



**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

---

**FACULTAD DE INGENIERÍA**

**Sistema de reconocimiento  
automático de neuronas en  
microscopía de  
epifluorescencia**

**TESIS**

Que para obtener el título de  
**Ingeniero en Computación**

**P R E S E N T A**

Fernando González Colín

**DIRECTORA DE TESIS**

Dr. Jimena Olveres Montiel



Ciudad Universitaria, Cd. Mx., 2019

A mis padres por todo su apoyo, confianza y cariño.

## **Agradecimientos**

Agradezco a mis padres por todo lo que han hecho por mí. A mis sinodales que me estuvieron guiando, asesorando, evaluando y alentando. A mis amigos con los que he aprendido, de los que he aprendido y que en muchas ocasiones me han inspirado y motivado a ser mejor. Finalmente, agradezco a los proyectos UNAM PAPIIT IA103119 e IN116917 por los apoyos recibidos.

# Índice general

|   |    |
|---|----|
| Resumen.....                                    | 2  |
| 1. Introducción.....                            | 3  |
| 1.1. Definición del problema.....               | 5  |
| 1.2. Objetivos.....                             | 7  |
| 2. Antecedentes de neurociencias.....           | 8  |
| 2.1. Imagenología de calcio.....                | 13 |
| 2.2. Experimento característico.....            | 14 |
| 2.3. Trabajos relacionados con el problema..... | 17 |
| 3. Procesamiento imágenes digitales.....        | 20 |
| 3.1. Píxel.....                                 | 21 |
| 3.2. Histograma.....                            | 23 |
| 3.3. Filtros.....                               | 28 |
| 3.4. Perfilado.....                             | 34 |
| 3.5. Detección de BLOBS.....                    | 40 |
| 3.6 Análisis multiescala.....                   | 43 |
| 3.7 Morfología matemática.....                  | 48 |
| 4. Definición de requerimientos.....            | 53 |
| 5. Diseño e implementación.....                 | 56 |
| 5.1 Segmentación.....                           | 57 |
| 5.2 Validación.....                             | 63 |
| 6. Pruebas y resultados.....                    | 68 |
| 7. Conclusiones.....                            | 73 |
| 8. Referencias.....                             | 76 |

## Resumen

La comprensión de la mente humana y sus padecimientos es uno de los retos más antiguos a los que se ha enfrentado el ser humano, a la vez que los avances científicos y tecnológicos han permitido desarrollar técnicas, experimentos y teorías cada vez más complejas. Sin embargo, con el desarrollo de nuevas técnicas también han aparecido nuevas dificultades. Aunque la microscopía de epifluorescencia es una técnica novedosa que permite realizar experimentos en tejido cerebral y adquirir videos de la actividad de múltiples neuronas con resolución de célula única, las características de los videos hacen que la identificación de las células nerviosas observadas sea una labor engorrosa para los investigadores lo que implica, principalmente, una desventaja en productividad. En problemas similares de imagenología donde se requiere realizar reconocimiento y segmentación de objetos ha sido aplicado con éxito el procesamiento de imágenes digitales. El desarrollo de nuevos algoritmos y computadoras más potentes han hecho que esta disciplina evolucione rápidamente y tenga utilidad en muchas áreas de estudio, desde la astronomía hasta la medicina. En esta tesis se propone un sistema basado en procesamiento de imágenes digitales para automatizar el proceso de identificar neuronas en videos de microscopía de epifluorescencia y con ello apoyar a los neurocientíficos en sus investigaciones. Esta implementación ha sido pensada para ser robusta al mismo tiempo que se mantiene accesible para los usuarios al no exigir demasiadas prestaciones de cómputo. Además, los cambios de fluorescencia a través del tiempo son empleados para validar los elementos segmentados y proporcionar datos valiosos de la actividad de cada neurona a los usuarios más allá de la segmentación, todo lo anterior sin la necesidad de realizar complejas configuraciones de parámetros, en menor tiempo que el dedicado a hacer la actividad manualmente y con rendimiento equiparable al del experto.

## **1. Introducción**

El ser humano en su inmensa curiosidad ha sido capaz de desvelar la naturaleza de muchos de los fenómenos que le rodean, empezando con aquellos que puede percibir, aunque, su curiosidad no se ha limitado a aquello que le rodea, preguntas sobre cómo escucha, ve, siente, huele, saborea, coordina los movimientos de su cuerpo, aprende, recuerda, olvida, ama, odia y se emociona, lo han llevado a buscar la comprensión de su propio cuerpo y con ello a controlar los padecimientos que le acechan.

Evidencias sugieren que desde la prehistoria, los primeros humanos ya concebían al cerebro como un elemento esencial para la vida. Desde entonces, con el paso de los años su comprensión fue siendo cada vez más profunda. Desde los primeros diagnósticos de daño cerebral en el Antiguo Egipto, pasando por la idea desarrollada en Grecia de que estructuras específicas proporcionan funciones específicas, las disecciones romanas, la anatomía renacentista, la teoría celular, y las primeras técnicas para la observación de células nerviosas hasta llegar a la doctrina de la neurona a finales del siglo XIX, numerosas teorías fueron desarrolladas, sin embargo, muchas no serían probadas hasta la invención del microscopio electrónico en la década de 1950, lo que a su vez desencadenó una revolución científica: la revolución de las neurociencias [1].

Con nuevos métodos de observación y experimentación, teorías fueron desarrolladas, probadas y refutadas, sin embargo, la comunidad científica terminó por darse cuenta que para lograr comprender el sistema nervioso es necesario conocer muchas cosas que están involucradas, desde la estructura molecular del agua hasta el electromagnetismo y teoría de redes. En un escenario así, resultó evidente que la mayor

oportunidad de lograr avances significativos en las investigaciones vendría de un enfoque multidisciplinario [1]; biólogos, químicos, físicos, matemáticos o ingenieros trabajando en conjunto para alcanzar un mismo objetivo, cada uno aportando un enfoque diferente con los cuales tender puentes hacia la comprensión integral del sistema nervioso.

La aparición de nuevas tecnologías ha sido un factor de gran importancia para el desarrollo de las neurociencias pues ha permitido, además de la realización de nuevos experimentos con sofisticados sistemas de observación, realizar el análisis de datos a través de herramientas de cómputo así como generar modelos y simulaciones del sistema nervioso [2]. Entre esas tecnologías se encuentra el procesamiento de imágenes digitales debido a que provee un poderoso enfoque para el análisis de estructuras biológicas. Tradicionalmente este tipo de análisis se ha basado en la intervención humana, lo que puede llevar a juicios subjetivos y con ello aumenta la probabilidad de error, mientras que en un análisis computacional, los juicios se realizan con base en medidas cuantitativas. Además, el procesamiento de imágenes digitales es un área en la que se ha hecho mucha investigación y desarrollo por lo que cuenta con abundantes referencias y fuentes de información para cubrir todos los aspectos que implica que esta disciplina [3].

Existen dos principales motivaciones para la aplicación de procesamiento de imágenes digitales: mejorar la imagen y extraer características de ella [3]. La segunda es la de mayor importancia en neurociencias debido a que esta es la que permite extraer datos importantes para la elaboración de inferencias, teorías o diagnósticos. Sin embargo, ambas motivaciones implican procesar imágenes mediante operaciones matemáticas. De hecho el procesamiento de imágenes digitales tiene sus bases en métodos matemáticos de procesamiento de

señales la cual a su vez ha tenido aplicación en disciplinas como la física, ingeniería, biología, sismología y economía por mencionar algunas [4].

### **1.1. Definición del problema**

En el Departamento de Neurociencia Cognitiva del Instituto de Fisiología Celular (IFC) de la UNAM se realizan estudios de frontera en el área de las neurociencias. Los investigadores utilizan distintos enfoques y abordajes experimentales para alcanzar un mejor entendimiento de los procesos neuronales y cerebrales y eventualmente poder controlar o prevenir algunas de las patologías del sistema nervioso más comunes (por ejemplo, enfermedad de Parkinson, enfermedad de Alzheimer y epilepsia). La complejidad de esta tarea es tal, que requiere de los esfuerzos conjuntos de diversas disciplinas, es decir, para resolver los problemas actuales de las neurociencias se tiene que hacer acopio de todo lo que esté al alcance: matemáticas, lógica, cómputo, biología, física, economía, química, ingeniería, patología, psicología, entre otras disciplinas. Aunque los recientes avances tecnológicos han impulsado enormemente el desarrollo en neurociencias, aún queda mucho trabajo por hacer, es aquí donde la Ingeniería en Computación, con ayuda de su formación académica, debe desempeñar un rol importante en estos avances científicos.

La técnica de imagenología de calcio intracelular permite realizar estudios en cortes de tejido cerebral y observar decenas de neuronas de manera simultánea, además de otorgar resolución de célula única empleando microscopía de epifluorescencia. Los videos que se obtienen en tales experimentos requieren de análisis minuciosos por parte de los investigadores con el fin de extraer información significativa de ellos. Uno de los procesos de mayor demora es la identificación manual de neuronas individuales funcionales durante los experimentos (una neurona funcional es aquella de la que podemos observar cambios de fluorescencia



asociados a actividad neuronal en forma de potenciales de acción). Esta tarea requiere de experiencia y mucho tiempo, dependiendo de la cantidad de datos a procesar, requiere en promedio de una hora para identificar las neuronas en un tramo de experimento de 30 minutos. El trabajo es tedioso debido a que el investigador tiene que revisar cuadro por cuadro del video buscando manchas relativamente pequeñas que muestren cambios en la intensidad de sus píxeles; conforme pasa el tiempo de análisis, la vista comienza a cansarse y así aumenta la probabilidad de omitir regiones de interés, además del error intrínseco humano en este tipo de análisis, por estos motivos resulta necesario el uso de herramientas automatizadas que apoyen en esta labor.

Se han hecho esfuerzos por automatizar el proceso de detección de neuronas en videos de microscopía de epifluorescencia [5, 6, 7]. Sin embargo, esta tarea es difícil debido a que las neuronas presentan alta heterogeneidad en forma y tamaño, bordes difusos y fluorescencia poco uniforme. Además, el tamaño de la población observada varía con cada experimento y hay una significativa presencia de ruido debido a que todos los datos son obtenidos con cámaras que son muy sensibles a la luz. Los métodos tradicionales de procesamiento de imágenes digitales como umbralización, normalización y ecualización han resultado ser insuficientes, mientras que técnicas de aprendizaje profundo requieren de enormes cantidades de muestras etiquetadas a mano para entrenamiento y validación.

En un escenario como el planteado, se propone un proyecto que aborda el problema con técnicas más sofisticadas de procesamientos de imágenes digitales y reconocimiento de patrones, capaz de reconocer, segmentar y localizar automáticamente las neuronas contenidas en vídeos de microscopía de epifluorescencia.

## **1.2. Objetivos**

Desarrollar un sistema capaz de identificar, segmentar y localizar neuronas contenidas en los videos obtenidos de observaciones hechas por microscopía de epifluorescencia. El sistema estará contenido en una biblioteca de programación accesible para los investigadores, de modo que debe ser de fácil uso, su funcionamiento debe depender en lo mínimo de la supervisión humana y los resultados obtenidos deberán estar en un formato adecuado para su fácil interpretación y uso futuro. Además, dado que esta tarea es actualmente desempeñada por los investigadores manualmente, el sistema debe tener un rendimiento similar o superior al del experto para mostrar su utilidad tanto en identificación como en tiempo de ejecución.

## **2. Antecedentes de neurociencias**

El cerebro humano contiene aproximadamente 85 mil millones de neuronas [8], células nerviosas altamente especializadas en distintas funciones biológicas, y cuya principal característica es la generación y propagación de señales eléctricas (potenciales de acción). Estas neuronas sufren cambios metabólicos durante el transcurso de la vida de los individuos de tal manera que continuamente están perdiendo y formando nuevas conexiones dentro del cerebro. Al conectarse unas neuronas con otras surgen complejas redes de comunicaciones de las que se cree emergen funcionalidades tanto sensoriales como cognitivas. De modo que conocer la dinámica neuronal de estas redes neuronales biológicas permitiría entender, detectar y tratar eficientemente distintos desordenes neuronales como la Enfermedad de Parkinson y la Epilepsia [9, 10], entre otros. Para ello es preciso comprender los componentes estructurales de la neurona, los mecanismos que le permiten manipular señales, los patrones de conexión, las relaciones entre patrones de actividad con las diferentes funciones, así como la manera en que las conexiones se ven modificadas por la experiencia y el uso.

Estructuralmente la neurona se compone de cuatro partes: el soma, las dendritas, el axón y las terminales sinápticas (Figura 2.1). El soma es el cuerpo de la neurona, realiza las funciones metabólicas de la célula así como la producción de diferentes enzimas, es el contenedor de las diversas estructuras generales de las células, especialmente contiene en su núcleo el material genético necesario para sintetizar proteínas, tiene forma esférica y representa la parte más ancha de la neurona con un diámetro promedio de 20  $\mu\text{m}$ . Las dendritas son unas estructuras tubulares, delgadas que se extienden desde el soma como ramas de árboles, sirven como las antenas receptoras de la neurona pues a través de ellas la neurona puede recibir señales de otras neuronas. El axón, al igual que las dendritas, es una estructura tubular que se extiende desde

el soma, puede extenderse mucho más que las dendritas y tener una longitud desde 0.1 mm hasta 2 m, también se ramifica y en la punta de cada rama se encuentran las terminales sinápticas, estructuras con forma de botón por las cuales la neurona envía señales a otras células [11, 12, 13].

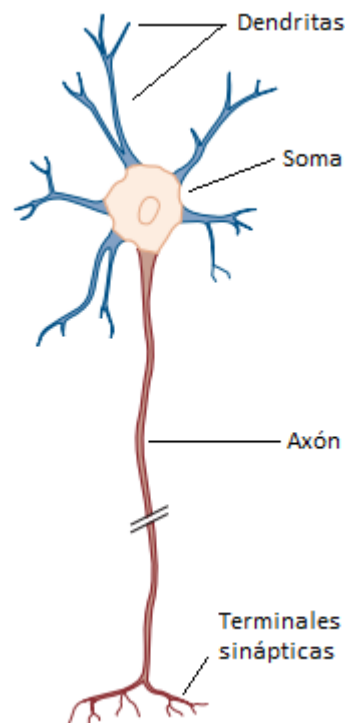


Figura 2.1. Estructura de la neurona. Modificado de [11]

Un elemento muy importante en la estructura de las neuronas es la llamada membrana celular, esta delimita la región que comprende físicamente la neurona con el medio extracelular. Está formada por una bicapa de fosfolípidos con la parte no polar de la molécula orientada hacia el centro, de tal manera que le confiere una alta estabilidad y capacidad de deformación. A lo largo de su superficie contiene una amplia variedad de complejos de proteínas (canales iónicos, receptores ionotrópicos y metabotrópicos) que responden a distintas clases de estímulos (mecánicos, químicos, eléctricos, etc.) para controlar su apertura y cierre y de esta manera regular el flujo de iones, principalmente  $\text{Na}^+$ ,  $\text{Ca}^{2+}$ ,  $\text{Mg}^{2+}$  y  $\text{Cl}^-$ , hacia el interior y exterior de la célula. En ese sentido es importante

la diferencia de potencial que existe entre el espacio intracelular y el medio extracelular. Con el medio extracelular como referencia, la diferencia de potencial entre ambos elementos es de aproximadamente -70 mV, este es el conocido potencial de reposo de las neuronas.

Los iones antes mencionados pueden moverse hacia adentro o hacia afuera de la membrana celular de distintas formas, cuando esto sucede se puede medir una corriente eléctrica a través de la membrana. A medida que los iones positivos fluyen hacia el exterior de la membrana o los iones negativos fluyen hacia su interior a través de los canales iónicos, la diferencia de potencial se vuelve más negativa en un proceso llamado hiperpolarización. En cambio, si el flujo de iones provoca que el potencial eléctrico tienda a ser menos negativo el proceso es llamado despolarización. Dependiendo del proceso causado por el flujo de iones la sinapsis puede ser excitatoria en el caso de la hiperpolarización o inhibitoria en el caso de la despolarización.

Cuando la membrana celular está lo suficientemente despolarizada (supera un umbral) se genera un potencial de acción, un evento de aproximadamente 1-2 ms de duración que eleva el voltaje de la neurona hasta niveles positivos de 20-60 mV (Figura 2.2). Cada neurona presenta distintos patrones de generación de potenciales de acción dependiendo de los distintos tipos de canales y receptores que posee en su membrana. Una vez generado, el potencial de acción viaja por el axón de la neurona hasta la terminación sináptica donde regularmente se comunica con otra neurona [12].

La conexión entre dos neuronas recibe el nombre de sinapsis y se compone en lo general de tres elementos: presinapsis, espacio sináptico y postsinapsis (Figura 2.3). La presinapsis comprende el punto de salida de la señal de una neurona emisora a través de la terminal sináptica donde sufre la transducción de señal eléctrica en una señal química, la cual se

propaga a través del espacio sináptico y es recibida por la postsinapsis, el punto de entrada de la señal en la neurona receptora, en esta parte la señal es reconvertida en señal eléctrica y continua su recorrido a través de la neurona receptora. Aunque lo común es que las terminales sinápticas de una neurona hagan conexión con otras neuronas a través de sus dendritas también ocurre, con menor frecuencia, que la conexión se da entre las terminales simpáticas y el axón o el soma de la neurona receptora, incluso entre los mismos axones. En este tipo de sinapsis (sinapsis química) nunca hay contacto físico entre dos neuronas, la transmisión de la señal siempre se da a través del espacio sináptico que hay entre ambas células [1].

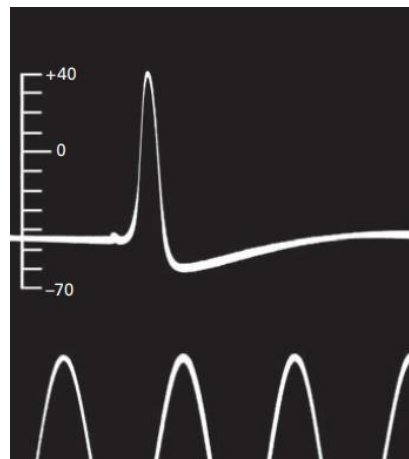


Figura 2.2. Primer registro de un potencial de acción, hecho el 1939 por Hodgkin y Huxley donde el eje vertical indica la magnitud del potencial de acción en mV y el horizontal indica el tiempo y los pulsos mostrados en la parte inferior de la imagen se encuentran separados por 2 ms. Modificado de [11]

Las señales químicas que se propagan a través del espacio sináptico, son llevadas a cabo por pequeñas moléculas llamadas neurotransmisores, estas son proteínas almacenadas y liberadas por vesículas sinápticas ubicadas en el interior de la presinapsis. La señal química para la liberación de estos neurotransmisores es el arribo de un potencial de acción a la terminal sináptica, lo cual ocasiona un ligero cambio de voltaje que es detectado por complejos de proteínas

incrustados en la membrana celular (en este caso específicamente por el canal de calcio activado por voltaje), de esta manera se liberan en el espacio sináptico y viajan hacia la postsinapsis.

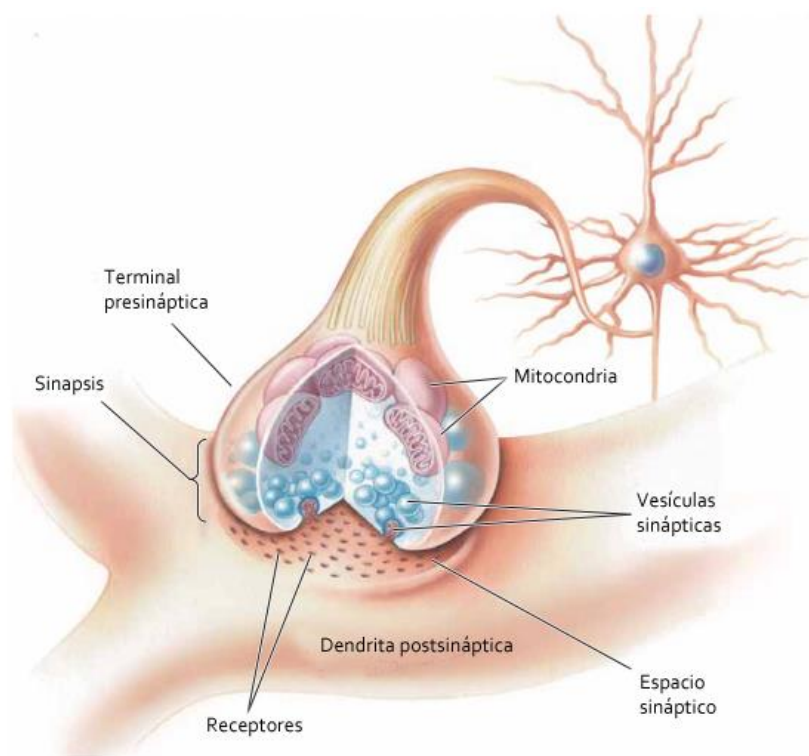


Figura 2.3. Estructura de la sinapsis. Modificado de [1]

Las respuestas a muchas de las preguntas sobre el funcionamiento del sistema nervioso requieren comprender cómo se comunican las neuronas y cuáles son los distintos patrones de comunicación entre ellas. Resulta de vital importancia el desarrollo de métodos de registro de actividad neuronal a gran escala. Diversos enfoques y propuestas han surgido con los años y conforme se desarrollan nuevas tecnologías también se desarrollan nuevas técnicas para este fin. En el presente trabajo los datos adquiridos se obtuvieron con ayuda de la técnica de imagenología de calcio.

## 2.1. Imagenología de calcio

La imagenología de calcio es una técnica que permite censar de manera indirecta la actividad eléctrica de las neuronas, hace uso de proteínas que adquieren propiedades fluorescentes cuando se unen químicamente al  $\text{Ca}^{2+}$ . Estas moléculas se hacen entrar o se expresan genéticamente dentro de los somas de las neuronas y resultan en un sensor de la actividad eléctrica de las neuronas debido que cuando estas disparan potenciales de acción dejan entrar  $\text{Ca}^{2+}$  al mismo tiempo, por lo tanto bajo la luz de estimulación adecuada para cada fluoróforo podemos ver emitir luz a estas proteínas exactamente en el mismo instante de cuando están disparando potenciales de acción [13]. En la Figura 2.4 para una neurona observada con esta técnica se hace la comparación de la señal eléctrica registrada mediante un electrodo y su señal de cambio de fluorescencia.

Esta técnica ha resultado ser muy útil para mapear interacciones neuronales a través de extensas poblaciones de neuronas con resolución de célula única, dependiendo del microscopio pueden llegarse a registrar de 100 a 500 neuronas al mismo tiempo en una porción de tejido nervioso [9, 14, 15]. Las interacciones observadas requieren de análisis previos de la actividad individual de las neuronas. El primer paso siempre consiste en obtener la señal correspondiente al cambio de fluorescencia por cada neurona, para lo cual se necesita la previa identificación individual de cada uno de los cuerpos celulares presentes en el experimento, posteriormente es necesario procesar estas señales para determinar la ocurrencia de los potenciales de acción. Finalmente, un conjunto de datos que muestren los potenciales de acción de cada neurona de una población observada en función del tiempo permite a los investigadores describir la actividad neuronal de la muestra, buscar patrones, generar hipótesis y conclusiones. El presente trabajo busca implementar una herramienta de software que agilice tales



procedimientos, de tal manera que los investigadores puedan obtener y procesar cada vez una mayor cantidad de datos, de la misma manera abre la puerta a nuevas posibilidades de experimentación [16].

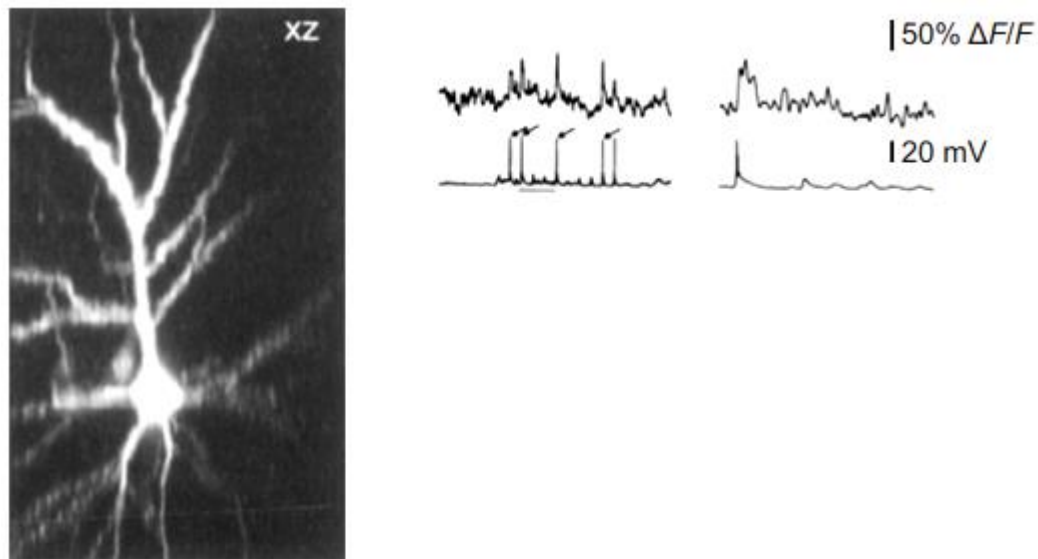


Figura 2.4. Para la neurona de la imagen a la izquierda se comparan los registros obtenidos (derecha) por imagenología de calcio (superior) y mediante electrodo (inferior) [17]

## 2.2. Experimento característico

En el Laboratorio de Neurociencias Cognitivas del Instituto de Fisiología Celular de la UNAM (IFC), se realizan experimentos sobre animales en el marco de las normas mexicanas para la producción, cuidado y uso de animales de laboratorio (NOM-062-ZOO-1999). En un experimento característico, una rata Wistar de 20 a 36 días postnatal es anestesiada con 85 mg/kg de ketamina y 15 mg/kg de xilacina, entonces rápidamente su cerebro es extraído y colocado en una solución salina de (en mM) 126 NaCl, 3.5 KCl, 26 NaHCO<sub>3</sub>, 1 MgCl<sub>2</sub>, 1.5 CaCl<sub>2</sub>, 1.2 NaH<sub>2</sub>PO<sub>4</sub>, 11 glucosa, 300 mOsm/L, pH=7.4, saturada con 95% O<sub>2</sub> y 5% CO<sub>2</sub> y a 4°C de temperatura. Las rebanas permanecen en esta solución a temperatura ambiente (21 ~ 25°C) por al menos 1 hora antes de las observaciones. Posteriormente las rebanadas son incubadas en oscuridad de 20 a 25

minutos a temperatura ambiente en presencia de 1-10  $\mu\text{M}$  fluo 4-AM (Molecular Probes  $\text{\textcircled{R}}$ , Life technologies, Ciudad de México) en 0.1% de dimetilsulfóxido (DMSO) y la misma solución salina.

Para la observación se emplea un microscopio equipado con objetivo 20X de inmersión en agua y 0.95 NA (XLUMPlanFI, Olympus, Center Valley, PA). Para observar los cambios espontáneos de la fluorescencia se estimula la preparación con pulsos de luz de 488 nm y duración de 15 a 50 ms con una fuente de iluminación Lambda LS (Sutter Instruments, Petaluma, CA) conectada al microscopio mediante fibra óptica. El área de observación consta de 750 X 750  $\mu\text{m}$ . Entonces se adquieren videos mediante una cámara digital con enfriamiento (CoolSnap K4, Photometrics, Tucson, AZ) a una tasa de 100 a 250 ms/cuadro y en intervalos de 5 a 10 minutos. Cada video consta de 3600 cuadros y resolución de 512 X 512 píxeles. En total los videos suman aproximadamente 30 minutos de grabación [18].

Para el procesamiento de los videos, los expertos etiquetan las neuronas con ayuda de ImageJ [19] las regiones de interés ROIs donde se considera que pueden haber neuronas y guardan las coordenadas  $(x,y)$  del centro, radio  $r$  y un identificador numérico de cada ROI en un archivo llamado RoiSet.zip con un formato propio de esta herramienta (Figura 2.5). Posteriormente, mediante la ecuación Ec. 2.1 para cada ROI se calcula su señal de cambio de fluorescencia (también llamada transitorio de calcio), donde  $S_r$  es el valor de la señal de la ROI en un cuadro,  $I_r$  es el promedio de la intensidad de los píxeles que conforman la ROI e  $I_B$  es el promedio de los píxeles contenidos en un área circular con radio  $R$  que rodea la ROI de radio  $r$  y  $R$  es  $k$  veces mayor que  $r$ . Finalmente las señales son procesadas para determinar la ocurrencia de potenciales de acción, aquellas regiones de interés que no presentan actividad son descartadas y los resultados son expresados en una matriz binaria  $N \times F$

donde  $N$  es el número de ROIs,  $F$  el número de cuadros, 0 estado de reposo y 1 actividad asociada a potenciales de acción (Figura 2.6).

$$S_r = \frac{I_r - I_B}{I_r}$$

Ec. 2.1

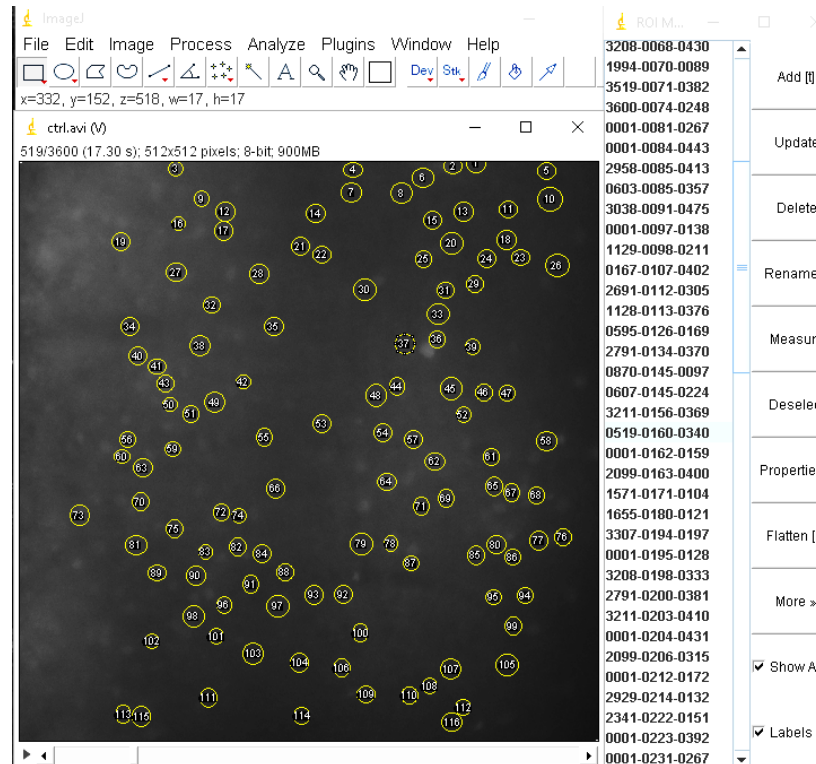


Figura 2.5. Etiquetado manual con ImageJ, de lado izquierdo se observa un instante de un video sobre el cual se muestran las etiquetas hechas por el experto y del lado derecho se muestra los datos contenidos en el RoiSet.zip

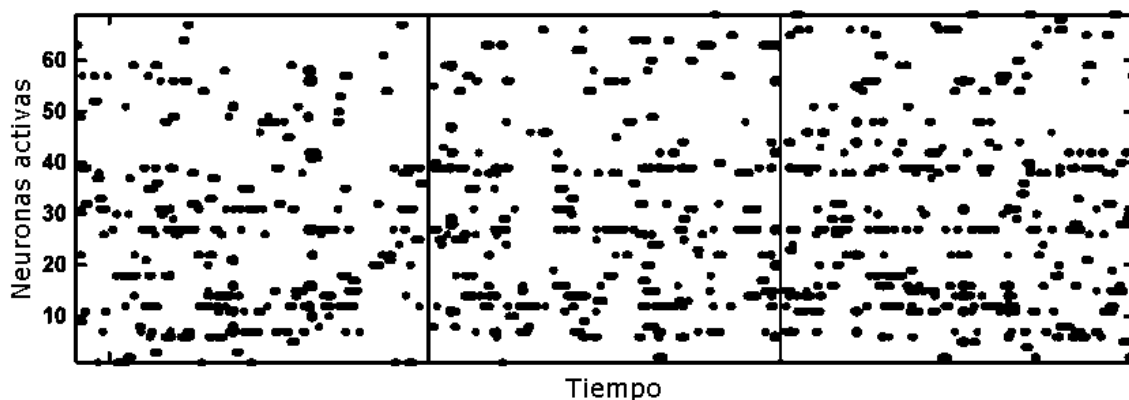


Figura 2.6. Ejemplo de matriz de actividad [9], cada fila corresponde a una neurona diferente, las columnas corresponden a instantes de tiempo, las líneas verticales indican la división entre cada video del experimento y los puntos negros actividad neuronal

### 2.3. Trabajos relacionados con el problema

Una de las tareas relacionadas a la extracción de datos en los experimentos más exhaustivas, de mayor demora y probabilidad de error humano es la identificación manual de neuronas en los videos obtenidos de la experimentación. En ese sentido son requeridas herramientas automatizadas que apoyen a los investigadores en tareas de detección y segmentación de neuronas en el material adquirido. Han sido ya muchos los esfuerzos por desarrollar sistemas que satisfagan esta necesidad, sin embargo, el estado del arte aún no es satisfactorio por lo que permanece como un problema abierto dada su complejidad [5]. Las herramientas de apoyo desarrolladas, como las empleadas en el IFC suponen la inversión de mucho tiempo, la experiencia del investigador y los resultados pueden incluso llegar a ser subjetivos.

S. Banglieto et al [5] han propuesto un método para segmentar neuronas de imágenes con resolución de 2048 X 2048 y 1024 X 1024 píxeles obtenidas de experimentos para observar células nerviosas en retina de ratón y en larvas de *Drosophila* mediante microscopia de fluorescencia. Las imágenes, como en los videos obtenidos en el IFC, contienen un par de cientos de neuronas. Su objetivo es segmentar de las neuronas tanto el soma como el árbol dendrítico con la finalidad de mapear las conexiones que hay entre ellas. Su método plantea para la segmentación del soma un algoritmo basado en detección de *blobs* con análisis multiescala seguido por uno de *active contour*, transformación *watershed* y umbralización. Al evaluar con etiquetas hechas manualmente, su método logró detectar entre el 75 y 89.19% de las células identificadas manualmente.

El reciente desarrollo de técnicas de aprendizaje profundo para procesamiento de imágenes también ha encontrado su aplicación en imagenología celular, como es mostrado en el trabajo de

J. Chen et al [6] donde presentan una novedosa arquitectura de redes convolucionales completamente conectadas (FCN) llamada *Complete Bipartite Networks* (CB-NET) que es probada en segmentación de neuronas de danio cebra en imágenes adquiridas con microscopía confocal bicolor. Debido a que este tipo de arquitecturas son de aprendizaje supervisado requirieron de muestras etiquetadas a mano para su entrenamiento y validación, una ventaja importante, es que a diferencia de las arquitecturas tradicionales, esta no requiere de grandes volúmenes de muestras para su entrenamiento, en cambio han usado siete *datasets* cada uno con 55 imágenes y aumentado de datos. Además este método involucra explícitamente el uso de análisis multiescala y requiere de post-procesamiento en los resultados arrojados por la arquitectura, donde se aplica umbralización y morfología matemática.

Un tercer trabajo, es el presentado por M. Dorostkat et al [7] en el que presentan SARFIA (*Semi-Automated Routines for Functional Image Analysis*), un sistema para el análisis de microcircuitos en imágenes adquiridas de observaciones de neuronas de danio cebra con microscopia de fluorescencia. En este trabajo plantean el uso de filtros de suavizado, Laplaciano y umbralización para la detección de regiones de interés, además su modelo de datos de salida contempla la localización y tamaño de las regiones identificadas tal como ocurre en los experimentos del IFC para el cálculo de los potenciales de acción.

Los tres trabajos mencionados anteriormente integran elementos interesantes y útiles, el análisis multiescala es una constante en estos métodos así como el uso de otras técnicas de procesamiento de imágenes digitales como filtros de suavizado, el operador Laplaciano y umbralización, incluso en el primer trabajo emplean técnicas aún más complejas como *active contour* y *watershed*. Aunque los enfoques de aprendizaje profundo plantean poderosas opciones, aún requieren alguna etapa de procesamiento de imágenes. Además, la necesidad de muestras

etiquetadas a mano para su entrenamiento y validación representan una importante desventaja debido a que en esta área de aplicación es difícil disponer de tantas muestras etiquetadas aunque esto podría ser un tanto compensado con el uso de técnicas de aumento de datos.

### 3. Procesamiento de imágenes digitales

Si bien la adquisición de imágenes digitales ha impulsado campos de la investigación científica al brindar medios para registrar observaciones, es necesario contar con métodos para extraer información significativa de ellas. Este trabajo está enfocado particularmente en imágenes en escala de grises, las cuales son un conjunto de datos llamados píxeles y ordenados en un arreglo de  $N$  filas y  $M$  columnas, cada píxel posee una posición única dentro del arreglo en las coordenadas  $(x,y)$  y un valor de intensidad  $I$  en una escala de  $L-1$  niveles de gris (donde 0 es blanco y  $L-1$  es negro) tal que  $x$ ,  $y$  e  $I$  son valores discretos dentro de un rango definido, así una imagen está definida como la función matemática:

$$f(x, y) = I, \quad x \in [0, N - 1], \quad y \in [0, M - 1], \quad I \in [0, L - 1]$$

El procesamiento de imágenes digitales consiste en llevar a cabo cálculos a partir de los valores numéricos de los píxeles de una imagen, dependiendo del resultado es posible definir tres tipos de procesamiento: de bajo, medio y alto nivel. El procesamiento de bajo nivel involucra operaciones simples de reducción de ruido, mejoramiento de contraste o perfilado y se caracteriza por tener por entrada y salida una imagen. El procesamiento de medio nivel comprende tareas de segmentación (fragmentar imágenes en regiones de interés) y descripción de regiones en formatos que permitan su procesamiento y clasificación (reconocimiento), en este nivel se tiene por salida el conjunto de características extraídas de una imagen de entrada. Finalmente, el procesamiento de alto nivel consiste en darle sentido a las características extraídas de una imagen de entrada, esto requiere de tareas cognitivas y conforma un completo análisis de imágenes digitales [20].

### 3.1. Píxel

Aunque el procesamiento de una imagen digital consiste en operar los píxeles contenidos en ella, en muchas ocasiones no es suficiente con conocer los atributos propios de cada píxel (coordenadas e intensidad) y es necesario considerar el contexto o vecindario en el que se encuentra cada uno.

Sea  $P$  un píxel con coordenadas  $(x,y)$  contenido en una imagen,  $N_8(P)$  los ocho píxeles que rodean a  $P$ ,  $N_4(P)$  los cuatro píxeles que rodean a  $P$  en los ejes horizontal y vertical y  $N_D(P)$  los cuatro píxeles que rodean a  $P$  en los ejes diagonales; el vecindario de  $P$  puede ser descrito en términos de los conjuntos  $N_8(P)$ ,  $N_4(P)$  y  $N_D(P)$  (Figura 3.1). Un dato importante que se pueden obtener del vecindario de un píxel es la conectividad (Figura 3.2), dos píxeles contiguos que cumplen con criterios específicos de similitud, como el nivel de gris, se dicen que están conectados [21].

|             |           |             |
|-------------|-----------|-------------|
| $(x-1,y-1)$ | $(x,y-1)$ | $(x+1,y-1)$ |
| $(x-1,y)$   | $P(x,y)$  | $(x+1,y)$   |
| $(x-1,y+1)$ | $(x,y+1)$ | $(x+1,y+1)$ |

Figura 3.1. Vecindario de  $P$

$$N_8(P) = \left\{ \begin{array}{l} (x-1, y-1), (x, y-1), (x+1, y-1), (x-1, y), (x+1, y), \\ (x-1, y+1), (x, y+1), (x+1, y+1) \end{array} \right\}$$



$$N_4(P) = \{(x, y - 1), (x - 1, y), (x + 1, y), (x, y + 1)\}$$

$$N_D(P) = \{(x - 1, y - 1), (x + 1, y - 1), (x - 1, y + 1), (x + 1, y + 1)\}$$

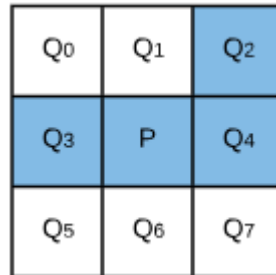
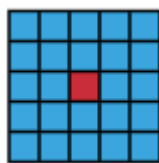
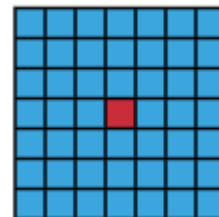


Figura 3.2.  $P$  tiene conectividad con  $Q_2$ ,  $Q_3$  y  $Q_4$  considerando el color como criterio de similitud

A veces es necesario conocer el contexto en el que se encuentra  $P$  considerando un área de mayor tamaño que los ocho píxeles que le rodean, en estos casos son consideradas regiones llamadas ventanas, cada ventana es una matriz generalmente cuadrada donde  $P$  es su píxel central (Figura 3.3), además es un subconjunto de la imagen original y posee  $n$  filas y  $m$  columnas, donde  $n \ll N$  y  $m \ll M$ .



(a)



(b)

Figura 3.3. En (a) se muestra una ventana de 5 X 5 píxeles mientras que en (b) se muestra una de 7 X 7. El color rojo indica el píxel central  $P$  y en azul los píxeles considerados en el contexto de  $P$

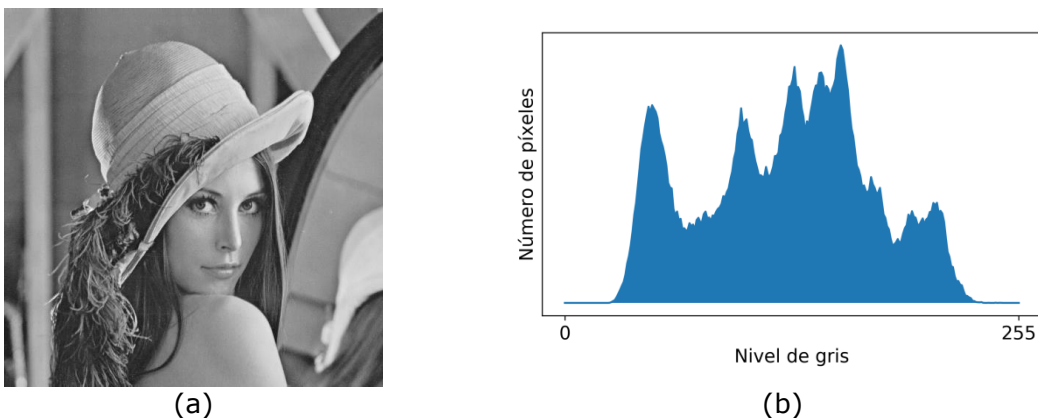
### 3.2. Histograma

El histograma es un concepto importante en estadística que aplicado en imágenes digitales permite conocer información descriptiva de la misma, específicamente la frecuencia con que aparecen cada uno de los valores de gris en la imagen (Figura 3.4) [22]. Así, el histograma de una imagen está definido como:

$$H(r_k) = n_k, \quad k \in [0, L - 1] \quad \text{Ec. 3.1}$$

En la ecuación anterior  $r_k$  es el  $k$ -ésimo nivel de gris y  $n_k$  es el número de píxeles que en la imagen tienen el valor  $r_k$ . Al dividir la ecuación Ec. 3.1 entre el número total de píxeles que hay en la imagen se obtiene la probabilidad  $P(r_k)$  de ocurrencia de  $r_k$ .  $P(r_k)$  también recibe el nombre de histograma normalizado [24].

$$P(r_k) = \frac{n_k}{n}, \quad P(r_k) \in [0, 1] \quad \text{Ec. 3.2}$$



Obtener información descriptiva de una imagen a través de su histograma resulta conveniente para ciertos propósitos de mejoramiento o segmentación. Una operación común es la umbralización que consiste en binarizar en intervalos de menor tamaño que el de la escala de grises

y existen diferentes enfoques en su uso, uno de ellos consiste en asignar el máximo valor (blanco) a los píxeles dentro del intervalo y cero (negro) a los que está fueran, de modo que se obtiene una imagen binaria por lo que este enfoque es llamado binarización (Figura 3.5), con esto es posible eliminar o segmentar regiones de la imagen por su nivel de gris [22].

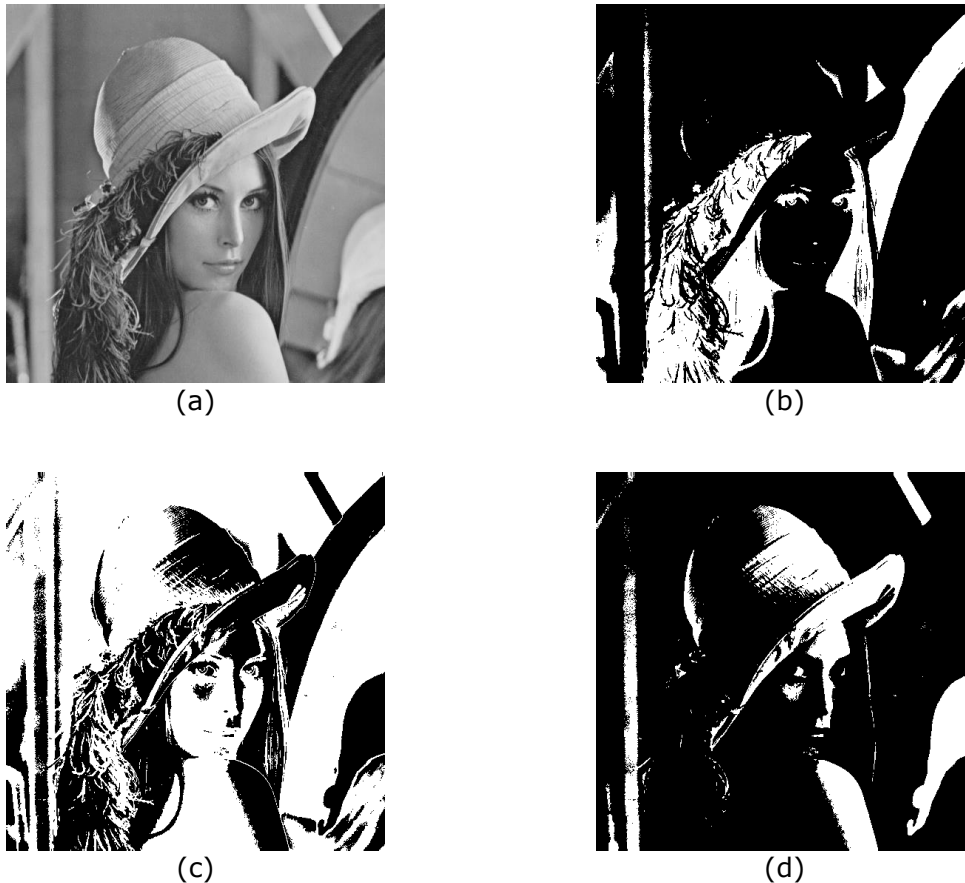


Figura 3.5. En (b), (c) y (d) se muestra el resultado de binarizar (a) en los intervalos  $[0,84]$ ,  $[85,170]$  y  $[171,255]$  respectivamente

La distribución de los valores de gris en un histograma indica el contraste de una imagen, cuando los valores de gris se concentran en una región estrecha de la escala de grises entonces se trata de una imagen con bajo contraste, si esta región comprende los valores más altos de la escala entonces es una imagen brillante y es oscura cuando la región comprende los valores más bajos de la escala. Asimismo, una distribución uniforme de los valores de gris se traduce en una imagen con alto

contraste y por lo tanto con mejor calidad en el sentido de que hay una adecuada proporción en los niveles de gris [23] (Figura 3.6). La mejora de imágenes a través de modificar su histograma es una práctica común en el procesamiento de imágenes digitales, también recibe el nombre de ecuilización del histograma y generalmente forma parte de etapas de preprocesamiento [20].

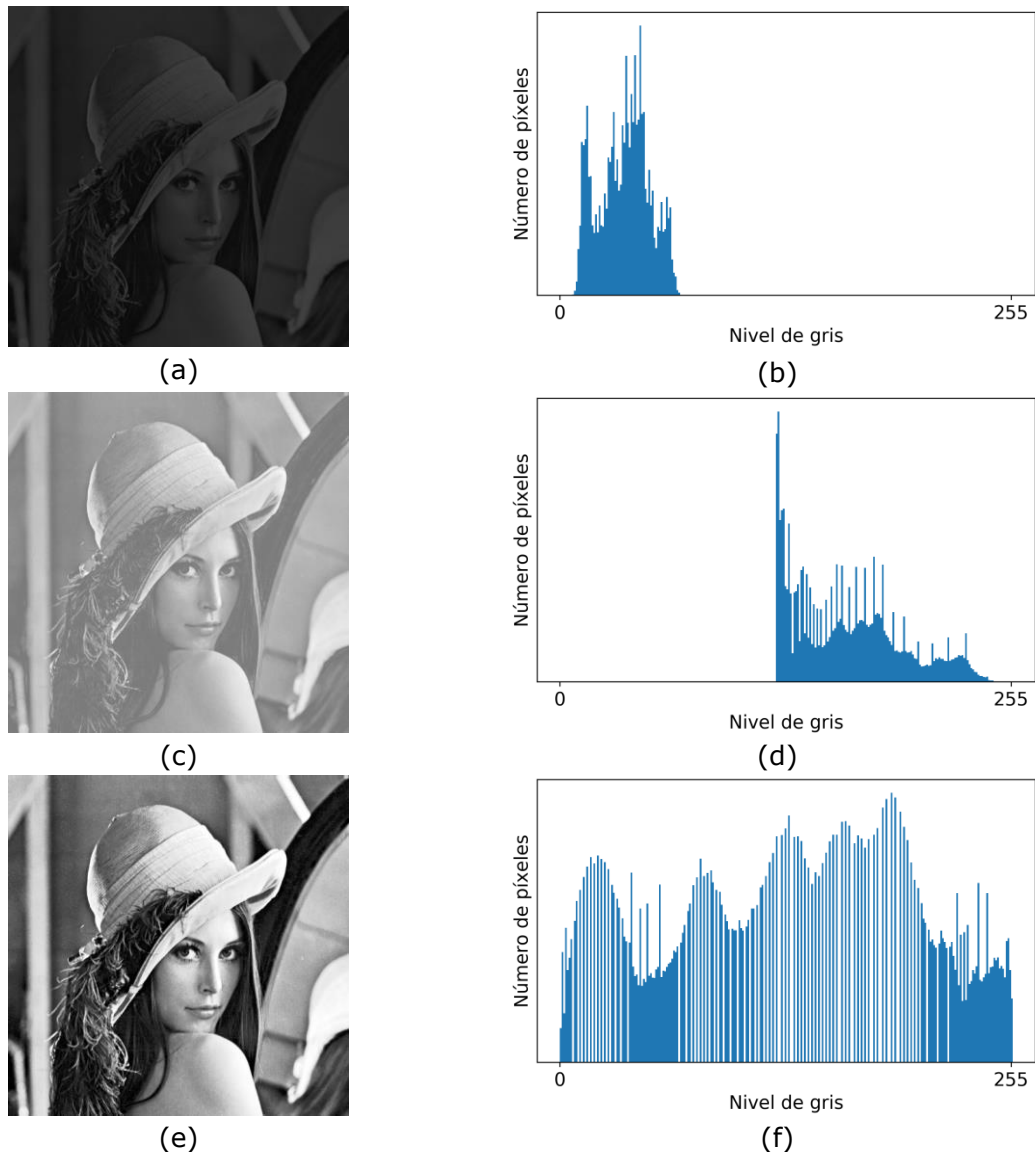


Figura 3.6. (a) y (c) son dos imágenes de bajo contraste, oscura en el primer caso y brillante en el segundo. Mientras que (e) es una imagen con alto contraste. Las gráficas (b), (d) y (f) son los histogramas de las imágenes (a), (c) y (e) respectivamente

Sea el histograma normalizado  $P(r_k)$  y la función  $s=T(r)$ , tal que  $T(r)$  produce un valor de gris  $s$  por cada valor de gris  $r$  en la imagen original y satisface las condiciones:

- $T(r)$  produce un solo valor  $s$  para cada  $r$  y  $r \in [0, 1]$
- $T(r) \in [0, 1]$

La primera condición garantiza la existencia de la transformada inversa de  $T(r)$ , mientras que la segunda garantiza que los valores producidos por  $T(r)$  estarán en el mismo rango que  $r$ . Entonces, sea  $P_r(r)$  la función de densidad de probabilidad de  $r$  y  $P_s(s)$  la de  $s$ , se tiene que:

$$P_s(s) = P_r(r) \frac{dr}{ds} \quad \text{Ec. 3.3}$$

La ecuación Ec. 3.3 es posible debido a que  $T(r)$  posee transformada inversa; como  $P_s(s)$  está determinada por  $P_r(r)$  y la transformada  $T(r)$  existe, se tiene que:

$$T(r) = \int_0^1 P_r(r) dr \quad \text{Ec. 3.4}$$

Para dominios discretos se tiene que:

$$P_r(r_k) = \frac{n_k}{n}, k = 0, 1, 2, \dots, L - 1 \quad \text{Ec. 3.5}$$

$$S_k = T(r_k) \quad \text{Ec. 3.6}$$

$$T(r_k) = \sum_{j=0}^k P_r(r_j) \quad \text{Ec. 3.7}$$

La ecualización de un histograma consiste en realizar la transformación  $T(r_k)$  y tiene la ventaja de ser fácil de implementar y no

requerir elevadas prestaciones de hardware debido a que computacionalmente no es costosa y opera con información extraída directamente de la imagen sin necesidad de parámetros adicionales [20].

La ecualización del histograma es una operación global, es decir, opera sobre todos los píxeles de la imagen, aunque esto ofrece una mejora, pequeñas regiones pueden verse afectadas negativamente por regiones de mayor tamaño aun estando en extremos opuestos de la imagen. Un enfoque para resolver este problema consiste en ecualizar de manera particular cada píxel considerando su contexto usando ventanas, diferentes tamaños de ventana ofrecen resultados diferentes [21] (Figura 3.7). Del mismo modo que ecualizar globalmente no siempre es la mejor opción, tampoco lo es usar siempre tamaños muy reducidos de ventana.

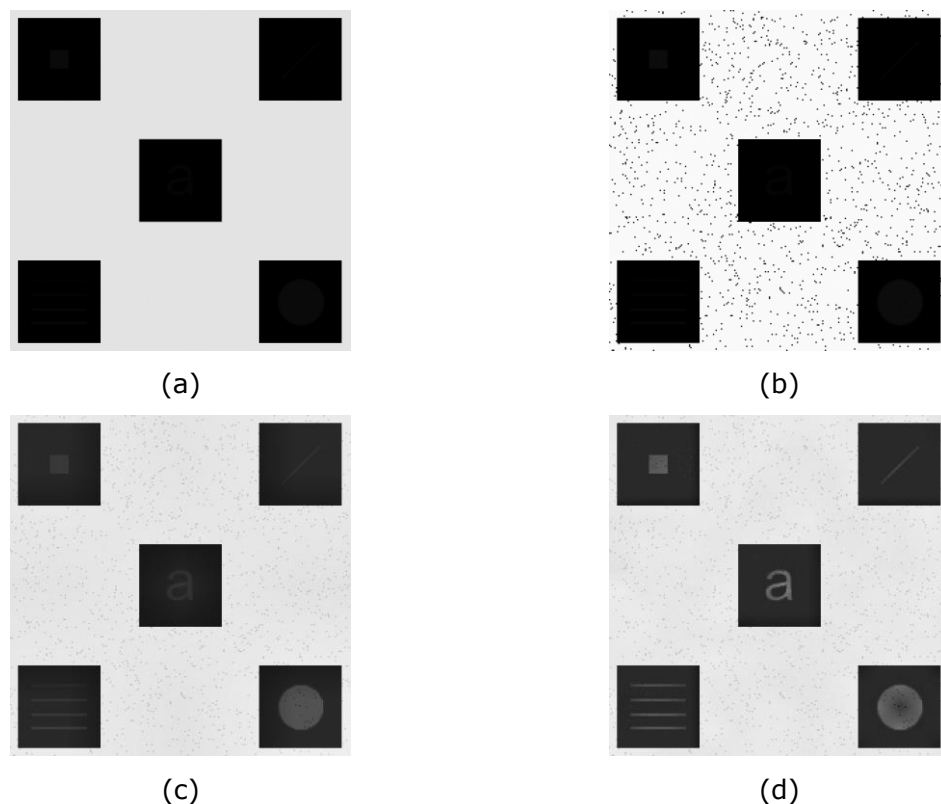


Figura 3.7. En (b) se tiene el resultado de ecualizar globalmente (a), mientras que en (c) y (d) se muestra el resultado de aplicar ecualización local con ventanas de 3 X 3 y 5 X 5 respectivamente

### 3.3. Filtros

Una imagen puede ser pensada como una señal finita en el espacio y en particular como una señal discreta tratándose de una imagen digital. Esto permite filtrar una imagen haciendo la convolución de la imagen con un *kernel* [20]. Un *kernel* es una matriz de coeficientes, al igual que lo es una imagen pero de tamaño mucho menor, por lo tanto un *kernel* también está definido como una función matemática. Sean  $f(x)$  y  $g(x)$  dos señales continuas, la convolución de ambas se denota como  $f(x)*g(x)$  y está dada por la ecuación Ec. 3.8.

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(t)g(x - t)dt \quad \text{Ec. 3.8}$$

Si  $f(x)$  y  $g(x)$  son señales discretas entonces la expresión es:

$$f(x) * g(x) = \sum_{t=-\infty}^{\infty} f(t)g(x - t) \quad \text{Ec. 3.9}$$

Aplicado a funciones de dos variables como el caso de una imagen o un *kernel* se tiene la expresión:

$$f(x, y) * g(x, y) = \sum_{\alpha=-\infty}^{\infty} \sum_{\beta=-\infty}^{\infty} f(\alpha, \beta)g(x - \alpha, y - \beta) \quad \text{Ec. 3.10}$$

Al aplicar la ecuación Ec. 3.10 en imágenes digitales hay que tomar en cuenta que las imágenes son funciones finitas por lo que tienen un rango definido, de modo que los rangos de  $\alpha$  y  $\beta$  en las sumatorias deben comprender las dimensiones de la imagen. Así, si se tiene una imagen digital  $g(x,y)$  de  $N \times M$  píxeles y un *kernel*  $f(x,y)$  de  $A \times B$  coeficientes,  $f(x,y)*g(x,y)$  está dada por la expresión:

$$f(x, y) * g(x, y) = \sum_{\alpha=-s}^s \sum_{\beta=-t}^t f(\alpha, \beta)g(x - \alpha, y - \beta) \quad \text{Ec. 3.11}$$

Donde  $s = \frac{A-1}{2}$  y  $t = \frac{B-1}{2}$

La ecuación Ec. 3.11 implica un proceso iterativo y lineal, por lo que este proceso es también llamado filtrado lineal [21]. El resultado es calculado para cada píxel con coordenadas  $(x,y)$  en la imagen y este proceso puede ser ilustrado de manera mecánica como si el *kernel* fuera desplazado por toda la imagen y para cada posición se multiplicaran punto a punto los coeficientes del *kernel* con el valor de los píxeles sobre los que está colocado, se sumaran los productos y se asignara el resultado al píxel correspondiente al centro del filtro (Figura 3.8).

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 4 | 6 | 5 | 9 | 5 | 3 |
| 3 | 8 | 5 | 3 | 8 | 0 |
| 0 | 7 | 3 | 7 | 1 | 2 |
| 9 | 7 | 1 | 3 | 7 | 3 |
| 3 | 8 | 6 | 0 | 0 | 4 |
| 5 | 1 | 6 | 7 | 7 | 8 |

(a)

|     |     |     |
|-----|-----|-----|
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |
| 1/9 | 1/9 | 1/9 |

(b)

$$(3*1+8*1+0*1+7*1+1*1+2*1+3*1+7*1+3*1)/9=3.8$$

|   |   |   |   |   |   |
|---|---|---|---|---|---|
| 4 | 6 | 5 | 9 | 5 | 3 |
| 3 | 8 | 5 | 3 | 8 | 0 |
| 0 | 7 | 3 | 7 | 1 | 2 |
| 9 | 7 | 1 | 3 | 7 | 3 |
| 3 | 8 | 6 | 0 | 0 | 4 |
| 5 | 1 | 6 | 7 | 7 | 8 |

(c)

|   |   |   |   |     |   |
|---|---|---|---|-----|---|
| 4 | 6 | 5 | 9 | 5   | 3 |
| 3 | 8 | 5 | 3 | 8   | 0 |
| 0 | 7 | 3 | 7 | 3.8 | 2 |
| 9 | 7 | 1 | 3 | 7   | 3 |
| 3 | 8 | 6 | 0 | 0   | 4 |
| 5 | 1 | 6 | 7 | 7   | 8 |

(d)

Figura 3.8. En (a) se muestra una imagen de 6 X 6 píxeles, en (b) un filtro de 3 X 3 coeficientes, en (c) es calculado el resultado de operar el filtro sobre la imagen en una posición arbitraria y en (d) el resultado es asignado al píxel correspondiente al centro del filtro



La convolución es una operación que posee las propiedades de la conmutatividad, distributividad y asociatividad. La propiedad conmutativa de la convolución implica que no importa el sentido en que se realice la convolución, el resultado será siempre el mismo, mientras que la distributiva implica que la convolución de una función con la suma de dos funciones es equivalente a convolucionar las funciones por separado y sumar los resultados. Es decir, se obtiene el mismo resultado al sumar dos imágenes o *kernels* y después convolucionar con un tercer elemento o hacer la convolución primero por separado y sumar los resultados. Finalmente la propiedad asociativa implica que al convolucionar tres funciones es indiferente el orden en el que se haga [21].

Otro enfoque de filtrado de imágenes consiste en operaciones no lineales por lo que recibe el nombre de filtrado no lineal, estos no hacen uso del *kernel* ni de la operación convolución, en cambio aplican operaciones sobre los píxeles contenidos en una ventana colocada sobre la imagen y el resultado es asignado al píxel correspondiente al centro de la misma, del mismo modo que con los filtros lineales, la ventana es desplazada por toda la imagen de modo que la operación es realizada para cada píxel en ella [20, 21]. De hecho la ecualización local es un filtro no lineal debido a que cumple con las características antes mencionadas, otro ejemplo de operación usada en este tipo de filtros es la mediana.

Entre la amplia gama de filtros, se encuentran los filtros de suavizado que son ampliamente utilizados por su utilidad para eliminar el ruido de las imágenes (Figura 3.9), este tipo de filtros también son llamados promediadores debido a que la configuración de los coeficientes del *kernel* corresponde a obtener el promedio del valor de los píxeles sobre los que está siendo aplicado el filtro [20]. Al asignar a un píxel el valor promedio de su contexto se reducen variaciones bruscas de intensidad; el ruido, al ser un fenómeno que se caracteriza por variaciones aleatorias abruptas en los valores de gris en una imagen, es

posible reducirlo aplicado filtros de suavizado. Sin embargo, los bordes de objetos también se caracterizan por cambios bruscos de nivel de gris por lo que los bordes también se ven suavizados, lo que implica la pérdida de detalles. Además, a mayor tamaño de *kernel* mayor es el efecto del filtro.

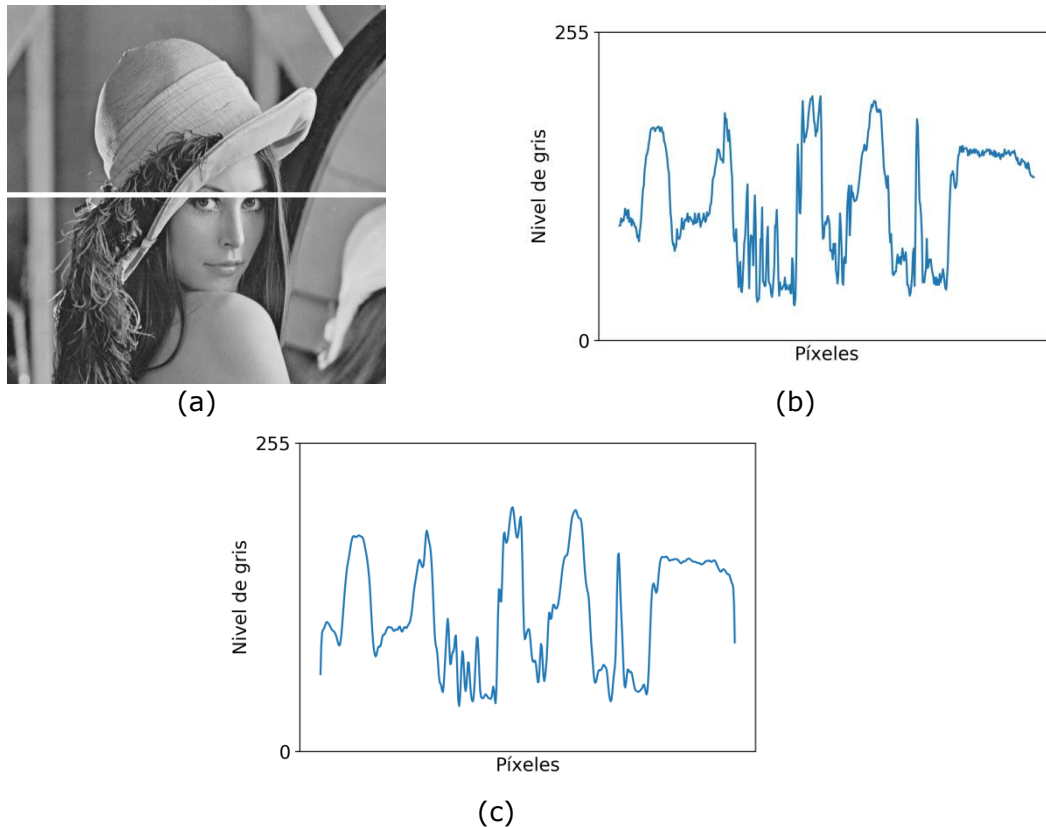


Figura 3.9. En (b) se muestra el comportamiento de los valores de gris correspondientes a la franja blanca sobre la imagen (a) y en (c) es mostrada la misma región después de ser aplicado un filtro de suavizado. Como resultado se obtiene una señal menos ruidosa, de hecho estos filtros son también conocidos como pasabajas

Hay muchas configuraciones de coeficientes para filtros de suavizado, la más simple corresponde al filtro de media que consiste en un *kernel* formado de valores iguales y mayores a cero, este filtro tiene la desventaja de producir efectos indeseables en la imagen filtrada llamados efectos de bloque debido a que pareciera que hay bloques con la forma del *kernel* dentro de la imagen. Para compensar esta desventaja puede ser empleado un filtro circular o uno ponderado donde el píxel central es el que tiene más peso. Aunque estas dos opciones ayudan a reducir el

efecto de bloque, no son una solución completa ya que el problema en realidad persiste solo que en diferente intensidad. Al ponderar los coeficientes de un *kernel* con una distribución de probabilidad Gaussiana se evita tal efecto y con ello se obtienen resultados uniformes. Este tipo de filtros basados en la distribución Gaussiana reciben el nombre de filtros Gaussianos (Figura 3.10). Los coeficientes de filtros promediadores deben ser divididos entre la suma de todos los coeficientes, con esto se garantiza que los nuevos valores de gris estarán dentro del rango de la escala de grises original [24].



Figura 3.10. En (b) se muestra el resultado de aplicar un filtro de media de 13 X 13 en (a), en (c) y (d) después de aplicar a (a) un filtro Gaussiano de 13 X 13 y 19 X 19 respectivamente. Es posible apreciar el efecto de bloque en (b) como sutiles líneas horizontales y verticales que no están presentes en (c) ni en (d)

Es posible aproximar una distribución Gaussiana a través de la función binomial (Figura 3.11) dada por la ecuación Ec. 3.12, donde  $x=0, 1, 2, \dots, N$  y  $N$  es el orden del filtro, de modo que los coeficientes del

kernel están dados por la expresión  $[f_N(x)]^T * [f_N(x)]$ . En la Tabla 3.1 se muestra la construcción de un filtro Gaussiano de orden  $N=2$ .

$$f_N(x) = \frac{N!}{x!(N-x)!} \quad \text{Ec. 3.12}$$

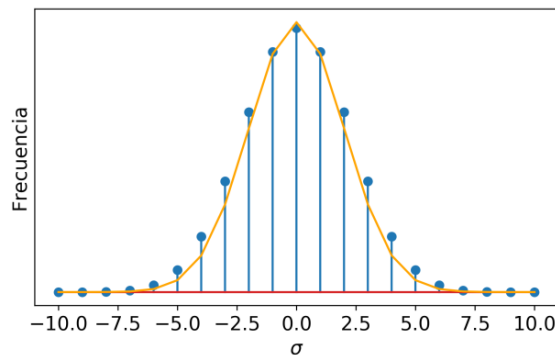


Figura 3.11. Aproximación de una distribución Gaussiana (amarillo) con una función binomial (azul)

Tabla 3.1. Construcción de un filtro Gaussiano de orden 2

| $N$ | $f_N(x) = \frac{N!}{x!(N-x)!}$               | $[f_2(x)]$            | $[f_2(x)]^T * [f_2(x)]$   |
|-----|--|-----------------------|---|
| 2   | $f_2(0) = 1$<br>$f_2(1) = 2$<br>$f_2(2) = 1$ | $[1 \quad 2 \quad 1]$ | $\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$ |

Entre los filtros no lineales también existen filtros de suavizado, el más popular es el de mediana, donde para todos píxeles contenidos en la ventana del filtro se calcula la mediana y se asigna el resultado al píxel central [23]. Aunque lo más común es emplear la mediana también puede ser usado el valor máximo o mínimo. Este filtro es particularmente útil con el tipo de ruido de sal y pimienta que se caracteriza por la presencia

aleatoria de puntos blancos y negros por toda la imagen [24]. En la Figura 3.12 Se muestra un ejemplo de imagen con ruido de sal y pimienta a la que se le aplica este filtro.



(a)



(b)

Figura 3.12. En (b) se muestra el resultado después de aplicar un filtro de mediana a la imagen (a), la cual presenta ruido de sal y pimienta

### 3.4. Perfilado

El efecto contrario de los filtros de suavizado es acentuar las variaciones bruscas en los niveles de gris, esto con el propósito de hacer más notorios los bordes de los objetos contenidos en una imagen, este proceso es conocido como perfilado y existen configuraciones de *kernel* que desempeñan este efecto. Normalmente el perfilado se realiza después de haber eliminado el ruido de una imagen en etapas previas de preprocesamiento, de lo contrario el ruido se vería amplificado empeorando la calidad de la imagen [20].

Los filtros de suavizado trabajan promediando los píxeles de una región, siendo el promedio análogo a la integración, entonces los filtros de perfilado operan mediante derivación espacial. Debido a que la respuesta de la derivada es proporcional al grado de variación de los valores de gris en la región donde es aplicado el filtro, el resultado es el realce de los

bordes y otro tipo de variaciones como el ruido, mientras que las regiones con poca variación son mitigadas.

La derivada de una función discreta está definida en términos de diferencias, así la primera derivada de una función discreta de una variable está dada por la expresión:

$$\frac{df}{dx} = f(x + 1) - f(x) \quad \text{Ec. 3.13}$$

Para funciones de dos variables:

$$\frac{\partial f}{\partial x} = f(x + 1, y) - f(x, y) \quad \text{Ec. 3.14}$$

$$\frac{\partial f}{\partial y} = f(x, y + 1) - f(x, y) \quad \text{Ec. 3.15}$$

Primera derivada compuesta:

$$\frac{\partial f}{\partial x \partial y} = \frac{\partial f}{\partial x} + \frac{\partial f}{\partial y} \quad \text{Ec. 3.16.1}$$

$$\frac{\partial f}{\partial x \partial y} = f(x + 1, y) - f(x, y) + f(x, y + 1) - f(x, y) \quad \text{Ec. 3.16.2}$$

$$\frac{\partial f}{\partial x \partial y} = f(x + 1, y) + f(x, y + 1) - 2f(x, y) \quad \text{Ec. 3.16.3}$$

La expresión de la ecuación Ec. 3.16.1 también recibe el nombre de gradiente.

La primera derivada tiene por características que:

- Es cero en regiones donde no hay variaciones de intensidad
- Es diferente de cero en regiones donde hay variaciones de intensidad y al final de una rampa o escalón
- Es diferente de cero a lo largo de una rampa

La segunda derivada de una función discreta de dos variables está dada por:

$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial f}{\partial x} f(x, y) - \frac{\partial f}{\partial x} f(x - 1, y) \quad \text{Ec. 3.17.1}$$

$$\frac{\partial^2 f}{\partial x^2} = [f(x + 1, y) - f(x, y)] - [f(x, y) - f(x - 1, y)] \quad \text{Ec. 3.17.2}$$

$$\frac{\partial^2 f}{\partial x^2} = f(x - 1, y) + f(x + 1, y) - 2f(x, y) \quad \text{Ec. 3.17.3}$$

$$\frac{\partial^2 f}{\partial y^2} = \frac{\partial f}{\partial y} f(x, y) - \frac{\partial f}{\partial y} f(x, y - 1) \quad \text{Ec. 3.18.1}$$

$$\frac{\partial^2 f}{\partial y^2} = [f(x, y + 1) - f(x, y)] - [f(x, y) - f(x, y - 1)] \quad \text{Ec. 3.18.2}$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y + 1) + f(x, y - 1) - 2f(x, y) \quad \text{Ec. 3.18.3}$$

La segunda derivada compuesta:

$$\frac{\partial^2 f}{\partial x \partial y^2} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad \text{Ec. 3.19.1}$$

$$\frac{\partial^2 f}{\partial x \partial y^2} = f(x - 1, y) + f(x + 1, y) - 2f(x, y) + f(x, y + 1) + f(x, y - 1) - 2f(x, y) \quad \text{Ec. 3.19.2}$$

$$\frac{\partial^2 f}{\partial x \partial y^2} = f(x - 1, y) + f(x + 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y)$$

Ec. 3.19.3

La segunda derivada tiene por características que:

- Es cero en regiones donde no hay variaciones de intensidad
- Es diferente de cero en regiones donde hay sobresaltos de intensidad y al final de una rampa o escalón
- Es cero a lo largo de una rampa
- Responde con mayor intensidad que la primera derivada en presencia de líneas delgadas, puntos aislados y pequeños detalles en general.

Debido a que la expresión de la ecuación Ec. 3.19.1 también es conocida con el nombre de Laplaciano, este filtro recibe el nombre de filtro Laplaciano. Además es un filtro isotrópico 90°, lo que significa que en los ejes horizontal y vertical es invariante a la dirección de las variaciones de intensidad en la imagen sobre la que es aplicado. En la Figura 3.13 se muestra la construcción del filtro Laplaciano a partir de la ecuación 3.19.3 y el resultado de aplicarlo sobre una imagen.

La segunda derivada tiene la particularidad de generar una doble respuesta como puede verse en la Figura 3.14 (c), el cruce por cero durante la respuesta indica con precisión la ubicación del borde mientras que el orden de las espigas indica el sentido del cambio. Conocer los bordes de los objetos contenidos en una imagen supone una ventaja en tareas de segmentación, sin embargo, su uso tiene como desventaja la amplificación del ruido. Por ello es necesario suprimir el ruido antes de aplicar un filtro Laplaciano [20].





(a)



(b)

|            |            |            |
|------------|------------|------------|
| 0          | $f(x,y-1)$ | 0          |
| $f(x-1,y)$ | $-4f(x,y)$ | $f(x+1,y)$ |
| 0          | $f(x,y+1)$ | 0          |

=

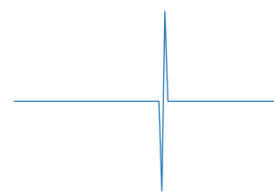
|   |    |   |
|---|----|---|
| 0 | 1  | 0 |
| 1 | -4 | 1 |
| 0 | 1  | 0 |

(c)

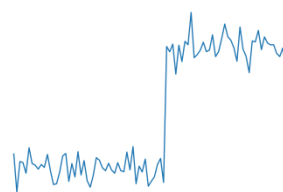
Figura 3.13. En (b) se muestra el resultado de aplica filtro Laplaciano (c), basado en la ecuación Ec. 3.19.3, en la imagen (a)



(a)



(b)



(c)



(d)

Figura 3.14. (a) Es una señal escalón, (c) una señal escalón con ruido y (b) y (d) son el resultado de aplicar un filtro Laplaciano

El proceso de suavizar y aplicar un filtro Laplaciano es un proceso de dos pasos y no es posible sumar un filtro de suavizado con uno Laplaciano debido a que la entrada del segundo filtro es la salida del primero. En este caso conviene aprovechar la propiedad asociativa de la convolución y hacer la convolución de dos filtros, uno Gaussiano para suavizado y uno Laplaciano para detección de bordes. El resultado de la operación anterior matemáticamente se interpreta como la segunda derivada de la función Gaussiana y recibe el nombre de Laplaciano de Gaussiana o LoG [22].

Sea la función Gaussiana, su primera y segunda derivada:

$$h(r) = -e^{-\frac{r^2}{\sigma^2}} \quad \text{Ec. 3.20.1}$$

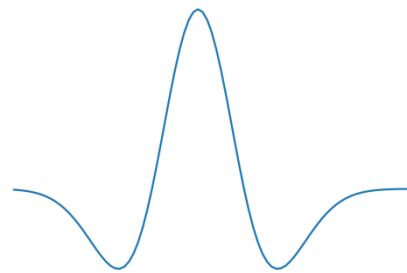
$$\frac{d}{dr}h(r) = \frac{r}{\sigma^2}e^{-\frac{r^2}{\sigma^2}} \quad \text{Ec. 3.20.2}$$

$$\frac{d^2}{dr^2}h(r) = -\frac{r^2-\sigma^2}{\sigma^4}e^{-\frac{r^2}{\sigma^2}} \quad \text{Ec. 3.20.3}$$

La ecuación Ec. 3.20.3 corresponde a la función LoG a partir de la cual es posible calcular los coeficientes del filtro que lleva el mismo nombre. En la Figura 3.15 se muestra el resultado de aplicar un filtro construido con esta función sobre una imagen de entrada.



(a)



(b)



(c)

|    |    |    |    |    |
|----|----|----|----|----|
| 0  | 0  | -1 | 0  | 0  |
| 0  | -1 | -2 | -1 | 0  |
| -1 | -2 | 16 | -2 | -1 |
| 0  | -1 | -2 | -1 | 0  |
| 0  | 0  | -1 | 0  | 0  |

(d)

Figura 3.15. El filtro (d) se obtiene a partir de la respuesta de una función LoG como la mostrada en (b). En (c) se muestra el resultado de aplicar el filtro (d) en (a)

### 3.5. Detección de BLOBS

El término *BLOB* (*Binary Large Object*) hace referencia a un grupo de píxeles conectados en una imagen binaria y de forma casi circular, un corte perpendicular a un *blob* mostraría una función escalón como la mostrada en la Figura 3.14. (a). Este tipo de estructuras son de particular interés debido a que es posible detectar objetos a través de puntos singulares como *blobs*. Debido a la forma del escalón unitario, un *blob* posee bordes que pueden ser detectados aplicando un operador Laplaciano o un LoG.

En la ecuación Ec. 3.20.3 además de la variable independiente  $r$  también está presente la desviación estándar  $\sigma$  o también llamada escala. Al modificar la escala de la función se modifica proporcionalmente el

ancho de la curva característica de un LoG (Figura 3.16), también llamada sombrero mexicano por su forma.

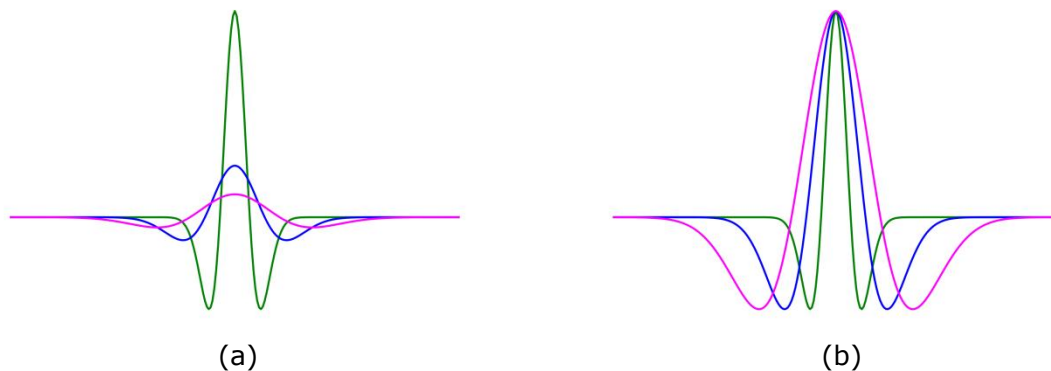


Figura 3.16. En (a) se muestra en verde, azul y rosa las respuestas de la función LoG con  $\sigma=0.2$ ,  $\sigma=0.4$  y  $\sigma=0.6$  respectivamente. En (b) se muestran las mismas respuestas pero normalizadas

LoGs de diferentes escalas tienen curvas de diferentes proporciones y también responden de diferente manera ante una misma señal escalón. En la Figura 3.17 se muestran tres casos donde un escalón unitario es convolucionado con un LoG de diferente  $\sigma$  en cada caso. El ancho de la curva aumenta proporcionalmente conforme aumenta su  $\sigma$  y la respuesta que genera también es cada vez de mayor amplitud. Cuando un LoG es muy estrecho como para cubrir la mayor parte del escalón ocurre el caso de la Figura 3.17 (b) donde es generada una doble respuesta en cada borde, a medida que la  $\sigma$  aumenta la curva es capaz de cubrir una mayor porción del escalón y su respuesta tiene mayor amplitud al mismo tiempo que las espigas positivas se van juntando hasta que finalmente se juntan en una sola de una amplitud aún mayor como ocurre en el caso de la Figura 3.17 (f). En ese sentido, el operador LoG es particularmente útil detectando *blobs*, pues se puede determinar que existe un *blob* donde la respuesta del LoG tiene solo una espiga positiva de amplitud máxima [20].

En la Figura 3.18 se muestra la aplicación del filtro de la Figura 3.15 (d) para detectar estrellas en una imagen de la NASA mediante detección de *blobs*. La respuesta del filtro muestra regiones circulares con máxima intensidad las cuales indican la presencia de un *blob* (estrella), sin embargo, la respuesta de las estrellas de mayor tamaño es en forma de aros con menor intensidad, estos aros son la doble respuesta del LoG. Esto significa que la escala  $\sigma$  del filtro no es suficientemente grande para detectar esas estructuras de mayor tamaño, la solución más evidente es filtrar con un LoG de mayor tamaño, sin embargo, como se puede ver en la Figura 3.18 (c) la desventaja es la pérdida de detalles y de *blobs* que ya habían sido realizados en la Figura 3.18 (b), esto supone la necesidad de emplear LoGs con diferentes escalas y para cada respuesta obtener información de interés, este procedimiento de filtrar una imagen con escalas sucesivas recibe el nombre de análisis múltiescala [11].

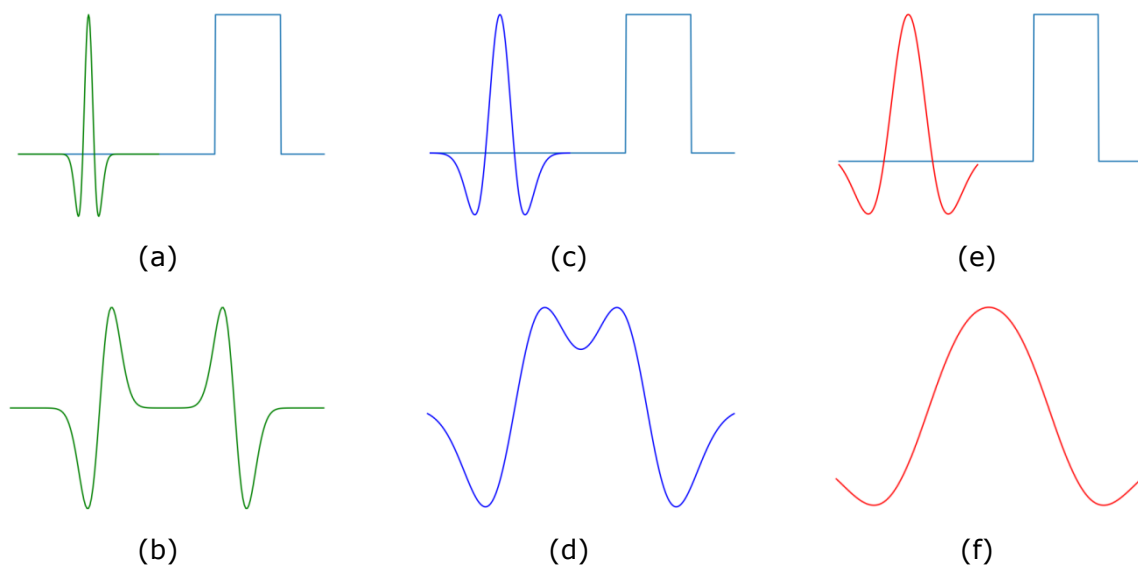


Figura 3.17. En (a), (c) y (e) se muestra en cada una dos señales, la del lado izquierdo corresponde a una función LoG con  $\sigma=0.2$ ,  $\sigma=0.5$  y  $\sigma=0.8$  respectivamente y la del lado derecho a un escalón. (b), (d) y (f) son las respuestas de convolucionar las señales mostradas en (a), (c) y (e) respectivamente

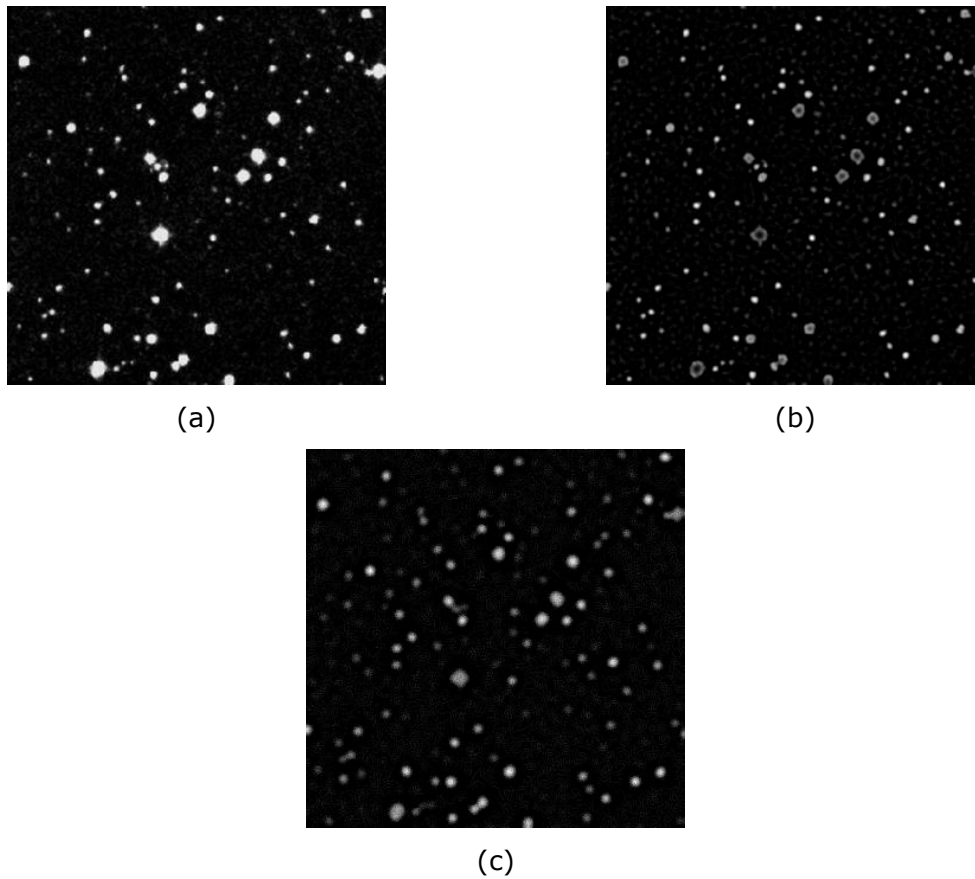


Figura 3.18. (a) es una fotografía de estrellas, las cuales son un ejemplo de *blobs*. En (b) y (c) se muestra la respuesta de un aplicar un LoG con escalas diferentes siendo mayor la de (c)

### 3.6 Análisis multiescala

Un LoG permite extraer de una imagen *blobs* con un tamaño determinado por la escala  $\sigma$  del LoG, con una escala pequeña se pueden resaltar blobs pequeños y los grandes se verán atenuados y presentaran una doble respuesta en forma de aro. Mientras que con una escala grande se pueden resaltar blobs de mayor tamaño pero los pequeños son atenuados, difuminados e incluso suprimidos como puede observarse en la Figura 3.18. Para extraer todos los blobs de todos los tamaños o los de un rango de tamaño es necesario hacerlo por escalas, es decir aplicar un análisis multiescala.

Aunque en principio suena simple realizar un análisis multiescala, en realidad el proceso implica discretizar la función de la ecuación Ec. 3.20.3 para calcular los *kernel* para cada escala que se vaya a emplear, lo cual computacionalmente es complejo y costoso. Debido a que la implementación de la derivada está en términos de diferencias, es posible calcular la derivada de una función Gaussiana mediante diferencias (Figura 3.19), de este modo es posible aproximar un función LoG mediante una diferencia de Gaussianas mejor conocida como DoG [25]. Además, la convolución de dos Gaussianas de misma escala tiene por respuesta otra Gaussiana de mayor escala que las anteriores. Esto implica que es posible filtrar una imagen sucesivamente con un solo filtro Gaussiano y obtener el mismo resultado que si hubiera sido filtrada con un *kernel* de mayor escala. Al final, el proceso de análisis multiescala consiste en convolucionar una imagen de entrada con un filtro Gaussiano  $K$  veces, obtener la diferencia de la convolución  $k-1$  y  $k$  y determinar los puntos de valor máximo [21] (Figura 3.20).

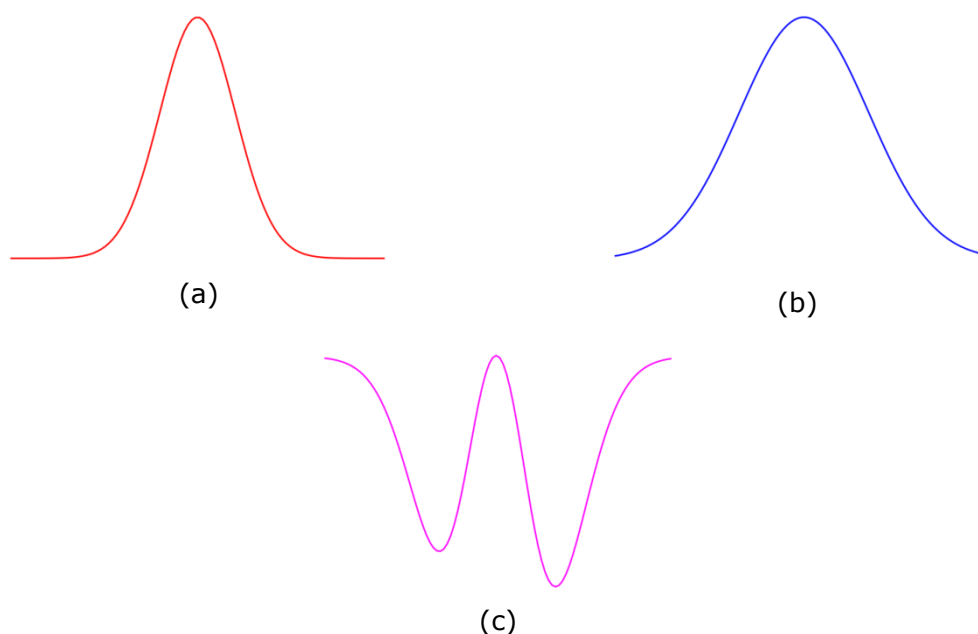


Figura 3.19. En (c) se muestra el resultado de restar las funciones Gaussianas (a) – (b), donde (b) tiene una  $\sigma$  mayor que (a). Se puede apreciar que la diferencia de Gaussianas ofrece una buena aproximación de un LoG

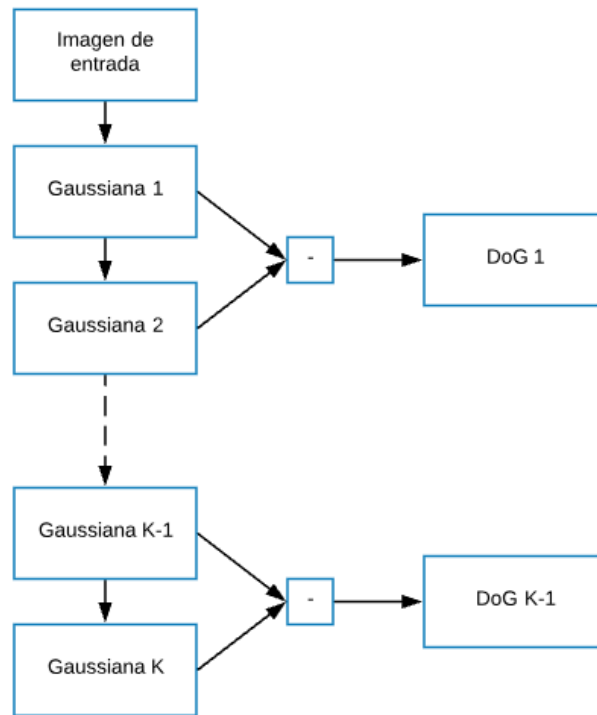


Figura 3.20. Diagrama de análisis multiescala

Conforme aumenta el número de convoluciones sucesivas con el *kernel*, cada  $\sigma_k$  será mayor que la  $\sigma_{k-1}$ , en ocasiones es conveniente que la diferencia de escala entre cada convolución sea mayor o menor y esto es controlable mediante el tamaño del filtro usado, sin embargo, mientras mayor es el *kernel* mayor es el costo computacional de la convolución. Un enfoque que permite aumentar rápidamente la escala sin tener que incrementar el costo computacional es el submuestreo, es decir, reducir la resolución de la imagen [20]. El submuestreo puede hacerse de varias maneras y se requiere aplicar un filtro de aproximación, el cual es generalmente un Gaussiano, antes de submuestrear para evitar el efecto de bloque. Una opción para submuestrear una imagen es eliminar columnas pares o impares y lo mismo para las filas, sin embargo, esto supone una medida arbitraria y la pérdida de información. Un enfoque más adecuado es el de la interpolación, el cual consiste en generar el valor de un píxel en la nueva imagen interpolando los valores en una ventana  $n \times n$  de la imagen original [25]. Aunque el submuestreo puede ser realizado con diferentes proporciones, la más usual es la  $2 \times 2$ , es



decir, un píxel en la nueva imagen es calculado con los valores de una ventana  $2 \times 2$  de la imagen original, de modo que al final, si las dimensiones de la imagen de entrada son  $m \times n$ , la imagen resultante tendrá dimensiones  $(m/2) \times (n/2)$ .

Al agregar el submuestreo al análisis multiescala se está agregando también un análisis multiresolución (Figura 3.21). Después de realizar la diferencia de Gaussianas con muestras de menor resolución es necesario un sobremuestreo para hacerlas crecer hasta el tamaño original para así poder compararlas con las demás muestras y determinar la máxima respuesta de cada *blob* [21].

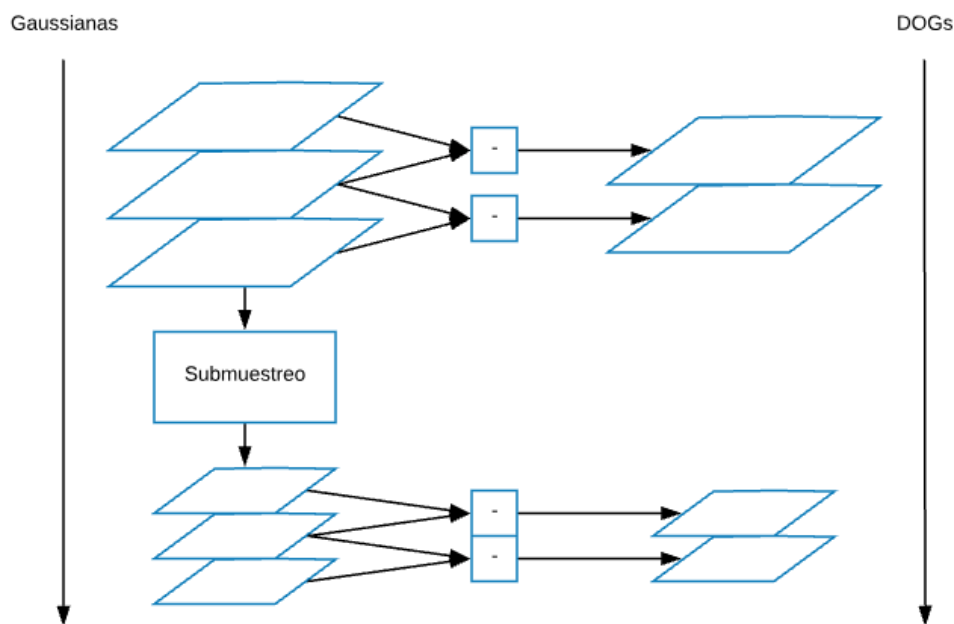


Figura 3.21. Análisis multiescala y multiresolución

Después de conseguir realizar los *blobs* de diversos tamaños es posible aplicar algún umbral para segmentar las regiones de máxima intensidad y con ello terminar de segmentar los *blobs*. Sin embargo, un umbral global tiene el mismo problema que la ecualización global: pequeñas regiones podría resaltar en su vecindario lo suficiente para ser de interés, pero tener una intensidad menor en comparación con otras regiones de mayor tamaño y así verse influenciadas negativamente. Por

ello, al igual que con la ecualización, en muchas ocasiones conviene hacer el procesamiento de manera local y existen varios enfoques para determinar el umbral en un vecindario, por ejemplo mayor a la media, mayor a la mediana, igual al máximo o empleando la desviación estándar [24]. Una vez más, son los filtros Gaussianos los que ofrecen un enfoque bastante interesante, el cual consiste en filtrar una imagen como la de la Figura 3.22 (c) con un filtro Gaussiano esto produce *blobs* de menor amplitud y más anchos, de modo que al obtener los píxeles de la imagen antes de filtrar que son mayores a sus homólogos en la imagen filtrada se obtienen las regiones de máxima intensidad de la imagen considerando su vecindario (Figura 3.23). Al final de aplicar el umbral se obtiene una imagen de ceros y unos sobre la que es posible aplicar otro tipo de técnicas dada su naturaleza binaria.

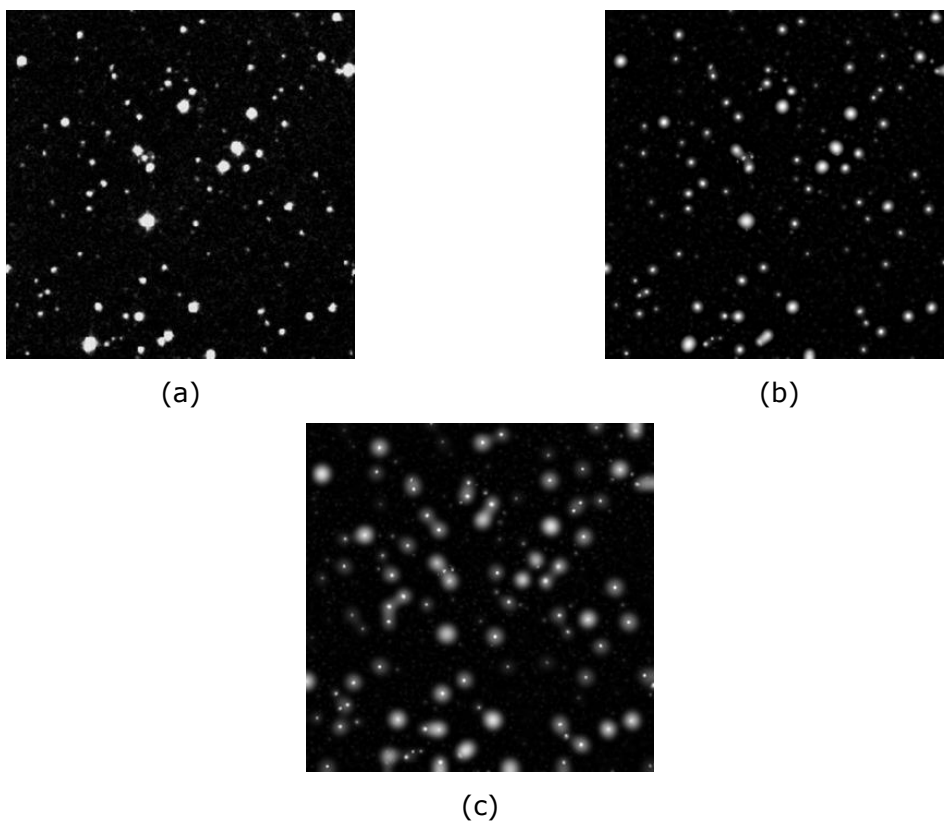


Figura 3.22. En (b) se muestra el resultado aplicar en (a) análisis mutiescala con un filtro Gaussiano de 5 x 5 y 20 DoGs. Mientras que en (c) se realizó un análisis con un filtro Gaussiano 5 x 5 y dos submuestreos, cada uno con 5 DoGs

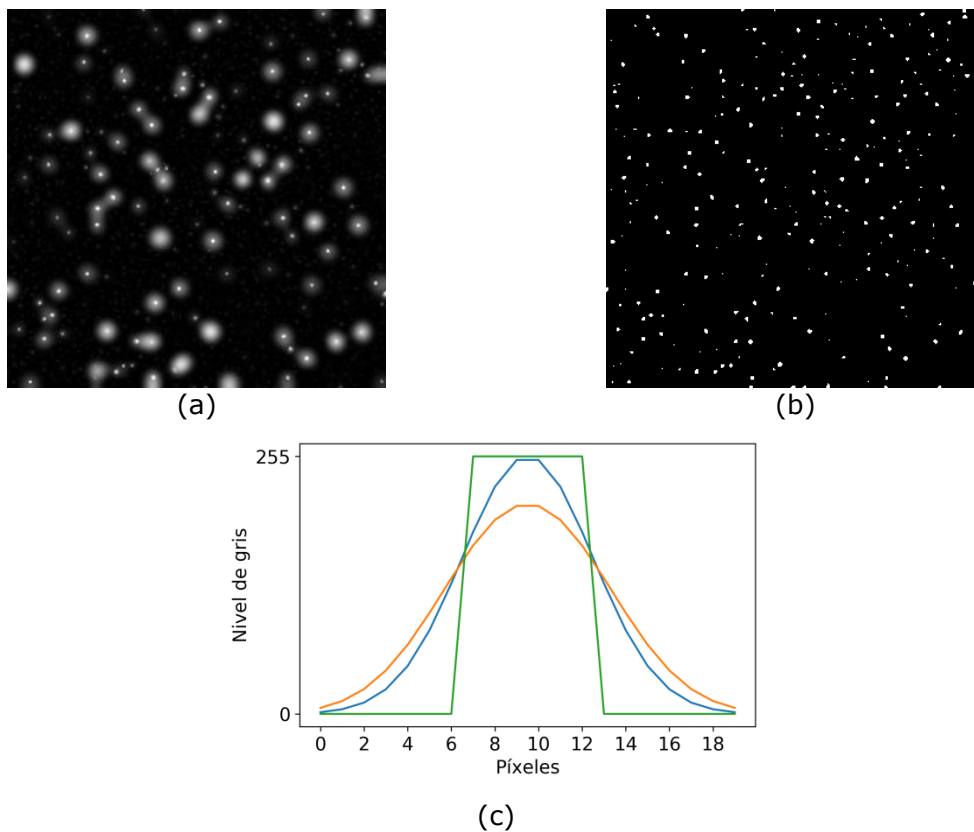


Figura 3.23. En (b) se muestra el resultado de binarizar la imagen (a) con ayuda de un filtro Gaussiano. En (c) se muestra en azul la curva de un *blob* vista en una dimensión, en naranja la respuesta después de aplicar un filtro Gaussiano a la curva azul y en verde, de forma binaria, las posiciones donde la curva azul es mayor que la naranja

### 3.7 Morfología matemática

Después de realzar y segmentar los *blobs* de una imagen de entrada, se obtiene una imagen binaria donde los píxeles que pertenecen a un *blob* tienen 1 por valor o blanco, un *blob* también es un objeto dentro de la imagen y como tal tiene propiedades estructurales como área, circularidad y bordes. Matemáticamente es posible definir un objeto como un conjunto. El conjunto universal de una imagen binaria y se denota como  $Z^2$  debido a que al solo existir dos valores de intensidad, negro y blanco, solo se requiere un par coordenado  $(x,y)$  para representar un elemento que pertenece a un conjunto. Al definir a los objetos contenidos

en una imagen como conjuntos es posible aplicar operaciones matemáticas sobre ellos con el objetivo de modificar o describir su estructura. Ciertamente esto podría ser aplicado a imágenes en escala de grises, en tal caso además de las coordenadas  $(x,y)$  se consideraría un tercer valor que es el de gris, de modo que ahora la imagen sería representada como el conjunto universal  $Z^3$  [20].

Sea la imagen  $Z^2$ , los conjuntos  $A$  y  $B$ , tal que  $B \subseteq Z^2$  y  $A \subseteq Z^2$  y las operaciones entre los conjuntos  $A$  y  $B$  que se muestra en la Tabla 3.2, al tratarse de una imagen binaria donde un píxel pertenece o no a uno o más conjuntos, se pueden definir estas operaciones de conjuntos como operaciones de álgebra booleana tal como se muestra en la Tabla 3.3.

Tabla 3.2. Operaciones de conjuntos

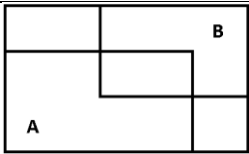

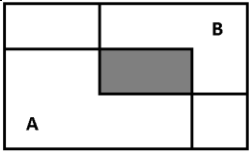
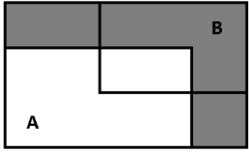

| Operación  |   |
|------------|--|
| $A \cup B$ |  |
| $A \cap B$ |  |
| $A^c$      |  |
| $A - B$    |  |

Tabla 3.3. Equivalencia de operaciones de conjuntos con operaciones booleanas

| Operación de conjuntos | Operación booleana |
|------------------------|--------------------|
| $A \cup B$             | A OR B             |
| $A \cap B$             | A AND B            |
| $A^c$                  | NOT A              |
| $A - B$                | A AND (NOT B)      |

Como es posible observar en la Tabla 3.2, al operar los objetos como conjuntos se obtienen objetos con estructuras diferentes. El término morfología en multitud de disciplinas hace referencia a la estructura de entidades, en el caso de procesamiento de imágenes digitales, la morfología hace referencia la estructura de los objetos contenidos en una imagen (como los *blobs*) y en particular la morfología matemática lidia con cómo modificar dichas estructuras con las operaciones matemáticas previamente mostradas o extraer datos importantes como bordes, esqueletos, área, circularidad, convexidad entre otros [23].

Una vez dicho que cada *blob* en una imagen binaria  $Z^2$  es un conjunto matemático de píxeles, para modificar su forma con operaciones como las de la Tabla 3.3 hace falta un segundo conjunto con el cual realizar la operación, con excepción de la *NOT*. Este segundo conjunto es llamado Elemento Estructurante o *SE* (por su nombre en inglés *Structuring Element*) el cual, es operado en cada posición de  $Z^2$ , muy similar al *kernel* y la convolución, con la diferencia de que en este caso se trata de una operación no lineal, es decir, la respuesta del *SE* sobre  $Z^2$  depende de las condiciones definidas para la operación, debido a que se trata de una imagen binaria la respuesta de un *SE* es 0 o 1 dependiendo de si las condiciones de la operación se cumplen. Además, a diferencia de un *kernel*, en un *SE* es necesario especificar una posición central, de modo que su respuesta sea asignada al píxel  $P$  que coincida con tal posición [20].

Sea el conjunto  $A$ , tal que  $A \subseteq Z^2$ ,  $B$  un  $SE$  y la operación  $A \ominus B = \{p \in Z^2: (B)_p \subseteq A\}$ , en la Figura 3.24 se muestran los resultados (d) y (e) de aplicar la operación  $A \ominus B$  con (a) y los  $SE$  (b) y (c) respectivamente. En ambos se casos, se obtuvo una estructura más delgada y pequeña que la original, por tal motivo la operación  $A \ominus B$  recibe el nombre de erosión o adelgazamiento [23].

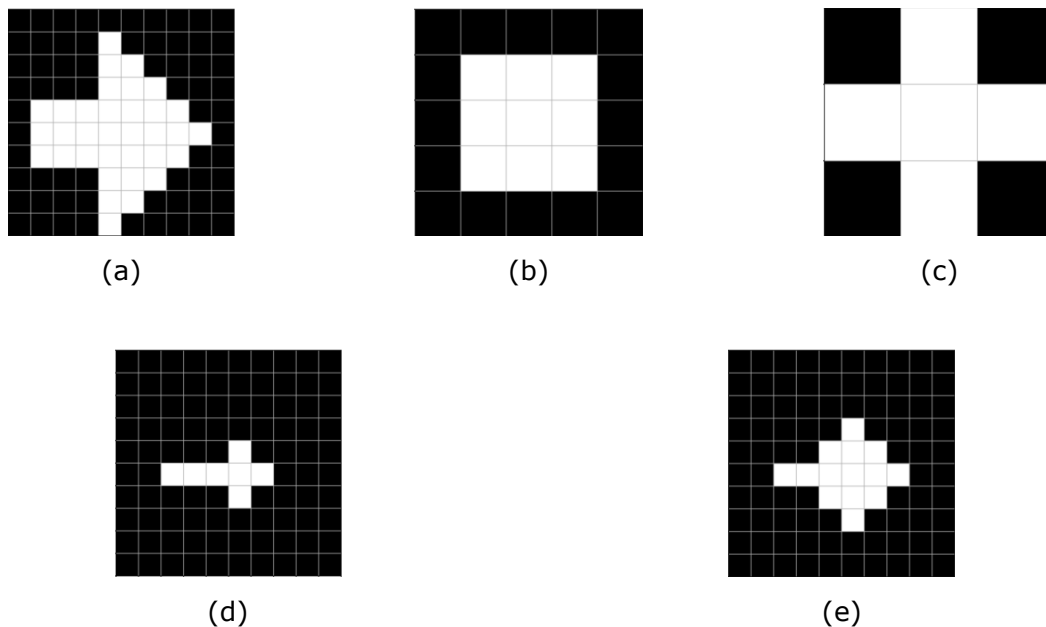


Figura 3.24. (a) Es una imagen binaria de entrada, (d) y (e) son el resultado de realizar erosión con los  $SE$  (b) y (c) respectivamente

En la Figura 3.25 se muestra en (c) y (f) el resultado de aplicar la operación  $A \oplus B = \{p \in Z^2: P = a + b, a \in A, b \in B\}$  en (a) con el  $SE$  (b) y en (d) con el  $SE$  (e) respectivamente. En ambos resultados se obtiene un objeto más ancho que el original, por tal motivo la operación  $A \oplus B$  es llamada dilatación o ensanchamiento [21]. Es importante notar que los resultados mostrados en el Figura 3.24 son las entradas de la dilatación en la Figura 3.25 y a pesar de haber usado el mismo  $SE$  con el que se erosionó, los resultados no fueron las entradas en la erosión, es decir, aunque la dilatación es la operación opuesta a la erosión no se cumple que  $(A \oplus B) \ominus B = A$ , así como tampoco se cumple que  $(A \ominus B) \oplus B = A$ , lo

anterior lejos de ser un problema, significa una característica de la morfología matemática que se puede aprovechar para combinar erosión con dilatación de muchas formas para dar origen a operaciones más complejas como lo son la apertura y el cierre.

La apertura se define como  $A \circ B = (A \ominus B) \oplus B$ , aprovecha la erosión para eliminar de la imagen pequeños objetos que no son de interés y la dilatación para aproximar a su tamaño original a aquellos que sí lo son [21]. Otro efecto de esta operación consiste en que suaviza los contornos de un objeto. La operación de cierre se define como  $A \cdot B = (A \oplus B) \ominus B$ , en este caso se aprovecha el ensanchamiento de los objetos para rellenar pequeños huecos y así obtener una estructura uniforme, mientras que una siguiente etapa de erosión aproxima los objetos a su tamaño original [21]. En este proceso también se obtiene contornos más suaves. A la vez, apertura y cierre pueden ser usados combinados para eliminar objetos indeseables con apertura y rellenar los huecos presentes en los objetos con cierre. Realmente es posible realizar infinidad de combinaciones con infinidad de  $SE$  de acuerdo a los objetivos del procesamiento.

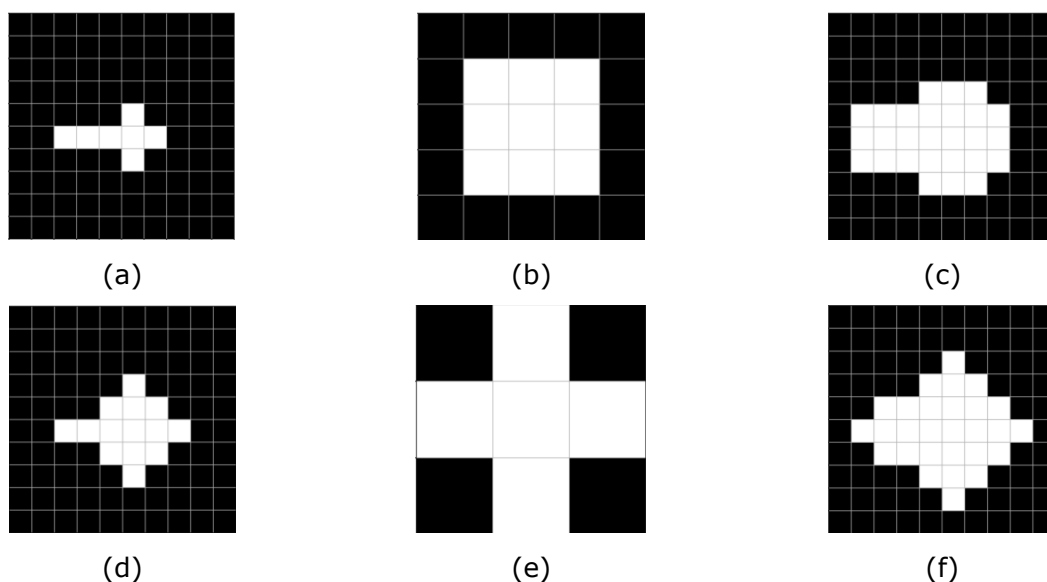


Figura 3.25. En (c) y (f) se muestra el resultado de dilatar (a) y (b) con (b) y (e) respectivamente

## 4. Definición de requerimientos

De la definición del problema en la sección 1.1 se tiene la necesidad de desarrollar un sistema que identifique de manera automática las neuronas contenidas en los videos generados de un experimento característico como el descrito en la sección 2.2. Debido a que los investigadores en el IFC operan bajo protocolos y métodos establecidos, la adopción del nuevo sistema no debe implicar la alteración en la manera en que se realizan las actividades previas o posteriores, sino más bien se busca automatizarlo. Para ello es necesario definir los requerimientos de los expertos como usuarios, así como los datos de entrada y de salida del sistema.

Además de indicar la ruta donde se encuentran los videos del experimento a procesar, el usuario desea poder indicar un tamaño mínimo de neurona, tal que aquellas ROIs que tengan un tamaño menor al indicado serán automáticamente descartadas. Asimismo, para el cálculo de los transitorios debe poder especificar la proporción del área a considerar respecto al radio de cada ROI y si serán consideradas las regiones de interés que puedan llegar a estar dentro de dicha área. En la Tabla 4.1 se describen los datos de entrada para el sistema.

El usuario espera que el sistema entregue por resultados: un archivo donde estén contenidas las coordenadas  $(x,y)$  de las neuronas, su radio  $r$  y un identificador numérico con el mismo formato de los RoiSet.zip generados con ImageJ y la matriz de actividad almacenada en un archivo .csv. Es decir, las salidas del sistema deben corresponder en formato con los archivos generados actualmente al final de un experimento. En la Tabla 4.2 se describen los datos de salida del sistema.



Tabla 4.1. Descripción de los datos de entrada del sistema.

| Entrada                                     | Descripción   | Características   |
|---|---|---|
| Ruta  | Ruta del directorio donde se encuentran almacenados los videos a analizar   | Cadena de caracteres  |
| Videos                                      | Videos obtenidos durante un experimento característico, en promedio cada experimento consta de cuatro a seis vídeos                         | Videos en formato .AVI con resolución de 512 X 512 píxeles, en escala de grises de 255 niveles, 3600 cuadros por video y frecuencia de 20 cuadros por segundo |
| Área mínima de neurona                      | Toda posible neurona detectada será descartada automáticamente si tiene un área menor que el indicado en este dato                          | Entero sin signo  |
| Método para calcular transitorios de calcio | Indica si el transitorio de calcio de cada neurona será calculado considerando las neuronas vecinas   | Booleano  |
| Proporción del radio $R$ respecto a $r$     | Indica la proporción del radio $R$ del área circular considerada para calcular los transitorios de calcio respecto al radio $r$ de cada ROI | Entero sin signo  |

Otros requerimientos definidos por el usuario son:

- El sistema debe estar contenido en un librería de programación
- El sistema no debe requerir de complejas implementaciones de hardware
- El tiempo del procesamiento debe ser menor que el tiempo que actualmente los investigadores dedican al etiquetado de los videos

Tabla 4.2. Descripción de los datos de salida del sistema

| Salida              | Descripción   | Características   |
|---------------------|---|---|
| RoiSet.zip          | Formato de archivo en el que ImageJ guarda las etiquetas hechas por el experto en un video  | Por cada etiqueta se guardan las coordenadas $(x,y)$ de su centro, un identificador numérico y el radio de la neurona. Todos los datos son de tipo entero sin signo |
| Matriz de actividad | Matriz de $N \times F$ donde $N$ es el número de neuronas detectadas y $F$ el número total de cuadros en el experimento. En la matriz 0 representa estado de reposo y 1 actividad asociada a potenciales de acción. | Matriz binaria en formato .csv  |

## 5. Diseño e implementación

Como se puede apreciar en los trabajos relacionados de la sección 2.3, las técnicas de procesamiento de imágenes digitales ofrecen un poderoso enfoque y han sido probadas con éxito en otras tareas relacionadas con las ciencias biomédicas [4]. Por tal motivo, para el desarrollo del Sistema de Reconocimiento Automático de Neuronas en Microscopía de Epifluorescencia (SRANME), se ha optado por el uso de técnicas de procesamiento de imágenes digitales buscando mayor robustez que las implementaciones hechas en los trabajos relacionados.

SRANME está diseñado en dos etapas: Segmentación y validación. Durante la segmentación se aplican técnicas de procesamiento de imágenes digitales para identificar potenciales neuronas y se construye un mapa de ROIs con la misma resolución que la de los videos; en la segunda etapa, mediante el cálculo de los transitorios de calcio y potenciales de acción se validan las ROIs que presentan actividad neuronal y con ellas se construye la Matriz de actividad y el Mapa de neuronas, finalmente, de este último se genera el RoiSet.zip.

En la etapa de segmentación son segmentadas un conjunto de potenciales neuronales y tiene por salida un mapa de regiones de interés ROIs, en este mapa cada región de interés ha sido etiquetada de tal modo que todos los píxeles que pertenecen a esta tienen el mismo valor y es único para cada una. La etapa de validación consiste en evaluar cada ROI mediante su señal de actividad, recibe por entrada los videos del experimento sin ningún tipo de procesamiento y el mapa de ROIs generado en la primera etapa, por salida entrega una matriz de actividad de las neuronas confirmadas donde los renglones corresponden a cada una de ellas y las columnas al total de cuadros considerando todos los videos y el Mapa de neuronas resultante de haber eliminado aquellas ROIs que no presentaron potenciales de acción. En la Figura 5.1 se

muestra flujo de trabajo del SRANME hasta generar las salidas esperadas por el usuario.

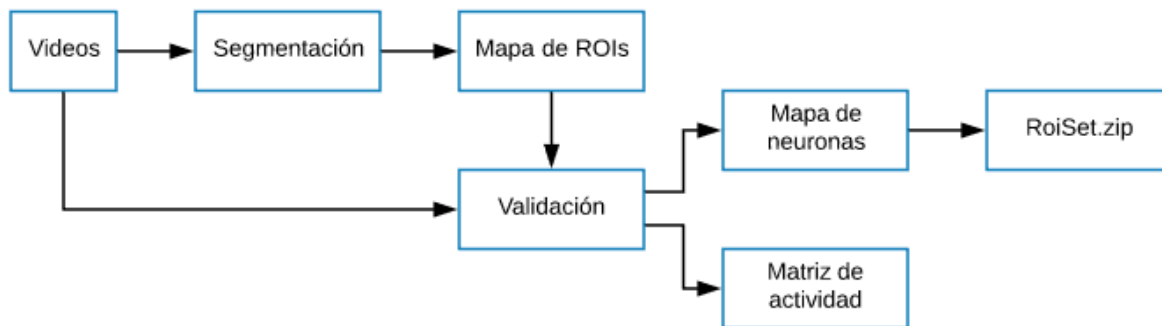


Figura 5.1. Procesamiento de los videos hasta generar las salidas esperadas

## 5.1 Segmentación

La etapa de segmentación está formada por cuatro procesos (Figura 5.2), el primero mejora el contraste de la imagen en cuestión, el siguiente suprime en lo posible el ruido para entonces continuar con la identificación de neuronas mediante detección de *blobs*, finalmente se tiene el mapeo en donde se genera un mapa con las ROIs segmentadas.

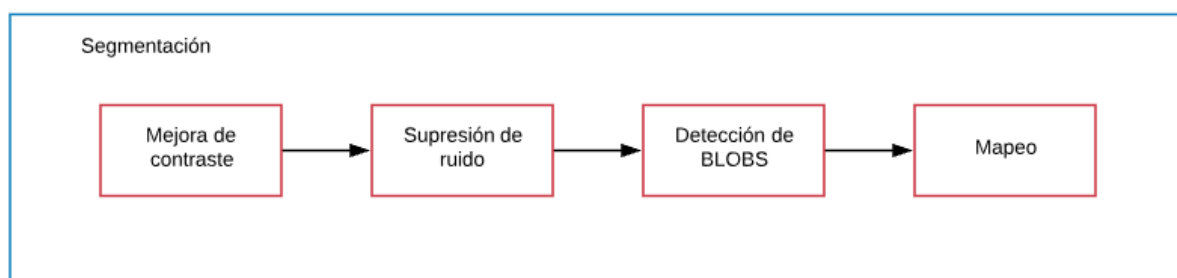


Figura 5.2. Procesos hechos durante la etapa de segmentación

En esta etapa se recibe un video a la vez, se carga de manera secuencial cada cuadro del video y es ecualizado localmente utilizando una ventana 3 x 3 para mejorar el contraste. Para suprimir el ruido se emplean los cuadros ya ecualizados para formar una matriz de tres dimensiones correspondiente a tres cuadros consecutivos de modo que tiene una dimensión 3 X 512 X 512. A continuación es convolucionada con

un filtro Gaussiano 3 X 3 X 3 y el resultado es llamado Imagen A (Figura 5.3). La convolución en tres dimensiones sigue la misma mecánica que en dos dimensiones, en este caso el filtro es construido convolucionando un vector columna con cada vector renglón de un filtro Gaussiano 3 X 3 (Figura 5.4). En la Figura 5.5 se muestran los resultados de aplicar los procesos de mejorar de contraste y supresión de ruido.

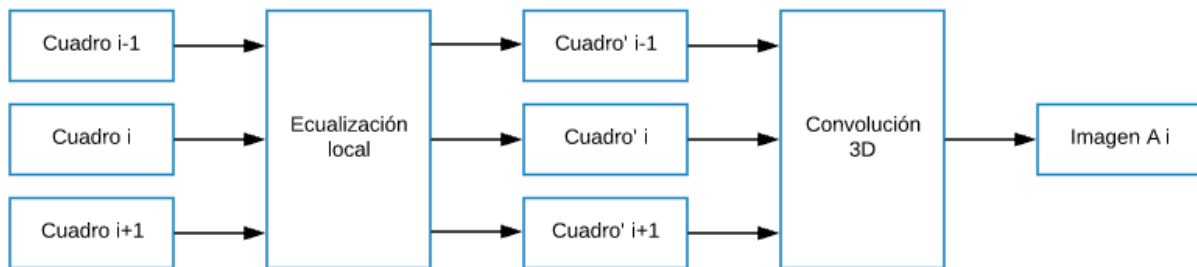


Figura 5.3. Proceso de mejoramiento de la imagen mediante ecuación local y convolución 3D

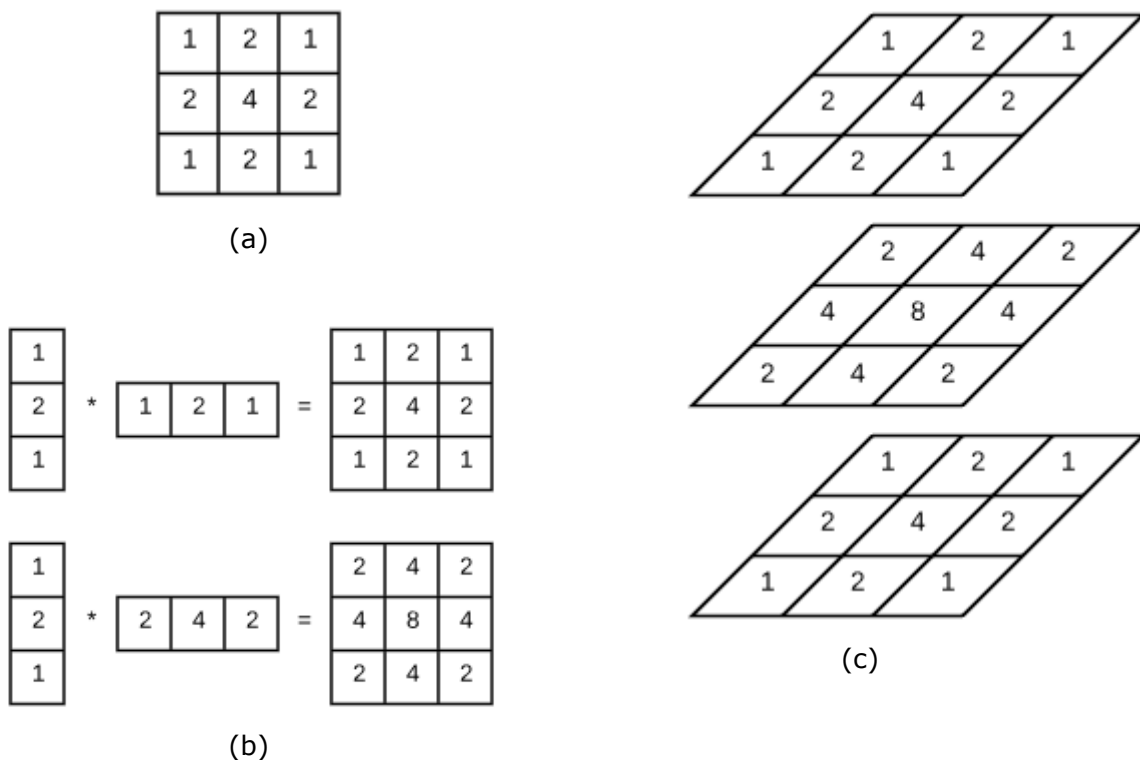


Figura 5.4. En (a) se muestra un filtro Gaussiano 3 X 3, en (b) los resultados de convolucionar un vector columna por cada vector fila del filtro (a) y en (c) la configuración un filtro Gaussiano 3 X 3 X 3 a partir de los resultados en (b)

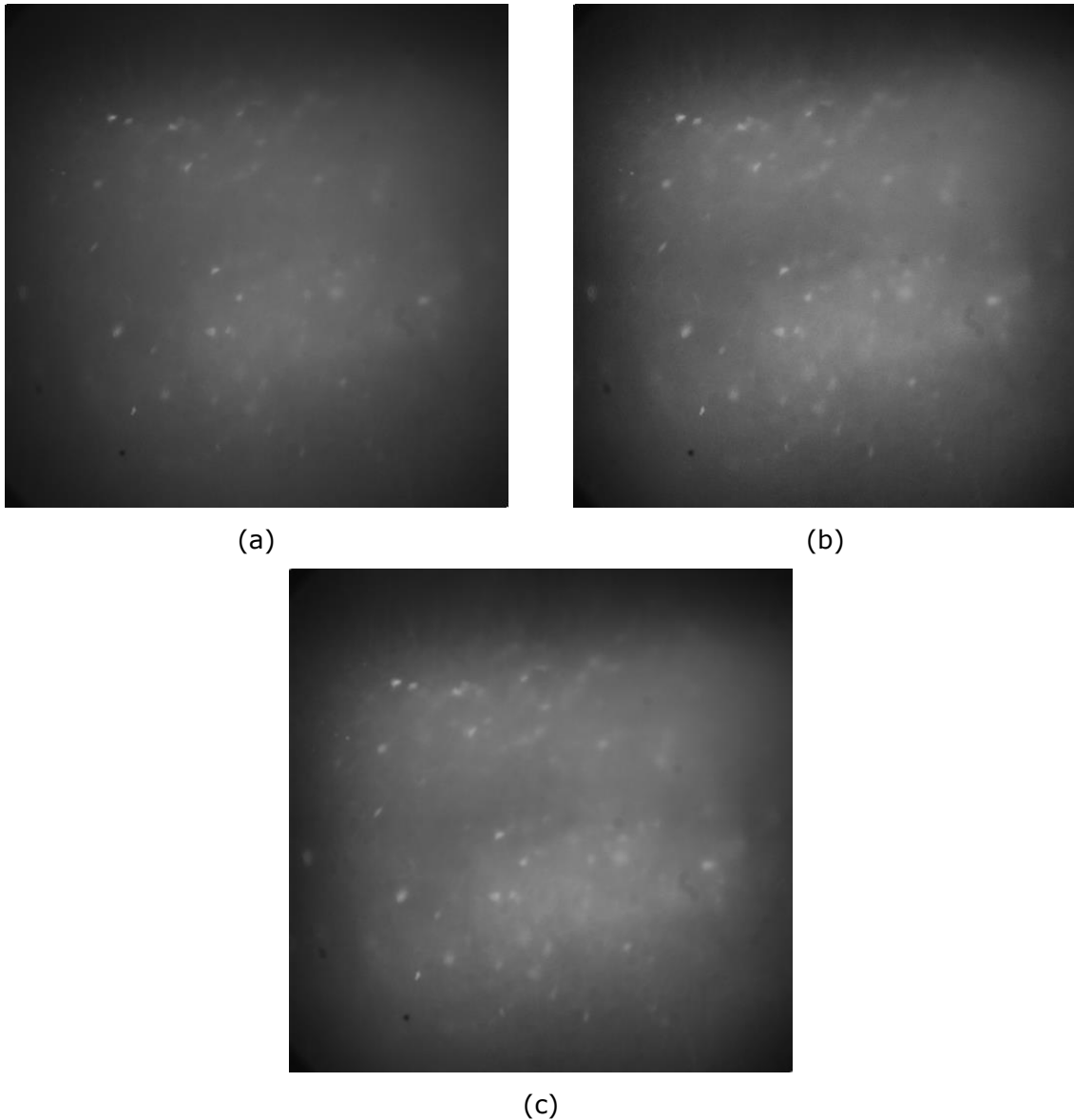


Figura 5.5. En (a) se muestra un cuadro de video sin procesamiento, en (b) el resultado aplicar ecualización local y en (c) el resultado de la convolución 3D, esta última es un ejemplo de Imagen A del diagrama de bloques de la Figura 5.3

Para la detección de *blobs* la Imagen A es sometida a un análisis multiescala y multiresolución en un proceso de tres diferencias de Gaussianas por resolución con un filtro Gaussiano 5 x 5, la primera resolución corresponde a la original del video y la segunda al resultado de un submuestreo 2 X 2, entonces se genera una Imagen B donde el valor de cada píxel corresponde al máximo valor alcanzado en todas las

diferencias de Gaussianas (Figura 5.6), aquellas DoGs que corresponden a la menor resolución deben ser sobremuestreadas para que tengan el tamaño original y así poder comparar el valor de cada píxel, para evitar efectos indeseados se emplea interpolación lineal con el sobremuestreo.

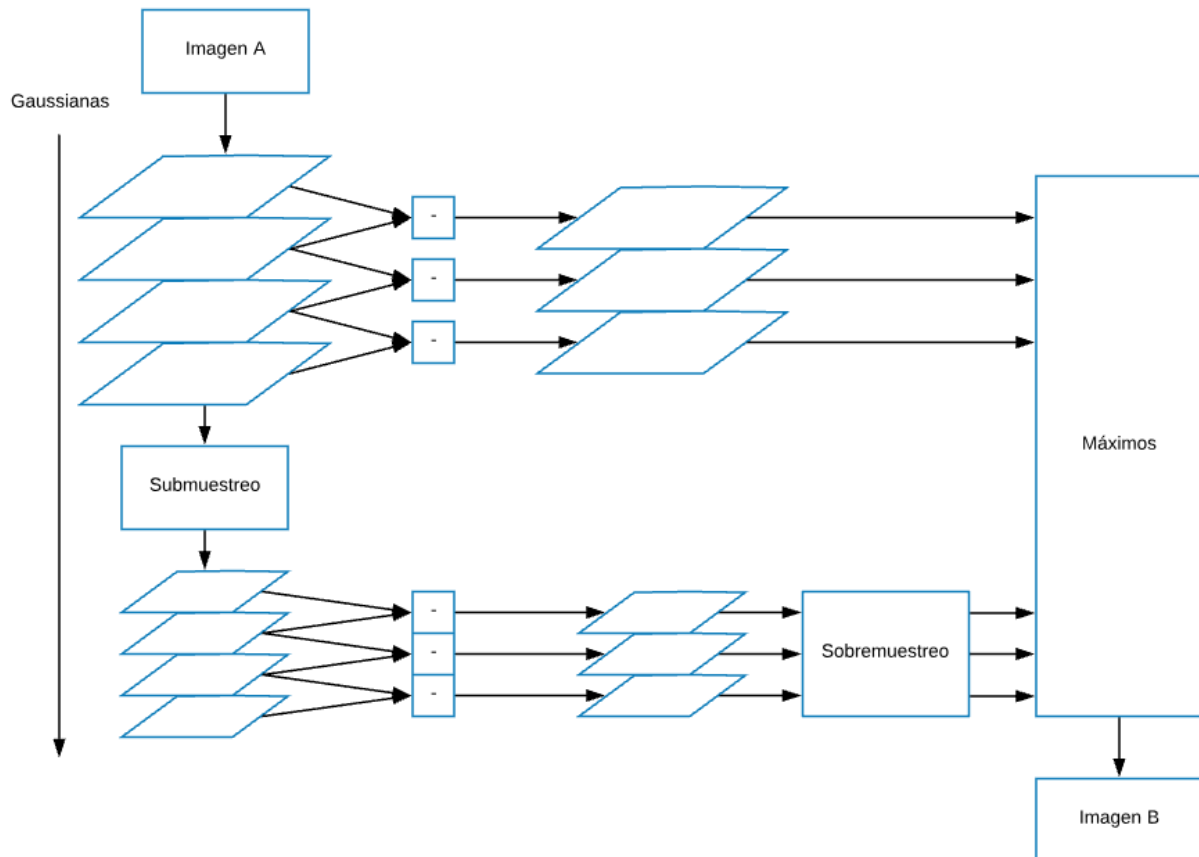


Figura 5.6. Procesamiento multiescala y multiresolución aplicado para generar una Imagen B para cada Imagen A

Con el fin de segmentar los *blobs* detectados se aplica umbralización a la Imagen B con un umbral de tres veces la desviación estándar de los valores de la Imagen B, las imágenes binarias que resultan de este proceso son mezcladas cada cinco cuadros aplicado la operación booleana *OR* para generar una imagen donde están contenidos todos los píxeles que podrían pertenecer a una neurona y es llamada Imagen C (Figura 5.7). En la Figura 5.8 se muestra un ejemplo de las imágenes B y C generadas de los procesos ilustrados en las figuras 5.6 y 5.7 respectivamente.

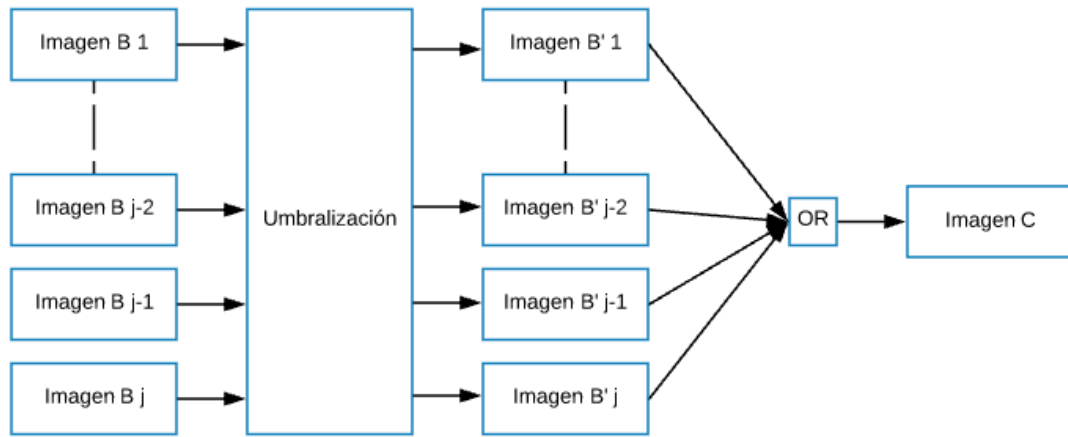


Figura 5.7. Proceso para segmentar los blobs detectados en el análisis multiescala y multiresolución y generar la Imagen C aplicando una operación *OR* con las imágenes B binarizadas

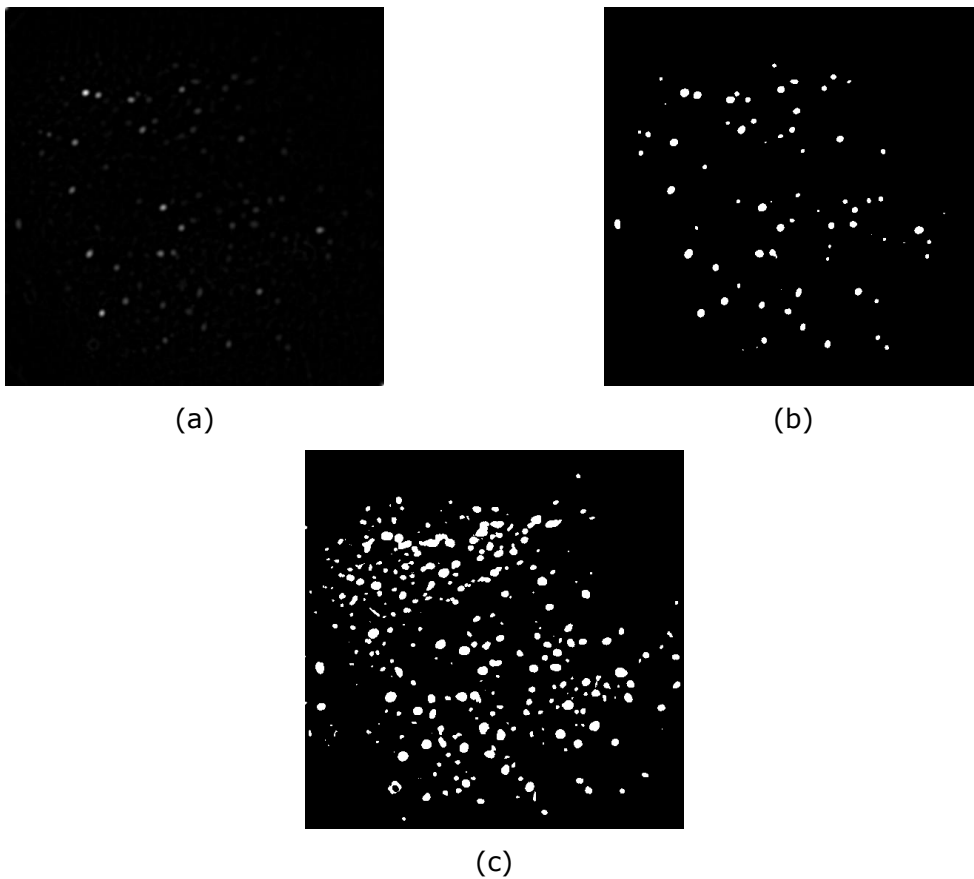


Figura 5.8. En (a) se muestra el resultado del análisis multiescala y multiresolución de la Imagen A en la Figura 5.5 (c). En (b) el resultado de aplicar umbralización a (a) y en (c) el resultado de aplicar *OR* con todas las imágenes binarias del experimento



Hasta este punto se han segmentado todas las posibles neuronas, sin embargo, falta delimitar cada una, darles una etiqueta y eliminar aquellas que no cumplen con un tamaño mínimo. Para ello, en el mapeo primero se agrupan las ROIs por píxeles conectados y aquellas que no cumplen con el área mínima indicada por el usuario son eliminadas y el resultado es llamado Imagen D. Para dividir cada ROI en la que puede haber dos o más neuronas unidas se realiza una serie de erosiones controladas, cada ROI es erosionada hasta que su área es menor o igual a 36 píxeles, entonces se le asigna una etiqueta y se guarda el número de erosiones realizadas hasta ser etiquetada. Después tiene lugar un proceso de dilatación controlada donde cada ROI etiquetada es dilatada el mismo número de veces que fue erosionada con la limitación de área de la ROI original, esto permite dividir dos o más regiones de interés traslapadas y mantener su tamaño original. Para evitar conflictos en los bordes, los píxeles donde hay traslape son descartados. Al final de este proceso se obtiene un mapa preliminar con potenciales neuronas llamado Mapa de ROIs. En la Figura 5.9 se muestra el proceso de mapeo de la Imagen C, pasando por la Imagen D, hasta llegar al Mapa de ROIs, mientras que en la Figura 5.10 se muestra un ejemplo de Imagen D, las salidas de los procesos de erosión y dilatación y del Mapa de ROIs generado.

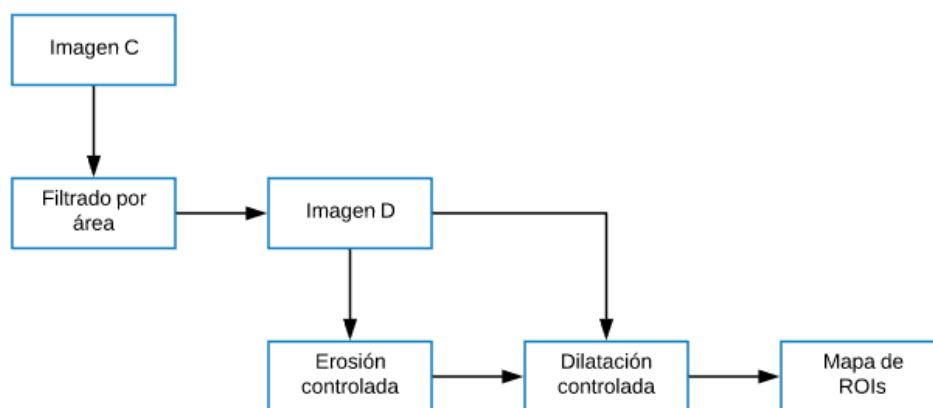


Figura 5.9. Proceso de mapeo de las ROIs como último pasó de la etapa de segmentación

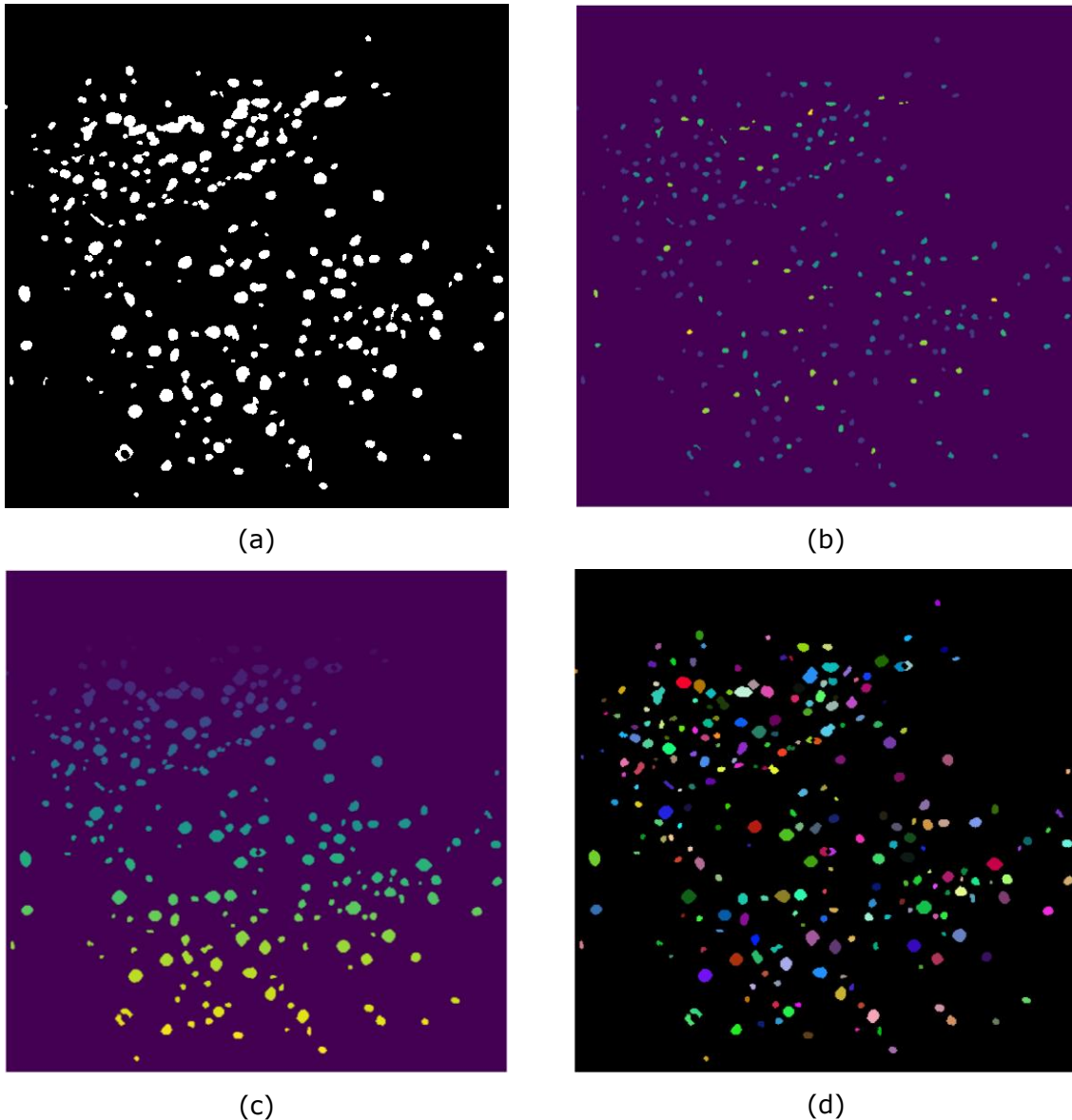


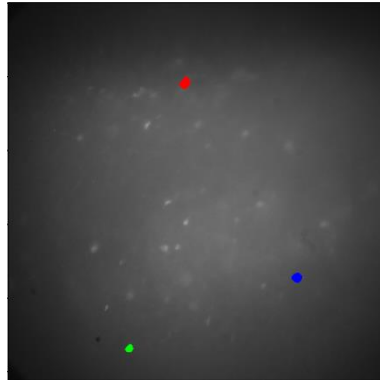
Figura 5.10. En (a) se muestra la Imagen D después de eliminar pequeños objetos, tan pequeños como para ser considerados ROIs. En (b) se muestra el resultado de aplicar erosión controlada en (a) , cada intensidad de color representa un núcleo de ROI diferente. (c) es el resultado de dilatar los núcleos en (b) teniendo como referencia a (a). (d) es el Mapa de ROIs donde cada color representa una ROI diferente

## 5.2 Validación

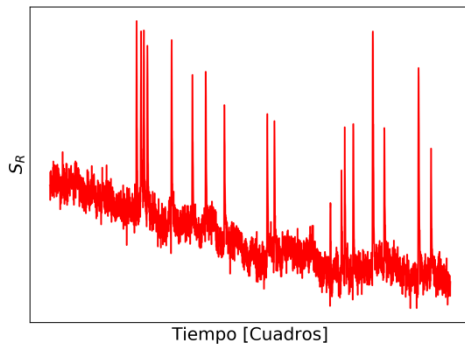
Para validar que una ROI realmente corresponde a una neurona es necesario analizar su señal de cambio de fluorescencia a través del experimento, para ello se requiere volver a cargar de manera secuencial

cada cuadro de cada video del experimento, la señal asociada a cada neurona es llamada transitorio de calcio y es calculada de acuerdo a la descripción del experimento característico de la sección 2.2 y la ecuación Ec. 2.1. Todo este proceso es realizado empleando los videos originales sin ningún procesamiento y los valores definidos por el usuario para la proporción del radio  $R$  del área alrededor de la ROI a considerar respecto al radio  $r$  de la ROI y si serán consideradas las ROIs vecinas.

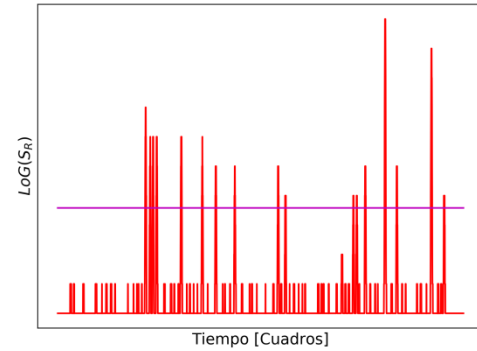
Después de obtener el transitorio de calcio de cada ROI el siguiente paso es determinar los potenciales de acción en ellos, para esto es necesario identificar sobre saltos en la señal en forma de espigas (potenciales de acción), para ello es convolucionada la señal con un filtro LoG y al resultado se le aplica umbralización con umbral de dos desviaciones estándar (Figura 5.11), si hay espigas con magnitud superior al umbral entonces se considera que hubo potenciales de acción y el resultado es guardado en la Matriz de actividad, una matriz binaria donde las filas corresponden a las neuronas validadas y las columnas a cada instante del experimento de forma secuencial, donde 0 representa estado de reposo y 1 actividad neuronal. Finalmente, los datos son reorganizados eliminando las ROIs que no presentaron actividad y se genera el Mapa de neuronas. En Figura 5.12 se muestra el proceso de validación y en la Figura 5.13 un ejemplo de las salidas: Mapa de neuronas y Matriz de actividad, para un Mapa de ROIs de entrada.



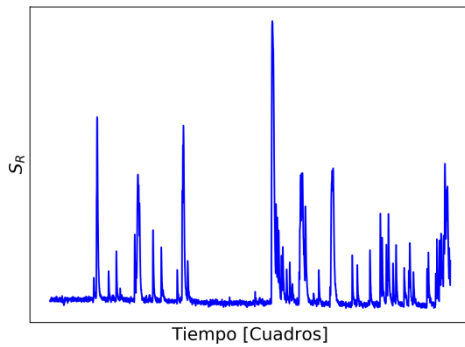
(a)



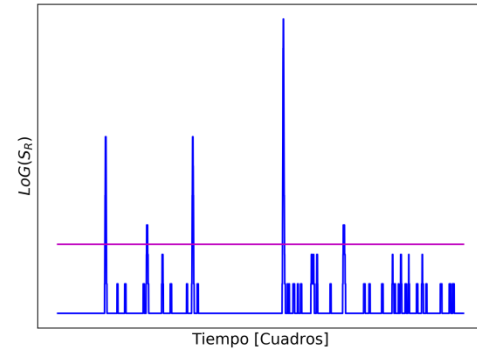
(b)



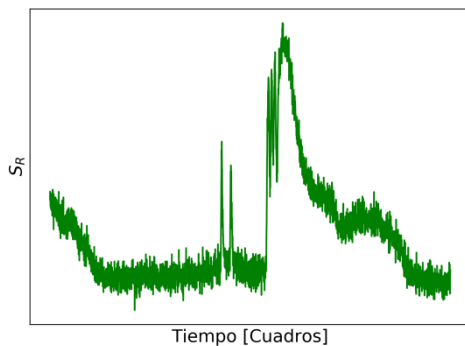
(e)



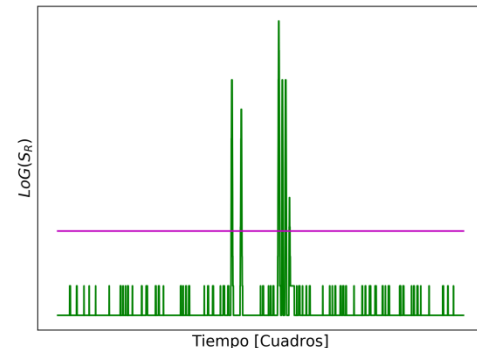
(c)



(f)



(d)



(g)

Figura 5.11. En (a) se muestran tres ROIs en color rojo, azul y verde, (b), (c) y (d) corresponden a sus respectivos transitorios de calcio y (e), (f) y (g) a las espigas detectadas, donde la línea morada representa el umbral de dos desviaciones estándar

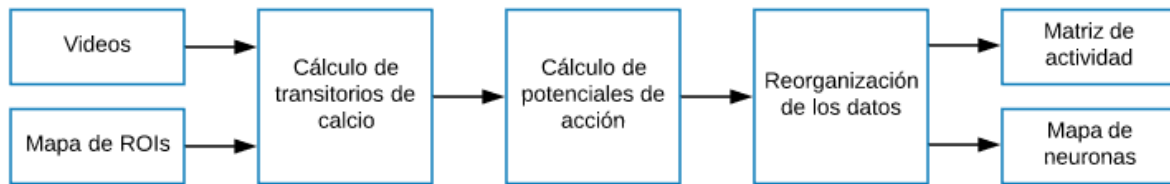


Figura 5.12. Etapa de validación de las ROIs segmentadas

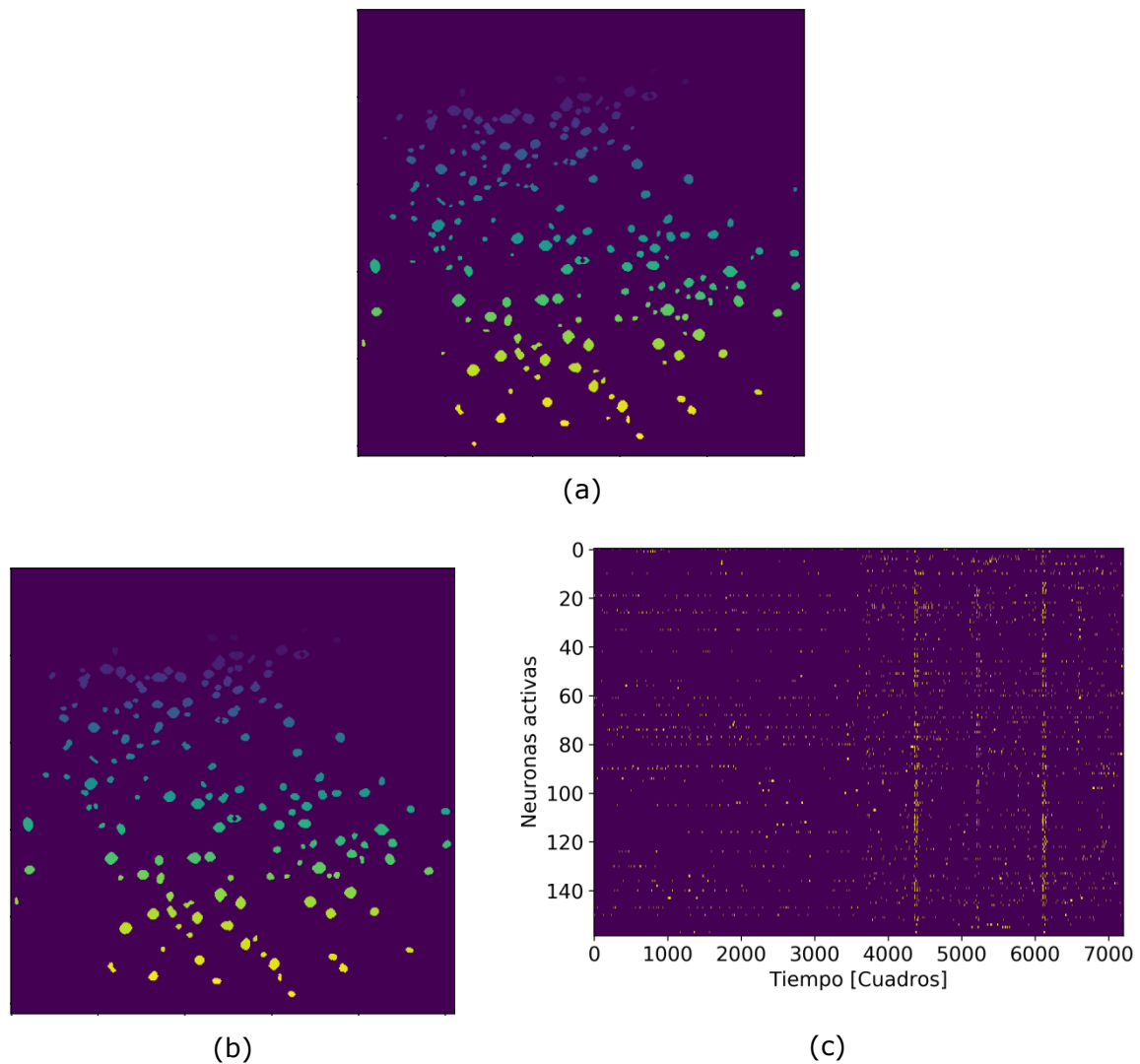


Figura 5.13. En (a) se muestra el Mapa de ROIs, en (b) el Mapa de neuronas y en (c) la Matriz de actividad. Siendo (b) y (c) el resultado final del procesamiento

Aunque SRANME está diseñado para requerir de la menor intervención humana posible para su operación, cuenta con un conjunto de parámetros de fácil configuración que pueden ser modificados por el

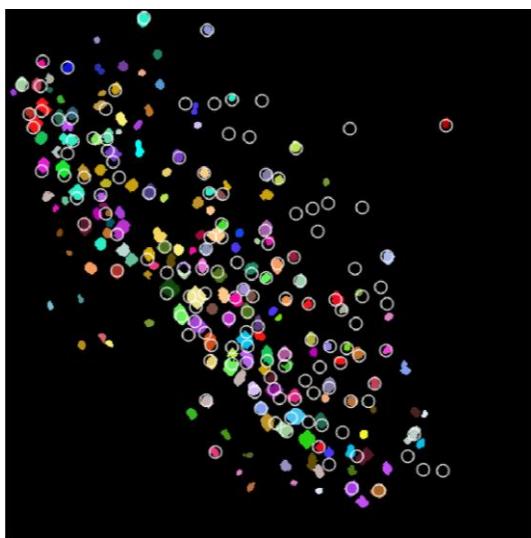
usuario si lo considera conveniente. En la Tabla 5.1 son descritos estos parámetros por etapa y proceso.

Tabla 5.1. Parámetros configurables del sistema

| Etapa        | Proceso   | Parámetro  |
|--------------|---|--|
| Segmentación | Mejora de contraste<br>(Figura 5.3)               | Tamaño de la ventana usada para ecualización local             |
|              | Supresión de ruido<br>(Figura 5.3)                | Orden del filtro Gaussiano empleado para la convolución 3D     |
|              | Detección de blobs<br>(Figura 5.6)                | Orden del filtro Gaussiano para el análisis multiescala        |
|              |   | Número de DOGs por resolución                                  |
|              |   | Número de submuestreos   |
|              | Detección de blobs<br>(Figura 5.7)                | Número de desviaciones estándar empleado para la umbralización |
| Validación   | Cálculo de potenciales de acción<br>(Figura 5.11) | Orden del filtro LoG   |
|              |   | Número de desviaciones estándar empleado para la umbralización |

## 6. Pruebas y resultados

Se realizaron pruebas con 5 experimentos, cada uno consta de 4 a 6 videos y han sido previamente analizado por el experto como cualquier experimento característico descrito en la sección 2.2, por lo que se cuenta con las etiquetas hechas a mano, estas serán en primera instancia el marco de referencia, sin embargo, es necesario considerar el error humano.



Videos: 4

Neuronas identificadas por el experto: 138

Neuronas identificadas por el sistema: 195

Número de coincidencias: 112

Porcentaje de coincidencias: 81.2%

Tiempo de ejecución: 10:41 [mm:ss]

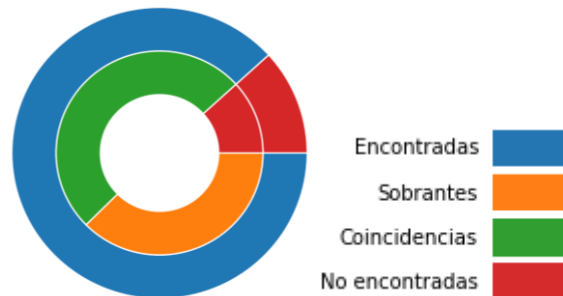
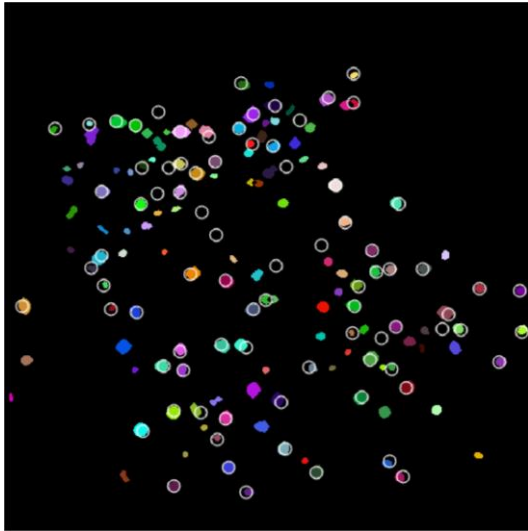


Figura 6.1. Experimento 1



Videos: 4  
 Neuronas identificadas por el experto: 83  
 Neuronas identificadas por el sistema: 140  
 Número de coincidencias: 73  
 Porcentaje de coincidencias: 87.9%  
 Tiempo de ejecución: 09:26 [mm:ss]

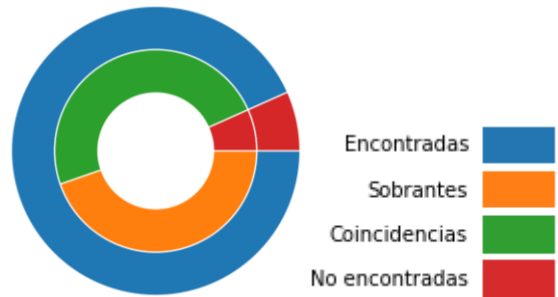
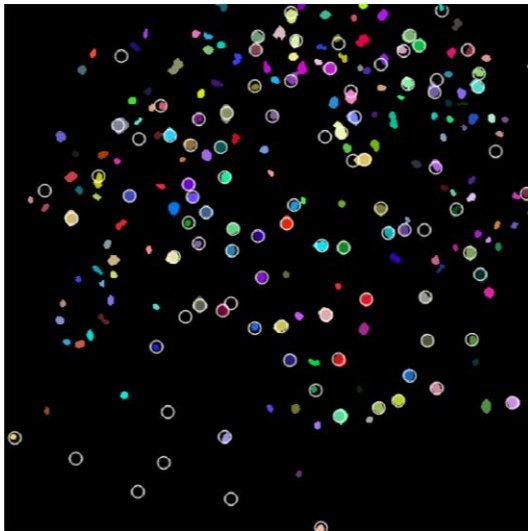


Figura 6.2. Experimento 2



Videos: 5  
 Neuronas identificadas por el experto: 93  
 Neuronas identificadas por el sistema: 202  
 Número de coincidencias: 81  
 Porcentaje de coincidencias: 87.1%  
 Tiempo de ejecución: 12:40 [mm:ss]

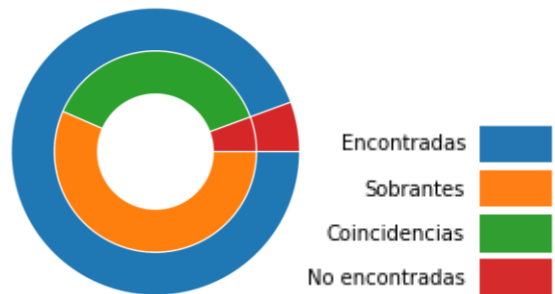
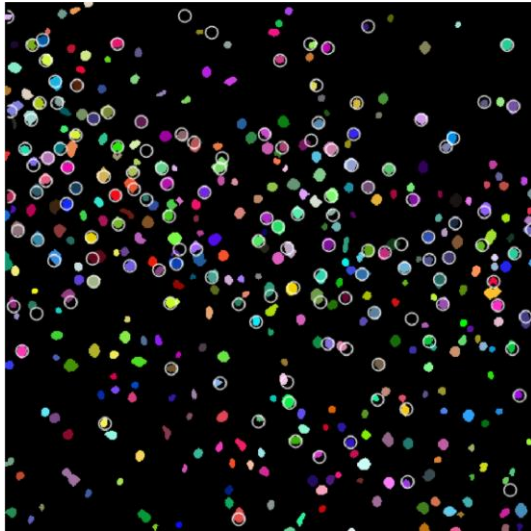


Figura 3.3. Experimento 3





Videos: 6  
 Neuronas identificadas por el experto: 137  
 Neuronas identificadas por el sistema: 403  
 Número de coincidencias: 132  
 Porcentaje de coincidencias: 96.3%  
 Tiempo de ejecución: 16:54 [mm:ss]

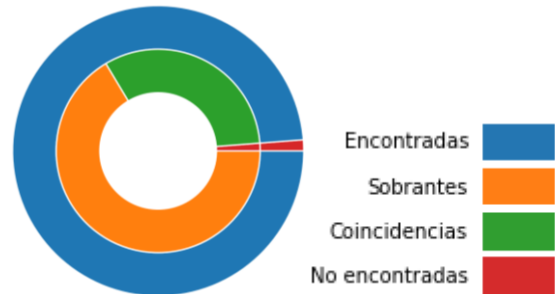
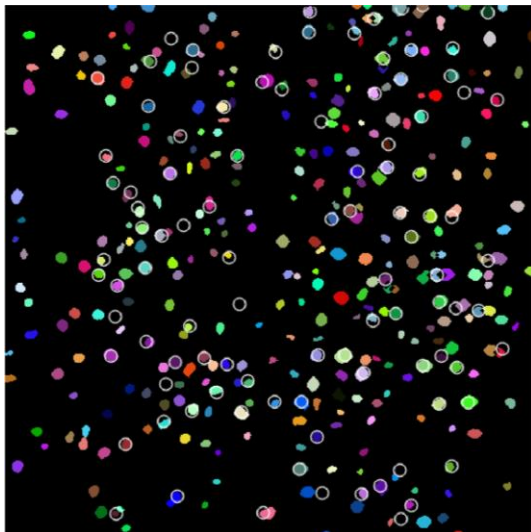


Figura 6.4. Experimento 4



Videos: 5  
 Neuronas identificadas por el experto: 99  
 Neuronas identificadas por el sistema: 348  
 Número de coincidencias: 91  
 Porcentaje de coincidencias: 91.9%  
 Tiempo de ejecución: 14:18 [mm:ss]

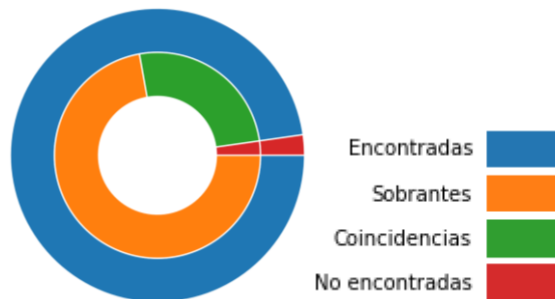


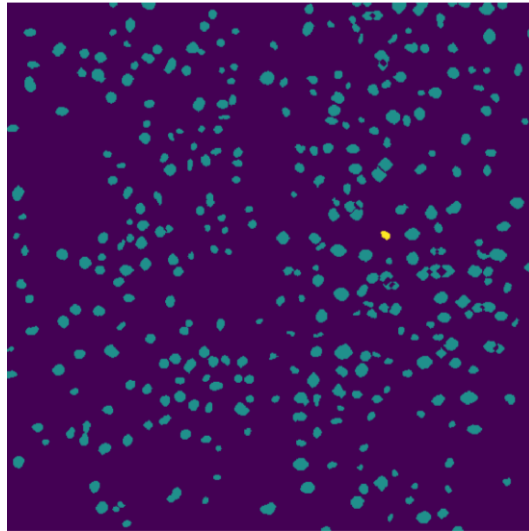
Figura 6.5. Experimento 5

En las gráficas de los experimentos se muestra en color azul todas las neuronas identificadas por el sistema, en rojo aquellas neuronas que el experto identificó pero el sistema no, siendo esta considerablemente más pequeña que la región verde, correspondiente a aquellas que fueron

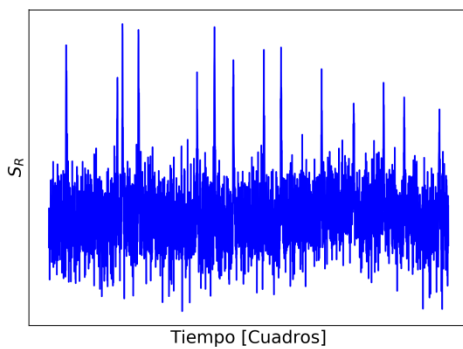
identificadas tanto por el experto como por el sistema. En tanto que la región de color naranja corresponde a neuronas identificadas por el sistema pero no el experto. En la Figura 6.6 se muestra un ejemplo de estas neuronas que el experto no identificó pero el sistema sí.

Las neuronas identificadas por el sistema pero no por el experto requieren una particular atención pues podrían marcar una clara frontera de las limitaciones humanas en esta actividad, es decir, que podría significar que el sistema tiene la capacidad de identificar neuronas que pasan desapercibidas al ojo humano. Los mismos expertos reconocen que el error humano en esta actividad puede llegar a ser considerable por factores como la experiencia, la capacidad visual y el cansancio. Por lo tanto, el hecho de que el sistema haya detectado un mayor número de neuronas en los videos no es una sorpresa para ellos, al contrario, representa una ventaja importante tener una herramienta automática, que además de identificar la mayoría de las neuronas que ellos pueden observar, también pueda identificar aquellas que pasaron desapercibidas para ellos.

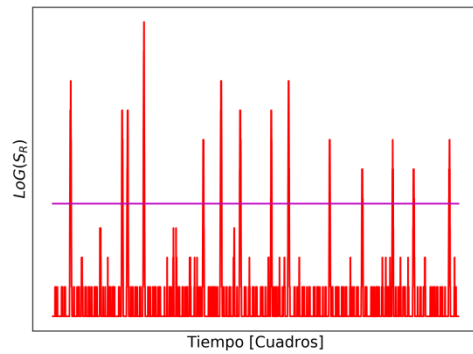
El fenómeno de que "el sistema ve más que el experto" también se puede apreciar en los casos donde la ROIs marcada por el experto parece solo cubrir una pequeña parte de la neurona identificada por el sistema, esto se puede interpretar como que el experto solo fue capaz de observar una pequeña parte de la neurona y esa fue la que etiquetó, esto durante la etapa del cálculo de los transitorios de calcio puede generar problemas debido a que solo se contempla una parte de la neurona y la otra es considerada parte de la intensidad del contexto.



(a)



(b)



(c)

Figura 6.6. En (a) se muestra, sobresaliendo en amarillo, una neurona del Experimento 5 que no fue etiquetada por el experto, en (b) su correspondiente transitorio de calcio y en (c) el resultado de aplicarle el operador LoG. Este es un ejemplo de una ROI que presenta actividad neuronal y que pasó desapercibida por el experto

Respecto a los requerimientos de hardware, se empleó una computadora con CPU Core i3-2100, memoria de RAM de 4 GB y sistema operativo Windows 10, esta configuración representa el estándar de prestaciones en equipos de gama baja-media a la fecha del desarrollo. En cuanto al tiempo de ejecución, toma en promedio 02:50 [mm:ss] por video llegar a la Matriz de Actividad y al Mapa de neuronas, siendo en todos los casos menor al tiempo destinado por el experto a hacer el mismo análisis.

## 7. Conclusiones

Durante el desarrollo de esta tesis se habló de la importancia de las investigaciones en neurociencias y de cómo han sido impulsadas por el avance de la tecnología. Se definieron los problemas que enfrentan los investigadores del Departamento de Neurociencia Cognitiva del IFC para la identificación manual de neuronas en videos que obtienen mediante microscopía de epifluorescencia. Con el fin de apoyarlos en tal tarea se desarrolló SRANME, un sistema basado en procesamiento de imágenes digitales. Con este sistema se lograron resultados satisfactorios en el sentido de que en menos de 20 minutos y sin la necesidad de altas prestaciones de computo, es capaz de reconocer arriba del 80% de las neuronas identificadas por el experto e incluso aquellas que fueron omitidas por él, siendo reconocido que esto representa una importante mejora en el desempeño de sus actividades.

Por lo visto en los resultados, los objetivos fueron alcanzados, pues se obtuvo un desempeño promedio superior al 80% identificado las neuronas etiquetadas manualmente y además el sistema fue capaz de identificar otras que pasaron desapercibidas para los expertos, lo que supone una ventaja importante para ellos. Los datos de salida del sistema, la Matriz de actividad y el Mapa de neuronas proporcionan más información significativa que la que se obtenía mediante un análisis manual, tienen un formato adecuado para su interpretación y posteriores análisis.

Los resultados de desempeño obtenidos fueron alcanzados sin complejas implementaciones de hardware, incluso queda demostrado que con prestaciones estándar para equipos de gama baja-media se puede realizar todo el proceso en menor tiempo que el invertido por el experto, lo que supone un importante ahorro de tiempo y con ello mayor productividad.

Las tecnologías empleadas para el desarrollo del software son muy accesibles y de código abierto por lo que no son requeridas licencias ni algún tipo de software especializado, de modo que la aceptación y adaptación al sistema no debería suponer un problema. Además el sistema está desarrollado en forma de módulos que el usuario pueda usar de forma directa y amigable, aunque la intención fue la de reducir la cantidad de parámetros configurables por el usuario, si se desea, estos pueden ser rápidamente modificados e incluso la implementación permite su modificación fácilmente, esto con la intención de poder darle mantenimiento al sistema, realizar modificaciones de acuerdo a las necesidades de los usuarios y continuar desarrollando nuevos módulos.

La capacidad del sistema de continuar creciendo a través de nuevos desarrollos fue prevista bajo la premisa de que aún hay mucho trabajo por hacer, pues esta implementación brinda una herramienta de apoyo a los investigadores del IFC para una de las muchas tareas que desempeñan, en ese sentido es necesario resaltar que no se busca desplazar al experto de tareas que le son propias, se busca proporcionar herramientas automatizadas que le permita a los expertos ser más productivos en sus actividades y con ellos acelerar sus investigaciones.

SRANME ofrece un enfoque más robusto que las implementaciones desarrolladas en los trabajos relacionados, pues se han considerado en la mejorar de la imagen cuadros consecutivos y convolución en tres dimensiones para suprimir el ruido que representa uno de los principales problemas en este tipo de videos. La baja resolución del video hace imposible el uso de transformaciones *watershed* como lo hace S. Banglieto [5] por lo que se ha propuesto otro método para la división de *blobs* traslapados mediante iteraciones controladas de erosión y dilatación. Finalmente, el análisis de los transitorios de calcio para eliminar aquellas ROIs que no presentan actividad neuronal, así como la Matriz de actividad ofrece datos de mayor fiabilidad.

Entre los posibles trabajos futuros que pueden surgir de esta tesis se encuentran mejoras para su aplicación en videos donde el tejido observado presenta movimiento, la implementación de algoritmos de aprendizaje automático para agrupar aquellas neuronas con actividad similar en grupos funcionales y algoritmos para analizar la Matriz de actividad con base en la teoría de redes. Incluso el Mapa de neuronas podría ser usado para entrenar arquitecturas de aprendizaje profundo como *U-Net* [26] y aprovechado para trazar la red de los grupos funcionales determinados y con ello aproximar la topología de las redes neuronales.

## 8. Referencias

- [1] M. F. Bear, B. W. Connors and M. A. Paradiso, *Neurosciences: Exploring the Brain*. 4th Edition. China: Wolters Kluwer, 2016.
- [2] A. J. H. Newton, R. A. McDougal, M. L. Hines and W. W. Lytton, "Using NEURON for Reaction-Diffusion Modeling of Extracellular Dynamics", *Frontiers in Neuroinformatics* 12, 2018.
- [3] J. Sekulska, S. Jezewski, R. Ambroziak, M. Kuzan'ski, "Digital image processing methods in biological structure recognition - a short review", *2nd International Conference on Perspective Technologies and Methods in MEMS Design*, pp. 61-64, 2006.
- [4] A. Procházka and O. Vyšata, "History and biomedical applications of digital signal and image processing", in *International Workshop on Computational Intelligence for Multimedia Understanding (IWCIM), Paris*, pp. 1-5, 2014.
- [5] S. Baglietto, I. E. Kepiro, G. Hilgen, E. Sernagor, V. Murino and D. Sona, "Automatic Segmentation of Neurons from Fluorescent Microscopy Imaging", *BIOESTEC, CCIS 881*, pp. 121-133, 2018.
- [6] J. Chen, S. Banerjee, A. Grama, W. J. Scheirer and D. Z. Chen, "Neuron Segmentation Using Deep Complete Bipartite Networks", *MICCAI 2017, Part II, LNCS 10434*, pp. 21-29, 2017.
- [7] M. M. Dorostkar, E. Dreosti, B. Odermatt and L. Lagnado, "Computational processing of optical measurements of neuronal and synaptic activity in networks". *Journal of Neuroscience Methods* 188, pp. 141-150, 2010.
- [8] G. M. Shepherd, *The Synaptic Organization of the Brain*. 5th Edition. New York: Oxford University Press, 2004.

- [9] J. Pérez, M. Duhne, E. Lara, V. Plata, D. Gasca, E. Galarraga, A. Hernandez and J. Bargas, "Pathophysiological signatures of functional connectomics in parkinsonian and dyskinetic striatal microcircuit", *Neurobiology of Disease*, 91, pp. 347-61, 2016.
- [10] M.Serrano et al, "Epileptogénesis experimental: estudios in vitro en un modelo animal", *Arch Neurocién (Mex) INNN*, 2017.
- [11] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. A. Siegelbaum and A. J. Hudspeth, *Principles of Neural Science*. 5th Edition. United States: Mc Graw Hill, 2013.
- [12] P. Dayan and L. F. Abbott, *Theoretical Neuroscience*. 1st Edition. London: The MIT Press, 2005.
- [13] T. W. Chen, T. J. Wardill, Y. Sun, S. R. Pulver, S. L. Renninger, A. Baohan, E. R. Schreiter, R. A. Kerr, M. B. Orger, V. Jayaraman, L. L. Looger, K. Svoboda and D. S. Kim, "Ultrasensitive fluorescent proteins for imaging neuronal activity", *Nature volume 499*, pp. 295–300, 2013.
- [14] T. Badea, J. Goldberg, B. Mao and R. Yuste, "Calcium Imaging of Epileptiform Events with Single-Cell Resolution", *Journal of Neurobiology* 48, pp. 215-227. 2001.
- [15] R. Cossart, Y. Ikegaya and R. Yuste, "Calcium imaging of cortical networks dynamics". *Cell Calcium* 37, pp. 451–457, 2005.
- [16] L. Carrillo, W. Yang, J. K. Miller, D. S. Peterka and R. Yuste, "Imaging and Optically Manipulating Neuronal Ensembles", *Annual Review of Biophysics* 46, pp. 271–293, 2017.
- [17] C. Grienberger and A. Konnerth, "Imaging Calcium in Neurons", *Neuron* 73, pp. 862-885, 2012.



- [18] J. E. Pérez, "Características fisiopatológicas de la enfermedad de Parkinson y la discinesia en el conectoma funcional del microcircuito estriatal". Tesis doctoral, Departamento de Neurociencia Cognitiva, Instituto de Fisiología Celular de la UNAM. Ciudad Universitaria, CDMX, 2016.
- [19] J. Broeke, J. M. Mateos and J. Pascau, *Image Processing with ImageJ*. 2nd Edition. Birmingham: Packt Publishing, 2015.
- [20] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. 4th Edition. Pearson, 2017.
- [21] R. Szeliski, *Computer Vision: Algorithms and Applications*. 1st Edition. Springer, 2010.
- [22] M. S. Nixon and A. S. Aguado, *Feature Extraction and Image Processing*. 1st Edition. Great Britain: Newnes, 2002.
- [23] J. C. Russ, *The Image Processing Handbook*. 5th Edition. Taylor & Francis Group, 2007.
- [24] M. Petrou and C. Petrou, *Image Processing: The Fundamentals*. 2nd Edition. Singapore: John Wiley & Sons, 2010.
- [25] W. Burger and M. J. Burge, *Principles of Digital Image Processing: Core Algorithms*. 1st Edition, London: Springer, 2009.
- [26] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation". *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, pp. 234–241, 2015.