 - Selected Papers from the 13th Annual Digital Forensics Research Conference Europe, p. 302064. ISSN: 2666-2817.
- [19] Thomas Gloe and Rainer Böhme. «The 'Dresden Image Database' for benchmarking digital image forensics». In: *Proceedings of the 2010 ACM symposium on applied computing*. 2010, pp. 1584–1590.
- [20] Chiara Galdi, Frank Hartung, and Jean-Luc Dugelay. «SOCRaTES: A Database of Realistic Data for SOurce Camera REcognition on Smartphones.» In: *ICPRAM*. 2019, pp. 648–655.
- [21] Dasara Shullani, Marco Fontani, Massimo Iuliani, Omar Al Shaya, and Alessandro Piva. «Vision: a video and image dataset for source identification». In: *EURASIP Journal on Information Security* 2017.1 (2017), p. 15.

Apéndice A

Planificación temporal y diagrama de Gantt

La ejecución de este TFG se ha desarrollado a lo largo de cinco meses, diferenciándose distintas tareas que abarcan desde el estudio del estado del arte hasta la redacción de la memoria. En la Tabla A.1 se detalla el esfuerzo estimado de horas invertidas en cada una de las actividades del proyecto.

Tabla A.1: Desglose de tareas y horas invertidas en el proyecto.

Descripción de la tarea	Horas estimadas
Investigación preliminar y estado del arte	25 h
Configuración entorno	5 h
Configuración Noiseprint	25 h
Configuración PRNU	25 h
Búsqueda y preparación del conjunto de datos	10 h
Configuración de métricas	10 h
Pruebas de funcionamiento base	40 h
Pruebas de WhatsApp	20 h
Pruebas de Instagram	20 h
Pruebas de filtros	35 h
Redacción de la memoria	90 h
Total	305 h

A continuación, en la Figura A.1, se ilustra la distribución temporal de estas tareas a lo largo del desarrollo del proyecto, agrupadas por bloques funcionales para facilitar su visualización cronológica.

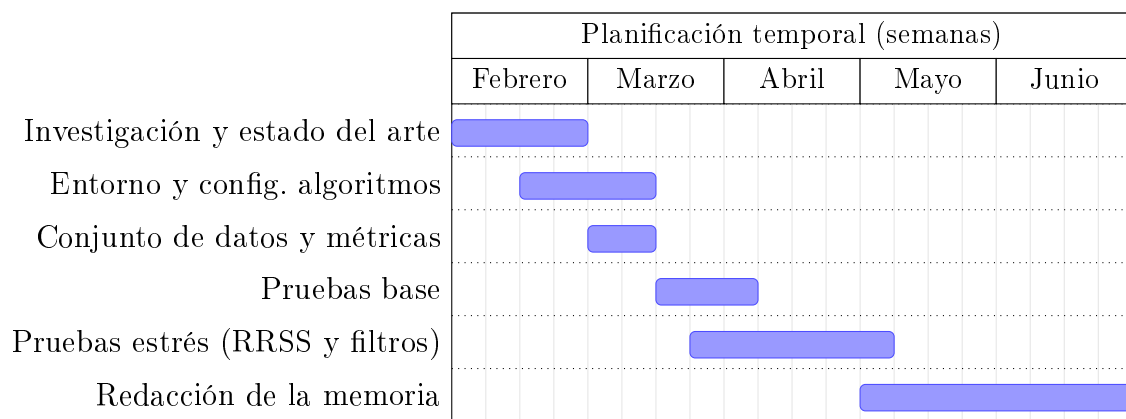


Figura A.1: Diagrama de Gantt del desarrollo del TFG.