

CONESCAPANHONDURAS2025paper27.pdf

 Institute of Electrical and Electronics Engineers (IEEE)

Document Details

Submission ID

trn:oid:::14348:477772900

Submission Date

Jul 31, 2025, 11:53 PM CST

Download Date

Aug 12, 2025, 12:25 PM CST

File Name

CONESCAPANHONDURAS2025paper27.pdf

File Size

2.3 MB

5 Pages




2,638 Words

14,227 Characters

9% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Top Sources

- 8%  Internet sources
- 4%  Publications
- 0%  Submitted works (Student Papers)

Integrity Flags




0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Top Sources

- 8%  Internet sources
- 4%  Publications
- 0%  Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

| | | | |
|----|-------------|---|-----|
| 1 | Internet | research.edgehill.ac.uk | 1% |
| 2 | Internet | accedacris.ulpgc.es | 1% |
| 3 | Internet | www.coursehero.com | 1% |
| 4 | Internet | marineenergyjournal.org | <1% |
| 5 | Internet | kclpure.kcl.ac.uk | <1% |
| 6 | Internet | investigacion.unirioja.es | <1% |
| 7 | Publication | Graciela Ramirez-Alonso, Olanda Prieto-Ordaz, Roberto Lopez-Santillan, Manuel ... | <1% |
| 8 | Internet | inis.iaea.org | <1% |
| 9 | Internet | beei.org | <1% |
| 10 | Internet | researchberg.com | <1% |
| 11 | Internet | www.cotarai.org | <1% |

12

Internet

www.labdeurgencias.com.ar

<1%

13

Publication

Teixeira, Ana Filipa Pereira. "A Representação da Mulher em Cargos Relevantes d...

<1%

14

Internet

blog.csdn.net

<1%

Consola de Procesamiento de Imágenes Médicas Basado en Matlab

Abstract— Este artículo presenta el desarrollo de una consola de procesamiento de imágenes médicas en MATLAB, diseñada para mejorar y analizar imágenes con fines diagnósticos. La herramienta sigue un flujo de trabajo estructurado en cinco etapas: Carga de imagen, preprocesamiento, procesamiento, postprocesado y guardado de resultados. Se implementó técnicas como filtros de mediana, media y gaussiano para reducción de ruido, segmentación, comparación en tiempo real de histogramas e imágenes completa, mapas de color entre otros.

Keywords: Procesamiento de imágenes médicas, MATLAB, Filtros, Diagnóstico por imagen.

I. INTRODUCCIÓN

El uso de imágenes médicas es indispensable para el diagnóstico adecuado de distintas patologías y por ende es uno de los pilares de la medicina moderna. El desarrollo de tecnologías de imagen nace de la necesidad humana de examinar el interior del cuerpo humano de la manera menos invasiva posible.

Las imágenes médicas nacen a finales del siglo XIX, en 1895 Wilhelm Conrad Roentgen descubrió los Rayos X mientras experimentaba con rayos catódicos en tubos de alto vacío, creando por primera vez una impresión de la mano de su esposa sobre una placa sensible a estos rayos. Por este descubrimiento se le entregó el Premio Nobel de Física en el año 1901. En latinoamérica, México fue el primer país en experimentar con los Rayos X, en 1896 (menos de un año después del descubrimiento de Roentgen) el Dr. Antonio F. Lopez publicó los principios físicos de los Rayos X en el periódico El Estandarte.

El desarrollo tecnológico de la radiología y las imágenes médicas ha permitido a una enorme cantidad de científicos ser galardonados con Premios Nobel por sus aportaciones a la humanidad, algunos de ellos aparte del Profesor Roentgen son Henry Becquerel, Pierre y Marie Curie, Godfrey Hounsfield, Alan Cormack y más recientemente Paul Lauterbur y Raymond Damadian. A lo largo del tiempo se desarrolló nueva tecnología para obtener imágenes, se sustituyó el uso de mecanismos analógicos por los sistemas digitales más precisos, se redujo considerablemente el tiempo y radiación necesarios para obtener estas imágenes, reduciendo así la exposición del paciente, entre muchos más avances del área.

Sin embargo las imágenes médicas no son creadas únicamente a través de rayos X, en ellas se incluye el Ultrasonido, la Resonancia Magnética, la Tomografía por Emisión de Positrones (PET), Bioluminiscencia y Fluorescencia, entre otros métodos. Para el correcto diagnóstico de patologías, es necesario procesar las imágenes obtenidas a través de estos procesos, el equipo envía los datos crudos a la computadora, la cual a través de algoritmos puede modificar los valores de la imagen para obtener su mejor versión y reducir la necesidad de necesitar más exposiciones.

II. DESCRIPCIÓN DE LA CONSOLA DE PROCESAMIENTO

Esta consola surge como una oportunidad de explotar las herramientas disponibles en matlab desde una aproximación enfocada a la salud y optimizando los recursos disponibles. Para ello es necesario separar el proceso en ____ etapas, las cuales son:

A. Carga de imagen

A través de este proceso es posible ingresar una imagen al procesador, esta debe estar en el formato admitido por el sistema y tener un navegador que facilite al usuario la selección de dicho documento. Otro método de obtención de los datos de imagen puede ser directamente desde el sensor o cámara en tiempo real. En esta etapa también se deberá convertir el formato a uno operativo (Por ejemplo, de DICOM a PNG) y comprobar la validez de los datos ingresados.

B. Preprocesamiento

Una vez cargada, la imagen debe ser preprocesada para facilitar el manejo del usuario, el usuario tendrá a disposición los controles necesarios para reducir el ruido en imagen, convertir la imagen de color a escalas de grises (si fuere necesario) o elegir un solo canal para procesar, redimensionar o recortar la imagen, corregir el contraste o brillo, entre otros.

C. Procesamiento de imagen

Es la etapa principal del análisis, en esta se aplican algoritmos según el objetivo del usuario. Algunas de las funciones más comunes son la detección de bordes, la segmentación (por ejemplo la umbralización, clustering, watershed, etc), extraer características (como la textura, los contornos, las formas), detectar objetos o regiones de interés, análisis morfológico.

D. Postprocesamiento

Una vez realizado el procesamiento de la imagen y obtenidos los objetivos del usuario, es necesario realizar un postprocesamiento, en este se mejora, interpreta o refina los resultados del procesamiento, se prepara la imagen para la visualización o análisis. Algunas de las funciones son la superposición de resultados sobre la imagen original (como en el realce de bordes), eliminación de falsos positivos, agrupamiento o etiquetado de objetos, cálculo de métricas y la preparación de los datos para implementar procesos de machine learning.

E. Guardado de imagen (Output)

Es la etapa final, en esta se salvan los resultados de las etapas anteriores, es necesario guardar las imágenes en formatos que permitan su fácil visualización y futuras ediciones (comúnmente en PNG, TIFF y DICOM), también se debe ofrecer opciones para exportar los resultados de manera numérica (por ejemplo CSV, JSON o XLSX), generar reportes o visualizaciones, enviar los resultados a bases de datos o sistemas médicos.

III. DESARROLLO DEL PROCESADOR DE IMÁGENES MÉDICAS

El procesador de imágenes ofrece una herramienta interactiva e intuitiva para la visualización, manipulación y análisis básico de imágenes médicas en formatos comunes.

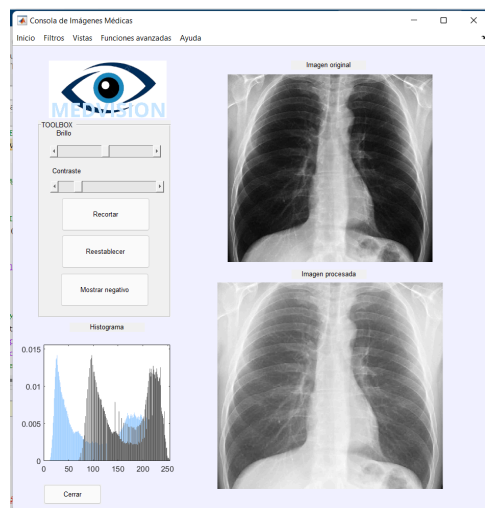


Fig. 1 Interfaz de usuario de la consola de procesamiento de imágenes

Está estructurado en cinco etapas principales: Carga, preprocesamiento, procesamiento, postprocesado y guardado. A continuación describimos las características de cada una:

A. Carga de imagen

Permite importar imágenes desde el sistema de archivos en múltiples formatos, comunes en la web: .png, .jpg, .jpeg, .webp, .bmp, .gif, .tiff. Al cargar la imagen se realiza una conversión automática a escala de grises para facilitar el posterior procesamiento, especialmente en las tareas de filtrado y segmentación, que trabajan de manera más eficiente en una sola banda de intensidad.

B. Preprocesamiento

Esta etapa prepara la imagen para su análisis, aplica mejoras visuales y reducción de ruido, estas son las herramientas más accesibles al usuario, pues son las más utilizadas. Incluye funciones como las siguientes:

- **Contraste con transformación gamma:**

Se realiza a partir de una curva de contraste gamma, ajusta la distribución de intensidades para ampliar el rango dinámico visible.

- **Ajuste de brillo:**

Permite al usuario modificar manualmente la luminosidad general de la imagen restando o sumando al valor de intensidad de los píxeles.

- **Visualización del negativo:**

Invierte las intensidades de los píxeles, permite resaltar estructuras que no son tan sencillas de visualizar en la vista original.

- **Filtros de suavizado:**

Estos filtros permiten reducir el ruido de alta frecuencia en las imágenes, se han implementado los siguientes:

- **Filtro de mediana:** Es útil para eliminar ruido impulsivo (sal y pimienta) pero reduce la precisión en los bordes.
- **Filtro de media:** Suaviza la imagen promediando los píxeles vecinos según el kernel elegido por el usuario.
- **Filtro gaussiano:** Suaviza con mayor control sobre la dispersión (sigma), permite mantener los bordes y reduce la distorsión en la imagen.

C. Procesamiento

Esta es la etapa central, donde se aplican algoritmos que extraen o resaltan la información relevante de la imagen:

- **Detección y realce de bordes:**

Es útil para identificar las estructuras anatómicas a través de las transiciones bruscas de las intensidades de los píxeles.

- **Segmentación:**

Se utiliza para realzar las diferentes estructuras de la imagen, binariza la imagen según un umbral, los procesos integrados son:

- **Umbralización con Otsu:** Es una segmentación automática que elige el mejor umbral para la detección.
- **Umbral personalizado:** Permite al usuario establecer un umbral específico para la binarización.

- **Mapas de falso color:**

Es una transformación visual que aplica paletas de color para mejorar la interpretación de las estructuras anatómicas, en este proyecto se integraron los mapas JET, COLD, HOT y BONE.

D. Postprocesado

Para análisis, presentación de los resultados, además de posibles calibraciones se incluyen:

- **Detección de bordes aislados:**

Permite visualizar de manera aislada los bordes que el programa ha identificado en la imagen, se utiliza para corroborar el funcionamiento de la herramienta.

- **Generación de ruido gaussiano:**

Útil para simular condiciones adversas y evaluar la robustez de los filtros y algoritmos aplicados, de igual manera funciona para calibrar el sistema.

- **Vista comparativa:**

A medida que el usuario modifica la imagen original, se despliega la imagen modificada y en el histograma es posible observar la diferencia entre el histograma original y el modificado, de esta manera se evidencian los cambios aplicados.

E. Guardado

Esta etapa permite guardar los cambios realizados a la imagen original, de esta manera las imágenes pueden compartirse para ser visualizadas y analizadas con mayor precisión. La imagen se guarda en formato .png para reducir la compresión.

IV. DESCRIPCIÓN DE LA IMPLEMENTACIÓN DEL CÓDIGO Y LAS HERRAMIENTAS UTILIZADAS.

A. Carga de imagen

A través de la función `uigetfile()` es posible abrir el navegador de archivos, dentro de esta función se debe especificar los formatos admitidos y el texto de la ventana.

Con la función `fullfile()` extraemos la ruta completa del archivo para cargarla en matlab desde una carpeta diferente a la del `.m` donde está alojado el programa.

Asignamos el nombre "im" a la imagen leída por la función `imread()`, convertimos los valores de la imagen a valores de 8 bits con la función `uint8()`.

Además, en caso de ser una imagen a color, la convertimos a escala de grises utilizando la siguiente sentencia:

```
if size(im,3) == 3
    im = rgb2gray(im)
```

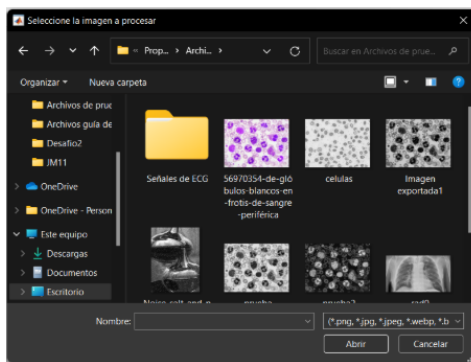


Fig. 2 Interfaz de carga de imágenes

B. Histograma y actualización

El histograma original se presenta automáticamente al momento de abrir la imagen a procesar y se ancla al fondo del gráfico, sobre él se presenta el histograma de la imagen procesada.

Con cada modificación a la imagen se actualiza el histograma modificado y es posible compararlos en tiempo real con el original.

C. Recorte:

Permite recortar una parte de la imagen que se convierte en nuestra región de interés ROI, de esa manera podemos deshacernos de partes innecesarias de la imagen cargada.

Se abre una figura nueva donde se solicita elegir una sección para recortar una función `getrect()` marcando un rectángulo con el mouse sobre la imagen procesada para poder recortarla, luego de ello se actualiza la imagen procesada utilizando la función `imcrop()`.

D. Contraste:

Se utiliza una transformación gamma a la imagen procesada, el valor de transformación se obtiene a través de un slider y se aplica la fórmula de transformación.

TABLA I
COMPARACIÓN EN CONTRASTES

| Niveles de contraste | | |
|----------------------|----------------|----------------|
| Contraste original | Contraste bajo | Contraste alto |
| | | |

E. Brillo

En esta función se aplica una modificación a los valores de intensidad de la imagen a través de barridos con funciones `For` anidadas, el valor del brillo se obtiene a través de un slider.

TABLA II
COMPARACIÓN EN BRILLO

| Niveles de brillo | | |
|-------------------|-------------|-------------|
| Brillo original | Brillo bajo | Brillo alto |
| | | |

F. Negativo

Invierte los valores de intensidad de la imagen, para este proceso se establece que el nuevo valor de los píxeles será 255 menos el valor de cada píxel.

TABLA III
COMPARACIÓN CON NEGATIVO

| Original | Negativo |
|----------|----------|
| | |

G. Restablecer

Carga nuevamente la imagen original para revertir todos los cambios realizados.





H. Guardar

Utiliza la función `uiputfile()`, permite elegir el formato de la imagen guardada entre `.png`, `.jpg` y `.bmp`.

I. Filtro de media

Genera una ventana de texto donde solicita el tamaño del kernel para evaluar cada píxel utilizando, si el usuario no especifica un valor, automáticamente aplica un kernel de 3.




TABLA IV
COMPARACIÓN DE KERNELS DE FILTRO DE MEDIA

| Resultado de los diferentes tamaños de kernel | | | |
|---|---|---|---|
| Imagen original | Kernel 3x3 | Kernel 5x5 | Kernel 7x7 |
|  |  |  |  |

J. Filtro de mediana

Genera una ventana popup donde solicita el tamaño del kernel, una vez establecido se utiliza la función medfilt().





TABLA V
COMPARACIÓN DE KERNELS DE FILTRO DE MEDIANA

| Resultado de distintos tamaños de Kernel | | |
|---|---|---|
| Imagen original | Kernel 3x3 | Kernel 5x5 |
|  |  |  |

K. Filtro Gaussiano

Se crea un popup donde solicita el tamaño del kernel y el valor sigma para el filtro, si el usuario no modifica los textbox utilizará los valores predeterminados 3 y 0,5 respectivamente.

TABLA VI
COMPARACIÓN EN KERNELS Y SIGMA EN FILTRO GAUSSIANO

| Resultado de los diferentes configuraciones de sigma y tamaño de kernel | | | |
|---|---|---|---|
| Imagen original | Kernel 3x3 sigma 0.8 | Kernel 5x5, sigma 1.0 | Kernel 7x7, sigma 1.5 |
|  |  |  |  |

Se utiliza la función fspecial() y se especifica que será un filtro gaussiano con la sentencia 'gaussian', aquí mismo se define el tamaño del kernel y el valor de sigma asignado al filtro en el popup anterior.


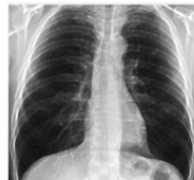
L. Detección y realce de bordes

Se utilizan diferentes máscaras para detectar el cambio brusco de intensidad en las intensidades, las máscaras son las siguientes:

```
sobel_lr = [1 0 -1; 2 0 -2; 1 0 -1];  
sobel_rl = [-1 0 1; -2 0 2; -1 0 1];  
sobel_tb = [1 2 1; 0 0 0; -1 -2 -1];  
sobel_bt = [-1 -2 -1; 0 0 0; 1 2 1];
```

Para cada una de las máscaras se aplica una convolución utilizando la función conv2(), se suman los valores absolutos de los bordes, se normalizan los resultados y se sobreponen en la imagen modificada.



TABLA VII
REALCE DE BORDES

| Realce de bordes | |
|--|---|
| Imagen original | Bordes realzados |
|  |  |

M. Segmentación

Se implementa una función de segmentación con la función strtrim(), el valor del umbral se obtiene con un popup preguntando al usuario el valor que desea utilizar, si el usuario no modifica el valor en la textbox, entonces utilizará el valor Otsu para la imagen.


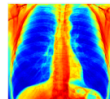
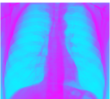
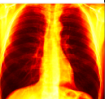
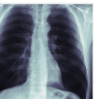
TABLA VIII
RESULTADOS DE BINARIZACIÓN

| Binarización | |
|---|--|
| Imagen original | Segmentación |
|  |  |

N. Mapas de falso color

Se utilizaron 4 mapas, se disponen en un sistema switch-case, se utiliza la función ind2rgb(), se especifica que se aplicará sobre la imagen procesada y se define el mapa a implementar a través de los casos. Si se vuelve a pulsar en el mismo case, entonces el mapa de color se desactivará.

TABLA IX
COMPARACIÓN EN FALSO COLOR

| Mapas de color | | | | |
|---|--|---|---|---|
| Imagen original | JET | Cool | Hot | Bone |
|  |  |  |  |  |

O. Bordes aislados

Se utilizan las máscaras definidas anteriormente, pero esta vez no se sobreponen en la imagen modificada sino sobre un fondo negro, esto permite visualizar de manera más precisa las estructuras.

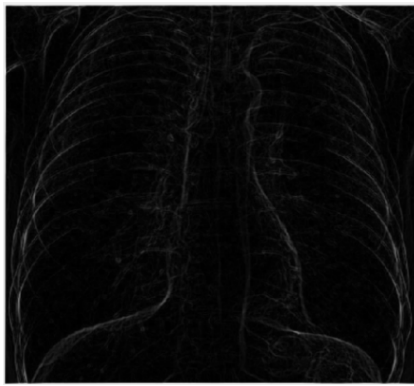


Fig. 3 Bordes aislados.

P. Agregar ruido gaussiano

Permite agregar ruido gaussiano para probar los filtros implementados y calibrarlos, abre un popup donde solicita la varianza del ruido (que define la intensidad del ruido) y el factor (que define el tamaño del grano), luego genera una imagen de ruido y la sobrepone a la imagen procesada.

TABLA X

COMPARACIÓN EN FACTOR Y VARIANZA EN RUIDO GUASSIANO

| Opciones de ruido gaussiano | | | |
|-----------------------------|-------------------------|-------------------------|-------------------------|
| Imagen original | Factor 3, varianza 0.02 | Factor 5, varianza 0.06 | Factor 8, varianza 0.10 |
| | | | |

Q. Visualización de mapa 3D

A partir de la imagen procesada, se genera una superficie 3D usando la función Surf, el eje z representa la intensidad de la imagen, se aplica un mapa de color JET para visualizar de mejor manera las diferencias en las intensidades. Permite rotar la imagen, guardar la figura en 3D e interactuar marcando puntos especiales en la figura.

La visualización en 3D de las estructuras podría permitir al usuario identificar de mejor manera las estructuras presentes en la imagen, facilitando el diagnóstico.

TABLA XI

COMPARACIÓN EN IMAGEN Y MAPA 3D

| Mapeo 3D | |
|-----------------|---------|
| Imagen original | Mapa 3D |
| | |

IV. CONCLUSIONES

El procesamiento de imágenes médicas es una herramienta fundamental para el diagnóstico clínico, permite mejorar la calidad visual, extraer características relevantes y facilitar el análisis de estructuras anatómicas. En esta investigación se desarrolló una consola de procesamiento de imágenes médicas basada en MATLAB, estructurada en cinco etapas principales (carga, preprocesamiento, procesamiento, postprocesado y guardado).

Los resultados demuestran que la consola es eficaz en la manipulación y realice de imágenes médicas, facilitando tareas como la reducción de artefactos, la identificación de regiones de interés y la visualización de estructuras mediante técnicas avanzadas. Futuras mejoras podrían incluir la integración de algoritmos de deep learning para automatizar procesos de diagnóstico y la expansión de formatos compatibles, como DICOM, para una mejor aplicabilidad en entornos clínicos.

REFERENCIAS

- [1] K. Doi, "Diagnostic imaging over the last 50 years: research and development in medical imaging science and technology," *Physics in Medicine and Biology*, vol. 51, no. 13, pp. R5–R27, Jun. 2006, doi: 10.1088/0031-9155/51/13/r02.
- [2] J. L. Semmlow, *Biosignal and medical image processing*. 2008. doi: 10.1201/9780203024058.
- [3] R. Koprowski, "Hyperspectral imaging in medicine: image pre-processing problems and solutions in Matlab," *Journal of Biophotonics*, vol. 8, no. 11–12, pp. 935–943, Feb. 2015, doi: 10.1002/jbio.201400133.
- [4] Q. Deng and X. Chen, "An image processing toolbox based on MATLAB GUI," *Journal of Physics Conference Series*, vol. 2010, no. 1, p. 012137, Sep. 2021, doi: 10.1088/1742-6596/2010/1/012137.
- [5] E. B. Magrab, *An Engineer's Guide to MATLAB: With Applications from Mechanical, Aerospace, Electrical, Civil, and Biological Systems Engineering*. Pearson, 2011.