

Trabajo Fin de Grado  
Grado en Ingeniería de las Tecnologías de  
Telecomunicación

Herramienta de ayuda al diagnóstico de lesiones  
vasculares basada en imágenes termográficas

Autor: Alejandro Torelli Durán

Tutoras: Begoña Acha Piñero, María del Carmen Serrano Gotarredona

Dpto. Teoría de la Señal y Comunicaciones  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2020





Trabajo Fin de Grado  
Grado en Ingeniería de las Tecnologías de Telecomunicación

# **Herramienta de ayuda al diagnóstico de lesiones vasculares basada en imágenes termográficas**

Autor:

Alejandro Torelli Durán

Tutoras:

Begoña Acha Piñero

Catedrática

María del Carmen Serrano Gotarredona

Catedrática

Dpto. de Teoría de la Señal y Comunicaciones

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2020



Trabajo Fin de Grado: Herramienta de ayuda al diagnóstico de lesiones vasculares basada en imágenes termográficas

Autor: Alejandro Torelli Durán

Tutoras: Begoña Acha Piñero, María del Carmen Serrano Gotarredona

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2020

El Secretario del Tribunal



*A mi familia*

*A mis compañeros*

*A mis docentes*





# Agradecimientos

---

En primer lugar, agradezco a mis tutoras, la Dra. D<sup>a</sup>. Begoña Acha Piñero y la Dra. D<sup>a</sup>. María del Carmen Serrano Gotarredona, su dedicación, agrado y buenas maneras para conmigo, no sólo durante la elaboración de este Trabajo de Fin de Grado, sino durante todo mi periplo universitario.

El siguiente agradecimiento excede los límites de este Trabajo de Fin de Grado: A mi familia. Papá, mamá y Javi, todo lo que fui, soy y seré se debe a vosotros tres. Gracias por llevarme siempre en volandas, por soportar los golpes conmigo y por enseñarme (y demostrarme) que “no puedo” no es una opción. Lo más bonito de mis metas es cruzarlas juntos.

Por último, doy gracias a todos los compañeros, amigos y profesores por ayudarme a llegar hasta aquí. De entre todas esas estrellas, hay dos que brillan muy por encima del resto: Ángela y Violeta. Habéis brindado la luz que necesitaba el final de este camino y, lo que es más importante, mi vida. Éste es solamente el primero de muchos senderos.

*Alenadro Torelli Durán*

*Sevilla, 2020*



# Resumen

---

El tratamiento de imágenes y la medicina caminan juntos desde 1895, cuando Whilhelm Konrad Röntgen descubrió los Rayos X [1]. Décadas de evolución han hecho posible que la imagen médica se convierta en una herramienta fundamental para la medicina. Métodos como la radiografía, la resonancia magnética o el ultrasonido son conocidos por todos [2].

En los últimos años, la termografía infrarroja (TIR) está tomando importancia en diferentes campos médicos [3]. Y es precisamente aquí donde se enmarca este Trabajo de Fin de Grado: El tratamiento digital de imágenes infrarrojas para el estudio de lesiones vasculares, tales como los hemangiomas<sup>1</sup>.

Hay muchas evidencias de que los hemangiomas están relacionados con una mayor velocidad de la sangre. Por ello, una cámara IR y un adecuado tratamiento de la imagen pueden ayudar al estudio y diagnóstico de esta enfermedad. Dicho tratamiento de imagen, que será el contenido de este Trabajo de Fin de Grado, será realizado con MATLAB. Consiste en añadir un fondo negro a cada imagen que elimine las partes que no son objeto de estudio; y en realizar mapas de color dependientes de la temperatura (generales y relativos a cada punto que el médico quiera estudiar) que permitan establecer probabilidades de encontrar niveles de velocidad en sangre similares o, en otras palabras, probabilidades de encontrar nuevos hemangiomas (presentes y futuros).

Por último, este Proyecto también pretende afianzar la relación entre velocidad en sangre y lesiones vasculares.

---

<sup>1</sup>Los hemangiomas son tumores no cancerosos causados por el crecimiento anormal de los vasos sanguíneos. Comúnmente se los conoce como marcas rojas de nacimiento [4].



# Abstract

---

Image processing and medicine walk together since 1895, when Whilhelm Konrad Röntgen discovered the X-rays [1]. Decades of evolution have made possible that medical image becomes an essential tool for medicine. Methods like radiography, magentic resonance or ultrasound are known by everyone [2].

In the last few years, infrared thermography (IRT) is getting stronger in several medical fields [3]. And this is where this End-of-degree Project belongs: Digital image processing for studying vascular lesions, such as hemangiomas<sup>2</sup>.

Lots of evidences let us know that hemangiomas are related to a higher blood speed. That's why an IR camera and an appropriate image preocessing can help study and diagnose hemangiomas. This image processing, which will be the content of the End-of-degree Project, will be performed with MATLAB. It consists in adding a black background to every image that eliminates the parts that are not necessary for studying; and it also consists in developing colored maps that estimates temperature levels (general levels and relative levels that depend on the exact part of the image which is going to be studyied by the doctor) which let us establish priors for finding specific blood speeds or, in other words, establish priors for finding new hemangiomas (those that exist now and those that might exist in the future).

Finally, this Project also tries to strengthen the relation between blood speed and vascular lesions.

---

<sup>2</sup>Hemangiomas are non-cancerous tumors caused by anormal growth of blood vessels. They are commonly known as strawberry birthmarks [4].



# Índice

---

<b>Agradecimientos</b>	<b>ixx</b>
<b>Resumen</b>	<b>xi</b>
<b>Abstract</b>	<b>xiii</b>
<b>Índice</b>	<b>xv</b>
<b>Índice de Tablas</b>	<b>xvii</b>
<b>Índice de Figuras</b>	<b>xix</b>
<b>1 Introducción</b>	<b>1</b>
<b>2 Recursos empleados</b>	<b>5</b>
2.1. <i>Cámara IR</i>	5
2.2. <i>MATLAB</i>	6
2.2.1 Función de apertura de archivos .BIN	6
<b>3 Desarrollo de la Herramienta</b>	<b>9</b>
3.1. <i>Eliminación del fondo de la imagen</i>	9
3.1.1. Técnicas de detección de bordes	10
3.1.2. Técnicas de umbralización	11
3.1.3. Técnicas de umbralización con relleno de huecos	16
3.2. <i>Normalización y tratamiento previo de la imagen</i>	24
3.3. <i>Mapa de probabilidades de detección de lesiones vasculares</i>	25
3.3.1. Mapa de probabilidades generalizado	25
3.3.2. Mapa de probabilidades relativo usando Pixel Select	34
<b>4 Resultados</b>	<b>43</b>
<b>5 Conclusiones y líneas futuras</b>	<b>69</b>
<b>Anexo A: Funciones internas de MATLAB</b>	<b>71</b>
<b>Referencias</b>	<b>83</b>
<b>Glosario</b>	<b>85</b>





# Índice de Tablas

---

Tabla 4-1. Valoración esquemática de resultados.

68



# Índice de Figuras

---

Figura 1-1. Diagrama explicativo	2
Figura 2-1. Mapa de color Jet	7
Figura 2-2. Gráfico de altura de temperatura de color	8
Figura 2-3. Imagen en falso color	8
Figura 3-1. Imagen original en falso color	10
Figura 3-2. Resultado de aplicación de Máscara Sobel	10
Figura 3-3. Relleno de huecos sobre contorno no cerrado	11
Figura 3-4. Plano R, plano G y plano B de la imagen de la Figura 3-1	12
Figura 3-5. Plano S de la imagen	13
Figura 3-6. Máscara binaria tras umbralización con level=0.3	13
Figura 3-7. Imagen resultante tras aplicar máscara con level=0.3	15
Figura 3-8. Máscara binaria tras umbralización con level=0.9	15
Figura 3-9. Imagen resultante tras aplicar máscara con level=0.9	16
Figura 3-10. Máscara binaria tras umbralización con level=0.6	17
Figura 3-11. Máscara binaria tras umbralización con level=0.6 e imfill	17
Figura 3-12. Captura de MATLAB en el momento de rellenar los posibles huecos restantes	18
Figura 3-13. Hueco clickeado con detalle de asterisco cian y morado	18
Figura 3-14. Máscara definitiva	19
Figura 3-15 a. Ejemplo 2 a tamaño real	19
Figura 3-15 b. Ejemplo 2 ampliado para observar en detalle	20
Figura 3-16 a. Ejemplo 3 a tamaño real	20
Figura 3-16 b. Ejemplo 3 ampliado para observar en detalle	20
Figura 3-17 a. Ejemplo 4 a tamaño real	21
Figura 3-17 b. Ejemplo 4 ampliado para observar en detalle	21
Figura 3-18 a. Ejemplo 5 a tamaño real	21
Figura 3-18 b. Ejemplo 5 ampliado para observar en detalle	22
Figura 3-19 a. Ejemplo 6 a tamaño real	22
Figura 3-19 b. Ejemplo 6 ampliado para observar en detalle	22
Figura 3-20 a. Ejemplo 7 a tamaño real	23
Figura 3-20 b. Ejemplo 7 ampliado para observar en detalle	23
Figura 3.21 a. Ejemplo 8 a tamaño real	23
Figura 3-21 b. Ejemplo 8 ampliado para observar en detalle	24
Figura 3-22 a. Ejemplo 9 a tamaño real	24
Figura 3-22 b. Ejemplo 9 ampliado para observar en detalle	24

Figura 3-23. Aumento de contraste en imagen, junto a variación de histograma	26
Figura 3-24. Resultado de aplicar imadjust	27
Figura 3-25. Resultado de aplicar Imadjust e Imsharpen	28
Figura 3-26. Máscara de extracción de zona de máxima probabilidad	29
Figura 3-27. Imagen original con zona de máxima probabilidad en rojo	30
Figura 3-28. Máscara de extracción de zona de probabilidad media, sin filtrar	30
Figura 3-29. Máscara de extracción de zona de probabilidad media, filtrada	31
Figura 3-30. Imagen original con zona de máxima probabilidad en rojo y zona de probabilidad media en amarillo	32
Figura 3-31. Máscara de extracción de zona de probabilidad baja, sin filtrar	33
Figura 3-32. Máscara de extracción de zona de probabilidad baja, filtrada	33
Figura 3-33. Mapa de probabilidades generalizado	34
Figura 3-34. Captura de MATLAB en el momento de seleccionar la zona a estudiar	35
Figura 3-35. Zoom a la ventana “Figura 18” de la Figura 3-34	35
Figura 3-36. Selección de la zona a estudiar	36
Figura 3-37. Máscara de extracción de Zona de máxima probabilidad relativa al píxel seleccionado	37
Figura 3-38. Imagen original con zona de máxima probabilidad relativa en rojo	38
Figura 3-39. Máscara de extracción de Zona de media probabilidad relativa al píxel seleccionado	39
Figura 3-40. Imagen original con zona de máxima probabilidad relativa en rojo y media probabilidad relativa en amarillo	39
Figura 3-41. Máscara de extracción de Zona de baja probabilidad relativa al píxel seleccio	40
Figura 3-42. Mapa de probabilidades relativo usando Pixel Select	40
Figura 3-43. Imagen con zona blanca, correspondiente a temperaturas relativas muy elevadas	41
Figura 4-1. Leyenda Capítulo 4	44
Figura 4-2. Resultados en Paciente 1	45
Figura 4-3. Resultados en Paciente 2	47
Figura 4-4. Resultados en Paciente 3	49
Figura 4-5. Resultados en Paciente 4	51
Figura 4-6. Resultados en Paciente 5	53
Figura 4-7. Resultados en Paciente 6	55
Figura 4-8. Resultados en Paciente 7	57
Figura 4-9. Resultados en Paciente 8	58
Figura 4-10. Resultados en Paciente 9	60
Figura 4-11. Resultados en Paciente 10	62
Figura 4-12. Resultados en Paciente 11, caso 1	64
Figura 4-13. Resultados en Paciente 11, caso 2	66
Figura A-1. Extracto de la página de ayuda de la función colormap	71
Figura A-2. Extracto de la página de ayuda de la función edge	72
Figura A-3. Extracto de la página de ayuda de la función figure	72
Figura A-4. Extracto de la página de ayuda de la función find	73

Figura A-5. Extracto de la página de ayuda de la función <code>fliplr</code>	73
Figura A-6. Extracto de la página de ayuda de la función <code>fopen</code>	74
Figura A-7. Extracto de la página de ayuda de la función <code>fread</code>	75
Figura A-8. Extracto de la página de ayuda de la función <code>im2bw</code>	75
Figura A-9. Extracto de la página de ayuda de la función <code>imadjust</code>	76
Figura A-10. Extracto de la página de ayuda de la función <code>imclearborder</code>	76
Figura A-11. Extracto de la página de ayuda de la función <code>imdilate</code>	77
Figura A-12. Extracto de la página de ayuda de la función <code>imfill</code>	78
Figura A-13. Extracto de la página de ayuda de la función <code>imsharpen</code>	78
Figura A-14. Extracto de la página de ayuda de la función <code>mean</code>	79
Figura A-15. Extracto de la página de ayuda de la función <code>mesh</code>	79
Figura A-16. Extracto de la página de ayuda de la función <code>rgb2hsv</code>	80
Figura A-17. Extracto de la página de ayuda de la función <code>strel</code>	81



# 1 INTRODUCCIÓN

---

*Los proyectos que hemos completado demuestran lo que ya sabemos. Los proyectos futuros deciden lo que vamos a aprender.*

*- Mohsin Tiwana -*

La motivación del presente Proyecto se encuentra íntimamente ligada con la investigación médica. Profesionales de la medicina han encontrado evidencias de la relación entre diferentes lesiones vasculares y la velocidad de la sangre. Para profundizar en este planteamiento, y a través de la colaboración con Departamento de Teoría de la Señal y Comunicaciones de la Universidad de Sevilla, se proporciona apoyo tecnológico e informático que ayude a avanzar en estas investigaciones. Así, y como objetivo último, se pretende que lo que un día nació como una simple idea, en un futuro llegue a ser una herramienta cotidiana en cualquier hospital. Es decir, ayudar en el diagnóstico, estudio y tratamiento de enfermedades.

Así, y concretando más en este Trabajo de Fin de Grado, el propósito y los objetivos que persigue están relacionados con el estudio de imágenes termográficas. A partir de las imágenes infrarrojas de los pacientes se va a desarrollar una Herramienta que adecúe las condiciones de dichas imágenes y que, además, realice un análisis gráfico relacionado con la temperatura corporal de los pacientes.

Las condiciones a adecuar en las imágenes son básicamente las relacionadas con información sobrante. Es decir, el único contenido importante y susceptible de estudio en las imágenes es el que corresponde al cuerpo del paciente, por lo que hay que eliminar el resto. Expresado en un lenguaje más coloquial, eliminar el fondo de la imagen.

Discriminando el fondo de la imagen no sólo se consigue un estudio más preciso de la imagen analizada, sino que se otorga un carácter de homogeneidad a todos los pacientes estudiados y por estudiar. Por supuesto, es de suma importancia que este tratamiento a la imagen no deje información residual. Pero aún es mucho más importante que no elimine áreas susceptibles de estudio, ya que la omisión de un único píxel puede marcar la diferencia entre localizar una lesión o no hacerlo.

Por ello se realizarán los tratamientos de imagen necesarios para asegurar que la imagen resultante cumpla los objetivos fijados, a la vez que mantenga el 100% de la información útil que contenía la imagen original. Para realizar estas tareas (además de otras pertenecientes a este primer “bloque”, como la de normalización, o la de lectura de archivos binarios) se usará el software MATLAB.

Por otro lado, y antes de explicar en qué consiste el análisis a las imágenes térmicas propuesto para este Trabajo de Fin de Grado, es necesario explicar el porqué de la utilización de este tipo de imágenes. Para ello, se va a usar como ejemplo una lesión vascular muy común y recurrente en los casos tratados en este Proyecto, los hemangiomas:

Como se ha mencionado en páginas anteriores, los hemangiomas son tumores benignos que aparecen como consecuencia del desarrollo anormal de los vasos sanguíneos. Pueden estar presentes al nacer, pero el caso más común es el de la aparición en las primeras semanas de vida. De los hemangiomas puede derivar la aparición de hemorragias, así como dificultades en la respiración y alimentación, o problemas de crecimiento y visión. Si la tumoración ocurre en vasos muy internos del cuerpo, se suele recurrir a técnicas como la resonancia magnética o el TAC para su diagnóstico [5].

El hemangioma infantil es el tumor vascular más frecuente en niños. En muchos casos crecen durante un tiempo, para pasar posteriormente a un proceso de involución que puede desembocar incluso en la desaparición del tumor. Destacan generalmente por su tono rojo o azul [6]. Resaltar que la parte médica ligada a este Proyecto pertenece a la rama de la pediatría.

El estudio de hemangiomas, así como el de otras enfermedades vasculares, que abarca este Trabajo de Fin de Grado está centrado en una característica muy concreta de los mismos: La relación entre su aparición y el aumento de velocidad en la sangre.

Hay evidencias de que la sangre viaja a velocidades mayores en zonas que presentan ciertos tipos de lesiones. Y, como es sabido, el incremento de velocidad en las partículas genera calor. Por tanto, es en este punto donde entran a escena las imágenes termográficas, puesto que permiten discriminar las distintas zonas del cuerpo según su temperatura. De la relación entre las zonas lesionadas y el mapa de temperatura corporal nace este Proyecto, que pretende establecer niveles de concordancia entre valores de temperatura específicos y lesiones vasculares. Es decir, a partir de una imagen infrarroja se va a obtener un mapa de probabilidades de encontrar hemangiomas u otras lesiones en ella. Este hecho dispararía las posibilidades de estudio, permitiendo observar la extensión real de la lesión, dónde se podrían encontrar nuevas lesiones no detectadas con otros métodos, o incluso dónde se podrían formar en un futuro. Incluso la información que no refleje la imagen térmica es importante, ya que no todas las lesiones vasculares presentan altas velocidades de sangre. Por ello, el hecho de que una lesión tenga alto flujo o no también ayuda al médico a diagnosticar el tipo de lesión vascular y predecir su evolución.

Por tanto, el análisis de las imágenes termográficas referido unas líneas más arriba consistirá en la creación de mapas de probabilidad de localización de lesiones vasculares. De hecho, y con el objetivo de ampliar el abanico de recursos técnicos para el personal médico encargado del estudio, se realizarán dos tipos de mapa: Uno genérico y otro relativo al punto (temperatura) de la imagen que elija el profesional. Ambos mapas serán desarrollados usando el software MATLAB. Por supuesto, los dos mapas serán impresos de forma independiente sobre la propia imagen original, obteniendo así dos nuevas imágenes que serán presentadas en pantalla.

Por otro lado, es importante destacar que dentro de los límites de este Trabajo de Fin de Grado se encuentran el desarrollo de la Herramienta, y la realización de un análisis o estudio sobre la concordancia entre las temperaturas corporales y las lesiones diagnosticadas por el personal médico pertinente (dicho análisis se encuentra en el Capítulo 4 y el Capítulo 5 de este Documento). No se pretende realizar un diagnóstico alternativo que contradiga las opiniones médicas. También queda fuera de los límites del Trabajo de Fin de Grado la realización de fotografías a los pacientes, debido a que en todo momento se trabaja con imágenes ya capturadas por profesionales médicos en la consulta.

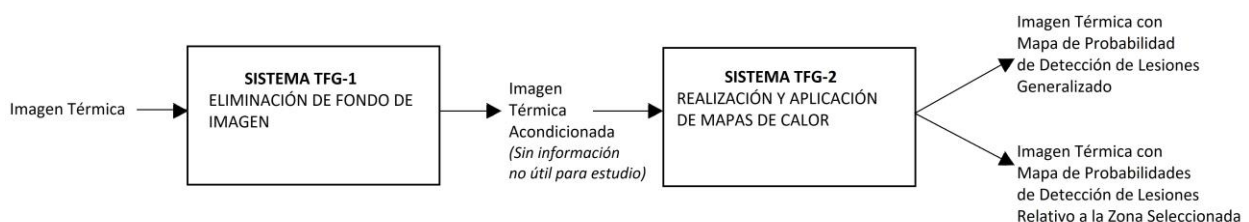


Figura 1-1. Diagrama explicativo.



Finalmente, se adjunta a estas líneas un diagrama (Figura 1-1) que resume de manera muy simplificada las tareas que lleva a cabo la Herramienta a desarrollar en este Trabajo de Fin de Grado. Cabe destacar nuevamente que las entradas al primer sistema son proporcionadas por el personal médico.



## 2 RECURSOS EMPLEADOS

---

*La ciencia no conoce su deuda a la imaginación.*

*- Ralph Waldo Emerson -*

Para poder desarrollar una herramienta que lleve a cabo los estudios sobre lesiones vasculares requeridos en este Trabajo de Fin de Grado, se necesitarán principalmente dos elementos: Imágenes termográficas captadas en pacientes reales (acompañadas de un diagnóstico médico que permita contrastar los resultados obtenidos), y un medio sobre el que llevar a cabo el tratamiento digital de dichas imágenes.

Una cámara IR será la encargada de tomar capturas a los pacientes, permitiendo la obtención de imágenes termográficas. Éstas serán analizadas, tratadas y presentadas usando el programa MATLAB, considerado uno de los medios más potentes para el tratamiento de imágenes.

### 2.1. Cámara IR

Las lesiones de todos los pacientes estudiados han sido capturadas utilizando la misma cámara infrarroja. Es decir, todas las imágenes termográficas analizadas y tratadas en el presente Proyecto provienen del mismo equipo fotográfico.

Como toda cámara infrarroja (también llamada cámara térmica), su funcionamiento se basa en la formación de imágenes a partir de la emisión infrarroja de los objetos que detecta [7]. Cada imagen capturada por esta cámara estará disponible para este Proyecto en dos formatos con características diferentes:

1. La imagen original, en escala de grises, queda almacenada como archivo binario .BIN. Se trata de la imagen que captura y almacena la propia cámara. El nombre de estas imágenes es 'rawX.bin' (sin comillas y siendo X el número de la captura) y su tamaño es siempre el mismo: 9.37 KB.
2. La imagen coloreada (es decir, en falso color), se almacena usando el conocido formato de imagen .JPG [8]. El proceso de falso coloreado no lo realiza la cámara IR, sino que es tarea del software MATLAB. El nombre de estas imágenes es 'rgbX.jpg' (sin comillas y siendo X el número de la captura) y su tamaño oscila entre los 5 KB y los 7 KB.

Llama la atención el reducido espacio que ocupa la información almacenada, tanto en un formato como en otro. Esto se debe a que las imágenes que captura la cámara IR son muy pequeñas. Su resolución es de tan solo 80 x 60 píxeles, es decir, hasta dos órdenes de magnitud menos que las imágenes obtenidas por casi cualquier cámara de smartphone actual. Sin duda nos encontramos ante una característica ciertamente limitante (como se demostrará a lo largo del presente Documento) que, sin embargo, no supondrá en ningún momento un obstáculo insalvable para que se alcancen los objetivos marcados en el Proyecto.

## 2.2. MATLAB

La potencia del software MATLAB es de sobra conocida. Científicos e ingenieros de todo el mundo utilizan MATLAB como instrumento fundamental en campos tan diversos como el cálculo, álgebra, transmisión de datos, estadística, física o tratamiento de imágenes [9]. MATLAB contiene un gran número de funciones que implementan los principales algoritmos de análisis de datos y procesamiento de señales e imágenes, por lo que es una herramienta tan usada en docencia e investigación. Esto añadido a su potencia en la realización de cálculos y procesamiento de datos, el sencillo lenguaje de programación (propio) que posee, la formidable capacidad de representación de los resultados y lo intuitivo de su GUI hacen de MATLAB un medio sobradamente preparado para alcanzar los objetivos de este Trabajo Fin de Grado.

Como indica su propio nombre, MATLAB, abreviatura de *MATRIX LABORATORY* [10], encuentra su esencia en el trabajo con matrices. Al fin y al cabo, una imagen es una matriz bidimensional (o tridimensional, si trabajamos en espacios de color como RGB y HSV) en la que cada pixel se corresponde con un elemento de dicha matriz. Esto, unido a lo expuesto en el párrafo anterior, posiciona a MATLAB como referencia puntera para el tratamiento de imágenes.

Con el objetivo de agilizar la lectura y comprensión de este Documento, las distintas funciones internas empleadas en el desarrollo de la Herramienta que abarca el presente Trabajo de Fin de Grado se irán desglosando y explicando conforme vayan apareciendo en sus correspondientes capítulos. No obstante, si se desea ampliar dicha información, se recomienda al lector acudir al Anexo A, donde se encuentra el contenido completo que proporciona la ayuda de MATLAB (a través de su web oficial, [www.mathworks.com](http://www.mathworks.com)) en relación a las funciones internas más relevantes que se han usado durante el desarrollo del Proyecto.

Las representaciones gráficas son de importancia capital en este Trabajo de Fin de Grado, en tanto en cuanto constituyen la forma de expresar los resultados obtenidos. No hay que olvidar que el objetivo final es proporcionar al médico una imagen cargada de información útil (y seleccionable por él mismo) que sólo es posible obtener mediante tratamiento digital.

MATLAB muestra dichas representaciones en diferentes ventanas, que serán capturadas y agregadas como muestras a este Documento para que el lector pueda comprobar los resultados como si estuviera ejecutando el programa.

### 2.2.1 Función de apertura de archivos .BIN

La función *read\_bin\_thermal\_image* no ha sido realizada ni modificada por mí, sino que se trata de un archivo cedido por la Tutoría de este Trabajo de Fin de Grado. Esta función lee una imagen almacenada en formato binario (.BIN) y la convierte en imagen de falso color.

Las imágenes capturadas por la cámara IR se guardan en formato .BIN (y no en otros más comunes, como .JPG). Por tanto, para poder trabajar con estas imágenes, será necesario abrir y manejar archivos .BIN.

El problema de este tipo de archivos en formato binario [11] es que no pueden ser abiertos de forma nativa por MATLAB. Se necesita pues una función que realice esta tarea.

A continuación se muestra cómo *read\_bin\_thermal\_image* es capaz de llevarlo a cabo:

```
fid = fopen('raw1.bin', 'r');
A=fread(fid,'uint16');
```

Se ha usado como ejemplo la imagen 'raw1.bin', siguiendo el formato de nombre de archivos usado por la cámara IR: 'rawX.bin'. Puede observarse que el archivo se lee como una variable del tipo matriz de enteros sin signo de 16 bits (uint16) [12].

En este punto aún no se puede representar la imagen en pantalla, ya que se encuentra "desordenada". Por ello, será necesario almacenar el contenido útil de la variable "A" (matriz de tamaño 480x1) en la variable "IM" (matriz de tamaño 80x60), de suma importancia puesto que será la imagen base sobre la que realizar el tratamiento requerido en este Trabajo de Fin de Grado.

```
IM=zeros(N,M);  
for i=1:N  
    for j=1:M  
  
        IM(i,j)=A(k);  
        k=k+1;  
  
    end  
end
```

Una vez almacenado el contenido de la imagen en un formato adecuado para MATLAB, la función *read\_bin\_thermal\_image* le aplicará un mapa de color mediante la función integrada *colormap*. En este caso, el mapa elegido es *Jet* [13], muy común entre los mapas térmicos porque abarca desde los colores fríos hasta los cálidos, lo que mejora la comprensión visual de la información contenida en la imagen. Es necesario remarcar que la imagen coloreada no será objeto de tratamiento, sino que lo será su versión original en escala de grises “IM”.



Figura 2-1. Mapa de color Jet. Recuperada de [www.mathworks.com](http://www.mathworks.com)

Además de aplicar un mapa de color adecuado, la función *read\_bin\_thermal\_image* normaliza la imagen almacenada en IM y presenta en pantalla un muy útil gráfico que representa cada pixel con una altura según la temperatura del falso color aplicado (Figura 2-2). Finalmente, también presenta en pantalla la imagen completa en falso color, junto a una barra vertical que hace las veces de leyenda para las temperaturas de color aplicadas (Figura 2-3).

```
MinVal=min(A(3:end));  
MaxVal=max(A(3:end));  
k=3;  
IM=IM-MinVal;  
mesh(IM)  
figure  
imagesc(fliplr(IM),[0 MaxVal-MinVal]),colormap(jet),colorbar('vert')
```

Por tanto, la función *read\_bin\_thermal\_image* supone la verdadera piedra de partida para este Trabajo de Fin de Grado, debido a que no solamente realiza la fundamental tarea de leer la imagen original captada por la cámara IR y almacenarla en formato compatible para MATLAB, sino que además realiza unas presentaciones en pantalla que serán muy importantes para el estudio a realizar.

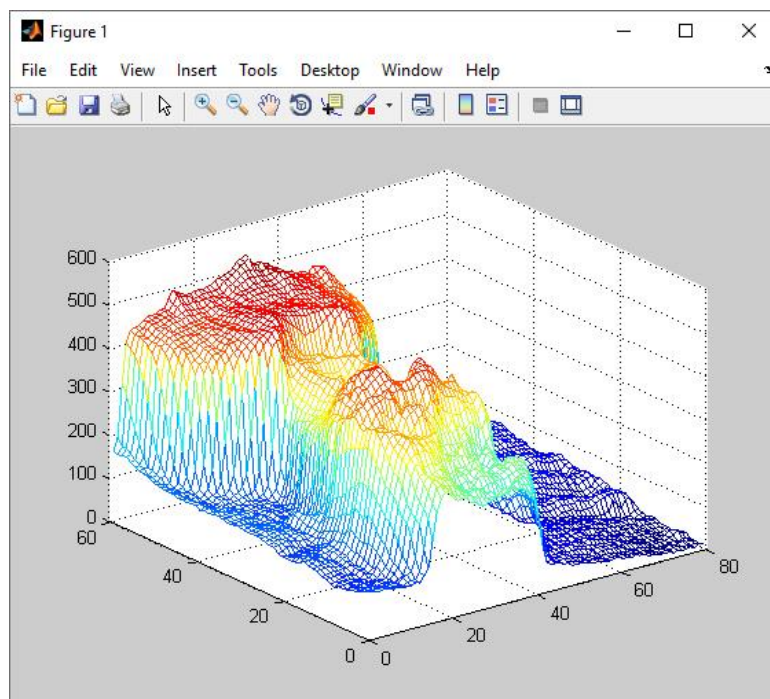


Figura 2-2. Gráfico de altura de temperatura de color.

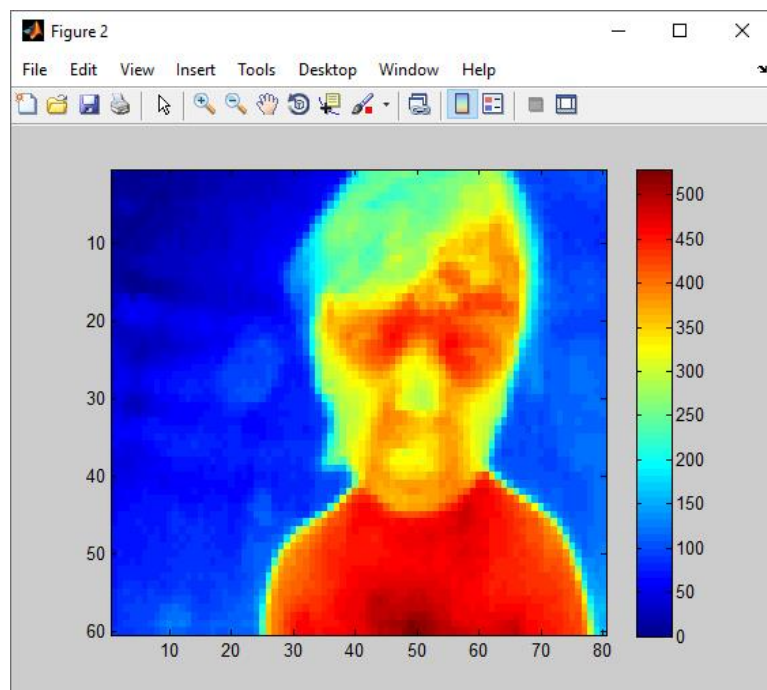


Figura 2-3. Imagen en falso color.

# 3 DESARROLLO DE LA HERRAMIENTA

---

*Si cierras la puerta a todos los errores, también la verdad se quedará fuera.*

*- Rabindranath Tagore -*

El despliegue de la solución al problema propuesto será el objetivo de la sección que nos ocupa. Se sustenta en dos pilares fundamentales: Establecer unas condiciones comunes a todas las imágenes a capturar, a través de la normalización selectiva y de la eliminación del fondo; y elaborar mapas de probabilidad de detección de lesiones vasculares según la velocidad de circulación de la sangre, es decir, la temperatura captada por la cámara IR.

## 3.1 Eliminación del fondo de la imagen

El primer paso para poder realizar un estudio concreto y fidedigno de la imagen es el de discriminar las partes de la misma que no sean útiles, evitando así que puedan alterar el resultado final. En los casos que abarca este Proyecto (imágenes termográficas de cuerpos), esto se traduce en “borrar” el fondo para que dé la impresión de que la captura se ha realizado con un telón negro de fondo.

La implantación de un fondo negro también resulta más cómoda e intuitiva al ojo humano, ya que todo lo que aprecie en las imágenes distinto del color negro será susceptible de ser analizado. No existirá la duda sobre si cierta parte pertenece al cuerpo o al fondo, algo para nada trivial tratándose de imágenes térmicas en escala de grises.

Dada la naturaleza matricial de MATLAB, el camino lógico para conseguir este objetivo pasa por crear una máscara que abarque la figura humana. Es decir, el cuerpo blanco (valor 1) sobre el fondo negro (valor 0). Obteniendo esta máscara, aplicarla a la imagen original es un proceso casi inmediato. Simplemente hay que multiplicar la imagen original por los unos y ceros que forman la imagen máscara, lo cual no es más que una multiplicación de matrices, operación extremadamente sencilla para MATLAB.

Hay que recordar que la solución encontrada debe ser válida para todas las imágenes termográficas capturadas por la cámara IR, tanto las analizadas en este Trabajo de Fin de Grado como las correspondientes a nuevos pacientes en el futuro. De nada nos serviría una herramienta válida únicamente para algunas imágenes.

Por todo ello, se han usado diferentes técnicas y métodos con el objetivo de conseguir generar la máscara deseada para cada imagen. Las siguientes subsecciones y apartados explicarán los más relevantes, en orden cronológico.

### 3.1.1 Técnicas de detección de bordes

La detección de bordes es una de las técnicas de tratamiento de imagen más usadas en MATLAB, debido a su robusto funcionamiento y a la multitud de posibilidades posteriores que brinda al programador [14]. Su naturaleza casa con la técnica que se quiere aplicar a las imágenes del Proyecto, por lo que resulta obvio optar por esta solución.

El objetivo es utilizar técnicas de detección de bordes para trazar la silueta humana y, a continuación, rellenar el interior de blanco, quedando el resto de la imagen en negro. Esto nos brindaría la máscara deseada y lo haría de forma inteligente, sin importar de qué imagen se trata.

Se utiliza una máscara de Sobel para detectar bordes. A través de la función interna de MATLAB *edge*, se aplica esta máscara binaria a cada una de las tres componentes del espacio RGB para juntar los resultados en una sola imagen:

```

I=imread('rgb1.jpg');
bsobelR=edge(I(:, :, 1),'sobel');
bsobelG=edge(I(:, :, 2),'sobel');
bsobelB=edge(I(:, :, 3),'sobel');
bsobelRGB=bsobelB | bsobelG | bsobelB;
figure, imshow (bsobelRGB);

```

La imagen original que servirá de ejemplo para toda la sección, en falso color, se representa en la Figura 3-1.



Figura 3-1. Imagen original en falso color.

Y el resultado de aplicar la detección de bordes se muestra en la Figura 3-2.

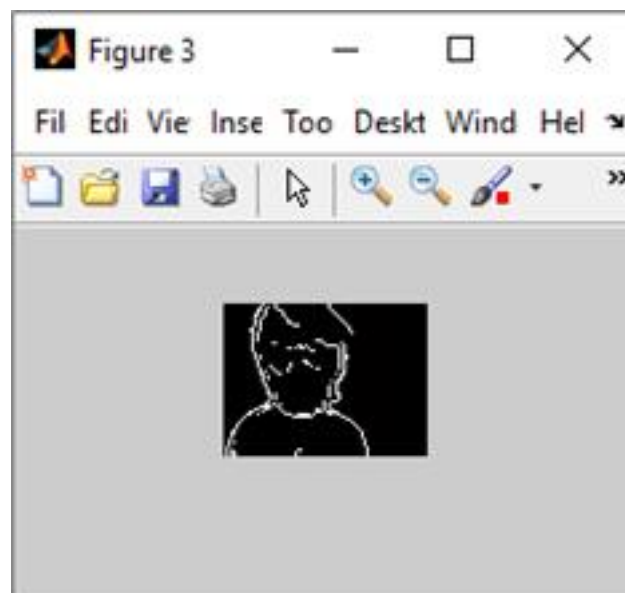


Figura 3-2. Resultado de aplicación de Máscara Sobel.

Puede observarse un problema que llama la atención: El contorno no aparece completamente cerrado. Si tratamos de rellenar el interior de la figura en color blanco, al no tener un contorno cerrado, se “saldrá” y el resultado no será ni mucho menos el deseado, como puede observarse en la Figura 3-3. Para rellenar se ha usado la función *imfill*.



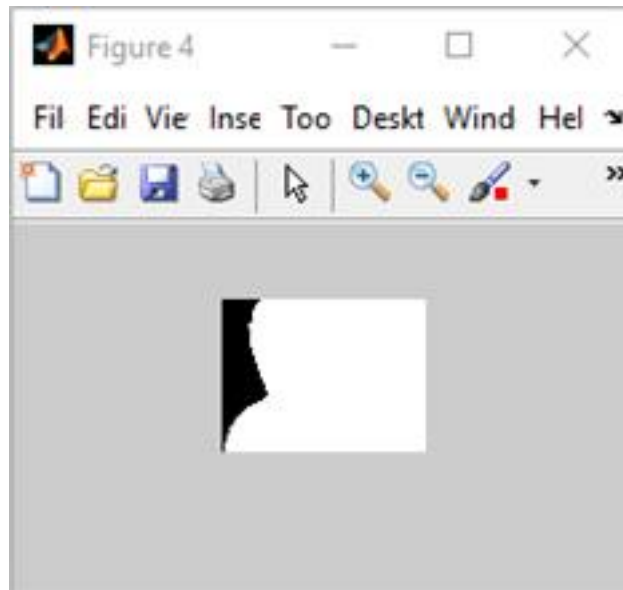


Figura 3-3. Relleno de huecos sobre contorno no cerrado.

Por supuesto, este problema no está carente de solución. Un buen ejemplo sería aplicar a la Figura 3-2 un tratamiento basado en:

1. Crear elementos de estructuración lineal mediante la función integrada *strel*, eligiendo los parámetros adecuados de longitud y ángulo.
2. Dilatar la máscara usando *imdilate* y los elementos de estructuración calculados en el punto anterior.
3. Rellenar los huecos interiores con *imfill*.
4. Eliminar los objetos conectados en el borde usando la función *imclearborder*, ajustando el parámetro conectividad de píxeles.
5. Suavizar el objeto, nuevamente usando *strel* y ajustando de nuevo distintos parámetros, según la estructura morfológica elegida para tal fin.

De esta forma se obtienen resultados muy cercanos a los deseados. El inconveniente se encuentra en que la realización de los puntos anteriores conlleva la elección de unos parámetros con valores no comunes a todas las imágenes. Por tanto, para cada imagen habría que buscar, sondear y seleccionar varios valores óptimos, lo cual choca frontalmente con el dinamismo necesario para estudiar varias imágenes a la vez o para revisar decenas de imágenes en poco tiempo. Los valores que resultan aceptables para unas imágenes resultan no serlo en absoluto para otras. En definitiva, esta técnica no se puede considerar óptima porque necesita mucho tratamiento individualizado (es decir, manual) para cada imagen si se quiere obtener un resultado adecuado.

### 3.1.2 Técnicas de umbralización

Dado el elevado número de parámetros a ajustar para obtener resultados deseados mediante técnicas de detección de bordes, la solución natural pasa por encontrar un método que reduzca el número de ajustes necesarios.

Por ello se ha elegido la umbralización. Ésta consiste en establecer un nivel de referencia dentro de una escala del 0 (negro absoluto) al 1 (blanco absoluto). Todos los píxeles que estén por debajo de ese nivel de referencia pasarán a tener valor 0 y todos los que estén por encima tendrán 1 como valor [15].

La primera duda que surge al pensar en este método es: ¿Cómo se va a elegir un valor que únicamente discrimine el fondo, dejando la silueta como está? Es una duda razonable, debido a que el método de umbralización por excelencia en MATLAB, *im2bw*, así como la propia naturaleza teórica de la técnica, actúan sobre imágenes en escala de grises, por lo que el fondo y la figura humana no están tan diferenciadas como para hacer una discriminación exenta de problemas. Como muestra, en la Figura 3-4 se muestran los planos R, G y B de la imagen mostrada en la Figura 3-1. Es fácil observar que ninguno de los tres casos se puede

establecer un límite que abarque a la figura y no al fondo, y viceversa.

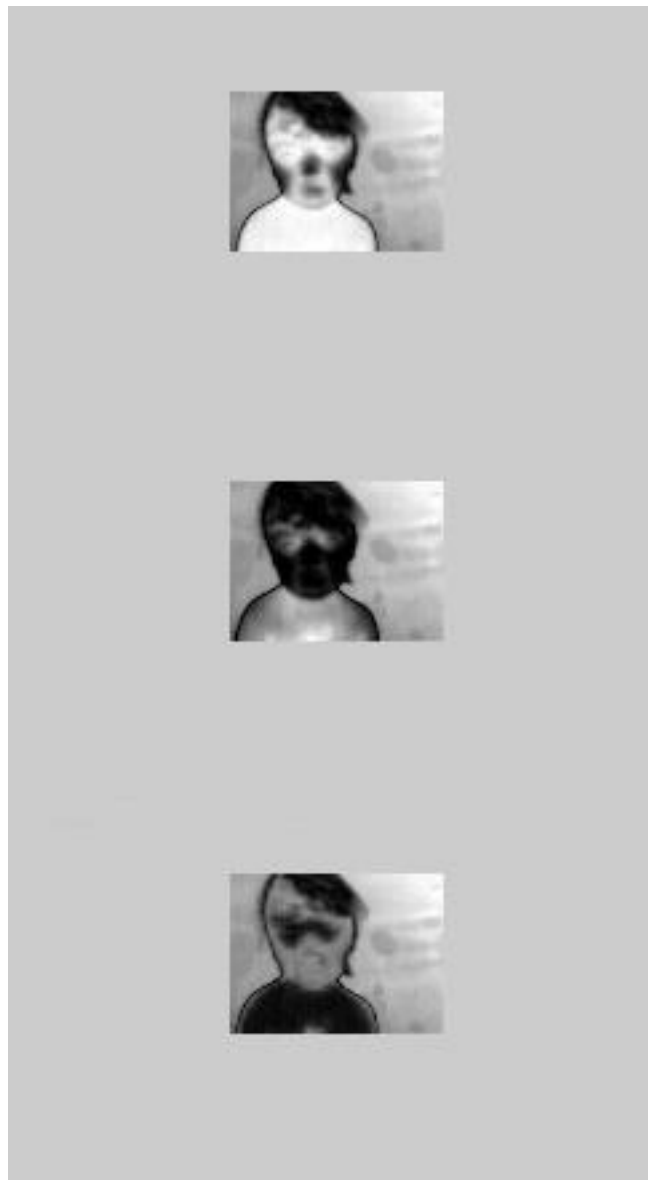


Figura 3-4.Plano R (arriba), plano G (centro) y plano B (abajo) de la imagen de la Figura 3-1.

La solución a este problema pasa por cambiar el espacio de color. Si observamos la imagen original (Figura 3-1), podemos intuir que la diferencia principal entre la forma humana y el fondo es la saturación. Por ello, podemos migrar la imagen desde el espacio de color RGB hasta el espacio de color HSV [16], para quedarnos con su segundo plano (S, *Saturation*). MATLAB realiza esta conversión mediante la función `rgb2hsv`:

```

I=imread('rgb1.jpg');
I=double(I)/255;      %Se normaliza el valor de I para que esté entre 0 y 1
HSV=rgb2hsv(I);
figure,imshow (HSV(:, :,2));
  
```

El resultado de la última línea de código, correspondiente a la representación del plano S de la imagen, se muestra en la Figura 3-5<sup>3</sup>. Como puede observarse, estamos ante el plano idóneo sobre el que trabajar para

<sup>3</sup>La Figura 3-5 aparece ampliada respecto a su tamaño original (80x60 píxeles) porque se considera importante observarla al detalle (como se referenciará más adelante), a pesar de que ello suponga una pérdida de calidad evidente. Esto mismo ocurrirá con otras figuras en lo sucesivo.

discriminar el fondo de la imagen.



Figura 3-5.Plano S de la imagen.

El siguiente es aplicar la umbralización. Mediante la función interna *im2bw* se elegirá un valor de referencia que marque el límite entre píxeles blancos y píxeles negros. Observando la Figura 3-5 nos damos cuenta de que no es necesario un valor alto para generar la máscara binaria, ya que el fondo es totalmente negro y la figura humana tiene una luminosidad bastante alta en general. Se elegirá por ello el valor 0.3 como umbral (parámetro *level* de *im2bw*). A continuación se muestra el código empleado, así como la máscara binaria resultante (Figura 3-6):

```
mascara=im2bw(HSV(:,:,2), 0.3);  
figure, imshow (mascara);
```



Figura 3-6.Máscara binaria tras umbralización con level=0.3.

Ya puede aplicarse esta máscara binaria a la imagen en falso color. Concretamente, se filtrará cada uno de los tres planos de la imagen en falso color (que está en espacio RGB). Este proceso es sencillo en MATLAB, puesto que se puede realizar mediante un barrido de la matriz (se recorre cada píxel de la máscara y, si su valor es 0, también lo será el mismo píxel correspondiente a cada plano de la imagen; en caso contrario, el píxel de cada plano se queda como estaba). El código necesario se muestra a continuación:

```

[filas, columnas]=size (mascara);      %Número de filas y columnas (píxeles) de la imagen
lr=l(:, :, 1);      %Sacamos cada plano
lg=l(:, :, 2);
lb=l(:, :, 3);
lr2=reshape(lr, 1, filas*columnas);      %Cambiamos el tamaño de los planos para el barrido
lg2=reshape(lg, 1, filas*columnas);
lb2=reshape(lb, 1, filas*columnas);
mascararshp=reshape(mascara, 1, filas*columnas);

%Barrido de la imagen
for(j=1:1:filas*columnas)
if(mascararshp(j)==0)
lr2(j)=0;lg2(j)=0;lb2(j)=0;
end
end

%Reordenado de tamaños y almacenamiento en "Ifinal"
lr3=reshape(lr2, filas, columnas);
lg3=reshape(lg2, filas, columnas);
lb3=reshape(lb2, filas, columnas);
Ifinal(:, :, 1)=lr3;
Ifinal(:, :, 2)=lg3;
Ifinal(:, :, 3)=lb3;

```

Esto supone la solución más evidente para el programador genérico. Sin embargo, MATLAB permite conseguir el mismo objetivo usando mucho menos código y variables, ahorrando recursos de memoria y brindando una solución más elegante, gracias a su naturaleza basada en el trabajo con matrices. Basta con una simple multiplicación de matrices (al tratarse de máscaras binarias, si multiplicamos el valor de cada uno de sus píxeles (0 ó 1) por el píxel equivalente en cada plano de la imagen a filtrar, el resultado puede ser cero, o el mismo valor que tenía antes). El código necesario sería el siguiente:

```

Ifinal(:, :, 1)=l(:, :, 1).* mascara;
Ifinal(:, :, 2)=l(:, :, 2).* mascara;
Ifinal(:, :, 3)=l(:, :, 3).* mascara;

```

Por supuesto, la matriz *Ifinal* es la misma en ambos casos, representada en la Figura 3-7.

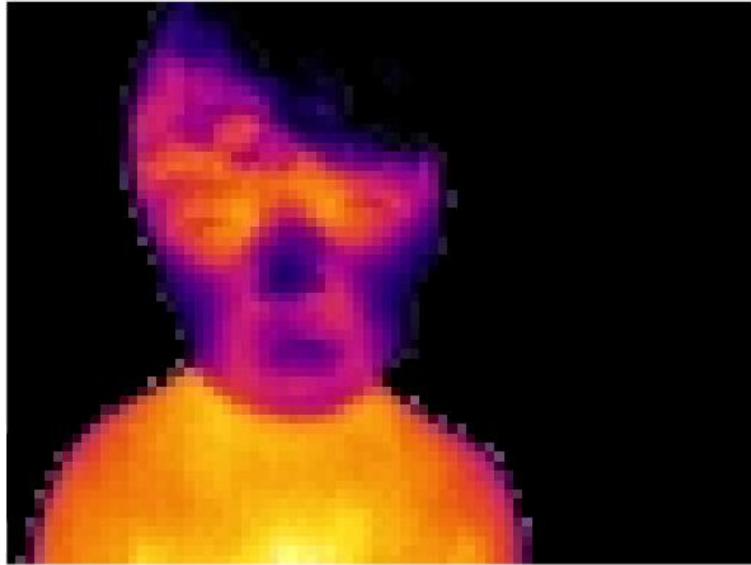


Figura 3-7. Imagen resultante tras aplicar máscara con  $level=0.3$ .

Si se compara con la imagen original (Figura 3-1), se observa que el objetivo de colocar un fondo negro está cumplido. Además, no existe pérdida alguna de información útil, es decir, de la parte que desprende calor. Sin embargo, al ampliar la imagen como se ha hecho en la Figura 3-7 se observa un molesto pixelado en tonos grises sobre la silueta humana, a modo de aura. Este efecto apenas sería apreciable en una imagen de varios Megapixels, pero en las capturas tomadas por la cámara IR (recordemos que su tamaño es de  $80 \times 60$ , es decir, 4800 píxeles) cada píxel importa, puesto que su tamaño relativo es considerablemente alto. Hay que tener en cuenta también que sólo se está mostrando una imagen de ejemplo, pero en otras capturas estudiadas el efecto es aún más nocivo. Por todo ello, esto constituye una aberración intolerable para un Proyecto de esta magnitud.

Puesto que con el método de umbralización sólo contamos con un parámetro, *level*, resulta evidente que el arreglo del problema pasa por cambiar este nivel de umbralización. Con el fin de amenizar la lectura y evitar las iteraciones del inevitable proceso de ensayo y error, se van a mostrar los resultados obtenidos si se utiliza un *level* bastante alto, por ejemplo 0.9.



Figura 3-8. Máscara binaria tras umbralización con  $level=0.9$ .



Figura 3-9. Imagen resultante tras aplicar máscara con  $level=0.9$ .

Aunque es evidente que el *level* seleccionado es demasiado alto, puede observarse al comparar la Figura 3-7 y la Figura 3-9 resulta que las pixelaciones grises se han acabado. La Figura 3-8 deja ver claramente que las líneas que conforman la silueta humana son más suaves que en el caso con  $level=0.3$ . Sin embargo, el gran problema de subir el umbral es la aparición de “islas negras” que eliminan parte del contenido útil para estudiar, lo cual es totalmente intolerable. Por ello, resulta obligatorio buscar una solución de consenso, la cual será tratada en la siguiente subsección.

### 3.1.3 Técnicas de umbralización con relleno de huecos

Ya hemos visto cómo la umbralización supone un importante acercamiento al resultado deseado, pero no está carente de defectos. Si se establece un nivel de umbralización bajo, aparecen molestos artefactos; si se establece un nivel de umbralización alto, parte del cuerpo humano también será filtrado como si fuera el fondo de la imagen, dando lugar a huecos negros que eliminan parte de la información útil.

La solución, como se adelantó antes, pasa por el consenso. Estableciendo un nivel de umbralización intermedio (se elige  $level=0.6$ ), se eliminan por completo las molestas pixelaciones de las transiciones cuerpo-fondo. Un gran número de imágenes examinadas incluso se encuentran exentas de huecos negros... pero no todas. En otro número no menos importante de imágenes aparecen islas negras, eso sí, mucho menos extensas y numerosas que con niveles de umbralización altos. Para erradicar este tipo de huecos se puede usar *imfill*.

Esta función interna de MATLAB rellena los agujeros existentes en imágenes binarias. Según la sintaxis elegida, lo hará automáticamente o según ciertos criterios. Para la solución elegida se aplicará *imfill* automático, ya que permite eliminar casi todos (o todos) los posibles huecos que tenga la máscara binaria de umbral 0.6:

```
bw=im2bw(HSV(:,:,2), 0.6);
bwfill=imfill(bw, 'holes');
```

La Figura 3-10 muestra la máscara con  $level=0.6$ . En ella se observan algunos huecos pequeños pero de difícil localización (rodeados de un halo rojo para facilitar su visualización), así como un hueco de mayor envergadura situado en la parte inferior (rodeado de azul). La Figura 3-11 muestra la misma máscara, una vez tratada con *imfill*. Puede observarse que los pequeños huecos han desaparecido, quedando en este ejemplo sólo un hueco por rellenar.

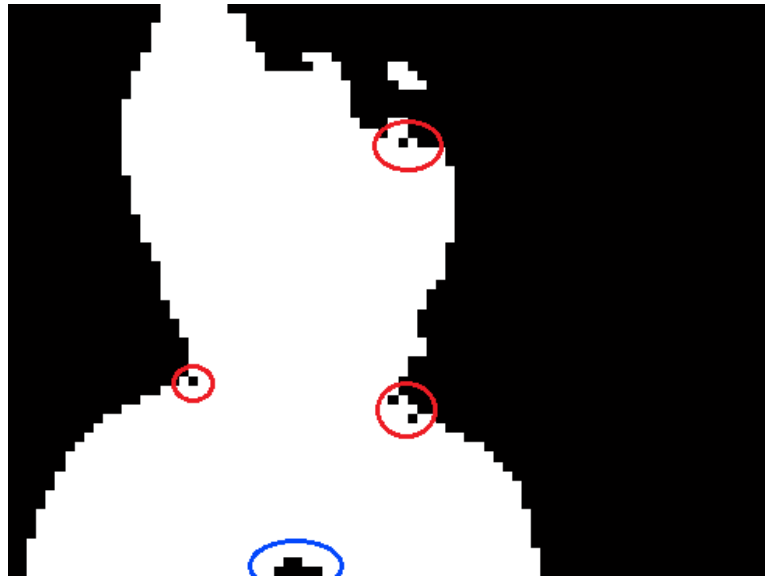


Figura 3-10. Máscara binaria tras umbralización con level=0.6.

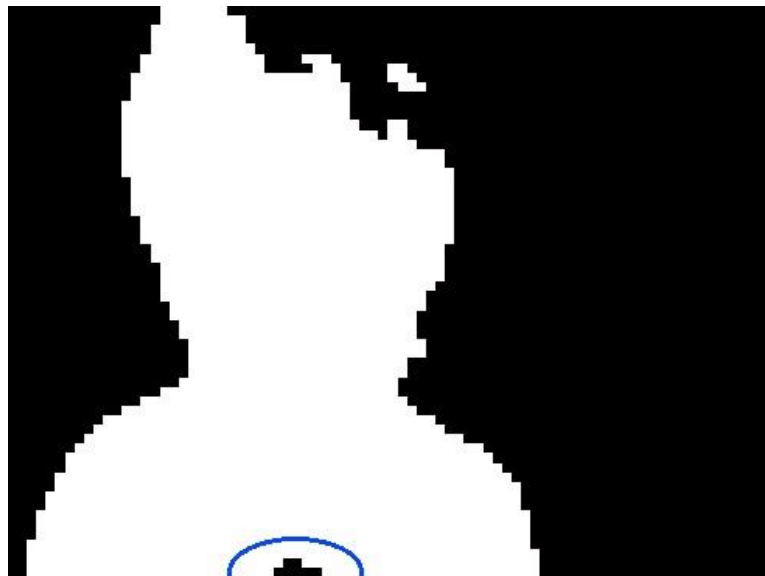


Figura 3-11. Máscara binaria tras umbralización con level=0.6 e imfill.

Una imagen libre de pixelaciones y, en el peor de los casos, sólo con algún hueco por cubrir se puede considerar un estado bastante próximo al ideal. Obviamente esta tara no puede ser pasada por alto, por lo que se recurre a la solución definitiva: Una segunda iteración de *imfill* en la que es el usuario quien, en caso de que quede algún hueco, lo seleccione a través de una interfaz user-friendly. El código necesario es bastante sencillo:

```
mascara=imfill (bwfill);
```

Respecto a los pasos que tiene que seguir el usuario, simplemente cuando ejecute la Herramienta única y completa de este Trabajo Fin de Grado *tfgtorelli.m*, le aparecerá en pantalla la máscara ya umbralizada y con el primer pase de *imfill*. Junto a ella, la imagen en falso color, que servirá de referencia, además de un tutorial con el siguiente texto:

#### *TUTORIAL #1*

*Con el objetivo de conseguir un área útil de estudio lo más fideligna posible,*

*haga click sobre los huecos que pudieran faltar para rellenar dicho área.*

*Puede valerse de la imagen original ("Figure 3") como referencia.*

*Para finalizar, haga doble click sobre el último hueco encontrado, o presione Enter.*

*En caso de que no encuentre ningún hueco a rellenar, presione Enter.*

Dicho tutorial, así como el resto de elementos descritos en el párrafo anterior, pueden ser vistos en la Figura 3-12. Se trata de una captura de pantalla de la ventana de comandos de MATLAB, así como de las figuras emergentes que aparecen en ese punto de la ejecución de la Herramienta.

Cuando el usuario seleccione los huecos en el caso de que los haya (Figura 3-13), la máscara estará completa. En nuestro caso, existe el claro hueco hacia el centro de la parte inferior de la imagen. La Figura 3-14 muestra el resultado definitivo.



Figura 3-12. Captura de MATLAB en el momento de rellenar los posibles huecos restantes.



Figura 3-13. Hueco clickeado con detalle de asterisco cian y morado.





Figura 3-14. Máscara definitiva.

De esta forma, se ha obtenido una máscara que permite añadir un fondo negro a la imagen. Y lo hace con una capacidad de filtrar todo el contenido susceptible de estudio altamente satisfactoria, quedando la imagen además libre de pixelaciones y otras aberraciones. No existen valores ajustables por el usuario que entorpezcan el estudio mediante el ensayo y error in situ. Únicamente en los casos que surja algún pequeño hueco, y si el usuario lo ve conveniente, podrá pulirlo mediante un sencillo y rápido gesto gracias a la cómoda interfaz gráfica.

Para dar veracidad a las anunciadas bondades de esta sub-herramienta de eliminación de fondo, englobada en la Herramienta completa de estudio de lesiones vasculares mediante el tratamiento de imágenes, se expondrán algunos ejemplos más en las siguientes Figuras. Estos ejemplos se mostrarán en el tamaño original de la imagen, para que sirva de vista general; y en un tamaño más ampliado (el obtenido al maximizar la ventana emergente de MATLAB) para que el lector pueda observarlos al detalle. Cabe recalcar de nuevo que la parte “térmica” de las imágenes queda intacta, mientras que el resto de la imagen queda en negro. A lo largo de los próximos puntos de este Documento el lector podrá comprobar la importancia del filtrado en las imágenes para cumplir los objetivos de este Proyecto.

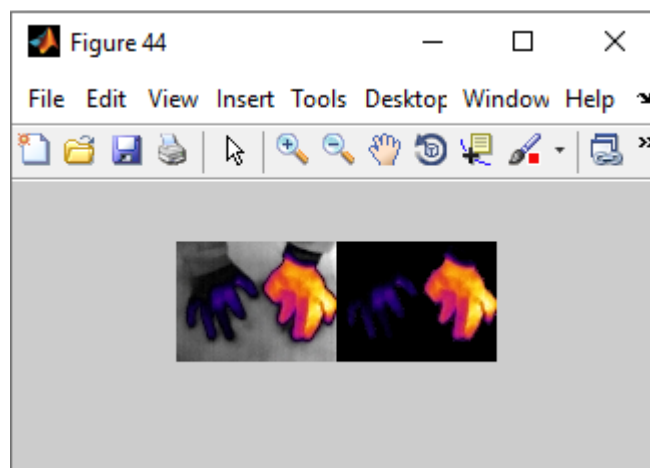


Figura 3-15 a. Ejemplo 2 a tamaño real. Imagen original (izquierda) e imagen procesada (derecha).

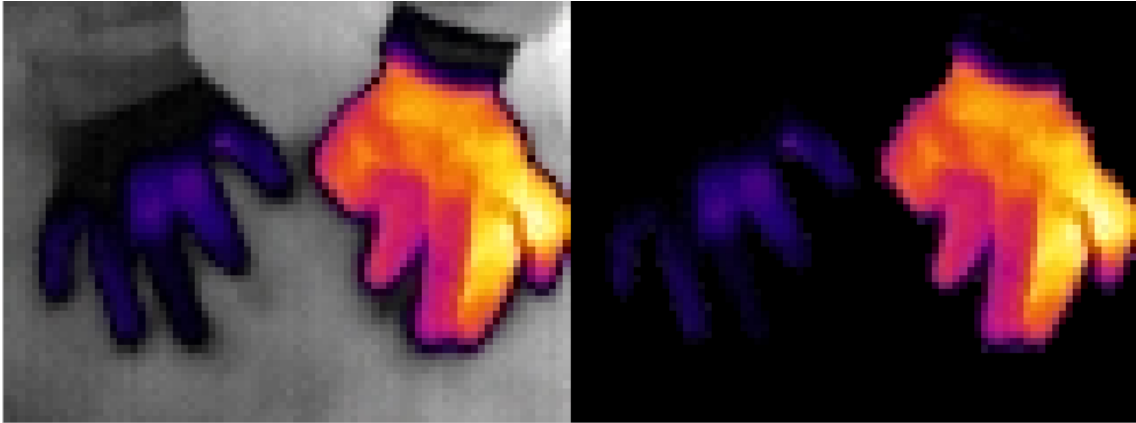


Figura 3-15 b.Ejemplo 2 ampliado para observar en detalle.

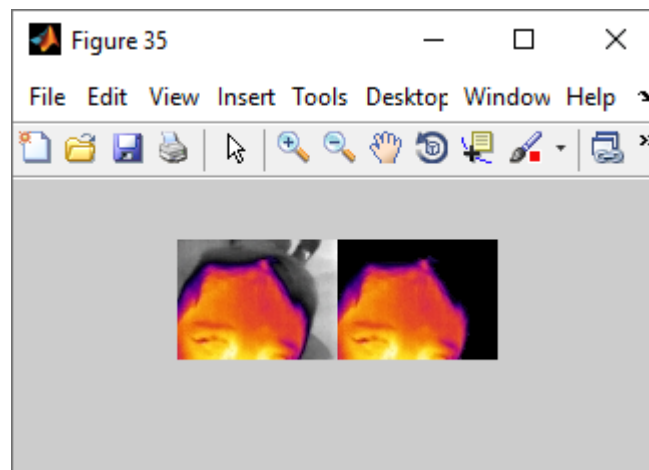


Figura 3-16 a.Ejemplo 3 a tamaño real. Imagen original (izquierda) e imagen procesada (derecha).

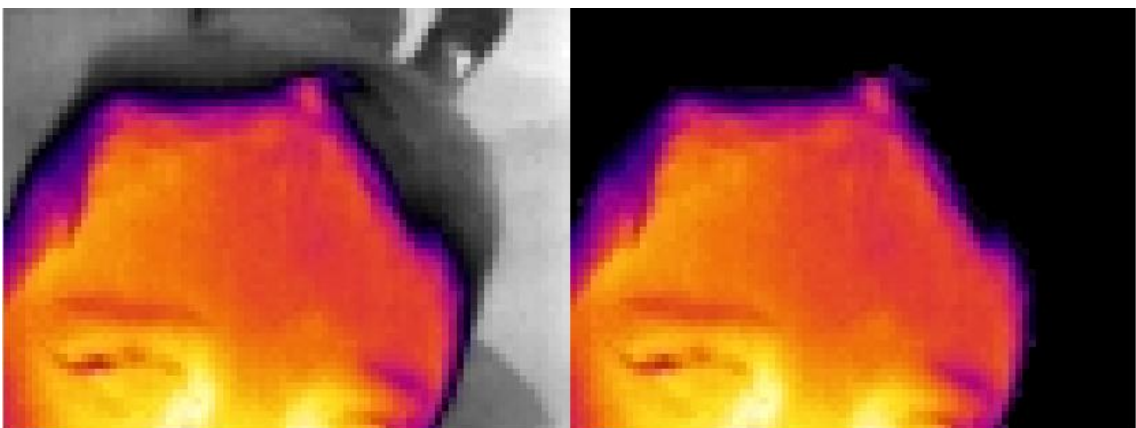


Figura 3-16 b.Ejemplo 3 ampliado para observar en detalle.

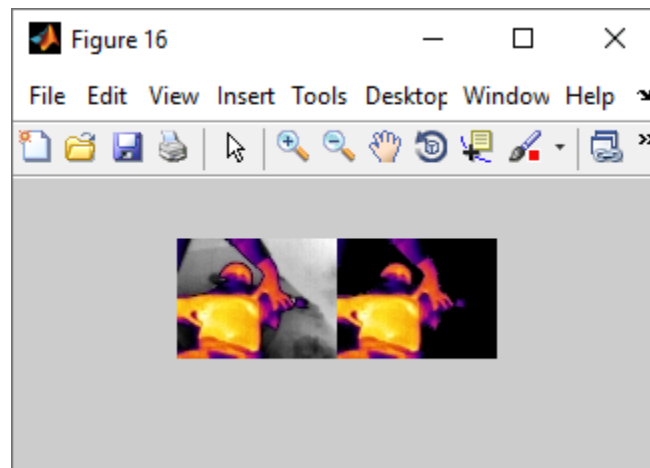


Figura 3-17 a. Ejemplo 4 a tamaño real. Imagen original (izquierda) e imagen procesada (derecha).

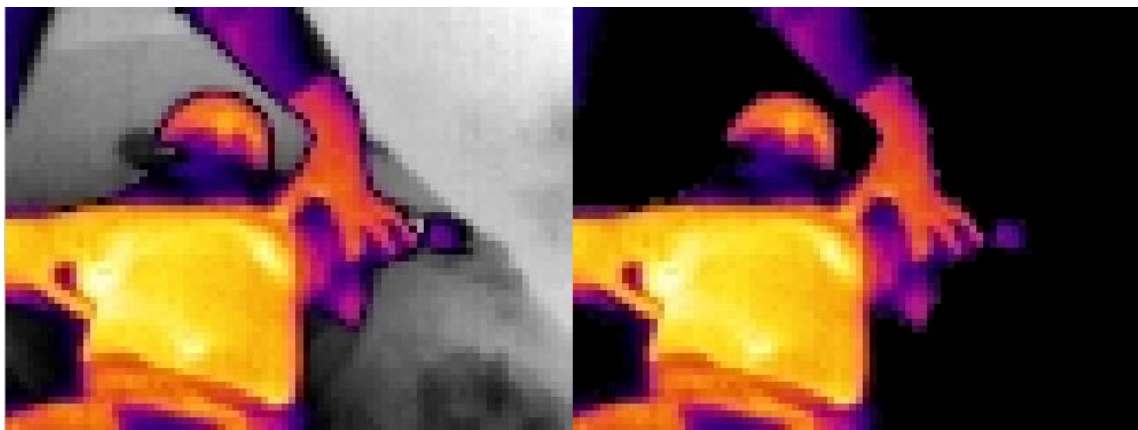


Figura 3-17 b. Ejemplo 4 ampliado para observar en detalle.

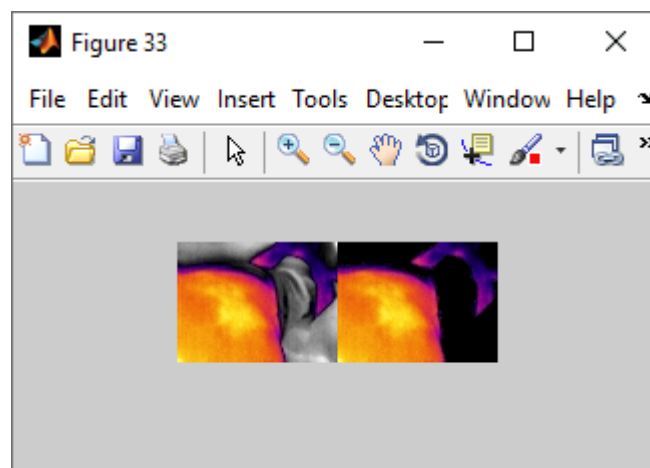


Figura 3-18 a. Ejemplo 5 a tamaño real. Imagen original (izquierda) e imagen procesada (derecha).

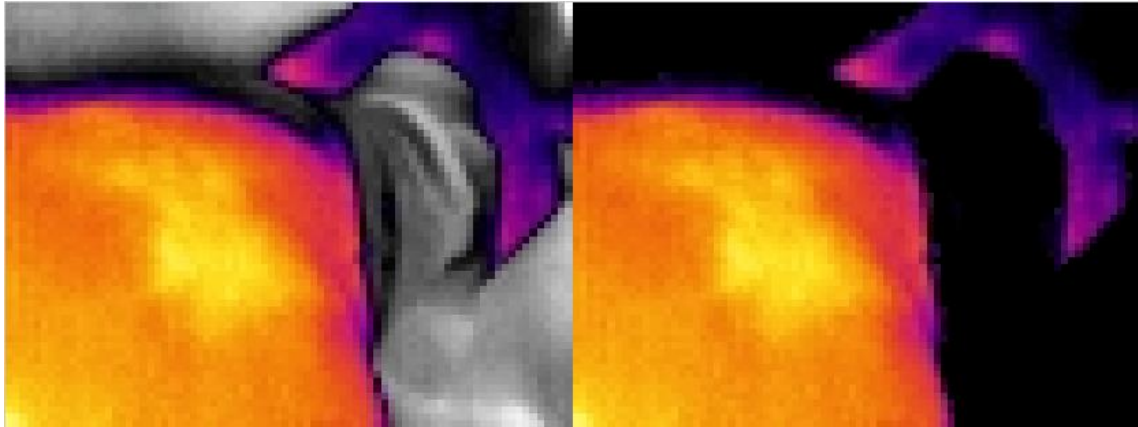


Figura 3-18 b. Ejemplo 5 ampliado para observar en detalle.

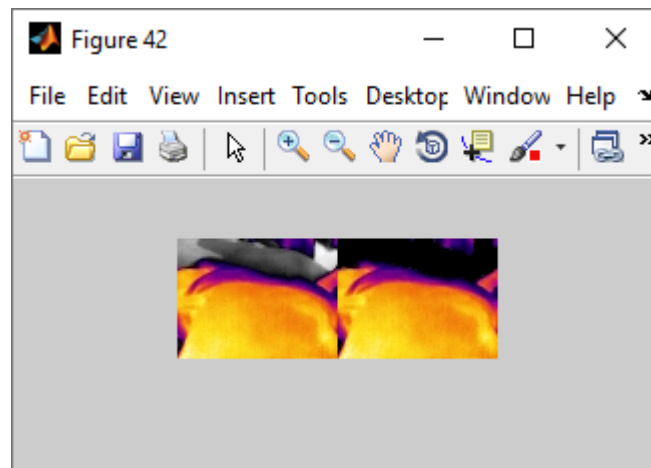


Figura 3-19 a. Ejemplo 6 a tamaño real. Imagen original (izquierda) e imagen procesada (derecha).

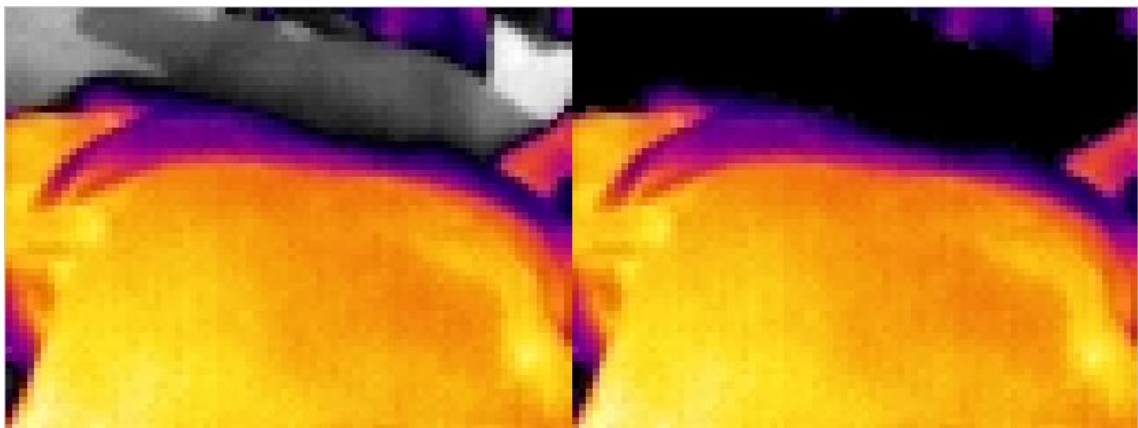


Figura 3-19 b. Ejemplo 6 ampliado para observar en detalle.

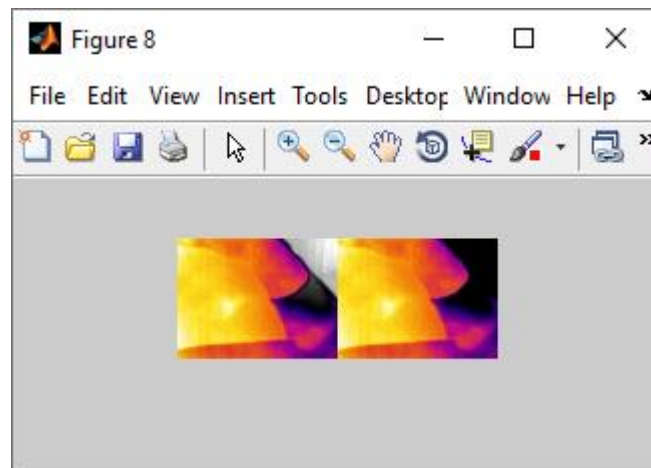


Figura 3-20 a. Ejemplo 7 a tamaño real. Imagen original (izquierda) e imagen procesada (derecha).

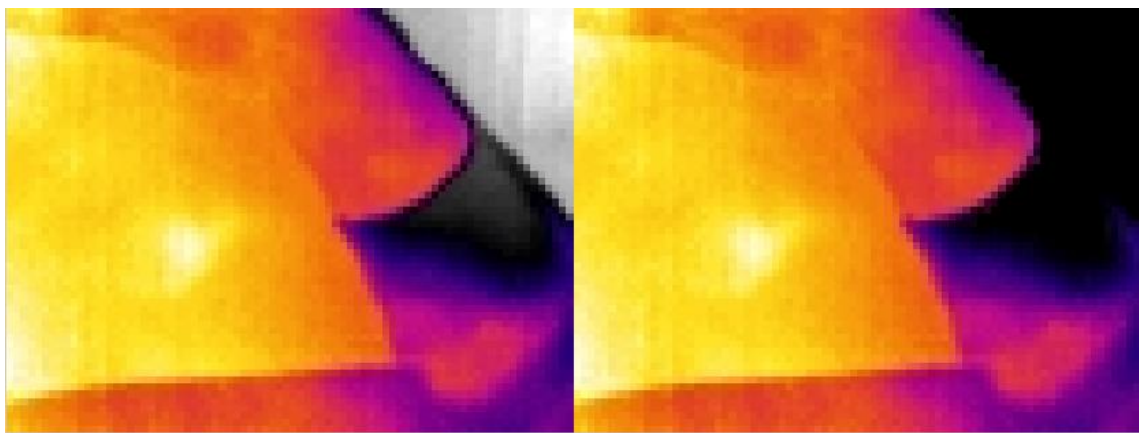


Figura 3-20 b. Ejemplo 7 ampliado para observar en detalle.

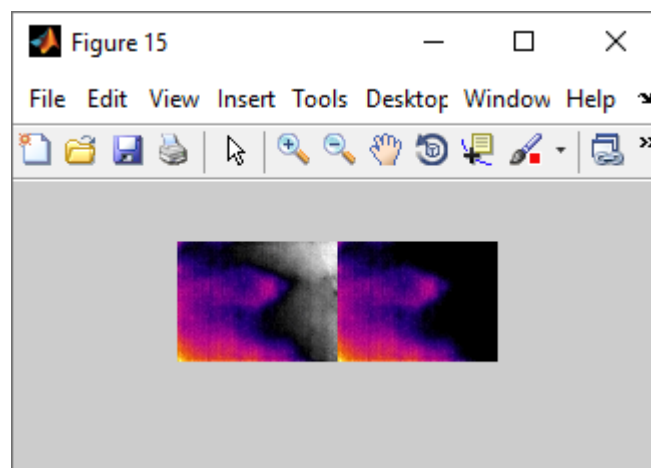


Figura 3-21 a. Ejemplo 8 a tamaño real. Imagen original (izquierda) e imagen procesada (derecha).

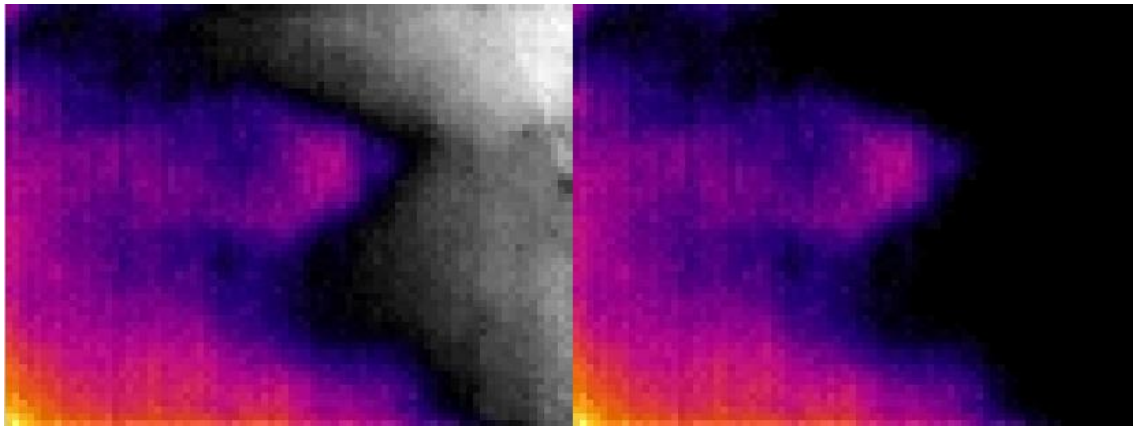


Figura 3-21 b. Ejemplo 8 ampliado para observar en detalle.

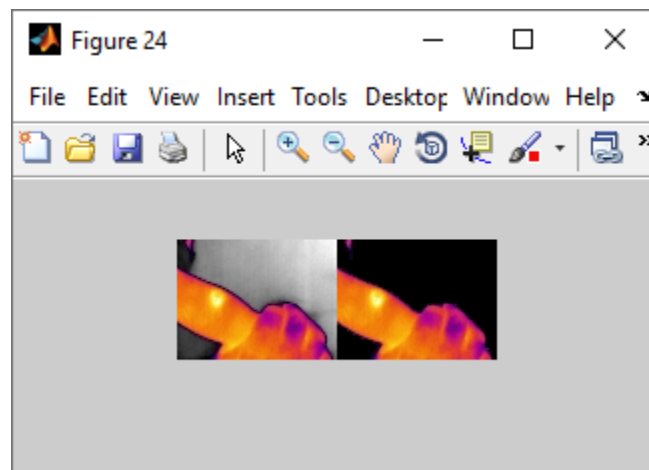


Figura 3-22 a. Ejemplo 9 a tamaño real. Imagen original (izquierda) e imagen procesada (derecha).

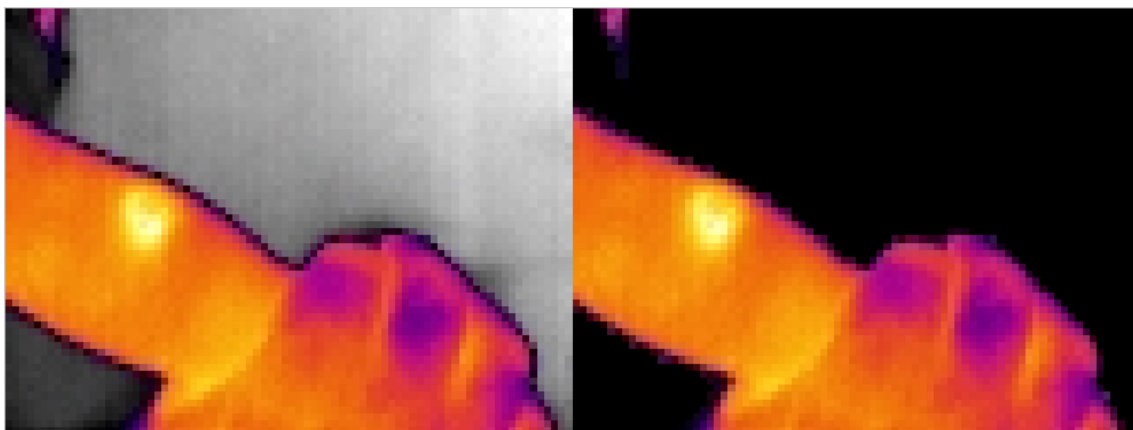


Figura 3-22 b. Ejemplo 9 ampliado para observar en detalle.

### 3.2 Normalización y tratamiento previo de la imagen

Se trata de una pequeña aunque importante parte del código centrada en normalizar la imagen a estudiar, así como en prepararla y almacenarla para los tratamientos que le restan. Los dos pasos principales son:

1. Normalización de la imagen obtenida en *read\_bin\_thermal\_image*: Una de las tareas acometidas por la función tratada en la Subsección 2.2.1 *read\_bin\_thermal\_image* era la de extraer la imagen del

archivo .BIN y almacenarla como una variable de MATLAB para su posterior tratamiento. En esta parte del código se normaliza su valor para que cada pixel esté entre 0 y 1. Esta práctica se repetirá de forma muy común durante los tratamientos restantes del el Proyecto.

2. Aplicar la mascara obtenida a la imagen original: En la Sección 3.1 se obtuvo la máscara que permitía eliminar el fondo de la imagen y quedarnos solamente con la parte que contiene información relevante. Durante el transcurso de la sección, y con el objetivo de que el contenido visual resultara más accesible al lector y al usuario final de la Herramienta, se aplicó dicha máscara a la imagen en falso color RGB. Pero resulta evidente pensar que esa misma máscara se puede aplicar a la imagen original en escala de grises. De hecho, la verdadera utilidad de la máscara aparece cuando se aplica a la imagen original en escala de grises extraída por *read\_bin\_thermal\_image*. Sobre esta imagen se realizarán los estudios de detección de hemangiomas u otras lesiones. El código empleado para ello es muy sencillo:

```
IM2=IM .* mascara;      %IM es la imagen original almacenada como variable MATLAB
```

### 3.3 Mapa de probabilidades de detección de lesiones vasculares

La sección que nos ocupa es la que brinda los resultados más importantes a este Trabajo de Fin de Grado. Gracias a la información contenida en la imagen (no siempre visible), y a través de un adecuado tratamiento digital de la misma, se pueden rescatar datos muy útiles y mostrar estudios muy interesantes sobre el tema que nos ocupa. Estos procedimientos, que serán expuestos en las próximas líneas, han permitido desarrollar los dos tipos de mapa de probabilidades de detección de lesiones vasculares empleados en este Proyecto: Los generales y los relativos.

#### 3.3.1 Mapa de probabilidades generalizado

Se trata de un mapa de probabilidades único para cada imagen. Como se ha ido explicando a lo largo del Documento, la base teórica sobre la que se sustenta este Proyecto es la relación entre la aparición de lesiones vasculares y el aumento local de la velocidad de circulación de la sangre. Y como ese incremento de velocidad conlleva una mayor temperatura en la zona, la relación se puede reformular de una manera muy primitiva como “lesiones-temperatura”. Por supuesto, es un concepto que hay que tratar con suma delicadeza, ya que la correspondencia no es, ni mucho menos, sencilla.

Antes de entrar en detalle, es necesario recalcar que las condiciones aplicadas en la toma de imágenes por la cámara IR no son para nada homogéneas. La falta de uniformidad en criterios como distancia y ángulo de disparo, temperatura de la sala o iluminación, entre otros, imposibilita prácticamente que haya dos imágenes iguales. Incluso si hablamos de un mismo paciente y una misma sesión de fotos, es muy difícil encontrar dos imágenes con el mismo nivel de brillo en la misma zona (típicamente una mancha o lesión vascular). Este hecho dificulta aún más la búsqueda de unos niveles universales que determinen dónde hay lesiones y dónde no las hay.

Por ello, la solución pasa por realizar un tratamiento común a todas las imágenes a estudiar, con el objetivo de homogeneizar en lo posible sus características. No se puede realizar un tratamiento manual e individualizado a cada imagen por parte del programador en busca de los valores adecuados, ya que supondría una solución inviable en cuanto a coste temporal, computacional y humano.

El primer gran punto a tratar será el de conseguir resaltar más las manchas potenciales (es decir, las zonas localmente calientes) respecto al resto del cuerpo. Se trata de una condición indispensable para poder presentar al usuario un mapa correctamente delimitado. Una manera muy común de conseguir este efecto en una imagen es modificar su contraste. Un mayor contraste aumenta las diferencias entre el color negro y el blanco mediante la reasignación de los valores de la intensidad de la imagen [17].

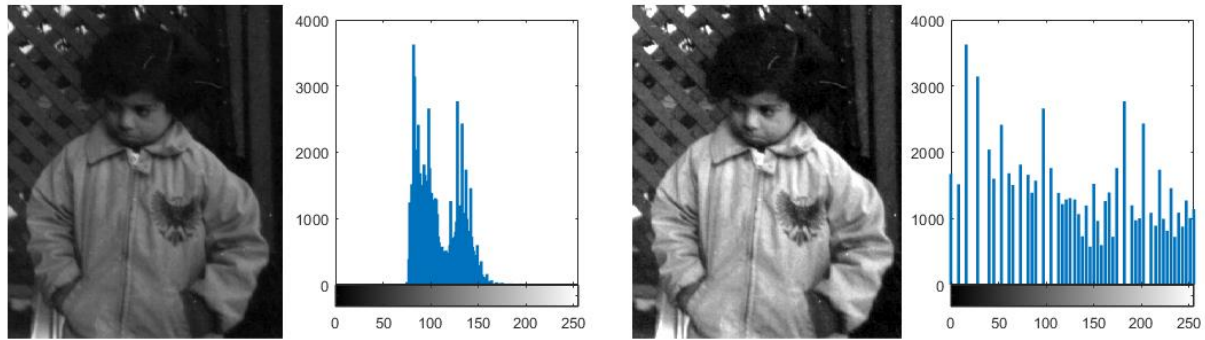


Figura 3-23. Aumento de contraste en imagen, junto a variación de histograma. Recuperada de [www.mathworks.com](http://www.mathworks.com)

MATLAB dispone de casi una veintena de funciones para ajustar el contraste. Aunque pueden ser aplicadas a imágenes en color, son funciones que trabajan fundamentalmente con imágenes en escala de grises. Tras estudiarlas y probar sus resultados en una gran variedad de imágenes termográficas, se ha decidido optar por la combinación de dos funciones: *imadjust* e *imsharpen*.

La función *imadjust* es la más usada en cuanto a modificación de la intensidad. La sintaxis es:

```
J = imadjust(I,[low_in; high_in],[low_out; high_out],gamma)
```

Siendo *low\_in* y *high\_in* los valores que se mapean a *low\_out* y *high\_out*. Es decir, los valores inferiores a *low\_in* pasarán a tener valor *low\_out*; y los valores superiores a *high\_in* pasarán a tener valor *high\_out*. El parámetro *gamma* modifica la forma de la curva de la intensidad. Si su valor es mayor que uno, pondera hacia valores de salida más bajos; si es menor que uno, pondera hacia valores de salida más altos; si es igual a uno, la curva no será modificada.

El tratamiento *imadjust* que se asigna en la Herramienta es:

```
g1=imadjust (IM2, [0 1], [0 1], 2.5);
g1=imadjust (g1);
```

En la primera línea de este código se observa que sólo se cambia la curva de intensidad para oscurecer más salida, sin modificar los valores máximos y mínimos. En la segunda línea se hace un tratamiento generalizado, que consiste simplemente en saturar el 1% inferior y el 1% superior a todos los píxeles de la imagen, aumentando el contraste de la imagen.

Para ilustrar este paso y los que se mostrarán a continuación, se utiliza el ejemplo de la Figura 3-15. Se trata de un paciente cuyo diagnóstico advierte de malformación arteriovenosa en la mano izquierda (derecha de la imagen). Se observa una mayor temperatura en esa mano y, dentro de ésta, también se aprecia que hay dedos más afectados por las lesiones (meñique, anular, ligeramente en menor medida el pulgar). Estas características hacen que sea un buen ejemplo para explicar los resultados.

Así, el efecto de aplicar *imadjust* a la imagen se observa en la Figura 3-24. Puede verse claramente que la imagen tratada (derecha de la Figura) consigue un mayor contraste entre las zonas oscuras y las claras, distinguiéndose así mucho mejor entre unas y otras.



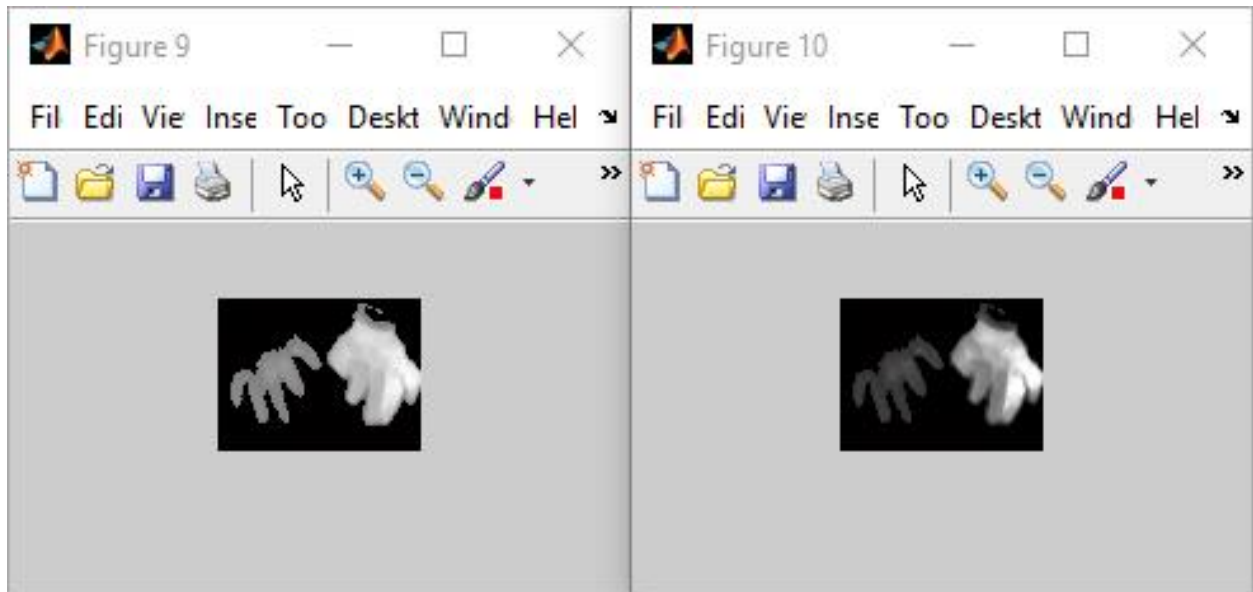


Figura 3-24.Resultado de aplicar imadjust.

Una vez obtenida esta imagen, el siguiente paso es afilarla (también conocido como aumentar la “nitidez”, o directamente el “sharp” o “sharpness”) para poder delimitar las zonas de una manera más eficiente. Es un paso fundamental para establecer un mapa que estime dónde hay probabilidad alta de encontrar lesiones vasculares, dónde hay probabilidad media y dónde hay probabilidad baja. En la imagen actual, los límites entre unas y otras son confusos.

MATLAB permite llevar a cabo esta tarea mediante la función `imsharpen`, que afila la imagen usando enmascaramiento desafilado. Su sintaxis es:

```
B = imsharpen(A,Name,Value,...);
```

Los pares Nombre-Valor son parámetros que monitorizan diferentes aspectos del enmascarado. El tratamiento `imsharpen` aplicado a nuestras imágenes será:

```
g1=imsharpen (g1,'Radius',10,'Amount',1);
```

Los parámetros usados son *Radius* (desviación estándar del filtro paso de baja gaussiano) y *Amount* (fuerza del afilado). El valor alto de *Radius* ayuda a resaltar posibles lesiones que se manifiestan a baja temperatura (casos muy repetidos e igualmente importantes, como veremos más adelante). La fuerza *Amount* tiene un pequeño incremento, ya que su valor por defecto es 0.8.

El resultado final de estos tratamientos sobre la imagen de ejemplo se muestra en la Figura 3-25.

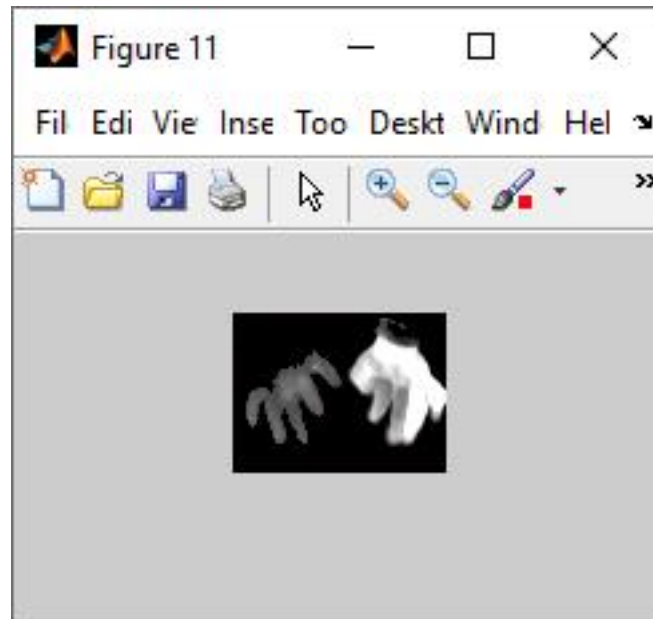


Figura 3-25. Resultado de aplicar *Imadjust* e *Imsharpen*.

Esta imagen ya permite una clara diferenciación entre niveles de temperatura o probabilidad de encontrar lesión. Lo más interesante de este tratamiento es que homogeneiza los resultados de las distintas imágenes. Es decir, consigue que todas las imágenes, a pesar de que tienen condiciones de capturas excesivamente diferentes, se muevan en los mismos niveles de intensidad y puedan ser evaluadas, por tanto, bajo los mismos criterios de discriminación entre niveles. Supone una solución eficiente al que es sin duda el gran problema del análisis de estas imágenes.

El siguiente gran punto en el desarrollo de la Herramienta será el de aplicar los mapas de color a la imagen. Dado el tratamiento que se ha realizado a la imagen y la necesidad de encontrar una solución válida para todos los casos estudiados y los futuros, se eligen tres niveles estándar que delimitarán las zonas de probabilidad alta, media y baja. Cada píxel de la imagen que pertenezca a una de estas áreas será pintado de un color, según los tres niveles de probabilidad. Se sobreentiende que la parte que quede sin colorear será la que tenga una probabilidad de detección de lesión vascular aún menor que “baja”.

Los niveles aplicados a las imágenes homogeneizadas son 0.945, 0.8 y 0.7, presentados como variables *level1*, *level2* y *level3*, respectivamente. Llama la atención que el valor *level1* se ha establecido como un 5% superior a la referencia lógica de 0.9. Debido a las características de la imagen una vez tratada, el valor que marcará las temperaturas más altas será muy elevado. Se puede decir que ese 5% suaviza la transición entre las zonas de temperatura alta y las de temperatura media:

```
level1=0.945;
level2=0.8;
level3=0.7;
```

Para extraer la mancha principal se realizará la función *im2bw*, que ya fue presentada en la subsección 3.1.2. El nivel a aplicar será *level1*. Es decir, se creará una máscara en la que todo nivel de píxel mayor que *level1* pasará a ser de color blanco puro (valor 1), mientras que los valores inferiores inferiores pasarán a ser negro puro (valor 0).

```
g2=im2bw(g1, level1);
```

La máscara de nuestro ejemplo se muestra en la Figura 3-26.

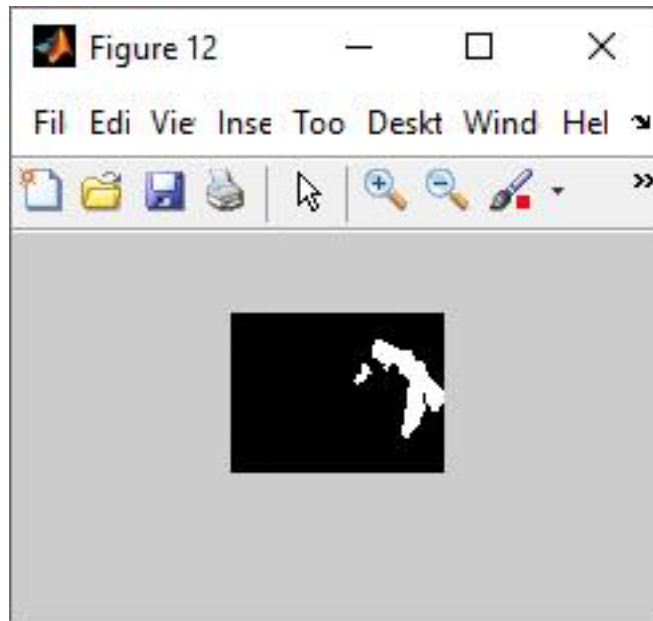


Figura 3-26. Máscara de extracción de zona de máxima probabilidad.

A continuación, hay que colorear esa mancha principal o zona de máxima probabilidad y aplicarla a la imagen original. El color que identifica típicamente a las altas temperaturas es el rojo, que en el espacio RGB se consigue con los valores 1, 0, 0 (respectivamente). Dadas las características de la máscara (valores 0 ó 1) y de la imagen original (valores entre 0 y 1), si creamos una nueva imagen RGB cuyo contenido en cada plano sea la imagen original en escala de grises más (en el plano R) o menos (en los planos G y B) los valores de dicha máscara, se obtendrá como resultado la imagen en escala de grises con la mancha principal en rojo. Por supuesto, y para que cada plano de la nueva imagen siga acotado entre 0 y 1, habrá que normalizar la imagen. Este proceso se muestra en el siguiente código y los resultados se pueden ver en la Figura 3-27:

```
%Colorear la zona de probabilidad máxima en rojo.
```

```
IM3(:, :, 1) = IM2 + g2;
```

```
IM3(:, :, 2) = IM2 - g2;
```

```
IM3(:, :, 3) = IM2 - g2;
```

```
%Limitar IM3 a valores [0,1]
```

```
IM3(IM3 < 0) = 0;
```

```
IM3(IM3 > 1) = 1;
```

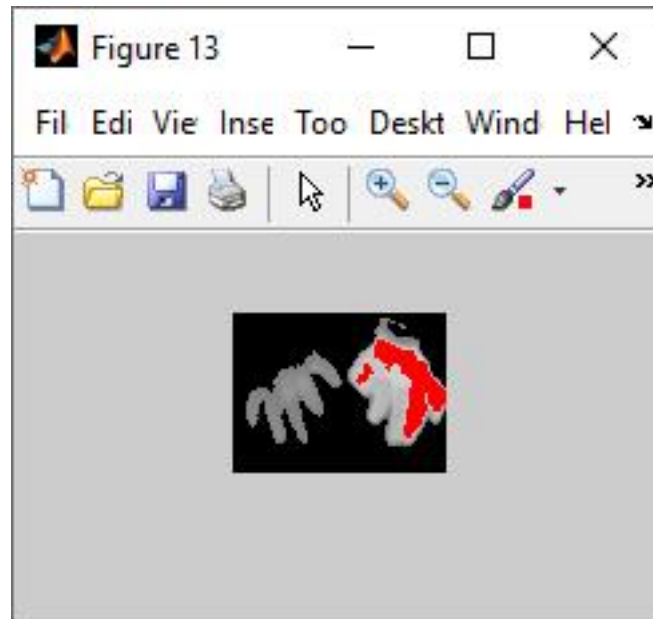


Figura 3-27. Imagen original con zona de máxima probabilidad en rojo.

El siguiente paso es extraer la mancha secundaria o zona de probabilidad media. Es un proceso similar al de las manchas principales, ya que se sigue usando la función *im2bw* con el nivel fijado a *level2*. Sin embargo, en este caso habrá que tener en cuenta que la función transforma en 1 a todo píxel superior a 0.8, por lo cual la nueva máscara se “pisa” con la máscara principal (recordemos que ponía a 1 todo píxel superior a 0.945). Es decir, la nueva máscara necesita un filtro paso de alta. Una vez más, la naturaleza optimizada para matrices que ofrece MATLAB evita que esto sea un problema grave, ya que sólo hay que restar a la máscara recién obtenida (Figura 3-28) la máscara principal que se halló antes (Figura 3-26). Esto hace las veces de un sencillo paso de alta. El código necesario es:

```
g3=im2bw(g1, level2);
g4=g3-g2;      %Recordemos que g2=im2bw(g1, level1);
```

La máscara final se muestra en la Figura 3-29.

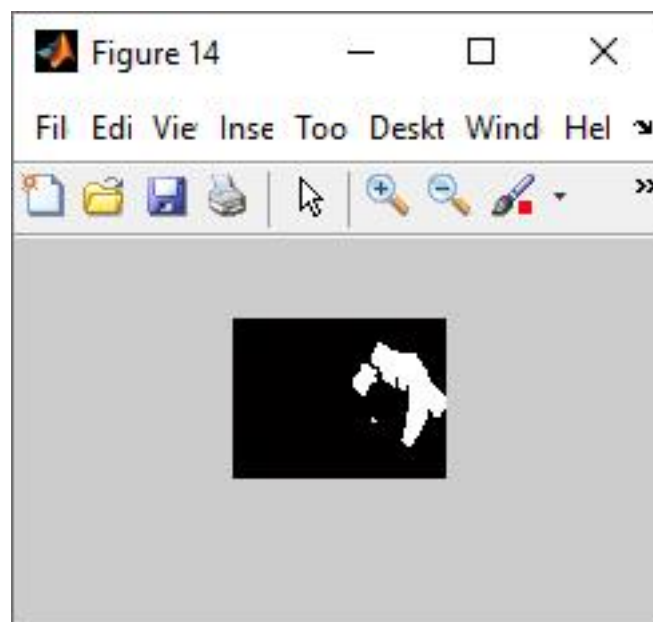


Figura 3-28. Máscara de extracción de zona de probabilidad media, sin filtrar.

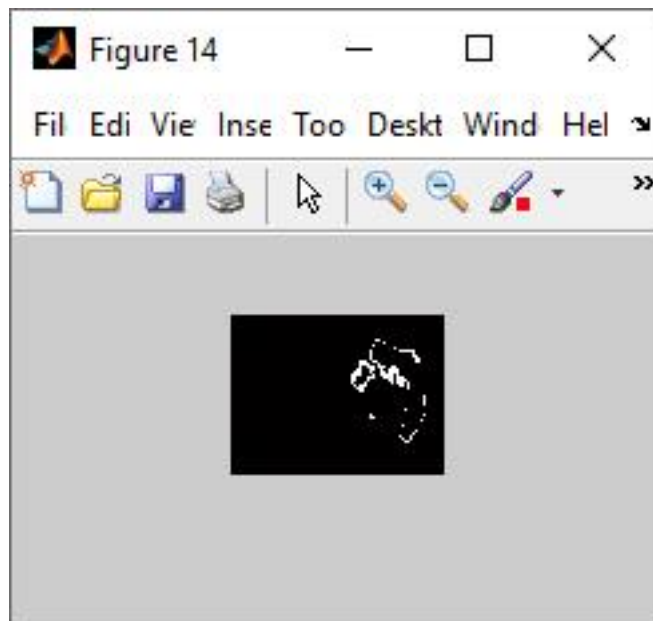


Figura 3-29. Máscara de extracción de zona de probabilidad media, filtrada.

Respecto al color al que aplicar a la zona media, lo más intuitivo es elegir el amarillo (1, 1, 0 en RGB), y aplicarlo de la misma forma en que se hizo con la zona de máxima probabilidad. A continuación se muestra el código usado para aplicar el color y normalizar:

```
%Colorear la zona de probabilidad media en amarillo.
```

```
IM3(:,:,1)=IM3(:,:,1)+g4;
```

```
IM3(:,:,2)=IM3(:,:,2)+g4;
```

```
IM3(:,:,3)=IM3(:,:,3)-g4;
```

```
%Limitar IM3 a valores [0,1]
```

```
IM3(IM3<0)=0;
```

```
IM3(IM3>1)=1;
```

En la Figura 3-30 se muestra el resultado hasta este punto.

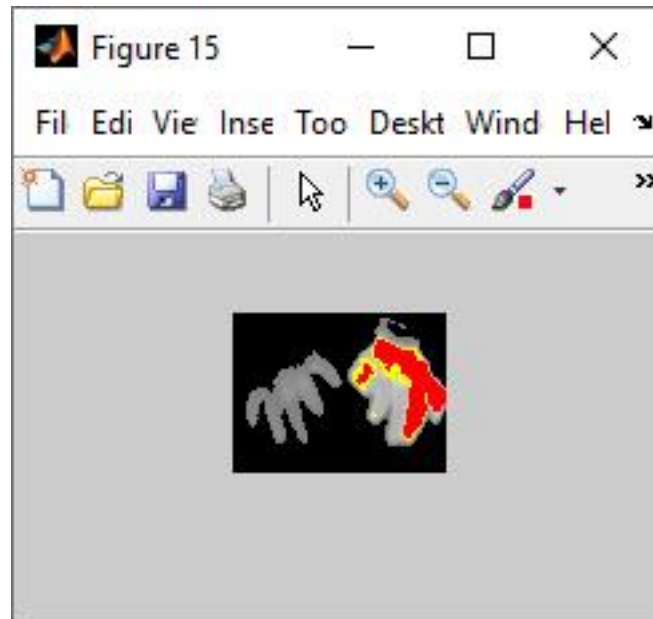


Figura 3-30. Imagen original con zona de máxima probabilidad en rojo y zona de probabilidad media en amarillo.

El último paso es extraer una zona de probabilidad baja. Se procederá de la misma forma que se hizo para la zona de probabilidad media, es decir, aplicando un filtro paso de alta a la máscara obtenida con *im2bw* y nivel marcado por *level3*. Este filtrado se realizará restando la máscara secundaria sin filtrar (la que transformaba en 1 a todo píxel superior a 0.8) a la máscara recién obtenida.

Por otro lado, se le aplicará un color frío, cian (0, 1, 1 en RGB) y se normalizará de nuevo. Las máscaras sin y con filtrado aparecen en la Figura 3-31 y en la Figura 3-32, respectivamente. El código usado para la zona de probabilidad baja se muestra a continuación:

```
g5=im2bw(g1,level3);
g6=g5-g3;

%Colorear la zona de probabilidad baja en cian.
IM3(:,:,1)=IM3(:,:,1)- g6;
IM3(:,:,2)=IM3(:,:,2)+ g6;
IM3(:,:,3)=IM3(:,:,3)+ g6;

%Limitar IM3 a valores [0,1]
IM3(IM3<0)=0;
IM3(IM3>1)=1;
```

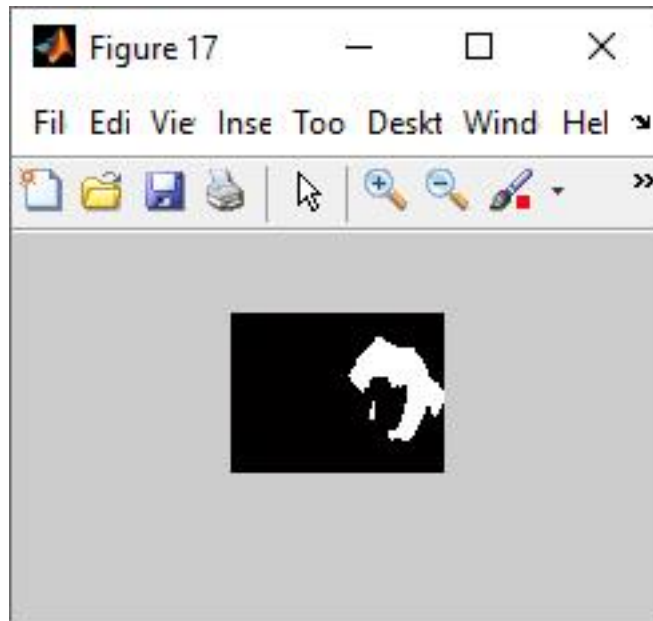


Figura 3-31. Máscara de extracción de zona de probabilidad baja, sin filtrar.

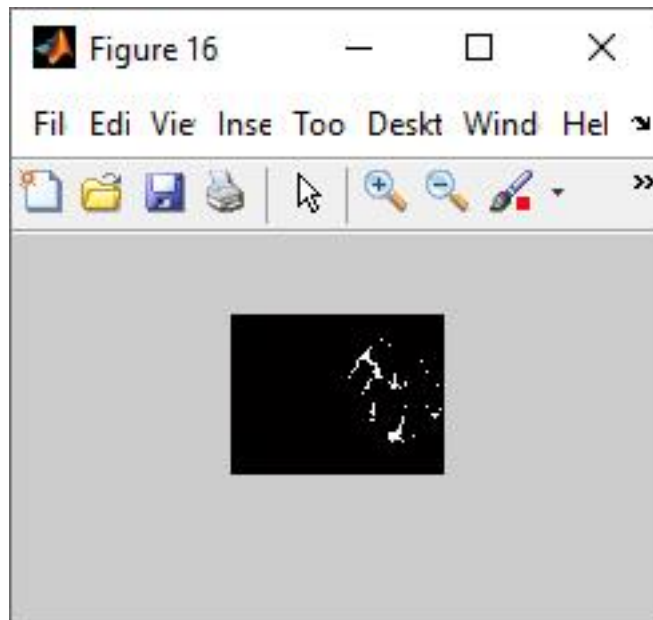


Figura 3-32. Máscara de extracción de zona de probabilidad baja, filtrada.

Finalmente, se muestra la imagen de nuestro ejemplo, una vez aplicado el mapa de probabilidades generalizado (Figura 3-33). Puede observarse que se ajusta al diagnóstico médico (éste y otros resultados serán expuestos con mayor detalle en el Capítulo 4).

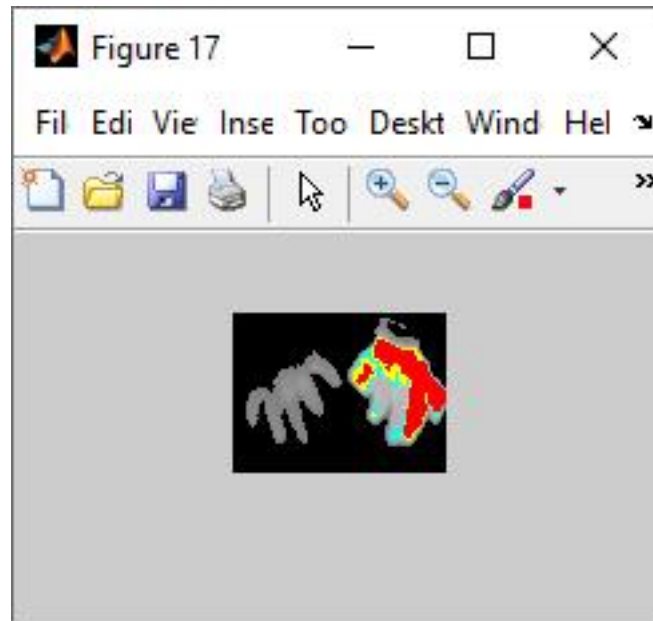


Figura 3-33. Mapa de probabilidades generalizado.

### 3.3.2 Mapa de probabilidades relativo usando Pixel Select

Una vez cumplido el principal objetivo fijado en este Trabajo Fin de Grado (creación de un mapa de probabilidades de detección de hemangiomas y otras enfermedades vasculares según la velocidad de la sangre, presentado en pantalla sobre la imagen original), cabe preguntarse si no puede darse una vuelta de tuerca más, añadir algún método que afine los resultados y ofrezca un mayor abanico de soluciones al usuario final. La propuesta para resolver tal pregunta es la implantación de mapas de probabilidad inteligentes.

Se trata de una segunda iteración que realiza la Herramienta, una vez que el mapa general de probabilidades ha sido presentado en pantalla. Para dar comienzo a la realización de este segundo mapa, se imprime por pantalla la imagen original sin mapas de color agregados. De esta forma, el usuario dispone del tiempo que estime necesario para comparar la imagen original, la imagen mapeada y el propio diagnóstico.

Si el usuario estima conveniente realizar un nuevo estudio centrado en una lesión (o posible/futurible lesión) en particular, sólo tendrá que clicar en la imagen la zona del cuerpo en la que se encuentra dicho objeto de segundo estudio. A esta técnica se le ha llamado Pixel Select. La Herramienta se encarga entonces de realizar un nuevo estudio relativo a la zona seleccionada. La Figura 3-34 muestra lo que aparece en las ventanas de MATLAB en ese momento. Además de la imagen mapeada y la imagen “limpia” que hemos citado antes, se despliega un nuevo tutorial que explica el funcionamiento de esta parte de la Herramienta. El mensaje es el siguiente:

*TUTORIAL #2*

*Para realizar un estudio más exhaustivo sobre un nivel de temperatura determinado,  
haga doble click sobre un punto de la imagen que pertenezca a dicho nivel.*

El código empleado para realizar la función descrita es:

```
[columna, fila, levels] = impixel(IM2original);
level=levels(1);
```

Para seleccionar el píxel de manera interactiva, se ha utilizado la función *impixel* de MATLAB, que devuelve las coordenadas  $x$  e  $y$  del píxel seleccionado, así como el nivel de brillo del píxel seleccionado en cada plano



RGB, almacenado en el vector *levels*. Cabe destacar que, dado que se trata de una imagen de un solo plano, los tres valores del vector *levels* serán exactamente iguales. Para trabajar con mayor comodidad, se almacena cualquiera de ellos en la variable *level*, que será de gran importancia para el resto del desarrollo.

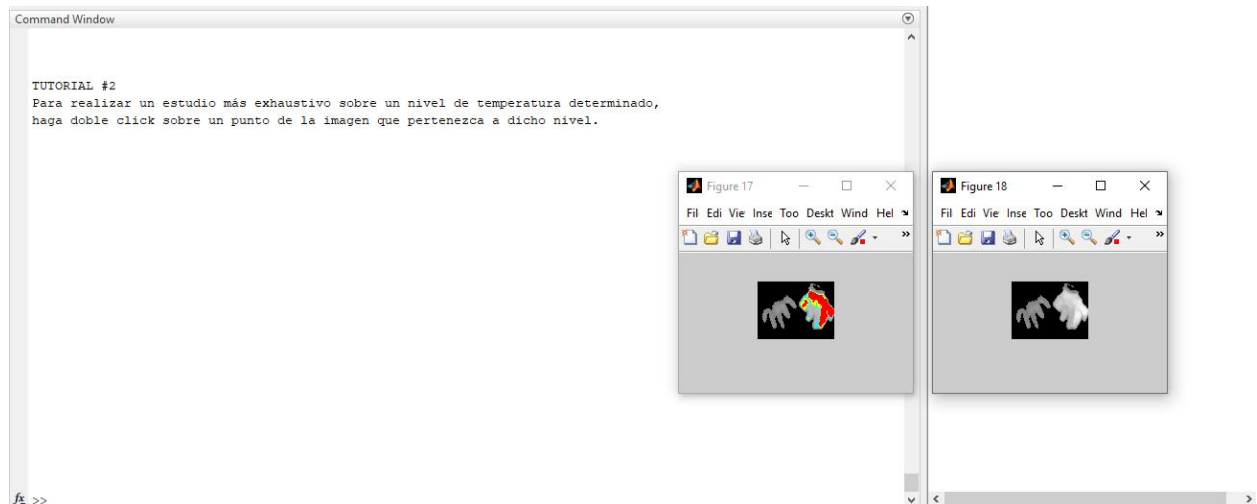


Figura 3-34. Captura de MATLAB en el momento de seleccionar la zona a estudiar (en la ventana de nombre “Figura 18”).

Hagamos un zoom en la ventana de nombre “Figura 18” que puede verse en la Figura 3-34, para que el lector de este Documento tenga una vista más aproximada a la que tiene el usuario final de la Herramienta:

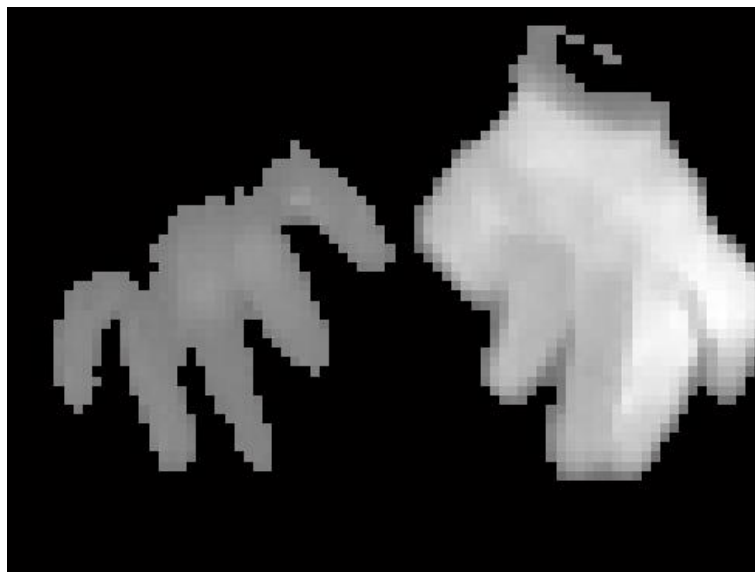


Figura 3-35. Zoom a la ventana “Figura 18” de la Figura 3-34.

Si observamos con detenimiento, se puede apreciar que los dedos meñique y anular de la mano izquierda (derecha de la imagen) tienen una temperatura aún mayor que el resto de la mano. Se va a usar este pretexto para elaborar el ejemplo que ilustra estas líneas: Hacer un estudio más exhaustivo de la zona con mayor temperatura en la imagen.

Lo primero es interactuar con la ventana de MATLAB para seleccionar la parte de la imagen a estudiar. La Figura 3-36 ilustra dónde se ha hecho doble click para el ejemplo que nos ocupa (marcado en azul). El nivel de brillo del píxel seleccionado quedará guardado en la variable *level*, como se dijo anteriormente.

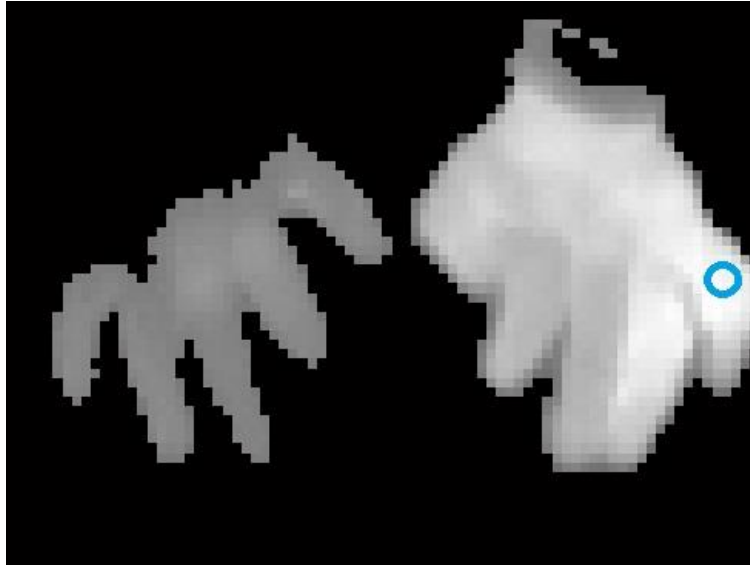


Figura 3-36. Selección de la zona a estudiar.

El siguiente paso es realizar un tratamiento a la imagen para conseguir los resultados deseados. Este tratamiento es independiente del que se realizó para obtener el mapa de probabilidades generalizado, ya que ahora se realiza bajo otras condiciones y con la presencia de la variable *level*.

El tratamiento comienza ajustando el contraste y añadiendo nitidez para optimizar la localización de manchas potenciales, de manera similar a como se hizo para obtener el mapa de probabilidades generalizado (Subsección 3.3.1):

```
g1=imadjust (IM2, [0 1], [0 1], 2.5);
g1=imadjust (g1);
```

Sólo en el caso de que el nivel del punto/mancha (variable *level*) sea menor que 0.9 se hará uso de *imsharpen*. Esta decisión es tomada porque aplicar *imsharpen* en niveles mayores que 0.9 supone una granularización del contraste demasiado gruesa para el tipo de mapa que se quiere realizar ahora:

```
if (level<0.9)
    g1=imsharpen (g1,'Radius',5,'Amount',2);
end
```

Por supuesto, habrá que normalizar la matriz tratada *g1* y almacenar en la variable *level* el nuevo valor que tiene el píxel seleccionado tras el tratamiento. Para ello, nos ayudamos de sus coordenadas *x* e *y*, que fueron obtenidas como resultado de la función *impixel* y almacenadas en las variables *columna* y *fila*, respectivamente.

El siguiente paso es extraer las manchas principales, es decir, las zonas del cuerpo humano que tienen un nivel de temperatura más similar al que tiene el píxel seleccionado. Es un procedimiento similar al empleado en la Subsección 3.3.1, pero la horquilla de temperaturas que abarca la zona se traza de manera muy diferente.

En este caso, la zona de probabilidad máxima abarcará desde un valor inferior en un 3% al del píxel seleccionado, hasta un valor que se encuentre aproximadamente a medio camino entre el nivel de brillo del píxel seleccionado y 1. Dado que, después del tratamiento, los valores de las manchas en las imágenes suelen ser bastante altos (casi siempre por encima de 0.9), el límite superior establecido será del mismo orden que el límite inferior en estos casos.

Para casos inferiores a 0.9, y teniendo en cuenta una vez más que la imagen tratada tiene los valores altos más altos y los bajos más bajos debido al aumento de contraste y nitidez, se deberán ir relajando (aumentando) los límites de la zona de máxima probabilidad conforme vaya cayendo el valor del píxel seleccionado, para obtener así un resultado más fideligno. Las condiciones establecidas en el párrafo anterior permiten que esto

sea posible. De todas formas, y para desacelerar ligeramente esta caída, se usa un factor 0.45 para valores de *level* inferiores a 0.9.

Resumiendo, se puede decir que entre 0.9 y 0.91 hay un abanico de información mucho más amplio que entre 0.7 y 0.71. A continuación, el código empleado para establecer estas condiciones:

```
ga=im2bw(g1, level-0.03*level);

if (level<0.9)
    levelgb=0.45*( 1-(level-0.03*level))+(level-0.03*level);
else
    levelgb=0.5*( 1-(level-0.03*level))+(level-0.03*level);
end

%Si levelgb se queda por debajo de level->Valor manual(caso extremo)
levelgb(levelgb<level)=level+0.15;

%levelgb nunca mayor que 1
levelgb(levelgb>1)=1;

gb=im2bw(g1,levelgb);
g2=ga-gb;
```

La máscara de extracción de zona de máxima probabilidad relativa al píxel seleccionado se muestra en la Figura 3-37.

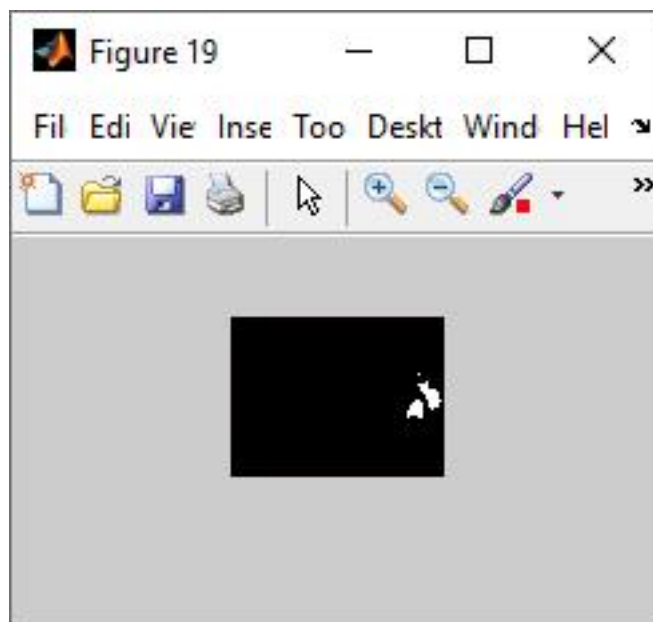


Figura 3-37. Máscara de extracción de Zona de máxima probabilidad relativa al píxel seleccionado.

Tras extraer la máscara principal, el siguiente paso es colorearla y aplicarla a la imagen original. Se trata de un procedimiento muy parecido al estudiado en la Subsección 3.3.1. Para conseguir una mayor familiarización del usuario con la Herramienta, se vuelve a emplear el color rojo para distinguir la zona de probabilidad máxima.

```
%Colorear la zona de máxima probabilidad relativa al píxel seleccionado en rojo.
```

```

IM3(:,:,1)=IM2+g2;
IM3(:,:,2)=IM2-g2;
IM3(:,:,3)=IM2-g2;

%Limitar IM3 a valores [0,1]
IM3(IM3<0)=0;
IM3(IM3>1)=1;

```

Para el ejemplo que ilustra estas líneas, la imagen original con la zona de máxima probabilidad relativa coloreada en rojo se muestra en la Figura 3-38.

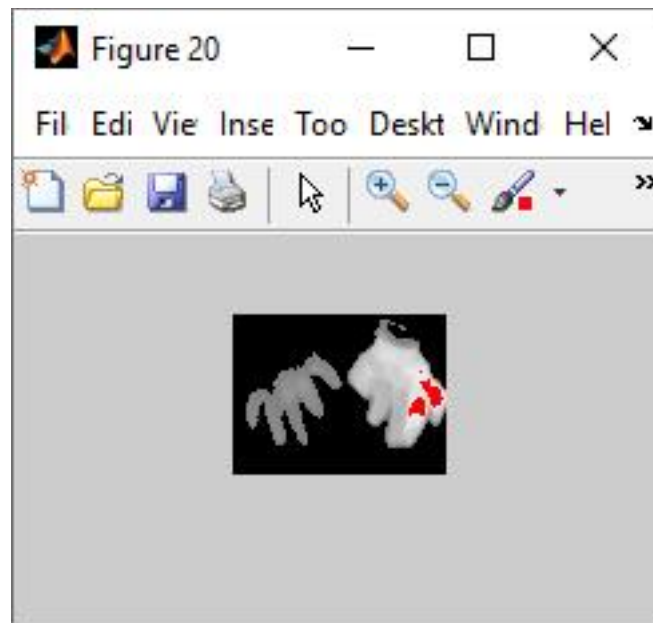


Figura 3-38. Imagen original con zona de máxima probabilidad relativa en rojo.

Una vez extraída, coloreada y aplicada la zona de máxima probabilidad relativa al píxel seleccionado, se podría dar por concluido el segundo mapa. Sin embargo, para que dicho mapa no accuse falta de información respecto a su homónimo mapa generalizado, se incluirá una zona de probabilidad media bajo el límite inferior.

La realización resulta bastante más sencilla que la del área de probabilidad máxima, ya que ahora el límite superior lo marca el comienzo de la zona roja. Se ha establecido un nivel de descenso del 10% respecto a *level* y el código empleado es:

```

g3=im2bw(g1, level-0.1*level);
g4=g3-ga;    %Recordemos que ga=im2bw(g1, level-0.03*level);

```

Para colorear, se vuelve a optar por el color amarillo:

```

%Colorear la zona de media probabilidad relativa al píxel seleccionado en amarillo.
IM3(:,:,1)=IM3(:,:,1)+g4;
IM3(:,:,2)=IM3(:,:,2)+g4;
IM3(:,:,3)=IM3(:,:,3)-g4;

%Limitar IM3 a valores [0,1]
IM3(IM3<0)=0;

```

```
IM3(IM3>1)=1;
```

La máscara obtenida se muestra en la Figura 3-39. La imagen coloreada con las zonas roja y amarilla puede observarse en la Figura 3-40.

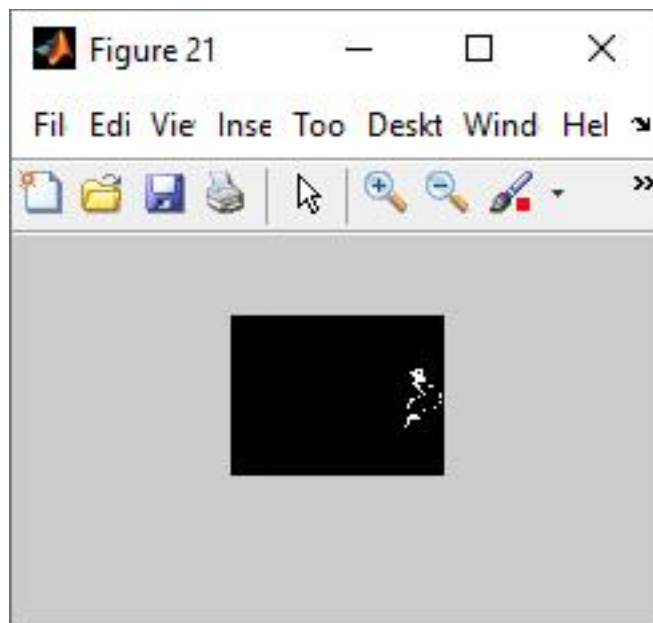


Figura 3-39. Máscara de extracción de Zona de media probabilidad relativa al píxel seleccionado.

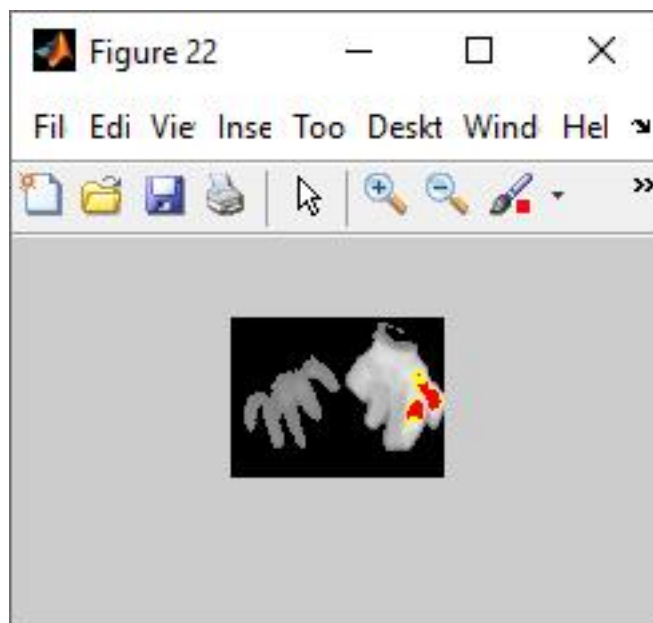


Figura 3-40. Imagen original con zona de máxima probabilidad relativa en rojo y media probabilidad relativa en amarillo.

Finalmente se añadirá una zona de baja probabilidad relativa. En este caso, el límite inferior lo marca un descenso del 20% en *level*. El código empleado para la elaboración de la máscara, coloreado (cian), aplicación sobre la imagen y normalización es:

```
g5=im2bw(g1, level-0.2*level);
g6=g5-g3;
%Colorear la zona de probabilidad baja en cian.
IM3(:,:,1)=IM3(:,:,1)- g6;
```

```

IM3(:,:,2)=IM3(:,:,2)+ g6;
IM3(:,:,3)=IM3(:,:,3)+ g6;

%Limitar IM3 a valores [0,1]
IM3(IM3<0)=0;
IM3(IM3>1)=1;

```

La Figura 3-41 presenta la máscara obtenida, y la Figura 3-42 muestra el resultado final de la imagen de ejemplo con mapeado relativo usando Pixel Select.

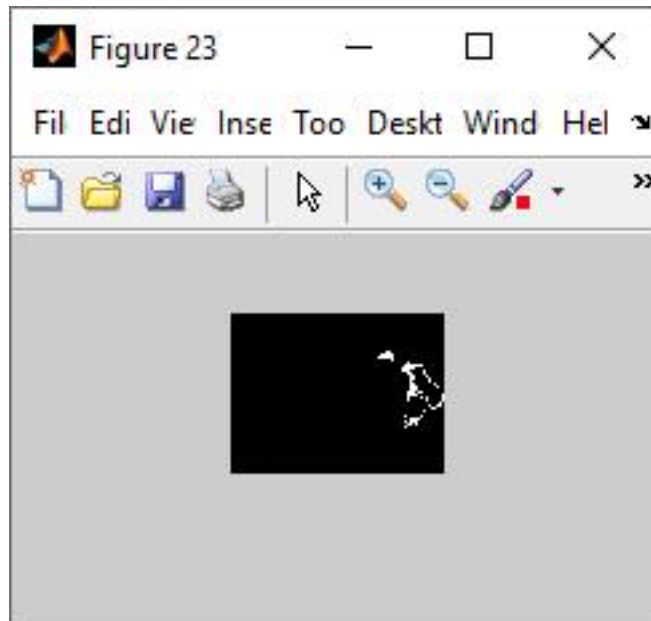


Figura 3-41. Máscara de extracción de Zona de baja probabilidad relativa al píxel seleccionado.

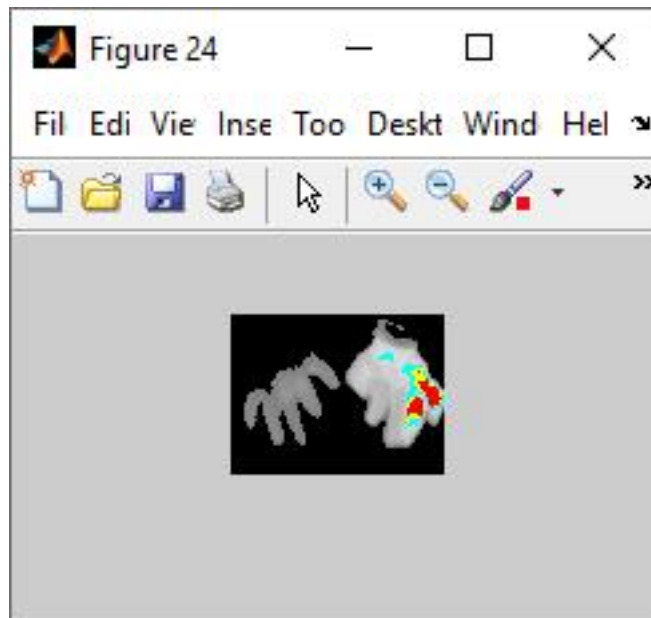


Figura 3-42. Mapa de probabilidades relativo usando Pixel Select.

Aunque los resultados se analizarán en el Capítulo 4, cabe destacar que la imagen mostrada en la Figura 3-42 corresponde con lo que cabría esperar cuando se comenzó el ejemplo: La zona roja indica dónde se pueden encontrar los niveles de temperatura más similares al del punto seleccionado (Figura 3-36). Es decir, la extensión más probable de la lesión seleccionada, así como los lugares del resto del cuerpo donde hay más

probabilidades de encontrar lesiones como la seleccionada. Las zonas amarilla y cian describen lo mismo, pero con probabilidad media y baja, respectivamente.

Como última idea, cabe añadir que en el ejemplo expuesto durante este Capítulo se ha hecho Pixel Select sobre la zona más brillante de la imagen. Esto será lo normal en la mayoría de los casos, puesto que los hemangiomas y otras lesiones suelen tener valores de brillo muy elevados y el usuario clickea sobre píxeles con valores muy cercanos a 1.

Sin embargo, si se decide estudiar una zona que no sea la más brillante de la imagen, puede aparecer un nuevo rango de temperaturas en dicha imagen, sobre el cual no se ha hablado hasta ahora: Las temperaturas muy superiores a las del píxel seleccionado.

Este caso se ilustra en la Figura 3-43. En ella se puede observar una zona circular de color blanco (o gris muy claro), en la parte centro-izquierda de la imagen. Pues bien, ese color blanco no ha sido generado artificialmente por la Herramienta, sino que ya estaba en la imagen original. Simplemente se trata de una zona con temperatura muy superior a la del píxel seleccionado y, por tanto, se considera fuera de los márgenes de probabilidad establecidos por la Herramienta. De ahí que no esté coloreada en rojo, amarillo o cian (probabilidad alta, media y baja, respectivamente).

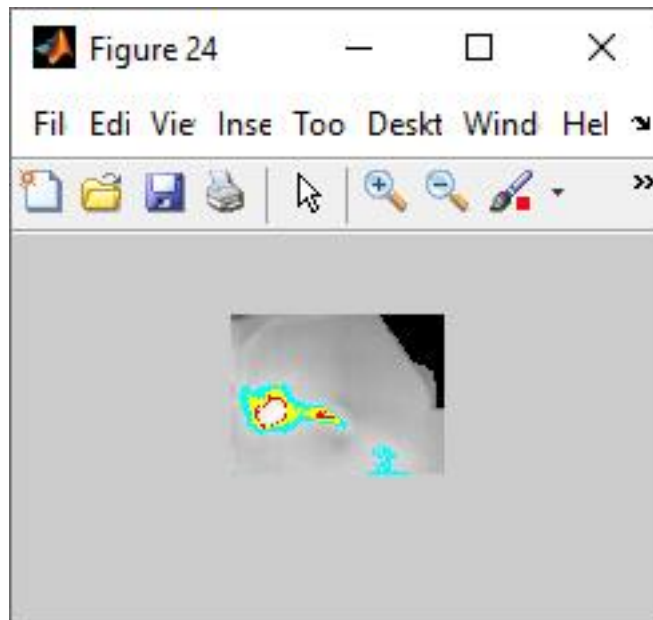


Figura 3-43. Imagen con zona blanca, correspondiente a temperaturas relativas muy elevadas.





# 4 RESULTADOS

---

*En esencia, la ingeniería consiste en usar la ciencia para encontrar soluciones creativas y prácticas. Es una noble profesión.*

*- Isabel II de Inglaterra -*

Conocer el alcance de la Herramienta desarrollada en este Trabajo de Fin de Grado exige analizar su rendimiento. A lo largo de este Capítulo se llevará a cabo esta tarea mediante la exposición, explicación y valoración de los resultados. De entre todos los casos estudiados antes, durante y después de la realización del software, se ha elegido una docena representativa de ellos. Todos ellos han sido proporcionados por la Tutoría de este Trabajo de Fin de Grado, bajo el amparo de profesionales de la medicina.

A través del análisis de los casos expuestos no sólo se discutirán fallos y aciertos, sino que se irá tejiendo la comprensión de la propia idiosincrasia de este Trabajo de Fin de Grado: Los objetivos que se buscan, los inconvenientes más comunes, los diferentes grados de dificultad para localizar lesiones, la gran variedad de casos médicos existentes, las fortalezas y debilidades de la Herramienta, o la importancia de los factores externos.

Antes de entrar de lleno en el análisis de resultados, es preciso dar una serie de directrices básicas sobre el contenido que el lector va a encontrar en las próximas páginas de este Documento, empezando por la manera de presentar la información e imágenes de cada paciente.

La primera información que se obtiene de un paciente es un número que lo identifica inequívocamente respecto al resto. Por supuesto, cualquier referencia al nombre real y cualquier información visual susceptibles de poner en riesgo la privacidad de sus datos personales ha sido convenientemente omitida.

El siguiente contenido relacionado con el paciente es un pequeño diagnóstico médico que advierte del tipo de lesión que padece. Cantidad y tamaño de lesiones, o el hecho de que éstas conlleven cambios de temperatura detectados conforman contenido adicional susceptible de aparecer en este parte médico.

A continuación, las imágenes del paciente. Con el objetivo de mostrar la información visual de manera ordenada y cómoda, las capturas siempre se presentarán con el mismo formato. Se trata de una sola figura que contiene seis imágenes equiespaciadas y del mismo tamaño, en este orden:

- Parte superior izquierda de cada figura: Imagen de referencia, proporcionada por el doctor. Según el paciente, puede tratarse de una imagen térmica en falso color con un círculo rojo que rodea a la lesión, o una imagen no térmica en color, como las que realiza cualquier cámara convencional.
- Parte superior derecha de cada figura: Imagen original capturada por la cámara IR, en escala de grises.
- Parte central izquierda de cada figura: Mapa de probabilidades generalizado, aplicado a la imagen original.

- Parte central derecha de cada figura: Mapa de probabilidades relativo usando Pixel Select, aplicado a la imagen original
- Parte inferior izquierda de cada figura: Mapa de probabilidades generalizado, acompañado de una serie de símbolos que valoran el grado de concordancia entre el diagnóstico y el mapa de probabilidades generalizado.
- Parte inferior derecha de cada figura: Mapa de probabilidades relativo, acompañado de una serie de símbolos que valoran el grado de concordancia entre el diagnóstico y el mapa de probabilidades relativo.

Como se ha indicado en los dos últimos puntos, y con el objetivo (para nada sencillo) de hacer algo menos abstracta la valoración de los resultados, cada mapa en cada paciente irá acompañado de una serie de símbolos, cuyo significado se detalla en la Figura 4.1.



**CONCORDANCIA ALTA:** Cuando el contenido del mapa coincide en colocación, color y forma con lo que se espera en ese punto.



**CONCORDANCIA MEDIA:** Cuando el contenido del mapa coincide en dos de tres entre colocación, color y forma, o coincide en las tres de manera no muy exacta con lo que se espera en ese punto.



**CONCORDANCIA BAJA:** Cuando el contenido del mapa coincide en uno de tres entre colocación, color y forma, o coincide en las tres de manera muy inexacta con lo que se espera en ese punto.



**DISCORDANCIA BAJA:** Cuando el contenido del mapa no coincide con el esperado en ese punto, pero el desajuste es muy leve en color, forma y tamaño, o está justificado.



**DISCORDANCIA MEDIA:** Cuando el contenido del mapa no coincide con el esperado en ese punto, pero el desajuste es leve a moderado en color, forma y tamaño, o está justificado que exista, pero no su grado.



**DISCORDANCIA ALTA:** Cuando el contenido del mapa no coincide con el esperado en ese punto, con gran desajuste en color, forma y tamaño, y/o no está justificado.

Figura 4-1.Leyenda Capítulo 4.

Por último, el análisis de los resultados. Se trata de desglosar la información anterior, comentar aspectos importantes de la misma, valorar y comparar resultados y sacar pequeñas conclusiones de cada caso.

Los pacientes irán en páginas diferentes para facilitar al lector su búsqueda o los saltos entre unos y otros.

- Paciente 1

*Diagnóstico médico:* Malformación arteriovenosa en la mano izquierda. Mayor temperatura en la mano que tiene la lesión y, dentro de ésta, incrementos de temperatura en los dedos afectados. La sesión altera el reparto de flujo sanguíneo entre las dos manos.

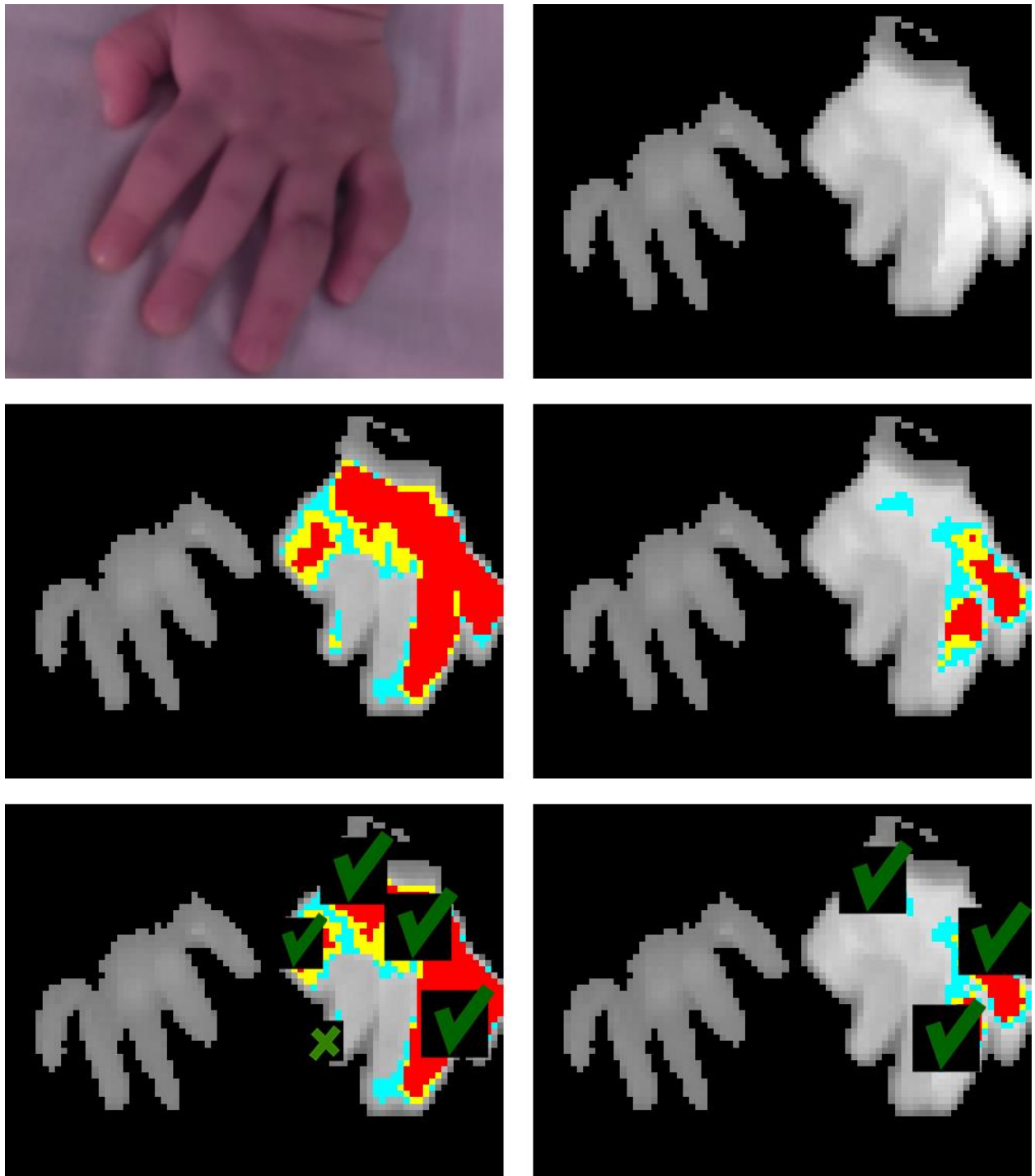


Figura 4-2. Resultados en Paciente 1.

*Análisis de resultados:* Se trata de la sesión empleada durante parte del Capítulo 3 como ejemplo. En este caso se cuenta con una imagen real de referencia. Las zonas visiblemente afectadas coinciden con el mapa generalizado (centro izquierda y abajo izquierda en la Figura), incluso en grado. En el dedo índice, por encima de la articulación interfalángica distal [18], aparece una pequeña mancha cian con 3 píxeles amarillos que puede ser provocada por la presión que hace el paciente con la mano sobre ese

punto (aunque todas las imágenes de este paciente presentan un aumento de temperatura local en esa zona, por lo que podría ser un nuevo foco).

En cuanto al mapa relativo Pixel Select (centro derecha y abajo derecha en la figura), se ha seleccionado la falange medial del dedo meñique [19], puesto que pertenece a la zona más lesionada. También se observa alto grado de concordancia de resultados, ya que la zona roja coincide con las partes más abultadas y el escalado de colores también parece correcto.

- Paciente 2

*Diagnóstico médico:* Hemangioma ulcerado. Paciente visto anteriormente en varias sesiones.

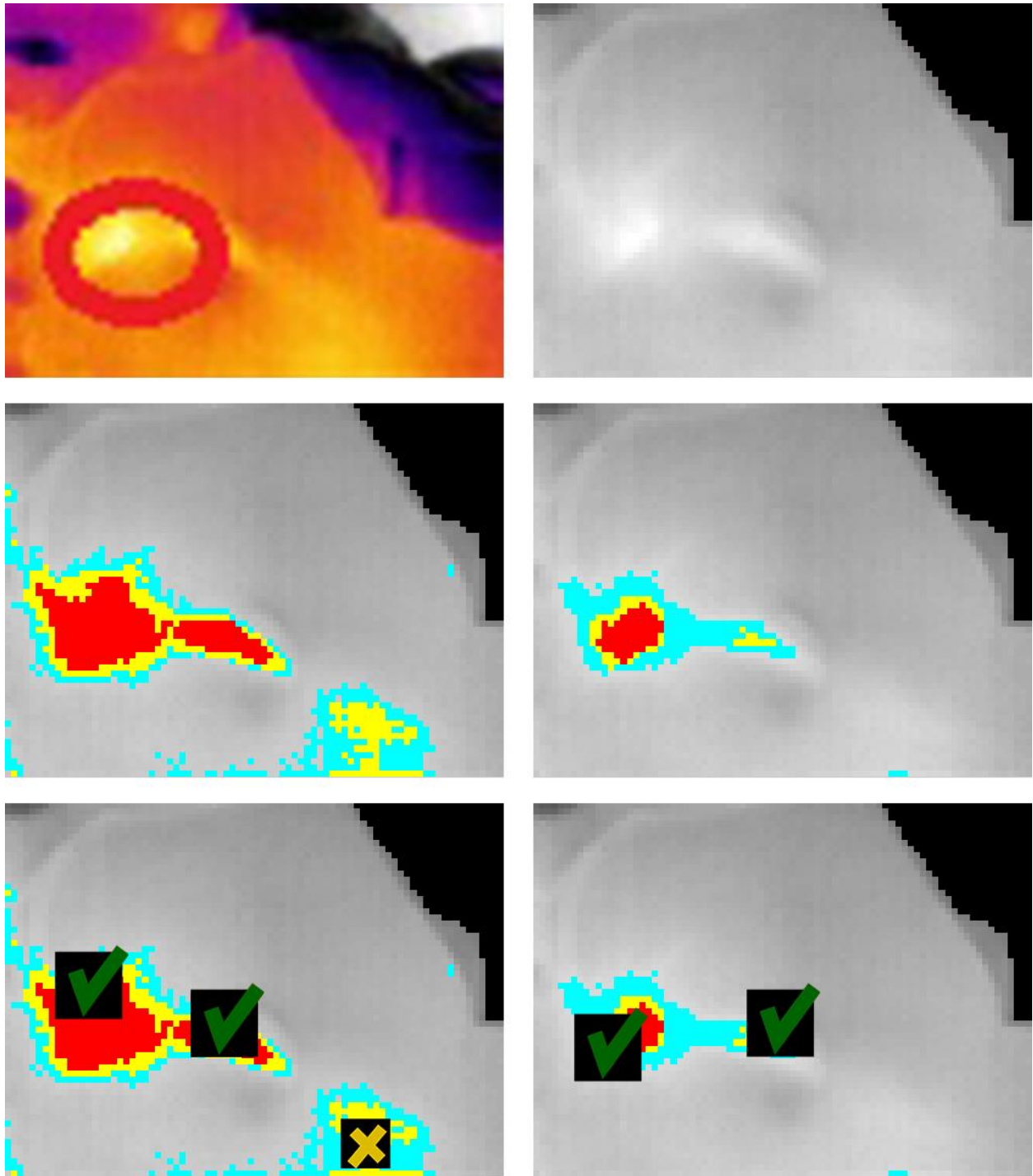


Figura 4-3. Resultados en Paciente 2.

*Análisis de resultados:* El hemangioma es detectado tal y como indica la referencia del doctor (se debe tener en cuenta que la imagen estudiada tiene ligeramente más zoom y una mínima variación de ángulo respecto a la imagen de referencia). En el mapa generalizado (centro izquierda y abajo izquierda en la Figura) aparece una mancha amarilla en la parte inferior de la imagen hacia la derecha. Al estar en esa parte de la imagen, se presupone causada por una fuente de calor externo, como una mano que sujeta el cuerpo del paciente. En posteriores ejemplos se demostrará que este hecho ocurre con relativa frecuencia.

En cuanto al mapa relativo (centro derecha y abajo derecha en la Figura), se ha seleccionado el punto

de calor máximo (parte interior izquierda del círculo rojo en la imagen de referencia). El resultado es un interesante estudio de las temperaturas del hemangioma, ya que se ve claramente cómo, a medida que se avanza hacia la derecha por el hemangioma, la temperatura va bajando (transición amarilla y zona cian) para, posteriormente, volver a subir la temperatura (amarillo) al adentrarnos en el segundo foco de calor. Al no llegar en ningún momento al color rojo, se deduce que la temperatura del segundo foco es inferior a la del primero, que posee una amplia zona roja. También se puede apreciar que la mancha de abajo no es detectada en este caso, al ser de una temperatura relativa bastante inferior a la del píxel seleccionado.

- Paciente 3

*Diagnóstico médico:* Hemangioma en involución en la zona del labio inferior de la boca. Lado derecho de la foto. No se aprecian variaciones térmicas significativas en la lesión.

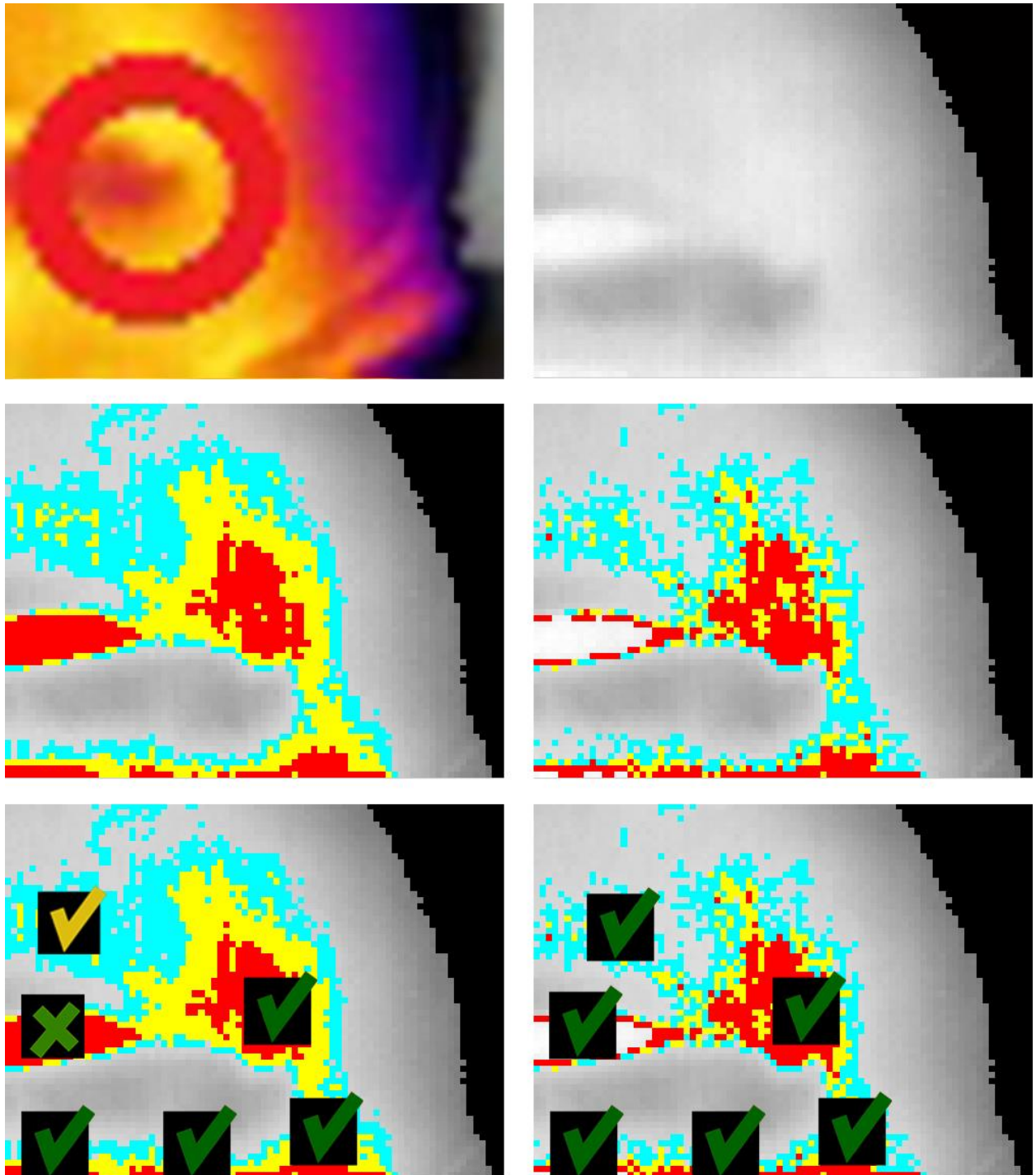


Figura 4-4. Resultados en Paciente 3.

*Análisis de resultados:* Se trata de un caso con resultados muy interesantes debido a que el diagnóstico advierte que no se aprecian cambios significativos en temperatura, pese a lo cual, con esta Herramienta, se consigue realizar un mapa que demuestra que sí hay diferencias de temperatura entre la zona lesionada y el resto. En el mapa generalizado (centro izquierda y abajo izquierda en la Figura), se observa cómo la parte que rodea al labio izquierdo tiene una temperatura sensiblemente superior al resto de la cara, que tiende a degradarse a cian para luego desaparecer. Destacar también que la zona roja ovalada en la parte central izquierda de la imagen se ha marcado como fallo pequeño porque se trata de la boca ligeramente abierta del paciente, la cual genera el calor que se refleja en el mapa. Se

trata pues de una discordancia justificada. Es importante recordar que los pacientes son niños de edad muy temprana, por lo que es difícil evitar que ocurran estos hechos o que se necesite una tercera persona para sujetar los cuerpos en el momento de la captura, lo que se traduce en fuentes de calor no deseadas.

El mapa relativo (centro izquierda y abajo izquierda en la Figura) muestra unos resultados similares a los del primer mapa. Quizá un poco más afinados, como es normal dadas las características del método. Se ha seleccionado el píxel correspondiente a la parte superior derecha en el interior del círculo. Esta vez se trata de un punto al azar dentro de la zona marcada por el doctor, ya que en las imágenes originales no se aprecian diferencias de temperatura, como bien advertía el informe médico. El mapa demuestra que la boca tiene mayor temperatura que el hemangioma, reafirmando la hipótesis expuesta en el párrafo anterior.



- Paciente 4

*Diagnóstico médico:* Hemangioma en involución en la frente, junto a la ceja derecha. Se ve un pequeño puntito en la zona de lesión.

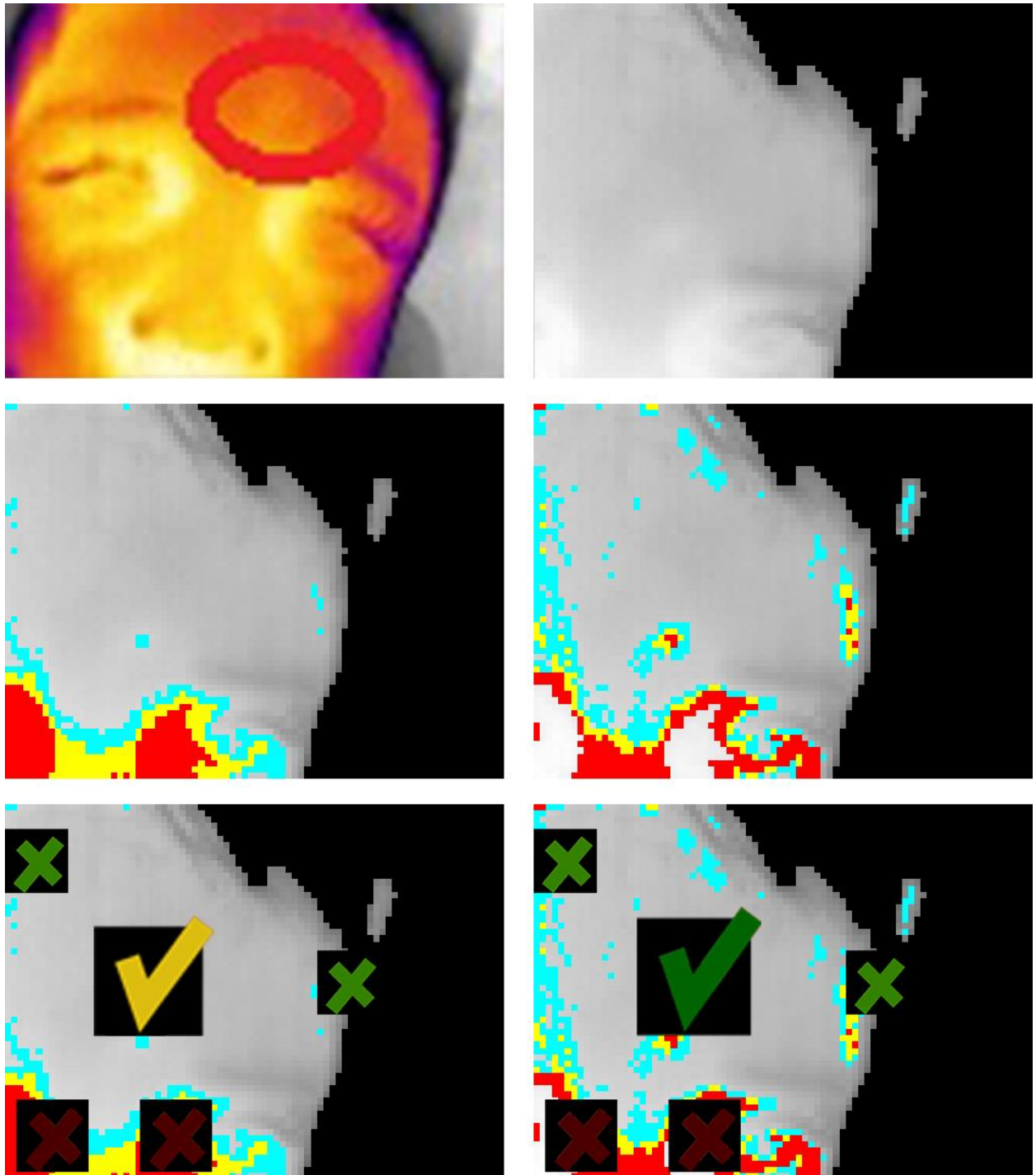


Figura 4-5. Resultados en Paciente 4.

*Análisis de resultados:* Hemangioma muy pequeño y difícil de localizar, casi inapreciable. En el mapa generalizado (centro izquierda y abajo izquierda en la Figura) aparece como un punto de tan solo 4 píxeles. Es detectado como zona de probabilidad de detección baja. En el resto de la frente, las pequeñas zonas en cian corresponden a partes situadas debajo del pelo, lo que provoca ese calor. Puede observarse que arriba a la derecha aparece una pequeña cantidad de píxeles correspondientes a piel del cuerpo humano (es decir, píxeles en tonos grises) que parecen salidos “de la nada”, pero en realidad se trata simplemente de un hueco libre entre dos zonas cubiertas por cabello. Por ello, se concluye que es muy recomendable apartar el pelo de la cara en este tipo de sesiones, ya que provoca

calor y elimina zona útil susceptible de estudio. Por último, llama la atención la elevada temperatura que alcanza la cara por debajo de las cejas. Incluso en la imagen de referencia (arriba izquierda en la Figura) se puede observar un incremento desmesurado de la temperatura rodeando la nariz, siendo aún más evidente en la zona bajo ésta y cerca de los ojos. Desconozco la causa de este hecho.

Para el mapa relativo se ha seleccionado el pequeño hemangioma indicado por el doctor. Ahora se distingue muy claramente la evolución de la temperatura en dicho hemangioma. Siguen molestando las zonas de calor bajo el pelo. El área marcada bajo los ojos, aunque se ha reducido de forma moderada, sigue siendo considerable. En conclusión, las condiciones de captura enturbian ligeramente el importante resultado de detectar un hemangioma muy complicado sobre el que incluso se puede observar la evolución de su calor, a pesar de sus reducidos tamaño y horquilla inicial de temperatura.

- Paciente 5

*Diagnóstico médico:* Hemangiomas de color azul en la espalda. Mayor temperatura en la zona lesionada.

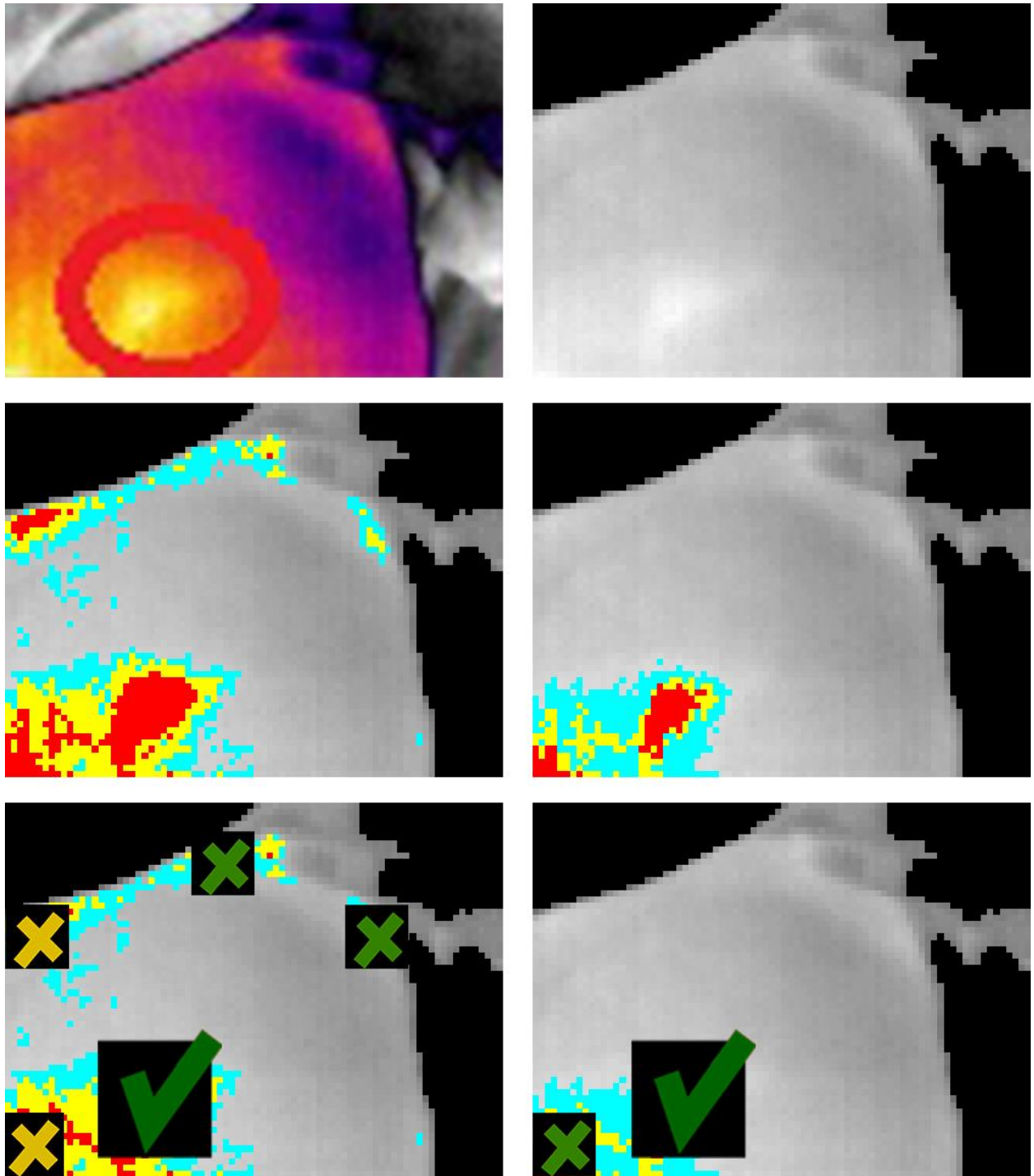


Figura 4-6. Resultados en Paciente 5.

*Análisis de resultados:* Antes de empezar con el análisis de los mapas generados por la Herramienta, se va a hacer un pequeño estudio de las imágenes situadas en la fila de arriba de la Figura (referencia del doctor, en falso color a la izquierda; imagen original en escala de grises a la derecha). Si nos fijamos en la parte de arriba a la derecha en el cuerpo, se ve perfectamente cómo una mano adulta sujeta el cuerpo del paciente. Como se dijo antes, este gesto se realiza de forma instintiva para que el paciente (niño pequeño) no se mueva y así obtener una buena fotografía. Pero la realidad es que ejecutar esta acción provoca que la imagen quede completamente alterada. Se observa perfectamente que el dedo pulgar adulto que está presionando sobre el cuerpo desencadena una franja de calor

importante, provocando un cambio de temperatura grande con respecto a la zona que está por debajo y por encima del dedo. Debido a que la presión se realiza sobre una zona de baja temperatura y a que la Herramienta se ha desarrollado tratando de mermar estos hechos en la medida de lo posible, su impacto en los mapas no es muy elevado (el mapa relativo incluso elimina el efecto del dedo y otros como el de la ropa). Pero una segunda mano está presumiblemente ejerciendo presión cerca del hemangioma, es decir, en una zona caliente. Esto provoca una subida de calor sobre una zona que ya tiene temperatura alta (zona del hemangioma), provocando la aparición de una zona muy luminosa en la esquina inferior izquierda, muy difícil de obviar incluso usando el método Pixel Select.

Entrando puramente en el análisis de los mapas, se destaca que el mapa generalizado (centro izquierda y abajo izquierda en la Figura) detecta sin ningún problema el hemangioma. Lo explicado en el párrafo anterior resta limpieza al conjunto, pero el hemangioma y la evolución de su temperatura tienen la información intacta y lista para ser estudiada sin ningún impedimento.

El método Pixel Select (centro derecha y abajo derecha en la Figura) se ha usado para elegir un punto brillante del hemangioma. El resultado es, como de costumbre, más preciso que en el primer mapa. Imagen casi carente de artefactos, lo que proporcionará una mayor comodidad visual a la hora de realizar un estudio sobre el paciente.

- Paciente 6

*Diagnóstico médico:* Hemangioma de color azul en el torso. Se aprecia claramente una mayor temperatura en la zona lesionada.

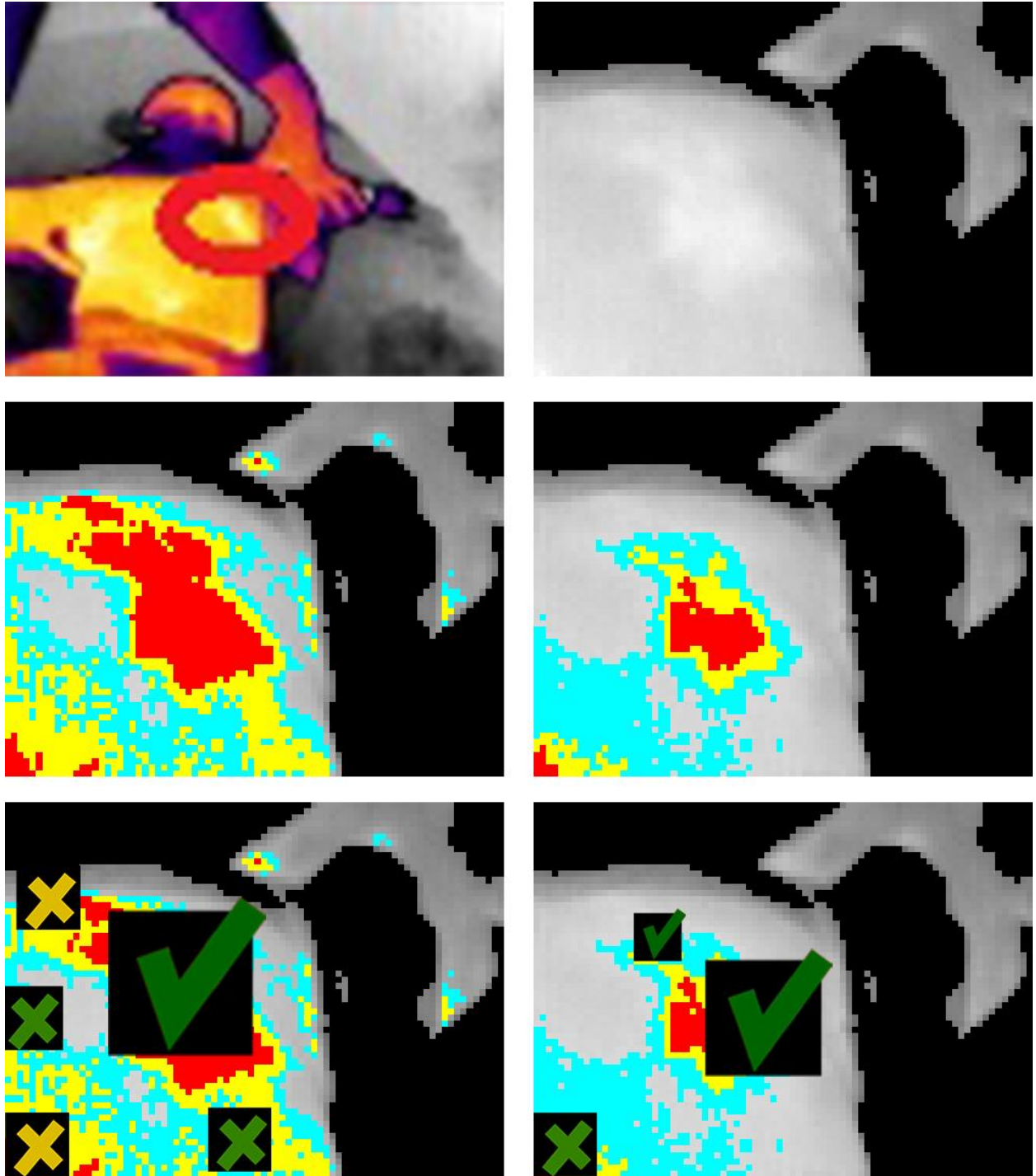


Figura 4-7. Resultados en Paciente 6.

*Análisis de resultados:* La imagen de referencia (arriba a la izquierda en la Figura) evidencia el problema expuesto y desplegado en el Paciente 5: La alteración cromática que supone presionar sobre el cuerpo del paciente, mucho más aún si se hace en zona de altas temperaturas. La mano derecha adulta que presiona el costado derecho del niño provoca un incremento de temperatura que hace que sólo el círculo rojo colocado por el doctor permita distinguir cuál es un hemangioma y cuál no. Aunque la imagen estudiada (arriba a la derecha en la Figura) tenga un mayor zoom, no queda exenta de los efectos de esta acción.

Aun así, el hemangioma vuelve a ser detectado de manera muy eficiente por la Herramienta a través del mapa generalizado (centro izquierda y abajo izquierda en la Figura). Se pueden distinguir claramente sus distintas temperaturas.

En el mapa relativo (centro derecha y abajo derecha en la Figura) se ha elegido un píxel de la parte baja del hemangioma, ya que a simple vista parece tener mayor intensidad. Efectivamente, el segundo mapa demuestra que la mayor temperatura del hemangioma se encuentra en la parte baja, lo que presumiblemente implica que el origen de la mancha se encuentra ahí. A medida que se escala por el hemangioma, la temperatura se va reduciendo, como muestra el citado mapa relativo.

- Paciente 7

*Diagnóstico médico:* Hemangioma en tratamiento y en involución situado en el brazo derecho. Zona rojiza, muestra una mayor actividad térmica.

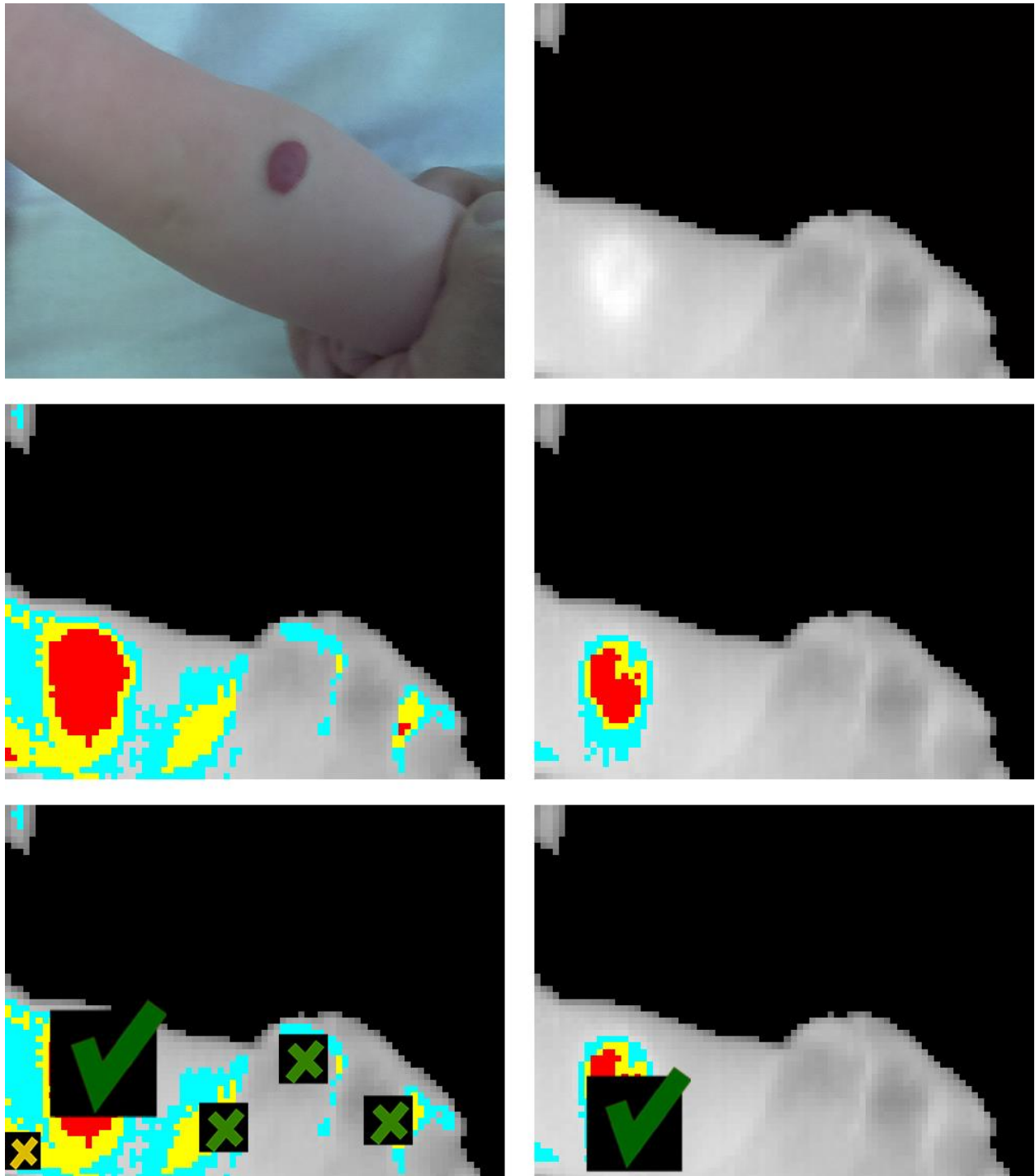


Figura 4-8. Resultados en Paciente 7.

*Análisis de resultados:* Clara detección del hemangioma por parte del mapa de probabilidades generalizado (centro izquierda y abajo izquierda en la Figura). La mano que sujeta el brazo del paciente provoca ligeros artefactos que no afectan a la vista general de la imagen.

Con el método Pixel Select (centro derecha y abajo derecha en la Figura) eligiendo un píxel del hemangioma se brinda un mapa aún más nítido y libre de artefactos. Cierra de manera eficiente el contorno del hemangioma y permite ver cómo se distribuye el calor dentro de él. Resultado eficiente y libre de dudas.

- Paciente 8

*Diagnóstico médico:* Hemangioma infantil. Se ha estado tratando durante años y se ha reducido. Se aprecia aún una zona en la que temperatura es más alta que la de la zona no lesionada.

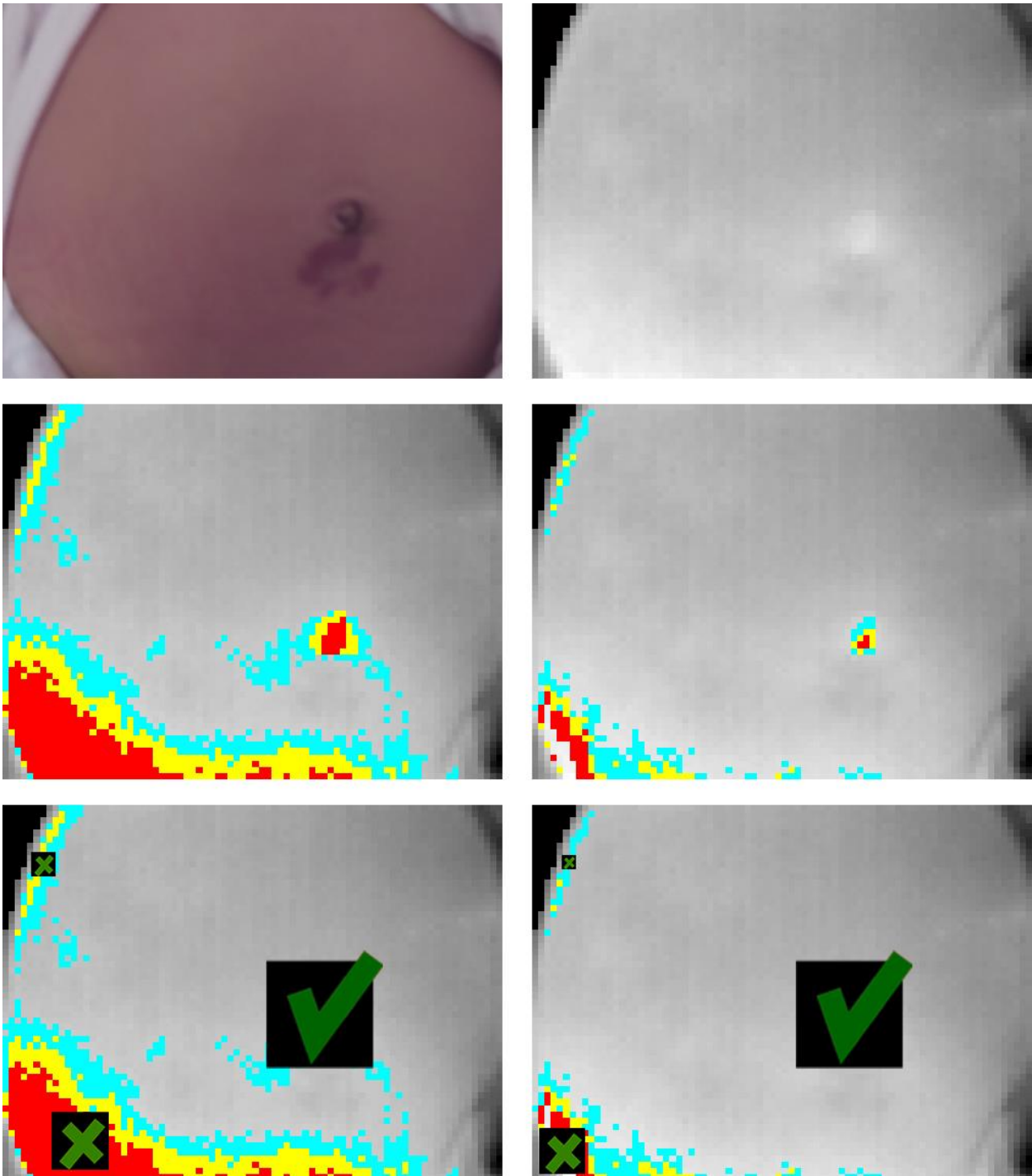


Figura 4-9. Resultados en Paciente 8.

*Análisis de resultados:* Cumple su cometido, localizando claramente la zona luminosa del hemangioma que advierte el diagnóstico médico. Fuentes externas de calor en la parte baja del vientre y en el costado derecho, pero la imagen de referencia (arriba a la izquierda en la figura) despeja cualquier duda sobre su origen: La ropa del paciente.

Es destacable el hecho de que el mapa generalizado (centro izquierda y abajo izquierda en la Figura) brinda una mayor información que el mapa relativo (centro derecha y abajo derecha en la Figura), habiéndose seleccionado esta vez un píxel entre los más luminosos del hemangioma. De nuevo



resultados limpios y eficientes, esta vez con la salvedad del calor expulsado a través del ropaje.

- Paciente 9

*Diagnóstico médico:* Hemangioma en involución en el centro del pecho. Llega a existir variación térmica en el contorno de la lesión.

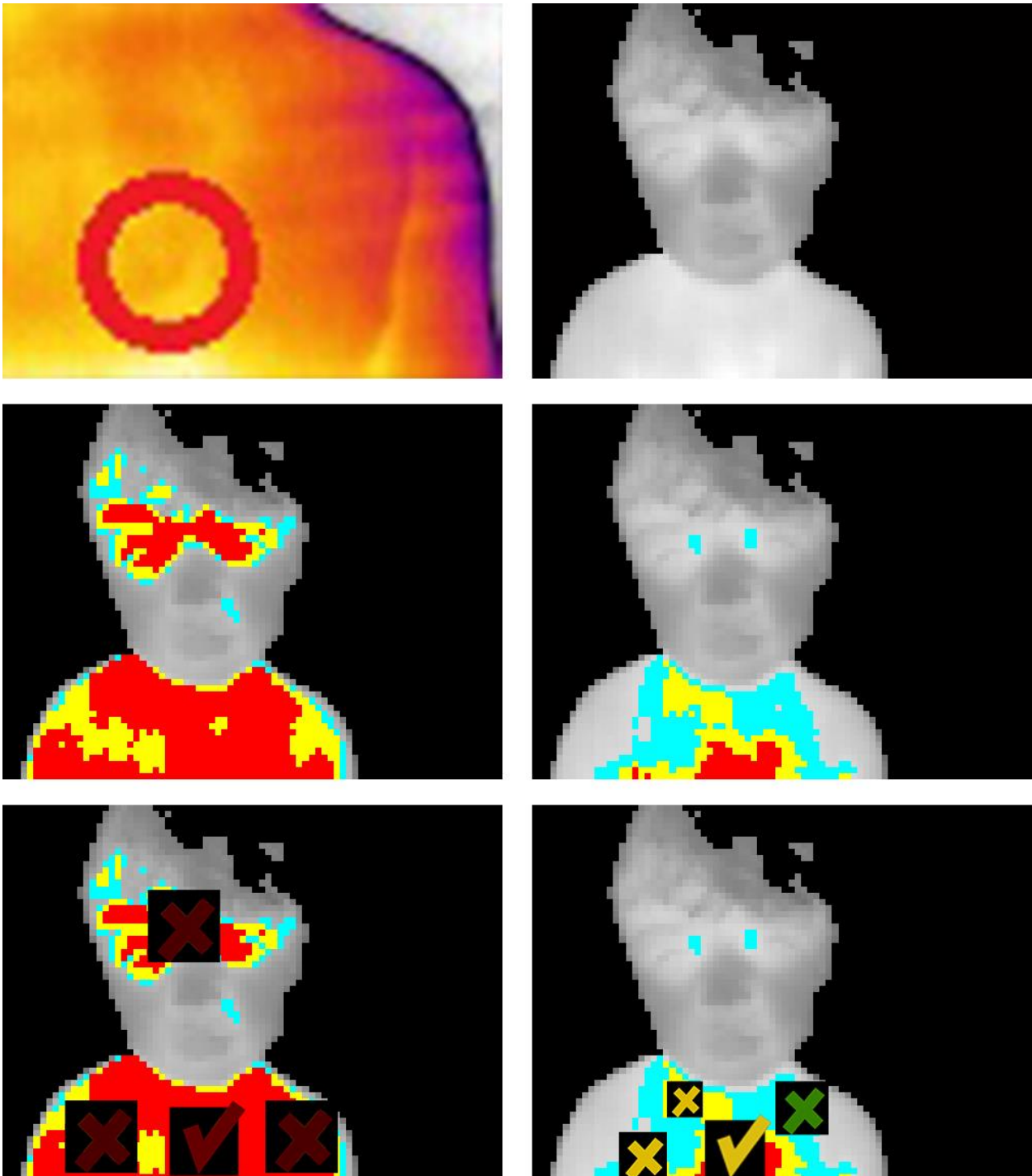


Figura 4-10. Resultados en Paciente 9.

*Análisis de resultados:* Realizar un análisis responsable y representativo de los resultados implica incluir casos en los que dichos resultados no son los deseables. Este ejemplo es, con diferencia, el que peores resultados ha brindado. La experiencia y conocimientos adquiridos a lo largo de este Trabajo de Fin de Grado me permiten elaborar un argumento que explica lo ocurrido: La fotografía está tomada desde una distancia considerable, lo que, unido a la baja resolución de la cámara (80x60 píxeles), provoca que cada píxel abarque una región muy grande del cuerpo. El hecho evidente de que cada píxel sólo pueda tener un color, hace que la información almacenada en la fotografía sea bastante

imprecisa. Así, los resultados del mapa generalizado (centro izquierda y abajo izquierda en la Figura) muestran resultados nada aclaratorios. Es cierto que la zona que según el informe contiene el hemangioma (centro del pecho) está en rojo, pero aun así la información es imprecisa.

Con el método Pixel Select (centro derecha y abajo derecha en la Figura) punteando sobre la región lesionada se obtiene una mejora significativa en los resultados, acotando mucho más los colores y proporcionando una información mucho más intuitiva y cercana al informe médico. A pesar de ello, y haciendo uso del carácter crítico que requiere este tipo de estudios, es necesario decir que el resultado dista de ser óptimo.

- Paciente 10

*Diagnóstico médico:* Hemangioma en el labio superior de la boca. Lado izquierdo de la foto.

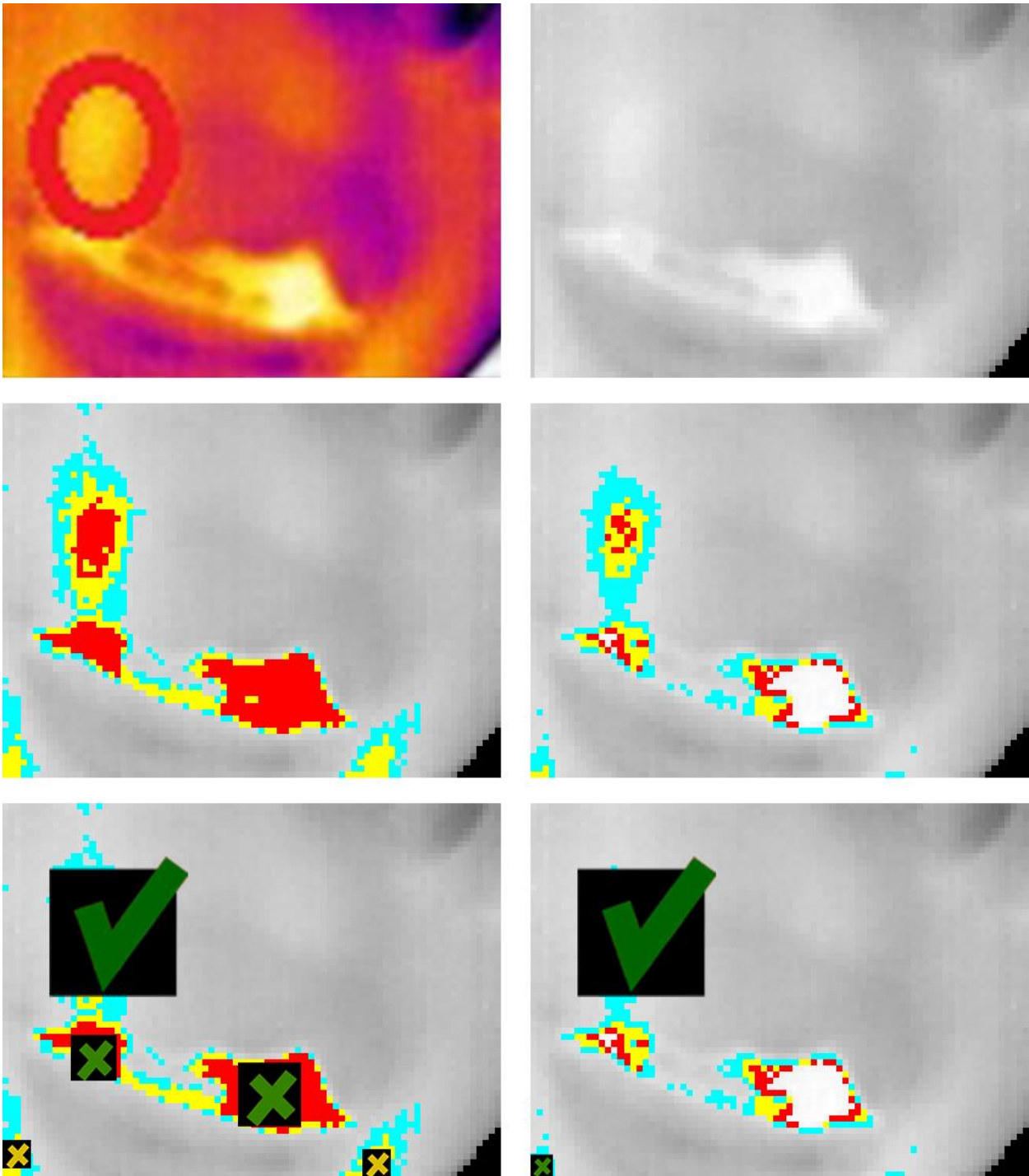


Figura 4-11. Resultados en Paciente 10.

*Análisis de resultados:* Caso importante porque se vuelve a detectar un hemangioma bajo condiciones que no son sencillas. El aumento de temperatura del hemangioma es ligero, y la consecución de resultados aclaratorios se complica por el hecho de que justo donde acaba el hemangioma por abajo se tiene de nuevo una boca abierta. Haciendo un ejercicio de agudeza visual sobre la imagen de referencia y la imagen original en escala de grises (arriba izquierda y arriba derecha de la Figura, respectivamente), pueden llegar a observarse incluso los dientes inferiores. La expulsión de aire y la dificultad para distinguir formas en imágenes termográficas hacen que sea muy difícil dilucidar dónde empieza el hemangioma y donde acaba el calor expulsado por el aliento.

Sin embargo, el mapa de probabilidades generalizado (centro izquierda y abajo izquierda en la Figura) deja ver perfectamente dónde están esos límites, proporcionando así condiciones mucho mejores para el estudio por parte del doctor.

El mapa selectivo (centro derecha y abajo derecha en la Figura) seleccionando un punto de entre los rodeados por el círculo rojo en la imagen de referencia, continúa por la misma línea de resultados. Las características del método permiten que la boca vaya presidida por zonas “blancas” que informan estar en un rango de temperaturas superior al del hemangioma.

- Paciente 11.1

*Diagnóstico médico:* Malformación capilar venosa y linfática de bajo flujo (síndrome de Klippel Trenaunay). No se observan variaciones térmicas en la lesión. Esta lesión es interesante para comparar con otras de alta flujo, en las cuales sí se detectan variaciones de temperatura.

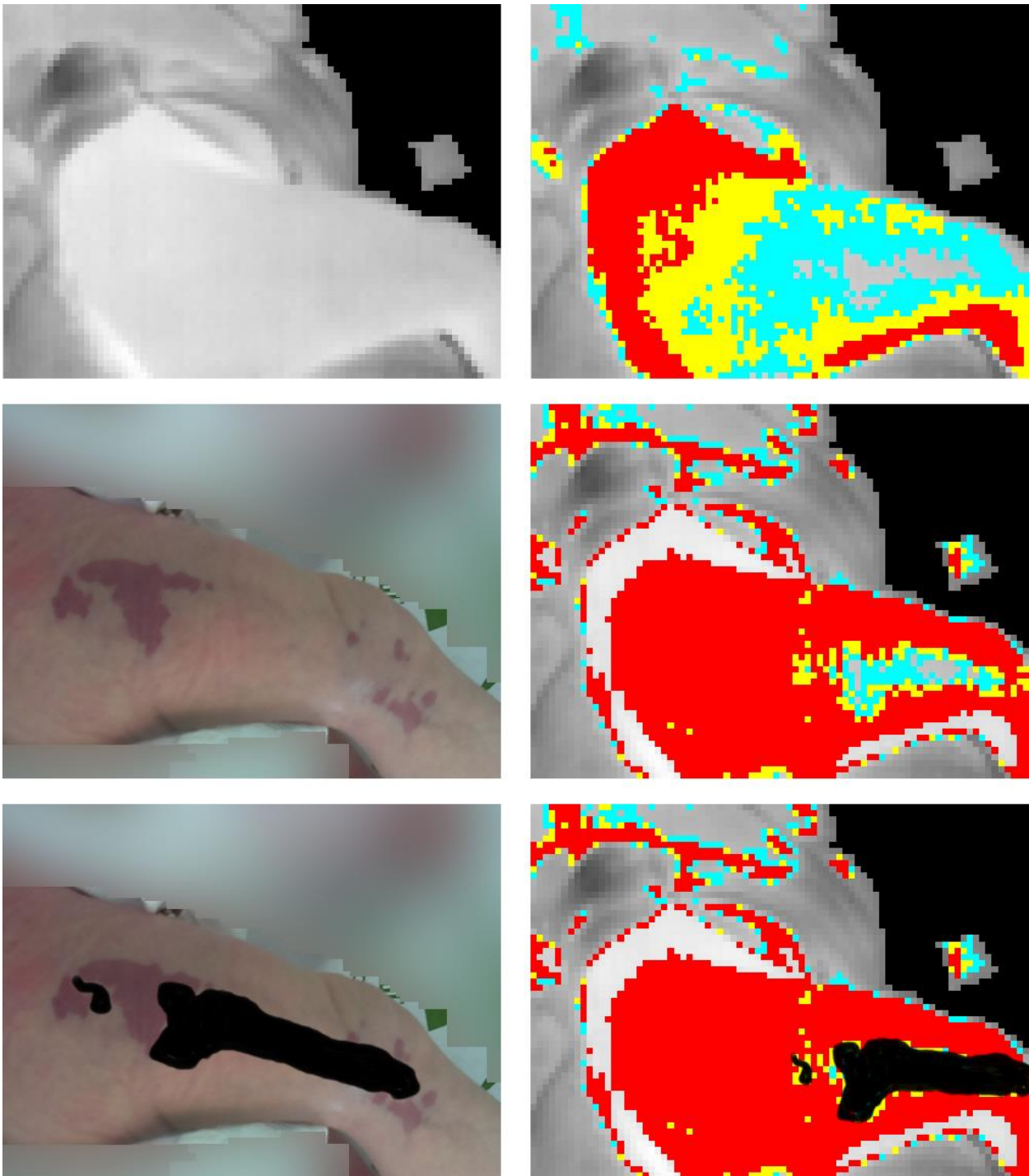


Figura 4-12. Resultados en Paciente 11, caso 1.

*Análisis de resultados:* Los casos 11 y 12 son muy especiales. Ambos pertenecen a un mismo paciente, aquejado de síndrome de Klippel Trenaunay, una afección poco frecuente que a menudo consiste en, entre otros, la aparición en la piel de manchas en vino de Oporto [20] [21]. A pesar de que a primera vista pueda parecer que se trata de los peores mapas de cuantos se han expuesto, lo cierto es que son, sin duda alguna, los que brindan resultados más interesantes.

Debido a que son casos que se salen de lo habitual, se van a tratar como tal, empezando por la propia

estructura de la Figura 4-12 y de la Figura 4-13. El esquema de imágenes explicado al principio del Capítulo será diferente para el Paciente 11, con el fin de mejorar la comprensión visual del conjunto. Así, la Figura 4-12 lleva el siguiente reparto de imágenes:

- Parte superior izquierda de la Figura: Imagen original en escala de grises.
- Parte central izquierda de la Figura: Imagen de referencia (imagen no térmica).
- Parte inferior izquierda de la Figura: Imagen de referencia con zona a destacar pintada de forma manual, en negro.
- Parte superior derecha de la Figura: Mapa de probabilidades generalizado (no se usará).
- Parte central derecha de la Figura: Mapa de probabilidades relativo, eligiendo un píxel cualquiera de entre los pertenecientes a la lesión en la imagen de referencia.
- Parte inferior derecha de la Figura: Mapa de probabilidades relativo al que se traslada la misma zona destacada en negro y dibujada a mano, escalada uniformemente y rotada según la variación de zoom y ángulo existente respecto a la imagen de referencia.

El pronóstico médico nos informa de que en este caso no se detectan variaciones térmicas en la zona lesionada. Y, efectivamente, realizar una pasada normal con la Herramienta sólo produce resultados aparentemente vacíos. Sin embargo, y tras un extenso estudio de los resultados, se puede llegar a la conclusión de que la Herramienta sí detecta una muy tenue variación de temperatura entre la zona con lesiones y la zona sin lesiones.

Para ello, hemos hecho Pixel Select en la zona donde existe lesión, en la mancha más grande. Esta parte es cian en el mapa generalizado, es decir, de probabilidad baja. Así, al hacer Pixel Select sobre ella, el cian pasa a ser rojo, abriéndose tres nuevos “niveles hacia abajo”: Amarillo relativo, cian relativo y “por debajo de cian relativo” (es decir, sin color). Pues bien, si marcamos el contorno exterior de la lesión grande, situada aproximadamente a la altura de la mediación del cuádriceps, y continuamos estableciendo un tubo imaginario “libre de lesiones” que conecte con las lesiones de la rodilla, habremos creado un dibujo con forma parecida a una pistola. Es fácil observar que esta “pistola” es como el mar que une dos islas, la de la lesión del cuádriceps y la de las lesiones de la rodilla.

Mirando ahora el mapa de probabilidades relativo (centro derecha y abajo derecha en la Figura), podemos comprobar que si trasladamos la “pistola” a su región equivalente (con la lógica rotación y escalado, ya que ambas imágenes están tomadas desde un punto ligeramente diferente), ésta encaja perfectamente en la zona “amarillo-cian-sin color”. Es decir, aunque sea muy leve, existe una diferencia palpable de temperatura en las lesiones de la pierna.

Si se observan una vez más las cuatro imágenes inferiores de la Figura, se pueden apreciar detalles de hasta qué punto encajan ambas zonas, y más detalles como que la zona de “la pistola” es la única bajada local de temperaturas que se encuentra en toda la pierna. Y son las lesiones las que marcan los límites de la “pistola”. Incluso se puede trazar una pequeña zona entre las lesiones del cuádriceps que también tiene su homónimo en el mapa relativo.

Se trataría pues, de un resultado de potencial importancia, fruto del análisis de los resultados arrojados por la Herramienta.

- Paciente 11.2

*Diagnóstico médico:* Malformación capilar venosa y linfática de bajo flujo (síndrome de Klippel Trenaunay). No se observan variaciones térmicas en la lesión. Esta lesión es interesante para comparar con otras de alta flujo, en las cuales sí se detectan variaciones de temperatura.

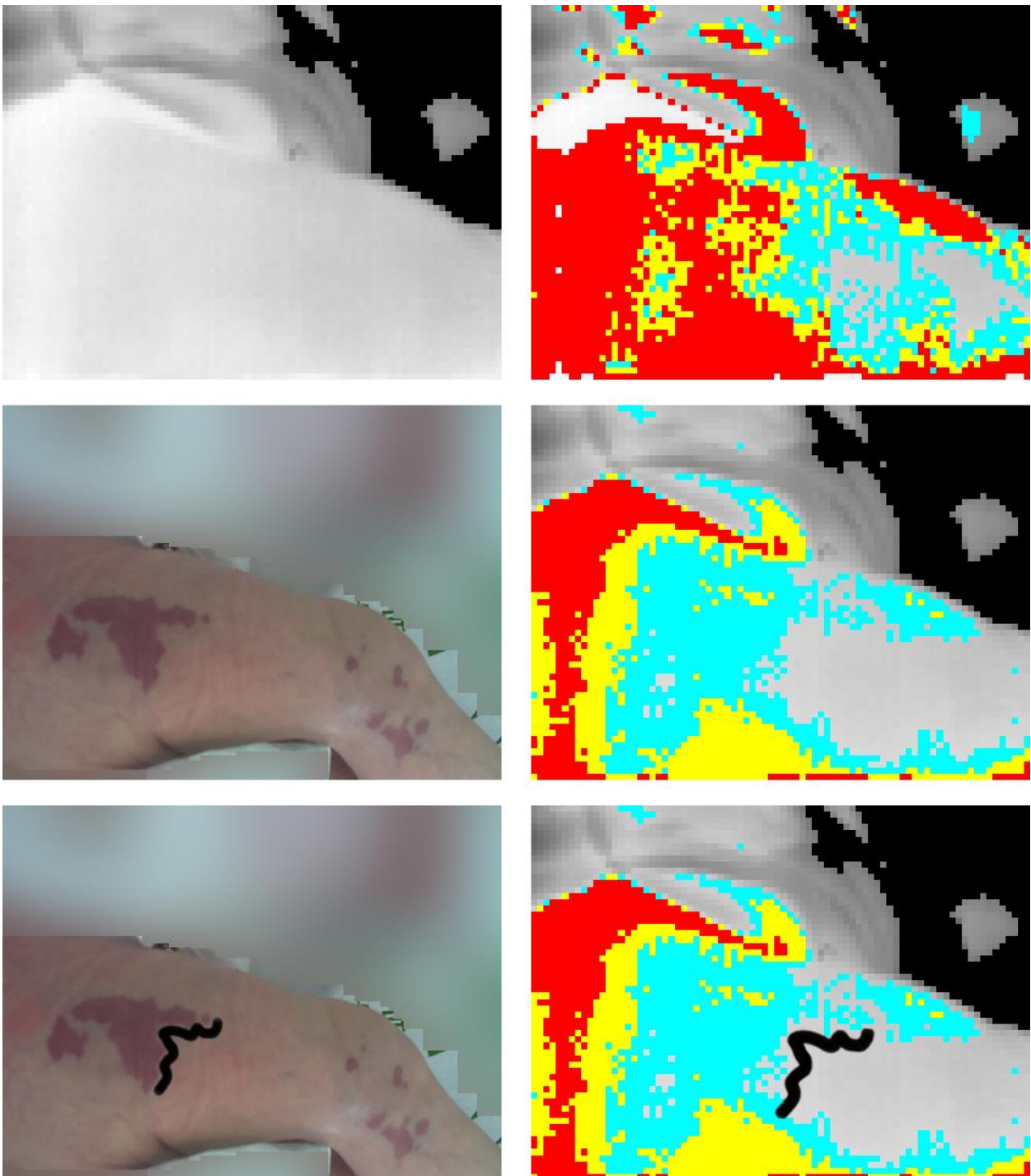


Figura 4-13. Resultados en Paciente 11, caso 2.

*Análisis de resultados:* El caso 2 del paciente 11 (aquejado de síndrome de Klippel Trenaunay) supone una vuelta de tuerca más a lo expuesto en el caso 1. En esta ocasión, la distribución de las imágenes en la Figura 4-14 es:

- Parte superior izquierda de la Figura: Imagen original en escala de grises.



- Parte central izquierda de la Figura: Imagen de referencia (imagen no térmica).
- Parte inferior izquierda de la Figura: Imagen de referencia con zona a destacar pintada de forma manual, en negro.
- Parte superior derecha de la Figura: Mapa de probabilidades relativo a un píxel de lesión (no se usará).
- Parte central derecha de la Figura: Mapa de probabilidades generalizado.
- Parte inferior derecha de la Figura: Mapa de probabilidades generalizado, al que se traslada la misma zona destacada en negro y dibujada a mano, escalada uniformemente y rotada según la variación de zoom y ángulo existente respecto a la imagen de referencia.

Puede observarse que esta vez se hace el experimento con el mapa generalizado, para usar uno distinto aunque tenga menor nivel de detalle. La imagen de referencia y la imagen en escala de grises son diferentes a las del caso 1, aunque pertenecientes a la misma sesión que aquéllas.

En este caso se trabaja únicamente con el contorno exterior de la lesión grande situada a la altura del cuádriceps, dibujando una figura que traza dicho contorno. Sería equivalente a la culata de la “pistola”, pero dibujada algo más en detalle. Nuevamente se obtiene una figura completamente trasladable a su homónimo mapeado. Las conclusiones extraídas en el caso 11.1 son igualmente aplicables a este caso 11.2.

Tras el análisis de estos doce casos en once pacientes, se cerrará el capítulo con un esquema que trata de cuantificar los resultados obtenidos. Es necesario recordar que los símbolos de concordancia y discordancia aplicados a los mapas, a pesar de ser una buena herramienta visual, no suponen un método de cuantificación absoluto, ya que la elección de uno u otro lleva implícitas una serie de decisiones intangibles, como el impacto visual o comprensivo que pueda tener un fallo en el conjunto de la imagen. Aun así, y haciendo hincapié en la necesidad de leer y comprender todos los análisis realizados en este Capítulo antes de consultar el esquema, debido a que éste es un complemento de los análisis y no un sustituto, se presenta a continuación el citado esquema de valoración de resultados:

Tabla 4-1. Valoración esquemática de resultados.

Paciente	Localización de lesión, <i>mapa generalizado</i>	Localización de lesión, <i>mapa relativo</i>	Grado de concordancia en localización de lesión, <i>mapa generalizado</i>	Grado de concordancia en localización de lesión, <i>mapa relativo</i>	Grado de discordancias justificadas, <i>mapa generalizado</i>	Grado de discordancias justificadas, <i>mapa relativo</i>	Grado de discordancias no justificadas con suficiente certeza, o de origen desconocido, <i>mapa generalizado</i>	Grado de discordancias no justificadas con suficiente certeza, o de origen desconocido, <i>mapa relativo</i>
1	Sí	Sí	Alto	Alto	Nulo	Nulo	Bajo	Nulo
2	Sí	Sí	Alto	Alto	Nulo	Nulo	Bajo-Medio	Nulo
3	Sí	Sí	Alto	Alto	Bajo	Nulo	Bajo-Medio	Bajo
4	Sí	Sí	Medio	Alto	Bajo	Bajo	Medio-Alto	Medio-Alto
5	Sí	Sí	Alto	Alto	Bajo-Medio	Bajo	Nulo	Nulo
6	Sí	Sí	Alto	Alto	Medio	Bajo-Medio	Nulo	Nulo
7	Sí	Sí	Alto	Alto	Bajo-Medio	Nulo	Nulo	Nulo
8	Sí	Sí	Alto	Alto	Medio	Bajo	Nulo	Nulo
9	Sí	Sí	Bajo	Medio	Alto	Medio	Nulo	Nulo
10	Sí	Sí	Alto	Alto	Bajo	Nulo	Bajo	Bajo
11.1	Sí	Sí	Bajo	Bajo	Medio-Alto	Alto	Nulo	Nulo
11.2	Sí	Sí	Bajo	Bajo	Medio-Alto	Alto	Nulo	Nulo

Haciendo en escueto balance general de cara a la información de la tabla, se puede decir que destaca la localización del 100% de las lesiones, el alto grado de concordancia de los mismos con el resultado esperado y el hecho de que la mayoría de las discordancias proviene de motivos justificados<sup>4</sup>. (ropa, pelo, presión ejercida por manos para sujetar a los pacientes debido a su corta edad, etc.), lo que se traduce en que la Herramienta es sensible a estos agentes externos (estas ideas se desarrollarán en el siguiente Capítulo). También es relevante el hecho de que los resultados más interesantes de todo el Capítulo provienen de los dos casos con peor puntuación analítica según la Tabla 4-1, es decir, Paciente 11.1 y Paciente 11.2. Esto corrobora que la idea de que la Tabla 4-1 es un complemento del contenido desarrollado durante el Capítulo 4, no un resumen, ni un elemento que lo pueda sustituir.

<sup>4</sup>Este porcentaje sería mucho mayor si se tuviera en cuenta que la mayoría de las discordancias no justificadas provienen muy probablemente de factores externos (los mismos que provocan las discordancias justificadas), pero son casos en los que no se pueden asegurar con certeza total, por ejemplo porque no se vea la mano que presiona en la imagen.

# 5 CONCLUSIONES Y LÍNEAS FUTURAS

---

*Si el resultado confirma la hipótesis, has hecho una medida. Si el resultado es contrario a la hipótesis, has hecho un descubrimiento.*

*- Enrico Fermi -*

La experiencia y conocimientos adquiridos a lo largo de este Trabajo de Fin de Grado, recogidos en los Capítulos 1 a 4 de este Documento, proporcionan las herramientas necesarias para elaborar una serie de conclusiones que permiten arrojar luz sobre el presente y futuro del tema que nos ocupa.

Empezamos por el primer gran objetivo del Proyecto, la eliminación del fondo de la imagen, a modo de telón negro, para que la información visual se reduzca al cuerpo humano objeto de estudio. Un vistazo rápido por las decenas de figuras contenidas en este Documento es más que suficiente para concluir que el objetivo está cumplido, en tanto en cuanto la Herramienta realiza la función que se espera de ella. De esta forma, un gran número de condiciones externas aleatorias dejan de ser un obstáculo para el estudio, ya que son suprimidas mediante el software.

Por supuesto, esta parte del código es extrapolable a otros estudios de características similares. Cualquier tratamiento de imágenes térmicas se puede valer de esta función para eliminar su información no deseada. Además, el carácter matricial de MATLAB y las facilidades que otorga al usuario para operar con los elementos nulos de las matrices (píxeles de fondo en la imagen), permiten que estemos ante un resultado escalable. En todo momento se tiene información exacta de la localización de estos píxeles, por lo que resulta casi inmediato realizar acciones futuras sobre esta parte de las imágenes, si el programador así lo estimara conveniente (por ejemplo, un cambio de color).

En cuanto a la segunda meta capital de este Trabajo de Fin de Grado, la extracción de mapas de color, no es aventurado asegurar que lleva implícitas las reflexiones más trascendentales, puesto que esta parte del código es la encargada de materializar la esencia del Proyecto.

Basándonos sobre todo en el contenido del Capítulo 4, no es difícil concluir que la Herramienta es capaz de detectar hemangiomas y otras lesiones de manera eficiente. Pero esta conclusión no queda completa sin desarrollar la idea de que estamos ante una herramienta sensible a perturbaciones externas, dentro de un entorno que para nada está exento de ellas.

¿Nos encontramos ante un problema sin solución? En absoluto. De hecho, estamos ante un paso evolutivo muy común: La aparición de nuevos problemas a medida que se va profundizando en la solución.

La falta de homogeneización en las condiciones de toma de capturas provoca que la propia Herramienta sea la encargada de suplir esta carencia. Y aunque los resultados sean satisfactorios, no cabe duda de que un futuro criterio más unificado a la hora de realizar las fotografías permitirá mejorarlos. Haciendo uso de una analogía: Por muy potentes que sean los programas informáticos de edición de imágenes que venden cientos de millones

de copias por todo el mundo, el resultado de un simple selfie cuando se enfoca la cara correctamente antes de hacer dicha foto siempre será más satisfactorio que el resultado obtenido si se hace el mismo selfie desenfocado y dejamos que sea el software de edición quien enfoque la cara.

A continuación se proponen una serie de condiciones susceptibles de ser unificadas para perfeccionar los resultados en el futuro:

- Iluminación en la sala.
- Temperatura en la sala.
- Distancia entre la cámara y el paciente.
- Ausencia de ropa en la zona susceptible de estudio, ampliando dicha zona con un perímetro de seguridad adecuado.
- Ausencia de otras partes del cuerpo del paciente sobre el área susceptible de estudio, como el cabello.
- Ausencia de terceras personas ejerciendo presión sobre el cuerpo del paciente, ampliando esta zona con un perímetro de seguridad adecuado.

A estos puntos pueden añadirse algunas recomendaciones para el futuro, como aumentar la resolución de la cámara IR o, en su defecto, realizar fotografías en planos muy cercanos a las distintas partes que conforman el cuerpo o la zona a estudiar, para después unir las todas a modo de puzle, formando una imagen de alta resolución.

Todas estas condiciones propuestas y recomendaciones se tornan familiares cuando se piensa en las técnicas de imagen médica más conocidas, como la radiografía o la resonancia magnética. Por todos es sabido que son métodos indispensables para muchos diagnósticos, pero que requieren de no pocas instrucciones que son dadas al paciente antes e incluso durante el proceso. Si la técnica tratada en este Trabajo de Fin de Grado llega a prosperar y estandarizarse algún día, lo hará estando atada inevitablemente a este tipo de requisitos que permiten maximizar la información obtenida.

El futuro de la Herramienta, por tanto, debe ir asociado a un *feedback* que haga más productiva la cooperación médico-ingeniero. Si el software recibe imágenes más homogeneizadas, no sólo podrá devolver mejores resultados, sino que además el programador podrá incorporarle actualizaciones y mejoras a través de nuevas versiones para ajustarse aún más a los objetivos propuestos.

Por otro lado, es importante hacer una reflexión sobre la importancia de la imagen médica. El lector de este Documento habrá podido comprobar, sobre todo a través del Capítulo 4, que existe una enorme variedad de diagnósticos diferentes. Las lesiones vasculares se manifiestan de maneras muy dispares, tanto en el exterior como en el interior del cuerpo. Esto no hace sino poner más trabas al equipo médico encargado de la detección del alcance exacto de la lesión y, por tanto, de su correcto estudio y tratamiento para erradicarla.

Y precisamente porque la finalidad última es eliminar problemas de salud, desde estas líneas se intenta poner de manifiesto la importancia de aunar conocimientos. Gracias a la contribución del campo de la ingeniería, el campo de la medicina puede alcanzar nuevas metas, y viceversa.

# ANEXO A: FUNCIONES INTERNAS DE MATLAB

**D**urante el desarrollo del presente Documento, se han ido presentando las diferentes funciones internas de MATLAB empleadas en este Trabajo de Fin de Grado. Cada vez que ha aparecido una nueva función en algún fragmento de código (de mi autoría) de cuantos se encuentran en este Documento, se ha explicado el motivo de su elección y se ha hecho un breve repaso de su funcionamiento.

No obstante, este Anexo incluye información adicional que complementa y detalla lo que ya se ha expuesto sobre estas funciones integradas. De hecho, los contenidos de las siguientes páginas *no* serán nuevas explicaciones personales, sino que directamente se tomarán capturas de pantalla de la sección de ayuda perteneciente a la web oficial de MathWorks, desarrollador del programa MATLAB: <https://www.mathworks.com/>

Las funciones serán presentadas en orden alfabético, e incluirán la definición, sintaxis y descripción proporcionadas por MathWorks:

**colormap**
R2019b

View and set current colormap collapse all in page

---

**Syntax**

```
colormap map
colormap(map)
colormap(target,map)

cmap = colormap
cmap = colormap(target)
```

---

**Description**

`colormap map` sets the colormap for the current figure to one of the predefined colormaps. If you set the colormap for the figure, then axes and charts in the figure use the same colormap. The new colormap is the same length (number of colors) as the current colormap. When you use this syntax, you cannot specify a custom length for the colormap. To learn more about colormaps, see [What Is a Colormap?](#) example

---

`colormap(map)` sets the colormap for the current figure to the colormap specified by `map`. example

---

`colormap(target,map)` sets the colormap for the figure, axes, or chart specified by `target`, instead of for the current figure. example

---

`cmap = colormap` returns the colormap for the current figure as a three-column matrix of RGB triplets. example

---

`cmap = colormap(target)` returns the colormap for the figure, axes, or chart specified by `target`. example

Figura A-1.Extracto de la página de ayuda de la función colormap. Recuperada de [www.mathworks.com](https://www.mathworks.com)

**edge**

R2019b

Classification edge

[expand all in page](#)**Syntax**

```
E = edge(tree,TBL,ResponseVarName)
E = edge(tree,X,Y)
E = edge(__,Name,Value)
```

**Description**

`E = edge(tree,TBL,ResponseVarName)` returns the classification edge for `tree` with data `TBL` and classification `TBL.ResponseVarName`.

`E = edge(tree,X,Y)` returns the classification edge for `tree` with data `x` and classification `Y`.

`E = edge(__,Name,Value)` computes the edge with additional options specified by one or more `Name,Value` pair arguments, using any of the previous syntaxes. For example, you can specify observation weights.

Figura A-2.Extracto de la página de ayuda de la función `edge`. Recuperada de [www.mathworks.com](http://www.mathworks.com)

**figure**

R2019b

Create figure window

[collapse all in page](#)**Syntax**

```
figure
figure(Name,Value)
f = figure(__)
```

```
figure(f)
figure(n)
```

**Description**

`figure` creates a new figure window using default property values. The resulting figure is the [current figure](#).

`figure(Name,Value)` modifies properties of the figure using one or more name-value pair arguments. For example, `figure('Color','white')` sets the background color to white. [example](#)

`f = figure(__)` returns the `Figure` object. Use `f` to query or modify properties of the figure after it is created. [example](#)

`figure(f)` makes the figure specified by `f` the current figure and displays it on top of all other figures. [example](#)

`figure(n)` finds a figure in which the `Number` property is equal to `n`, and makes it the current figure. If no figure exists with that property value, MATLAB® creates a new figure and sets its `Number` property to `n`.

Figura A-3.Extracto de la página de ayuda de la función `figure`. Recuperada de [www.mathworks.com](http://www.mathworks.com)

## find

R2019b

Find indices and values of nonzero elements

[collapse all in page](#)

### Syntax

```
k = find(X)
k = find(X,n)
k = find(X,n,direction)
```

```
[row,col] = find( __ )
[row,col,v] = find( __ )
```

### Description

`k = find(X)` returns a vector containing the [linear indices](#) of each nonzero element in array `X`. [example](#)

- If `X` is a vector, then `find` returns a vector with the same orientation as `X`.
- If `X` is a multidimensional array, then `find` returns a column vector of the linear indices of the result.
- If `X` contains no nonzero elements or is empty, then `find` returns an empty array.

`k = find(X,n)` returns the first `n` indices corresponding to the nonzero elements in `X`. [example](#)

`k = find(X,n,direction)`, where `direction` is 'last', finds the last `n` indices corresponding to nonzero elements in `X`. The default for `direction` is 'first', which finds the first `n` indices corresponding to nonzero elements. [example](#)

`[row,col] = find( __ )` returns the row and column subscripts of each nonzero element in array `X` using any of the input arguments in previous syntaxes. [example](#)

`[row,col,v] = find( __ )` also returns vector `v`, which contains the nonzero elements of `X`. [example](#)

Figura A-4.Extracto de la página de ayuda de la función `find`. Recuperada de [www.mathworks.com](http://www.mathworks.com)

## fliplr

R2019b

Flip array left to right

[collapse all in page](#)

### Syntax

```
B = fliplr(A)
```

### Description

`B = fliplr(A)` returns `A` with its columns flipped in the left-right direction (that is, about a vertical axis). [example](#)

If `A` is a row vector, then `fliplr(A)` returns a vector of the same length with the order of its elements reversed. If `A` is a column vector, then `fliplr(A)` simply returns `A`. For multidimensional arrays, `fliplr` operates on the planes formed by the first and second dimensions.

Figura A-5.Extracto de la página de ayuda de la función `fliplr`. Recuperada de [www.mathworks.com](http://www.mathworks.com)

**fopen**

R2019b

Open file, or obtain information about open files

[collapse all in page](#)**Syntax**

```
fileID = fopen(filename)
fileID = fopen(filename,permission)
fileID = fopen(filename,permission,machinefmt,encodingIn)
[fileID,errmsg] = fopen( __ )
```

```
fIDs = fopen('all')
```

```
filename = fopen(fileID)
[filename,permission,machinefmt,encodingOut] = fopen(fileID)
```

**Description**

`fileID = fopen(filename)` opens the file, `filename`, for binary read access, and returns an integer file identifier equal to or greater than 3. MATLAB® reserves file identifiers 0, 1, and 2 for standard input, standard output (the screen), and standard error, respectively. [example](#)

If `fopen` cannot open the file, then `fileID` is -1.

`fileID = fopen(filename,permission)` opens the file with the type of access specified by `permission`.

`fileID = fopen(filename,permission,machinefmt,encodingIn)` additionally specifies the order for reading or writing bytes or bits in the file using the `machinefmt` argument. The optional `encodingIn` argument specifies the character encoding scheme associated with the file. [example](#)

`[fileID,errmsg] = fopen( __ )` additionally returns a system-dependent error message if `fopen` fails to open the file. Otherwise, `errmsg` is an empty character vector. You can use this syntax with any of the input arguments of the previous syntaxes. [example](#)

`fIDs = fopen('all')` returns a row vector containing the file identifiers of all open files. The identifiers reserved for standard input, output, and error are not included. The number of elements in the vector is equal to the number of open files. [example](#)

`filename = fopen(fileID)` returns the file name that a previous call to `fopen` used when it opened the file specified by `fileID`. The output `filename` is resolved to the full path. The `fopen` function does not read information from the file to determine the output value.

`[filename,permission,machinefmt,encodingOut] = fopen(fileID)` additionally returns the permission, machine format, and encoding that a previous call to `fopen` used when it opened the specified file. If the file was opened in binary mode, `permission` includes the letter 'b'. The `encodingOut` output is a standard encoding scheme name. `fopen` does not read information from the file to determine these output values. An invalid `fileID` returns empty character vectors for all output arguments. [example](#)

Figura A-6.Extracto de la página de ayuda de la función `fopen`. Recuperada de [www.mathworks.com](http://www.mathworks.com)



**fread**

R2019b

Read data from binary file

[collapse all in page](#)**Syntax**

```

A = fread(fileID)
A = fread(fileID,sizeA)
A = fread(fileID,sizeA,precision)
A = fread(fileID,sizeA,precision,skip)
A = fread(fileID,sizeA,precision,skip,machinefmt)
[A,count] = fread( __ )

```

**Description**

**A = fread(fileID)** reads data from an open binary file into column vector **A** and positions the file pointer at the end-of-file marker. The binary file is indicated by the file identifier, **fileID**. Use **fopen** to open the file and obtain the **fileID** value. When you finish reading, close the file by calling **fclose(fileID)**. example

**A = fread(fileID,sizeA)** reads file data into an array, **A**, with dimensions, **sizeA**, and positions the file pointer after the last value read. **fread** populates **A** in column order.

**A = fread(fileID,sizeA,precision)** interprets values in the file according to the form and size described by **precision**. The **sizeA** argument is optional. example

**A = fread(fileID,sizeA,precision,skip)** skips the number of bytes or bits specified by **skip** after reading each value in the file. The **sizeA** argument is optional. example

**A = fread(fileID,sizeA,precision,skip,machinefmt)** additionally specifies the order for reading bytes or bits in the file. The **sizeA** and **skip** arguments are optional. example

**[A,count] = fread( \_\_ )** additionally returns the number of characters that **fread** reads into **A**. You can use this syntax with any of the input arguments of the previous syntaxes.

Figura A-7.Extracto de la página de ayuda de la función **fread**. Recuperada de [www.mathworks.com](http://www.mathworks.com)

**im2bw**

R2019b

Convert image to binary image, based on threshold

[collapse all in page](#)**Syntax**

```

BW = im2bw(I,level)
BW = im2bw(X,cmap,level)
BW = im2bw(RGB,level)

```

**Description**

**BW = im2bw(I,level)** converts the grayscale image **I** to binary image **BW**, by replacing all pixels in the input image with luminance greater than **level** with the value 1 (white) and replacing all other pixels with the value 0 (black). example

This range is relative to the signal levels possible for the image's class. Therefore, a **level** value of 0.5 corresponds to an intensity value halfway between the minimum and maximum value of the class.

**BW = im2bw(X,cmap,level)** converts the indexed image **X** with colormap **cmap** to a binary image.

**BW = im2bw(RGB,level)** converts the truecolor image **RGB** to a binary image.

Figura A-8.Extracto de la página de ayuda de la función **im2bw**. Recuperada de [www.mathworks.com](http://www.mathworks.com)

**imadjust**
R2019b

Adjust image intensity values or colormap collapse all in page

---

**Syntax**

```
J = imadjust(I)
J = imadjust(I,[low_in high_in])
J = imadjust(I,[low_in high_in],[low_out high_out])
J = imadjust(I,[low_in high_in],[low_out high_out],gamma)

J = imadjust(RGB,[low_in high_in], __)
newmap = imadjust(cmap,[low_in high_in], __)
```

---

**Description**

`J = imadjust(I)` maps the intensity values in grayscale image `I` to new values in `J`. By default, `imadjust` saturates the bottom 1% and the top 1% of all pixel values. This operation increases the contrast of the output image `J`. example

You optionally can perform contrast adjustment using a GPU (requires Parallel Computing Toolbox™).

This syntax is equivalent to `imadjust(I,stretchlim(I))`.

---

`J = imadjust(I,[low_in high_in])` maps intensity values in `I` to new values in `J` such that values between `low_in` and `high_in` map to values between 0 and 1.

---

`J = imadjust(I,[low_in high_in],[low_out high_out])` maps intensity values in `I` to new values in `J` such that values between `low_in` and `high_in` map to values between `low_out` and `high_out`. example

---

`J = imadjust(I,[low_in high_in],[low_out high_out],gamma)` maps intensity values in `I` to new values in `J`, where `gamma` specifies the shape of the curve describing the relationship between the values in `I` and `J`.

---

`J = imadjust(RGB,[low_in high_in], __)` maps the values in truecolor image `RGB` to new values in `J`. You can apply the same mapping or unique mappings for each color channel. example

---

`newmap = imadjust(cmap,[low_in high_in], __)` maps the values in colormap `cmap` to new values in `newmap`. You can apply the same mapping or unique mappings for each color channel.

Figura A-9.Extracto de la página de ayuda de la función `imadjust`. Recuperada de [www.mathworks.com](http://www.mathworks.com)

**imclearborder**
R2019b

Suppress light structures connected to image border collapse all in page

---

**Syntax**

```
J = imclearborder(I)
J = imclearborder(I,conn)
```

---

**Description**

`J = imclearborder(I)` suppresses structures in image `I` that are lighter than their surroundings and that are connected to the image border. Use this function to clear the image border. For grayscale images, `imclearborder` tends to reduce the overall intensity level in addition to suppressing border structures. The output image, `J`, is grayscale or binary, depending on the input. example

---

`J = imclearborder(I,conn)` specifies the pixel connectivity, `conn`. example

Figura A-10.Extracto de la página de ayuda de la función `imclearborder`. Recuperada de [www.mathworks.com](http://www.mathworks.com)

## imdilate

Dilate image

R2019b

[collapse all in page](#)

### Syntax

```
J = imdilate(I,SE)
J = imdilate(I,nhood)
J = imdilate(__,packopt)
J = imdilate(__,shape)
```

### Description

`J = imdilate(I,SE)` dilates the grayscale, binary, or packed binary image `I`, returning the dilated image, `J`. `SE` is a structuring element object or array of structuring element objects, returned by the `strel` or `offsetstrel` functions. [example](#)

You optionally can perform the dilation using a GPU (requires Parallel Computing Toolbox™).

`J = imdilate(I,nhood)` dilates the image `I`, where `nhood` is a matrix of 0s and 1s that specifies the structuring element neighborhood. `imdilate` determines the center element of the neighborhood by `floor((size(nhood)+1)/2)`.

This syntax is equivalent to `imdilate(I,strel(nhood))`.

`J = imdilate(__,packopt)` specifies whether `I` is a packed binary image.

This syntax is not supported on a GPU.

`J = imdilate(__,shape)` specifies the size of the output image. [example](#)

Figura A-11.Extracto de la página de ayuda de la función `imdilate`. Recuperada de [www.mathworks.com](http://www.mathworks.com)

**imfill**
R2019b

Fill image regions and holes collapse all in page

---

### Syntax

```

BW2 = imfill(BW,locations)
BW2 = imfill(BW,locations,conn)
BW2 = imfill(BW,'holes')
BW2 = imfill(BW,conn,'holes')

I2 = imfill(I)
I2 = imfill(I,conn)

BW2 = imfill(BW)
BW2 = imfill(BW,0,conn)
[BW2, locations_out] = imfill(BW)

```

---

### Description

**BW2 = imfill(BW,locations)** performs a flood-fill operation on background pixels of the input binary image BW, starting from the points specified in `locations`. example

You optionally can perform the flood-fill operation using a GPU (requires Parallel Computing Toolbox™).

**BW2 = imfill(BW,locations,conn)** fills the area defined by `locations`, where `conn` specifies the connectivity.

**BW2 = imfill(BW,'holes')** fills holes in the input binary image BW. In this syntax, a hole is a set of background pixels that cannot be reached by filling in the background from the edge of the image. example

**BW2 = imfill(BW,conn,'holes')** fills holes in the binary image BW, where `conn` specifies the connectivity. example

**I2 = imfill(I)** fills holes in the grayscale image I. In this syntax, a hole is defined as an area of dark pixels surrounded by lighter pixels. example

**I2 = imfill(I,conn)** fills holes in the grayscale image I, where `conn` specifies the connectivity. example

**BW2 = imfill(BW)** displays the binary image BW on the screen and lets you define the region to fill by selecting points interactively with the mouse. To use this syntax, BW must be a 2-D image.

Press **Backspace** or **Delete** to remove the previously selected point. Shift-click, right-click, or double-click to select a final point and start the fill operation. Press **Return** to finish the selection without adding a point.

This syntax is not supported on a GPU.

**BW2 = imfill(BW,0,conn)** lets you override the default connectivity as you interactively specify locations.

This syntax is not supported on a GPU.

**[BW2, locations\_out] = imfill(BW)** returns the locations of points selected interactively in `locations_out`. To use this syntax, BW must be a 2-D image. example

This syntax is not supported on a GPU. ⬆

Figura A-12.Extracto de la página de ayuda de la función `imfill`. Recuperada de [www.mathworks.com](http://www.mathworks.com)

**imsharpen**
R2019b

Sharpen image using unsharp masking collapse all in page

---

### Syntax

```

B = imsharpen(A)
B = imsharpen(A,Name,Value)

```

---

### Description

**B = imsharpen(A)** sharpens the grayscale or truecolor (RGB) input image A by using the unsharp masking method. example

**B = imsharpen(A,Name,Value)** uses name-value pairs to control aspects of the unsharp masking. example

Figura A-13.Extracto de la página de ayuda de la función `imsharpen`. Recuperada de [www.mathworks.com](http://www.mathworks.com)

**mean**
R2019b

Average or mean value of array [collapse all in page](#)

---

### Syntax

```
M = mean(A)
M = mean(A,'all')
M = mean(A,dim)
M = mean(A,vecdim)
M = mean(__,outtype)
M = mean(__,nanflag)
```

---

### Description

`M = mean(A)` returns the **mean** of the elements of `A` along the first array dimension whose size does not equal 1. [example](#)

- If `A` is a vector, then `mean(A)` returns the mean of the elements.
- If `A` is a matrix, then `mean(A)` returns a row vector containing the mean of each column.
- If `A` is a multidimensional array, then `mean(A)` operates along the first array dimension whose size does not equal 1, treating the elements as vectors. This dimension becomes 1 while the sizes of all other dimensions remain the same.

---

`M = mean(A,'all')` computes the mean over all elements of `A`. This syntax is valid for MATLAB® versions R2018b and later. [example](#)

---

`M = mean(A,dim)` returns the mean along dimension `dim`. For example, if `A` is a matrix, then `mean(A,2)` is a column vector containing the mean of each row. [example](#)

---

`M = mean(A,vecdim)` computes the mean based on the dimensions specified in the vector `vecdim`. For example, if `A` is a matrix, then `mean(A,[1 2])` is the mean of all elements in `A`, since every element of a matrix is contained in the array slice defined by dimensions 1 and 2. [example](#)

---

`M = mean(__,outtype)` returns the mean with a specified data type, using any of the input arguments in the previous syntaxes. `outtype` can be 'default', 'double', or 'native'. [example](#)

---

`M = mean(__,nanflag)` specifies whether to include or omit NaN values from the calculation for any of the previous syntaxes. `mean(A,'includenan')` includes all NaN values in the calculation while `mean(A,'omitnan')` ignores them. [example](#)

Figura A-14.Extracto de la página de ayuda de la función mean. Recuperada de [www.mathworks.com](http://www.mathworks.com)

**mesh**
R2019b

Mesh surface plot [collapse all in page](#)

---

### Syntax

```
mesh(X,Y,Z)

mesh(Z)
mesh(Z,C)
mesh(__,C)

mesh(ax, __)
mesh(__,Name,Value)
s = mesh(__)
```

---

### Description

`mesh(X,Y,Z)` creates a mesh plot, which is a three-dimensional surface that has solid edge colors and no face colors. The function plots the values in matrix `Z` as heights above a grid in the `x-y` plane defined by `X` and `Y`. The edge colors vary according to the heights specified by `Z`. [example](#)

---

`mesh(Z)` creates a mesh plot and uses the column and row indices of the elements in `Z` as the `x-` and `y-`coordinates.

---

`mesh(Z,C)` additionally specifies the color of the edges.

---

`mesh(__,C)` additionally specifies the color of the edges. [example](#)

---

`mesh(ax, __)` plots into the axes specified by `ax` instead of the current axes. Specify the axes as the first input argument.

---

`mesh(__,Name,Value)` specifies surface properties using one or more name-value pair arguments. For example, 'FaceAlpha',0.5 creates a semitransparent mesh plot. [example](#)

---

`s = mesh(__)` returns the chart surface object. Use `s` to modify the mesh plot after it is created. For a list of properties, see [Surface Properties](#). [example](#)

Figura A-15.Extracto de la página de ayuda de la función mesh. Recuperada de [www.mathworks.com](http://www.mathworks.com)

**rgb2hsv** R2019b

Convert RGB colors to HSV [collapse all in page](#)

---

**Syntax**

```
HSV = rgb2hsv(RGB)
hsvmap = rgb2hsv(rgbmap)
```

---

**Description**

`HSV = rgb2hsv(`[RGB](#)`)` converts the red, green, and blue values of an RGB image to hue, saturation, and value (HSV) values of an HSV image. [example](#)

`hsvmap = rgb2hsv(`[rgbmap](#)`)` converts an RGB colormap to an HSV colormap. [example](#)

Figura A-16. Extracto de la página de ayuda de la función `rgb2hsv`. Recuperada de [www.mathworks.com](http://www.mathworks.com)

**strel**

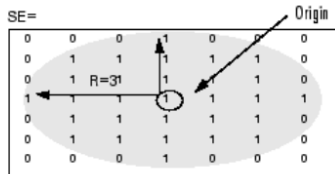
R2019b

Morphological structuring element

[expand all in page](#)**Description**

A `strel` object represents a flat morphological *structuring element*, which is an essential part of morphological dilation and erosion operations.

A flat structuring element is a binary valued neighborhood, either 2-D or multidimensional, in which the `true` pixels are included in the morphological computation, and the `false` pixels are not. The center pixel of the structuring element, called the *origin*, identifies the pixel in the image being processed. Use the `strel` function (described below) to create a flat structuring element. You can use flat structuring elements with both binary and grayscale images. The following figure illustrates a flat structuring element.



To create a nonflat structuring element, use `offsetstrel`.

**Creation****Syntax**

```
SE = strel(nhood)
SE = strel('arbitrary',nhood)
```

```
SE = strel('diamond',r)
SE = strel('disk',r,n)
SE = strel('octagon',r)
SE = strel('line',len,deg)
SE = strel('rectangle',[m n])
SE = strel('square',w)
```

```
SE = strel('cube',w)
SE = strel('cuboid',[m n p])
SE = strel('sphere',r)
```

**Description**

`SE = strel(nhood)` creates a flat structuring element with specified neighborhood `nhood`.

You can also use the syntax `SE = strel('arbitrary',nhood)` to create a flat structuring element with a specified neighborhood.

`SE = strel('diamond',r)` creates a diamond-shaped structuring element, where `r` specifies the distance from the structuring element origin to the points of the diamond.

`SE = strel('disk',r,n)` creates a disk-shaped structuring element, where `r` specifies the radius and `n` specifies the number of line structuring elements used to approximate the disk shape. Morphological operations using disk approximations run much faster when the structuring element uses approximations. [example](#)

`SE = strel('octagon',r)` creates an octagonal structuring element, where `r` specifies the distance from the structuring element origin to the sides of the octagon, as measured along the horizontal and vertical axes. `r` must be a nonnegative multiple of 3.

`SE = strel('line',len,deg)` creates a linear structuring element that is symmetric with respect to the neighborhood center, with approximate length `len` and angle `deg`. [example](#)

`SE = strel('rectangle',[m n])` creates a rectangular structuring element of size `[m n]`.

`SE = strel('square',w)` creates a square structuring element whose width is `w` pixels. [example](#)

`SE = strel('cube',w)` creates a 3-D cubic structuring element whose width is `w` pixels.

`SE = strel('cuboid',[m n p])` creates a 3-D cuboidal structuring element of size `[m n p]`.

`SE = strel('sphere',r)` creates a 3-D spherical structuring element whose radius is `r` pixels. [example](#)

Figura A-17.Extracto de la página de ayuda de la función `strel`. Recuperada de [www.mathworks.com](http://www.mathworks.com)





# REFERENCIAS

---

- [1] C. Tomé López, «El descubrimiento de los rayos X,» [En línea]. Available: <https://culturacientifica.com/2019/07/16/el-descubrimiento-de-los-rayos-x/>.
- [2] IMT Innovation, «Medical Imaging,» [En línea]. Available: <https://imtinnovation.com/digital-health/medical-imaging/>.
- [3] E. Hidalgo Salvador, F. Álvarez González, A. Salvador Luna, «Aplicación de la termografía infrarroja en medicina legal. ¿Prueba válida para la objetivación de los síndromes dolorosos? Disfunción temporomandibular,» [En línea]. Available: [http://scielo.isciii.es/scielo.php?script=sci\\_arttext&pid=S1135-76062014000200002](http://scielo.isciii.es/scielo.php?script=sci_arttext&pid=S1135-76062014000200002).
- [4] D. Turbert, E. M. Jiménez, «What is Hemangioma?,» [En línea]. Available: <https://www.aao.org/eye-health/diseases/hemangioma>.
- [5] Clínica Universidad de Navarra, «Dignóstico y Tratamiento de los Hemangiomas en la Clínica,» [En línea]. Available: <https://www.cun.es/enfermedades-tratamientos/enfermedades/hemangiomas>.
- [6] Instituto Nacional del Cáncer, «Diccionario de cáncer. Hemangioma infantil,» [En línea]. Available: <https://www.cancer.gov/espanol/publicaciones/diccionario/def/hemangioma-infantil>.
- [7] Fluke Corporation, «Cómo funcionan las cámaras infrarrojas,» [En línea]. Available: <https://www.fluke.com/es-es/informacion/mejores-practicas/aspectos-basicos-de-las-medidas/termografia/como-funcionan-las-camaras-infrarrojas>.
- [8] Joint Photographic Experts Group, «About JPEG,» [En línea]. Available: <https://jpeg.org/about.html>.
- [9] Mathworks, «MATLAB. Matemáticas. Gráficos. Programación.,» [En línea]. Available: <https://es.mathworks.com/products/matlab.html>.
- [10] Mathworks, «Documentación. Matrices y arreglos,» [En línea]. Available: [https://es.mathworks.com/help/matlab/learn\\_matlab/matrices-and-arrays.html](https://es.mathworks.com/help/matlab/learn_matlab/matrices-and-arrays.html).
- [11] File.org, «Opening BIN files,» [En línea]. Available: <https://file.org/extension/bin>.
- [12] Mathworks, «Documentación. Uint16,» [En línea]. Available: <https://es.mathworks.com/help/matlab/ref/uint16.html>.
- [13] Mathworks, «Documentación. Jet,» [En línea]. Available: <https://es.mathworks.com/help/matlab/ref/jet.html>.
- [14] B. Acha Piñero, C. Serrano Gotarredona, «Fundamentos de Procesamiento de Imagen,» *apuntes*, tema 7,

2016-2017.

- [15] B. Acha Piñero, C. Serrano Gotarredona, «Fundamentos de Procesamiento de Imagen,» *apuntes*, tema 7, 2016-2017.
- [16] B. Acha Piñero, C. Serrano Gotarredona, «Fundamentos de Procesamiento de Imagen,» *apuntes*, tema 3, 2016-2017.
- [17] Mathworks, «Documentación. Ajuste de contraste,» [En línea]. Available: <https://es.mathworks.com/help/images/contrast-adjustment.html>.
- [18] M. T. Angulo Carrere, A. Álvarez Méndez, Y. Fuentes Peñaranda, «Biomecánica clínica. Biomecánica de la Extremidad Superior. Exploración de la Mano,» [En línea]. Available: <http://www.revistareduca.es/index.php/reduca-enfermeria/article/viewFile/749/765>.
- [19] Clínica Cristo Rey, «¿Cuántos huesos hay en la mano?,» [En línea]. Available: [clinicacristorey.com/noticias/Huesos-de-la-mano.html](http://clinicacristorey.com/noticias/Huesos-de-la-mano.html).
- [20] A. C. Edens Hurst, David Zieve, «Klippel-Trenaunay syndrome,» [En línea]. Available: <https://medlineplus.gov/ency/article/000150.htm>.
- [21] M. A. Baños Bernad, «Diferenciación entre hemangioma plano y la mancha en vino de Oporto,» [En línea]. Available: <https://www.institutcataladelpcu.com/diferenciacion-del-hemangioma-plano-la-mancha-vino-oporto/>.

# Glosario

---

.BIN: Formato de archivo binario	3
.JPG/.JPEG: Formato de compresión de imágenes ( <i>Joint Photographic Experts Group</i> )	3
GUI: Interfaz Gráfica de Usuario ( <i>Graphical User Interface</i> )	4
HSV: Matiz, Saturación, Valor ( <i>Hue, Saturation, Value</i> )	4
IR: Infrarrojo ( <i>Infrared</i> )	xi
IRT: Infrared Thermography	xiii
RGB: Rojo, Verde, Azul ( <i>Red, Green, Blue</i> )	4
TAC: Tomografía Axial Computerizada	2
TIR: Termografía Infrarroja	xi