



Casa abierta al tiempo

Universidad Autónoma Metropolitana

Posgrado en Ciencias y Tecnologías de la Información

División de Ciencias Básicas e Ingeniería

Unidad Iztapalapa

“IDENTIFICACIÓN Y LOCALIZACIÓN DE CÉLULAS GLIALES
EN IMÁGENES DE TEJIDO CEREBRAL ”

Tesis que presenta

Lic. Karla Mandujano López

Matrícula

2183802193

Para obtener el grado de

Maestro en Ciencias

(Ciencias y Tecnología de la información)

Directores

Dra. Graciela Román Alonso

Dr. Juan Villegas Cortez

Jurado

Presidente Dr. Leonardo Trujillo Reyes

Secretario Dr. Juan Villegas Cortez

Vocal Dr. Rene Mackinney Romero

Unidad Iztapalapa

Universidad Autónoma Metropolitana

División de Ciencias Básicas e Ingeniería

Ciudad de México, Iztapalapa, noviembre del 2022



Universidad Autónoma Metropolitana

Posgrado en Ciencias y Tecnologías de la Información

Identificación y Localización de Imágenes de Tejido Cerebral

Tesis que presenta

Karla Mandujano López

Para obtener el grado de

Maestro en Ciencias

(Ciencias y Tecnología de la información)

Asesores

Dr. Juan Villegas Cortez

Dra. Graciela Román Alonso

Unidad Iztapalapa,

Universidad Autónoma Metropolitana

División de Ciencias Básicas e Ingeniería

Ciudad de México, 21 de noviembre de 2022

Karla Mandujano López
Universidad Autónoma Metropolitana

IDENTIFICACIÓN Y LOCALIZACIÓN DE
CÉLULAS GLIALES EN IMÁGENES DE
TEJIDO CEREBRAL

*Para mi familia
Sin la cual este trabajo
no habría sido posible.*

RESUMEN

Se propone un sistema de procesamiento digital de imágenes para la identificación automática de células gliales de tejido cerebral. El propósito es apoyar al especialista, que al seleccionar una región de interés (ROI) en una imagen de tejido cerebral, se procede a buscar esa ROI en toda la imagen en cuestión. La metodología propuesta trabaja a nivel textura por lo que primero aplica un preprocesamiento a toda la imagen: se extraen atributos CBIR (Content-based Image Retrieval) probados en la literatura, dando como resultado la posibilidad de analizar de forma local a una imagen específica para localizar regiones similares en la misma imagen. Nuestra propuesta hace uso de una discriminación de los patrones conformados usando métrica euclidiana, tras implementar el método *K*-Means para una auto-organización de los patrones de textura conformados. Se han realizado pruebas con un total de 312 imágenes experimentales de laboratorio obtenidas de un microscopio Zeiss Axyo Manager en formato TIFF, y se han obtenido resultados satisfactorios bajo la supervisión del especialista en biología celular. Adicionalmente, con el fin de acelerar el tiempo de análisis de un conjunto grande de imágenes, el sistema propuesto integra técnicas de paralelismo para aprovechar la capacidad de cómputo de un sistema multiprocesador.

Palabras Clave: Células gliales, Procesamiento Digital de Imágenes, Clasificación, CBIR, K-NN, Reconocimiento de Patrones

ABSTRACT

A digital image processing system is proposed for the automatic identification of glial cells of brain tissue. The purpose of this approach is to support the specialist, so that when selecting a region of interest (ROI) on a brain tissue image, proceed to search for that ROI in the entire image in question. The proposed methodology works at the texture level, so it first applies a preprocessing to the entire image: CBIR (Content-based Image Retrieval) attributes found in the literature, resulting in the possibility of analyzing locally the texture of a specific image to locate similar regions in the same image. Our proposal applies a discrimination of the conformed patterns using Euclidean metrics, after implementing the *K*-Means method for a self-organization of the conformed texture patterns. Tests have been carried out with a total of 312 laboratory images, purely experimental obtained from microscope Zeiss Axyo Manager in TIFF file format, and promising results have been obtained under the supervision of the specialist in cell biology. Additionally, in order to speed up the analysis time of a large set of images, the proposed system integrates parallelism techniques to take advantage of the computational capacity of a multiprocessor system.

Keywords: Glial cells, Digital Image Processing, Classification, CBIR, K-NN, Pattern Recognition

CONTENIDO

Contenido	IX
Lista de figuras	XI
Lista de tablas	XIII
Lista de algoritmos	XV
Lista de acrónimos	XVII
Dependencias lógicas de capítulos	XIX
1 Introducción	1
1.1 Descripción de la problemática a tratar	1
1.1.1 Definición del problema	1
1.1.2 Pregunta de investigación fundamental	2
1.1.3 Hipótesis	2
1.1.4 Contribución.	2
1.1.5 Justificación	3
1.2 Objetivos	3
1.2.1 Objetivo general.	3
1.2.2 Objetivos particulares	3
2 Antecedentes	5
2.1 Marco teórico.	5
2.1.1 Células gliales	5
2.1.2 Procesamiento digital de imágenes.	9
2.2 Estado del arte.	12
2.2.1 Escaneo Cerebral	14
3 Propuesta	19
3.1 Metodología general.	19
3.2 <i>Integración de paralelismo</i>	25
3.3 Algoritmo paralelo.	26
4 Experimentos	27
5 Resultados	31
6 Descripción del sistema	35

Contenido	X
7 Conclusiones y Trabajo a futuro	41
8 Glosario	43
Bibliografía	45
Anexo 1	47
Anexo 2	61

LISTA DE FIGURAS

Figura 2.1	Fotografía tomada por él microscopio la cual muestra una célula glial.	6
Figura 2.2	Tipos de células gliales.	7
Figura 2.3	Paradigma de la Imagen Digital.	9
Figura 2.4	Pasos fundamentales en el procesamiento de imágenes digitales [14].	11
Figura 2.5	Contador de hematología automatizado.	13
Figura 2.6	Tomografía por emisión de positrones.	14
Figura 2.7	Tomografía por emisión de positrones.	15
Figura 2.8	Tomografía por emisión de positrones.	16
Figura 3.1	Metodología general de un sistema de visión y reconocimiento de imágenes.	19
Figura 3.2	Metodología propuesta para análisis por textura de imágenes de células gliales.	22
Figura 3.3	Sembrado de puntos en la imagen.	23
Figura 3.4	Construcción de ventana de 10x10 pixeles.	23
Figura 3.5	Propuesta de arquitectura paralela para implementar de forma masiva la metodología general.	25
Figura 4.1	Ejemplo de una célula glial en campo claro.	28
Figura 4.2	Ejemplo de una célula glial en fluorescencia.	28
Figura 4.3	Imagen celular con la ROI seleccionada por el usuario y las tres ROI de tamaño 10x10 más parecidas, considerando un emparrillado regular de la imagen.	29
Figura 4.4	Imagen celular con la ROI seleccionada por el usuario y las tres ROI de tamaño 10×10 más parecidas, considerando un emparrillado aleatorio de la imagen.	29
Figura 4.5	Imagen celular con la ROI seleccionada por el usuario y las tres ROI de tamaño 30×30 más parecidas, considerando un emparrillado aleatorio de la imagen.	30
Figura 5.1	Proporción de las imágenes vs ROI vs ventanas.	32
Figura 5.2	Imagen celular con la ROI seleccionada aleatoriamente y las cinco ROI de tamaño más parecidas, considerando un emparrillado aleatorio donde delimita un objeto	32
Figura 5.3	Imagen celular con la ROI seleccionada aleatoriamente y las cinco ROI de tamaño más parecidas, considerando un emparrillado aleatorio donde delimita varios objetos.	33
Figura 5.4	Comparación por tiempo en procesos.	34
Figura 5.5	Comparación de secuencial vs paralelismo	34

Figura 6.1	Carpeta de proyecto.	35
Figura 6.2	Carpeta de <i>Ejecutable</i>	36
Figura 6.3	Archivo <i>Ejecutable</i>	36
Figura 6.4	Carpeta 'Img_prueba'.	36
Figura 6.5	Imagen a analizar.	37
Figura 6.6	Botón <i>Abrir</i>	37
Figura 6.7	Imagen seleccionada.	38
Figura 6.8	Seleccionar objeto.	38
Figura 6.9	Objeto recortado.	39
Figura 6.10	Cerrar objeto recortado.	39
Figura 6.11	Cerrar ventana de la imagen seleccionada.	40
Figura 6.12	Resultado.	40

LISTA DE TABLAS

Tabla 4.1	Resolución de imágenes.	27
Tabla 5.1	Tiempo que tardó el algoritmo paralelo con los diferentes procesos. .	33

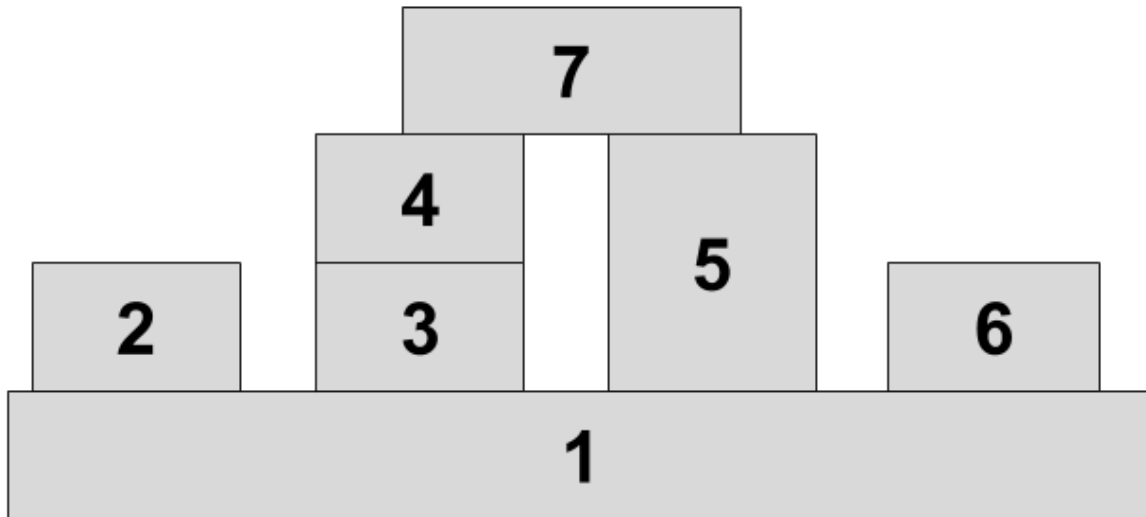
LISTA DE ALGORITMOS

Algoritmo 3.1	ejecutado simultáneamente por cada proceso i	26
---------------	--	----

LISTA DE ACRÓNIMOS

IA	Inteligencia Artificial.....	4
CBIR	Content-based Image Retrieval.....	43
K-NN	K-Nearest-Neighbor	34
HSI	Hue, Saturation, Lightness	44
RGB	Red, Green, Blue	
DB	Database	
CAD	Computer-Aided Design	12

DEPENDENCIAS LÓGICAS DE CAPÍTULOS



CAPÍTULOS

Capítulo 1	Introducción
Capítulo 2	Antecedentes
Capítulo 3	Propuesta
Capítulo 4	Experimentos
Capítulo 5	Resultados
Capítulo 6	Descripción del sistema
Capítulo 7	Conclusiones y Trabajo a futuro

1

INTRODUCCIÓN

1.1 Descripción de la problemática a tratar

Para nosotros como seres humanos, seres inteligentes y dinámicos, el sentido de la vista es el más común que usamos, prácticamente pasa desapercibido que de los cinco sentidos que tenemos actualmente sea el más explotado [21]; un ejemplo de lo refinado que es nuestro sentido de la vista es el “reconocimiento automático” que hacemos de imágenes, ya sea en tipo, clasificación o la identificación de objetos [25].

Actualmente se tiene el problema de identificación de objetos, o puntos, en imágenes de tejidos celulares; estas muestras son adquiridas para poder diagnosticar enfermedades en pacientes acorde con un propósito médico. Para cada paciente se tiene que hacer el análisis visual de muestras de tejidos por parte de un médico especializado, o personal debidamente capacitado, pero siendo una tarea repetitiva y con particularidades para cada persona, es que no es posible que una sola persona haga un correcto análisis visual de dichas muestras de tejido [25].

La glía es un grupo heterogéneo de células neuronales que cumplen funciones diversas en la fisiología del cerebro. Con base en su morfología, fisiología y localización en el cerebro podemos identificar los siguientes principales tipos de glía [7]: microglía, oligodendroglía, glía NG2 y Astroglía.

1.1.1 Definición del problema

La búsqueda de patrones en células gliales que advierten a tener una enfermedad es complicada, ya que solo personas especializadas lo pueden hacer, esto conlleva mucho tiempo y además aquí también influye el sentido de la vista, ya que esto se observa a través de un microscopio y el ojo humano no puede alcanzar a apreciar todo. Por lo

anteriormente expuesto, es deseable poder contar con un sistema automatizado, instalado en un equipo de cómputo, que permita hacer el análisis visual de un tipo de muestra específico, y muestre un análisis primario de la imagen a la persona que la supervise, y en corto tiempo ayude a realizar la detección de objetos deseada sobre las muestras. [4]

1.1.2 Pregunta de investigación fundamental

- ¿Se podría a través de la Inteligencia Artificial (IA) apoyar el análisis de imágenes de tejido cerebral a nivel celular, para la localización y/o identificación de células gliales para un mejor diagnóstico médico?
- ¿En las células gliales es posible encontrar patrones que conduzcan a una enfermedad con ayuda del Procesamiento Digital de Imágenes?

1.1.3 Hipótesis

Al llevarse a cabo el procesamiento de imágenes digitales de tejido cerebral, por medio del estudio de la textura de la imagen, seremos capaces dar un análisis preliminar del contenido de la imagen a partir de ella misma, con base al criterio visual de un especialista de la salud.

1.1.4 Contribución

Se proporciona un análisis automático del contenido de la imagen de tejido cerebral desde la perspectiva de la textura, superando las características de color que pueden ser alteradas por los reactivos químicos del preparado de las muestras, contribuyendo con una nueva metodología para el campo de estudio de la imagenología médica de este tipo de imágenes.

1.1.5 Justificación

Por definición, las enfermedades neurodegenerativas comprenden tanto la pérdida progresiva e irreversible de neuronas; así como la proliferación descontrolada de algunos otros tipos celulares en el caso de cáncer. Todo esto conlleva a una pérdida del control óptimo de aquellas funciones reguladas a nivel nervioso.

La determinación microscópica de muestras de pacientes con padecimientos neurodegenerativos sigue siendo considerada una herramienta muy útil para el diagnóstico y evolución de estas enfermedades. Bajo técnicas diversas de histoquímica e inmunofluorescencia contra marcadores específicos es posible obtener un estimado en biopsias de estos pacientes sobre la disminución en el número de neuronas o la proliferación descontrolada de oligodendrocitos o astrocitos en casos de tumores cerebrales.

1.2 Objetivos

1.2.1 Objetivo general

Desarrollar una metodología computacional que realice el análisis de la imagen de tejido cerebral para identificar y localizar células gliales.

1.2.2 Objetivos particulares

- Hacer un preprocesamiento adecuado de las imágenes de tejido cerebral (campo claro y fluorescencia).
- Extraer las características de las imágenes aplicando técnicas CBIR¹ de textura.
- Generar un clasificador de distancia que detecte una región de interés.
- Desarrollar un sistema de conteo de células, basado en la caracterización desde su textura.

¹CBIR es un sistema que tiene dos etapas principales: la inserción de datos y el procesamiento de consultas. Se define en el estado del arte.

2

ANTECEDENTES

2.1 Marco teórico

2.1.1 Células gliales

El cerebro está conformado por neuronas y células gliales, la relación entre estos dos tipos de células es fundamental para mantener la homeostasis cerebral [7]. Las neuronas tienen la habilidad de comunicarse entre sí de manera rápida y eficiente a través de señales eléctricas o químicas que se traducen en forma de potenciales de acción. Todas las células neurales que no producen potenciales de acción están agrupadas en lo que se conoce como glía [5, 7].

La glía es el grupo de células del sistema nerviosos más abundante en el cerebro. Sin embargo, durante mucho tiempo se le consideró sólo como un elemento de soporte neuronal, que no cumplía ninguna función importante. Hoy en día se sabe que la glía participa en la formación, operación y modulación de los circuitos sinápticos [5, 23]. En la Figura 2.1 se observa una célula glial, con sus diferentes núcleos.

Durante medio siglo, se creía que el cerebro humano contenía unos 100 mil millones de neuronas y un mil millones de células gliales, con una proporción de glía: neurona de 10:1. Un nuevo método de conteo, el fraccionador isotrópico, ha desafiado la idea de que la glía supera en número a las neuronas y revivió una pregunta que se pensaba que se había resuelto [5].

El fraccionador isotrópico recientemente validado demuestra una relación glía: neurona de menos de 1: 1 y un número total de menos de 100 mil millones de células gliales en el cerebro humano. Una encuesta de evidencia original muestra que los datos histológicos siempre apoyaron una proporción 1: 1 de glía a neuronas en todo el cerebro humano, y un rango de 40-130 mil millones de células gliales [5].

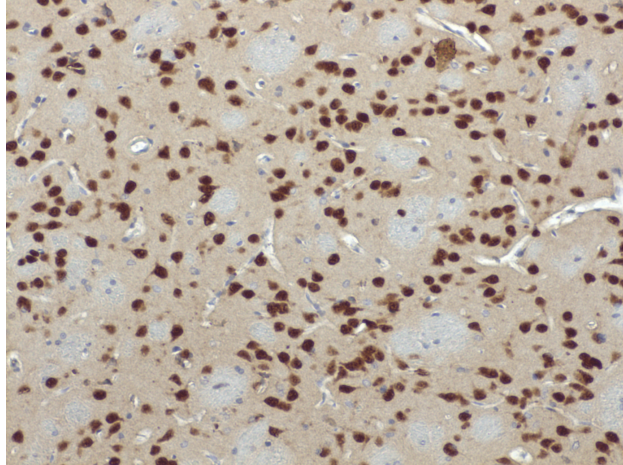


Figura 2.1: Fotografía tomada por el microscopio la cual muestra una célula glial.

La excitabilidad de las células nerviosas depende en gran medida del ácido glutámico, el principal neurotransmisor excitador en el cerebro de mamíferos. La síntesis y el metabolismo del ácido glutámico, involucra una estrecha relación entre las neuronas y las células gliales. Una alteración entre los sistemas neuro-gliales glutamatergicos puede producir la muerte de las células nerviosas a través de un mecanismo denominado excitotóxico [7, 23].

La muerte excitotóxica se ha asociado a la pérdida neuronal que se observa después de un episodio isquémico y en algunas enfermedades neurodegenerativas. Por otro lado, las interacciones neurogliales pueden funcionar como sistemas de neuroprotección que se activan para contrarrestar el daño excitotóxico a través de una intensa interacción metabólica que involucra la estimulación de la glucólisis y la producción de sustratos metabólicos como el piruvato y el lactato, así como la síntesis de glutatión, entre otros [7, 23].

La glía es un grupo heterogéneo de células neurales como se muestra en la Figura 2.2 que cumplen funciones diversas en la fisiología del cerebro. Con base en su morfología, fisiología y localización en el cerebro podemos identificar los siguientes principales tipos de glía [2, 23]:

- Microglía,
- Oligodendroglía,
- Glía NG2,
- Astroglía.

La microglía se encarga de vigilar que el cerebro conserve su integridad, al reaccionar de manera inmediata ante cualquier daño que se produzca. En caso de infección, la microglía combate a los organismos nocivos, fagocitándolos y removiendo también las células muertas. La microglía, incluso, puede participar en la remodelación sináptica durante el

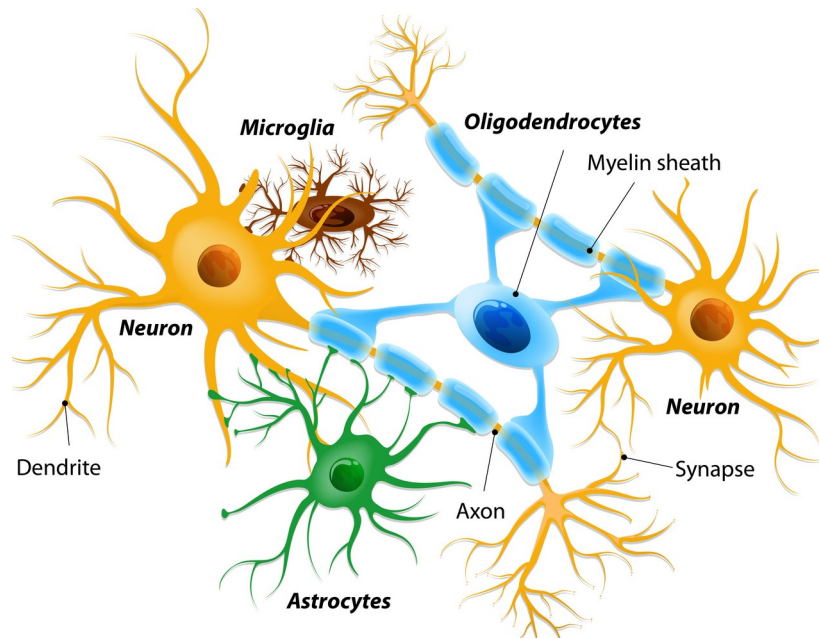


Figura 2.2: Tipos de células gliales.

desarrollo del sistema nervioso central, al remover conexiones inapropiadas. Adicionalmente, la microglía se vuelve activa en enfermedades neurodegenerativas [2, 23].

El papel de la Oligodendroglía es facilitar la comunicación eléctrica entre las neuronas. Comprende a los oligodendrocitos, que se ubican en el sistema nervioso central y las células de Schwann, que están presentes en el sistema nervioso periférico. Ambos tipos gliales producen mielina, una lipoproteína que envuelve a los axones de las neuronas y hace más eficiente la comunicación neuronal, al acelerar la conducción eléctrica de los impulsos nerviosos [5, 23].

Las moléculas de lípidos son componentes clave de la compleja estructura y función del cerebro, ya que los lípidos representan alrededor del 50 % del peso seco del cerebro. Se ha demostrado que la composición lipídica de las membranas de las células neuronales y gliales afecta la función celular y la neurotransmisión [2].

Los oligodendrocitos contribuyen al soporte metabólico necesario para el adecuado funcionamiento axonal. Cuando se enferman, se presenta un déficit metabólico y en la producción de mielina. Las consecuencias de la desmielinización y el déficit metabólico de la oligodendroglía produce problemas cognitivos y motores, como los que se presentan en enfermedades neurodegenerativas como la esclerosis múltiple o las leucodistrofias [2, 23].

La astrogliá comprende a los astrocitos, las células endimales y la glía radial. La palabra astrocito significa “célula en forma de estrella” [23]. Los astrocitos, las células gliales más grandes y más numerosas del sistema nervioso central, desempeñan una variedad de funciones importantes en la regulación de la homeostasis, el aumento de la plasticidad

sináptica y el suministro de neuroprotección, lo que ayuda a mantener la función normal del cerebro [9, 23].

Al mismo tiempo, los astrocitos pueden participar en la respuesta inflamatoria y desempeñar un papel clave en la progresión de las enfermedades neurodegenerativas. Los astrocitos reactivos están fuertemente inducidos por numerosas condiciones patológicas en el SNC. La reactividad de los astrocitos se caracteriza inicialmente por la hipertrofia del soma y los procesos, desencadenados por diferentes moléculas. Cualquier desregulación de estos mecanismos es una condición común para las enfermedades neurodegenerativas [9, 23].

Una gran cantidad de estudios en modelos de ratones han demostrado que los astrocitos desempeñan un papel complejo en la patogénesis de las enfermedades neurodegenerativas, y la disfunción de los astrocitos puede contribuir a la muerte neuronal o al proceso de trastornos neuronales [9].

Se ha encontrado que los astrocitos reactivos siempre pierden su papel de apoyo y obtienen una función tóxica en la progresión de las enfermedades neurodegenerativas. Los astrocitos reactivos responden a las lesiones en el SNC durante el proceso de trastornos neurológicos causados por enfermedades neurodegenerativas, como la enfermedad de Alzheimer, Parkinson, esclerosis lateral amiotrófica y esclerosis múltiple [9, 23].

El trastorno bipolar (TB) es una enfermedad neurobiológica compleja. Es probable que tanto las neuronas como las células gliales se vean afectadas en el TB, pero la forma en que estos tipos de células cambian a nivel estructural y funcional aún es en gran parte desconocida. Numerosos estudios postmortem han investigado cambios celulares estructurales en el TB. Sin embargo, dado que estos estudios muestran una gran variedad en términos de preguntas de investigación, región del cerebro estudiada, tamaño de la muestra, metodología y resultado, sigue siendo difícil llegar a una conclusión directa basada en sus resultados [10].

Hoy en día, la recuperación de imágenes es un campo de investigación muy activo porque la mayoría de la información en Internet son imágenes, aproximadamente 73 %. Vale la pena mencionar que la información relacionada con las imágenes no está bien organizada. La mayoría de las técnicas y sistemas conocidos para la recuperación de imágenes utilizan algún método para agregar metadatos como subtítulos, palabras clave o descripciones a las imágenes, de modo que la recuperación se puede realizar con las palabras de anotación. Sin embargo, la anotación manual de la imagen es lenta, laboriosa y costosa [17].

Durante la última década, los aumentos dramáticos en el poder computacional y la mejora en los algoritmos de análisis de imágenes han permitido el desarrollo de poderosos enfoques analíticos asistidos por computadora para los datos radiológicos. Con la reciente aparición de escáneres digitales de diapositivas completas, las diapositivas de histopatología de tejidos ahora se pueden digitalizar y almacenar en forma de imagen digital [19].

En consecuencia, la histopatología del tejido digitalizado se ha convertido ahora en

una cuestión de aplicación de análisis de imágenes computarizados y técnicas de aprendizaje automático. Análogamente al papel de los algoritmos de diagnóstico asistido por computadora (CAD- Computer-aided Design) en imágenes médicas para complementar la opinión de un radiólogo, los algoritmos CAD han comenzado a desarrollarse para la detección de enfermedades, el diagnóstico y la predicción del pronóstico para complementar la opinión del patólogo [19].

2.1.2 Procesamiento digital de imágenes

El interés en los métodos de procesamiento de imágenes digitales se deriva de dos áreas principales: mejora de la información gráfica para la interpretación humana; y procesamiento de datos de imagen para almacenamiento, transmisión y representación para la percepción autónoma de la máquina ver Figura 3.1. Una imagen puede definirse como una función bidimensional, $f(x, y)$, donde x e y son coordenadas y la amplitud de f en cualquier par de coordenadas (x, y) se denomina intensidad o gris nivel de la imagen en ese punto. Cuando x , y , y los valores de amplitud de f son todas cantidades finitas y discretas, llamamos a la imagen una imagen digital [17].

El campo del procesamiento de imágenes digitales se refiere al procesamiento de imágenes digitales por medio de una computadora digital. Una imagen digital está compuesta por un número finito de elementos, cada uno de los cuales tiene una ubicación y un valor en particular, estos elementos se denominan elementos de imagen o píxeles [14].

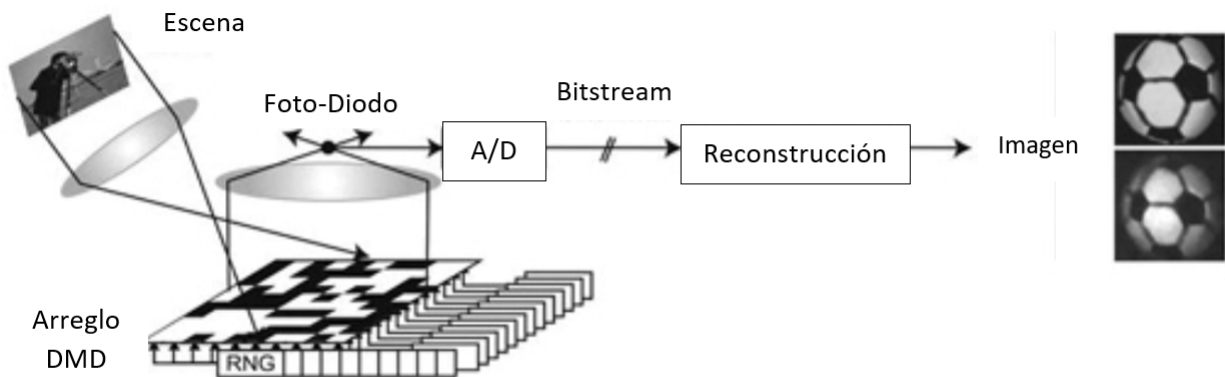


Figura 2.3: Paradigma de la Imagen Digital.

Una imagen digital desde el computo es una representación numérica, puede ser bina-

ria, en escala de grises o en color [22].

El objetivo de la visión por computadora es utilizar las computadoras para emular la visión humana, incluido el aprendizaje y la capacidad de hacer inferencias y tomar acciones basadas en aportes visuales. Esta área en sí es una rama de la inteligencia artificial cuyo objetivo es emular la inteligencia humana [14].

Existen tres tipos de procesos computarizados [14]:

- Nivel bajo: Implican operaciones primitivas, como el preprocesamiento de imágenes para reducir el ruido, la mejora del contraste y el enfoque de la imagen, este se caracteriza por el hecho de que tanto sus entradas como sus salidas son imágenes.
- Nivel medio: Implica tareas como la segmentación, la descripción de esos objetos para reducirlos a una forma adecuada para el procesamiento por computadora y la clasificación de objetos individuales, este se caracteriza por el hecho de que sus entradas generalmente son imágenes, pero sus salidas son atributos extraídos de esas imágenes.
- Nivel alto: Implica 'dar sentido' a un conjunto de objetos reconocidos, como en el análisis de imágenes, y, en el extremo más alejado del continuo, realizar las funciones cognitivas normalmente asociadas con la visión .

Pasos fundamentales en el procesamiento de imágenes digitales [14] ver Figura 2.4:

- Adquisición de imágenes
- Mejora de la imagen
- Procesamiento de imágenes de color
- Wavelets
- Compresión
- Procesamiento morfológico
- Segmentación
- Representación y descripción
- Reconocimiento

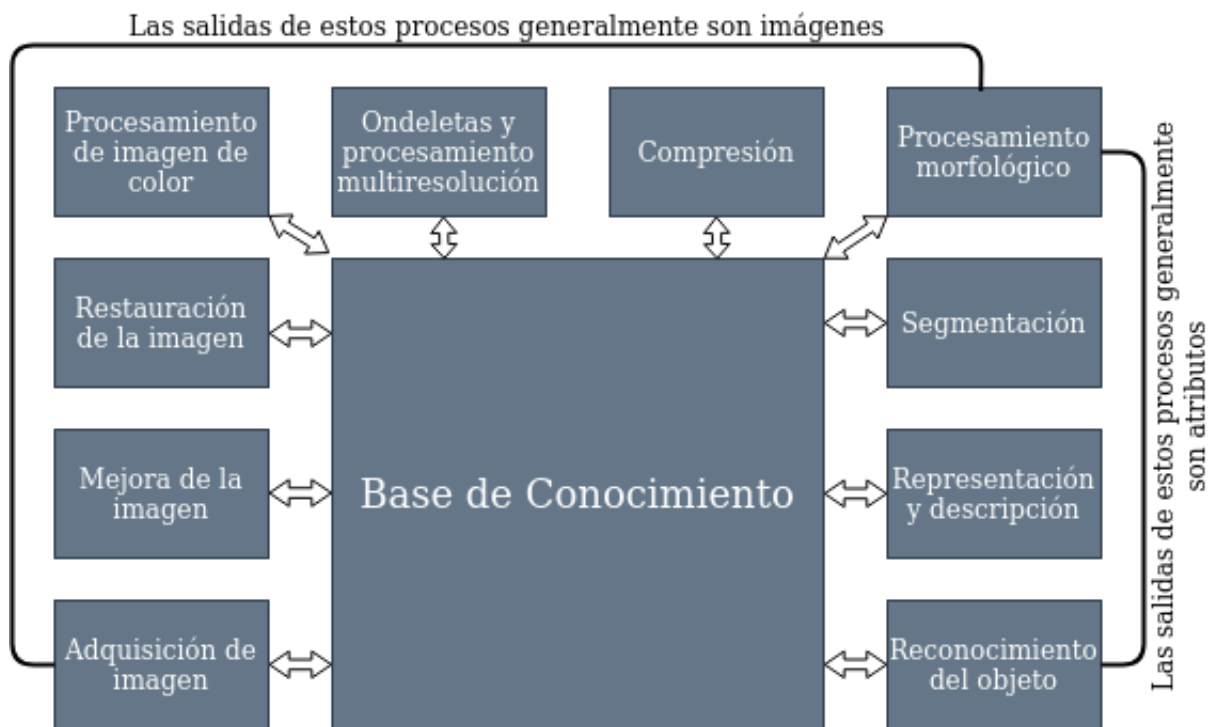


Figura 2.4: Pasos fundamentales en el procesamiento de imágenes digitales [14].

2.2 Estado del arte

En la última década, con el aumento de la capacidad de almacenamiento en la red o en la nube y la masificación de las redes sociales, la disponibilidad por Internet de un gran volumen de imágenes digitales con diversas características es posible; y el hecho de que tales recursos no estén clasificados ha propiciado el desarrollo de herramientas que traten de paliar este problema. Hablando de cantidades reportadas en el año 2017, se estimó en un billón la cantidad de imágenes digitales acorde a una investigación reportada por el servicio MYLIO¹, tal que el 85 % de ellas fueron tomadas por un teléfono móvil. Ante esta realidad se tiene la posibilidad de crear nuevos repositorios de bases de datos de imágenes, pero el reto es clasificarlas acorde a su contenido, de forma no-supervisada, y con una alta eficiencia de forma simple y rápida, todo ello orientado al trabajo fundamental de organizar, proteger y compartir.

En particular, en el campo de la ciencia los grandes avances en la tecnología digital, así como en el campo de la microscopía de luz en los últimos años, determinaron que la imagen celular digital había adquirido una importancia creciente en la biología celular [20]. En el diagnóstico médico, el recuento de células sanguíneas juega un papel muy importante, ya que su aumento o disminución hace que se produzcan muchas enfermedades. Existen diferentes técnicas de recuento de células sanguíneas, el método convencional de conteo manual bajo microscopio requiere mucho tiempo y produce resultados inexactos [4], puede tener algunas limitaciones inherentes de los observadores humanos [20].

Por estas limitaciones existen ya nuevos métodos de adquisición y más sofisticados, como los empleados en la detección de alto contenido, generalmente producen una gran cantidad de datos que exige el poder de las computadoras para analizarlos [20], si bien existen soluciones de hardware como el Contador de hematología automatizado² ver Imagen 2.5, los países en desarrollo no son capaces de disponer de máquinas costosas e inasequibles en todos los laboratorios de los hospitales [4].

Por lo cual al no tener el hardware necesario, podemos hacer uso de el análisis de imágenes por computadora en microscopía celular ya que puede abordar diversas tareas como clasificación y conteo celular, estudios sobre estructuras subcelulares o estudios sobre células vivas, entre otras, sin necesidad de grandes máquinas costosas. También se tiene utilidad en campos como patología, biotecnología vegetal, toxicología, desarrollo de fármacos y otros [20].

¹Página oficial: Here's How Many Digital Photos Will Be Taken in 2017. Eric Perret - <https://focus.mylio.com/tech-today/heres-how-many-digital-photos-will-be-taken-in-2017-repost-oct>, Fecha de consulta: 19-mayo-2020.

²Página oficial: Lighting the way with diagnostigs- <http://sysmex.com/>, Fecha de consulta: 05 de septiembre, 2020.



Figura 2.5: Contador de hematología automatizado.

Para proporcionar una alternativa rentable basada en software y una alternativa eficiente para reconocer y analizar las células del tipo sanguínea [20], es necesario aportar una metodología para hacer un conteo automático de células basado en el procesamiento digital de imágenes. Diferentes etapas deben ser consideradas, desde la adquisición de la imagen, preprocesarla, mejorarla, segmentarla, la extracción de características, aplicación de operaciones morfológicas y la detección de anomalías, se pueden considerar diferentes problemáticas, como la restauración de la imagen, el filtrado de ruido y la reducción del desenfoque. La segmentación de imágenes es también un procesamiento importante aplicado mediante diferentes algoritmos para incluir la separación de los agregados celulares [20].

Para este propósito, se han implementado pocas técnicas de preprocesamiento y postprocesamiento en la imagen de las células sanguíneas para proporcionar una imagen mucho más clara y limpia [4], es aquí donde existe un campo amplio de investigación. Una de las técnicas que ha dado resultados exitosos es la técnica CBIR (Content-Based Image Retrieval) la cual ha sido probada en identificación y clasificación de imágenes de interés [3], tales como los escenarios naturales y la clasificación de rostros humanos. Muchos sistemas y herramientas CBIR se han desarrollado para la consulta de imágenes [1], tal como Query By Image Content (QBIC) desarrollado en los años 90s por IBM [11], que permite al usuario consultas dentro de una gran base de datos con base a atributos de la imagen.

CBIR es un sistema que tiene dos etapas principales: la inserción de datos y el procesamiento de consultas. La inserción de imágenes, es un método no supervisado para la extracción de información (tal como color, forma o texturas), de la base de datos para ser guardados y autoagrupados por medio de un algoritmo para su uso futuro. En el procesamiento de consultas, mediante una imagen de muestra, se obtienen sus características, para ser comparadas con métricas específicas hacia los racimos de datos auto agrupados en la etapa previa, y así obtener el valor dentro de la clasificación previamente hecha de

la base de datos [15].

Anteriormente la clasificación de las imágenes se llevaba a cabo de forma manual y por medio de metadatos insertados en las imágenes tales como etiquetas de texto, pero, en una base de datos muy grande conlleva un costo enorme para su clasificación [18]. Con el uso de CBIR la identificación y clasificación de células sanguíneas se puede automatizar como se ha aplicado a otro tipo de imágenes (rostros y escenarios naturales), ya que se realiza por medio de características de la misma imagen sin necesidad de metadatos, etiquetas o intervención externa, esta propuesta es presentada en el presente trabajo.

Para hacer el reconocimiento de patrones en imágenes de tejido cerebral buscamos desarrollar una metodología computacional que realice el análisis de la imagen de tejido cerebral y la clasifique en una enfermedad neurodegenerativa.

2.2.1 Escaneo Cerebral

Exploración del cerebro, cualquiera de los métodos de diagnóstico para detectar anomalías intracraneales.

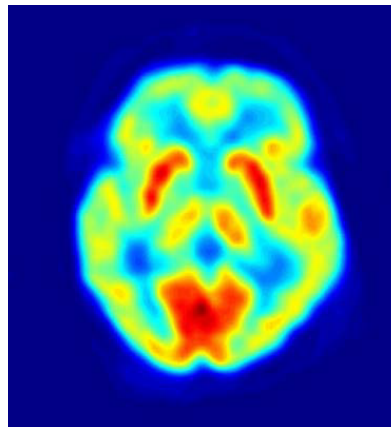


Figura 2.6: Tomografía por emisión de positrones.

El más antiguo de los procedimientos de exploración del cerebro que todavía se utiliza es un procedimiento simple y relativamente no invasivo llamado exploración de isótopos. Se basa en la tendencia de ciertos isótopos radiactivos a concentrarse selectivamente en tumores y lesiones de vasos sanguíneos ver Figura 2.6. El procedimiento implica la inyección de un isótopo radiactivo (como tecnecio-99 o yodo-131) en un vaso sanguíneo que irriga la región craneal. A medida que la sustancia se localiza dentro del cerebro, se descompone y emite rayos gamma. La concentración de rayos en un sitio determinado,

medida por un dispositivo móvil de detección de radiación, puede revelar la presencia, la forma y, a menudo, el tamaño de la anomalía intracraneal. En muchos casos, la exploración de isótopos ha sido reemplazada por la exploración axial computarizada. tomografía (CAT) o tomografía computarizada (TC) [26].

La tomografía computarizada es un procedimiento en el que se toman rayos X del cerebro desde muchos ángulos diferentes ver Figura 2.7. Una fuente de rayos X emite una serie de pulsos cortos de radiación a medida que ella y un detector electrónico giran alrededor de la cabeza del individuo que se está probando. Las respuestas del detector se envían a una computadora que analiza e integra los datos de rayos X de las numerosas exploraciones para construir una imagen transversal detallada del cerebro. Una serie de tales imágenes permite a los médicos localizar tumores cerebrales, abscesos cerebrales, coágulos de sangre y otros trastornos que serían difíciles de detectar con técnicas convencionales de rayos X [26].

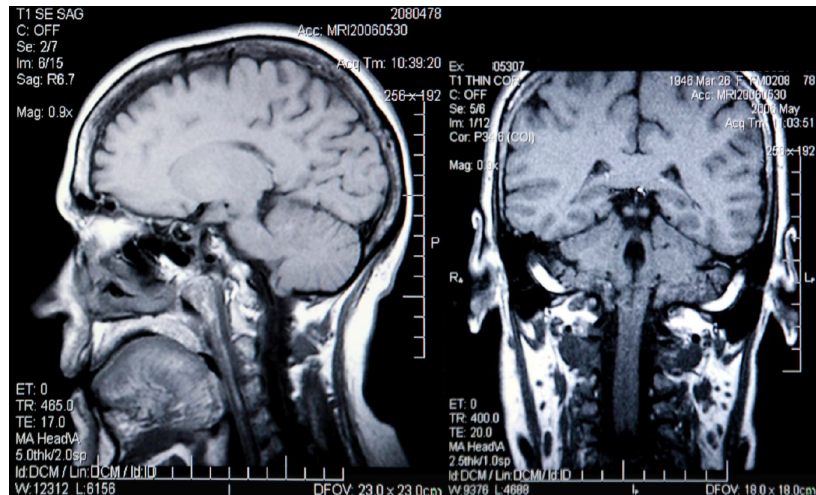


Figura 2.7: Tomografía por emisión de positrones.

Con el desarrollo a mediados de la década de 1970 del TAC, las tecnologías basadas en computadora han revolucionado el campo del diagnóstico médico. Una de las nuevas técnicas tomográficas más importantes es la resonancia magnética nuclear (RMN) ver Figura 2.8. Al igual que la TAC, la RMN genera imágenes de cortes delgados del cerebro (u otro órgano en estudio), pero lo hace sin el peligro de los rayos X u otra radiación ionizante. Además, la RMN puede revelar anomalías fisiológicas y bioquímicas, así como estructurales. (Aunque los beneficios de la RMN son innumerables, la técnica no se recomienda para personas con marcapasos, clips para aneurismas, prótesis metálicas grandes o dependencia de instrumentos que contienen hierro) [26].

La tomografía por emisión de positrones (PET, por sus siglas en inglés) es un procedimiento informático en el que se introduce un compuesto marcado con un marcador radiactivo en el cerebro (u otro órgano en estudio) y se realiza un seguimiento de su comportamiento. Esta información, con modelado por computadora, eventualmente produce

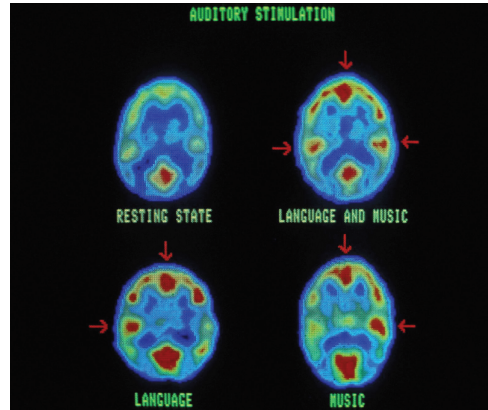


Figura 2.8: Tomografía por emisión de positrones.

una imagen transversal del proceso fisiológico en estudio [26].

Diagnóstico por imagen

Diagnóstico por imagen o también llamado diagnóstico por imágenes médicas, el uso de radiación electromagnética y ciertas otras tecnologías para producir imágenes de estructuras internas del cuerpo con el propósito de un diagnóstico preciso. El diagnóstico por imagen es aproximadamente equivalente a la radiología, la rama de la medicina que utiliza la radiación para diagnosticar y tratar enfermedades. Sin embargo, también se pueden utilizar otras tecnologías, incluido el ultrasonido, que emplea ondas sonoras para visualizar tejidos, y la endoscopia y métodos similares en los que un instrumento óptico flexible está equipado con una cámara para obtener imágenes [16].

Rayos X

Los rayos X, utilizados desde 1895, fueron el primer tipo de radiación para proporcionar imágenes del interior del cuerpo. Los rayos X atraviesan los tejidos corporales y también tienen la propiedad de oscurecer la película fotográfica cuando la golpean. A medida que penetran en los tejidos, los rayos X se absorben de manera diferencial, y los objetos más densos, como los huesos, absorben más rayos y, por lo tanto, impiden que alcancen la película. Los tejidos blandos, por otro lado, absorben menos rayos; El resultado es que en una fotografía de rayos X del interior del cuerpo, los huesos se muestran como áreas más claras y los tejidos blandos se muestran más oscuros en la película expuesta [16].

Un factor limitante en los rayos X cuando se usan solos es la incapacidad de distinguir entre tejidos blandos diferenciados adyacentes de aproximadamente la misma densidad (es decir, no es posible producir tonos contrastantes entre tales objetos en la película expuesta). Para obtener este contraste, se inyecta en el cuerpo un medio de contraste, una sustancia líquida o gaseosa que es comparativamente opaca a los rayos X (radiopaca) o comparativamente transparente para ellos. Los fluidos de medio de contraste se pueden inyectar en las cavidades corporales naturales, inyectar en el torrente sanguíneo y los vasos

linfáticos, tragar o introducir mediante enema para estudiar el tracto digestivo, o inyectar alrededor de los órganos para mostrar su contorno externo. Diferentes medios de contraste permiten así la obtención de imágenes de rayos X de tipos particulares de estructuras internas blandas, como las arterias y venas en la angiografía, el paso de la sangre a través del corazón en la angiocardiografía, la vesícula biliar y los canales biliares en la colecistografía, la médula espinal en la mielografía y la tracto urinario en urografía. Prácticamente cualquier parte del cuerpo puede examinarse en busca de alteraciones fisiológicas de las estructuras normales mediante análisis de rayos X. Las películas cinematográficas de rayos X pueden registrar los procesos corporales cuando los medios de contraste entran y salen de partes del cuerpo [16].

Se han desarrollado otras técnicas de imagen utilizando rayos X. En la tomografía, se pueden obtener imágenes de rayos X de estructuras internas profundas enfocando los rayos en un plano específico dentro del cuerpo. Una variación más compleja de esta técnica es la tomografía computarizada, conocida como tomografía computarizada [16].

Medicina Nuclear

El escaneo de isótopos radiactivos que se han inyectado en los tejidos es una especialidad médica llamada medicina nuclear. Tanto la exploración de isótopos como la fotografía de rayos X se utilizan en la exploración del cerebro. Una técnica de imagen relacionada con la exploración isotópica es la tomografía por emisión de positrones. Otro tipo de diagnóstico por imágenes es la resonancia magnética nuclear, que crea imágenes de cortes delgados del cuerpo utilizando ondas de radio de muy alta frecuencia. La ecografía es una técnica en la que se utilizan ondas sonoras de alta frecuencia para detectar anomalías en los órganos internos. Las variedades de radiación que se utilizan en las imágenes de diagnóstico continúan expandiéndose, junto con las técnicas para usarlas [16].

Endoscopia y procedimientos relacionados

Los procedimientos como la endoscopia, laparoscopia y colposcopia utilizan instrumentos ópticos generalmente flexibles que pueden insertarse a través de aberturas, ya sea de origen natural o quirúrgico, en el cuerpo. Muchos instrumentos de alcance están equipados con pequeñas cámaras de video que permiten al médico o al cirujano ver los tejidos que se examinan en un monitor grande. También se han diseñado varios osciloscopios para permitir la realización de una biopsia de tejido, en la que se recoge una pequeña muestra de tejido para el estudio histológico, junto con el análisis visual [16].

3

PROPUESTA

3.1 Metodología general

La metodología general de un sistema de visión y reconocimiento de imágenes en general se muestra en la Figura 3.1, acorde con [15], y se tienen variantes dependiendo el tipo de problemas a resolver, e.g. el reconocimiento de firmas escritas a mano [24], el reconocimiento de tipo de letra en documentos digitales [6], el reconocimiento de rostros [1].

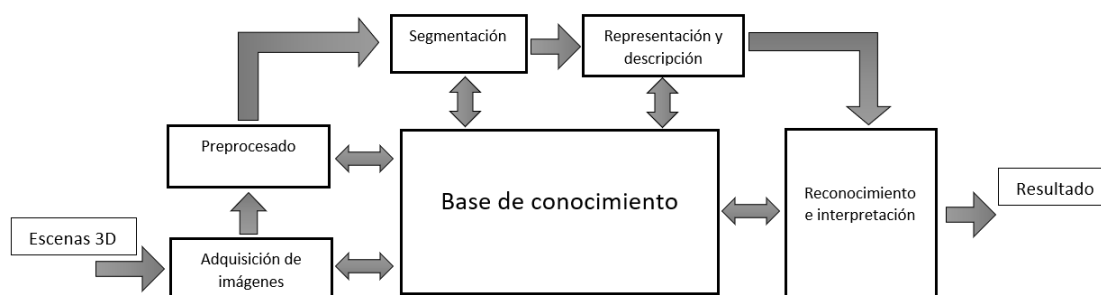


Figura 3.1: Metodología general de un sistema de visión y reconocimiento de imágenes.

Etapas de la metodología general de un sistema de visión y reconocimiento de imágenes:

- La adquisición de imágenes es el primer proceso que se muestra en la Figura 3.1. La adquisición podría ser tan simple como recibir una imagen que ya está en formato digital. Generalmente, la etapa de adquisición de imágenes implica un procesamiento previo, como el escalado.
- Preprocesado, esta etapa abarca diferentes sub-etapas como son:
 - La mejora de la imagen es una de las áreas más simples y atractivas del procesamiento de imágenes digitales. Básicamente, la idea detrás de las técnicas de mejora es resaltar los detalles que están oscurecidos o simplemente resaltar ciertas características de interés en una imagen. Un ejemplo familiar de mejora es cuando aumentamos el contraste de una imagen porque "se ve mejor".

Es importante tener en cuenta que la mejora es un área muy subjetiva del procesamiento de imágenes.

- La restauración de imágenes es un área que también se ocupa de mejorar la apariencia de una imagen. Sin embargo, a diferencia de la mejora, que es subjetiva, la restauración de la imagen es objetiva, en el sentido de que las técnicas de restauración tienden a basarse en modelos matemáticos o probabilísticos de degradación de la imagen. La mejora, por otro lado, se basa en las preferencias subjetivas humanas con respecto a lo que constituye un resultado de mejora "bueno".
 - El procesamiento de imágenes en color es un área que ha ido ganando importancia debido al aumento significativo en el uso de imágenes digitales en Internet. Son una serie de conceptos fundamentales en modelos de color y procesamiento de color básico en un dominio digital. El color también se utiliza como base para extraer características de interés en una imagen.
 - Las ondeletas son la base para representar imágenes en varios grados de resolución. En particular, se utiliza para la compresión de datos de imágenes y para la representación piramidal, en la que las imágenes se subdividen sucesivamente en regiones más pequeñas.
 - La compresión, como su nombre lo indica, se ocupa de las técnicas para reducir el almacenamiento requerido para guardar una imagen, o el ancho de banda requerido para transmitirla. Aunque la tecnología de almacenamiento ha mejorado significativamente durante la última década, no se puede decir lo mismo de la capacidad de transmisión. Esto es cierto particularmente en los usos de Internet, que se caracterizan por un contenido pictórico significativo. La compresión de imágenes es familiar para la mayoría de los usuarios de computadoras en forma de extensiones de archivo de imagen, como la extensión de archivo *jpg* utilizada en el estándar de compresión de imágenes JPEG (Joint Photographic Experts Group).
 - El procesamiento morfológico se ocupa de herramientas para extraer componentes de la imagen que son útiles en la representación y descripción de la forma. Comienza una transición de procesos que generan imágenes a procesos que generan atributos de imagen.
- Los procedimientos de segmentación dividen una imagen en sus partes u objetos constituyentes. En general, la segmentación autónoma es una de las tareas más difíciles en el procesamiento de imágenes digitales. Un procedimiento de segmentación riguroso lleva el proceso un largo camino hacia la solución exitosa de problemas de imágenes que requieren que los objetos se identifiquen individualmente. Por otro lado, los algoritmos de segmentación débiles o erráticos casi siempre garantizan un eventual fallo. En general, cuanto más precisa sea la segmentación, es más probable que el reconocimiento tenga éxito.
 - La representación y la descripción casi siempre siguen la salida de una etapa de segmentación, que generalmente son datos de píxeles sin procesar, que constituyen

el límite de una región (es decir, el conjunto de píxeles que separa una región de la imagen de otra) o todos los puntos de la región en sí. En cualquier caso, es necesario convertir los datos a un formato adecuado para el procesamiento informático. La primera decisión que debe tomarse es si los datos deben representarse como un límite o como una región completa. La representación de límites es apropiada cuando el foco está en las características de la forma externa, como esquinas e inflexiones. La representación regional es apropiada cuando el foco está en las propiedades internas, como la textura o la forma del esqueleto. En algunas aplicaciones, estas representaciones se complementan entre sí. La elección de una representación es solo una parte de la solución para transformar los datos en bruto en una forma adecuada para su posterior procesamiento informático. También se debe especificar el método para describir los datos de modo que se resalten las características de interés. La descripción, también llamada selección de características, se ocupa de extraer atributos que dan como resultado información cuantitativa de interés o que son básicos para diferenciar una clase de objetos de otra.

- El reconocimiento es el proceso que asigna una etiqueta (por ejemplo, "vehículo") a un objeto en función de sus descriptores.

Metodología propuesta para análisis por textura de imágenes de células gliales

Para el problema de la identificación y localización de las células gliales en imágenes digitales se propone el modelo mostrado en la Figura 3.2, en el que se plantean las siguientes etapas:

Etapas 1

- *Adquisición* de la base de datos de imágenes de tejido cerebral, todas ellas se encuentran en el espacio de color RGB.
- *Preprocesamiento*
 - Se aplica a cada una de las imágenes un filtro Gaussiano
 - Ecuilibración de imagen
 - Exportación de las imágenes del espacio de color RGB original, al espacio HSI, con el objetivo de trabajar con más información independiente del color de la imagen y buscando tener información de las texturas de las imágenes

Etapas 2

- El usuario selecciona una sola imagen (de referencia) de la pila de imágenes previamente procesadas en la etapa 1, y en dicha imagen selecciona una región de interés (ROI) con el cursor.

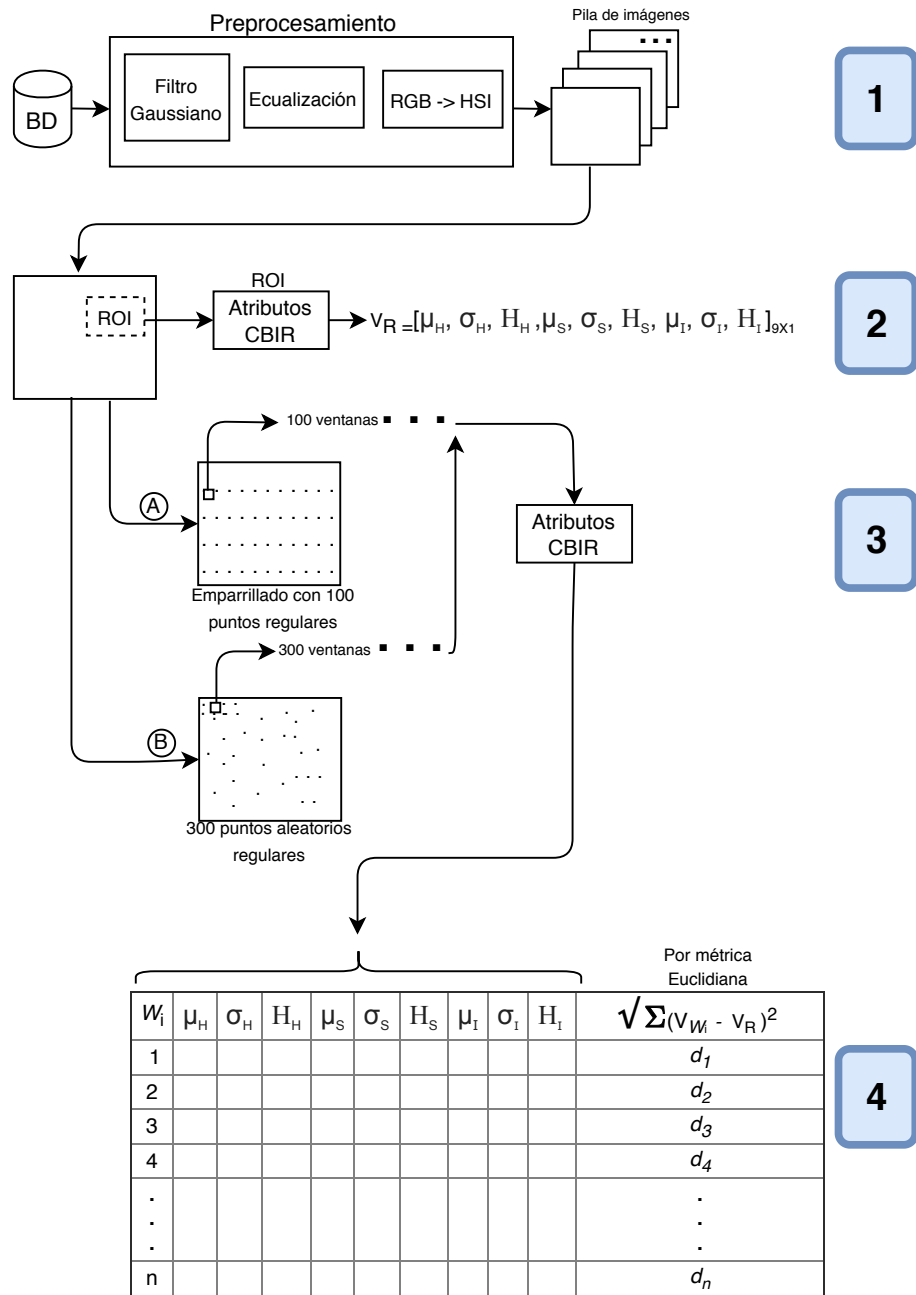
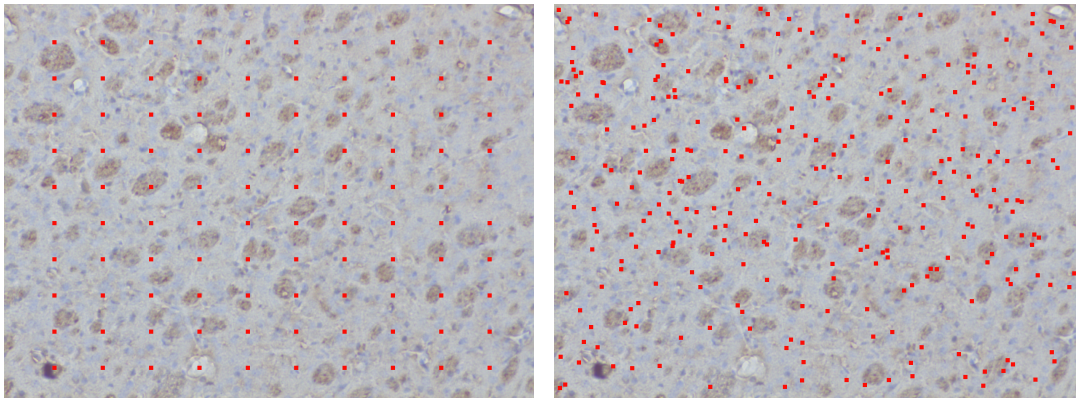


Figura 3.2: Metodología propuesta para análisis por textura de imágenes de células gliales.

- A continuación se procede a obtener los atributos, el sistema obtiene los atributos CBIR de la ROI, los cuales los disponemos en un vector de la forma: $V_R = [\mu_H, \sigma_H, H_H, \mu_S, \sigma_S, H_S, \mu_I, \sigma_I, H_I]_{9 \times 1}$, de tamaño 9×1 , ya que se tienen tres capas de la imagen y por cada capa se extraen los atributos: media, desviación estándar y homogeneidad.

Etapa 3

- En la imagen de referencia que el usuario seleccionó anteriormente se genera un emparejamiento con dos casos: regular de 100 puntos, o aleatorio de 300 puntos distribuidos uniformemente, ver Figura 3.3a.



(a) Sembrado de 100 puntos.

(b) Sembrado de 300 puntos.

Figura 3.3: Sembrado de puntos en la imagen.

- En cada punto generado se abre una ventana de tamaño 10×10 píxeles, ver Figura 3.4.

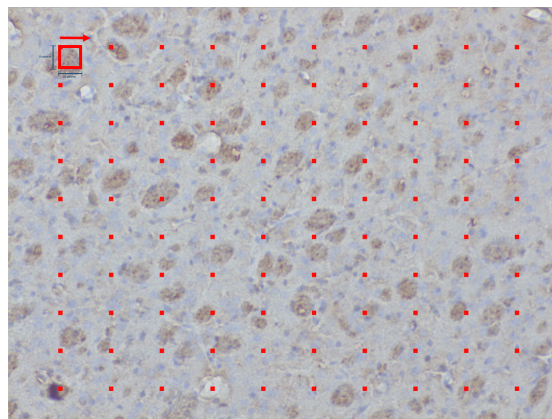


Figura 3.4: Construcción de ventana de 10x10 píxeles.

- A cada una de las (100 o 300) ventanas se extraen sus atributos CBIR teniendo como resultado una tabla de tamaño $(9 \times 100, \text{ o } 9 \times 300)$, nueve de los atributos CBIR y (100 o 300) por cada ventana según el caso.

Etapa 4

- Aplicación de un clasificador de distancia K-NN, con $K = 3, 5, 10$ o 30 [12]. A cada vector de la tabla obtenida se le aplica la métrica euclidiana contra el vector V_R y se guardan esas distancias en dicha tabla para después aplicar un ordenamiento ascendente y tomar las 3, 5, 10 o 30 regiones más parecidas a la ROI.

3.2

Integración de paralelismo

Uno de los problemas principales de la implementación de las pruebas semi-supervisadas automáticas es el tiempo de ejecución, es por esto que se propone una versión paralela, en la Figura 3.5 mostramos la arquitectura del sistema.

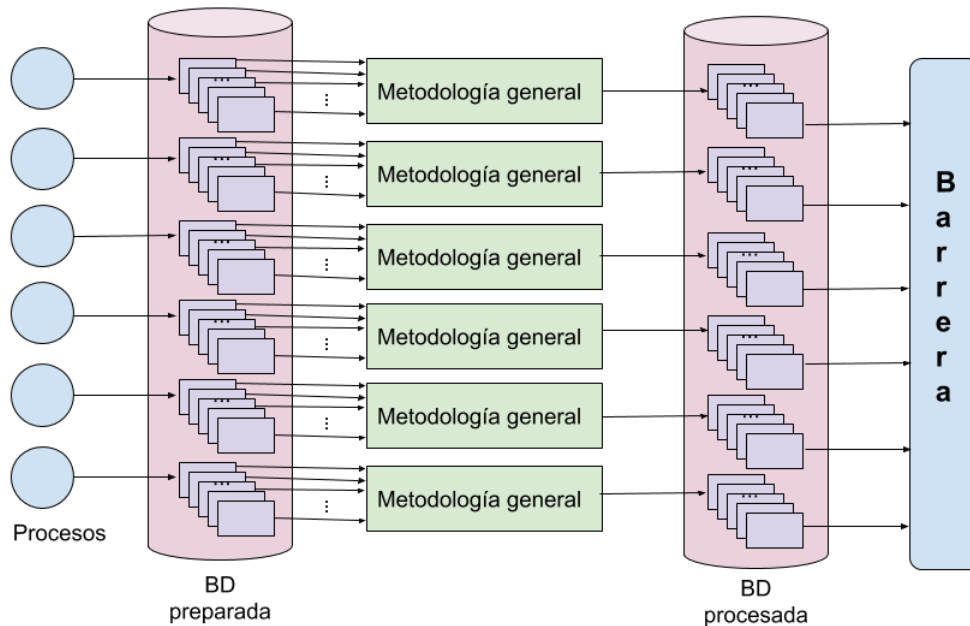


Figura 3.5: Propuesta de arquitectura paralela para implementar de forma masiva la metodología general.

Con el esquema de la Figura 3.2 ejemplificamos lo que hace cada uno de los procesos con cada imagen.

Para el paralelismo usamos la interfaz de paso de mensajes o MPI (sigla del inglés *Message Passing Interface*), es un estándar que define la sintaxis y la semántica de las funciones contenidas en una biblioteca de paso de mensajes, diseñada para ser usada en programas que exploten la existencia de múltiples procesadores. La ventaja de MPI sobre otras bibliotecas de paso de mensajes, es que los programas que utilizan la biblioteca son portables (dado que MPI ha sido implementado para casi toda arquitectura de memoria distribuida), y rápidos, (porque cada implementación de la biblioteca ha sido optimizada para el hardware en la cual se ejecuta).

3.3 Algoritmo paralelo

Algoritmo 3.1 ejecutado simultáneamente por cada proceso i

Entrada: subconjunto de imágenes (total de imágenes / n procesos).

```
1: para img en Imagenes hacer  
2:   filtroGaussiano()  
3:   ecualizacion()  
4:   RGBHSI()  
5:   seleccionROIaleatoria()  
6:   emparrillado()  
7:   extraccionCaracteristicas()  
8:   KNN()  
9:   guardaImagen(img_procesada)  
10: fin para  
11: BARRERA
```

Lo que hace este algoritmo es que cada proceso tiene un subconjunto de imágenes (total de imágenes / número de procesos). Una vez seleccionada la imagen realiza el preprocesamiento de la imagen que corresponde al filtro Gaussiano, la ecualización y convertir la imagen del espacio de color RGB al espacio de color HSI; una vez hecho esto se selecciona una ROI en la imagen de forma aleatoria y después se hace el emparrillado aleatorio de 300 puntos a la imagen y se extraen las características CBIR de cada uno de las subventanas que se abrieron en cada punto. Una vez hecho esto se lleva acabo el clasificador K- vecinos más cercanos y se guarda la imagen en disco. Esto se hace hasta acabar su montículo de imágenes y después cada proceso espera en una barrera a que todos los procesos terminen y acaba el programa.

4

EXPERIMENTOS

En este capítulo mostramos como y en donde se realizaron los distintos experimentos, como son: sembrado de 100 puntos regulares y sembrado de 300 puntos aleatorios con ventanas de interés de 10 píxeles y 30 píxeles, después se muestra una subsección de pruebas semi-supervisadas automáticas las cuales ponen a prueba nuestra metodología demostrando que en aquellas imágenes donde la ROI coincide con una sección de interés de la imagen, efectivamente se encuentran objetos similares a este.

Requerimientos de Hardware y Software para realizar la implementación:

- Procesador: Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz (12 CPUs), 2.2GHz
- Sistema operativo: MS Windows 10, 64 bits
- Memoria RAM: 8 GB
- Lenguaje: Python 3.8.1
- Librería: Open CV 4.1.2

Base de datos: se cuenta con un total de 312 imágenes, 190 de campo claro y 122 de fluorescencia, en formato TIF. En la tabla 4.1 se muestra la resolución de las imágenes.

Tipo	Resolución
Campo Claro	1392 × 1040 píxeles
Fluorescencia	355 × 402 a 1280 × 960 píxeles

Tabla 4.1: Resolución de imágenes.

Imágenes de campo claro

La generación de imágenes de campo claro es la forma más sencilla de microscopía en donde la luz pasa a través de, o se refleja desde un espécimen. La iluminación no se altera por los dispositivos que alteran las propiedades de la luz ver Figura 4.1.

En aplicaciones biológicas, la observación de campo claro se usa ampliamente para especímenes manchados, naturalmente pigmentados o altamente contrastados, montados sobre un portaobjetos de vidrio para microscopio. El espécimen se ilumina desde abajo y se observa desde arriba. El espécimen aparece brillante, pero más oscuro que el brillo en el fondo. Esta técnica se usa ampliamente en patología para observar secciones de tejido fijo o capas / frotis de célula. La generación de imágenes de campo claro no es muy útil para células vivas sin tinción o secciones de tejido sin tinción, ya que en la mayoría de los casos, la luz pasa a través de muestras transparentes o translúcidas con poca estructura o sin definición de la misma [8].

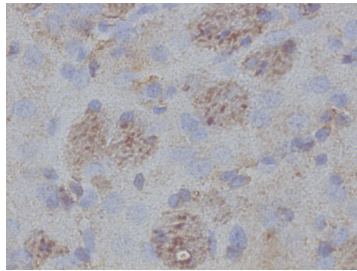


Figura 4.1: Ejemplo de una célula glial en campo claro.

Imágenes de fluorescencia

La fluorescencia es uno de los fenómenos físicos más utilizados en microscopía biológica y analítica, sobre todo por su alto grado de sensibilidad y especificidad. La fluorescencia es una forma de luminiscencia ver Figura 4.2.

Los microscopios de fluorescencia que se emplean en aplicaciones de investigación se basan en una serie de filtros ópticos: filtro excitación, divisor de rayos dicróico y filtro de emisión. El filtro de excitación selecciona las longitudes de onda que excitan un fluoróforo concreto dentro de la muestra, el filtro de emisión actúa como una especie de control de calidad, ya que solo permite pasar las longitudes de onda de interés emitidas por el fluoróforo. El espejo dicróico refleja luz en la banda de excitación y transmite luz en la banda de emisión, lo que permite la iluminación clásica de luz incidente para epifluorescencia [13].

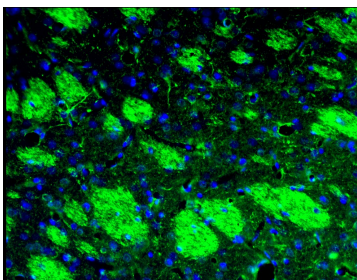


Figura 4.2: Ejemplo de una célula glial en fluorescencia.

Experimentos con diferentes emparrillados y subventanas

Las primeras pruebas se hicieron probando con dos formas de emparrillado la primera es un emparrillado regular de 100 puntos con subventanas de tamaño 10×10 pixeles como en la Figura 4.3 y las segundas pruebas fueron con un emparrillado de 300 puntos aleatorios con subventanas de tamaño 10×10 como en la Figura 4.4 y de 30×30 pixeles como en la Figura 4.5.

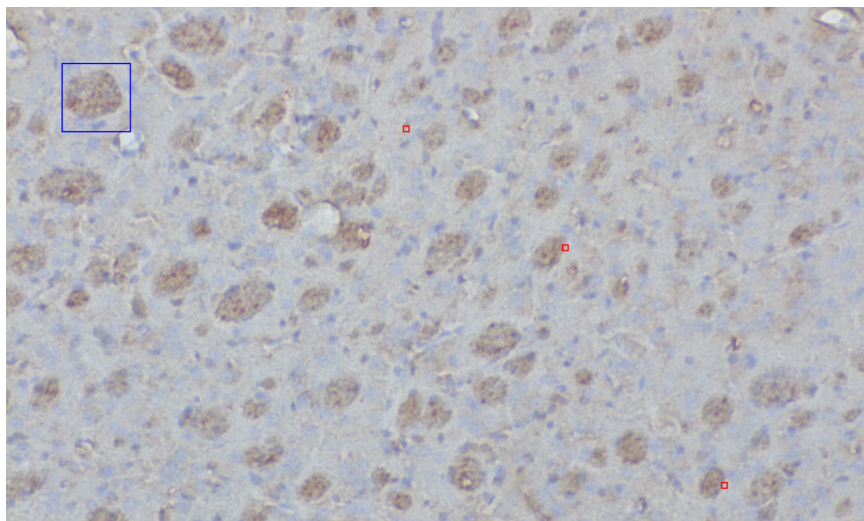


Figura 4.3: Imagen celular con la ROI seleccionada por el usuario y las tres ROI de tamaño 10×10 más parecidas, considerando un emparrillado regular de la imagen.

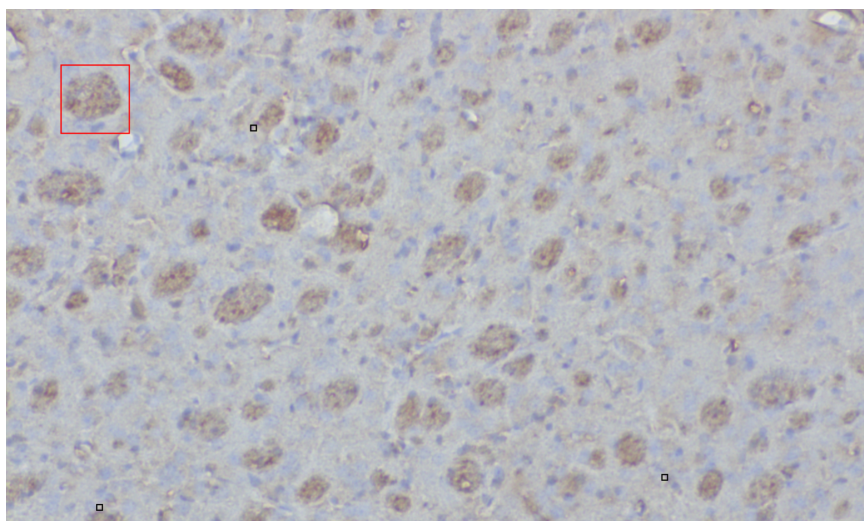


Figura 4.4: Imagen celular con la ROI seleccionada por el usuario y las tres ROI de tamaño 10×10 más parecidas, considerando un emparrillado aleatorio de la imagen.

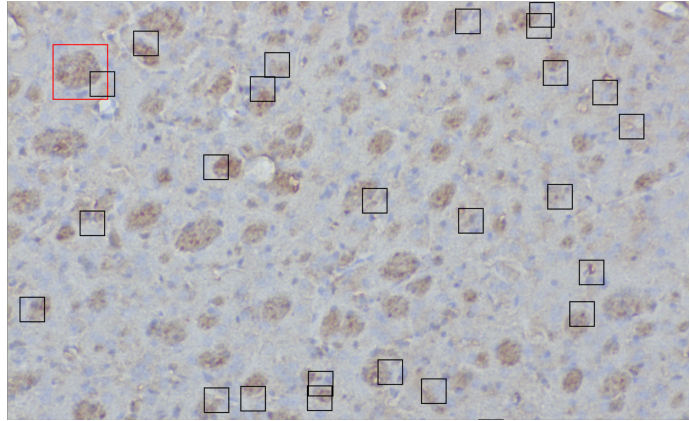


Figura 4.5: Imagen celular con la ROI seleccionada por el usuario y las tres ROI de tamaño 30×30 más parecidas, considerando un emparrillado aleatorio de la imagen.

Se realizaron diferentes pruebas con las imágenes de campo claro y de fluorescencia, lo que se quiere realizar es tener una imagen de células gliales y ya sea seleccionando una ROI de manera manual o una de manera aleatoria buscar ese objeto de interés en la misma imagen y mostrar esos objetos parecidos.

Criterio de pruebas semi-supervisadas automáticas

Con la finalidad de poder tener un criterio para analizar la posible efectividad de nuestra metodología, se partió de marcar o generar una ROI aleatoria sobre cada una de las imágenes de prueba.

Por lo cual se tienen dos tipos de pruebas una es de manera supervisada y otra de forma no supervisada. En la primera forma supervisada lo que se hace es que el médico especialista selecciona una ROI de forma manual y él da su punto de vista al decir si los objetos localizados son semejantes o no. Y la segunda forma de manera no supervisada, la ROI se selecciona de manera aleatoria en la imagen y se muestran los más parecidos.

5

RESULTADOS

Emparrillado de 100 puntos

En los primeros experimentos que se realizaron se eligió un emparrillado de 100 puntos regulares en la imagen, se selecciona la ROI y a partir de eso se buscaban las 3 ROI que más se parecían a los atributos CBIR de la selección, como se muestra en la Figura 4.4. Las ROI más parecidas se fijaron como ventanas de tamaño 10×10 , pero al observar los resultados estas ventanas eran muy pequeñas para el tamaño de la imagen que aproximadamente es de 1040×1392 .

Emparrillado de 300 puntos

Este experimento fue realizado con un emparrillado aleatorio de 300 puntos y de igual forma se obtuvieron las tres ROI que más se parecían con respecto a las características CBIR obtenidas de la región seleccionada, pero de igual forma las ventanas de tamaño 10×10 eran muy pequeñas ver Figura 4.4.

Por último, se ajustó el tamaño de las ventanas a un espacio de 30×30 pixeles y así fue más fácil de visualizar los resultados, como se muestra en la Figura 4.5.

Ajuste del tamaño de la ROI

El tamaño de las subventanas depende mucho del tamaño de las imágenes, ya que el tamaño que manejamos de las mismas es muy grande, una ventana de tamaño 10×10 pixeles es muy pequeña para visualizar los resultados. El tamaño de las ventanas para señalar los objetos parecidos al objeto señalado por el usuario en la ROI debería ser de un tamaño aproximado al tamaño del objeto de interés ver Figura 5.1.

Pruebas semi-supervisadas automáticas

Con la finalidad de poder tener un criterio para analizar la posible efectividad de nuestra metodología, se partió de marcar o generar una ROI aleatoria sobre cada una de las imágenes de prueba, de tal forma que se tiene la suerte de que caiga o no encerrando o delimitando una región verdadera de interés, i.e. una célula tipo glía, tal como se observa

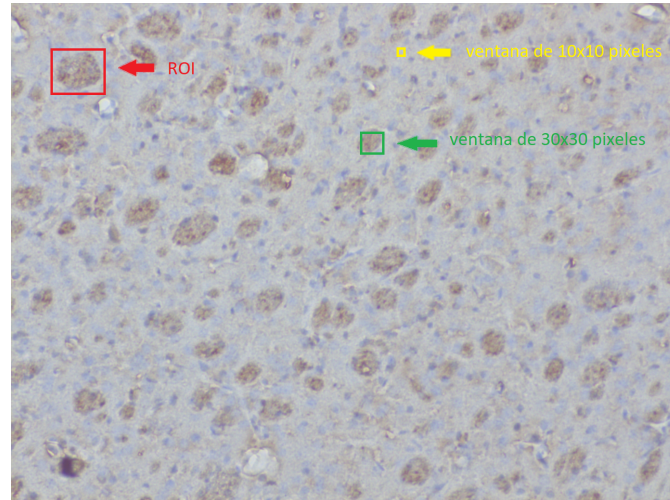


Figura 5.1: Proporción de las imágenes vs ROI vs ventanas.

en la Figura 5.2 la ROI seleccionada, ya que es aleatoria selecciona más de un objeto y los objetos que se muestran son similares, pero en la Figura 5.3 la ROI seleccionada contiene muchos objetos, esto debido a que la imagen tiene más objetos o fue realizada con un acercamiento más detallado, por lo cual no podemos decir si los objetos que encuentra son o no son parecidos.

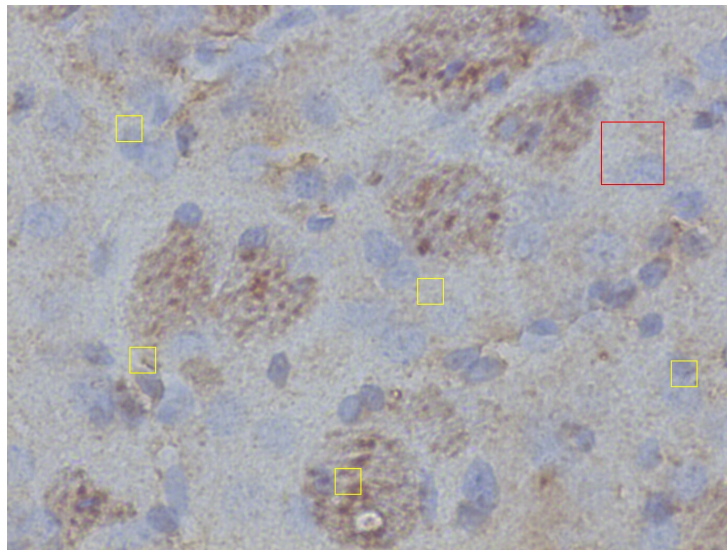


Figura 5.2: Imagen celular con la ROI seleccionada aleatoriamente y las cinco ROI de tamaño más parecidas, considerando un emparrillado aleatorio donde delimita un objeto

Se puede ver en las muestras presentadas que por el momento la ROI aleatoria para la batería de pruebas no precisamente cae encerrando una célula tipo glia, pero los resultados de las más parecidas aparentemente son satisfactorias para el profesional de la salud según su criterio de supervisión de las pruebas y su resultado.

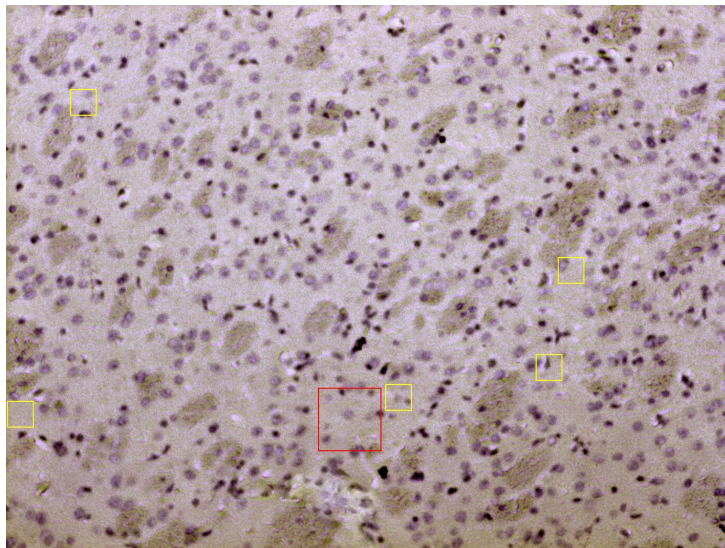


Figura 5.3: Imagen celular con la ROI seleccionada aleatoriamente y las cinco ROI de tamaño más parecidas, considerando un emparrillado aleatorio donde delimita varios objetos.

Resultados del algoritmo paralelo

Se hicieron pruebas con diferente números de procesos y los resultados se muestran en la Tabla 5.1 y en la Gráfica 5.5

Procesos	Tiempo min
2	109
4	66
6	50
8	84
10	64
12	55

Tabla 5.1: Tiempo que tardó el algoritmo paralelo con los diferentes procesos.

Se utilizó paralelismo, ya que al hacer las pruebas semisupervisadas el tiempo secuencial es 82.16 minutos, y al hacerlo concurrente se obtuvo un tiempo de 55.24 minutos el mejor tiempo.

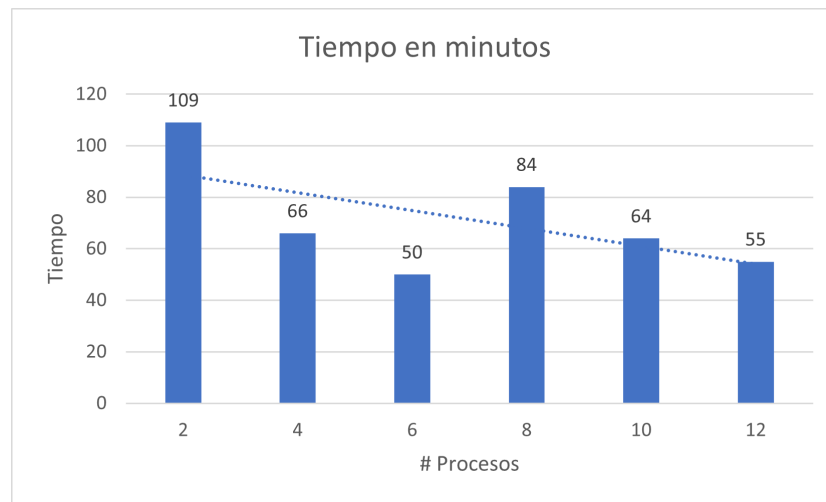


Figura 5.4: Comparación por tiempo en procesos.

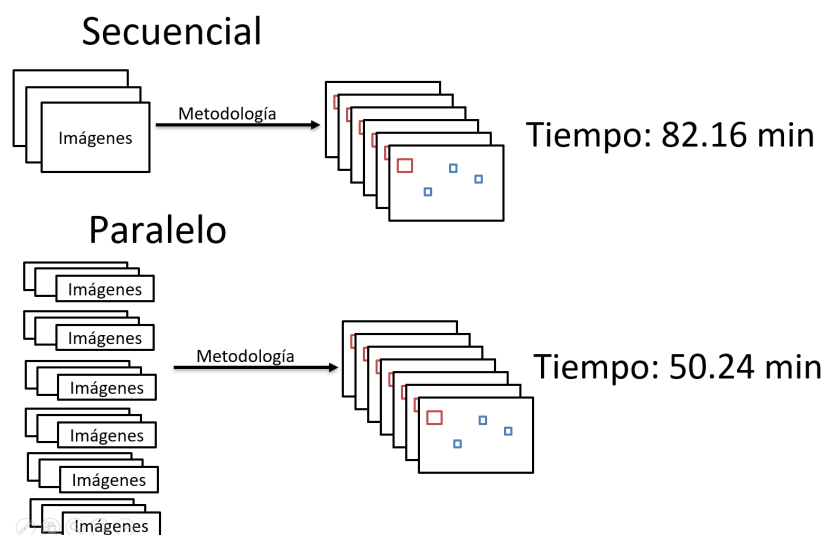


Figura 5.5: Comparación de secuencial vs paralelismo

6

DESCRIPCIÓN DEL SISTEMA

En esta sección mostramos la aplicación desarrollada para el personal profesional de la salud, para que pudieran probar nuestra propuesta de análisis de imágenes. Se dió una capacitación a distancia con el Dr. Edgar Rangel López, del Instituto Nacional de Neurología y neurocirugía Manuel Velasco Suárez, para que fuera operado y puesto a prueba a la distancia, ya que por la pandemia nacional del COVID-19, y de las restricciones de movilidad e ingreso al instituto únicamente del personal adscrito que no me fue posible asistirles en persona, pero gracias a la comunicación por internet y de telefonía móvil se pudo realizar la prueba operativa. En las siguientes partes se muestra la aplicación, su forma de ejecución, partes que la componen y los pasos de cómo se ejecuta una prueba de funcionamiento.

1. Primero seleccionamos la carpeta “proyectoGlias”.

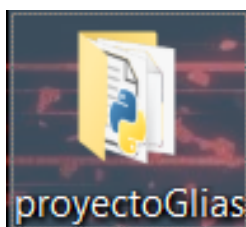


Figura 6.1: Carpeta de proyecto.

2. Después se selecciona la carpeta *Ejecutable*.
3. Dar doble click en el archivo *Ejecutable*.
4. Seleccionar la carpeta “Img_prueba”, en la cual se encuentran las imágenes de glías a estudiar.
5. Seleccionamos la imagen que se quiere analizar.

__pycache__	05/09/2020 03:41 p. m.	Carpeta de archivos	
build	05/09/2020 03:40 p. m.	Carpeta de archivos	
Ecuilizacion	05/09/2020 03:40 p. m.	Carpeta de archivos	
Ejecutable	14/10/2020 04:03 p. m.	Carpeta de archivos	
FGaussiano	05/09/2020 03:41 p. m.	Carpeta de archivos	
HSI	05/09/2020 03:41 p. m.	Carpeta de archivos	
Img_prueba	05/09/2020 03:41 p. m.	Carpeta de archivos	
resultado	05/09/2020 03:41 p. m.	Carpeta de archivos	
ClasicadorKnn	02/05/2020 06:17 p. m.	Python File	2 KB
Emparrillado	06/07/2020 06:48 p. m.	Python File	7 KB
Knn	30/04/2020 05:56 p. m.	Python File	1 KB
Preprocesamiento	21/04/2020 01:15 p. m.	Python File	2 KB
Principal	29/08/2020 11:45 a. m.	Python File	3 KB
Principal.spec	29/08/2020 11:52 a. m.	Archivo SPEC	1 KB
SeleccionROI	29/08/2020 11:44 a. m.	Python File	4 KB

Figura 6.2: Carpeta de *Ejecutable*.

Ejecutable	29/08/2020 11:55 a. m.	Aplicación	285,821 KB
------------	------------------------	------------	------------

Figura 6.3: Archivo *Ejecutable*.

__pycache__	05/09/2020 03:41 p. m.	Carpeta de archivos	
build	05/09/2020 03:40 p. m.	Carpeta de archivos	
Ecuilizacion	05/09/2020 03:40 p. m.	Carpeta de archivos	
Ejecutable	14/10/2020 04:03 p. m.	Carpeta de archivos	
FGaussiano	05/09/2020 03:41 p. m.	Carpeta de archivos	
HSI	05/09/2020 03:41 p. m.	Carpeta de archivos	
Img_prueba	05/09/2020 03:41 p. m.	Carpeta de archivos	
resultado	05/09/2020 03:41 p. m.	Carpeta de archivos	
ClasicadorKnn	02/05/2020 06:17 p. m.	Python File	2 KB
Emparrillado	06/07/2020 06:48 p. m.	Python File	7 KB
Knn	30/04/2020 05:56 p. m.	Python File	1 KB
Preprocesamiento	21/04/2020 01:15 p. m.	Python File	2 KB
Principal	29/08/2020 11:45 a. m.	Python File	3 KB
Principal.spec	29/08/2020 11:52 a. m.	Archivo SPEC	1 KB
SeleccionROI	29/08/2020 11:44 a. m.	Python File	4 KB

Figura 6.4: Carpeta 'Img_prueba'.

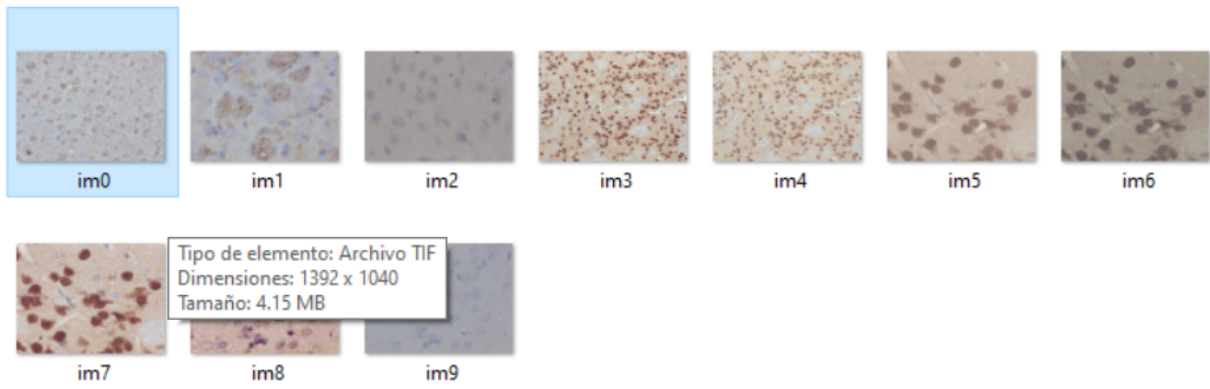


Figura 6.5: Imagen a analizar.

6. Dar click en el botón *Abrir*.

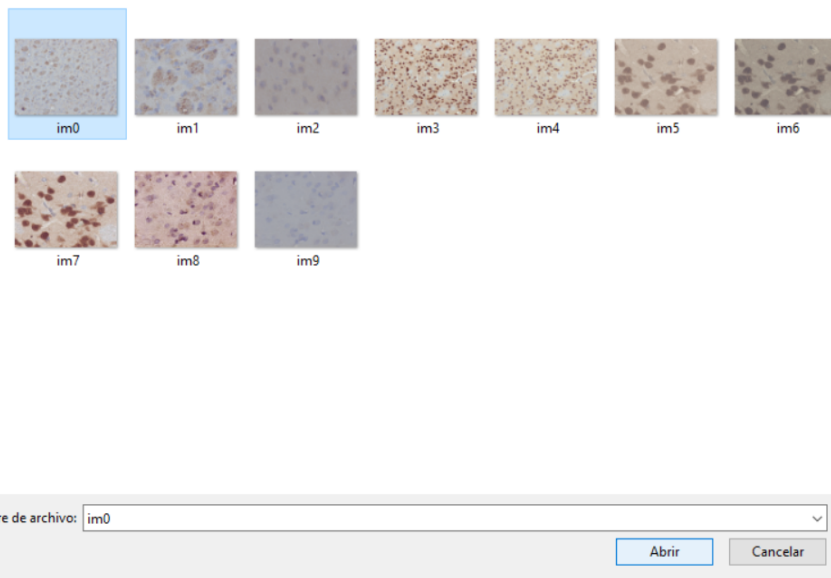


Figura 6.6: Botón *Abrir*.

7. Se abre la imagen que se selecciono.

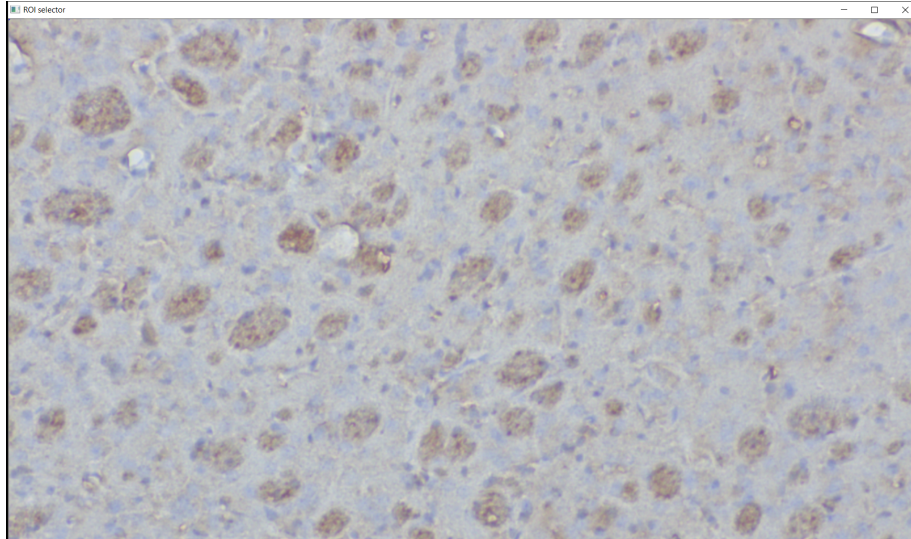


Figura 6.7: Imagen seleccionada.

8. Con el cursor seleccionar el objeto del cual se quieren buscar los objetos más parecidos a este.

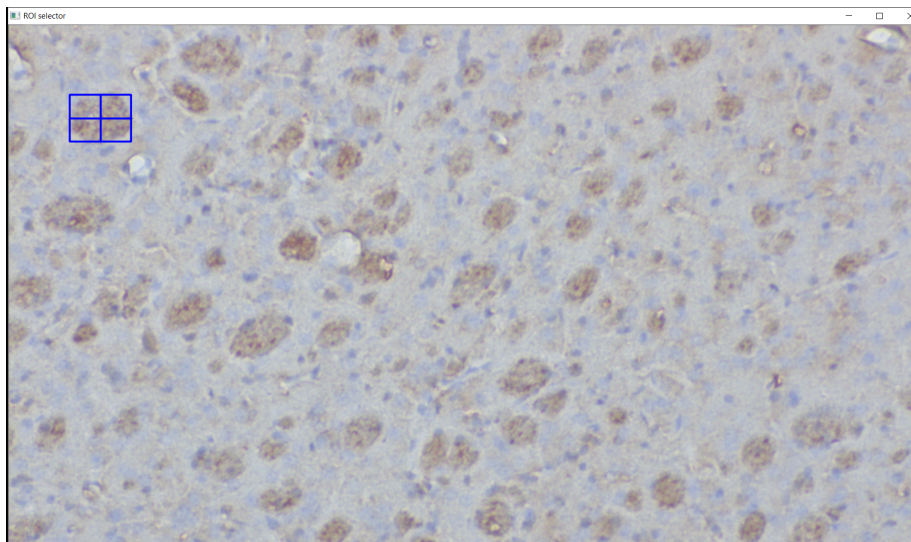


Figura 6.8: Seleccionar objeto.

9. Se abrirá otra ventana con el objeto que se selecciono.

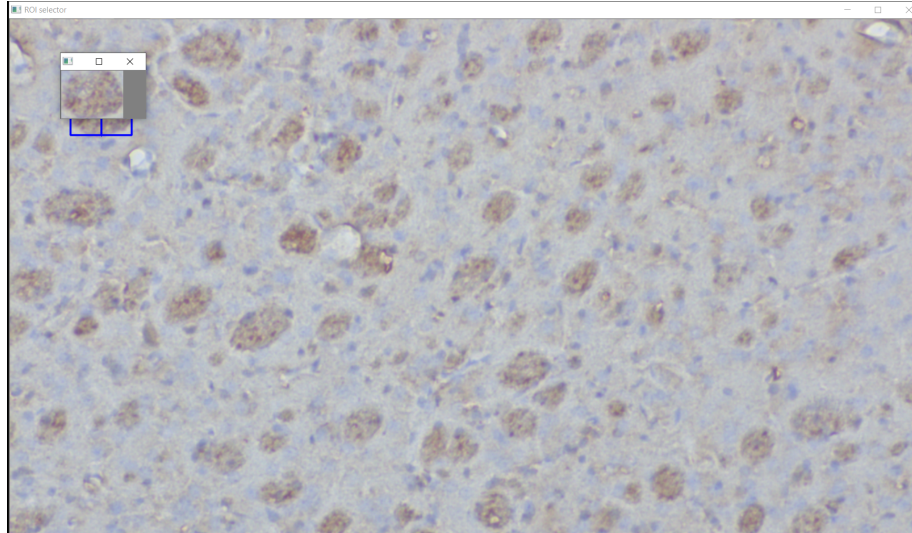


Figura 6.9: Objeto recortado.

10. Cerrar la ventana del objeto recortado.

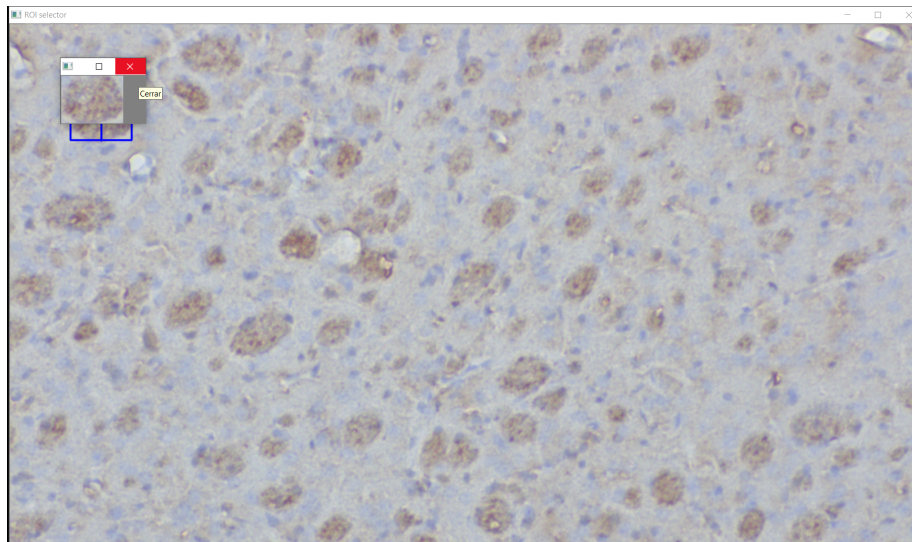


Figura 6.10: Cerrar objeto recortado.

11. Cerrar ventana de la imagen seleccionada.

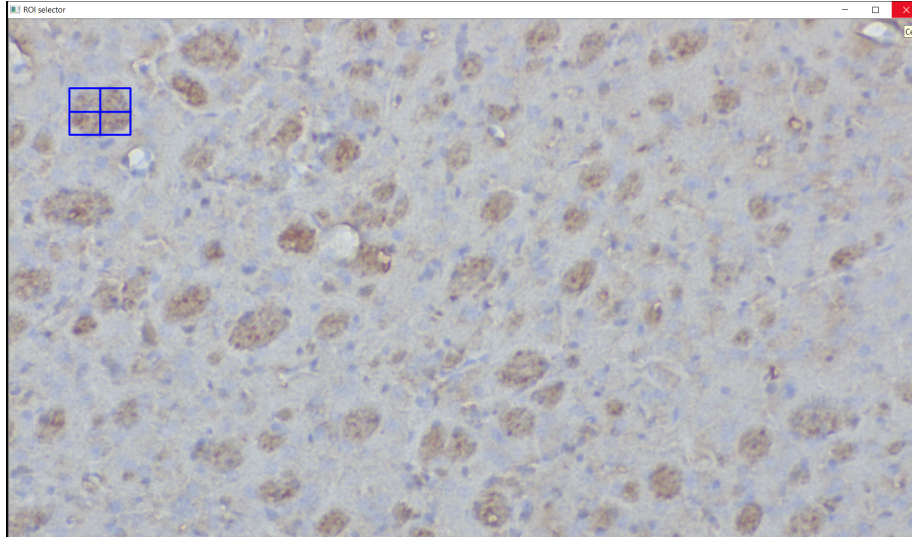


Figura 6.11: Cerrar ventana de la imagen seleccionada.

12. Esperar a que se muestre el resultado. Se abrirá una ventana la cual muestra con un recuadro rojo el objeto seleccionado y con recuadros amarillos los más parecidos a este.

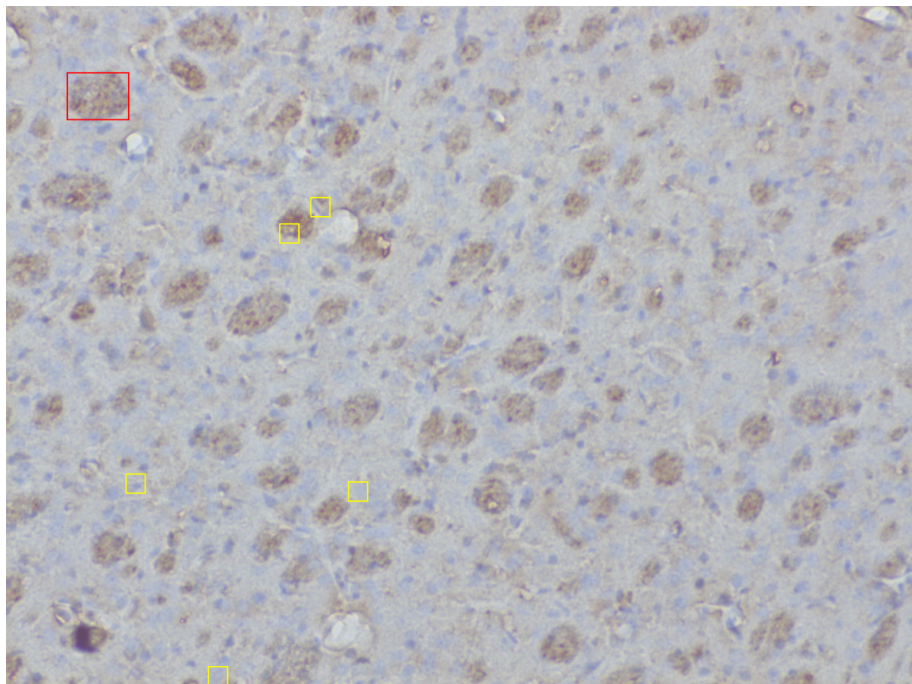


Figura 6.12: Resultado.

7

CONCLUSIONES Y TRABAJO A FUTURO

Conclusiones

Hemos presentado una metodología para estudiar las células tipo glia en tejido cerebral a partir de sus atributos de textura, dentro de una imagen obtenida experimentalmente en microscopio, de tal forma que este enfoque no ha sido reportado en la literatura y este es nuestro principal aporte.

A partir de estos primeros resultados vemos que sí es posible poder brindar un apoyo para la localización de este tipo de células dentro de las imágenes sin importar si es de tipo campo claro o fluorescencia, ya que no impacta el color y sus variaciones dado que hacemos uso de la textura y no del color como atributos para conformar el patrón de caracterización de nuestro objeto de estudio. Se observa que con los tres atributos CBIR el algoritmo es capaz de encontrar objetos parecidos al seleccionado por el usuario. Se observa que con los tres atributos CBIR el algoritmo es capaz de encontrar objetos parecidos al seleccionado por el usuario.

A través de la Inteligencia Artificial se pudo apoyar el análisis de imágenes de tejido cerebral, localizando las células gliales más parecidas a la que señalaba el especialista, dándole una herramienta rápida para poder analizar un número más amplio de imágenes al día, ya que normalmente sin esta herramienta a lo más analizan solo 3 imágenes, por el cansancio ocular que producen estas.

Con ayuda del procesamiento digital de imágenes, fue posible mejorar la calidad de estas para así encontrar las características necesarias para poder señalar los objetos más parecidos a estas, y con la ayuda del especialista, si se encuentra un número determinado de objetos en dicha imagen, se puede determinar si esta enferma o no la glía.

A partir de una cantidad 'relevante' para el profesional de la salud, es indicador de una enfermedad en el paciente, la metodología desarrollada apunta hacia automatizar la detección de este tipo de células de interés como ROI de la imagen de estudio.

Se observó que el uso de cómputo paralelo permitió la mejora del rendimiento en un 38.85 %.

Se hizo un artículo para COMIA 2020, con ponencia virtual y publicación del artículo aceptado próximo a aparecer en línea.

Trabajo a futuro

- Se espera que en la ROI seleccionada por el usuario en ella se haga un preprocesamiento para solo encerrar el contorno deseado con un algoritmo de detección automática y así no se busquen regiones que no sean de nuestro interés, para esto se pretende aplicar un procesamiento morfológico y en base a esto se pretende tener una nueva capa de procesamiento automático de la imagen.
- Afinar con técnicas de IA la localización automatizada de posibles ROI's.
- Analizar imágenes de forma masiva con técnicas de Deep Learning.
- Estudiar la posibilidad de extender nuestra metodología en el estudio de la pandemia del COVID-19, a partir de datos o imagenología médica.
- Analizar la estructura de otras arquitecturas paralelas.

8

GLOSARIO

Tema	Definición	Pág
Content-based Image Retrieval	Es un sistema de búsqueda para recuperar imágenes basándose en su contenido, refiriéndose en su contexto a colores, figuras, texturas o cualquier otra información que pueda derivarse de la propia imagen..	45
HSI	Se define un modelo de color en términos de sus componentes constituyentes. El modelo HSI se representa gráficamente como un cono doble o un doble hexágono. Los dos vértices en el modelo HSI se corresponden con el blanco y el negro, el ángulo se corresponde con el matiz, la distancia al eje con la saturación y la distancia al eje blanco-negro se corresponde a la luminancia.	43
Glía	Las células gliales o neuroglías son células del tejido nervioso, donde actúan en funciones auxiliares, complementando a las neuronas, que son las principales responsables de la función nerviosa.	4
Enfermedad Neurodegenerativa	Las enfermedades neurodegenerativas afectan varias actividades que el cuerpo realiza, como el equilibrio, movimiento, hablar, respirar y funciones del corazón. Muchas de estas enfermedades son genéticas. Las causas pueden ser alcoholismo, un tumor o un ataque cerebrovascular (ACV). Otras causas incluyen toxinas, químicos y virus. Otras veces, las causas se desconocen.	45

Pixel	Es la menor unidad homogénea en color que forma parte de una imagen digital.	43
Concurrencia	Se refiere a la habilidad de distintas partes de un programa, algoritmo, o problema de ser ejecutado en desorden o en orden parcial, sin afectar el resultado final. Los cálculos (operaciones) pueden ser ejecutados en múltiples procesadores, o ejecutados en procesadores separados físicamente o virtualmente en distintos hilos de ejecución.	4
KNN	Es un algoritmo basado en instancia de tipo supervisado de Machine Learning. Puede usarse para clasificar nuevas muestras (valores discretos) o para predecir (regresión, valores continuos). Al ser un método sencillo, es ideal para introducirse en el mundo del Aprendizaje Automático. Sirve esencialmente para clasificar valores buscando los puntos de datos “más similares” (por cercanía) aprendidos en la etapa de entrenamiento y haciendo conjeturas de nuevos puntos basado en esa clasificación.	4
Microglía	Son células neurogliales del tejido nervioso con capacidad fagocitaria y de soporte, que forman el sistema inmunitario del sistema nervioso central (SNC). Se originan durante el desarrollo a partir de precursores mesenquimales que, independientemente de la circulación sanguínea, penetran en el parénquima cerebral. Durante la vida adulta se ha propuesto que se pueden renovar por división in situ, a partir de precursores de la médula ósea, o desde monocitos que alcanzan el sistema nervioso a través de la sangre.	4

BIBLIOGRAFÍA

- [1] Cesar Benavides Alvarez. Sistema no-supervisado de clasificación de rostros con técnicas basadas en cbir. *Universidad Autónoma Metropolitana, Unidad Iztapalapa*, 2015.
- [2] C. N. Barber and D. M. Raben. Lipid Metabolism Crosstalk in the Brain: Glia and Neurons. 2019.
- [3] C. Benavides, J. Villegas, G. Roman, and C. Aviles. Face classification by local texture analysis through cbir and surf points. *IEEE Latin America Transactions*, 14(5):2418–2424, 2016.
- [4] Madhuri G Bhamare and D S Patil. Automatic Blood Cell Analysis By Using Digital Image Processing: A Preliminary Study. *International Journal of Engineering Research & Technology (IJERT)*, 2(9):3137–3141, 2013.
- [5] J. Bahney C. S. von Bartheld and S. Herculano-Houzel. The search for true numbers of neurons and glial cells in the human brain: A review of 150 years of cell counting. 2016.
- [6] Juan Villegas Cortez. Identificación de tipos de letras master in computer science thesis. *Universidad Autónoma Metropolitana, Unidad Azcapotzalco. Av. San Pablo 180, Col. Reynosa Tamaulipas, Deleg. Azcapotzalco, CP 02200, México DF. Jun. 2005.*, 2005.
- [7] O. Instituto Nacional de Neurología y Neurocirugía (Mexico) and Archivos de neurociencias L. MASSIEU. Instituto Nacional de Neurología y Neurología Manuel Velasco Suar. 2004.
- [8] Zivot Tecnología e innovación. <http://www.zivot.com.co/2/interes-ver>. 2020.
- [9] K. Li et al. Reactive Astrocytes in Neurodegenerative Diseases. 2019.
- [10] M. P. Boks F. A. J. Gigase, G. J. L. J. Snijders and L. D. de Witte. Neurons and glial cells in bipolar disorder: a systematic review of postmortem brain studies of cell number and size. 2019.
- [11] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Qian Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker. Query by image and video content: the qbic system. *Computer*, 28(9):23–32, 1995.
- [12] Keinosuke Fukunaga. *Introduction to statistical pattern recognition (2nd ed.)*. Academic Press Professional, Inc., San Diego, CA, USA, 1990.

- [13] Leica Microsystems GmbH. <https://www.leica-microsystems.com/es/aplicaciones/ciencias-biologicas/fluorescencia/>. Corporate Identity Communications, 2020.
- [14] R. C. Gonzalez and R. E. Richard E. Woods, Digital image processing. Prentice Hall, 2002.
- [15] Rafael C. Gonzalez and Richard E. (Richard Eugene) Woods. *Digital image processing*. Prentice Hall, 2002.
- [16] Diagnostic imaging. <https://bidi.uam.mx:6402/levels/collegiate/article/diagnostic-imaging/30251>. Encyclopædia Britannica, 2020.
- [17] J. Villegas-Cortez J. F. Serrano-Talamantes, C. Avilés-Cruz and J. H. Sossa-Azuela. Self organizing natural scene image retrieval. 2013.
- [18] B.Schiele J.Vogel. *Semantic modeling of natural scenes for content-based image retrieval*, volume 72. International Journal of Computer Vision, 2007.
- [19] A. Can A. Madabhushi N. M. Rajpoot M. N. Gurcan, L. E. Boucheron and B. Yener. Histopathological image analysis: a review. 2009.
- [20] Rubén Orozco Morales, Juan Valentín, and Lorenzo Ginori. *Cell Microscopy Imaging : A Review on Digital Image Processing Applications*. Number November. 2015.
- [21] V. S. Nalwa. A Guided Tour of Computer Vision. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc. 1993.
- [22] G. Olague. Evolutionary Computer Vision: The First Footprints. 2016.
- [23] Daniel Reyes H., Larissa Bulavina, and Tatyana Pivneva. La glia el pegamento de las ideas. *Ciencia*, (May):12–18, 2014.
- [24] Elías García Santillán. Detección y clasificación de objetos dentro de un salón de clases empleando técnicas de procesamiento digital de imágenes. 2008.
- [25] G. Sartori. Homo videns: la sociedad teleridigida. *Taurus*, 1998.
- [26] Brain scanning. <https://bidi.uam.mx:6402/levels/collegiate/article/brain-scanning/16179>. Encyclopædia Britannica, 2020.

ANEXO 1

Listado del archivo del programa “Principal” donde se llaman todas las clases del proyecto.

```
1 #librerias
2 from Preprocesamiento import Preprocesamiento
3 from SeleccionROI import SeleccionROI
4 from Emparrillado import Emparrillado
5 from ClasicadorKnn import ClasicadorKnn
6 from PIL import Image
7 import time
8 import math
9 import csv
10 import cv2
11 import numpy as np
12 from scipy.stats import kurtosis
13 from scipy.stats import skew
14
15 #Main
16 #Paso 1:Preprocesamiento de las im genes
17
18 print("Soy principal")
19 num_img=10
20
21 #preprocesamiento = Preprocesamiento(num_img,"C:/Users/Karla/Desktop/
    proyecto/Img_prueba/","im",".tif")
22 #preprocesamiento.filtroGaussiano()
23 #preprocesamiento.guardarImagenes("FG","C:/Users/Karla/Desktop/proyecto/
    FGaussiano/",".tif")
24 #preprocesamiento.ecualizacion()
25 #preprocesamiento.guardarImagenes("Ec","C:/Users/Karla/Desktop/proyecto/
    Ecualizacion/",".tif")
26 #preprocesamiento.RGBHSI()
27 #preprocesamiento.guardarImagenes("HSI","C:/Users/Karla/Desktop/proyecto
    /HSI/",".tif")
28
29
30 seleccionroi = SeleccionROI()
31 #numero de imagen seleccionada
32 numero,ruta = seleccionroi.seleccionarImagen()
33
34 #selecciona la imagen tipo TIF que es la que se muestra al usuario
35 coord1, coord2, coord3, coord4 =seleccionroi.recortaROI(ruta+"Img_prueba
    /im"+numero+".tif")
36 #se selecciona la imagen tipo HSI que es con la que se va a trabajar
37 imagenHSI = seleccionroi.buscaImagenHSI(ruta+"HSI/","HSI",numero,".tif")
```

```
38 vector = seleccionroi.vectorCBIR(imagenHSI,coord1, coord2, coord3,
    coord4)
39
40
41 emparrillado = Emparrillado()
42
43 #Se hace un emparrillado regular de 100 puntos con subventanas de
    tama o 10
44 #incaja tiene el identificador de cada cajita de tama o 10, junto con
    sus 4 coordenadas
45 #instancias son 100 en este caso por las 100 suventanas o cajitas
46 #instancias, infcaja = emparrillado.emparrilladoRegular(100,imagenHSI
    ,10)
47
48 knn = ClasicadorKnn()
49
50
51 #imagen = cv2.imread("C:/Users/Karla/Desktop/proyecto/Img_prueba/im"+
    numero+".tif")
52 #knn.vectoresCBIR(vector,instancias,100,infcaja,imagen,5)
53 #print(instancias.shape)
54 #print(infcaja)
55 tiempoIn = time.time()
56 instancias, infcaja = emparrillado.emparrilladoAleatorio(300,imagenHSI
    ,30)
57 imagen = cv2.imread(ruta+"Img_prueba/im"+numero+".tif")
58 imagen = knn.Colorea(imagen,coord1, coord2, coord3, coord4,(0,0,255))
59 knn.vectoresCBIR(vector,instancias,100,infcaja,imagen,5)
60
61
62 cv2.imwrite(ruta+"resultado/im"+numero+".tif",imagen)
63 tiempoFin = time.time()
64 print('El Proceso Tardo: ', (tiempoFin - tiempoIn)/60, 'Minutos')
65
66 #print(instancias.shape)
67 #print(infcaja)
```

Clase donde se realiza el preprocesamiento a las imágenes (Filtro Gaussiano, Ecuación y Conversión de RGB a HSI).

```
1 from PIL import Image, ImageOps
2
3 import time
4 import math
5 import csv
6 import cv2
7 import numpy as np
8
9
10 class Preprocesamiento:
11     #C:/Users/Karla/Desktop/proyecto/Img_prueba/im"
12     #constructor
13     def __init__(self,num_img,ruta,nombre,extension):
14         self.num_img = num_img
15         self.ruta = ruta
16         self.nombre = nombre
17         self.extension = extension
18         self.imagenes = self.carga_imagenes(num_img,ruta,nombre,extension)
19
20     def carga_imagenes(self,num_img,ruta,nombre,ext):
21         listaImagenes = []
22         for i in range(num_img):
23             image = cv2.imread(ruta+nombre+str(i)+ext)
24             #cv2.imshow("Imagen", image)
25             #cv2.waitKey(0)
26             listaImagenes.append(image)
27         return listaImagenes
28
29
30     def filtroGaussiano(self):
31         imagenFG = []
32         for i in range(self.num_img):
33             imagenFiltrada = cv2.GaussianBlur(self.imagenes[i],(3,3),0);
34             imagenFG.append(imagenFiltrada)
35         self.imagenes=imagenFG
36
37
38
39     def ecualizacion(self):
40         #https://code.tutsplus.com/es/tutorials/histogram-equalization-in-
41         #python--cms-30202
42         imagenE = []
43         for i in range(self.num_img):
44             img_to_yuv = cv2.cvtColor(self.imagenes[i],cv2.COLOR_BGR2YUV)
45             img_to_yuv[:, :, 0] = cv2.equalizeHist(img_to_yuv[:, :, 0])
46             hist_equalization_result = cv2.cvtColor(img_to_yuv, cv2.
47             COLOR_YUV2BGR)
48             imagenE.append(hist_equalization_result)
49         self.imagenes=imagenE
```

```
50 def RGBHSI(self):
51     imagenHSI=[]
52     for i in range(self.num_img):
53         HSI =cv2.cvtColor(self.imagenes[i], cv2.COLOR_RGB2HLS)
54         imagenHSI.append(HSI)
55     self.imagenes=imagenHSI
56
57 def guardarImagen(self,imagen,nombre,ruta,extension):
58     cv2.imwrite(ruta+nombre+extension,imagen)
59
60
61 def guardarImagenes(self,nombre,ruta,extension):
62     for i in range(self.num_img):
63         self.guardarImagen(self.imagenes[i],nombre+str(i),ruta,extension)
64
65 def dameImagenesTratadas(self):
66     return self.imagenes
```


Clase en la que se recorta el objeto seleccionado por el usuario, se convierte en HSI, se saca la media, desviación estándar y homogeneidad.

```
1 from PIL import Image, ImageOps
2
3 import time
4 import math
5 import csv
6 import cv2
7 import numpy as np
8 from tkinter import Tk
9 from tkinter.filedialog import askopenfilename
10
11 class SeleccionROI:
12
13
14
15
16
17 #Para saber el numero de la imagen que selecciono en RGB
18 def seleccionarImagen(self):
19
20     Tk().withdraw() # we don't want a full GUI, so keep the root window
    from appearing
21
22     filename = askopenfilename() # show an "Open" dialog box and return
    the path to the selected file
23
24     ruta = self.dameRuta(filename)
25
26     tamaño = len(filename) #el tamaño de la cadena
27     print("tamaño:", tamaño)
28     indice = filename.find("im") #encuentra la palabra im en la ruta y
    regresa el indice
29     print("indice:", indice)
30     #para extraer el número de la imagen seleccionada
31     if tamaño-indice==7: #si es un dígito
32         número=filename[indice+2]
33         print("número:", número)
34     else:
35         if tamaño-indice==8: #si es de dos dígitos
36             número=filename[indice+2:indice+4]
37             print("número:", número)
38         else:
39             número=filename[indice+2:indice+5] #si es de tres dígitos
40             print("número:", número)
41
42     return número, ruta
43
44 def dameRuta(self, filename):
45     tamaño = len(filename) #el tamaño de la cadena
46     indice = filename.find("Img_prueba") #encuentra la palabra en la
    ruta y regresa el indice
47     ruta = filename[:indice]
```

```

48     print("ruta:", ruta)
49     return ruta
50
51
52 def recortaROI(self, filename):
53     imagen = cv2.imread(filename)
54     #r=[x0,y0,x1,y1]
55     r = cv2.selectROI(imagen)
56
57     #imcrop I imagen[y0:y0+y1, x0:+x0+x1]
58     imCrop = imagen[int(r[1]):int(r[1]+r[3]), int(r[0]):int(r[0]+r[2])]
59     cv2.imshow("Image", imCrop)
60     cv2.waitKey(0)
61     return r[1], r[1]+r[3], r[0], r[0]+r[2]
62
63
64 #Para buscar la imagen en HSI
65 def buscaImagenHSI(self, ruta, nombre, num_img, ext):
66
67     imagen = cv2.imread(ruta+nombre+str(num_img)+ext)
68     #cv2.imshow("Imagen", imagen)
69     #cv2.waitKey(0)
70     return(imagen)
71
72
73 def vectorCBIR(self, imagenHSI, coord1, coord2, coord3, coord4):
74     instancias= []
75
76     recorte = imagenHSI[int(coord1):int(coord2), int(coord3):int(coord4)
77 ]
78     anchoImagen, altoImagen, capas = recorte.shape
79     nuevaImagen = recorte.reshape(capas, anchoImagen, altoImagen) #
80     reordeno la informacion
81
82     H = nuevaImagen[0] #separo las capas
83     S = nuevaImagen[1]
84     I = nuevaImagen[2]
85
86     hH, hS, hI= self.Homo(H, S, I, anchoImagen, altoImagen)
87
88     """ Atibutos para H (media, desviacion y homogeniedad) """
89     atributos = [np.mean(H), np.std(H), hH]
90
91     """ Atibutos para S (media, desviacion y homogeniedad) """
92     atributos.extend([np.mean(S), np.std(S), hS])
93
94     """ Atibutos para I (media, desviacion y homogeniedad) """
95     atributos.extend([np.mean(I), np.std(I), hI])
96
97     instancias.append(atributos)
98
99     return instancias

```

```
100
101 def Pij(self,cajaH, cajaS, cajaI):
102
103     sumaTotalH = np.sum(cajaH) #suma de todos los pixeles de mi
104     subventana H
105     sumaTotalS = np.sum(cajaS) #suma de todos los pixeles de mi
106     subventana S
107     sumaTotalI = np.sum(cajaI) #suma de todos los pixeles de mi
108     subventana I
109
110     probabilidadH = np.divide(cajaH,sumaTotalH)
111     probabilidadS = np.divide(cajaS,sumaTotalS)
112     probabilidadI = np.divide(cajaI,sumaTotalI)
113
114     return probabilidadH, probabilidadS, probabilidadI
115
116 def Homo(self,cajaH, cajaS, cajaI,ancho, alto):
117     PIJH, PIJS, PIJI = self.Pij(cajaH, cajaS, cajaI)
118     homogeniedadH = 0
119     homogeniedadS = 0
120     homogeniedadI = 0
121     for i in range(0,ancho):
122         for j in range(0,alto):
123             homogeniedadH+=PIJH[i,j]/(1+pow(i-j,2))
124             homogeniedadS+=PIJS[i,j]/(1+pow(i-j,2))
125             homogeniedadI+=PIJI[i,j]/(1+pow(i-j,2))
126
127     return homogeniedadH, homogeniedadS, homogeniedadI
128
129
130
131
```

Clase donde se realiza el emparrillado de 100 puntos regulares y 300 puntos aleatorios a la imagen y se extraen los atributos de cada subventana.

```

1 from PIL import Image, ImageOps
2
3 import time
4 import math
5 import csv
6 import cv2
7 import numpy as np
8 import random
9
10
11 class Emparrillado:
12
13
14     def dameMatrizConcurrenciaHSI(self, cajaH, cajaS, cajaI):
15         angulo = 0
16         distancia = 1
17
18         tamEspacioH = self.maxMax(cajaH) + 1 # [0-360]
19         tamEspacioS = self.maxMax(cajaS) + 1 # saturation [0, 1]
20         tamEspacioI = self.maxMax(cajaI) + 1 # intensity [0,1]
21
22         mCH=np.zeros((tamEspacioH,tamEspacioH))
23         mCS=np.zeros((tamEspacioS,tamEspacioS))
24         mCI=np.zeros((tamEspacioI,tamEspacioI))
25
26         contador = 0
27         memoria = []
28
29         for i in range(len(cajaH)):
30             for j in range(len(cajaH[0])):
31                 if(j+distancia<len(cajaH[0])):
32                     pivoteH = cajaH[i][j]
33                     subpivoteH = cajaH[i][j+distancia]
34                     mCH[pivoteH][subpivoteH] += 1
35
36                     pivoteS = cajaS[i][j]
37                     subpivoteS = cajaS[i][j+distancia]
38                     mCS[pivoteS][subpivoteS] += 1
39
40                     pivoteI = cajaI[i][j]
41                     subpivoteI = cajaI[i][j+distancia]
42                     mCI[pivoteI][subpivoteI] += 1
43
44
45
46         return mCH, mCS, mCI
47
48
49
50     def Pij(self, cajaH, cajaS, cajaI):
51

```

```

52     sumaTotalH = np.sum(cajaH) #suma de todos los pixeles de mi
subventana H
53     sumaTotalS = np.sum(cajaS) #suma de todos los pixeles de mi
subventana S
54     sumaTotalI = np.sum(cajaI) #suma de todos los pixeles de mi
subventana I
55
56     probabilidadH = np.divide(cajaH,sumaTotalH)
57     probabilidadS = np.divide(cajaS,sumaTotalS)
58     probabilidadI = np.divide(cajaI,sumaTotalI)
59
60     return probabilidadH, probabilidadS, probabilidadI
61
62 def Homo(self,cajaH, cajaS, cajaI):
63     mCH, mCS, mCI = self.dameMatrizConcurrenciaHSI(cajaH, cajaS, cajaI)
# Calcula matriz de concurrencia
64     #Tamanos de las matrices de concurrencia
65     anchoH, altoH = mCH.shape
66     anchoS, altoS = mCS.shape
67     anchoI, altoI = mCI.shape
68     #print("mCH Ancho= ",anchoH, " Alto= ", altoH)
69     #print("mCS Ancho= ",anchoS, " Alto= ", altoS)
70     #print("mCI Ancho= ",anchoI, " Alto= ", altoI)
71
72     pijH, pijS, pijI = self.Pij(mCH, mCS, mCI)
73
74     homogeniedadH = 0.0
75     for i in range(0,anchoH):
76         for j in range(0,altoH):
77             homogeniedadH+=pijH[i,j]/(1+pow(i-j,2))
78
79     homogeniedadS = 0.0
80     for i in range(0,anchoS):
81         for j in range(0,altoS):
82             homogeniedadS+=pijS[i,j]/(1+pow(i-j,2))
83
84     homogeniedadI = 0.0
85     for i in range(0,anchoI):
86         for j in range(0,altoI):
87             homogeniedadI+=pijI[i,j]/(1+pow(i-j,2))
88
89     return homogeniedadH, homogeniedadS, homogeniedadI
90
91 def maxMax(self, matriz):
92     maximo = -1
93     for lista in matriz:
94         for valor in lista:
95             if maximo <= valor:
96                 maximo = valor
97     return maximo
98
99
100
101 #empariillado regular de "100" puntos con subventanas de 10x10

```

```

102 def emparrilladoRegular(self,numPuntos,imagen,tama oEmparrillado):
103     instancias=[] #guardo las características de cada ventana (media,
104     desviacion y homogeneidad)*3
105     cont = 0 #para poner le identificador a cada caja
106     infCaja = [] #donde se guarda el identificador de cada caja y sus 4
107     coordenadas
108
109     anchoImagen, altoImagen, capas = imagen.shape #saco las propiedades
110     de la imagen
111     ancho = round(anchoImagen/(tama oEmparrillado+1)) #para obtener el
112     ancho de cada subventana
113     alto = round(altoImagen/(tama oEmparrillado+1)) #para obtener el
114     alto de cada subventana
115
116     nuevaImagen = imagen.reshape(capas,anchoImagen,altoImagen) #reordeno
117     la informaci n
118     H = nuevaImagen[0] #separo las capas
119     S = nuevaImagen[1]
120     I = nuevaImagen[2]
121
122     for i in range (0, tama oEmparrillado): #para recorrer el
123     emparrillado de 100
124         for j in range (0, tama oEmparrillado):
125
126             cajaH = H[ancho*(i+1):(ancho *(i+1))+10, alto*(j+1):(alto*(j+1))
127             +10]
128             cajaS = S[ancho*(i+1):(ancho *(i+1))+10, alto*(j+1):(alto*(j+1))
129             +10]
130             cajaI = I[ancho*(i+1):(ancho *(i+1))+10, alto*(j+1):(alto*(j+1))
131             +10]
132
133             hH, hS, hI= self.Homo(cajaH, cajaS, cajaI)
134
135             """ Atibutos para H (media,desviacion y homogeneidad)"""
136             atributos = [np.mean(cajaH), np.std(cajaH), hH]
137
138             """ Atibutos para S (media,desviacion y homogeneidad)"""
139             atributos.extend([np.mean(cajaS), np.std(cajaS), hS])
140
141             """ Atibutos para I (media,desviacion y homogeneidad)"""
142             atributos.extend([np.mean(cajaI), np.std(cajaI), hI])
143
144             instancias.append(atributos)
145
146             #guarda informacion de caja (coordenadas de cada cajita)
147             infCaja.append([cont,ancho*(i+1),(ancho *(i+1))+10, alto*(j+1),
148             (alto*(j+1))+10])
149             cont+=1
150
151     return np.array(instancias),np.array(infCaja)

```

```
145
146 def colorear(self, imagen ,superiorI, superiorD, inferiorI, inferiorD,
147 color):
148     grosor = 2
149     #Superior
150     for x in range(superiorI, superiorD+1):
151         for y in range( inferiorI, inferiorD+1):
152             if x < superiorI + grosor or x > superiorD - grosor or y <
153 inferiorI+grosor or y > inferiorD-grosor:
154                 imagen.itemset( (x, y, 0) , color[0])
155                 imagen.itemset( (x, y, 1) , color[1])
156                 imagen.itemset( (x, y, 2) , color[2])
157     return imagen
158
159 def damePuntoAleatorios(self, ancho, alto, numPuntos, margen):
160     puntosAleatorios = []
161     for i in range(numPuntos):
162         x = random.randint(0, ancho-1-margen)
163         y = random.randint(0, alto-1-margen)
164         puntosAleatorios.append( [i, x, y] )
165     return puntosAleatorios
166
167
168 def emparrilladoAleatorio(self, numPuntos, imagen, tamañoSubV):
169     instancias=[]
170     cont = 0
171     infCaja = []
172     margen = tamañoSubV
173
174     anchoImagen, altoImagen, capas = imagen.shape
175
176     puntos = self.damePuntoAleatorios( anchoImagen, altoImagen,
177 numPuntos, margen )
178     anchoCaja = margen
179     altoCaja = margen
180
181     nuevaImagen = imagen.reshape( capas, anchoImagen, altoImagen)
182     H = nuevaImagen[0]
183     S = nuevaImagen[1]
184     I = nuevaImagen[2]
185
186     for punto in puntos:
187         print("Estoy en el punto:", punto)
188         x = punto[1]
189         y = punto[2]
190         cajaH = H[x:x+anchoCaja , y:y+altoCaja]
191         cajaS = S[x:x+anchoCaja , y:y+altoCaja]
192         cajaI = I[x:x+anchoCaja , y:y+altoCaja]
193
194         hH, hS, hI= self.Homo(cajaH, cajaS, cajaI)
195
```

```
196     """ Calculos para H"""
197     atributos = [np.mean(cajaH), np.std(cajaH), hH]
198
199     """ Calculos para S"""
200     atributos.extend([np.mean(cajaS), np.std(cajaS), hS])
201
202     """ Calculos para I"""
203     atributos.extend([np.mean(cajaI), np.std(cajaI), hI])
204
205     instancias.append(atributos)
206
207     #GUARDA INFO DE CAJAS
208     infCaja.append( [ cont, x, x+anchoCaja, y, y+altoCaja ] )
209     cont += 1
210
211
212
213
214     return np.array(instancias) , np.array(infCaja)
215
216
217
218
219
220
```


Clase en la que se lleva a cabo el clasificador de K -vecinos más cercanos y da los objetos más parecidos al objeto seleccionado.

```

1 from PIL import Image, ImageOps
2
3 import time
4 import math
5 import csv
6 import cv2
7 import numpy as np
8 import random
9
10
11 class ClasicadorKnn:
12
13
14     def vectoresCBIR(self, vectorRecorte, vectorEmparrillado, numIns,
15         infCaja, imagen, numCol):
16         distancias = []
17         for i in range(numIns):
18             distancia = self.euclideana(vectorRecorte[0], vectorEmparrillado[i
19         ])
20             distancias.append([distancia, i])
21             orden = sorted(distancias, key=lambda student: student[0])
22             ganadoras = self.dameGanadoras(orden, numCol)
23
24             for i in range(numCol):
25                 imagen = self.Colorea(imagen, infCaja[ganadoras[i]][1], infCaja[
26                 ganadoras[i]][2], infCaja[ganadoras[i]][3], infCaja[ganadoras[i
27                 ]][4], (0, 255, 255))
28
29             cv2.imshow("imagenColoreada", imagen)
30             cv2.waitKey(0)
31
32     def euclideana(self, x, y):
33         distancia = 0
34         for i in range(len(x)):
35             distancia += pow(x[i]-y[i], 2)
36         return math.sqrt(distancia)
37
38     def dameGanadoras(self, lista, numGanadoras):
39         ganadoras = []
40         for i in range(numGanadoras):
41             ganadoras.append(lista[i][1])
42         return ganadoras
43
44     def Colorea(self, imagen, superiorIzq, superiorDer, infeiorIzq,
45         inferiorDer, color):
46         grosor = 2
47         for x in range(superiorIzq, superiorDer+1):
48             for y in range(infeiorIzq, inferiorDer+1):
49                 if(x<superiorIzq+grosor or x>superiorDer-grosor or y<infeiorIzq+

```

```
    grosor or y>inferiorDer-grosor):
47     imagen.itemset((x,y,0),color[0])
48     imagen.itemset((x,y,1),color[1])
49     imagen.itemset((x,y,2),color[2])
50     return imagen
```

ANEXO 2

Artículo de investigación sometido al Congreso Mexicano de Inteligencia Artificial COMIA 2020, celebrado en las fechas del 5 al 7 de agosto del 2020, bajo la organización de la Sociedad Mexicana de Inteligencia Artificial (SMIA), y la Universidad de Cd Juárez, Chihuahua. El artículo fue sometido a revisión ciega entre pares y aceptado para publicarse en el número especial de la revista indexada Research in Computing Science, del Centro de Investigación en Computación, del Insituto Politécnico Nacional; con la referencia tentativa a completar con la fecha de publicación como:

Karla Mandujano-López et al. *Identificación y localización de células gliales en imágenes de tejido cerebral*. Por publicarse en: Research in Computing Science (indexada en: Latindex, DBPL) Aceptado: 29 de junio, 2020. ISSN 18704069.

Adicionalmente, el trabajo de investigación fue seleccionado para presentarse en ponencia en el marco de Visión por Computadora y Reconocimiento de Patrones.



Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA

ACTA DE EXAMEN DE GRADO

No. 00088

Matrícula: 2183802193

Identificación y localización de células gliales en imágenes de tejido cerebral.



KARLA MANDUJANO LOPEZ
ALUMNA

REVISÓ

MTRA. ROSALIA SERRANO DE LA PAZ
DIRECTORA DE SISTEMAS ESCOLARES

Con base en la Legislación de la Universidad Autónoma Metropolitana, en la Ciudad de México se presentaron a las 11:00 horas del día 1 del mes de diciembre del año 2020 POR VÍA REMOTA ELECTRÓNICA, los suscritos miembros del jurado designado por la Comisión del Posgrado:

DR. LEONARDO TRUJILLO REYES
DR. RENE MACKINNEY ROMERO
DR. JUAN VILLEGAS CORTEZ

Bajo la Presidencia del primero y con carácter de Secretario el último, se reunieron para proceder al Examen de Grado cuya denominación aparece al margen, para la obtención del grado de:

MAESTRA EN CIENCIAS (CIENCIAS Y TECNOLOGIAS DE LA INFORMACION)

DE: KARLA MANDUJANO LOPEZ

y de acuerdo con el artículo 78 fracción III del Reglamento de Estudios Superiores de la Universidad Autónoma Metropolitana, los miembros del jurado resolvieron:

APROBAR

Acto continuo, el presidente del jurado comunicó a la interesada el resultado de la evaluación y, en caso aprobatorio, le fue tomada la protesta.

DIRECTOR DE LA DIVISIÓN DE CBI

DR. JESÚS ALBERTO OCHOA TAPIA

PRESIDENTE

DR. LEONARDO TRUJILLO REYES

VOCAL

DR. RENE MACKINNEY ROMERO

SECRETARIO

DR. JUAN VILLEGAS CORTEZ