



TRABAJO FIN DE MÁSTER

Máster Universitario en Microelectrónica: Diseño y Aplicaciones de Sistemas Micro/Nanométricos

Implementación de algoritmos de procesado de imágenes biomédicas en la plataforma Raspberry Pi

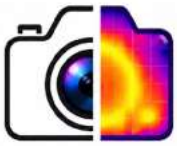
Autor
Rafael Rivero Serrano

Tutor
Juan Antonio Leñero Bardallo

Electrónica y Electromagnetismo

—

Sevilla, 6 de julio de 2026



Implementación de algoritmos de procesado de imágenes biomédicas en la plataforma Raspberry Pi

Autor
Rafael Rivero Serrano

Tutor
Juan Antonio Leñero Bardallo

Implementación de algoritmos de procesado de imágenes biomédicas en la plataforma Raspberry Pi:

Rafael Rivero Serrano

Palabras clave: Raspberry Pi, Hemangiomas, termografía infrarroja, análisis térmico, OpenCV, procesamiento de imagen

Resumen

Este TFM se centra en el desarrollo e implementación de algoritmos de procesamiento de imagen para el análisis de hemangiomas infantiles mediante termografía infrarroja. Para ello, se emplea una plataforma basada en Raspberry Pi equipada con una cámara visible y un sensor de imagen en el infrarrojo lejano (LWIR), capaz de adquirir y procesar imágenes térmicas en tiempo real. El trabajo consiste en el diseño y desarrollo de nuevas funcionalidades software integradas en forma de plugins, utilizando principalmente Python y bibliotecas de visión artificial como OpenCV. Entre las tareas abordadas se incluyen la segmentación de regiones de interés, la extracción de contornos, el cálculo de parámetros geométricos y térmicos de las lesiones y la visualización de resultados para facilitar su interpretación clínica.

Implementation of Biomedical Image Processing Algorithms on the Raspberry Pi Platform:

Rafael Rivero Serrano

Keywords: Raspberry Pi, Hemangiomas, Infrared Thermography, Thermal Analysis, OpenCV, Image Processing

Abstract

This Master's Thesis focuses on the development and implementation of image processing algorithms for the analysis of infantile hemangiomas using infrared thermography. The work is based on an existing Raspberry Pi platform equipped with both a visible-spectrum camera and a Long-Wave Infrared (LWIR) imaging sensor, capable of acquiring and processing thermal images in real time.
The main objective is the design and implementation of new software functionalities in the form of plugins, using Python and computer vision libraries such as OpenCV. The developed algorithms include image segmentation, contour extraction, computation of thermal and geometric lesion parameters, and visualization tools to assist clinical interpretation.

Yo, **Rafael Rivero Serrano**, alumno del **Máster Universitario en Microelectrónica: Diseño y Aplicaciones de Sistemas Micro/Nanométricos**, con DNI **28837147-T**, autorizo la ubicación de la siguiente copia de mi Trabajo Fin de Máster en la biblioteca del centro para que pueda ser consultada por las personas que lo deseen.

Fdo: *Rafael Rivero Serrano*

Sevilla a 6 de julio de 2026.

D. **Juan Antonio Leñero Bardallo**, Profesor del Área de Electrónica del Departamento Electrónica y Electromagnetismo de la Universidad de Sevilla.

Informa:

Que el presente trabajo, titulado ***Implementación de algoritmos de procesado de imágenes biomédicas en la plataforma Raspberry Pi,*** , ha sido realizado bajo su supervisión por **Rafael Rivero Serrano**, y autorizamos la defensa de dicho trabajo ante el tribunal que corresponda.

Y para que conste, expiden y firman el presente informe en Sevilla a 6 de julio de 2026.

El tutor:

Juan Antonio Leñero Bardallo

Yo, **Rafael Rivero Serrano**, alumno del **Máster Universitario en Microelectrónica: Diseño y Aplicaciones de Sistemas Micro/Nanométricos** de la **Facultad de Física de la Universidad de Sevilla**, con DNI **28837147-T**, declaro explícitamente que el trabajo presentado es original, entendido en el sentido que no he utilizado ninguna fuente sin citarla debidamente.

Fdo: **Rafael Rivero Serrano**

Sevilla, 6 de julio de 2026.

Agradecimientos

Agradecer a mi mujer y mis hijas, por tener esa infinita paciencia de aguantar mis horas desconectado de la familia para dedicárselo a los estudios. También a mis padres, sin ellos nada se hubiese hecho.

Índice general

1. Introducción **1**
- 1.1. Motivación . . . 1
- 1.2. Contexto clínico: hemangiomas infantiles . . . 2
- 1.3. Contexto tecnológico: termografía infrarroja y sistema portable . . . 3
- 1.4. Objetivos del trabajo . . . 4
- 1.5. Metodología general . . . 4
- 1.6. Estructura de la memoria . . . 5

2. Fundamentos **7**
- 2.1. Termografía infrarroja . . . 7
- 2.2. Plataformas embebidas para análisis térmico . . . 8
- 2.3. Sensores de imagen . . . 10
  - 2.3.1. Cámaras en el espectro visible . . . 10
  - 2.3.2. Sensores térmicos LWIR . . . 10
- 2.4. OpenCV y procesamiento digital de imagen . . . 10
  - 2.4.1. Biblioteca OpenCV . . . 11
  - 2.4.2. Operaciones básicas de procesamiento . . . 11
  - 2.4.3. Aplicación al análisis de imágenes térmicas . . . 11
- 2.5. Técnicas de segmentación y extracción de características . . . 12
  - 2.5.1. Umbralización automática mediante el método de Otsu . . . 12
  - 2.5.2. Operaciones morfológicas . . . 13
  - 2.5.3. Detección de contornos . . . 13

3. Plataforma de adquisición y procesamiento de imágenes **15**
- 3.1. Arquitectura general del sistema . . . 15
- 3.2. Componentes hardware . . . 17
  - 3.2.1. Unidad de procesamiento . . . 17
  - 3.2.2. Sistema de adquisición de imagen . . . 18
  - 3.2.3. Sistema de medida de temperatura . . . 18
  - 3.2.4. Interfaz de usuario . . . 18
  - 3.2.5. Sistema de alimentación . . . 18
- 3.3. Componentes software . . . 18
- 3.4. Flujo de funcionamiento de la plataforma . . . 19
- 3.5. Integración de nuevos algoritmos mediante plugins . . . 21

4. Desarrollo e implementación de plugins **23**
- 4.1. Objetivos de desarrollo . . . 23
  - 4.1.1. Plugin de eliminación del fondo térmico . . . 24
  - 4.1.2. Plugin de clasificación térmica mediante K-means . . . 28

4.1.3. Plugin 3: Análisis de diferencia térmica . . . 30
4.2. Prototipos explorados y comparativa . . . 32
4.2.1. Prototipo 1: Detección de bordes mediante Canny . . . 33
4.2.2. Prototipo 2: Detección de contornos . . . 34
4.2.3. Prototipo 3: Segmentación mediante Watershed . . . 36

5. Conclusiones **39**
5.1. Conclusiones generales . . . 39
5.2. Aportaciones del trabajo . . . 40
5.2.1. Resumen de las principales aportaciones . . . 41
5.3. Limitaciones del trabajo . . . 41
5.4. Líneas futuras . . . 42

Bibliografía **43**

A. Anexo A: Implementación de los algoritmos desarrollados **45**
A.1. Plugin de eliminación del fondo térmico . . . 45
A.1.1. Inicialización . . . 45
A.1.2. Procesamiento principal . . . 46
A.1.3. Eliminación del fondo . . . 46
A.1.4. Segmentación de la región caliente . . . 47
A.1.5. Limpieza morfológica y selección de región . . . 47
A.1.6. Detección de contornos y localización . . . 48
A.1.7. Obtención de parámetros . . . 49
A.1.8. Representación de resultados . . . 49
A.1.9. Detección de focos térmicos . . . 50
A.1.10. Conversión intensidad-temperatura . . . 51
A.1.11. Resumen del algoritmo . . . 51
A.2. Plugin de clasificación térmica mediante K-means . . . 51
A.2.1. Inicialización . . . 51
A.2.2. Clasificación térmica mediante K-means . . . 52
A.2.3. Representación de resultados . . . 53
A.2.4. Detección de focos térmicos . . . 53
A.2.5. Resumen del algoritmo . . . 54
A.3. Plugin de análisis de diferencia térmica . . . 54
A.3.1. Inicialización . . . 54
A.3.2. Segmentación de la región caliente . . . 55
A.3.3. Cálculo de parámetros térmicos . . . 55
A.3.4. Representación de resultados . . . 56
A.3.5. Resumen del algoritmo . . . 57

B. Manual de usuario **59**
B.1. Inicio de la plataforma . . . 59
B.2. Selección de los plugins . . . 61
B.3. Consideraciones finales . . . 61

Índice de figuras

2.1. Representación esquemática del espectro electromagnético. 8
2.2. Plataforma portátil de adquisición y procesamiento de imágenes térmicas desarrollada por el IMSE-CNM. [1]. 9
2.3. Ejemplo de segmentación basado en Otsu. 12
2.4. Ilustración de los pasos de procesamiento involucrados en la segmentación de imágenes. (a) Imagen termográfica original con valores de temperatura absolutos. (b) Histograma de la imagen termográfica original. Se resalta el umbral de temperatura (TH), determinado por el algoritmo de Otsu para la segmentación de la piel.(c) Imagen segmentada que muestra solo los valores de temperatura de la piel del paciente. (d) Líneas de contorno de temperatura extraídas de la imagen original. Las líneas de contorno con la temperatura más alta corresponden a la ubicación de la hiperplasia intimal (HI). (e) Detalle de las líneas de contorno de temperatura en la región afectada por la malformación. 14

3.1. Implementación final del sistema. (a) Vista frontal. Se han resaltado los componentes principales del sistema. (b) Vista posterior que muestra las dos interfaces de sensores de imagen que operan en paralelo. (c) Vista lateral que muestra los puertos Ethernet y USB disponibles. (d) Detalle de la conectividad interna con la pantalla táctil. 16
3.2. Elementos del sistema. 17
3.3. Flujo general de software. 19
3.4. Flujo general de funcionamiento de la plataforma de adquisición y procesamiento de imágenes utilizada durante el desarrollo del trabajo. 20

4.1. Foto de paciente en espectro visible mostrando la sospecha hemangioma. 24
4.2. Imagen en infrarrojo y con el plugin aplicado. 25
4.3. Imagen de otro paciente en visible. 27
4.4. Imagen de otro paciente en IR. 27
4.5. Funcionamiento del plugin de clasificación térmica mediante K-means. Izquierda: imagen infrarroja original. Derecha: resultado obtenido tras la clasificación de regiones térmicas y detección de focos de mayor temperatura. 28
4.6. Resultado del plugin de análisis de diferencia térmica. 31
4.7. Resultado del plugin de análisis de diferencia térmica en otro paciente. 31
4.8. Resultado obtenido mediante el algoritmo de detección de bordes de Canny sobre una imagen térmica. 33
4.9. Resultado obtenido mediante el plugin de detección de contornos. El algoritmo delimita la región de mayor tamaño detectada y calcula la temperatura media estimada de dicha zona. 35

4.10. Resultado obtenido mediante el algoritmo Watershed aplicado sobre una imagen térmica. 37

5.1. Esquema de desarrollo de TFM. 41

B.1. Ejecución de la aplicación VisionBio desde el escritorio de la Raspberry Pi. . . . 60
B.2. Confirmación de ejecución de la aplicación. 60
B.3. Interfaz principal de la aplicación durante la adquisición en tiempo real. 61
B.4. Interfaz con plugin aplicado. 61

Capítulo 1

Introducción

La ingeniería actual se caracteriza por su capacidad para integrarse en ámbitos muy diversos, permitiendo el desarrollo de soluciones tecnológicas aplicadas a problemas reales de la sociedad. Entre todos ellos, el ámbito sanitario representa uno de los campos donde la tecnología puede generar un mayor impacto, contribuyendo tanto al apoyo al diagnóstico como al seguimiento de patologías mediante herramientas cada vez más accesibles, precisas y no invasivas.
En particular, el estudio de los hemangiomas infantiles representa un escenario de gran interés debido a la importancia que tiene su detección y seguimiento durante las primeras etapas de desarrollo. Aunque una gran parte de estas lesiones evolucionan favorablemente, algunas pueden presentar complicaciones que requieren una monitorización adecuada por parte de los especialistas. En este contexto, la termografía infrarroja se presenta como una técnica prometedora, ya que permite obtener información relacionada con la distribución térmica de la piel de forma completamente no invasiva y sin contacto con el paciente.
Otro aspecto que ha resultado especialmente motivador es la utilización de plataformas de bajo coste basadas en Raspberry Pi. Tradicionalmente, muchas aplicaciones de imagen médica han estado asociadas a equipamiento complejo y costoso. Sin embargo, los avances recientes en sistemas embebidos, sensores térmicos y herramientas de software libre han abierto la posibilidad de desarrollar dispositivos portables capaces de ejecutar algoritmos avanzados de procesamiento de imagen en tiempo real. Esta aproximación permite acercar tecnologías de análisis térmico a entornos clínicos de una forma más accesible y flexible.
El presente trabajo se enmarca dentro de una línea de investigación previamente desarrollada por el Instituto de Microelectrónica de Sevilla (IMSE-CNM) y diversos trabajos relacionados con el empleo de termografía infrarroja para el análisis de lesiones vasculares. Sobre esta plataforma existente, el objetivo consiste en desarrollar nuevas funcionalidades software en forma de plugins que permitan extraer información de interés clínico a partir de imágenes térmicas obtenidas mediante sensores LWIR. Entre estas funcionalidades se incluyen algoritmos de segmentación, detección de regiones de interés, extracción de contornos y cálculo de parámetros relacionados con la evolución de las lesiones.

1.1. Motivación

La elección de este Trabajo Fin de Máster viene motivada por el interés en la aplicación de la tecnología a problemas reales del ámbito sanitario. Resulta especialmente interesante comprobar cómo todos los conocimientos en electrónica pueden integrarse para desarrollar herramientas con una utilidad directa en el diagnóstico y seguimiento de patologías médicas.
Uno de los aspectos más atractivos de este proyecto es su carácter multidisciplinar, ya que combina conocimientos de microelectrónica, visión artificial, procesamiento de imagen y sistemas

embebidos con aplicaciones clínicas reales. Esta combinación permite abordar un problema desde distintas perspectivas, integrando tanto el desarrollo tecnológico como las necesidades planteadas por los profesionales sanitarios.
Además, el trabajo se desarrolla sobre una plataforma termográfica real utilizada en investigación clínica, lo que aporta un componente práctico especialmente motivador. La posibilidad de diseñar e implementar algoritmos que posteriormente puedan ser utilizados para analizar imágenes de pacientes reales supone una oportunidad de participar en el desarrollo de herramientas con potencial aplicación médica.
Otro factor de interés es el empleo de plataformas hardware de bajo coste basadas en Raspberry Pi. En los últimos años, este tipo de sistemas ha demostrado que es posible desarrollar soluciones avanzadas utilizando recursos relativamente sencillos y accesibles. La combinación de una Raspberry Pi, sensores de imagen en el espectro visible e infrarrojo lejano y bibliotecas de procesamiento de imagen como OpenCV permite construir sistemas capaces de ejecutar algoritmos complejos en tiempo real, acercando tecnologías que tradicionalmente requerían equipamiento mucho más costoso.
Finalmente, la motivación principal de este trabajo radica en la oportunidad de contribuir al desarrollo de nuevas funcionalidades software para una plataforma existente, mediante la creación de plugins capaces de proporcionar información de interés clínico a partir de imágenes térmicas.

1.2. Contexto clínico: hemangiomas infantiles

El hemangioma infantil (HI) es el tumor vascular benigno más frecuente durante la infancia y una de las anomalías vasculares más comunes en recién nacidos y lactantes [2]. Se caracteriza por una proliferación anormal de vasos sanguíneos que da lugar a lesiones visibles en la piel y, en algunos casos, en tejidos más profundos. Aunque la mayoría de los hemangiomas presentan una evolución favorable y tienden a involucionar espontáneamente con el tiempo, determinadas lesiones pueden desarrollar complicaciones que requieren seguimiento médico y tratamiento especializado.
La incidencia estimada de los hemangiomas infantiles oscila entre el 2 por ciento y el 10 por ciento de los recién nacidos, siendo más frecuentes en niñas, prematuros y niños con bajo peso al nacer [2, 3]. A pesar de tratarse de una lesión benigna, algunos casos pueden provocar ulceraciones, alteraciones funcionales o secuelas estéticas permanentes, especialmente cuando afectan a zonas sensibles o presentan un crecimiento significativo durante los primeros meses de vida.
Desde el punto de vista clínico, el hemangioma infantil presenta dos etapas principales de evolución. La primera corresponde a la fase proliferativa, caracterizada por un crecimiento rápido de la lesión asociado a un aumento de la vascularización. Posteriormente tiene lugar una fase de involución progresiva durante la cual el hemangioma disminuye gradualmente de tamaño y actividad vascular [2]. La duración y evolución de estas etapas puede variar considerablemente entre pacientes, por lo que resulta necesario realizar un seguimiento periódico de la lesión para evaluar su evolución y detectar posibles complicaciones.
En función de la profundidad de la lesión, los hemangiomas infantiles suelen clasificarse en superficiales, profundos y mixtos [2]. Los hemangiomas superficiales presentan una coloración rojiza característica y son fácilmente identificables mediante inspección visual. Por el contrario, los hemangiomas profundos se desarrollan en capas internas del tejido y pueden resultar más difíciles de detectar debido a la escasa manifestación visible en la superficie de la piel. Los hemangiomas mixtos combinan características de ambos tipos y constituyen una situación intermedia entre las dos categorías anteriores.
Tradicionalmente, la evaluación clínica de estas lesiones se basa en la exploración física realizada por especialistas, complementada en determinados casos mediante técnicas de imagen como la ecografía Doppler o la resonancia magnética [3]. Sin embargo, estas técnicas pueden requerir

equipamiento especializado y no siempre resultan adecuadas para realizar controles frecuentes durante la evolución de la lesión.
En los últimos años, la termografía infrarroja ha despertado un interés creciente como herramienta complementaria para el análisis de hemangiomas infantiles. Diversos estudios han demostrado la existencia de una relación entre la actividad vascular de la lesión y la distribución térmica observada en la superficie cutánea [4]. Debido a ello, el análisis térmico puede proporcionar información objetiva sobre la evolución del hemangioma de forma no invasiva, sin contacto con el paciente y con un coste significativamente inferior al de otras técnicas de imagen. Estas características convierten a la termografía en una tecnología prometedora para el seguimiento clínico y la monitorización de este tipo de lesiones vasculares.

1.3. Contexto tecnológico: termografía infrarroja y sistema portable

La termografía infrarroja es una técnica de imagen no invasiva que permite obtener información sobre la distribución de temperaturas en la superficie de un objeto mediante la detección de la radiación infrarroja emitida de forma natural por los cuerpos. A diferencia de las cámaras convencionales, que capturan radiación dentro del espectro visible, las cámaras termográficas son capaces de detectar radiación en bandas del espectro infrarrojo, generando imágenes en las que cada píxel se encuentra asociado a una temperatura determinada [1].
En aplicaciones biomédicas, la termografía infrarroja ha despertado un creciente interés debido a su capacidad para proporcionar información fisiológica sin necesidad de contacto físico con el paciente. La distribución térmica de la piel está directamente relacionada con fenómenos como la perfusión sanguínea, el metabolismo tisular y la actividad vascular, permitiendo detectar alteraciones asociadas a diversas patologías [5, 4]. Esta característica resulta especialmente relevante en el estudio de lesiones vasculares, donde las variaciones en el flujo sanguíneo pueden traducirse en diferencias de temperatura detectables mediante sensores térmicos.
Los sistemas empleados en este tipo de aplicaciones suelen utilizar sensores que operan en la banda LWIR (Long Wave Infrared), comprendida aproximadamente entre los 8 y los 14 µm de longitud de onda. Esta región espectral resulta especialmente adecuada para el análisis térmico de tejidos biológicos debido a que la piel humana emite gran parte de su radiación térmica en dicho rango [1]. Gracias a ello, es posible obtener mapas térmicos de alta sensibilidad capaces de representar pequeñas variaciones de temperatura sobre la superficie cutánea.
Diversos estudios han demostrado la utilidad de la termografía infrarroja en el análisis de anomalías vasculares. En particular, se ha observado que lesiones con una elevada actividad vascular presentan temperaturas superiores a las del tejido circundante debido al incremento del flujo sanguíneo. Esta relación permite emplear imágenes térmicas como herramienta complementaria para la detección, caracterización y seguimiento de patologías como las malformaciones vasculares y los hemangiomas infantiles [5, 4].
Con el objetivo de aprovechar estas ventajas, diferentes grupos de investigación han desarrollado plataformas portables capaces de adquirir y procesar imágenes térmicas en tiempo real. Entre ellas destaca la plataforma desarrollada por el Instituto de Microelectrónica de Sevilla (IMSE-CNM), basada en una Raspberry Pi como unidad principal de procesamiento y equipada con una cámara visible y un sensor térmico LWIR [1]. Esta arquitectura proporciona una solución compacta, flexible y de bajo coste capaz de integrar funciones de adquisición, visualización y procesamiento de imágenes en un único dispositivo.
La utilización de sistemas embebidos basados en Raspberry Pi permite además implementar algoritmos de procesamiento de imagen directamente sobre la plataforma, facilitando el desarrollo de nuevas funcionalidades software adaptadas a las necesidades clínicas. Esta característica resul-

ta especialmente relevante en el contexto del presente Trabajo, cuyo objetivo principal consiste en desarrollar e integrar nuevos plugins de análisis de imagen sobre una plataforma termográfica ya existente, ampliando sus capacidades para la detección y caracterización de regiones de interés asociadas a hemangiomas infantiles.

1.4. Objetivos del trabajo

El objetivo principal de este trabajo es el desarrollo e integración de nuevas funcionalidades software para una plataforma portátil de termografía infrarroja orientada al análisis y seguimiento de hemangiomas infantiles. Para ello, se empleará una plataforma previamente desarrollada basada en Raspberry Pi y sensores de imagen en el espectro visible e infrarrojo lejano (LWIR), sobre la que se implementarán algoritmos de procesamiento de imagen capaces de extraer información de interés clínico a partir de imágenes térmicas.
Con el fin de alcanzar este objetivo general, se plantean los siguientes objetivos específicos:

- Estudiar el funcionamiento de la plataforma termográfica existente y los principios básicos de la termografía infrarroja aplicados al ámbito médico.
- Analizar las características clínicas de los hemangiomas infantiles y las necesidades asociadas a su seguimiento mediante técnicas de imagen no invasivas.
- Investigar técnicas de procesamiento digital de imagen susceptibles de ser aplicadas al análisis de imágenes térmicas, incluyendo segmentación, detección de contornos y extracción de características geométricas.
- Diseñar e implementar nuevos plugins software en lenguaje Python compatibles con la plataforma de procesamiento existente.
- Desarrollar algoritmos capaces de detectar regiones de interés asociadas a variaciones térmicas presentes en las lesiones estudiadas.
- Obtener parámetros cuantitativos de interés, tales como área, diámetro y características térmicas de las regiones detectadas.
- Evaluar el funcionamiento de las herramientas desarrolladas mediante imágenes obtenidas con el sistema termográfico.
- Analizar las posibilidades de ampliación futura de la plataforma mediante la incorporación de nuevos algoritmos de procesado y técnicas avanzadas de análisis de imagen.

1.5. Metodología general

Para alcanzar los objetivos planteados en este TFM se ha seguido una metodología estructurada en distintas fases, combinando el estudio bibliográfico, el análisis de una plataforma existente y el desarrollo de nuevas funcionalidades software orientadas al procesamiento de imágenes térmicas. En una primera fase se realizó una revisión bibliográfica sobre hemangiomas infantiles, termografía infrarroja y técnicas de procesamiento digital de imagen. Esta etapa permitió comprender tanto el contexto clínico del problema como las tecnologías empleadas para la adquisición y análisis de imágenes térmicas. Asimismo, se estudiaron trabajos previos relacionados con el uso de sensores infrarrojos para la monitorización de lesiones vasculares y con el desarrollo de plataformas portables de diagnóstico basadas en sistemas embebidos.

Posteriormente, se llevó a cabo un análisis de la plataforma de adquisición y procesamiento proporcionada para el desarrollo del trabajo. Dicha plataforma está basada en una Raspberry Pi e integra sensores de imagen en el espectro visible e infrarrojo lejano (LWIR), permitiendo la ejecución de algoritmos de procesamiento de imagen mediante aplicaciones desarrolladas en lenguaje Python. Durante esta fase se estudió la arquitectura del sistema, el funcionamiento de los distintos módulos software y el mecanismo de integración de nuevas funcionalidades mediante plugins.

Una vez comprendido el funcionamiento de la plataforma, se procedió al estudio y selección de algoritmos de procesamiento digital de imagen adecuados para el análisis de imágenes térmicas. Entre las técnicas consideradas se encuentran métodos de segmentación, detección de contornos, umbralización automática y operaciones morfológicas, todas ellas ampliamente utilizadas en aplicaciones de visión artificial y análisis biomédico.

La siguiente fase consistió en el desarrollo e implementación de nuevos plugins software capaces de integrarse en la plataforma existente. Estos plugins fueron diseñados para procesar las imágenes térmicas adquiridas por el sistema y extraer información de interés relacionada con las regiones de mayor actividad térmica. Para ello se empleó principalmente el lenguaje Python junto con bibliotecas de visión artificial como OpenCV.

Finalmente, se realizaron pruebas funcionales utilizando imágenes térmicas obtenidas mediante el sistema disponible. Los resultados obtenidos permitieron evaluar el comportamiento de los algoritmos desarrollados y analizar su capacidad para detectar regiones de interés y obtener parámetros cuantitativos asociados a las lesiones estudiadas. A partir de esta evaluación se identificaron posibles líneas de mejora y futuras ampliaciones de la plataforma.

1.6. Estructura de la memoria

La presente memoria se organiza en cinco capítulos, además de la bibliografía y dos anexos que contienen información complementaria sobre la implementación y el uso de la plataforma.

En el Capítulo 1 se presenta el contexto general del trabajo. Se describen la motivación del proyecto, las características principales de los hemangiomas infantiles, el interés de la termografía infrarroja para su seguimiento, los objetivos planteados y la metodología seguida.

El Capítulo 2 recoge los fundamentos y las tecnologías empleadas. En él se explican los principios básicos de la termografía infrarroja, las características de los sensores de imagen visibles y térmicos, el uso de OpenCV y las principales técnicas de segmentación, procesamiento morfológico y detección de contornos utilizadas durante el desarrollo.

En el Capítulo 3 se describe la plataforma portátil de adquisición y procesamiento de imágenes empleada en el proyecto. Se presentan su arquitectura general, sus componentes hardware y software, el flujo de funcionamiento de la aplicación y el procedimiento utilizado para integrar nuevos algoritmos mediante plugins.

El Capítulo 4 constituye la parte principal del trabajo. En este capítulo se describen los plugins desarrollados para el procesamiento de imágenes térmicas, los resultados obtenidos con imágenes de pacientes y los prototipos explorados durante el desarrollo. También se comparan técnicas como la detección de bordes mediante Canny, la detección de contornos y la segmentación mediante Watershed.

Finalmente, el Capítulo 5 reúne las conclusiones del trabajo, sus principales aportaciones y las limitaciones detectadas. Asimismo, se proponen distintas líneas futuras destinadas a mejorar los algoritmos desarrollados y ampliar las funcionalidades de la plataforma.

La memoria se completa con la bibliografía utilizada, un anexo dedicado a la implementación de los algoritmos desarrollados y un manual básico de usuario de la aplicación.

Capítulo 2

Fundamentos y tecnologías empleadas

En este capítulo se presentan los fundamentos teóricos y las tecnologías que constituyen la base del sistema desarrollado. En primer lugar, se describen los principios físicos de la termografía infrarroja y su aplicación en el ámbito biomédico, prestando especial atención a su utilidad para la detección y seguimiento de alteraciones vasculares cutáneas. Posteriormente, se introducen los dispositivos utilizados para la adquisición de imágenes térmicas y visibles, así como las principales técnicas de procesamiento digital de imagen empleadas para el análisis de la información obtenida.

2.1. Termografía infrarroja

La termografía infrarroja es una técnica de adquisición de imágenes que permite representar la distribución de temperaturas de una escena mediante la detección de la radiación infrarroja emitida por los cuerpos. A diferencia de las cámaras convencionales, que capturan la radiación reflejada dentro del espectro visible, las cámaras termográficas registran la energía emitida por los objetos debido a su temperatura, generando imágenes en las que cada píxel contiene información térmica asociada a una determinada región de la escena [1].

Desde el punto de vista físico, todos los cuerpos cuya temperatura sea superior al cero absoluto emiten radiación electromagnética. La cantidad y distribución espectral de dicha radiación dependen de la temperatura del objeto, lo que permite estimar su temperatura superficial mediante sensores adecuados. Esta propiedad constituye la base de funcionamiento de las cámaras termográficas utilizadas en aplicaciones médicas, industriales y científicas [6].

En aplicaciones biomédicas, la termografía presenta un interés especial debido a que la temperatura superficial de la piel está relacionada con diferentes procesos fisiológicos, entre ellos la perfusión sanguínea, el metabolismo celular y la actividad vascular. Alteraciones en estos procesos pueden provocar variaciones térmicas detectables mediante sensores infrarrojos, permitiendo obtener información complementaria a la proporcionada por otras técnicas de imagen convencionales [4, 5].

La mayoría de los sistemas termográficos empleados en medicina operan en la banda LWIR (Long Wave Infrared), comprendida aproximadamente entre los 8 y los 14 µm de longitud de onda. Esta región del espectro electromagnético resulta especialmente adecuada para el análisis de tejidos biológicos, ya que la piel humana emite gran parte de su energía térmica dentro de este intervalo espectral. Gracias a ello, es posible medir pequeñas diferencias de temperatura de forma no invasiva y sin necesidad de contacto físico con el paciente [1].

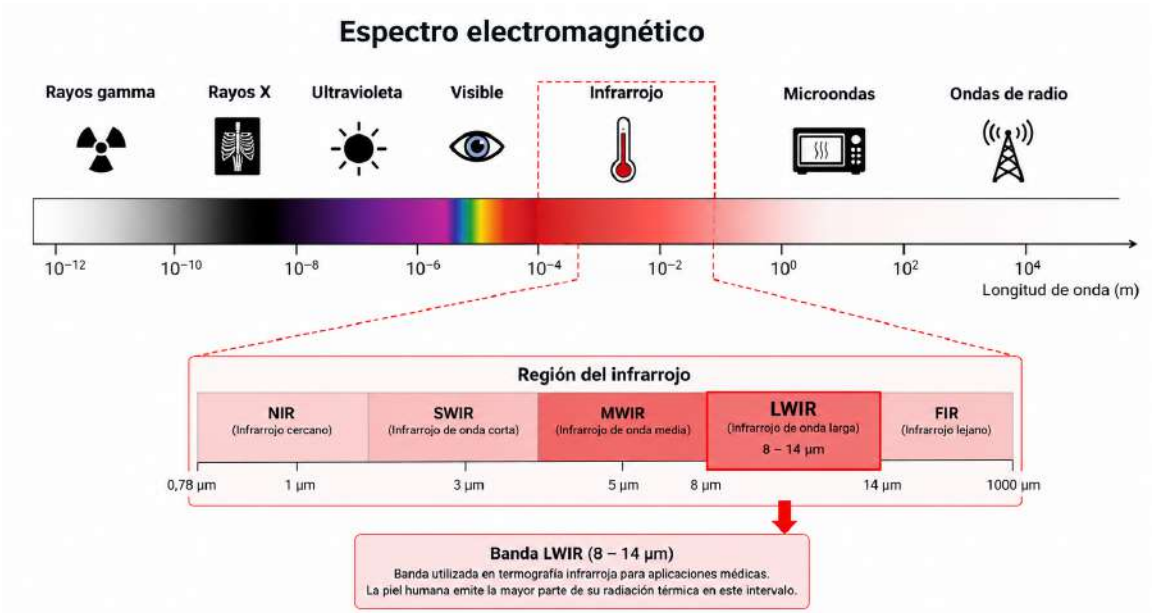


Figura 2.1: Representación esquemática del espectro electromagnético.

En los últimos años, la reducción del coste de los sensores infrarrojos y el aumento de su resolución han favorecido la aparición de nuevas aplicaciones clínicas basadas en análisis térmico. Entre ellas destacan el estudio de quemaduras, procesos inflamatorios, alteraciones vasculares y diferentes tipos de tumores superficiales. En particular, diversos trabajos han demostrado la utilidad de la termografía infrarroja para la evaluación de hemangiomas infantiles y malformaciones vasculares, debido a la estrecha relación existente entre la actividad vascular de estas lesiones y las variaciones de temperatura observadas en la superficie cutánea [4, 5].
Una de las principales ventajas de la termografía frente a otras técnicas de diagnóstico es su carácter completamente no invasivo. La adquisición de imágenes puede realizarse sin contacto con el paciente, sin emplear radiación ionizante y sin necesidad de procedimientos complejos. Además, los resultados se obtienen de forma inmediata, permitiendo realizar exploraciones repetidas durante el seguimiento clínico de una lesión sin causar molestias al paciente. Estas características hacen que la termografía constituya una herramienta especialmente atractiva para aplicaciones pediátricas, como el seguimiento de hemangiomas infantiles.

2.2. Plataformas embebidas para análisis térmico

El desarrollo de sensores térmicos de bajo coste y el aumento de la capacidad de procesamiento de los sistemas embebidos han favorecido la aparición de nuevas plataformas destinadas al análisis térmico en aplicaciones médicas. Estos sistemas permiten adquirir, procesar y visualizar imágenes termográficas utilizando dispositivos compactos, portables y energéticamente eficientes.
Un sistema embebido puede definirse como una plataforma informática diseñada para realizar tareas específicas dentro de una aplicación determinada. A diferencia de los ordenadores de propósito general, los sistemas embebidos se caracterizan por integrar hardware y software optimizados para una función concreta, ofreciendo una elevada flexibilidad y un reducido consumo energético.
En el ámbito de la imagen médica, las plataformas embebidas permiten implementar soluciones portables capaces de realizar tareas de adquisición y procesamiento de imágenes en tiempo real. Esta característica resulta especialmente interesante en aplicaciones clínicas donde se requiere

movilidad, facilidad de uso y disponibilidad inmediata de los resultados.

Entre las diferentes plataformas disponibles actualmente, Raspberry Pi se ha convertido en una de las soluciones más utilizadas en proyectos de investigación y desarrollo debido a su bajo coste, reducido tamaño y amplio soporte software. Su arquitectura permite ejecutar sistemas operativos basados en Linux y utilizar lenguajes de programación de alto nivel, como Python, facilitando la integración de bibliotecas de procesamiento de imagen y visión artificial.

En el contexto de la termografía médica, diversos trabajos han demostrado la viabilidad de emplear Raspberry Pi como unidad central de procesamiento para sistemas de adquisición térmica. Un ejemplo de ello es la plataforma desarrollada por el Instituto de Microelectrónica de Sevilla (IMSE-CNM), diseñada para la adquisición y procesamiento simultáneo de imágenes visibles e infrarrojas mediante una arquitectura modular basada en Raspberry Pi [1].

La plataforma integra una cámara convencional en el espectro visible junto con un sensor térmico LWIR, permitiendo obtener información complementaria de una misma escena. Mientras que la imagen visible proporciona detalles relacionados con la textura, color y localización de la lesión, la imagen térmica aporta información asociada a la distribución de temperatura superficial. La combinación de ambas fuentes de información permite desarrollar herramientas de análisis más completas para aplicaciones médicas y dermatológicas.

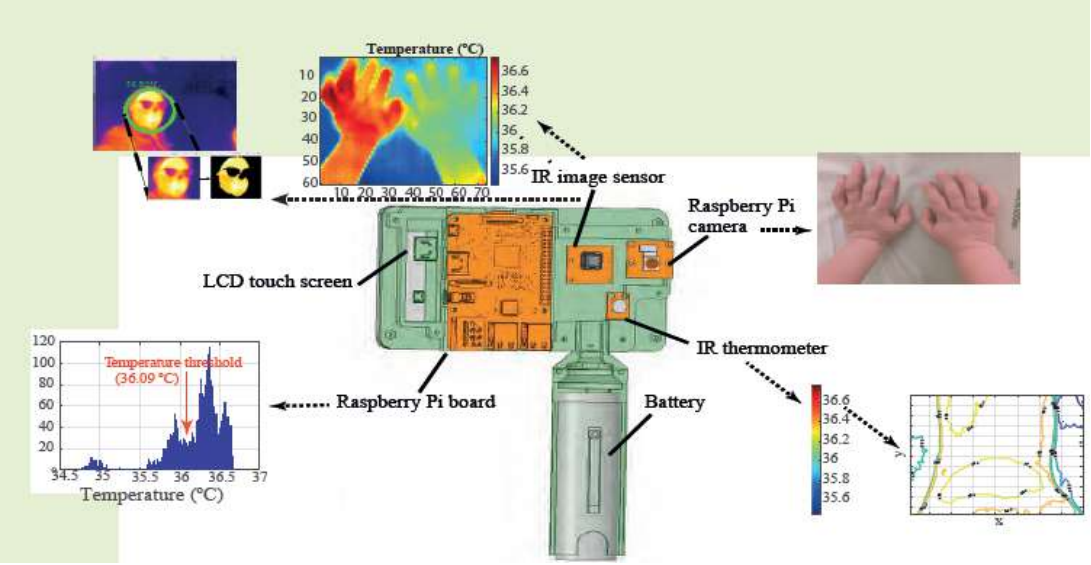


Figura 2.2: Plataforma portátil de adquisición y procesamiento de imágenes térmicas desarrollada por el IMSE-CNM. [1].

Además de la adquisición de imágenes, estas plataformas permiten ejecutar algoritmos de procesamiento digital directamente sobre el dispositivo. Gracias a ello es posible implementar funciones de segmentación, detección de regiones de interés, extracción de características y visualización de resultados en tiempo real, sin necesidad de recurrir a equipos externos de procesamiento.

Las características descritas convierten a las plataformas embebidas basadas en Raspberry Pi en una alternativa adecuada para el desarrollo de sistemas termográficos portables orientados a aplicaciones clínicas. Sobre este tipo de plataforma se desarrolla el presente proyecto, centrado en la implementación de nuevas funcionalidades software para el análisis de imágenes térmicas obtenidas durante el seguimiento de hemangiomas infantiles.

2.3. Sensores de imagen

Los sistemas de visión artificial requieren dispositivos capaces de transformar la información procedente del entorno en datos digitales susceptibles de ser procesados por un ordenador. Estos dispositivos reciben el nombre de sensores de imagen y constituyen uno de los elementos fundamentales de cualquier sistema de adquisición y procesamiento de imágenes.
Dependiendo de la región del espectro electromagnético utilizada para la adquisición, los sensores pueden proporcionar diferentes tipos de información sobre una misma escena. En el ámbito biomédico resulta especialmente interesante la combinación de imágenes visibles e infrarrojas, ya que ambas aportan información complementaria sobre la zona analizada.

2.3.1. Cámaras en el espectro visible

Las cámaras convencionales operan dentro del espectro visible, aproximadamente entre los 400 y 700 nm de longitud de onda. Estos dispositivos capturan la radiación reflejada por los objetos y generan imágenes que representan características como el color, la textura, la forma y la apariencia visual de la escena.
En aplicaciones médicas y dermatológicas, las imágenes visibles permiten identificar la localización de una lesión, analizar su aspecto superficial y documentar visualmente su evolución. Sin embargo, este tipo de imágenes no proporciona información directa sobre la distribución térmica ni sobre procesos fisiológicos internos relacionados con la vascularización o el metabolismo tisular.
Por este motivo, las cámaras visibles suelen emplearse junto con otros sensores complementarios capaces de proporcionar información adicional sobre la región estudiada. En particular, la combinación de imágenes visibles y térmicas permite obtener una caracterización más completa de las lesiones vasculares y facilita el desarrollo de herramientas avanzadas de análisis clínico.

2.3.2. Sensores térmicos LWIR

Los sensores térmicos utilizados en termografía médica operan habitualmente en la banda LWIR (Long Wave Infrared), comprendida entre aproximadamente 8 y 14 µm. A diferencia de las cámaras visibles, estos dispositivos no detectan la luz reflejada por los objetos, sino la radiación infrarroja emitida de forma natural debido a su temperatura.
Cada píxel de una imagen térmica representa una medida asociada a la energía infrarroja detectada en una determinada región de la escena. Mediante procesos de calibración adecuados, esta información puede transformarse en valores de temperatura superficial, permitiendo generar mapas térmicos de alta sensibilidad.
La utilización de sensores LWIR resulta especialmente adecuada para aplicaciones biomédicas debido a que la piel humana emite gran parte de su radiación térmica dentro de esta región espectral. Como consecuencia, pequeñas variaciones de temperatura asociadas a cambios en la vascularización o en la perfusión sanguínea pueden detectarse mediante técnicas de termografía infrarroja [4, 5].

2.4. OpenCV y procesamiento digital de imagen

El procesamiento digital de imagen constituye una de las áreas fundamentales de la visión artificial y engloba el conjunto de técnicas destinadas a mejorar, analizar o extraer información a partir de imágenes digitales. Estas técnicas permiten transformar los datos obtenidos por un sensor de imagen en información útil para una determinada aplicación, facilitando la detección de objetos, la segmentación de regiones de interés o la obtención de medidas cuantitativas [6].

En aplicaciones biomédicas, el procesamiento digital de imagen desempeña un papel especialmente relevante debido a la necesidad de analizar grandes cantidades de información visual de forma objetiva y reproducible. La utilización de algoritmos adecuados permite detectar patrones, identificar estructuras anatómicas y cuantificar características de interés clínico que pueden resultar difíciles de evaluar mediante inspección visual directa.
En el contexto de la termografía infrarroja, estas técnicas permiten procesar las imágenes térmicas adquiridas por los sensores LWIR con el objetivo de detectar regiones asociadas a variaciones de temperatura, delimitar contornos y extraer parámetros relacionados con las lesiones estudiadas. Estas funcionalidades constituyen la base de los algoritmos desarrollados durante este trabajo.

2.4.1. Biblioteca OpenCV

OpenCV (Open Source Computer Vision Library) es una biblioteca de código abierto ampliamente utilizada en aplicaciones de visión artificial y procesamiento digital de imagen. Fue desarrollada originalmente por Intel y actualmente constituye una de las herramientas más empleadas tanto en entornos académicos como industriales debido a su amplio conjunto de funciones y su compatibilidad con diferentes lenguajes de programación.
La biblioteca proporciona herramientas para la adquisición de imágenes, filtrado, segmentación, detección de contornos, extracción de características y análisis de objetos, entre muchas otras funcionalidades. Además, OpenCV puede integrarse fácilmente con lenguajes de alto nivel como Python, lo que facilita el desarrollo rápido de aplicaciones de procesamiento de imagen.
En este trabajo, OpenCV se emplea como herramienta principal para la implementación de los algoritmos desarrollados, proporcionando las funciones necesarias para procesar las imágenes térmicas obtenidas por la plataforma y extraer información relevante de las regiones de interés detectadas.

2.4.2. Operaciones básicas de procesamiento

Antes de aplicar técnicas avanzadas de análisis, las imágenes suelen someterse a diferentes etapas de preprocesado destinadas a mejorar su calidad o facilitar la posterior extracción de información. Entre las operaciones más habituales se encuentran el suavizado mediante filtros, la normalización de intensidades, la conversión entre formatos de imagen y la eliminación de ruido.
Una vez realizada esta etapa inicial, pueden aplicarse técnicas de segmentación para separar las regiones de interés del resto de la imagen. Estas técnicas permiten identificar áreas que presentan características diferenciadas respecto al entorno, como pueden ser variaciones significativas de temperatura en imágenes termográficas.
Tras la segmentación, es posible utilizar algoritmos de detección de contornos y extracción de características para obtener parámetros geométricos asociados a los objetos detectados. Entre las medidas más empleadas se encuentran el área, el perímetro, el diámetro equivalente o la posición de la región analizada, parámetros que resultan especialmente útiles en aplicaciones biomédicas y de seguimiento clínico [6].

2.4.3. Aplicación al análisis de imágenes térmicas

Las imágenes térmicas presentan características particulares que hacen necesario el empleo de técnicas específicas de procesamiento. En este tipo de imágenes, cada píxel representa información relacionada con la temperatura de una determinada región de la escena, permitiendo identificar áreas con un comportamiento térmico diferente al de su entorno.

En el presente trabajo se emplean técnicas de segmentación y detección de contornos para localizar regiones con temperaturas elevadas susceptibles de corresponderse con lesiones vasculares. Posteriormente, se calculan diferentes parámetros geométricos y térmicos que permiten caracterizar dichas regiones y facilitar su análisis.
Las herramientas proporcionadas por OpenCV permiten implementar estas funcionalidades de forma eficiente, constituyendo la base software sobre la que se desarrollan los plugins descritos en capítulos posteriores.

2.5. Técnicas de segmentación y extracción de características

Una de las etapas más importantes dentro del procesamiento digital de imagen es la segmentación. Su objetivo consiste en dividir una imagen en diferentes regiones con características similares, permitiendo separar los objetos o zonas de interés del resto de la escena. En aplicaciones biomédicas, la segmentación facilita la identificación de estructuras anatómicas o regiones asociadas a determinadas patologías, constituyendo habitualmente el primer paso para la posterior extracción de información cuantitativa.
En el caso de las imágenes térmicas, la segmentación permite localizar regiones cuya distribución de temperatura difiere significativamente del entorno. Estas regiones pueden corresponderse con zonas de elevada actividad vascular o con lesiones que presentan un comportamiento térmico distinto al tejido circundante. Una vez segmentadas, resulta posible calcular diferentes parámetros geométricos y térmicos que ayuden a caracterizar dichas regiones y evaluar su evolución.

2.5.1. Umbralización automática mediante el método de Otsu

El método de Otsu es una de las técnicas de segmentación más utilizadas en procesamiento digital de imagen. Fue propuesto por Nobuyuki Otsu en 1979 y se basa en la selección automática de un umbral que permita separar los píxeles de una imagen en diferentes clases de forma óptima. La idea fundamental consiste en analizar el histograma de intensidades de la imagen y determinar el valor de umbral que maximiza la separación entre los grupos de píxeles existentes. De este modo, es posible distinguir automáticamente regiones con características diferentes sin necesidad de definir manualmente un valor de referencia.
En el contexto de este trabajo, el método de Otsu se emplea para segmentar imágenes térmicas y detectar regiones que presentan temperaturas significativamente distintas respecto al entorno. La utilización de esta técnica permite automatizar el proceso de identificación de áreas potencialmente asociadas a lesiones vasculares, constituyendo una herramienta sencilla y eficaz para el análisis preliminar de imágenes termográficas.

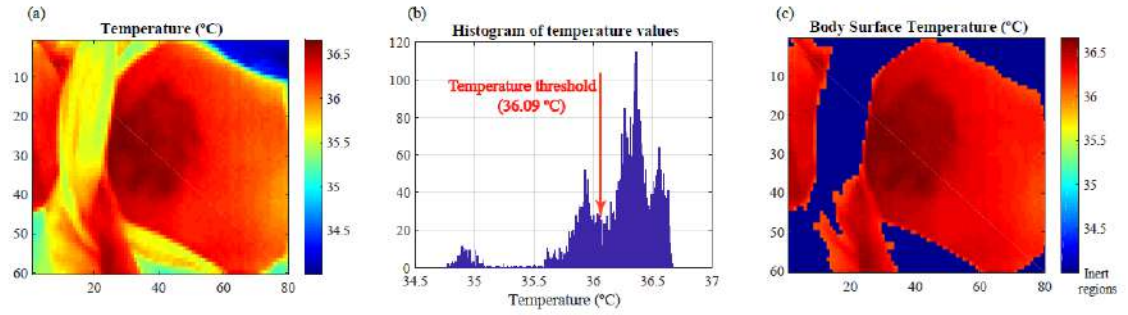


Figura 2.3: Ejemplo de segmentación basado en Otsu.

2.5.2. Operaciones morfológicas

Tras la segmentación inicial, es frecuente que la imagen binaria obtenida contenga pequeñas regiones aisladas, huecos o irregularidades que dificulten el análisis posterior. Para corregir estos problemas se utilizan operaciones morfológicas, técnicas de procesamiento basadas en la forma geométrica de los objetos presentes en la imagen.
Entre las operaciones más habituales se encuentran la erosión y la dilatación. La erosión reduce el tamaño de las regiones detectadas eliminando pequeños elementos aislados, mientras que la dilatación produce el efecto contrario, aumentando el tamaño de las regiones segmentadas. Combinando ambas operaciones es posible realizar procesos de apertura y cierre que permiten mejorar la calidad de las máscaras obtenidas y eliminar artefactos no deseados.
Estas técnicas son ampliamente utilizadas como etapa de preprocesado antes de la detección de contornos y la extracción de características geométricas.

2.5.3. Detección de contornos

Una vez segmentada la región de interés, resulta necesario delimitar sus fronteras con precisión. Para ello se emplean algoritmos de detección de contornos, cuyo objetivo consiste en localizar los píxeles que definen el borde de un objeto dentro de la imagen.
Los contornos permiten representar la forma de una región mediante un conjunto de puntos que describen su perímetro. A partir de esta información es posible calcular diferentes parámetros geométricos como el área ocupada, el perímetro, el diámetro aproximado o la posición de la región dentro de la imagen.
En los plugins desarrollados durante este trabajo se utilizan funciones de detección de contornos proporcionadas por OpenCV para identificar automáticamente las regiones segmentadas y obtener medidas cuantitativas asociadas a las mismas.

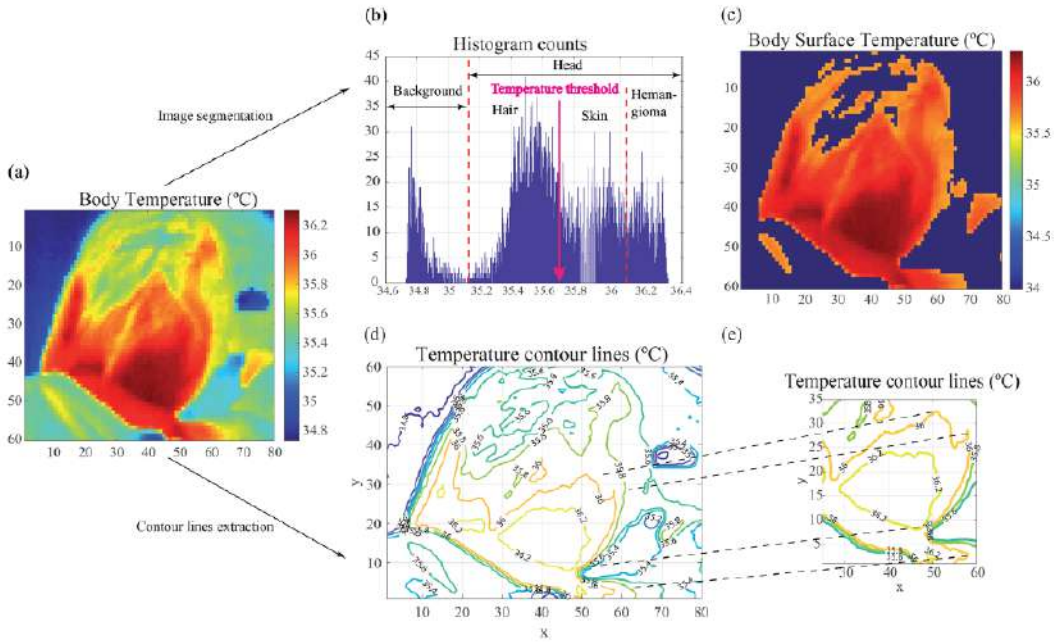


Figura 2.4: Ilustración de los pasos de procesamiento involucrados en la segmentación de imágenes. (a) Imagen termográfica original con valores de temperatura absolutos. (b) Histograma de la imagen termográfica original. Se resalta el umbral de temperatura (TH), determinado por el algoritmo de Otsu para la segmentación de la piel.(c) Imagen segmentada que muestra solo los valores de temperatura de la piel del paciente. (d) Líneas de contorno de temperatura extraídas de la imagen original. Las líneas de contorno con la temperatura más alta corresponden a la ubicación de la hiperplasia intimal (HI). (e) Detalle de las líneas de contorno de temperatura en la región afectada por la malformación.

Capítulo 3

Plataforma de adquisición y procesamiento de imágenes

Detallamos los distintos puntos de la plataforma utilizada y estudiada en este trabajo.

3.1. Arquitectura general del sistema

Este trabajo se desarrolla sobre una plataforma portátil de adquisición y procesamiento de imágenes diseñada para aplicaciones de termografía médica. Esta plataforma ha sido desarrollada por el Instituto de Microelectrónica de Sevilla (IMSE-CNM) con el objetivo de proporcionar una herramienta flexible y de bajo coste capaz de adquirir, procesar y visualizar imágenes térmicas en tiempo real dentro de entornos clínicos [1].
El sistema fue concebido para facilitar el estudio de patologías dermatológicas y anomalías vasculares mediante el uso combinado de imágenes térmicas e imágenes visibles. Para ello, integra diferentes sensores de adquisición junto con una unidad central de procesamiento encargada de ejecutar los algoritmos de análisis y mostrar los resultados obtenidos al usuario.
Una de las características más relevantes de la plataforma es su capacidad para adquirir simultáneamente información procedente de distintos sensores. Por una parte, las imágenes visibles permiten observar la apariencia externa de la lesión y su localización anatómica. Por otra, las imágenes térmicas aportan información relacionada con la distribución de temperatura superficial y la actividad vascular presente en la zona estudiada. La combinación de ambas fuentes de información proporciona una visión más completa del estado de la lesión y facilita su seguimiento clínico [4, 5].
La arquitectura del sistema ha sido diseñada siguiendo un enfoque modular, permitiendo incorporar nuevas funcionalidades software sin necesidad de modificar la estructura básica de la plataforma. Esta característica resulta especialmente importante en aplicaciones de investigación, donde los algoritmos de procesamiento evolucionan continuamente y deben adaptarse a nuevas necesidades clínicas.
La plataforma ha sido utilizada en estudios relacionados con el análisis de hemangiomas infantiles y otras anomalías vasculares en colaboración con el Área de Dermatología Pediátrica del Hospital Universitario Virgen del Rocío de Sevilla. En este contexto, el sistema permite obtener imágenes térmicas de forma rápida y no invasiva, facilitando la adquisición de información objetiva que puede complementar la evaluación realizada por los especialistas.
Como se muestra en la Figura 3.1, la plataforma integra los diferentes elementos hardware y software necesarios para la adquisición, procesamiento y visualización de imágenes térmicas y visibles. Esta arquitectura constituye la base sobre la que se desarrollan las nuevas funcionalidades implementadas en este trabajo.

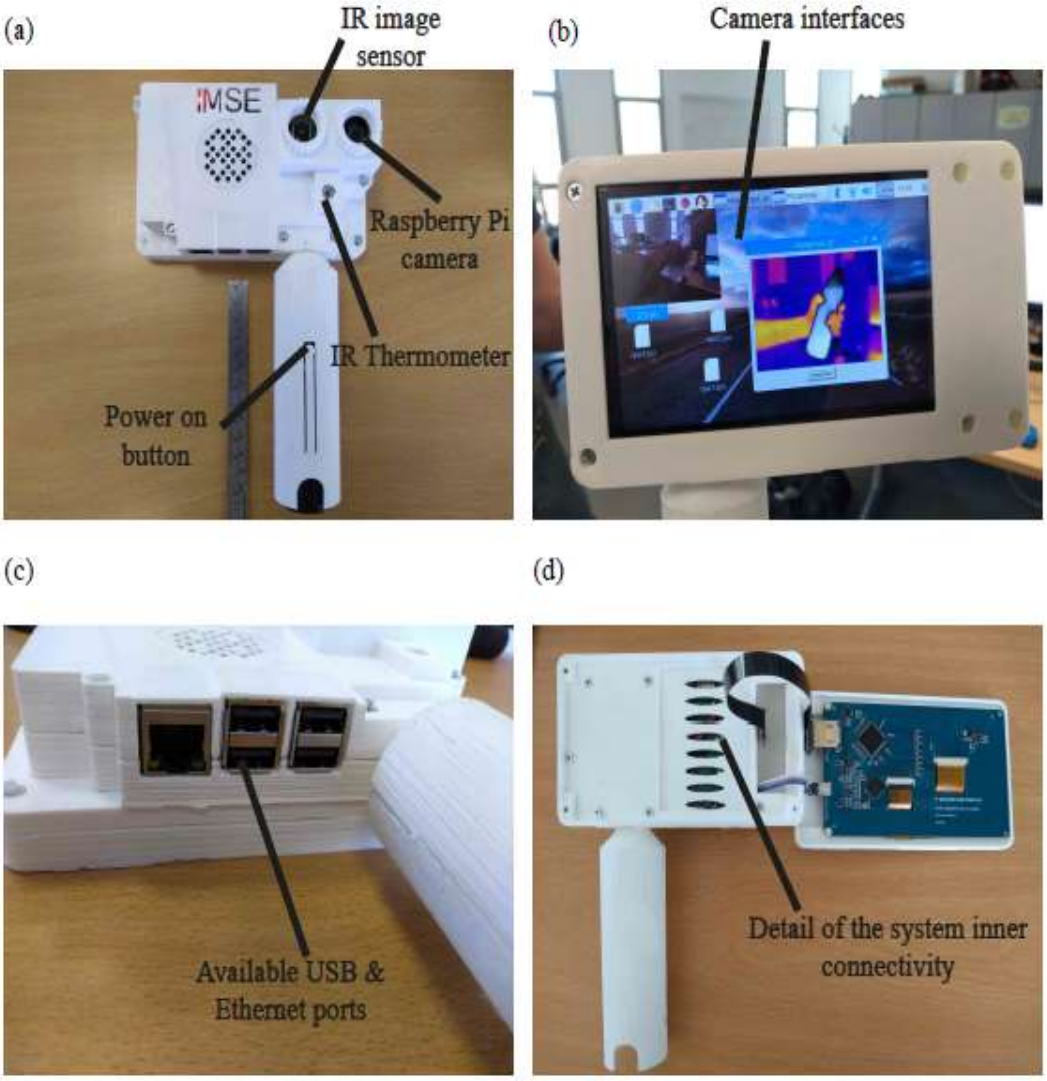


Figura 3.1: Implementación final del sistema. (a) Vista frontal. Se han resaltado los componentes principales del sistema. (b) Vista posterior que muestra las dos interfaces de sensores de imagen que operan en paralelo. (c) Vista lateral que muestra los puertos Ethernet y USB disponibles. (d) Detalle de la conectividad interna con la pantalla táctil.

3.2. Componentes hardware

La plataforma de adquisición y procesamiento de imágenes integra diversos elementos hardware encargados de la captura, procesamiento y visualización de la información obtenida durante las sesiones de análisis. La combinación de estos componentes permite construir un sistema portátil capaz de operar de forma autónoma en entornos clínicos y de investigación.

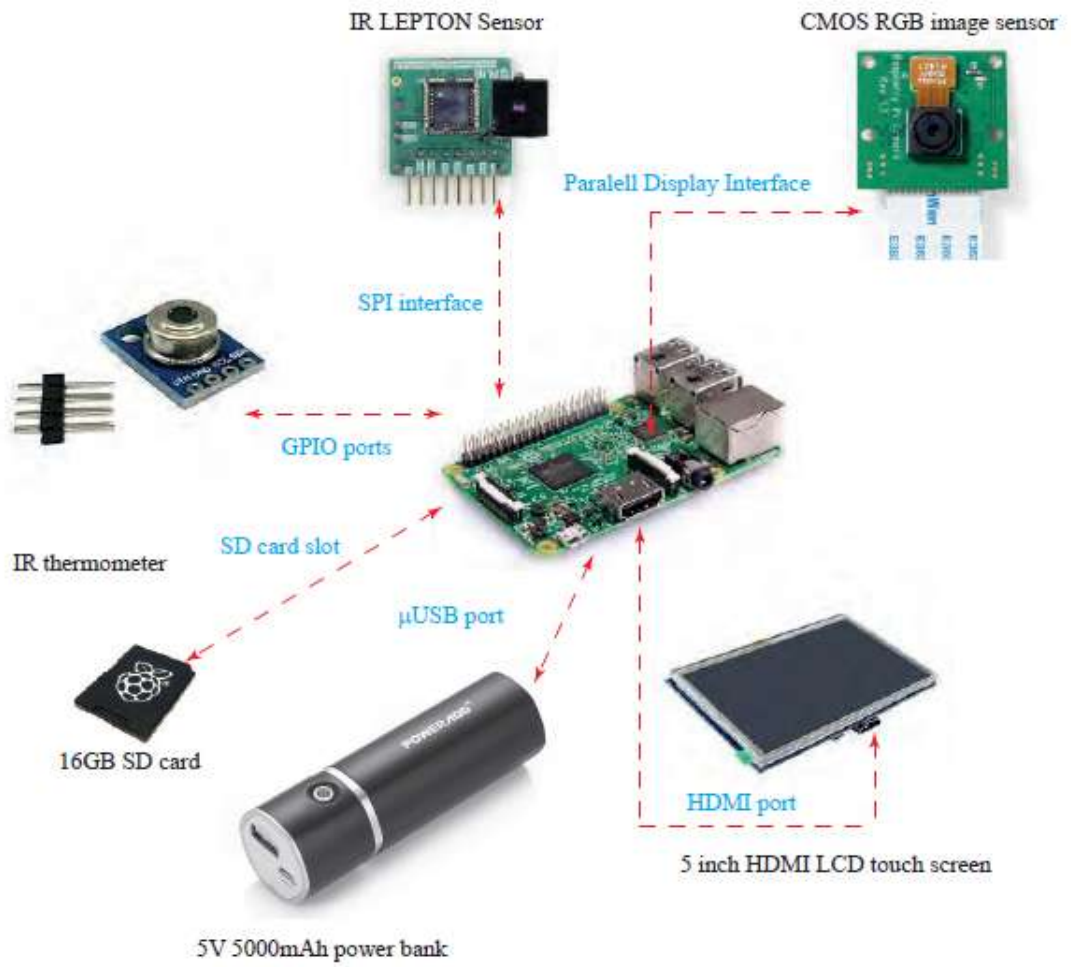


Figura 3.2: Elementos del sistema.

3.2.1. Unidad de procesamiento

La unidad central de procesamiento constituye el núcleo de la plataforma y es la encargada de coordinar el funcionamiento de todos los dispositivos conectados al sistema. Sobre ella se ejecutan los algoritmos de adquisición de imágenes, procesamiento digital y visualización de resultados.

La utilización de una plataforma basada en Raspberry Pi proporciona una solución compacta, de bajo coste y con suficiente capacidad de cálculo para ejecutar aplicaciones desarrolladas en lenguaje Python y bibliotecas de visión artificial como OpenCV. Además, su reducido consumo energético favorece la portabilidad del sistema y permite su utilización fuera de entornos de laboratorio [1].

3.2.2. Sistema de adquisición de imagen

La plataforma dispone de dos fuentes principales de información visual. La primera corresponde a una cámara convencional en el espectro visible, utilizada para capturar imágenes de la superficie cutánea y documentar visualmente la apariencia de las lesiones estudiadas.
La segunda corresponde a un sensor térmico que opera en la banda del infrarrojo lejano (LWIR). Este dispositivo permite obtener mapas de temperatura de la región analizada, proporcionando información relacionada con la actividad vascular y la distribución térmica de los tejidos. La combinación de ambas modalidades de imagen constituye una de las principales características de la plataforma desarrollada [1].

3.2.3. Sistema de medida de temperatura

La información proporcionada por la cámara térmica aporta además la temperatura, asi se facilita la calibración y validación de las medidas obtenidas durante la adquisición de imágenes. La utilización conjunta de sensores térmicos e instrumentos de medida específicos contribuye a mejorar la precisión de los datos empleados durante el análisis clínico.

3.2.4. Interfaz de usuario

La interacción con la plataforma se realiza mediante una pantalla táctil integrada en el propio dispositivo. Esta interfaz permite visualizar las imágenes adquiridas, acceder a las diferentes funcionalidades software disponibles y mostrar los resultados obtenidos por los algoritmos de procesamiento.
Gracias a esta solución, el sistema puede utilizarse de forma autónoma sin necesidad de equipos externos, facilitando su empleo en consultas médicas y sesiones de adquisición realizadas fuera del laboratorio.

3.2.5. Sistema de alimentación

Con el objetivo de garantizar la portabilidad del dispositivo, la plataforma incorpora un sistema de alimentación basado en batería. Esta característica permite realizar adquisiciones en diferentes entornos clínicos sin depender de conexiones permanentes a la red eléctrica.
La autonomía proporcionada por el sistema de alimentación contribuye a mejorar la flexibilidad de uso y facilita la realización de campañas de adquisición de datos en condiciones reales de funcionamiento.

3.3. Componentes software

Además de los elementos hardware descritos anteriormente, la plataforma dispone de una arquitectura software encargada de gestionar la adquisición de datos, el procesamiento de imágenes y la interacción con el usuario. Esta arquitectura ha sido diseñada con el objetivo de proporcionar un entorno flexible que permita incorporar nuevas funcionalidades de análisis sin necesidad de modificar la estructura básica del sistema [1].
El sistema opera sobre un entorno basado en Linux ejecutado en la Raspberry Pi, proporcionando los servicios necesarios para la gestión de dispositivos, la adquisición de imágenes y la ejecución de aplicaciones de procesamiento. La utilización de un sistema operativo ampliamente extendido facilita el desarrollo de nuevas herramientas y garantiza la compatibilidad con numerosas bibliotecas de software utilizadas en aplicaciones de visión artificial.

El desarrollo de las aplicaciones se realiza principalmente utilizando el lenguaje de programación Python. Este lenguaje presenta numerosas ventajas para aplicaciones científicas y de procesamiento de imagen, entre las que destacan su facilidad de programación, la disponibilidad de bibliotecas especializadas y la rápida integración de nuevos algoritmos.

Entre las herramientas software utilizadas destaca la biblioteca OpenCV, ampliamente empleada en aplicaciones de visión artificial y procesamiento digital de imagen. OpenCV proporciona funciones para filtrado, segmentación, detección de contornos, extracción de características y visualización de resultados, constituyendo la base de gran parte de los algoritmos implementados durante el desarrollo de este trabajo.

Una de las características más relevantes de la plataforma es su arquitectura modular basada en plugins. Este enfoque permite ampliar las funcionalidades disponibles mediante la incorporación de nuevos módulos software independientes, capaces de procesar las imágenes adquiridas y generar resultados específicos para cada aplicación.

Gracias a esta arquitectura, es posible desarrollar e integrar nuevos algoritmos sin necesidad de modificar el núcleo principal del sistema. Esta característica resulta especialmente adecuada en entornos de investigación, donde la incorporación continua de nuevas técnicas de análisis constituye una necesidad habitual.

El TFM se centra precisamente en el desarrollo de este tipo de extensiones software, implementando nuevos plugins orientados al procesamiento y análisis de imágenes térmicas obtenidas durante el seguimiento de hemangiomas infantiles. La modularidad de la plataforma permite que las herramientas desarrolladas puedan integrarse de forma sencilla dentro del entorno existente, favoreciendo su reutilización y futura ampliación.



Figura 3.3: Flujo general de software.

3.4. Flujo de funcionamiento de la plataforma

El funcionamiento de la plataforma puede dividirse en una serie de etapas consecutivas que abarcan desde la adquisición de datos hasta la presentación final de los resultados al usuario. Como

se muestra en la Figura 3.4, el sistema integra diferentes fuentes de información y herramientas software que permiten realizar el análisis de imágenes de forma automática y en tiempo real.
La primera etapa corresponde a la adquisición de datos. Durante esta fase se obtienen simultáneamente imágenes procedentes de la cámara visible y del sensor térmico infrarrojo. Esta información proporciona una descripción complementaria de la región analizada, combinando tanto características visuales como térmicas.
Una vez adquiridas las imágenes, los datos son transferidos a la unidad de procesamiento basada en Raspberry Pi. En esta etapa se realizan las tareas de preprocesado necesarias para preparar las imágenes para su posterior análisis. Estas operaciones pueden incluir normalización de intensidades, filtrado de ruido y conversión entre distintos formatos de representación.
Posteriormente, las imágenes son procesadas mediante algoritmos de visión artificial implementados utilizando Python y OpenCV. Durante esta fase se aplican técnicas de segmentación, detección de regiones de interés y extracción de características que permiten obtener información cuantitativa sobre la escena analizada.

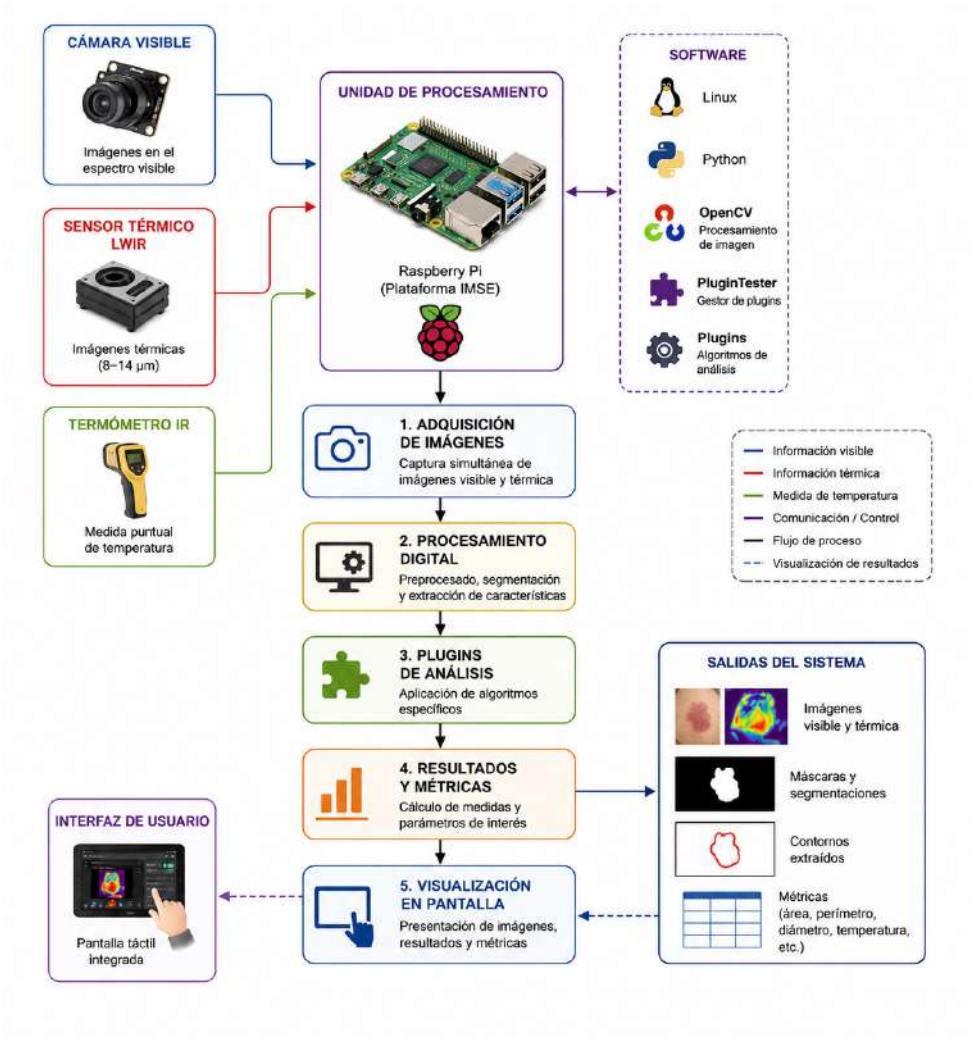


Figura 3.4: Flujo general de funcionamiento de la plataforma de adquisición y procesamiento de imágenes utilizada durante el desarrollo del trabajo.

La arquitectura modular del sistema permite incorporar algoritmos específicos mediante plugins independientes. Estos módulos software pueden implementar nuevas funcionalidades de análisis sin necesidad de modificar el núcleo principal de la plataforma, facilitando así la ampliación

continua de sus capacidades.
Los resultados generados por los plugins incluyen imágenes procesadas, máscaras de segmentación, contornos de las regiones detectadas y parámetros cuantitativos asociados a las mismas. Entre estos parámetros pueden encontrarse medidas geométricas, información térmica y otras características relevantes para el estudio clínico de las lesiones.
Finalmente, toda la información obtenida es presentada al usuario mediante la interfaz gráfica integrada en la plataforma. Esta visualización permite interpretar de forma sencilla los resultados generados por los algoritmos y facilita su utilización durante las sesiones de adquisición y análisis clínico.

3.5. Integración de nuevos algoritmos mediante plugins

Una de las características más relevantes de la plataforma utilizada en este trabajo es su arquitectura modular basada en plugins. Este enfoque permite ampliar las funcionalidades disponibles mediante la incorporación de módulos software independientes capaces de ejecutar algoritmos específicos sobre las imágenes adquiridas por el sistema.
Un plugin puede definirse como un componente software que se integra dentro de una aplicación principal para proporcionar nuevas funcionalidades sin modificar el núcleo del programa. Esta estrategia de diseño ofrece una elevada flexibilidad y facilita la incorporación progresiva de nuevas herramientas de análisis adaptadas a distintas necesidades clínicas o de investigación.
La plataforma desarrollada por el IMSE-CNM incorpora un entorno de ejecución que permite cargar dinámicamente diferentes plugins de procesamiento de imagen. Cada uno de estos módulos recibe como entrada las imágenes adquiridas por el sistema y genera como salida resultados específicos que pueden mostrarse directamente al usuario mediante la interfaz gráfica de la aplicación.
Gracias a esta arquitectura resulta posible desarrollar nuevas funcionalidades de forma independiente al resto del sistema. De este modo, los algoritmos pueden implementarse, probarse y modificarse sin afectar al funcionamiento general de la plataforma, favoreciendo la reutilización del código y simplificando las tareas de mantenimiento y actualización.
En el contexto de este proyecto, esta capacidad de extensión ha permitido desarrollar nuevos plugins orientados al análisis de imágenes térmicas de lesiones vasculares. Los módulos implementados incorporan técnicas de segmentación, detección de contornos y extracción de características geométricas con el objetivo de obtener información cuantitativa a partir de las regiones de interés detectadas.
Los capítulos siguientes describen detalladamente el diseño, implementación y validación de los plugins desarrollados, así como los resultados obtenidos tras su integración en la plataforma existente.
Como se observa en la Figura 3.3, los plugins constituyen el nivel de extensión de la plataforma, permitiendo incorporar nuevas funcionalidades sin modificar el núcleo principal del sistema.

Capítulo 4

Desarrollo e implementación de plugins

El presente capítulo describe el desarrollo de las funcionalidades software implementadas durante la realización de este trabajo. Partiendo de la arquitectura basada en plugins proporcionada por la plataforma de adquisición y procesamiento de imágenes, se diseñaron e integraron diferentes módulos orientados al análisis de imágenes térmicas de lesiones vasculares.

El trabajo realizado se centró en el estudio e implementación de técnicas de procesamiento digital de imagen capaces de identificar regiones de interés, extraer características geométricas y proporcionar información cuantitativa relevante para el seguimiento clínico de las lesiones. Para ello se emplearon herramientas como Python y OpenCV, aprovechando la flexibilidad ofrecida por la arquitectura modular del sistema.

En las siguientes secciones se describen el entorno de desarrollo utilizado, la metodología seguida durante la implementación y las distintas funcionalidades incorporadas a la plataforma mediante los plugins desarrollados.

4.1. Objetivos de desarrollo

Una vez analizada la plataforma de adquisición y procesamiento de imágenes descrita en los capítulos anteriores, se planteó el desarrollo de nuevas funcionalidades software orientadas al análisis de imágenes térmicas obtenidas durante el seguimiento de lesiones vasculares.

El objetivo principal consistió en diseñar e implementar nuevos plugins capaces de integrarse dentro de la arquitectura existente, aprovechando las capacidades de procesamiento proporcionadas por la plataforma y manteniendo la compatibilidad con el entorno software desarrollado previamente.

Para ello se estudiaron diferentes técnicas de procesamiento digital de imagen aplicables al análisis de imágenes térmicas, prestando especial atención a los métodos de segmentación automática, detección de contornos y extracción de características geométricas. Estas técnicas permiten identificar regiones de interés dentro de la imagen y obtener parámetros cuantitativos susceptibles de ser utilizados durante el seguimiento clínico de las lesiones.

El desarrollo realizado se centró en la implementación de algoritmos sencillos, interpretables y compatibles con el funcionamiento en tiempo real de la plataforma. Asimismo, se buscó que las herramientas desarrolladas pudieran servir como base para futuras ampliaciones orientadas a la incorporación de nuevas técnicas de análisis de imagen.

4.1.1. Plugin de eliminación del fondo térmico

Como primer desarrollo realizado se implementó un plugin(ThermalBackgroundRemoval.py) destinado a la eliminación del fondo térmico de las imágenes adquiridas por la plataforma. El objetivo principal de esta herramienta consiste en aislar automáticamente la región de mayor interés térmico, eliminando aquellas zonas de la escena que no aportan información relevante para el análisis posterior.

El algoritmo desarrollado parte de la imagen proporcionada por el sensor infrarrojo y realiza un proceso de segmentación basado en percentiles de intensidad. En lugar de emplear un umbral fijo, el plugin calcula automáticamente el nivel de intensidad correspondiente a las regiones con mayor temperatura relativa dentro de cada imagen, permitiendo adaptar el proceso de segmentación a diferentes condiciones de adquisición.

Una vez obtenida la máscara inicial, se aplican diferentes operaciones morfológicas para eliminar pequeñas regiones espurias y suavizar los contornos de la zona segmentada. Posteriormente, se selecciona la región candidata de mayor interés y se identifican en su interior aquellos focos que presentan temperaturas superiores al resto de la lesión.

Finalmente, el plugin genera una nueva imagen en la que únicamente se conserva la región térmica detectada, representándose el resto de la escena mediante un fondo negro. Sobre la imagen resultante se superpone el contorno de la región principal, un rectángulo delimitador y marcadores correspondientes a los focos de mayor temperatura. Además, se calcula información cuantitativa básica, como el área de la región segmentada expresada en píxeles y una estimación de la temperatura media de la lesión.

La principal utilidad de este plugin reside en simplificar la visualización de la información térmica relevante, reduciendo la influencia del fondo y facilitando tanto la inspección visual como el posterior procesamiento mediante algoritmos más avanzados. Asimismo, constituye una etapa de preprocesado sobre la que pueden desarrollarse futuras herramientas de análisis orientadas a la caracterización automática de lesiones vasculares.

A modo de ejemplo de visualización, usaremos la imagen de un paciente donde se observa la lesion en la parte lateral del vientre.



Figura 4.1: Foto de paciente en espectro visible mostrando la sospecha hemangioma.

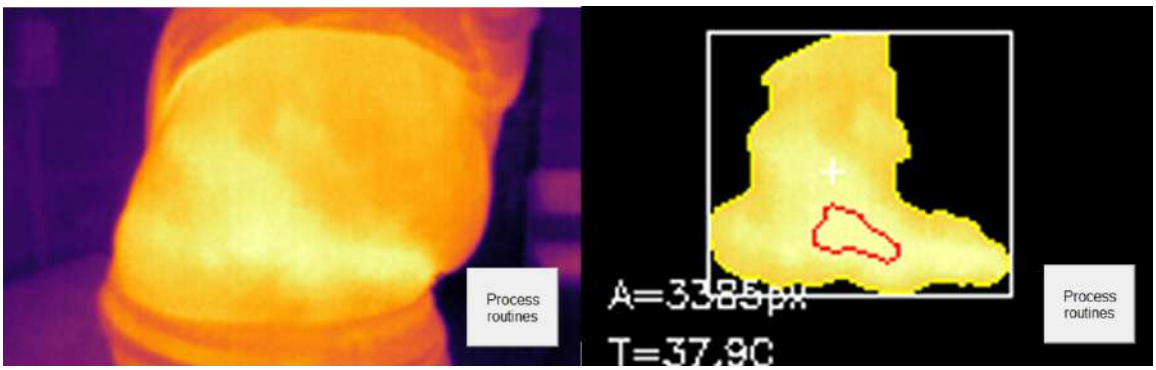


Figura 4.2: Imagen en infrarrojo y con el plugin aplicado.

Mostramos la figura con la captura en el espectro IR y con el plugin aplicado donde elimina el fondo y marca las zonas más altas de temperatura. Podemos comprobar como el plugin detecta y señala los puntos más calientes que en este caso son los de todo más brillantes. Además también elimina el fondo de la escena delimitando la región de interés. Muestra la región en pixeles y la temperatura media estimada.

Funcionamiento del algoritmo

El funcionamiento del plugin comienza con la recepción de la imagen infrarroja de entrada. En primer lugar, la imagen se transforma a un formato adecuado para su procesamiento posterior. Para ello, se asegura que la imagen se encuentra en formato de 16 bits y posteriormente se convierte de forma robusta a una imagen de 8 bits, evitando que valores extremos afecten de forma excesiva a la visualización y segmentación.

```
img_u16 = self._ensure_u16(img_in)
img8 = self._robust_u8(img_u16)
```

A partir de esta imagen normalizada se genera una representación en falso color, que permite visualizar de forma más intuitiva las diferencias de temperatura presentes en la escena.

```
thermal = cv.applyColorMap(img8, cv.COLORMAP_INFERNO)
```

La etapa principal del algoritmo consiste en segmentar la región caliente de interés. Para ello, no se utiliza un umbral fijo, sino un umbral adaptativo basado en percentiles de intensidad. De esta forma, el plugin selecciona automáticamente las zonas más calientes de cada imagen, adaptándose a las condiciones concretas de adquisición.

```
warm_thr = np.percentile(blur, self.hot_percentile)
_, warm = cv.threshold(blur, warm_thr, 255, cv.THRESH_BINARY)
```

Además, se calcula un segundo umbral más restrictivo para identificar los focos internos de mayor temperatura dentro de la región principal.

```
seed_thr = np.percentile(blur, self.hotspot_percentile)
_, seeds = cv.threshold(blur, seed_thr, 255, cv.THRESH_BINARY)
```

Tras obtener las máscaras binarias, se aplican operaciones morfológicas de apertura y cierre. La apertura permite eliminar pequeñas regiones aisladas debidas al ruido, mientras que el cierre ayuda a rellenar huecos y suavizar la región detectada.

```
warm = cv.morphologyEx(warm, cv.MORPH_OPEN, kernel, iterations=1)
warm = cv.morphologyEx(warm, cv.MORPH_CLOSE, kernel, iterations=3)
```

Posteriormente, se analizan las componentes conexas de la máscara resultante. El objetivo es conservar únicamente la región candidata más relevante, descartando regiones pequeñas o zonas que no contienen focos claramente calientes.

```
num, labels, stats, _ = cv.connectedComponentsWithStats(warm, connectivity=8)
```

Una vez obtenida la máscara final, el plugin elimina visualmente el fondo de la escena. Para ello, se crea una imagen negra y únicamente se copian sobre ella los píxeles pertenecientes a la región térmica detectada.

```
out = np.zeros_like(thermal)
out[mask == 255] = thermal[mask == 255]
```

Sobre la región segmentada se calcula el contorno principal mediante OpenCV. Este contorno permite delimitar visualmente la zona detectada y obtener información geométrica asociada a la misma.

```
contours, _ = cv.findContours(mask, cv.RETR_EXTERNAL, cv.CHAIN_APPROX_SIMPLE)
cnt = max(contours, key=cv.contourArea)
cv.drawContours(out, [cnt], -1, (0, 255, 255), 1)
```

Además del contorno, se dibuja un rectángulo de localización que facilita identificar la posición de la región detectada dentro de la imagen.

```
x, y, w, h = cv.boundingRect(cnt)
cv.rectangle(out, (x, y), (x + w, y + h), (255, 255, 255), 1)
```

El plugin también calcula el área de la región segmentada contando el número de píxeles pertenecientes a la máscara. Esta medida permite obtener una primera estimación cuantitativa del tamaño de la zona detectada.

```
area_px = int(np.count_nonzero(mask))
```

Para estimar la temperatura media de la región, se calcula la intensidad media de los píxeles contenidos en la máscara y posteriormente se transforma a temperatura mediante una función de conversión basada en la calibración utilizada por el sistema.

```
lesion_i = img8[mask == 255].astype(np.float32)
imean = float(np.mean(lesion_i)) if lesion_i.size else 0.0
t_mean = self.IR_to_temp(imean)
```

Finalmente, el plugin muestra sobre la imagen de salida los valores calculados, incluyendo el área de la región detectada y la temperatura media estimada.

```
text1 = f"A={area_px}px"
text2 = f"T={t_mean:.1f}C"
```

De este modo, el algoritmo permite transformar una imagen térmica completa en una representación más sencilla, donde se elimina el fondo de la escena y se resaltan únicamente las regiones térmicamente relevantes. Esto facilita la inspección visual de la lesión y proporciona una primera estimación cuantitativa de sus características.



Figura 4.3: Imagen de otro paciente en visible.

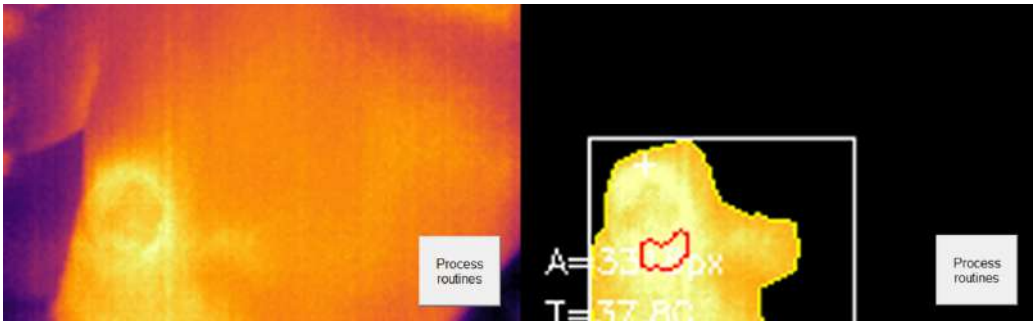


Figura 4.4: Imagen de otro paciente en IR.

4.1.2. Plugin de clasificación térmica mediante K-means

Como segundo desarrollo se implementó un plugin orientado a la clasificación automática de las diferentes regiones térmicas presentes en la imagen infrarroja mediante el algoritmo de agrupamiento K-means. A diferencia del plugin anterior, cuyo objetivo principal consistía en eliminar el fondo de la escena para aislar la región de interés, este desarrollo busca proporcionar una representación simplificada de la distribución térmica de la imagen.
El algoritmo clasifica automáticamente los píxeles en diferentes grupos según su intensidad térmica relativa, permitiendo diferenciar regiones correspondientes a temperatura ambiente, zonas templadas y áreas con mayor temperatura. Esta representación facilita la interpretación visual de la escena y permite identificar rápidamente aquellas regiones que presentan un comportamiento térmico diferenciado.
Además de la clasificación general, el plugin incorpora un procedimiento adicional para detectar los focos de mayor temperatura dentro de la imagen. Estas zonas se representan mediante contornos y rellenos en color rojo, destacando visualmente las regiones que podrían presentar un mayor interés desde el punto de vista clínico.
La Figura 4.5 muestra el funcionamiento del plugin sobre una imagen térmica real. A la izquierda se observa la imagen infrarroja original adquirida por la plataforma, mientras que a la derecha se presenta el resultado obtenido tras aplicar el algoritmo de clasificación. Puede apreciarse cómo la imagen queda dividida en diferentes regiones térmicas y cómo los focos de mayor temperatura aparecen resaltados en rojo. Además, se incorpora una pequeña leyenda que facilita la interpretación de cada una de las clases obtenidas.

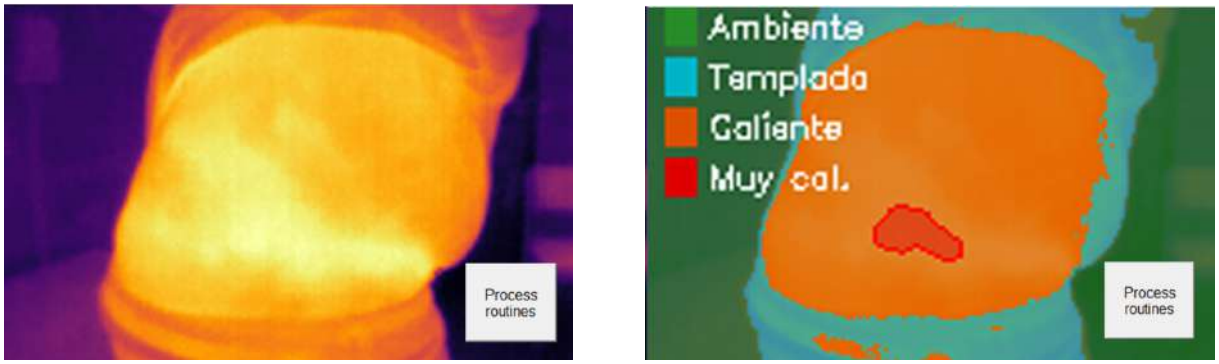


Figura 4.5: Funcionamiento del plugin de clasificación térmica mediante K-means. Izquierda: imagen infrarroja original. Derecha: resultado obtenido tras la clasificación de regiones térmicas y detección de focos de mayor temperatura.

Funcionamiento del algoritmo

El funcionamiento del plugin comienza con la recepción de la imagen térmica proporcionada por el sensor infrarrojo. En primer lugar, la imagen se normaliza al rango de 8 bits con el objetivo de homogeneizar los valores de intensidad y facilitar las etapas posteriores de procesamiento.

```
img8 = cv.normalize(img_in, None, 0, 255,
                    cv.NORM_MINMAX).astype(np.uint8)
```

Una vez normalizada la imagen, se aplica el algoritmo de agrupamiento K-means. Para ello, cada píxel de la imagen se considera una muestra independiente y se clasifica según su intensidad térmica. En este desarrollo se utilizan cuatro grupos o clases, permitiendo diferenciar automáticamente distintas regiones de temperatura presentes en la escena.

```
pixels = img8.reshape((-1, 1)).astype(np.float32)

_, labels, centers = cv.kmeans(
    pixels,
    self.K,
    None,
    criteria,
    attempts=10,
    flags=cv.KMEANS_RANDOM_CENTERS,
)
```

Tras finalizar el proceso de agrupamiento, los centros obtenidos se ordenan desde las temperaturas más bajas hasta las más elevadas. Esto permite asignar de forma consistente un significado físico a cada grupo, distinguiendo regiones correspondientes a temperatura ambiente, zonas templadas y regiones calientes.

```
order = np.argsort(centers)
```

Posteriormente, cada clase térmica se representa mediante un color específico. Para ello se define un mapa de colores que facilita la interpretación visual de la imagen segmentada, asignando colores diferentes a cada nivel de temperatura.

```
color_map_bgr = [
    (40,140,40),
    (40,140,40),
    (200,180,0),
    (0,80,220)
]
```

Con el objetivo de conservar parte de la información visual original, la imagen clasificada se combina con la imagen térmica en falso color utilizando una mezcla ponderada. De este modo se obtiene una representación más intuitiva en la que pueden apreciarse simultáneamente la distribución térmica y la clasificación realizada por el algoritmo.

```
output = cv.addWeighted(img_color,
                        0.30,
                        img_kmeans,
                        0.70,
                        0)
```

Como etapa adicional, el plugin incorpora un procedimiento para detectar los focos de mayor temperatura presentes en la imagen. Para ello se seleccionan únicamente los píxeles pertenecientes al percentil superior de intensidad térmica y, tras aplicar operaciones morfológicas de apertura y cierre para eliminar pequeñas regiones de ruido, se obtienen las zonas consideradas como puntos calientes.

```
hotspot_mask = self._detect_hotspots(img8)
```

Las regiones detectadas se representan sobre la imagen final mediante contornos y rellenos en color rojo, permitiendo identificar visualmente aquellas áreas que presentan una temperatura significativamente superior al resto de la escena.

```
cv.drawContours(output,
                [cnt],
                -1,
                (0,0,255),
                1)
```

Finalmente, el plugin incorpora una leyenda que indica el significado de cada una de las clases térmicas representadas. De este modo, el usuario puede interpretar de forma rápida la distribución de temperaturas observada y localizar fácilmente las regiones con mayor interés clínico.
El resultado final es una imagen clasificada en diferentes niveles térmicos donde se distinguen las zonas de temperatura ambiente, regiones templadas, áreas calientes y focos de máxima temperatura. Esta representación proporciona una visión simplificada de la escena y facilita la identificación visual de posibles anomalías térmicas susceptibles de ser analizadas mediante técnicas más avanzadas.

4.1.3. Plugin 3: Análisis de diferencia térmica

El tercer plugin desarrollado tiene como objetivo analizar el comportamiento térmico interno de la principal región caliente detectada en la imagen infrarroja. A diferencia de los plugins anteriores, que se centran principalmente en la segmentación y clasificación de la escena, este desarrollo incorpora un análisis cuantitativo de la distribución de temperaturas dentro de la lesión.
En primer lugar, el algoritmo identifica automáticamente la región caliente de mayor interés mediante un proceso de segmentación basado en percentiles de intensidad. Posteriormente, calcula la temperatura media estimada de dicha región y localiza el punto de máxima temperatura presente en su interior. A partir de estos dos valores se obtiene la diferencia térmica ($Delta$), parámetro que proporciona una estimación del contraste existente entre la temperatura media de la lesión y el foco de mayor actividad térmica.
Además del cálculo de estos parámetros, el plugin resalta visualmente los focos térmicos más intensos mediante contornos de color rojo y marca el punto de máxima temperatura dentro de la región segmentada. Sobre la imagen resultante también se muestran la temperatura media de la zona ($Tzona$), la temperatura máxima detectada ($Tmax$) y la diferencia térmica entre ambas ($Delta$), facilitando una interpretación inmediata de la distribución térmica observada.
La Figura 4.7 muestra el resultado obtenido tras aplicar el plugin sobre una imagen térmica real. Puede observarse cómo el algoritmo delimita automáticamente la región caliente, identifica los focos internos de mayor temperatura y presenta en pantalla los principales parámetros térmicos calculados durante el análisis.

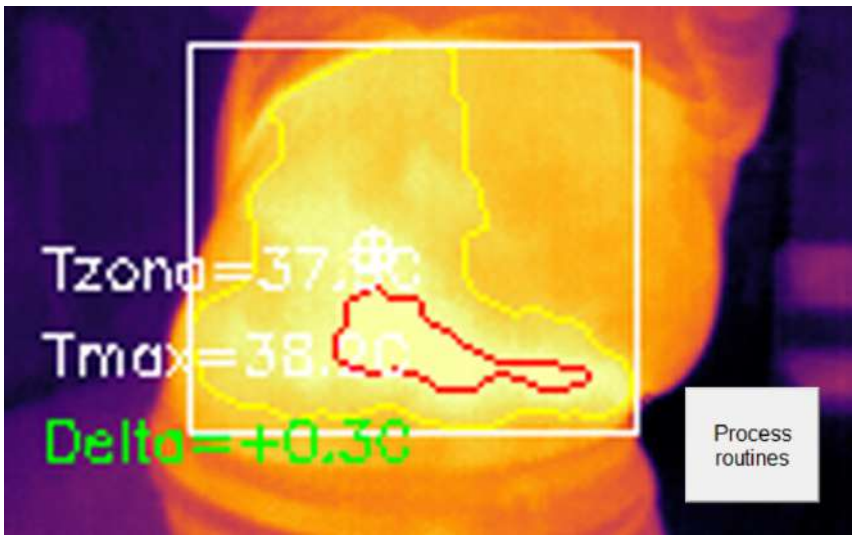


Figura 4.6: Resultado del plugin de análisis de diferencia térmica.

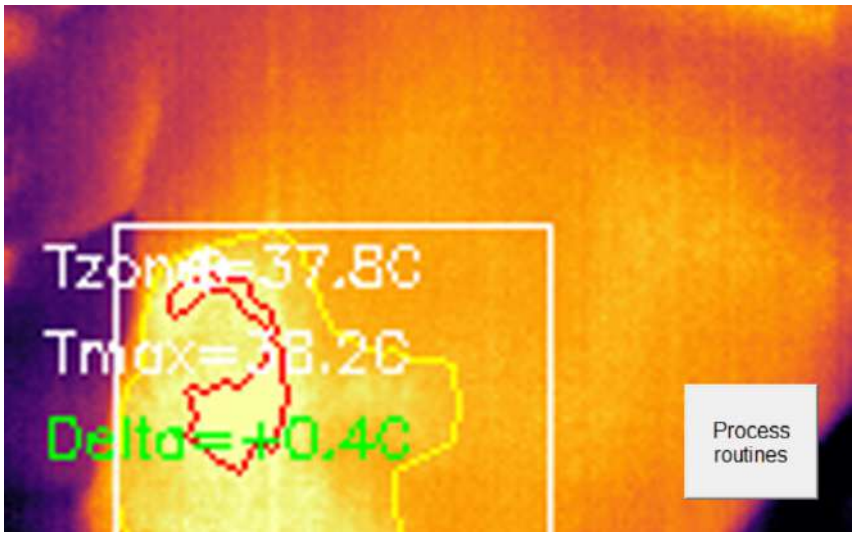


Figura 4.7: Resultado del plugin de análisis de diferencia térmica en otro paciente.

Funcionamiento del algoritmo

El funcionamiento del plugin comienza con la recepción de la imagen térmica procedente del sensor infrarrojo. En primer lugar, la imagen se convierte a un formato de 8 bits mediante una normalización robusta, lo que permite mejorar la visualización y homogeneizar el rango dinámico utilizado durante el procesamiento.

```
img8 = self.robust_u8(img_in)
output = cv.applyColorMap(img8, cv.COLORMAP_INFERNO)
```

A continuación, el algoritmo segmenta automáticamente la región caliente principal utilizando un umbral adaptativo basado en percentiles de intensidad. Para ello se genera una máscara binaria que posteriormente es refinada mediante operaciones morfológicas de apertura y cierre, eliminando pequeñas regiones de ruido y conservando únicamente la zona de interés.

```
mask = self.segment_hot_zone(img8)
```

Una vez obtenida la máscara definitiva, se localiza el contorno principal de la región segmentada y se calcula su rectángulo delimitador. Esta información permite representar gráficamente la localización de la lesión dentro de la imagen.

```
contours, _ = cv.findContours(mask,
                              cv.RETR_EXTERNAL,
                              cv.CHAIN_APPROX_SIMPLE)
```

Posteriormente se calcula la temperatura media estimada de toda la región segmentada. Para ello se obtiene el valor medio de intensidad de todos los píxeles pertenecientes a la máscara y posteriormente se transforma a temperatura mediante la función de calibración implementada en el plugin.

```
zone_values = img8[mask == 255]
temp_zone = self.IR_to_temp(zone_mean_i)
```

Como siguiente etapa se localiza el punto de máxima temperatura existente dentro de la región de interés. A partir de dicho valor se calcula la temperatura máxima y posteriormente la diferencia térmica respecto a la temperatura media de la zona.

```
_, max_i, _, max_loc = cv.minMaxLoc(masked_img)

temp_max = self.IR_to_temp(float(max_i))
delta_t = temp_max - temp_zone
```

Además del punto máximo, el algoritmo identifica pequeñas regiones cuya temperatura se encuentra próxima al máximo detectado. Estas zonas se consideran focos térmicos internos y se representan mediante contornos rojos sobre la imagen final, facilitando su identificación visual.

```
hotspot_mask = self.segment_hotspots_inside_zone(img8,
                                                 mask)
```

Finalmente, el plugin incorpora sobre la imagen los principales parámetros obtenidos durante el análisis: la temperatura media de la región ($Tzona$), la temperatura máxima detectada ($Tmax$) y la diferencia térmica entre ambas ($Delta$). De este modo, el usuario dispone de una representación visual de la lesión junto con una información cuantitativa que permite evaluar la existencia de focos térmicos especialmente activos y comparar diferentes adquisiciones realizadas durante el seguimiento clínico.

4.2. Prototipos explorados y comparativa

Durante el desarrollo del estudio para los algoritmos, no únicamente se implementaron los plugins presentados en la versión final de la plataforma. Con el objetivo de estudiar distintas estrategias de análisis de imágenes térmicas, se desarrollaron diversos prototipos basados en técnicas de procesamiento digital de imagen y visión artificial.
Cada uno de estos desarrollos fue evaluado atendiendo a diferentes criterios, entre ellos la estabilidad del algoritmo, la facilidad de interpretación de los resultados, la utilidad de la información proporcionada al usuario y la posibilidad de integración dentro de la plataforma existente. Este

proceso permitió identificar las soluciones más adecuadas para el entorno clínico y descartar aquellas que, aun siendo funcionales desde el punto de vista técnico, no aportaban una mejora significativa respecto a las herramientas finalmente seleccionadas.
En las siguientes subsecciones se describen algunos de los prototipos desarrollados durante esta fase de experimentación, así como las razones que motivaron su inclusión o descarte dentro de la versión final de la plataforma.

4.2.1. Prototipo 1: Detección de bordes mediante Canny

Como parte de la fase inicial de experimentación se desarrolló un primer prototipo basado en la detección de bordes utilizando el algoritmo de Canny, uno de los métodos más empleados en visión artificial para identificar cambios bruscos de intensidad dentro de una imagen. El objetivo de este desarrollo era comprobar si la delimitación automática de los contornos presentes en las imágenes térmicas podía facilitar la identificación de las regiones correspondientes a las lesiones.
El resultado obtenido se muestra en la Figura 4.8. Puede observarse que el algoritmo detecta correctamente un gran número de bordes presentes en la escena, incluyendo tanto los límites del brazo como pliegues de la ropa, sombras y otras estructuras no relacionadas con la lesión. Como consecuencia, la imagen obtenida presenta una elevada cantidad de información que dificulta la identificación de las regiones térmicamente relevantes.
Por este motivo, aunque el algoritmo funciona correctamente desde el punto de vista técnico, se consideró que su utilidad clínica era limitada para el objetivo de este trabajo. En consecuencia, se optó por desarrollar técnicas basadas en segmentación térmica, que permiten centrar el análisis en la distribución de temperaturas y obtener información cuantitativa más relevante para el estudio de lesiones vasculares.



Figura 4.8: Resultado obtenido mediante el algoritmo de detección de bordes de Canny sobre una imagen térmica.

Funcionamiento del algoritmo

El algoritmo comienza convirtiendo la imagen térmica original a un formato de 8 bits compatible con las funciones de OpenCV. Posteriormente se aplica una normalización de intensidades para

aprovechar todo el rango dinámico disponible y una representación en falso color que facilita la visualización de la imagen.

```
cv_img = self.from_cv16_to_cv8(img_in)
```

Antes de realizar la detección de bordes se aplica un filtro Gaussiano para reducir el ruido presente en la imagen. Este preprocesado evita que pequeñas variaciones de intensidad generen contornos espurios durante la etapa siguiente.

```
cv_img_blur = cv.GaussianBlur(cv_img, (3,3), 0)
```

Finalmente se utiliza el detector de bordes de Canny, que localiza automáticamente las zonas donde existen cambios bruscos de intensidad entre píxeles vecinos. El algoritmo emplea dos umbrales para distinguir entre bordes fuertes y débiles, obteniendo como resultado una imagen binaria en la que únicamente aparecen representados los contornos detectados.

```
edges = cv.Canny(
    image=cv_img_blur,
    threshold1=50,
    threshold2=150
)
```

Aunque el algoritmo permite detectar correctamente numerosos contornos presentes en la escena, el resultado obtenido incluye tanto los límites de la región de interés como otros elementos del entorno, generando una imagen con una elevada densidad de bordes. Esta característica dificulta la interpretación clínica de la información térmica y motivó que este prototipo no fuera seleccionado para formar parte de la versión final de la plataforma.

4.2.2. Prototipo 2: Detección de contornos

Como segunda alternativa se desarrolló un plugin basado en la detección automática de contornos mediante las herramientas proporcionadas por OpenCV. El objetivo de este desarrollo era delimitar las principales regiones presentes en la imagen térmica y obtener una primera estimación de la temperatura media asociada a cada una de ellas.
Para ello, el algoritmo realiza un preprocesado de la imagen, detecta los contornos de mayor tamaño y calcula la temperatura media de la región considerada más relevante. El resultado obtenido se muestra en la Figura 4.9, donde puede observarse el contorno principal superpuesto sobre la imagen térmica junto con la temperatura media estimada de la región seleccionada.
Aunque el algoritmo consigue detectar correctamente determinadas regiones de la imagen y calcular un parámetro térmico asociado, los contornos obtenidos dependen principalmente de las variaciones de intensidad presentes en la escena y no necesariamente de la distribución térmica de la lesión. Como consecuencia, en numerosas ocasiones aparecen delimitadas estructuras ajenas a la región de interés, reduciendo la utilidad clínica del resultado obtenido.
Por este motivo, este prototipo fue considerado una aproximación inicial al problema, sirviendo como base para el desarrollo posterior de algoritmos basados en segmentación térmica, los cuales proporcionan una representación más estable y una mejor caracterización de las lesiones estudiadas.

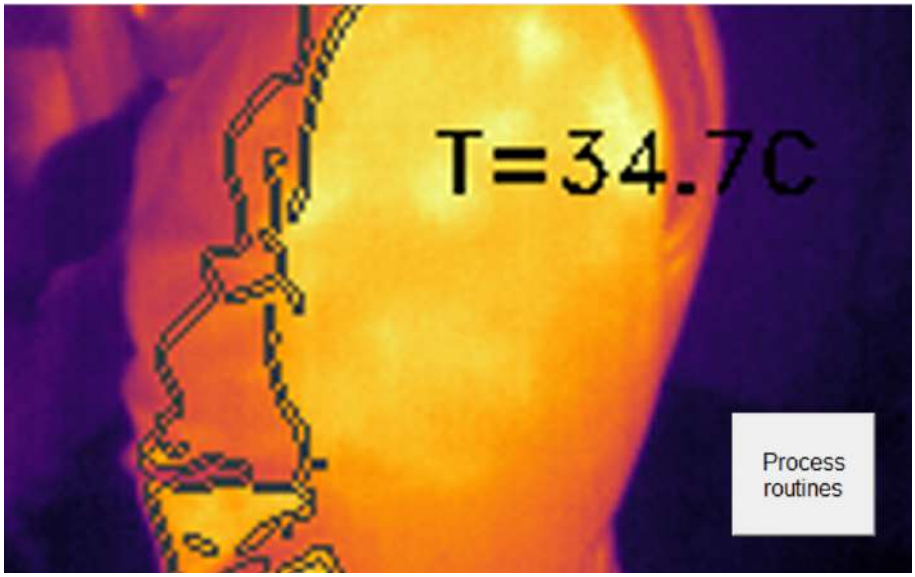


Figura 4.9: Resultado obtenido mediante el plugin de detección de contornos. El algoritmo delimita la región de mayor tamaño detectada y calcula la temperatura media estimada de dicha zona.

Funcionamiento del algoritmo

El algoritmo comienza convirtiendo la imagen térmica original a un formato de 8 bits compatible con las funciones de OpenCV. Posteriormente se aplica un filtro Gaussiano para reducir el ruido presente en la imagen y mejorar la estabilidad de la detección de contornos.

```
cv_img = self.from_cv16_to_cv8(img_in)

img_pre = cv.GaussianBlur(cv_img,
                          (5,5),
                          0)
```

A continuación, la imagen se transforma a escala de grises y se aplica un umbral de intensidad con el objetivo de eliminar regiones poco relevantes. Posteriormente se realiza una reducción de niveles de gris para simplificar la imagen antes de la detección de bordes.

```
img_pre = cv.cvtColor(img_pre,
                      cv.COLOR_BGR2GRAY)

_, img_pre = cv.threshold(img_pre,
                          115,
                          250,
                          cv.THRESH_TOZERO)

img_pre = self.color_reduce(img_pre,
                            32)
```

Una vez preprocesada la imagen, se aplica el detector de bordes de Canny y posteriormente una operación de dilatación para conectar pequeños segmentos de borde próximos entre sí, facilitando la detección de regiones cerradas.

```
img_canny = cv.Canny(img_pre,
                     50,
                     70)

img_pre = cv.dilate(img_canny,
                    kernel)
```

Los contornos obtenidos se analizan individualmente descartando aquellas regiones cuya superficie resulta demasiado pequeña o cuya geometría no presenta suficiente consistencia. Entre todos los contornos válidos se selecciona el de mayor área, que se considera la región principal de la imagen.

```
contours, hierarchy = cv.findContours(
    img_pre,
    cv.RETR_TREE,
    cv.CHAIN_APPROX_SIMPLE
)
```

Finalmente, el algoritmo calcula la temperatura media de la región seleccionada a partir de la intensidad media de los píxeles contenidos en su interior y muestra este valor directamente sobre la imagen procesada.

```
temp = self.get_area_mean_temp(cv_img,
                               to_draw_contour)
```

Aunque el algoritmo proporciona una delimitación automática de las principales regiones de la imagen, la información obtenida depende en gran medida de la presencia de contornos visibles y no de la distribución térmica real de la lesión. Por este motivo, el prototipo fue finalmente sustituido por algoritmos basados en segmentación térmica, que ofrecen resultados más robustos y una interpretación más adecuada para el análisis clínico de hemangiomas.

4.2.3. Prototipo 3: Segmentación mediante Watershed

Como parte del proceso de investigación también se desarrolló un prototipo basado en el algoritmo de segmentación Watershed. Esta técnica es ampliamente utilizada en procesamiento digital de imagen para separar regiones conectadas a partir de su relieve topográfico y constituye una de las herramientas más empleadas en aplicaciones biomédicas de segmentación.
El objetivo de este desarrollo era evaluar la capacidad del algoritmo para delimitar automáticamente la región térmica correspondiente a la lesión, aprovechando las diferencias de intensidad presentes en la imagen infrarroja. Para ello se utilizó una estrategia basada en umbralización automática mediante el método de Otsu y posterior aplicación del algoritmo Watershed.
La Figura 4.10 muestra el resultado obtenido durante las pruebas realizadas. Puede observarse cómo el algoritmo consigue delimitar correctamente el contorno general de la región caliente mediante una línea roja, separando el primer plano del fondo de la imagen.
Sin embargo, durante las pruebas realizadas se comprobó que el comportamiento del algoritmo dependía de forma significativa de la calidad de la segmentación inicial y de la distribución térmica presente en cada imagen. En determinadas situaciones aparecían sobresegmentaciones o delimitaciones poco consistentes, lo que dificultaba su utilización como herramienta de análisis automático dentro de la plataforma.
Por este motivo, aunque Watershed constituye una técnica de segmentación muy potente, finalmente se optó por emplear algoritmos basados en umbrales adaptativos y análisis de regiones

calientes, los cuales proporcionaban resultados más estables para el conjunto de imágenes empleado durante este trabajo.

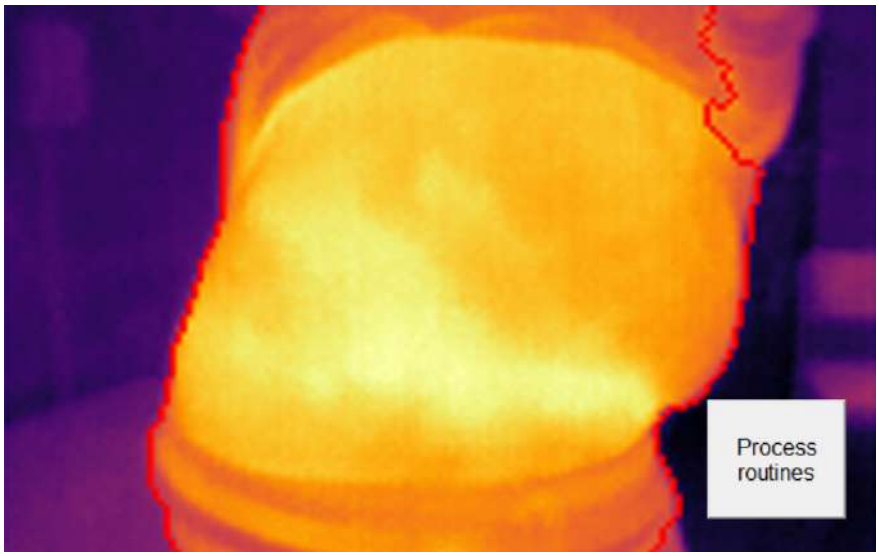


Figura 4.10: Resultado obtenido mediante el algoritmo Watershed aplicado sobre una imagen térmica.

Funcionamiento del algoritmo

El algoritmo comienza normalizando la imagen térmica al rango de 8 bits y generando una representación en falso color que facilita tanto la visualización como el procesamiento posterior.

```
img8 = cv.normalize(...)
image = cv.applyColorMap(...)
```

Posteriormente se convierte la imagen a escala de grises y se aplica una umbralización automática mediante el método de Otsu. Esta etapa permite obtener una primera separación entre la región caliente y el fondo de la escena.

```
_, thresh = cv.threshold(
    gray,
    0,
    255,
    cv.THRESH_BINARY +
    cv.THRESH_OTSU
)
```

A partir de la máscara obtenida se calculan las regiones correspondientes al fondo seguro y al primer plano seguro mediante operaciones morfológicas y transformada de distancia. Estas regiones sirven como marcadores de entrada para el algoritmo Watershed.

```
sure_bg = cv.dilate(...)
dist_transform =
    cv.distanceTransform(...)
```

Finalmente se aplica el algoritmo Watershed, que utiliza los marcadores previamente obtenidos para separar las diferentes regiones de la imagen. Las fronteras calculadas se representan en color rojo sobre la imagen térmica original, permitiendo visualizar el resultado de la segmentación.

```
markers = cv.watershed(
    image,
    markers
)

image[markers==-1]=[0,0,255]
```

Aunque el algoritmo consiguió delimitar correctamente algunas regiones de interés, las pruebas realizadas mostraron una elevada dependencia de la segmentación inicial y de la distribución térmica de cada imagen. Esta variabilidad motivó que el método no fuese incorporado a la versión final de la plataforma, aunque sí constituyó una etapa importante durante el proceso de investigación y permitió comparar diferentes estrategias de segmentación automática.

Capítulo 5

Conclusiones

A lo largo del trabajo se ha estudiado una plataforma de adquisición y procesamiento de imágenes térmicas, se han analizado diferentes técnicas de visión artificial y se han desarrollado nuevos plugins destinados a mejorar las capacidades de análisis del sistema.
En este capítulo se presentan las principales conclusiones extraídas tras la realización del proyecto, destacando las aportaciones realizadas, las limitaciones encontradas y las posibles líneas de trabajo que podrían desarrollarse en el futuro para continuar mejorando la plataforma.

5.1. Conclusiones generales

El presente proyecto ha permitido profundizar en el desarrollo de herramientas de procesamiento digital de imagen aplicadas al ámbito biomédico, integrando conocimientos de visión artificial, programación y análisis térmico dentro de una plataforma embebida basada en Raspberry Pi. El proyecto se ha desarrollado sobre una plataforma previamente existente, cuyo objetivo es facilitar el estudio de lesiones vasculares mediante la adquisición simultánea de imágenes en el espectro visible y en el infrarrojo lejano.
A lo largo del trabajo se ha realizado un estudio previo sobre las principales técnicas empleadas en el procesamiento de imágenes térmicas, analizando tanto el contexto clínico asociado a los hemangiomas infantiles como las tecnologías utilizadas actualmente en este tipo de aplicaciones. Esta fase permitió comprender las ventajas que ofrece la termografía infrarroja como herramienta complementaria para el seguimiento de este tipo de lesiones y sirvió como punto de partida para el diseño de nuevos algoritmos de procesamiento.
La principal aportación del trabajo ha consistido en el desarrollo e integración de diferentes plugins dentro de la arquitectura software de la plataforma. Estos módulos permiten incorporar nuevas funcionalidades sin modificar el núcleo principal del sistema, facilitando tanto la ampliación futura de la aplicación como la evaluación independiente de distintos algoritmos de procesamiento.
Durante el desarrollo se implementaron diferentes estrategias de segmentación y análisis térmico. Algunas de ellas pasaron a formar parte de la versión final de la plataforma al proporcionar resultados estables y fácilmente interpretables, mientras que otras fueron evaluadas únicamente como alternativas de investigación. Este proceso permitió comparar diferentes enfoques y seleccionar aquellas soluciones que mejor se adaptaban a las características de las imágenes térmicas disponibles y a las necesidades del entorno clínico.
Entre los desarrollos finalmente integrados destacan el plugin de eliminación del fondo térmico, el sistema de clasificación de regiones mediante K-means y el algoritmo de análisis de diferencia térmica. Estos desarrollos permiten mejorar la visualización de la información térmica, localizar

automáticamente las regiones de interés y proporcionar parámetros cuantitativos que pueden resultar útiles durante el seguimiento de lesiones vasculares.
En conjunto, los resultados obtenidos demuestran que la arquitectura basada en plugins constituye una solución adecuada para incorporar nuevas funcionalidades de procesamiento de imagen dentro de la plataforma existente. Asimismo, el trabajo desarrollado pone de manifiesto el potencial que presentan las técnicas de visión artificial y análisis térmico para asistir al personal sanitario en tareas de evaluación y seguimiento clínico, sirviendo además como base para futuras investigaciones orientadas a la mejora de este tipo de sistemas.

5.2. Aportaciones del trabajo

El trabajo realizado ha permitido ampliar las capacidades de una plataforma de adquisición y procesamiento de imágenes térmicas destinada al análisis de lesiones vasculares. Partiendo de una infraestructura previamente desarrollada, se han diseñado e implementado nuevas funcionalidades software que mejoran el tratamiento de las imágenes obtenidas por el sensor infrarrojo y facilitan la interpretación de la información térmica por parte del usuario.
Una de las principales aportaciones de este proyecto ha sido el desarrollo de una serie de plugins capaces de integrarse dentro de la arquitectura modular de la plataforma. Gracias a esta filosofía de diseño, las nuevas herramientas implementadas pueden incorporarse sin modificar el funcionamiento del núcleo principal de la aplicación, favoreciendo tanto la reutilización del código como la posibilidad de continuar ampliando el sistema mediante futuros desarrollos.
Además de la implementación de nuevos algoritmos, el trabajo ha permitido evaluar diferentes técnicas de procesamiento digital de imagen aplicadas al ámbito de la termografía infrarroja. Durante el desarrollo se estudiaron distintos enfoques de segmentación y extracción de información, comparando sus resultados hasta seleccionar aquellos que ofrecían un comportamiento más robusto y una representación más adecuada para el análisis de lesiones térmicas.
Entre las funcionalidades incorporadas destacan la segmentación automática de regiones calientes, la clasificación de zonas térmicas mediante técnicas de agrupamiento, así como la obtención de parámetros cuantitativos relacionados con la distribución de temperaturas. Estas herramientas permiten reducir la intervención manual del usuario y facilitan la obtención de información objetiva a partir de las imágenes adquiridas.
Otra aportación importante ha consistido en demostrar la viabilidad de realizar este tipo de procesamiento utilizando una plataforma embebida basada en Raspberry Pi. A pesar de tratarse de un sistema de bajo coste y reducido consumo energético, la plataforma ha mostrado capacidad suficiente para ejecutar los algoritmos desarrollados y visualizar sus resultados de forma integrada.
Finalmente, este trabajo establece una base sólida para la incorporación de nuevas herramientas de análisis térmico dentro de la plataforma. La arquitectura modular empleada permitirá que futuros desarrollos puedan integrarse de manera sencilla, ampliando progresivamente las capacidades del sistema y favoreciendo su aplicación en entornos clínicos y de investigación.



Figura 5.1: Esquema de desarrollo de TFM.

5.2.1. Resumen de las principales aportaciones

De forma resumida, las principales aportaciones realizadas durante este trabajo son las siguientes:

- Estudio del funcionamiento de una plataforma portátil para adquisición de imágenes térmicas.
- Desarrollo e integración de nuevos plugins dentro de la arquitectura software existente.
- Implementación de diferentes algoritmos de procesamiento y segmentación de imágenes térmicas.
- Evaluación comparativa de distintas técnicas de análisis, seleccionando aquellas que ofrecían mejores resultados.
- Obtención de parámetros cuantitativos útiles para el análisis térmico de lesiones vasculares.
- Generación de una base software que facilita la incorporación de futuras funcionalidades en la plataforma.

5.3. Limitaciones del trabajo

A pesar de los resultados obtenidos, el desarrollo realizado presenta una serie de limitaciones que deben tenerse en cuenta al interpretar el alcance de este proyecto.
En primer lugar, los algoritmos implementados han sido desarrollados y evaluados sobre un conjunto limitado de imágenes térmicas adquiridas mediante la plataforma disponible. Aunque estas imágenes han permitido validar el funcionamiento de los distintos plugins desarrollados, sería conveniente disponer de un mayor número de casos clínicos que permitieran realizar una evaluación más completa y estudiar el comportamiento de los algoritmos frente a una mayor variabilidad de lesiones y condiciones de adquisición.

Otra limitación importante está relacionada con la calibración espacial de las imágenes. Actualmente las medidas geométricas obtenidas, como el área de una lesión, se expresan en número de píxeles, ya que la plataforma no dispone de un procedimiento de calibración que permita convertir dichas medidas a unidades físicas como milímetros cuadrados. La incorporación de un sistema de calibración permitiría obtener parámetros cuantitativos más útiles para el seguimiento clínico. Asimismo, aunque los plugins desarrollados permiten identificar regiones calientes y extraer información térmica relevante, todavía no se dispone de un sistema automático de correspondencia entre la imagen infrarroja y la imagen visible. La incorporación de técnicas de registro entre ambas modalidades facilitaría la localización precisa de las lesiones y mejoraría la interpretación de los resultados obtenidos.
Por último, los algoritmos implementados han sido diseñados con el objetivo de demostrar la viabilidad de diferentes estrategias de procesamiento de imagen dentro de una plataforma embebida. No obstante, su utilización como herramienta de apoyo al diagnóstico requeriría una validación clínica más amplia y la colaboración continuada con especialistas médicos antes de su aplicación en un entorno asistencial.

5.4. Líneas futuras

El trabajo desarrollado constituye una base sobre la que resulta posible continuar ampliando las capacidades de la plataforma de análisis térmico. La arquitectura modular basada en plugins facilita la incorporación de nuevas funcionalidades sin necesidad de modificar el núcleo principal de la aplicación, permitiendo que el sistema evolucione de forma progresiva conforme se desarrollen nuevas técnicas de procesamiento de imagen.
Una de las líneas de trabajo más interesantes consiste en mejorar los algoritmos de segmentación automática de las lesiones. Durante el desarrollo del proyecto se exploraron diferentes estrategias, como el algoritmo Watershed, que demostró el potencial de este tipo de técnicas para delimitar regiones de interés. En futuras versiones sería interesante combinar este tipo de algoritmos con métodos adaptativos más robustos o con técnicas de aprendizaje automático que permitan mejorar la precisión de la segmentación.
Otra posible línea de desarrollo consiste en implementar un sistema de registro entre las imágenes adquiridas por el sensor infrarrojo y la cámara visible. La correspondencia automática entre ambas modalidades permitiría proyectar sobre la imagen visible la información obtenida del análisis térmico, facilitando la localización de la lesión y proporcionando una herramienta más intuitiva para el personal sanitario durante las sesiones clínicas.
Finalmente, una línea especialmente prometedora consiste en ampliar la validación clínica de los algoritmos desarrollados mediante el análisis de un mayor número de pacientes y diferentes tipos de lesiones. La colaboración con especialistas del Área de Dermatología Pediátrica del Hospital Universitario Virgen del Rocío permitiría evaluar el comportamiento de los plugins en situaciones reales de uso e identificar nuevas necesidades que pudieran dar lugar al desarrollo de futuras herramientas de apoyo al diagnóstico y seguimiento clínico.
En conjunto, todas estas líneas de trabajo muestran que la plataforma desarrollada posee un importante potencial de crecimiento. Los resultados obtenidos durante este TFM constituyen un primer paso hacia el desarrollo de sistemas cada vez más completos de análisis térmico, capaces de proporcionar información objetiva y de utilidad clínica para apoyar el estudio y seguimiento de lesiones vasculares mediante técnicas de visión artificial y procesamiento digital de imagen.

Bibliografía

[1] J. A. Lenero-Bardallo, R. de la Rosa-Vidal, R. Padial-Allué, J. Ceballos-Cáceres, Rodríguez-Vázquez, and J. Bernabéu-Wittel, "A customizable thermographic imaging system for medical image acquisition and processing," *IEEE Sensors Journal*, 2021.

[2] C. L'eaut'e-Labr'eze, J. I. Harper, and P. H. Hoeger, "Infantile haemangioma," *The Lancet*, vol. 390, no. 10089, pp. 85–94, 2017.

[3] D. P. Krowchuk, I. J. Frieden, A. J. Mancini, D. H. Darrow, *et al.*, "Clinical practice guideline for the management of infantile hemangiomas," *Pediatrics*, vol. 143, no. 1, 2019.

[4] J. A. Leñero-Bardallo, B. Acha, C. Serrano, J. A. Pérez-Carrasco, J. Ortiz-Álvarez, and J. Bernabéu-Wittel, "Thermography as a method for bedside monitoring of infantile hemangiomas," *Cancers*, vol. 14, no. 21, p. 5392, 2022.

[5] J. A. Lenero-Bardallo, C. Serrano, B. Acha, J. A. Pérez-Carrasco, and J. Bernabéu-Wittel, "Thermography for the differential diagnosis of vascular malformations," *Clinical and Experimental Dermatology*, 2020.

[6] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* Pearson, 4 ed., 2018.

Apéndice A

Anexo A: Implementación de los algoritmos desarrollados

En este anexo se recogen los fragmentos de código más representativos de los algoritmos desarrollados durante el desarrollo del trabajo. No se incluye el código completo de cada plugin, sino aquellas partes que permiten comprender la estructura interna del procesamiento realizado.

A.1. Plugin de eliminación del fondo térmico

El primer plugin implementado tiene como objetivo aislar la región térmica de interés y eliminar visualmente el fondo de la imagen infrarroja. Para ello, el algoritmo segmenta la zona caliente principal, detecta focos internos de mayor temperatura y representa sobre la imagen resultante diferentes parámetros de interés, como el área de la región detectada y una estimación de la temperatura media.

A.1.1. Inicialización

En primer lugar, el plugin define una serie de parámetros internos que controlan el comportamiento del algoritmo. Entre ellos destacan los percentiles utilizados para detectar la región caliente y los focos de mayor temperatura, así como los tamaños mínimos de área empleados para descartar pequeñas detecciones debidas al ruido.

Listing A.1: Inicialización del plugin de eliminación del fondo térmico.

```
class ThermalBackgroundRemoval(PluginBase):
    def __init__(self) -> None:
        super().__init__("Thermal Background", True)

        # Regi n caliente principal amplia
        self.hot_percentile = 82

        # Focos m s calientes dentro de la regi n
        self.hotspot_percentile = 98.5

        #  reas  m nimas
        self.min_area_px = 250
        self.min_hotspot_area_px = 8

        # Suavizado y morfolog a
        self.blur_ksize = 5
        self.morph_ksize = 5
```

El uso de percentiles permite que el algoritmo se adapte automáticamente a cada imagen, evitando depender de un umbral fijo. De esta forma, el plugin puede responder mejor ante cambios de iluminación térmica, distancia al paciente o variaciones de temperatura en la escena.

A.1.2. Procesamiento principal

El método principal del plugin es `ProcessImage()`, encargado de recibir la imagen infrarroja, preparar los datos y aplicar las distintas etapas de segmentación y visualización. En primer lugar, la imagen se convierte a un formato adecuado y posteriormente se genera una representación térmica en falso color.

Listing A.2: Preprocesado inicial y generación de la imagen térmica.

```
def ProcessImage(self, img_in):
    if img_in is None:
        return np.zeros((240, 320, 3), dtype=np.uint8)

    img_u16 = self._ensure_u16(img_in)
    img8 = self._robust_u8(img_u16)

    # Imagen t rmica en falso color
    thermal = cv.applyColorMap(img8, cv.COLORMAP_INFERNO)

    # Regi n caliente principal
    mask = self._segment_hot_percentile(img8)

    # Focos de mayor temperatura
    hotspots = self._segment_extra_hotspots(img8)
```

La imagen original procedente del sensor se transforma primero a una representación de 8 bits para facilitar su procesamiento. Posteriormente, la paleta `COLORMAP_INFERNO` permite generar una imagen visualmente más intuitiva, donde las zonas de mayor intensidad térmica aparecen resaltadas.

A.1.3. Eliminación del fondo

Una vez obtenida la máscara de la región caliente, se crea una imagen de salida inicialmente negra. Sobre ella se copian únicamente los píxeles pertenecientes a la región detectada y a los focos de mayor temperatura.

Listing A.3: Eliminación del fondo de la escena.

```
# Fondo negro
out = np.zeros_like(thermal)

# Mostrar regi n caliente principal
if mask is not None:
    out[mask == 255] = thermal[mask == 255]

# Mostrar tambi n focos m s calientes
if hotspots is not None:
    out[hotspots == 255] = thermal[hotspots == 255]
```

Este bloque es el que produce el efecto visual principal del plugin: el fondo de la escena desaparece y solo se conserva la zona térmicamente relevante. Esto facilita la interpretación de la imagen y permite centrar la atención en la posible región de interés clínica.

A.1.4. Segmentación de la región caliente

La segmentación de la región principal se realiza mediante un umbral calculado a partir de un percentil de intensidad. Antes de aplicar dicho umbral, se utiliza un filtro Gaussiano para reducir el ruido presente en la imagen.

Listing A.4: Segmentación de la región caliente mediante percentiles.

```
def _segment_hot_percentile(self, img8):
    k = self._odd(self.blur_ksize)
    blur = cv.GaussianBlur(img8, (k, k), 0)

    # M scara amplia: zona caliente general
    warm_thr = np.percentile(blur, self.hot_percentile)
    _, warm = cv.threshold(
        blur,
        warm_thr,
        255,
        cv.THRESH_BINARY
    )

    # M scara restrictiva: focos claramente m s calientes
    seed_thr = np.percentile(blur, self.hotspot_percentile)
    _, seeds = cv.threshold(
        blur,
        seed_thr,
        255,
        cv.THRESH_BINARY
    )
```

La máscara `warm` representa la región caliente general, mientras que la máscara `seeds` identifica zonas más restrictivas y de mayor intensidad. Esta doble segmentación permite seleccionar regiones amplias que además contienen puntos térmicamente significativos.

A.1.5. Limpieza morfológica y selección de región

Tras la umbralización, se aplican operaciones morfológicas para eliminar ruido y mejorar la forma de la región detectada. Posteriormente, se analizan las componentes conexas y se conserva la región más relevante.

Listing A.5: Limpieza morfológica y selección de componentes.

```
mk = self._odd(self.morph_ksize)
kernel = cv.getStructuringElement(
    cv.MORPH_ELLIPSE,
    (mk, mk)
)

# Limpiar zona amplia
warm = cv.morphologyEx(
    warm,
    cv.MORPH_OPEN,
    kernel,
```

```
    iterations=1
)

warm = cv.morphologyEx(
    warm,
    cv.MORPH_CLOSE,
    kernel,
    iterations=3
)

# Limpiar semillas calientes
seeds = cv.morphologyEx(
    seeds,
    cv.MORPH_OPEN,
    kernel,
    iterations=1
)

num, labels, stats, _ = cv.connectedComponentsWithStats(
    warm,
    connectivity=8
)
```

La apertura morfológica elimina pequeñas regiones aisladas, mientras que el cierre ayuda a rellenar huecos en el interior de la máscara. El análisis de componentes permite descartar regiones demasiado pequeñas y seleccionar únicamente aquellas que cumplen los criterios definidos.

A.1.6. Detección de contornos y localización

Una vez obtenida la máscara final, el plugin calcula el contorno principal de la región detectada. Además, se dibuja un rectángulo delimitador y se marca el punto de máxima intensidad térmica.

Listing A.6: Detección del contorno principal y localización de la región.

```
contours, _ = cv.findContours(
    mask,
    cv.RETR_EXTERNAL,
    cv.CHAIN_APPROX_SIMPLE
)

if contours:
    cnt = max(contours, key=cv.contourArea)

    # Contorno amarillo de la regi n principal
    cv.drawContours(out, [cnt], -1, (0, 255, 255), 1)

    # Rect ngulo blanco de localizaci n
    x, y, w, h = cv.boundingRect(cnt)
    cv.rectangle(
        out,
        (x, y),
        (x + w, y + h),
        (255, 255, 255),
        1
    )

    # Cruz en el m ximo t rmico
    masked_img = cv.bitwise_and(img8, img8, mask=mask)
    _, max_intensity, _, max_loc = cv.minMaxLoc(masked_img)
```

```
26      cv.drawMarker(
27          out,
28          max_loc,
29          (255, 255, 255),
30          cv.MARKER_CROSS,
31          7,
32          1
33      )
```

El contorno amarillo permite visualizar el límite de la región detectada, mientras que el rectángulo blanco facilita su localización dentro de la imagen. La cruz señala el punto de máxima intensidad térmica dentro de la zona segmentada.

A.1.7. Obtención de parámetros

Además de la representación visual, el plugin calcula parámetros cuantitativos básicos de la región detectada. Entre ellos se incluyen el área en píxeles y una estimación de la temperatura media.

Listing A.7: Cálculo de área y temperatura media estimada.

```
1  #  rea  en p xeles
2  area_px = int(np.count_nonzero(mask))
3
4  # Intensidad media dentro de la regi n
5  lesion_i = img8[mask == 255].astype(np.float32)
6  imean = float(np.mean(lesion_i)) if lesion_i.size else 0.0
7
8  # Temperatura media estimada de la regi n
9  t_mean = self.IR_to_temp(imean)
10
11 # Temperatura m xima estimada dentro de la regi n
12 t_max = self.IR_to_temp(float(max_intensity))
```

El área se obtiene contando los píxeles pertenecientes a la máscara segmentada. La temperatura media se estima a partir de la intensidad media de dichos píxeles, utilizando posteriormente una función de conversión de intensidad infrarroja a temperatura.

A.1.8. Representación de resultados

Los valores calculados se muestran directamente sobre la imagen de salida. Para evitar interferir con la región térmica detectada, el texto se coloca en una zona fija del fondo negro.

Listing A.8: Representación de parámetros sobre la imagen final.

```
1  text1 = f"A={area_px}px"
2  text2 = f"T={t_mean:.1f}C"
3
4  cv.putText(
5      out,
6      text1,
7      (8, h - 28),
8      cv.FONT_HERSHEY_SIMPLEX,
9      0.34,
10     (255, 255, 255),
11     1
12 )
```

```
13
14 cv.putText(
15     out,
16     text2,
17     (8, h - 12),
18     cv.FONT_HERSHEY_SIMPLEX,
19     0.34,
20     (255, 255, 255),
21     1
22 )
```

De esta forma, el usuario puede observar simultáneamente la región segmentada y los parámetros principales asociados a ella. En esta versión del plugin se muestran el área y la temperatura media, aunque el código también calcula la temperatura máxima para posibles ampliaciones posteriores.

A.1.9. Detección de focos térmicos

El plugin incluye una función específica para detectar pequeños focos de mayor temperatura. Para ello se emplea un percentil más restrictivo que el utilizado para la región caliente principal.

Listing A.9: Segmentación de focos térmicos de mayor intensidad.

```
1  def _segment_extra_hotspots(self, img8):
2      k = self._odd(self.blur_ksize)
3      blur = cv.GaussianBlur(img8, (k, k), 0)
4
5      thr = np.percentile(blur, self.hotspot_percentile)
6      _, th = cv.threshold(
7          blur,
8          thr,
9          255,
10         cv.THRESH_BINARY
11     )
12
13     mk = self._odd(self.morph_ksize)
14     kernel = cv.getStructuringElement(
15         cv.MORPH_ELLIPSE,
16         (mk, mk)
17     )
18
19     clean = cv.morphologyEx(
20         th,
21         cv.MORPH_OPEN,
22         kernel,
23         iterations=1
24     )
25
26     clean = cv.morphologyEx(
27         clean,
28         cv.MORPH_CLOSE,
29         kernel,
30         iterations=1
31     )
```

Esta función permite detectar regiones más pequeñas pero térmicamente más intensas. Dichos focos se filtran posteriormente por área mínima para evitar falsas detecciones debidas al ruido del sensor.

A.1.10. Conversión intensidad-temperatura

La función `IR_to_temp()` transforma los niveles digitales de intensidad infrarroja en una estimación de temperatura. Para ello se emplea una calibración lineal basada en dos puntos de referencia.

Listing A.10: Conversión de intensidad infrarroja a temperatura estimada.

```
def IR_to_temp(self, IR_value: float) -> float:
    T1 = 36.0
    T2 = 36.04
    R1 = 8250
    R2 = 8400

    a = (T2 - T1) / (R2 - R1)
    b = T2 - a * R2

    temp = IR_value * (16383.0 / 255.0) * a + b

    return round(temp, 1)
```

Aunque esta conversión no sustituye a una calibración radiométrica completa, permite ofrecer al usuario una estimación sencilla de la temperatura en grados Celsius, facilitando la interpretación de los resultados obtenidos.

A.1.11. Resumen del algoritmo

En conjunto, el plugin de eliminación del fondo térmico implementa una cadena de procesamiento compuesta por preprocesado, segmentación adaptativa, limpieza morfológica, detección de contornos, cálculo de parámetros y visualización final. La utilización de umbrales basados en percentiles permite adaptar el comportamiento del algoritmo a diferentes imágenes térmicas, mientras que las operaciones morfológicas y el filtrado por área contribuyen a reducir falsas detecciones.
El resultado final es una imagen simplificada en la que se elimina el fondo de la escena y se conserva únicamente la región térmica de interés junto con los focos de mayor intensidad, proporcionando una visualización clara y útil para el análisis posterior.

A.2. Plugin de clasificación térmica mediante K-means

El segundo plugin desarrollado incorpora un algoritmo de clasificación no supervisada basado en K-means. A diferencia del plugin anterior, cuyo objetivo principal era eliminar el fondo de la imagen, este algoritmo pretende dividir automáticamente la escena térmica en diferentes regiones según su nivel de temperatura.
La utilización de K-means permite agrupar píxeles con un comportamiento térmico similar, obteniendo una representación simplificada de la distribución de temperaturas presente en la imagen. Posteriormente, sobre dicha clasificación se resaltan los focos de mayor intensidad térmica, facilitando su identificación visual.

A.2.1. Inicialización

Durante la inicialización del plugin se definen los parámetros principales que controlan el proceso de clasificación. En primer lugar se establece el número de clases térmicas que serán obtenidas mediante el algoritmo K-means. Asimismo, se fija el percentil empleado para localizar los focos

de mayor temperatura y el área mínima utilizada para eliminar pequeñas detecciones debidas al ruido.

Listing A.11: Inicialización del plugin K-Means Thermal.

```
class KMeansThermal(PluginBase):

    def __init__(self):

        super().__init__("K-Means Thermal", True)

        # Clases t rmicas
        self.K = 4

        # Percentil para hotspots
        self.hotspot_percentile = 98.5

        #  rea  m nima
        self.min_hotspot_area_px = 8
```

En este desarrollo se seleccionan cuatro clases térmicas diferentes, suficientes para diferenciar zonas de ambiente, regiones templadas y áreas con elevada temperatura, manteniendo una representación sencilla e intuitiva.

A.2.2. Clasificación térmica mediante K-means

El procesamiento comienza normalizando la imagen procedente del sensor infrarrojo al rango de 8 bits. Posteriormente, todos los píxeles de la imagen se reorganizan en un único vector que servirá como entrada para el algoritmo de agrupamiento K-means.

Listing A.12: Preparación de la imagen para el algoritmo K-means.

```
img8 = cv.normalize(
    img_in,
    None,
    0,
    255,
    cv.NORM_MINMAX
).astype(np.uint8)

pixels = img8.reshape((-1,1)).astype(np.float32)
```

Una vez preparados los datos, se ejecuta el algoritmo K-means utilizando OpenCV. El procedimiento agrupa automáticamente los píxeles en cuatro clases según su intensidad térmica, obteniendo tanto la asignación de cada píxel como el valor medio correspondiente a cada grupo.

Listing A.13: Aplicación del algoritmo K-means.

```
_, labels, centers = cv.kmeans(
    pixels,
    self.K,
    None,
    criteria,
    attempts=10,
    flags=cv.KMEANS_RANDOM_CENTERS
)
```

El algoritmo empleado pertenece a las técnicas de aprendizaje no supervisado, ya que la clasificación se realiza sin necesidad de conocer previamente la temperatura correspondiente a cada región de la imagen.

A.2.3. Representación de resultados

Una vez obtenidos los diferentes grupos térmicos, estos se ordenan automáticamente desde las regiones más frías hasta las más calientes. Posteriormente se asigna un color específico a cada clase, generando una representación visual mucho más sencilla que la imagen térmica original.

Listing A.14: Asignación de colores a las clases térmicas.

```
color_map_bgr = [

    (40,140,40),

    (40,140,40),

    (200,180,0),

    (0,80,220)

]

for cluster_id in range(self.K):

    r = int(rank[cluster_id])

    img_kmeans[labels_2d==cluster_id] = color_map_bgr[r]
```

Finalmente, la clasificación obtenida se combina con la imagen térmica original mediante una mezcla ponderada. De esta forma se conserva parte de la información original del sensor mientras se resaltan visualmente las distintas regiones térmicas detectadas.

Listing A.15: Combinación entre imagen térmica y clasificación.

```
img_color = cv.applyColorMap(
    img8,
    cv.COLORMAP_INFERNO
)

output = cv.addWeighted(
    img_color,
    0.30,
    img_kmeans,
    0.70,
    0
)
```

A.2.4. Detección de focos térmicos

Además de la clasificación general, el plugin incorpora una segunda etapa destinada a localizar automáticamente los focos de mayor intensidad térmica presentes en la imagen.
Para ello se calcula un umbral utilizando un percentil elevado y posteriormente se aplican operaciones morfológicas destinadas a eliminar pequeñas regiones producidas por ruido.

Listing A.16: Detección automática de focos térmicos.

```
thr = np.percentile(
    blur,
    self.hotspot_percentile
)

_, mask = cv.threshold(
    blur,
    thr,
    255,
    cv.THRESH_BINARY
)

mask = cv.morphologyEx(...)
```

Las regiones válidas son finalmente representadas mediante un relleno rojo semitransparente y un contorno de mayor intensidad, permitiendo distinguir rápidamente aquellas zonas que presentan una temperatura superior al resto de la escena.
Asimismo, el plugin incorpora una pequeña leyenda que facilita la interpretación de las distintas categorías térmicas representadas mediante colores.

A.2.5. Resumen del algoritmo

El plugin desarrollado combina técnicas de agrupamiento mediante K-means con procedimientos clásicos de procesamiento digital de imagen para obtener una representación simplificada de la distribución térmica presente en la escena.
La utilización de cuatro clases térmicas permite diferenciar claramente las distintas regiones de temperatura, mientras que la detección independiente de focos calientes facilita la identificación de posibles zonas de interés clínico.
Como resultado final se obtiene una imagen en falso color que resume de forma sencilla la información térmica capturada por el sensor infrarrojo, facilitando la interpretación visual por parte del usuario y proporcionando una representación más intuitiva de la distribución espacial de temperaturas.

A.3. Plugin de análisis de diferencia térmica

El tercer plugin desarrollado tiene como objetivo realizar un análisis cuantitativo de la distribución térmica presente en la región caliente detectada. A diferencia de los algoritmos anteriores, este plugin no solo segmenta la lesión, sino que calcula diferentes parámetros térmicos que permiten evaluar la variación de temperatura existente en su interior.
Concretamente, el algoritmo estima la temperatura media de la región segmentada, localiza automáticamente el punto de máxima intensidad térmica y calcula la diferencia existente entre ambos valores. Este parámetro, denominado diferencia térmica o *Delta*, proporciona una medida sencilla de la heterogeneidad térmica de la lesión y facilita la identificación de posibles focos internos de mayor actividad.

A.3.1. Inicialización

Durante la inicialización del plugin se definen los parámetros necesarios para controlar las distintas etapas del procesamiento. Entre ellos destacan los percentiles utilizados para segmentar la región caliente principal, los umbrales empleados para localizar focos internos y el parámetro

`hotspot_margin`, encargado de controlar la sensibilidad en la detección de las zonas de máxima temperatura.

Listing A.17: Inicialización del plugin Thermal Delta Analysis.

```
class ThermalDeltaAnalysis(PluginBase):

    def __init__(self):

        super().__init__("Thermal Delta Analysis", True)

        self.hot_percentile = 82

        self.hotspot_percentile = 96.0

        self.min_area_px = 250

        self.min_hotspot_area_px = 8

        self.hotspot_margin = 3.0
```

La utilización de estos parámetros permite adaptar fácilmente el comportamiento del algoritmo a diferentes condiciones de captura, manteniendo una estructura flexible para futuras modificaciones.

A.3.2. Segmentación de la región caliente

El procesamiento comienza transformando la imagen procedente del sensor infrarrojo a un formato adecuado para su análisis y generando una representación térmica en falso color mediante OpenCV.
Posteriormente se aplica un procedimiento de segmentación basado en percentiles para aislar la región caliente principal. Este proceso es similar al utilizado en el primer plugin, aunque en este caso la región obtenida será utilizada posteriormente para realizar el análisis térmico interno.

Listing A.18: Segmentación de la región caliente.

```
img8 = self.robust_u8(img_in)

output = cv.applyColorMap(
    img8,
    cv.COLORMAP_INFERNO
)

mask = self.segment_hot_zone(img8)
```

Una vez obtenida la máscara binaria, el algoritmo localiza el contorno principal de la región detectada y dibuja sobre la imagen tanto el límite de la lesión como un rectángulo que facilita su localización dentro de la escena.

A.3.3. Cálculo de parámetros térmicos

La principal aportación de este plugin consiste en la obtención automática de parámetros térmicos representativos de la lesión. Para ello se calcula inicialmente la intensidad media correspondiente a todos los píxeles pertenecientes a la región segmentada.
Posteriormente se localiza el punto de máxima intensidad mediante la función `minMaxLoc()`, obteniendo así la temperatura máxima presente dentro de la lesión.

Listing A.19: Obtención de temperaturas características.

```
zone_values = img8[mask == 255].astype(np.float32)

zone_mean_i = float(np.mean(zone_values))

temp_zone = self.IR_to_temp(zone_mean_i)

masked_img = cv.bitwise_and(
    img8,
    img8,
    mask=mask
)

_, max_i, _, max_loc = cv.minMaxLoc(masked_img)

temp_max = self.IR_to_temp(float(max_i))
```

Una vez conocidas ambas temperaturas, el algoritmo calcula la diferencia térmica entre ellas, parámetro que constituye el principal indicador empleado durante este desarrollo para evaluar la presencia de focos internos de elevada temperatura.

A.3.4. Representación de resultados

Tras obtener los parámetros térmicos, el algoritmo identifica automáticamente aquellas regiones cuya temperatura se encuentra próxima al máximo detectado dentro de la lesión.
Para ello se utiliza un margen configurable respecto a la temperatura máxima, generando una máscara que únicamente conserva las zonas de mayor intensidad térmica.

Listing A.20: Detección de focos internos de mayor temperatura.

```
hotspot_mask = self.segment_hotspots_inside_zone(
    img8,
    mask
)

cv.drawContours(
    output,
    [hc],
    -1,
    (0,0,255),
    1
)
```

Finalmente, el plugin representa sobre la imagen la temperatura media de la región, la temperatura máxima detectada y la diferencia térmica existente entre ambas. Además, el punto correspondiente al máximo térmico se marca mediante una cruz, facilitando su localización visual.

Listing A.21: Representación de parámetros sobre la imagen final.

```
cv.putText(
    output,
    f"Tzona={temp_zone:.1f}C",
    ...
)

cv.putText(
```

```
    output,
    f"Tmax={temp_max:.1f}C",
    ...
)

cv.putText(
    output,
    f"Delta={delta_t:+.1f}C",
    ...
)
```

A.3.5. Resumen del algoritmo

El plugin de análisis de diferencia térmica amplía las funcionalidades desarrolladas en los algoritmos anteriores incorporando un análisis cuantitativo de la distribución de temperaturas dentro de la lesión.

Además de segmentar automáticamente la región caliente, el algoritmo calcula la temperatura media, determina el punto de máxima intensidad térmica y obtiene la diferencia existente entre ambos valores. Este parámetro permite identificar de forma sencilla la presencia de focos térmicos internos que pueden resultar de interés durante la evaluación clínica.

Como resultado final se obtiene una representación visual clara de la lesión junto con información cuantitativa que facilita la interpretación de la distribución térmica y proporciona una herramienta de apoyo para el análisis de imágenes infrarrojas.

Apéndice B

Manual de usuario

El presente anexo tiene como finalidad describir el funcionamiento básico de la plataforma empleada durante el desarrollo de este proyecto, así como el procedimiento necesario para utilizar los plugins desarrollados. Aunque la aplicación presenta una interfaz sencilla e intuitiva, se incluyen las principales pantallas y una breve explicación de cada una de ellas con el objetivo de facilitar su utilización.
En primer lugar, se muestra la interfaz principal de la aplicación y los elementos que la componen. Posteriormente, se describe el proceso de selección y ejecución de los distintos plugins implementados, así como la interpretación de la información mostrada durante el procesamiento de las imágenes térmicas.
Este manual está orientado a proporcionar una guía básica de uso para cualquier usuario que desee utilizar la plataforma, permitiendo comprender el significado de los resultados obtenidos y el funcionamiento general de los algoritmos desarrollados.

B.1. Inicio de la plataforma

La plataforma utilizada durante este proyecto se ejecuta directamente sobre la Raspberry Pi integrada en el sistema portátil de adquisición. El acceso a la aplicación se realiza desde el entorno gráfico del sistema operativo, donde se encuentra disponible el ejecutable *VisionBio*.
Para iniciar la aplicación únicamente es necesario ejecutar dicho archivo. Una vez lanzado el programa, éste inicializa automáticamente los dispositivos de adquisición y carga la interfaz principal, desde la que se gestionan tanto la visualización de las imágenes como la ejecución de los distintos algoritmos de procesamiento desarrollados.

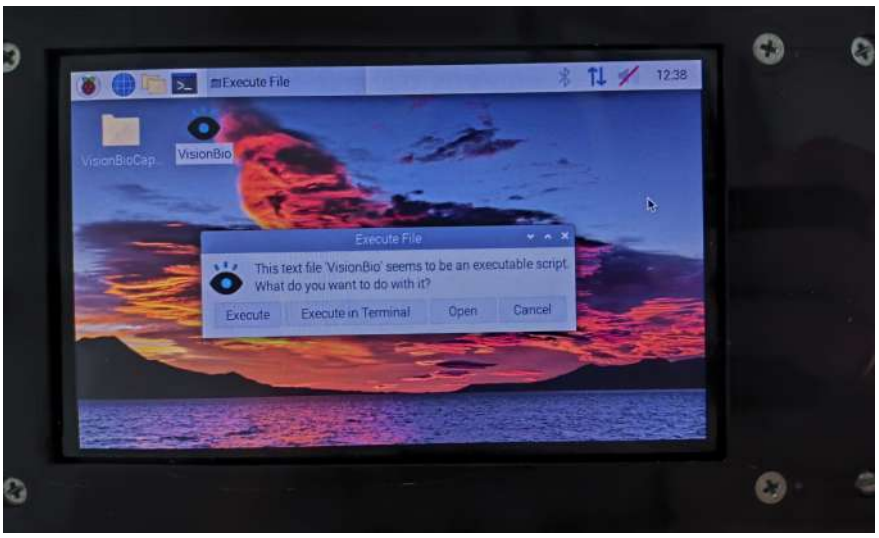


Figura B.1: Ejecución de la aplicación VisionBio desde el escritorio de la Raspberry Pi.

Al ejecutar la aplicación aparece un cuadro de diálogo solicitando la acción que se desea realizar sobre el archivo ejecutable. Para iniciar el programa únicamente es necesario seleccionar la opción *Execute*, tras lo cual se abrirá automáticamente la interfaz principal.

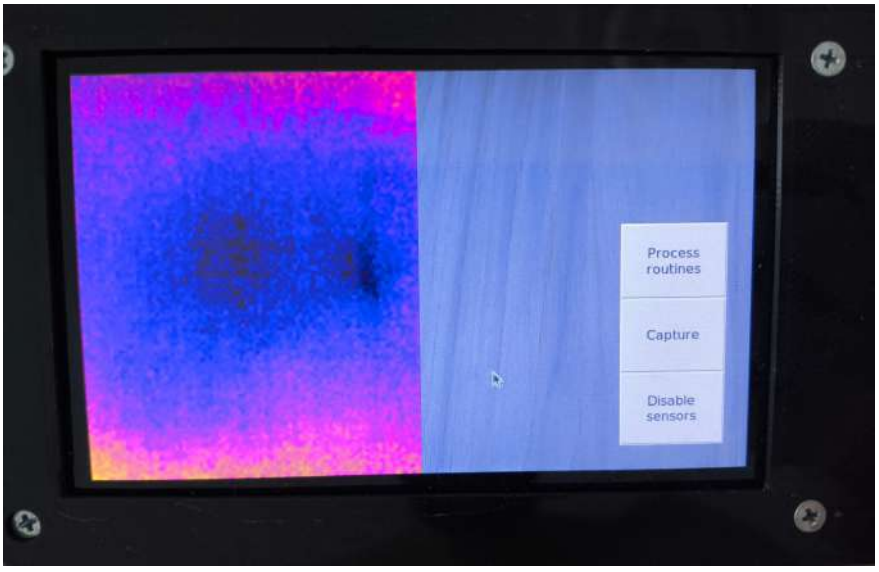


Figura B.2: Confirmación de ejecución de la aplicación.

Una vez completada la carga del programa, la plataforma muestra la interfaz principal, formada por dos zonas claramente diferenciadas. En la parte izquierda se visualiza en tiempo real la imagen capturada por el sensor infrarrojo, mientras que en la parte derecha aparece la imagen procedente de la cámara visible. Ambas adquisiciones se realizan de forma simultánea, permitiendo al usuario disponer de información térmica y visual durante toda la sesión.
En el lateral derecho de la interfaz se muestran los principales controles disponibles para el usuario. El botón *Process routines* permite acceder al listado de algoritmos disponibles para el procesamiento de la imagen térmica. El botón *Capture* realiza la adquisición simultánea de las imágenes visible e infrarroja, almacenándolas para su posterior análisis. Finalmente, el botón *Disable sensors* permite detener temporalmente la adquisición procedente de ambos sensores.

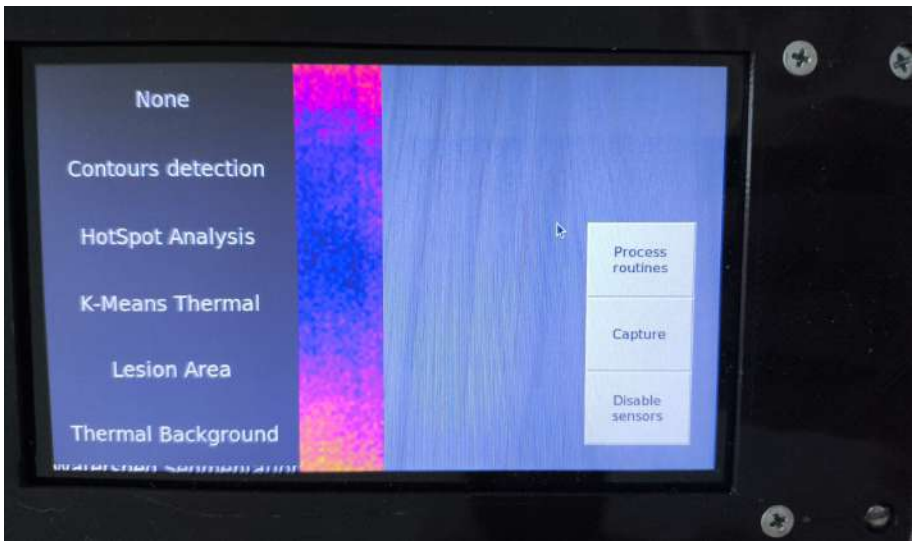


Figura B.3: Interfaz principal de la aplicación durante la adquisición en tiempo real.

B.2. Selección de los plugins

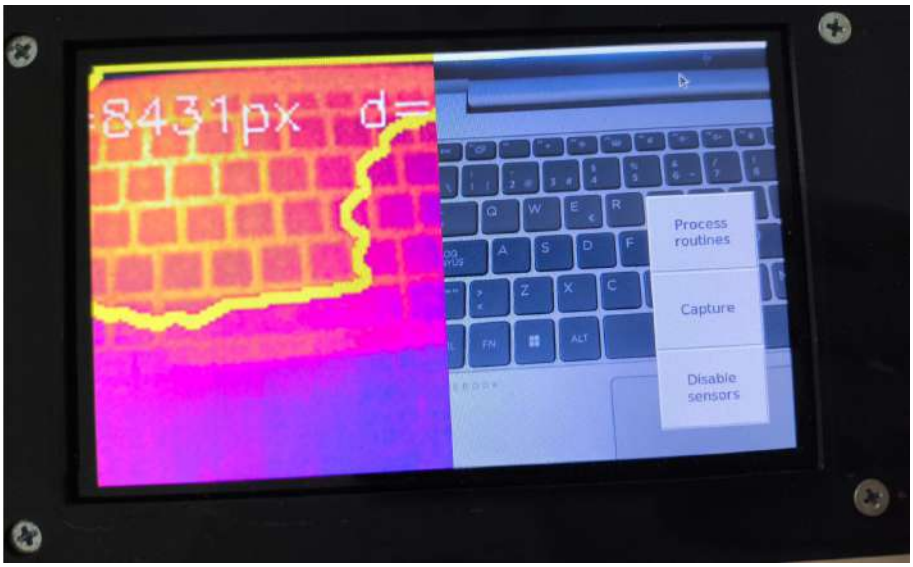


Figura B.4: Interfaz con plugin aplicado.

En este punto, podremos capturar la imagen para guardarla en sistema y su posterior análisis.

B.3. Consideraciones finales

La plataforma desarrollada presenta un funcionamiento sencillo e intuitivo, permitiendo que el usuario pueda realizar la adquisición y el procesamiento de imágenes térmicas con un número reducido de acciones. La integración de los distintos plugins dentro de una misma interfaz facilita la comparación entre algoritmos y permite visualizar de forma inmediata los resultados obtenidos sobre la imagen capturada.
Los plugins desarrollados durante este trabajo se integran de forma transparente en la aplicación, pudiendo seleccionarse desde el menú principal sin necesidad de realizar configuraciones adicio-

nales. Esto permite ampliar las capacidades de la plataforma manteniendo la misma experiencia de uso que el software original.
Aunque el sistema ha sido utilizado en un entorno de investigación, la estructura modular empleada durante el desarrollo facilita la incorporación de nuevos algoritmos de procesamiento en futuras versiones, contribuyendo a la evolución de la plataforma como herramienta de apoyo para el análisis de imágenes térmicas en el ámbito clínico.