



UDEM

Biblioteca

12 JUN 2012

Recepción de Tesis

13
2,500

Universidad de Monterrey



INGENIERO BIOMÉDICO

PROYECTO DE EVALUACIÓN FINAL

**“AUTOMATIZACIÓN DE ANÁLISIS DE IMÁGENES
FLUORESCENTES DE CÉLULAS CANCERÍGENAS”**

AUTOR: DAVID ALBERTO CARAVEO GARAY

ASESOR:

ING. JORGE TAKENAGA FUKUSHIMA

040.6102B
C262a
2011

SAN PEDRO GARZA GARCÍA, N.L

MAYO DE 2011

BIBLIOTECA
UNIVERSIDAD DE MONTERREY

San Pedro Garza García, 4 de Mayo de 2012

Por medio de la presente comunico a usted que he revisado el documento académico final cuyo título es *Automatización de análisis de imágenes fluorescentes de células cancerígenas* respecto a su forma y contenido, el cual es presentado por el alumno del Programa de Evaluación Final *David Alberto Caraveo garay* de la Universidad de Monterrey. Considero que el documento académico final tiene las características de ejecución y calidad que la Universidad de Monterrey demanda, además cumple con los objetivos y alcance planteada en el anteproyecto autorizado por el Comité de PEF al inicio del semestre.

Cordialmente,



M. en C. Jorge Takenaga Fukushima
Departamento de Ingeniería

c.c.p. Dr. Demófilo Maldonado Cortés, Director del Departamento de Ingeniería.

c.c.p. David Alberto Caraveo Garay

Agradecimientos

A Dios, por darme las bendiciones de salud, intelecto, y por otorgarme una familia que se preocupa por mí. Por estar ahí cuando más lo he necesitado y por guiarme correctamente en las decisiones que he realizado.

Agradezco principalmente a mis padres, Rey David Caraveo Bojórquez y a Carmen Belem Garay, por darme los valores que me han guiado en mi vida, por darme un buen ejemplo y enviarme a estudiar mi carrera que concluyó con este trabajo.

A mi hermana Silvina Caraveo Garay, por darme consejos y los ánimos para seguir adelante. Y a Alejandra Caraveo Garay por tenerme paciencia. A mi familia, en general por tener confianza en mí y apoyarme constantemente.

A mis compañeros de carrera; por las carnes asadas que nos unió más como amigos. Gracias a ellos pude aprender y realizarme como un buen ingeniero biomédico.

A mi nueva familia, que son mis amigos que hice en mi estancia en Monterrey: Mónica, Cecilia, Eduardo, Karla, Juan, Siria, Sergio, Luna, Pepe, Carito, Lety, Alex, Elizabeth, Gaby, Caty, Carlos (Tabas), Alán, Yuzakk, Mares, Eloina y a Gary. Mis amigos internacionales y nacionales, aunque no estuvieron toda mi carrera sé que siempre estarán ahí apoyándome. Gracias por su amistad y apoyo, saben que cuentan conmigo para siempre.

A mis grupos estudiantiles, por ellos mejore mis cualidades de liderazgo y de relaciones personales: el grupo SOMIB-UDEM, SIFE-UDEM, EWH-UDEM y a I-LINK

Y por último, a la Universidad de Monterrey, a sus profesores que me transmitieron sus conocimientos.

Índice

1	Introducción.....	1
2	Descripción del problema.....	2
3	Marco Teórico.....	5
3.1	Procesamientos de imágenes en Ingeniería Biomédica.....	5
3.2	Fluorescencia.....	5
3.2.1	Microscopio fluorescente.....	8
4	Procesamiento de imágenes.....	11
4.1	Transformaciones de intensidad y filtros espaciales.....	11
4.1.1	Funciones de transformaciones de intensidad básicas.....	11
4.2	Procesamiento de histogramas.....	12
4.2.1	Ecualización de histogramas.....	13
4.3	Filtrados Espaciales.....	13
4.3.1	Filtros espaciales de alisado.....	13
4.4	Segmentación de Imágenes.....	14
4.4.1	Método de Canny.....	15
4.4.2	Segmentación basada en umbrales.....	17
4.5	Morfología de la imagen.....	18
4.5.1	Etiquetado de imagen.....	20
4.6	Región de interés (ROI).....	21
4.7	Agrupamiento de K-medio.....	22
5	Metodología.....	24
6	Resultados:.....	39
7	Conclusiones:.....	41
8	Bibliografía.....	42

9	Anexo	45
9.1	Programa de conteo de imágenes	45
9.2	Programa ROI.....	49
9.3	Programa K-means.....	50
9.4	Programa: Detección de contorno Canny	53
9.5	Programa: Filtro Gaussiano	55
9.6	Programa: Derivadas	57

Índice de Figuras

Figura 1: Imagen usada por el patólogo.....	3
Figura 2: Diagrama de Jablonski (Lakowicz, 2006.).....	6
Figura 3: Imagen con los marcadores DAPI, Tedrametilrodamina (Lakowicz, 2006.)	7
Figura 4: Tejido celular obtenido con el microscopio fluorescente.....	10
Figura 5: Unión de nivel de grises en un histograma	18
Figura 6: Dilatación en los granos de arroz [a, b] en el lado derecho se aplicó erosión a un circuito integrado [c d] (González R & Woods, R, 2007)	20
Figura 7: A) Tejido celular renal B) Tejido celular renal con una ROI (Harandi, N, S Sadri, and N Moghaddam, 2009).....	22
Figura 8: El uso de k-medio en un imagen cerebral tomada de una resonancia magnética (Ilea, Dana E. and Whelan, Paul F., 2006)	23
Figura 9: Flujo grama del procesamiento de imágenes (Harandi, N, S Sadri, and N Moghaddam., 2009).....	24
Figura 10: Longitud de onda de los marcadores de Alexa (Life Technologies 2012)	25
Figura 11: Filtro para el marcador Alexa 488 (Zeiss, 2012)	26
Figura 12: Filtro para el marcador Alexa 588 (Zeiss, 2012)	27
Figura 13: Figura con los LUTs ya procesada	28
Figura 14: Pantalla para abrir una imagen en Matlab	29
Figura 15: Selección de áreas en la imagen.	30
Figura 16: Imagen con ROI seleccionada	31
Figura 17: Histograma del canal G de la imagen a procesar.	32
Figura 18: Valor en escala de gris en el contorno de una célula.....	33
Figura 19: Imagen con el detector de contornos de Canny.....	34
Figura 20: Filtro Gaussiano aplicado a la imagen con Canny.	35
Figura 21: Imagen que se le aplico el bwareaopen.....	35
Figura 22: Imagen que se le aplico una función morfológica de erosión.....	36
Figura 23: Plano X Y de las células.	37

Figura 24: Imagen con las células con cáncer encima de los ácidos nucleicos	37
Figura 25: Despliegue de la información al final el proceso de imagen.	38

1 Introducción.

El procesamiento de imágenes se ha desarrollado altamente en los últimos años para la optimización de las observaciones y diagnósticos realizados a través de imágenes médicas del cuerpo humano. Los algoritmos empleados en el procesamiento de imágenes tienen el objetivo de mejorar las características de las imágenes tales como intensidad, nitidez, etc., así como ampliar la obtención de información mediante las reconstrucciones en 3-D, identificación de contornos de interés, cuantificación de componentes biológicos, etc. Por otro lado, dichos esquemas no se han explotado en su totalidad en el proceso sobre conteo de células, por lo que actualmente no se encuentra disponible una herramienta de software orientada al conteo celular en el mercado actual.

Por otro lado, existe una gran variedad de técnicas para realizar el conteo celular en imágenes que permiten obtener información útil ya sea para diagnóstico o investigación. Dichas técnicas implican la detección de contornos y las operaciones morfológicas, las cuales ayudan a la obtención y manipulación de la forma básica de la imagen.

Por lo anterior, este trabajo propone la aplicación de un procesamiento de imágenes, en el lenguaje de Matlab como herramienta de software, para resaltar las características de las células cancerígenas presentes en las imágenes médicas. Se realizará un programa para automatizar el conteo de las células cancerígenas, y así contribuir a la emisión de diagnósticos más certeros y en menor tiempo.

2 Descripción del problema.

En la actualidad, no existe un método exacto para encontrar un porcentaje de probabilidad en que un tejido esté infectado por cáncer, ya que los patólogos utilizan su experiencia y agudeza visual para diagnosticar a un paciente que potencialmente sufre de esta enfermedad.

El patólogo especialista es el encargado de hacer el análisis de tejidos para generar un diagnóstico que vaya de acuerdo a la enfermedad del paciente. Lo que ellos realizan al evaluar las imágenes es obtener la cantidad aproximada del área que existe en el tejido a observar. En seguida, el patólogo vuelve a hacer un conteo sobre las células infectadas de cáncer. Al tener ya los dos números aproximados, el patólogo está usando un método cuantitativo, el cual no es reproducible por el hecho de estar usando su agudeza visual calculando densidad de cáncer en el tejido. El sentido de la vista puede llegar a tener limitaciones en la observación continua de distintas imágenes en un corto período de tiempo, por lo que es deseable tener un sistema automatizado que pueda hacer un trabajo reproducible. En la Figura 1 se puede observar la imagen que analiza el patólogo para obtener el conteo de células con el programa ImageJ.

El programa ImageJ es un software con la licencia gratuita distribuido por el Instituto Nacional de Salud (NIH, por sus siglas en inglés). El programa puede contar con las características que del usuario de forma manual, tales como el nivel de intensidad del umbral, que detector de contorno usar, etc., No obstante, el patólogo utiliza el ImageJ pero al obtener los datos de tres tejidos se puede tardar más de un día.

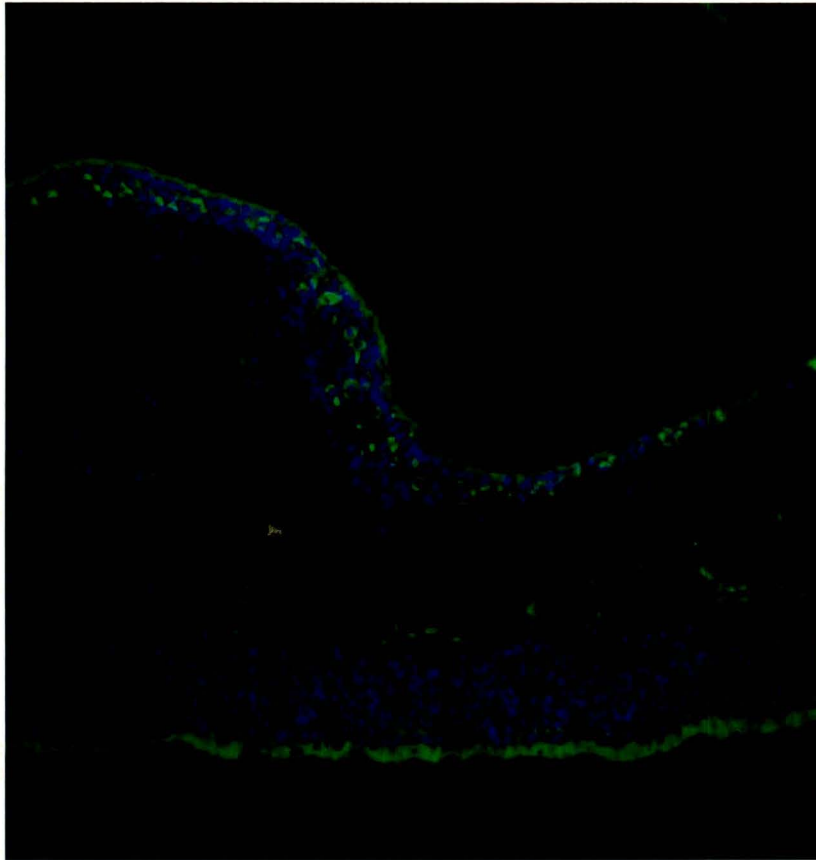


Figura 1: Imagen usada por el patólogo.

Teniendo como referencia imágenes de células no infectadas e infectadas de cáncer, los especialistas comparan las imágenes a través del método de cruces, donde pueden diagnosticar si un paciente padece de cáncer. Dicho método consiste en que el especialista compare las imágenes base con la del paciente, para que un tejido tenga una cruz o más es a partir de la comparación que hace el patólogo, este método puede variar según el patólogo, del tejido a analizar contra el tejido prueba. Al obtener una imagen de un tejido que no marque ninguna cruz, al ser comparada con la referencia, ésta se traduce en que el paciente no es positivo en el padecimiento del cáncer. Sin embargo, de obtenerse cuatro cruces, se revelaría que un paciente tiene una gran probabilidad de padecer cáncer.

El problema radica en que cada patólogo especializado puede realizar un diagnóstico diferente al mismo paciente, incluso pudiendo hacer variaciones en un solo caso; esto se debe a que la agudeza visual que tenía al hacer el primer diagnóstico puede que sea distinta al realizar la segunda evaluación. Teniendo

estas circunstancias en cuenta, comprendemos que al paciente se le puede dar un tratamiento erróneo, ocasionando un retraso en hallar el correcto, o en el peor de los casos, la muerte del paciente.

Existe un método llamado «selección de contorno» que consiste en obtener el contorno de las células de las imágenes a través de procesamientos de imágenes. Pero más allá de esto, se debe de tomar en cuenta el contraste y la distribución de la escala de grises de la imagen para realizar dicha acción. Para ello existen programas, como el ImageJ, que pueden facilitar el manejo de estas imágenes, haciendo máscaras automáticas alrededor de lo que éstos interpretan como una célula. No obstante, es común que estos programas arrojen conteos de células que no existen, o que no tomen en cuenta a las células que sí deberían de analizar. Además, debido a la interacción del patólogo con el programa para la aplicación y selección de parámetros de procesamiento, es posible la aplicación de un procesamiento deficiente de las imágenes.

Emplear el método de la “selección de contorno” de forma manual puede resultar desgastante y no tan exacto. Lo primero que se realiza es descomponer la imagen en los canales RGB (en español, los colores Rojo, Verde, Azul). Posteriormente, se obtiene una imagen de ocho bits, que sería una escala de grises. En base a dos de los canales, el patólogo a prueba y error, manipula los parámetros para la obtención de las características que funcionan en uno de los canales analizados, para posteriormente aplicarlos como base inicial al canal restante. La desventaja principal radica en el tiempo excesivo que requiere emplearse para analizar cada canal de forma individual, así como la tendencia natural del analista a sesgar los resultados tras asumir cantidades de estudios previos de dicha imagen.

3 Marco Teórico

3.1 Procesamientos de imágenes en Ingeniería Biomédica

El uso de las imágenes médicas, es decir, imágenes de las estructuras y morfología interna del cuerpo humano, comenzó en 1885, cuando los Rayos X fueron descubiertos por Roentgen, lo que trajo consigo la innovación en la medicina al proveer una herramienta para observar el interior del cuerpo humano sin ninguna cirugía invasiva [1]. La expresión «imagen médica» hace referencia a la obtención de datos relativos a la condición fisiológica o patológica basada en la interpretación de la imagen que representa una sección del cuerpo. [8]

El uso de la computadora para el diagnóstico de imágenes médicas también ha sido altamente estudiado en varios aspectos incluyendo generación, realces, mediciones, análisis y recuperación de imágenes. Cabe mencionar que procesamiento de imágenes médicas ha sido una de las áreas más importante desde que comenzó la era de procesamiento digital de imágenes. [1]

Una aplicación esperada de los estudios computarizados de imágenes de Rayos X es el diagnóstico automatizado. Hoy en día, el uso de “Diagnósticos Asistidos por Computadora” (CAD, por sus siglas en inglés) se ha hecho popular por estos tipos de procesamientos de imágenes médicas. [1][6]

El presente proyecto se enfocará en la adquisición de imágenes por medio de representaciones de microscopio fluorescente y algoritmos de reconstrucción.

3.2 Fluorescencia

Fluorescencia es ahora la metodología dominante usada extensivamente en biotecnología, flujo cito-métrico y diagnósticos médicos. Las imágenes fluorescentes pueden revelar la localización y medidas de moléculas intracelulares, a veces en el nivel de detección de una sola molécula.

Durante los pasados años los avances de la tecnología han proveído nuevos usos de fluorescencia. Estas tecnologías han sido adoptadas rápidamente y se han convertido en los métodos principales. Algunas de esas tecnologías son: dos-fotones o excitación multi-fotónica y microscopía confocal. [13]

La fluorescencia es usualmente excitada por la absorción de un simple fotón con una longitud de onda dentro la banda de absorción del fluoróforo. Los láseres con pulso de femtosegundos pueden excitar un fluoróforo por absorción. Dicha tecnología es reconocida por su facilidad de uso y compatibilidad con distintos microscopios. [13] Si la intensidad del láser es alta, un fluoróforo puede absorber simultáneamente de longitud de onda larga para alcanzar el siguiente estado cuántico; proceso que depende fuertemente en la intensidad de la luz y ocurre sólo en el punto focal del rayo del láser. Esto puede ser visto en el fotón, donde la emisión se produce sólo a partir de un solo punto dentro de la muestra. [13] [17]

Cuando una molécula fluorescente absorbe un fotón en la longitud de onda adecuada, un electrón es excitado a un estado de energía mayor y casi inmediatamente se colapsa y se regresa a su estado basal. En el proceso del colapso de energía la molécula puede liberar la energía absorbida como un fotón fluorescente. Desde que la energía es perdida en el proceso, la fluorescencia emitida del fotón típicamente exhibe una frecuencia menor de vibración y una longitud de onda más larga que el fotón excitado que él fue absorbido. [17]

La situación está representada gráficamente en el diagrama Jablonski en la Figura 2, la cual muestra una serie de estados de energía creciente como pila de líneas horizontales, representando la relajación individual al llegar al estado basal. k_{nr} es la tasa de decaimiento no radiactiva a S_0 (Estado basal), Γ es la tasa de emisión del fluoróforo.

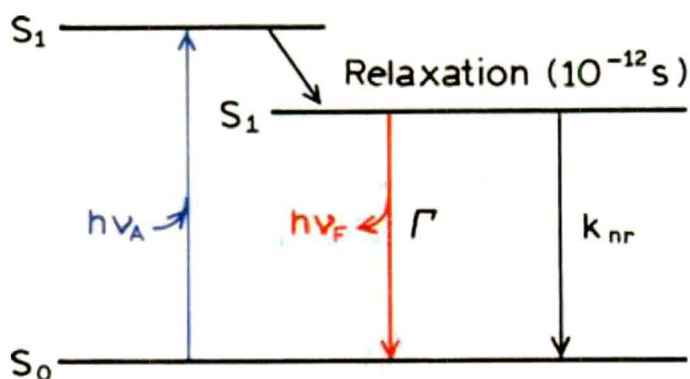


Figura 2: Diagrama de Jablonski (Lakowicz, 2006.)

En la excitación localizada de dos fotones se ha encontrado un uso extendido en microscopía de fluorescencia. Las excitaciones de múltiples fotones permiten imágenes de un solo plano focal de un microscopio, lo cual es una ventaja pues las imágenes de fluorescencia de otro modo distorsionarían la fluorescencia desde arriba y por debajo del plano focal.

La Figura 3, que se presenta a continuación, muestra un resultado obtenido a través de excitación multi-fotónica. Las células son etiquetadas con tres marcadores:

- DAPI para los ácidos nucleicos (4',6-diamidino-2-fenylindole), indicado con el color azul.
- Marcador Patman para membranas visualizado con el color verde
- Tedrametilrodamina para mitocondrias, que se muestra con el color rojo.

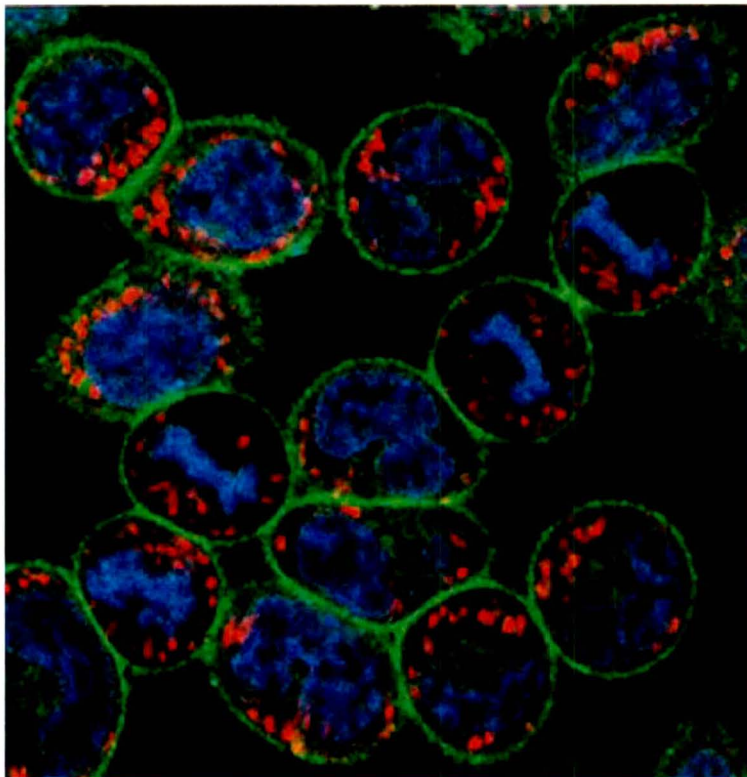


Figura 3: Imagen con los marcadores DAPI, Tedrametilrodamina (Lakowicz, 2006.)

Estas imágenes obtenidas con el microscopio confocal, son nítidas porque no hay una fase actual de fluorescencia que disminuya la resolución de la imagen. Dichas imágenes fueron obtenidas de varios laboratorios para su diagnóstico.

Actualmente, la tecnología basada en fluorescencia está siendo perfeccionada, lo cual ha permitido la obtención de imágenes celulares y detección de moléculas simples. Estos avances en la tecnología de fluorescencia están disminuyendo los costos y la complejidad de instrumentos previos. [13]

Hasta 1990 la espectroscopia multi-fotónica era considerada como un fenómeno exótico, usado principalmente en física, química y espectroscopia óptica. La absorción de dos fotones o la excitación requiere altos picos de energía para incrementar la probabilidad de que dos fotones sean simultáneamente viables para la absorción. [17]

Experimentos con multi-fotones requieren láseres complejos y potencia óptica alta, por eso no es posible el uso de excitación multi-fotónica (MPE, por sus siglas en inglés Multiphoton Excitation) en microscopía óptica, porque el uso de potencia alta podría dañar las muestras biológicas. Sorprendentemente, MPE es ampliamente usada en la microscopía de fluorescencia.

El uso de la microscopía multi-fotónica (MPM por sus siglas en inglés Multiphoton Microscopy) es posible debido a las propiedades favorables de los láseres de titanio-zafiro y el desarrollo del microscopio de escaneo-láser. MPM fue introducido en 1994 y ahora es usado extensamente en imágenes celulares.

3.2.1 Microscopio fluorescente.

Con la microscopía de luz ajustada para la fluorescencia, es posible examinar la distribución de una especie molecular simple en un espécimen. La microscopía fluorescente permite la visualización de moléculas específicas que fluorescen en presencia de luz excitadora. [13] [17]

La microscopía fluorescente contiene filtros especiales y emplean un método único de iluminación que produce imágenes fluorescentes emitidas desde las moléculas excitadas del espécimen. Los filtros son diseñados para aislar y

manipular los juegos de emisión y excitación, así como la longitud de las ondas fluorescentes. Una banda de longitudes de onda de excitación cortas es dirigida desde el iluminador y filtros hacia el espécimen, formando de esta manera una representación del plano de la imagen. [17]

Es importante notar algunas de las aplicaciones de la microscopia fluorescente, estas incluyen:

- *Determinación de las distribuciones intracelulares de las macromoléculas, formando estructuras como membranas, filamentos citosqueléticos y cromatina.* Fluorocromos conjugados y proteínas pueden ser usados para etiquetar los canales de la membrana y los canales iónicos.
- *Estudio de las dinámicas intracelulares de macromoléculas asociado con los procesos de unión de disociación y la difusión (FRAP de sus siglas en inglés Fluorescence Recovery After Photobleaching)*
- *Estudio de las proteínas mediante los estados de interacción, y la reacción de los mecanismos de transferencia de energía fluorescente.*
- *Determinación de las concentraciones intracelulares de iones.*

3.2.1.1 Filtros

Los juegos de filtros fluorescentes contienen tres filtros esenciales: filtro de excitación, espejo dicróico y filtro de barrera (o de emisión). El primer componente, el filtro de excitación transmite selectivamente una banda de longitudes de onda corta para excitar a un fluoróforo del espécimen. [17]

El segundo componente, el espejo dicróico refleja una longitud de onda corta hacia la lente del objetivo y la muestra, mientras que se transmite la luz fluorescente de longitud de onda larga hacia el detector. El tercer componente es la emisión o el filtro barrera, que transmite la banda de fluorescencia mientras bloquea cualquier excitación residual de longitud de onda corta.

Las longitudes de onda fluorescentes forman una imagen en el ojo del observador o en la cámara. Uno de los ajustes más importantes en la microscopía fluorescente es la alineación del iluminador. Como la amplitud de las señales de fluorescencia depende directamente de la cantidad de excitación, la iluminación desigual del objeto por una lámpara desalineada resultará en regiones brillantes y oscuras en la imagen de fluorescencia. [5] [17]

En la Figura 4 se puede apreciar una imagen obtenida con el microscopio de fluorescencia.

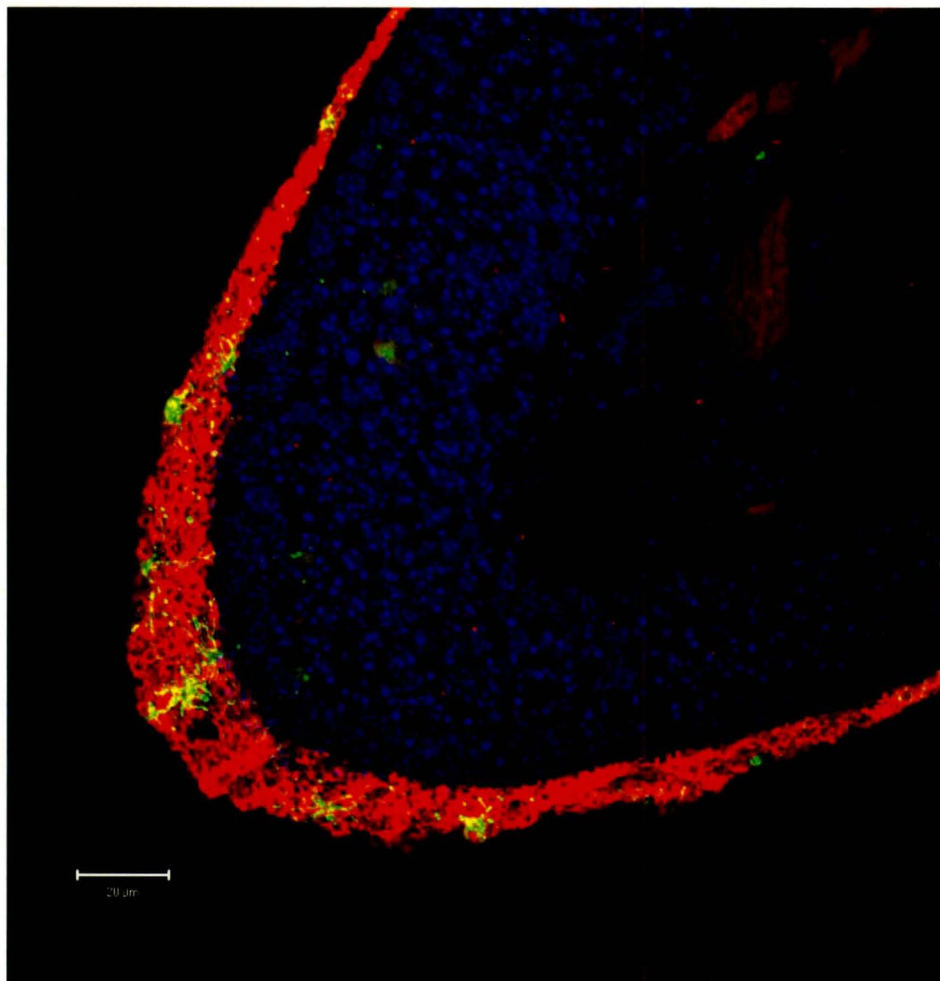


Figura 4: Tejido celular obtenido con el microscopio fluorescente.

4 Procesamiento de imágenes.

Una imagen puede ser definida como una función de dos dimensiones $f(x, y)$, donde x y y son coordenadas espaciales. La amplitud de la función f en cierto valor de coordenadas (x, y) representan lo que se conoce como la intensidad de la imagen en dicho punto. El término de nivel de gris se refiere a la intensidad de imágenes de un solo color. En cambio las imágenes de color están compuestas de una combinación de imágenes en 2-D, existen ciertas maneras de representar las imágenes en color. El sistema RGB representa una imagen a base a tres componentes individuales, los cuales son de color rojo, verde y azul. Las imágenes se componen de píxeles, los cuales son los puntos pequeños que dan la forma a la imagen. [8]

El procesamiento de imágenes involucra la aplicación de técnicas para lograr la mejora de la imagen o adaptarla para su uso en una computadora.

4.1 Transformaciones de intensidad y filtros espaciales.

4.1.1 Funciones de transformaciones de intensidad básicas.

La transformación de intensidad básica se encuentra entre la más simple de todas las técnicas de procesamientos de imágenes. Los valores de píxeles, antes y después del proceso, denotados por r y s respectivamente, están relacionados mediante una expresión de la forma $s = T(r)$, donde T es una transformación que mapea al píxel r dentro al valor s . Como estamos lidiando con valores digitales, en un ambiente de ocho bits, los valores que contiene T se encontrarán en un rango de 256 unidades. [8] [16] [19]

4.1.1.1 Los negativos de imágenes.

El negativo de una imagen con una intensidad de niveles en el rango de $[0, L-1]$ es obtenido tras usar la transformada negativa, la cual está dada por la siguiente expresión [8]:

$$s = L - 1 - r$$

Ecuación 1

Revisando la intensidad de niveles de una imagen después de la aplicación de la transformada se obtiene el equivalente a un negativo fotografía. Este tipo de procesamiento es particularmente adecuado para la mejora del detalle de blanco o gris, en referencia a las regiones oscuras de una imagen, especialmente cuando las áreas oscuras son dominantes en tamaño [16].

4.1.1.2 Transformaciones gamma (ley de potencias).

La transformación de la ley de potencias tiene una forma básica:

$$s = cr^\gamma$$

Ecuación 2

Donde c y γ son constantes positivas. En el caso de las transformaciones gamma, si se tienen valores fraccionarios de γ , los valores de s se vuelven más cercanos a $L-1$.

Esta transformada se usa mucho en los tubos de rayos catódicos (CRT), para poder corregir el fenómeno de *corrección gamma*. Por ejemplo, los CRT manejan un alto voltaje, el cual produce imágenes muy oscuras. [8]

4.2 Procesamiento de histogramas

El histograma de una imagen digital con intensidades en el rango $[0, L-1]$, es una función discreta $h(r_k) = n_k$, donde r_k es el k ésimo valor de intensidad y n_k es el número de píxeles en el rango de la imagen con intensidad r_k . Es de práctica común normalizar el histograma dividiendo cada uno de los componentes por el número total de píxeles en la imagen, lo cual se denota por el producto MN , donde, usualmente, M y N son el número de filas y columnas que existen en la imagen. Así, un histograma normalizado está dado por $p(r_k) = n_k/MN$, para $k = 0, 1, 2, \dots, L-1$.

Los histogramas son básicos para numerosas técnicas de procesamiento. La manipulación de éstos puede ser utilizada para el mejoramiento de imágenes. [8] [16]

4.2.1 Ecuación de histogramas.

La ecualización del histograma de una imagen es una transformación que pretende obtener, para una imagen, un histograma con una distribución uniforme. Es decir, se busca que exista el mismo número de píxeles para cada nivel de gris del histograma de una imagen monocroma. [8] [16] [19]

La función que cumple este criterio anteriormente descrito es la siguiente:

$$s_k = \frac{L - 1}{MN} \sum_{j=0}^k n_j$$

Ecuación 3

Donde MN es el número total de píxeles en la imagen, L es el número de posibilidades de intensidades en la imagen, k toma los valores de $0, 1, 2, 3 \dots L-1$ y n_j es el valor de intensidad que tienen en el valor j .

La Ecuación 3 se llama «ecualización de histograma» o la «transformación de linealización del histograma».

4.3 Filtrados Espaciales

Hay dos conceptos cercanamente relacionados que deben ser claramente entendidos cuando se realiza un filtrado espacial. Uno es *correlación* y el otro es *convolución*. Correlación es el proceso de mover la máscara filtradora sobre la imagen y calculando la suma de los productos en cada localización. La mecánica de la convolución es lo mismo, excepto que el filtro es rotado 180° . [8] [16] [19]

4.3.1 Filtros espaciales de alisado.

Los filtros alisados son utilizados para la reducción de ruido y para la aplicación del difuminado en las imágenes. El difuminado es aplicado para las tareas de procesamiento, tales como remover pequeños detalles de la imagen; la reducción del ruido puede ser realizada mediante el difuminado, utilizando un filtro lineal o no lineal.

Los filtros lineales se han llamado también «filtros de pasa baja», y su funcionamiento se basa en remplazar el valor de cada píxel en una imagen por el

promedio de los niveles de intensidad del filtro de la máscara. De este proceso resulta una imagen con una reducción en la *nitidez*, debido a las transformaciones en las intensidades. El propósito del filtrado es obtener una representación burda de los objetos de interés, tales como las intensidades de los objetos más pequeños. [8]

4.3.1.1 Filtro Gaussiano

Los filtros Gaussianos son una clase de filtro paso bajo, que está basado en la siguiente función de distribución de probabilidad

$$f(x) = e^{-\frac{x^2}{2\sigma^2}}$$

Ecuación 4

Donde σ es la desviación estándar. Un valor grande de σ produce una curva más plana que con un valor menor. Los filtros gaussianos son importantes debido a los siguientes casos [8] [12]:

1. Son matemáticamente bien comportados.
2. Son rotacionalmente simétricos, entonces son buenos puntos de partida para algunos algoritmos de detección de contorno.
3. Son separables; el filtro Gaussiano puede ser aplicado primero a una dimensión en el eje X, seguido por la otra dirección Y.
4. La convolución de dos Gaussianas resultan en otra Gaussiana.

Una función Gaussiana de dos dimensiones está dada por:

$$f(x, y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Ecuación 5

4.4 Segmentación de Imágenes

La segmentación subdivide una imagen dentro de sus regiones constituyentes u objetos. El nivel de interés del detalle sobre el cual la división es acarreada depende del problema a ser resuelto. La segmentación se puede dividir en diferentes aspectos como los que se mencionan a continuación [7] [15] [19]:

- Detección de contornos
 - Primera derivada
 - Segunda derivada
- Detector de Contorno de Canny
- Segmentación basado en umbrales

En la detección de contornos, existen varios tipos como se describió anteriormente. Los tipos referentes a la obtención de la primera derivada son los detectores de Prewitt, Sobel y Roberts. Para la segunda derivada, se emplean los métodos de detectores Laplaciano discreto, Marr-Hildreth y cruces por cero.

En la literatura [8] se menciona que el detector Canny es el más complejo, a pesar de no ser la última palabra para la selección de un detector de contornos.

4.4.1 Método de Canny

El método Canny implica la aplicación de procedimientos matemáticos de alta complejidad, por lo que no se explicará la totalidad de dicho modelo, sino los pasos más importantes para su logro; el método completo se encuentra desarrollado en el trabajo de John Canny en el 1986 [3]. El logro del detector de Canny esta basado en tres objetivos básicos:

1. *Tasa de error baja*: Todos los contornos idealmente deberían de ser detectados, sin obtenerse falsos resultados. Esto es, los bordes detectados tienen que ubicarse lo más cerca posible de los bordes verdaderos.
2. *Los puntos de los bordes deberían ser bien localizados*. La distancia entre el punto marcado como un contorno por el detector y el centro del contorno verdadero debería de ser el mínimo.
3. *Respuesta a un solo punto del contorno*. El detector debería regresar solamente un punto por cada pixel del contorno verdadero.

Los pasos a seguir para obtener el operador Canny, son los siguientes:

1. Realizar un filtro Gaussiano de dos dimensiones (x, y) para remover los ruidos innecesarios

2. Aplicar el proceso de convolución con la imagen original, sea $G(x, y)$ la función Gaussiana y sea $f(x, y)$ la imagen a filtrar. [8] [16] [19]

$$f_s(x, y) = G(x, y) * f(x, y)$$

Ecuación 6

3. Una vez hecho la convolución, el siguiente paso es calcular el gradiente, para lo cual se usa la primera derivada de la imagen para encontrar dichos gradientes.

$$g_x = \frac{\delta f_s}{\delta x}$$

Ecuación 7

Y,

$$g_y = \frac{\delta f_s}{\delta y}$$

Ecuación 8

4. Teniendo los componentes del gradiente se obtiene la magnitud del mismo.

$$M(x, y) = \sqrt{g_x^2 + g_y^2}$$

Ecuación 9

5. El siguiente paso es obtener la no supresión de los máximos (Nonmaxima Suppression, NMS por sus siglas en inglés). Para esto, hay que encontrar el ángulo de cada pixel de la imagen, la cual se obtiene con la tangente inversa de los componentes del gradiente, de acuerdo a la siguiente expresión:

$$\alpha(x, y) = \tan^{-1} \left[\frac{g_y}{g_x} \right]$$

Para el NMS, se divide los ángulos obtenidos en cuatro regiones, lo cual se puede apreciar en la Tabla 1

Tabla 1: Regiones del NMS

Región	Grados	Localización del pixel
1	0° a 22.5°	$y \leq 0 \ \& \ x > -y$
2	22.5° a 67.5	$x > 0 \ \& \ x \leq -y$
3	67.5° a 112.5	$x \leq 0 \ \& \ x > y$
4	112.5° a 157.5	$y < 0 \ \& \ x \leq y$
1	157.5° a 0°	$y \geq 0 \ \& \ x > -y$

6. Después del uso del NMS, ahora se puede aplicar el umbral para obtener una imagen binaria de los contornos. Canny propuso que era preferible emplear una vieja técnica de umbral con histéresis, la cual selecciona un umbral alto TH (High Threshold), y uno bajo TL (Low Threshold), para poder remover los falsos negativos. [3] [7]

4.4.2 Segmentación basada en umbrales.

Una imagen en escala de grises es convertida en una imagen binaria (blanco y negro). Primeramente, se elige un nivel de gris T en la imagen original y luego se convierte cada pixel, blanco o negro, en función de su valor de gris, es decir, si es mayor que o menor que T [3] [16] [20]:

Un pixel se convierte en: $\left\{ \begin{array}{l} \text{blanco si el nivel de gris es } > T \\ \text{negro sí el nivel de gris es } \leq T \end{array} \right.$

En el libro de *Introduction to Digital Image Processing* del autor Alasdair M. Andrew [16], se menciona que el método de umbral puede ser útil en remover detalles innecesarios de una imagen para concentrarse en los esenciales; de esta información se puede obtener lo necesario para investigar tamaños, formas y número de partículas en una imagen.

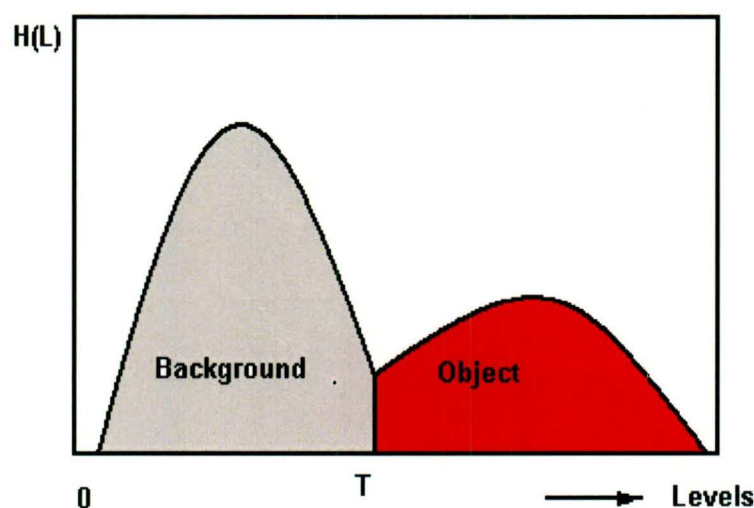


Figura 5: Unión de nivel de grises en un histograma

El método para seleccionar el nivel de interés para el umbral es a través de los histogramas. En la

Figura 5, se muestra cómo se escoge el nivel de grises en el punto de intersección del objeto y del fondo de la imagen. Pero cuando el histograma no se comporta con una distribución normal, existe un método de probabilidad alternativo para obtener el nivel de gris, conocido como el método de Otsu. [8] [16] [19]

$$\omega(k) = \sum_{i=0}^k p_i$$

Ecuación 11

Donde p_i es la probabilidad de que un pixel tenga el nivel de gris de i , k es el umbral en el nivel k .

4.5 Morfología de la imagen

La morfología es un amplio conjunto de operaciones de procesamiento de imágenes que las imágenes del proceso están basado por figuras. Operaciones

morfológicas aplican un elemento estructural para una imagen de entrada, creando una imagen de salida del mismo tamaño. En una operación morfológica, el valor de cada píxel de la imagen de salida se basa en una comparación del píxel correspondiente en la imagen de entrada con sus vecinos. Al elegir el tamaño y la forma de la zona, puede construir una operación morfológica que es sensible a las formas específicas en la imagen de entrada.

Las operaciones morfológicas más básicas son la dilatación y la erosión. La dilatación añade píxeles a los límites de los objetos en una imagen, mientras que la erosión elimina los píxeles en los bordes del objeto. El número de píxeles añadidos o retirados de los objetos en una imagen depende del tamaño y la forma del elemento estructural utilizado para procesar la imagen. En la dilatación morfológica y las operaciones de la erosión, el estado de cualquier píxel de la imagen de salida se determina mediante la aplicación de una regla para el píxel correspondiente y sus vecinos en la imagen de entrada. La norma que se utiliza para procesar los píxeles define la operación como una dilatación o la erosión una. En esta Tabla 2 se muestran las reglas tanto para la dilatación y la erosión. [15]

Tabla 2: Reglas para el uso de la dilatación y de erosión (MatWorks, 2010)

Operación	Regla
Dilatación	Los píxeles más allá del borde de la imagen se le asignan el valor mínimo que ofrecido el tipo de datos. Para las imágenes binarias, estos píxeles se supone que se establece en 0. Para escala de grises, el valor mínimo para uint8 imágenes es 0.
Erosión	Los píxeles más allá del borde de la imagen se le asignan el valor máxima permitida por el tipo de datos. Para imágenes binarias, estos píxeles se supone que se establece en 1. Para escala de grises, el valor máximo para uint8 imágenes es de 255.

Según Rafael González, la morfología es una rama del procesamiento de imágenes particularmente útil para analizar figuras en imágenes. El trabajo de *Medicals Diagnosis Based on Analysis and Processing* [10] utiliza lo que son las

herramientas morfológicas para encontrar las células que se encuentran completas, lo que aprueba el enunciado anterior de González [7].

En las imágenes siguientes se muestran los resultados de aplicar los elementos de morfología. En la Figura 6 a se muestran los contornos de los granos de arroz; en b se puede apreciar cómo se aplicó una dilatación; en la imagen c, se presenta el contorno de un circuito electrónico; y en la imagen d, se observa cómo está erosionada por un elemento morfológico.

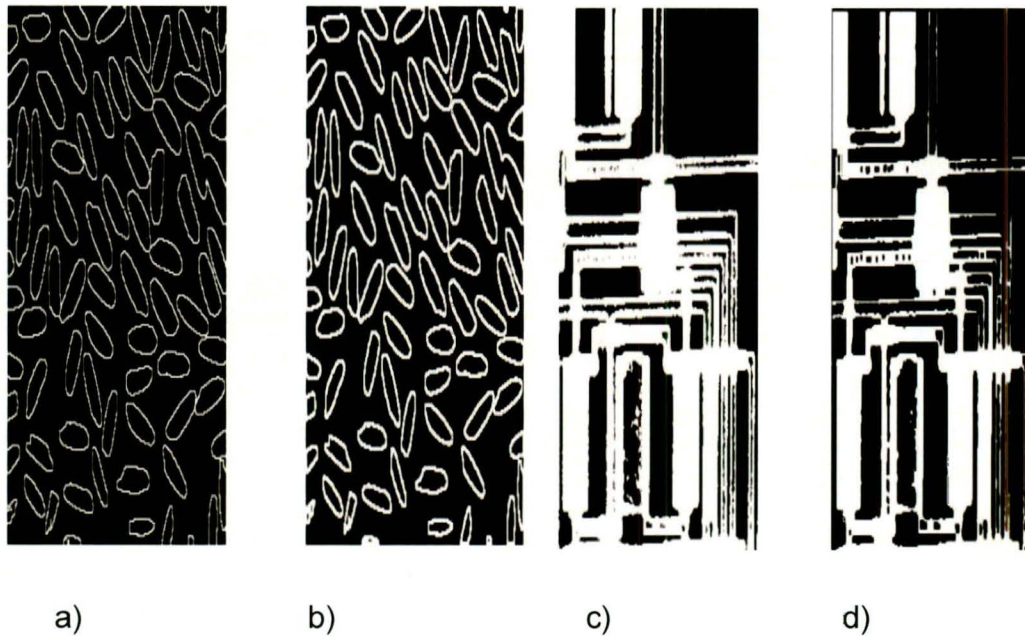


Figura 6: Dilatación en los granos de arroz [a, b] en el lado derecho se aplicó erosión a un circuito integrado [c d] (González R & Woods, R, 2007)

En el uso de la dilatación, los bordes de la imagen se engruesan, y en el caso contrario se obtiene la erosión. En el artículo de Sirikan Chucherd y Stanislav Makhanov [6] se señala que el uso de la erosión auxilia en el buen desempeño en el procesamiento de imágenes para la detección del cáncer.

4.5.1 Etiquetado de imagen

El etiquetado de imagen (image labeling) se refiere a cómo se enumeran los objetos en una imagen para obtener un conteo de todos ellos. En el artículo de Negar M. [9], se comenta que el uso del *etiquetado de imagen* ayuda para contar

las células del epitelio cervical. Además, el autor utiliza el método que propone Matlab para un mejor rendimiento en el algoritmo, el *bwlabel*.

4.6 Región de interés (ROI).

No se requiere aplicar un filtro a toda una imagen, sólo a una pequeña región de ella. El uso de una ROI es necesario para descartar de las imágenes de diagnóstico las áreas que no se requiere procesar, entonces se aplica una máscara para obtener el área deseada.

En el artículo *Automated cell counting and cluster segmentation using concavity detection and ellipse fitting techniques* [9] y en *Improving renal cell carcinoma classification by automatic region of interest selection* [6] los autores describen el uso que tiene un ROI para el procesamiento de imágenes. En la Figura 7 es posible observar que hay un tejido de renal sin tener una máscara de ROI; en seguida se percibe el resultado después de aplicarla.

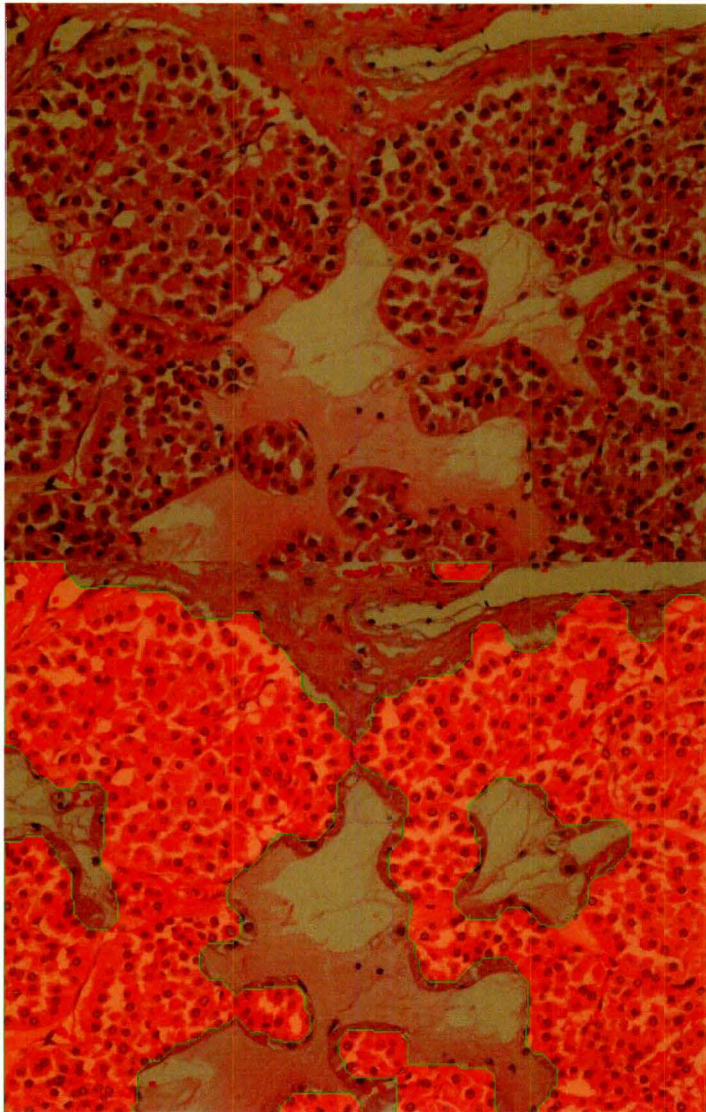


Figura 7: A) Tejido celular renal B) Tejido celular renal con una ROI (Harandi, N, S Sadri, and N Moghaddam, 2009)

4.7 Agrupamiento de K-medio

El K-Medio es una técnica de agrupamiento no jerárquico que sigue un procedimiento simple para clasificar un determinado conjunto de datos a través de un cierto número de conglomerados K , que se conocen a priori. Es un método de segmentación de imágenes, ya que agrupa en clústeres la representación. Existen varios artículos [6] [10] [12] que aprueban el uso de *k-medios*, como el primer paso para comenzar la segmentación de una imagen.

A continuación se muestran dos imágenes que indican cómo divide el *k-medio* una imagen, la imagen es obtenida de una resonancia magnética del

cerebro; la figura de la izquierda es la imagen original, y aplicando este método se observa cómo los colores diferencian las diferentes áreas de la imagen.

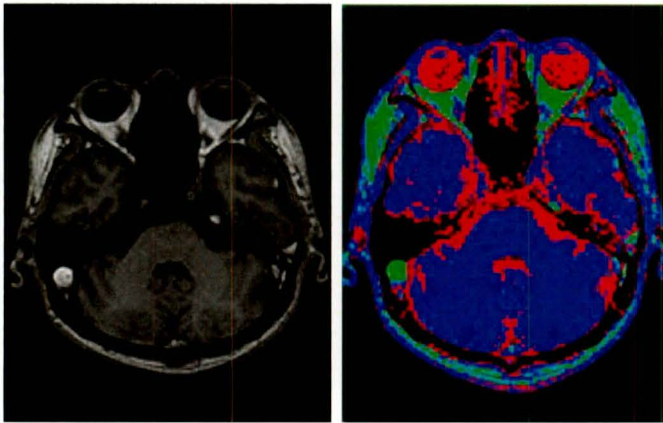


Figura 8: El uso de k-medio en un imagen cerebral tomada de una resonancia magnética (Ilea, Dana E. and Whelan, Paul F., 2006)

5 Metodología.

En la siguiente Figura 9 se puede observar un flujograma que indica los pasos que se siguieron para obtener el conteo de células cancerígenas, el cual hace referencia al trabajo de Negar M. [9] [11]

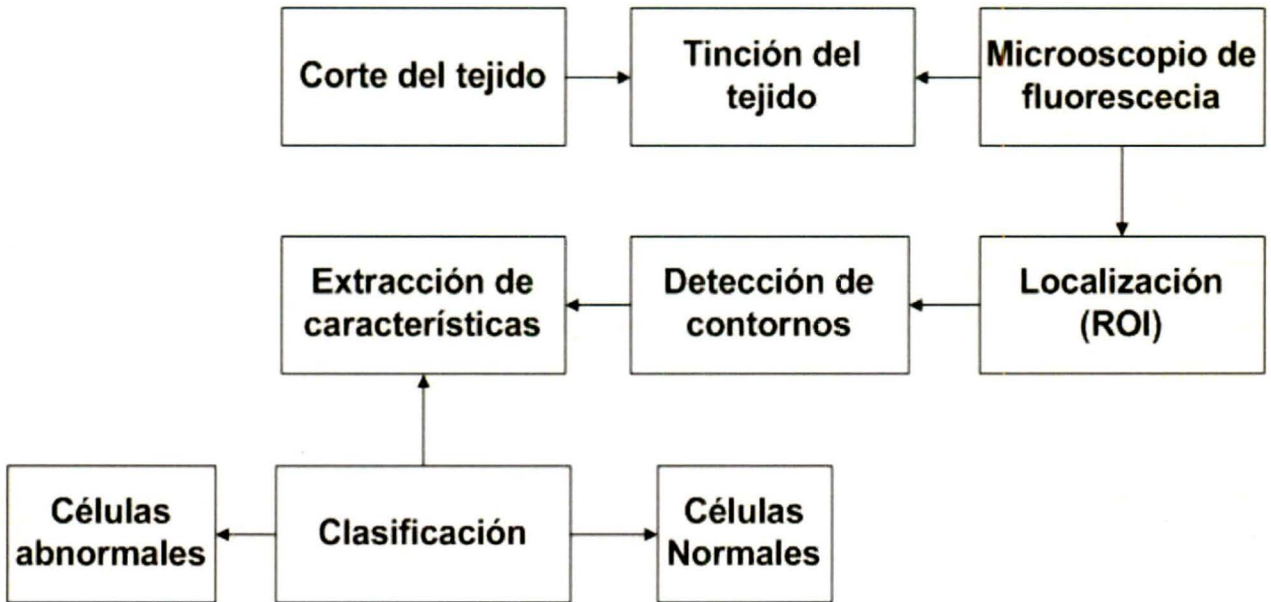


Figura 9: Flujo grama del procesamiento de imágenes (Harandi, N, S Sadri, and N Moghaddam., 2009)

En la adquisición del tejido celular se siguieron los pasos que se dicta en “Pancreatic β cells synthesize and secrete nerve growth factor” [18] y en artículo de Vidaltamayo, R en el 1996 [22], en esos artículos se menciona cómo se debe de cultivar el tejido celular.

En la tinción de tejidos celulares existen diferentes marcadores para lograr el objetivo de conteo de células. Esto depende de lo que se quiere teñir, es decir si queremos observar una proteína en específico o un organelo de la célula. En este caso, los marcadores que se utilizarán para lograr la tinción es el DAPI para identificar los ácidos nucleicos de las células; el antígeno mOR2, obtenido a partir de una rata, para marcar la proteína específica. Una vez reconocida la proteína específica, se crea un anticuerpo del antígeno anteriormente mencionado. Para esto, se debe emplear una especie superior, es decir, el antígeno utilizado en este caso es de conejo anti-rata.

Después de haber señalado la proteína, se marca con el fluoróforo. Los fluoróforos utilizados son de la marca *Alexa Fluor*® dye, en especial el Alexa Fluor® 488 (que tiene una longitud de onda de absorción de 496nm y de emisión de 519 nm) y el Alexa Fluor® 555 (que tiene una longitud de onda de absorción de 555 nm y de emisión de 565 nm). En la siguiente Figura 10 se muestran las longitudes de onda de los dos fluoróforos; la onda cuatro y la onda nueve representan respectivamente los marcadores fluorescentes utilizados.

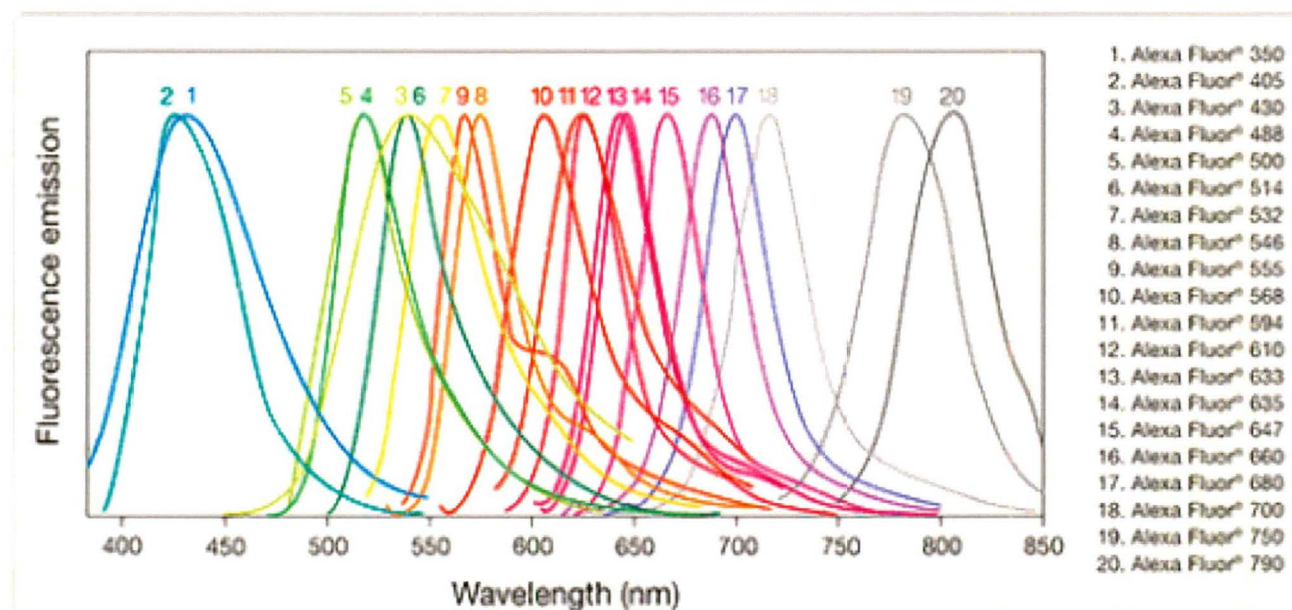


Figura 10: Longitud de onda de los marcadores de Alexa (Life Technologies 2012)

La etapa de amplificación se realizó con el microscopio de Zeiss, modelo Axiovert-135, con amplificación de 25X. Para el funcionamiento del microscopio se aplicaron los filtros para detectar la longitud de onda que emite el fluoróforo para obtener la imagen del elemento que se está observando.

El filtro que se ocupó para analizar las muestras con el marcador Alexa Fluor® 488 es el Filter Set 09 (número de serie 488009-9901-000) de la compañía Carls Zeiss. En la siguiente Figura 11, se observa el funcionamiento del filtro, que por sus características fue seleccionado.

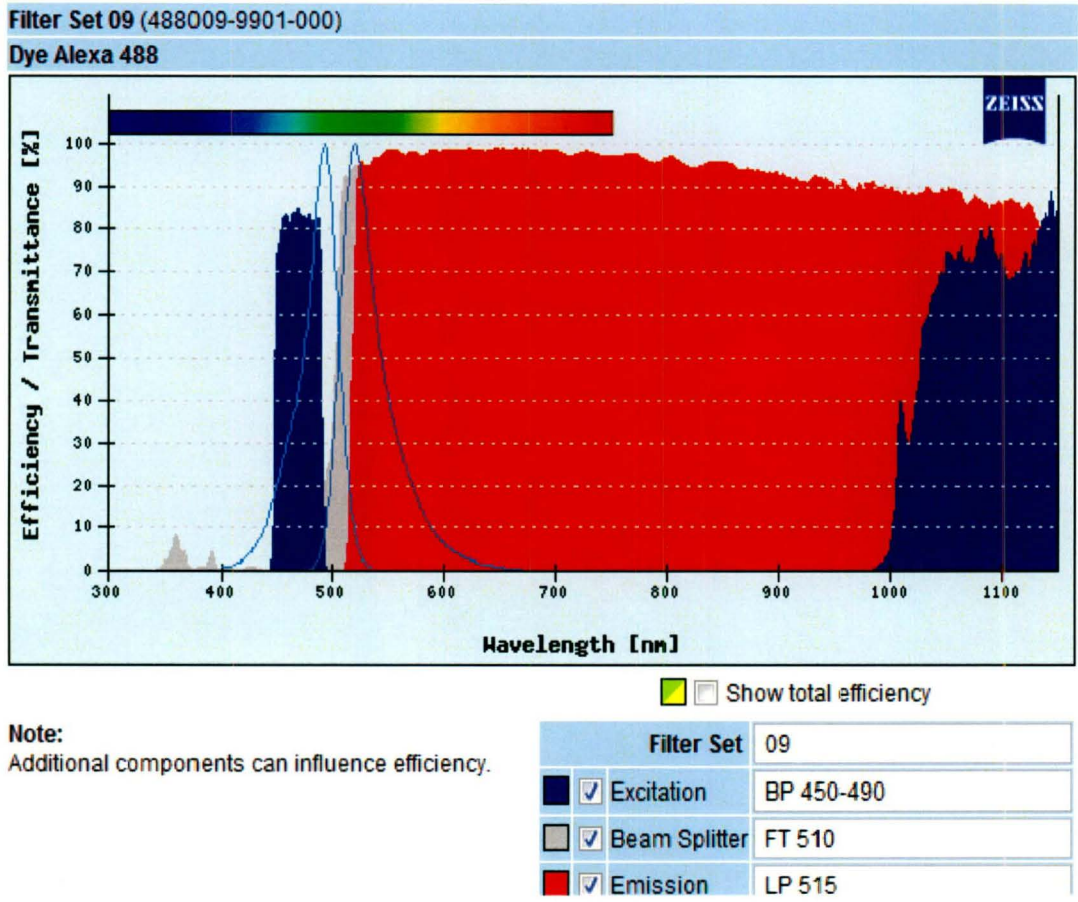


Figura 11: Filtro para el marcador Alexa 488 (Zeiss, 2012)

Para el fluoróforo Alexa Fluor® 555 existe el filtro *Filter Set 43* (número de serie 000000-1114-101) de la compañía Carls Zeiss, de igual manera se muestra en la siguiente Figura 12 las características del filtro mencionado anteriormente.

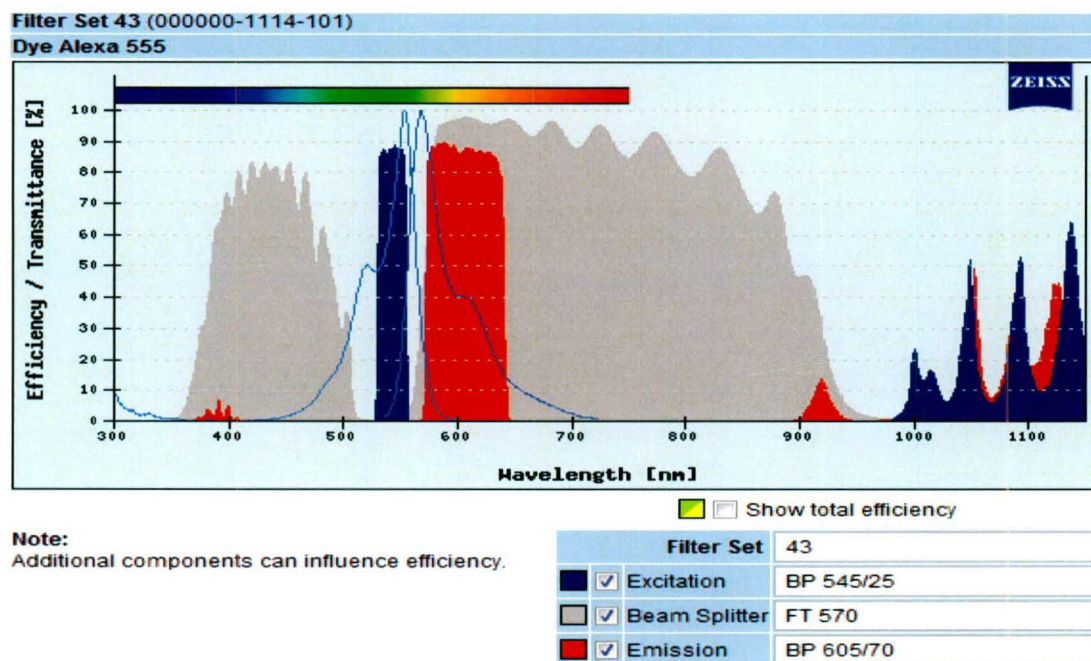


Figura 12: Filtro para el marcador Alexa 588 (Zeiss, 2012)

Terminado el proceso de amplificación de la imagen, ésta se encuentra en una escala de grises. Es necesario modificarla para que se encuentre en un formato RGB. Para ello, se procesa la imagen con un LUT (Look up-table) para poder realizar el cambio. En este caso se coloca un LUT con el color azul del formato RGB, los objetos que son coloreados con este color son los que se tiñeron con el marcador DAPI, el LUT que se emplea para agregar color al marcador de Alexa Fluor® 488 es el canal verde del formato RGB [17].

Cuando se aplica un LUT a una imagen en escala de grises no se alteran los píxeles de la imagen, solamente se altera la forma de mostrarlos. A continuación se ve en la Figura 13, un tejido celular ya con la aplicación del LUT. El tamaño de las imágenes son de 10 bits, o sea que la imagen tiene una resolución de 1024 píxeles por 1024 píxeles. [17]

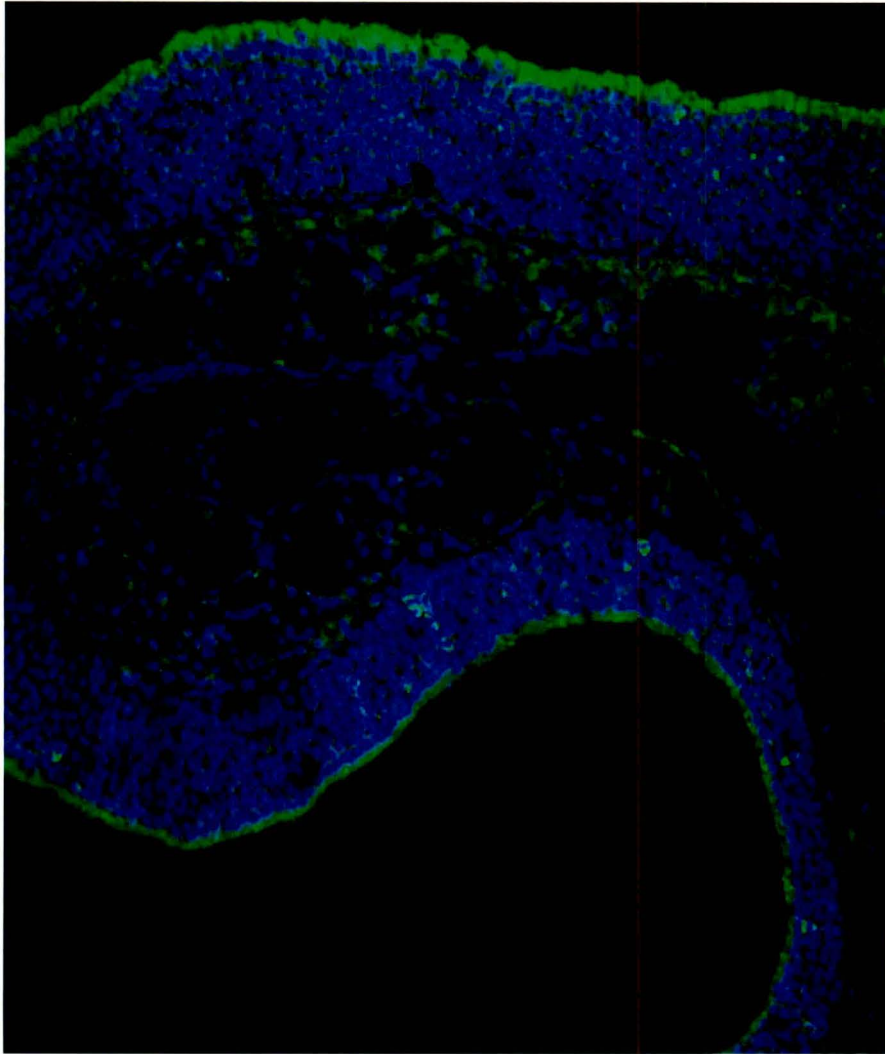


Figura 13: Figura con los LUTs ya procesada

Cuando se obtiene la imagen ya digitalizada en formato RGB, se comienza lo que es el procesamiento de imagen. El primer paso que se realiza es seleccionar la imagen que se desea analizar. A continuación se muestra la Figura 14 para seleccionar el tejido que se procesará.

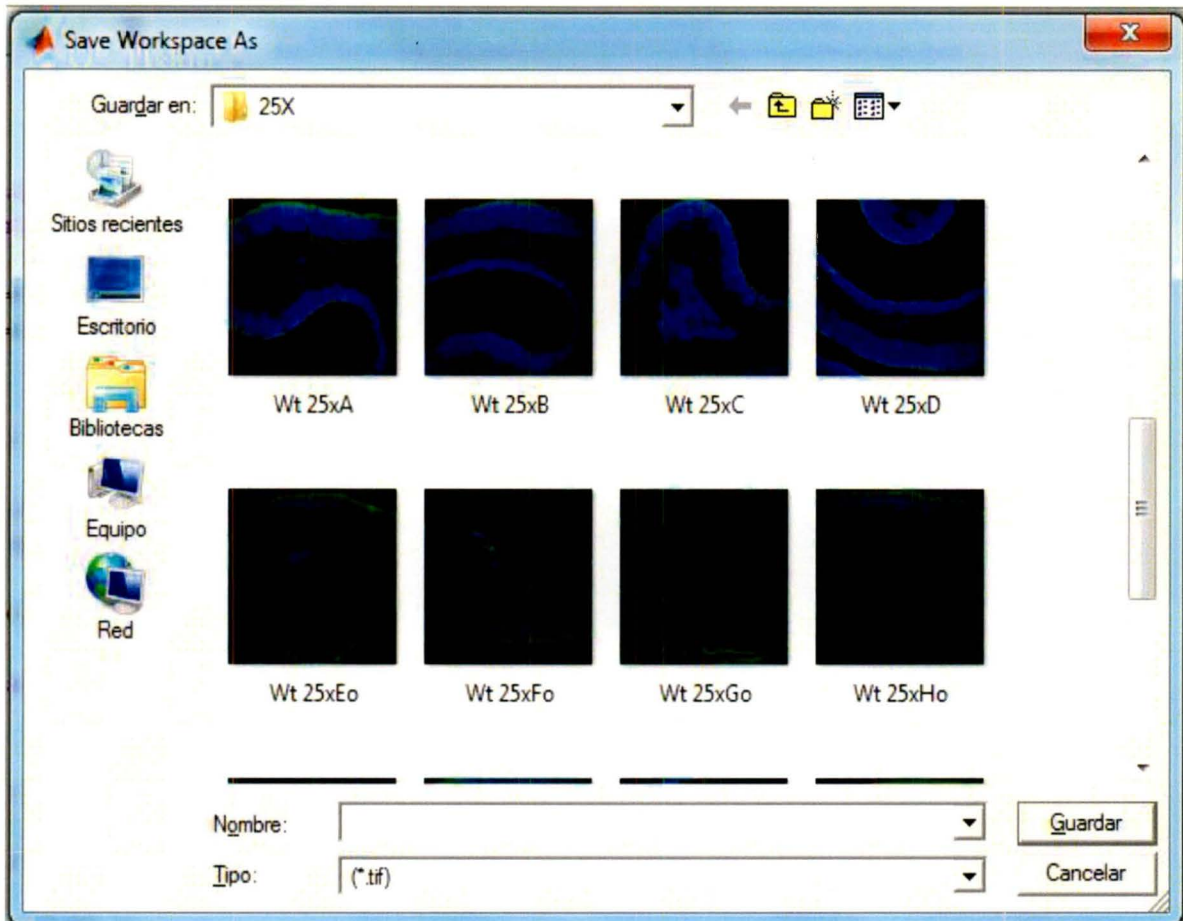


Figura 14: Pantalla para abrir una imagen en Matlab

Teniendo la imagen deseada, ahora se selecciona el número de áreas en que se divide la imagen. Por ejemplo, la Figura 15 cuenta con dos áreas a procesar; se despliega un mensaje de diálogo para que el usuario ingrese dicho número.

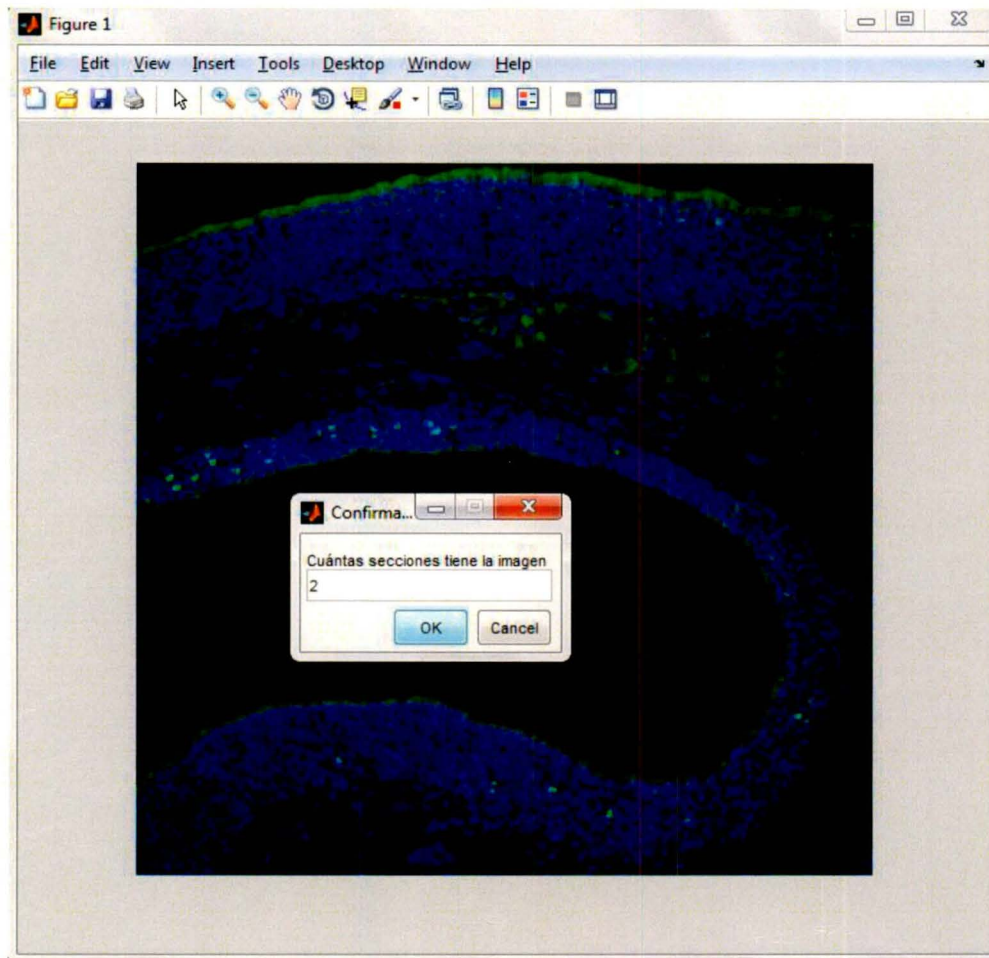


Figura 15: Selección de áreas en la imagen.

Teniendo el número de secciones a analizar, es posible comenzar con una región de interés (ROI). El trabajo de Qaiser Chaudry S. [5] demuestra que se puede obtener una ROI de manera automática. Para el beneficio del cliente se programó que él mismo seleccione dónde quiere tener su región de interés.

El software Matlab [15] cuenta con una función en su librería en la que se puede seleccionar, por medio de puntos, la región deseada por el usuario. Esta función es $[X Y] = \text{getpoints} ()$. La siguiente Figura 16 demuestra la aplicación de lo que se mencionó antes, el área que se encuentra encerrada entre los puntos que se ven en la imagen es el área de interés. La función explicada retorna las coordenadas $X Y$ en el plano de la imagen, después se requiere convertir dichas coordenadas en la ROI. Para ello, se emplea la función *roipoly*, que está disponible en la librería, lo cual genera una imagen binaria. Lo que se encuentra

dentro de la ROI será blanco (uno lógico) y lo que se encuentra fuera será negro (cero lógico).

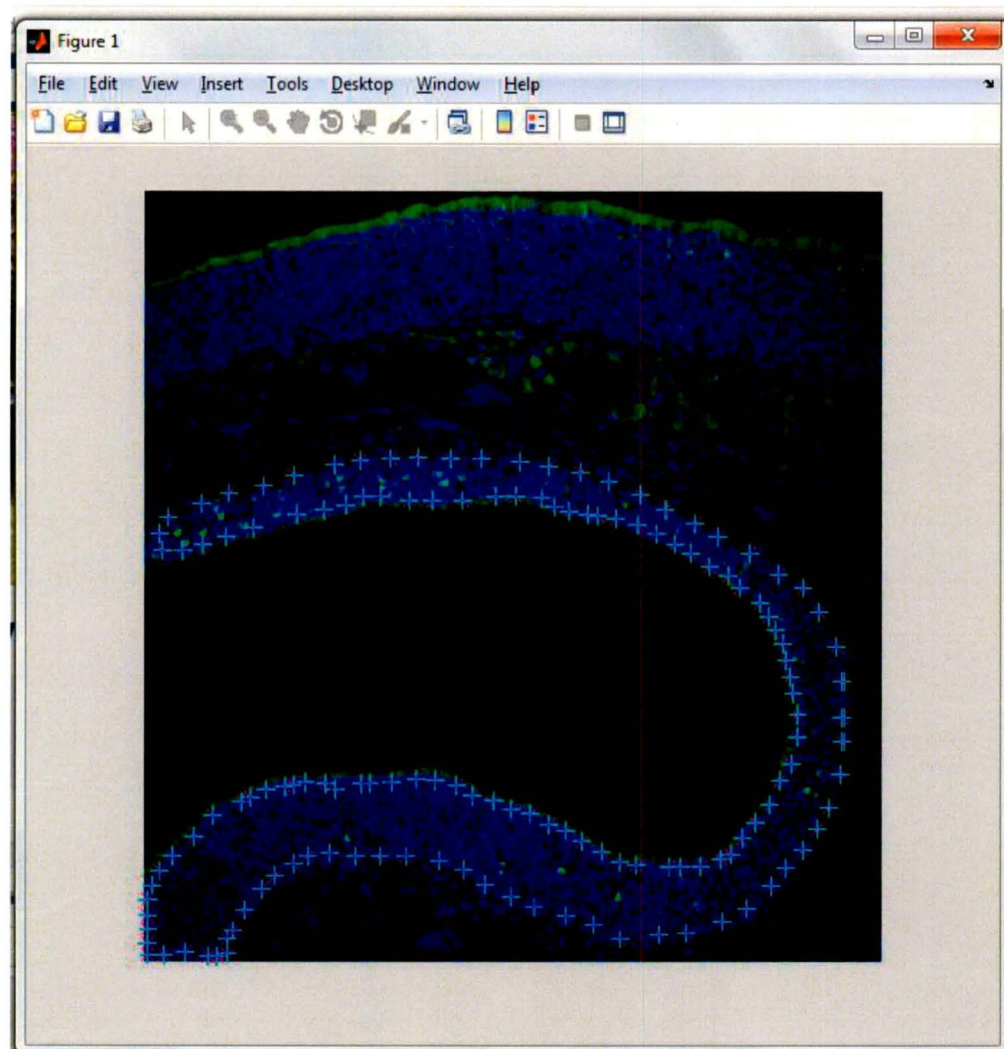


Figura 16: Imagen con ROI seleccionada

Para finalizar nuestra ROI, se debe aplicar una transformada negativa a la selección creada recientemente; esto se debe a que al realizar la resta de la imagen original de la ROI, la imagen que resulta es negativa, y las intensidades que se emplean para detectar los contornos no son lo suficientemente altas para detectar contornos. En el libro de González R. [7], se explica que la transformación negativa convierte los unos lógicos a ceros lógicos y los ceros lógicos son transformados en unos lógicos.

Terminando de analizar la región de interés, se desagruparán los componentes RGB en sus canales rojos, verdes y azules. Una vez separados los canales se pueden analizar las células según su marcador.

El siguiente paso es obtener la intensidad de las células que se desea analizar, para esto se requirió observar los histogramas de intensidad de la imagen. El valor que se obtuvo fue por prueba y error. En la siguiente Figura 17, se puede observar que del lado izquierdo, el histograma de la imagen se comporta con distribución de campana, con sesgo a la izquierda; del lado derecho de la misma figura se muestra la representación del canal verde de la imagen original, donde se puede apreciar que hay una gran cantidad de negros, siendo esa la razón por la que el histograma se recarga a la izquierda.

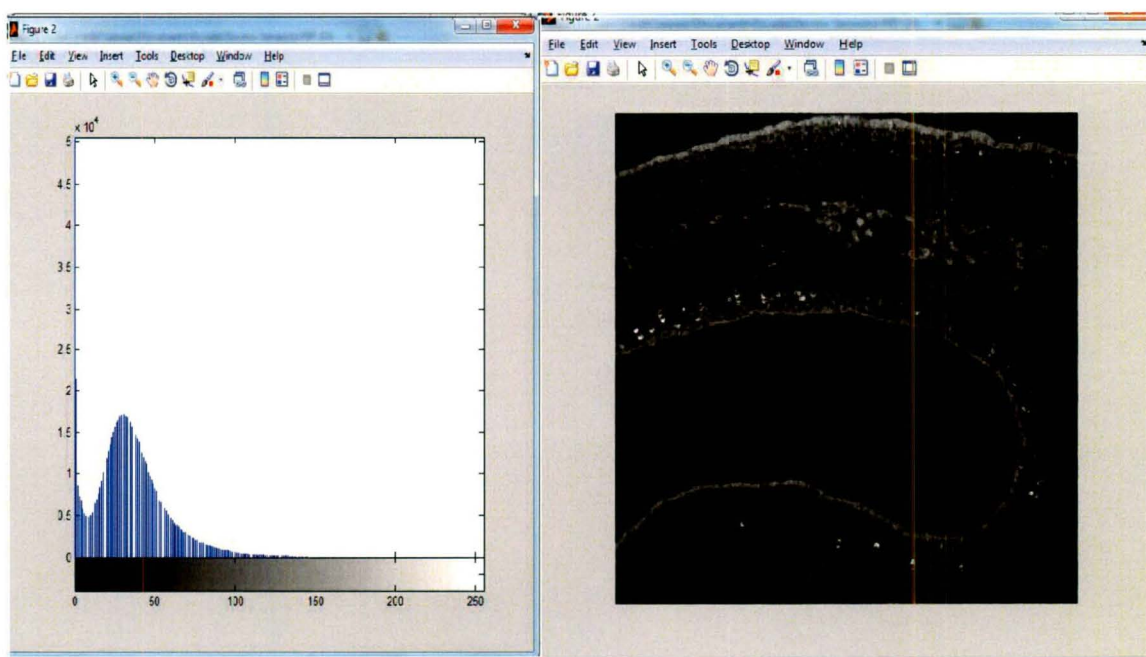


Figura 17: Histograma del canal G de la imagen a procesar.

Se observa en la Figura 18 la obtención del umbral a partir del canal verde, el cual es el valor en gris que se tiene del contorno de la célula. Se logró contar este valor con la función *data cursor*, la cual regresa la coordenada del pixel marcado y el valor de intensidad que tiene en la casilla. Para corroborar que dicho valor es el adecuado, se confirmó con el histograma visto el párrafo anterior con el método de las tres desviaciones estándar con respecto de la imagen ($\mu = 26 \sigma =$

26). En la siguiente imagen se ve la función *data cursor* sobre el borde de una célula, la cual muestra un valor de gris 101, como está dentro de las tres desviaciones es aceptable.

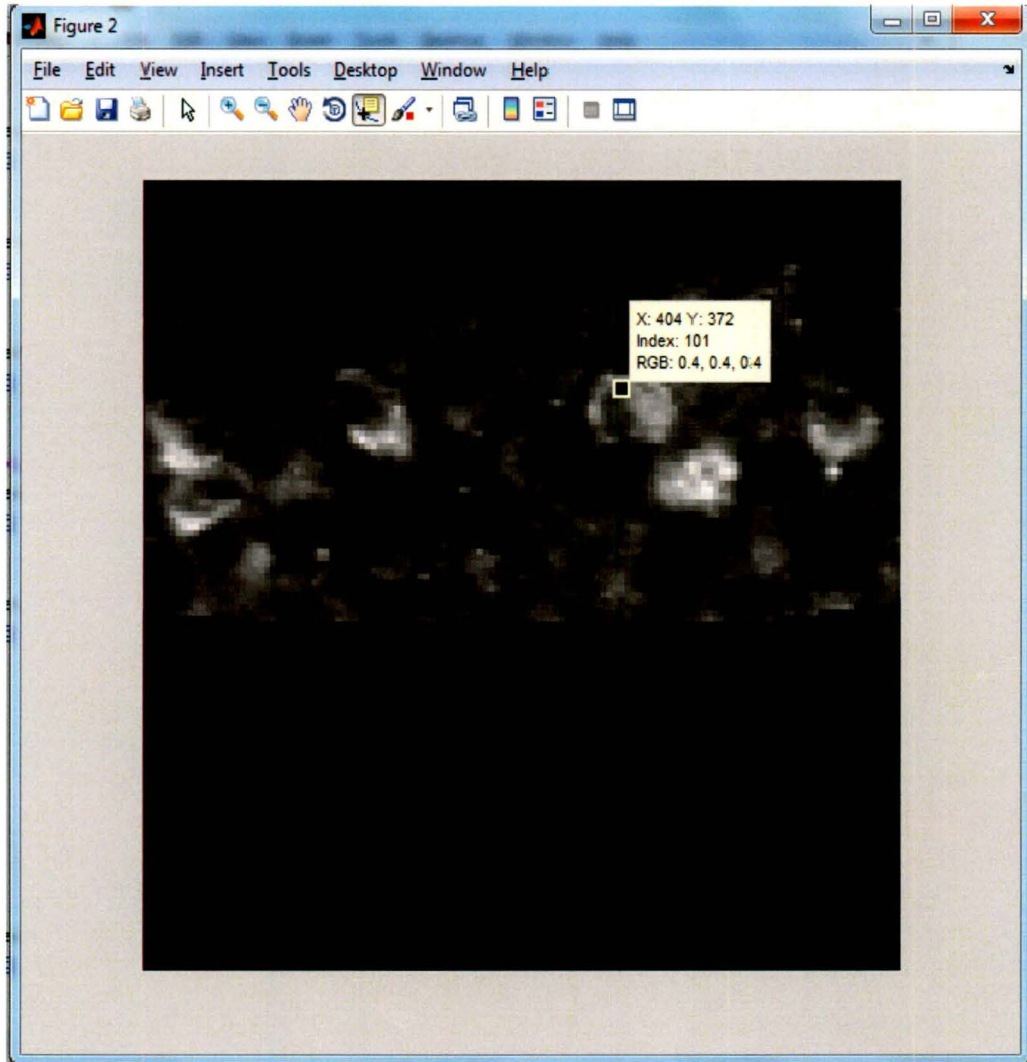


Figura 18: Valor en escala de gris en el contorno de una célula

Teniendo el nivel de umbral ya establecido se debe convertir la imagen en binaria; una vez hecho esto se aplica el algoritmo de K-media que se puede ver en los anexos “Programa: K-means”. Lo que realiza dicho es acomodar los elementos por secciones dependiendo del número de agrupamientos utilizados [10].

Teniendo separada la imagen por el *k-means* ya se podrá realizar la segmentación de la imagen. Se comienza con la detección de contornos, como se ha mencionado anteriormente. Uno de los mejores detectores de contornos

espaciales que existe es el Canny [3], se pueden observar las líneas de programación en el anexo “Programa: Canny”. En la siguiente Figura 19 se demuestra el resultado del uso de este detector.



Figura 19: Imagen con el detector de contornos de Canny

En el artículo de Sonal Kothari [12], se comenta que para eliminar el ruido de una imagen se tiene que aplicar un filtro Gaussiano, con un $\sigma = 1.2$. En la siguiente Figura 20 se puede observar la diferencia en la imagen con el detector Canny y una vez aplicado el filtro de Gaussiano.

Con el filtro aplicado se ve que el ruido presente en el detector Canny fue eliminado.

Para seguir eliminando datos que no son necesarios, se aplica un eliminador de objetos; esta función es *bwareaopen* la cual consiste en quitar objetos que tengan menos píxeles que un valor P . En la siguiente Figura 21 se percibe la eliminación de los objetos que no cumplen con dicha condición. [15]

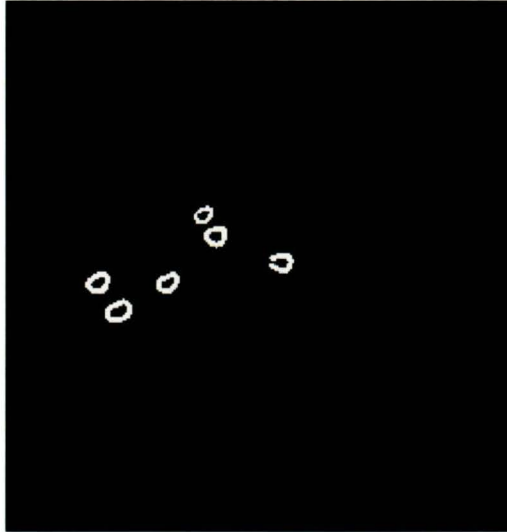


Figura 20: Filtro Gaussiano aplicado a la imagen con Canny.

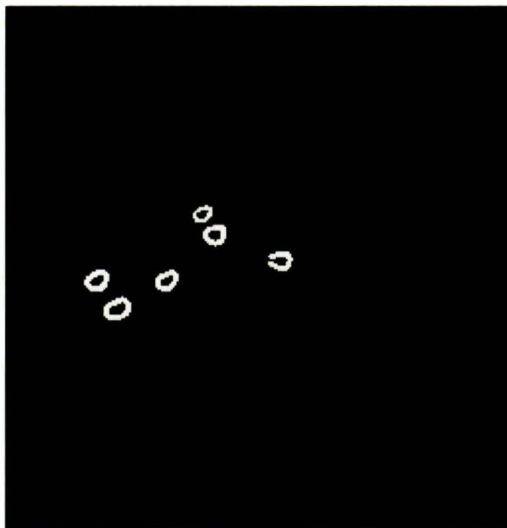


Figura 21: Imagen con elementos eliminados por un nivel de pixeles menor que P.

Si la imagen presenta que una célula se traslapa con otra, se debe usar una operación morfológica, la cual es la erosión. Para esto se ocupa un elemento morfológico el cual se llama “*strel*”. Este elemento permite realizar la función de la erosión, la cual es disminuir los bordes de los objetos, y así separar las células

que se encuentran unidas. En la siguiente Figura 22 se observa como se erosionó la imagen y eliminaron las partes innecesarias, tal como lo comentan los autores Sirikan Chucherd y Stanislav Makhanov [7].

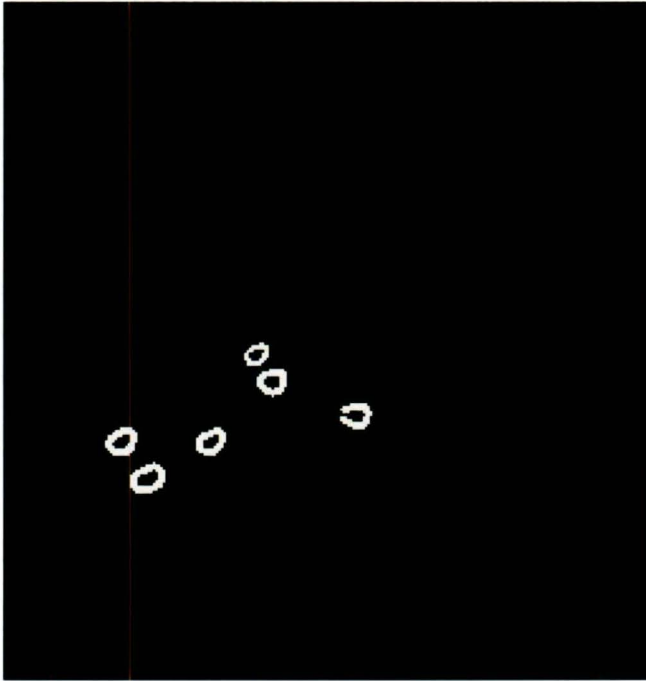


Figura 22: Imagen que se le aplico una función morfológica de erosión

Teniendo ya terminada la imagen con los procesos anteriores, es necesario saber en donde quedan las células en un plano X Y (espacial), para poder ver si existen ácidos nucleicos en esa región donde se encuentran los marcadores *Alexa Fluor*® 488. La siguiente Figura 23 muestra un plano X Y con las células.

Al final se acoplan los marcadores verdes (*Alexa Fluor*® 488) con los marcadores azules (DAPI), para corroborar cuáles son las células afectadas. Con la siguiente Figura 24 se puede observar cuáles son las células dañadas con cáncer

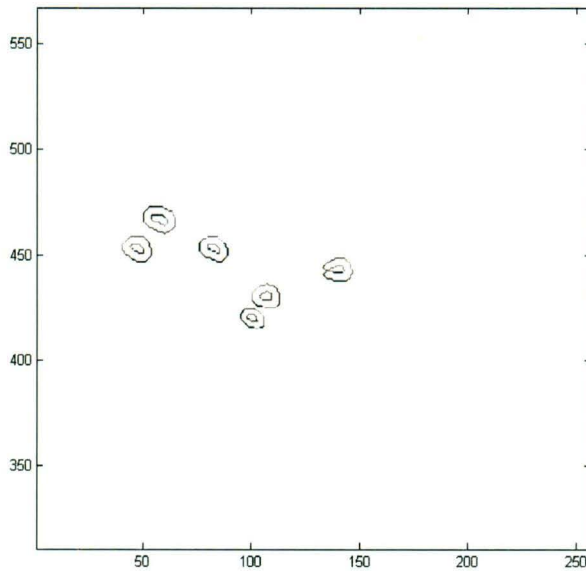


Figura 23: Plano X Y de las células.



Figura 24: Imagen con las células con cáncer encima de los ácidos nucleicos

Realizando todo lo anterior es posible contar las células con el método de *bwlabel* que maneja Matlab [7]; el método lo recomiendan diferentes autores [2] [9] que es un buen etiquetado de objetos.

Existe un método en Matlab [15] para obtener el área total de la región interesada, la cual es *bwarea*, el cual te da el área en pixeles teniendo la resolución del microscopio y cuál es el área total que ocupa una imagen de 1024 pixeles por 1024 pixeles se puede obtener el área en μM que ocupan en total las

células en la imagen. La siguiente Figura 25 demuestra la información desplegada en un cuadro de dialogo.

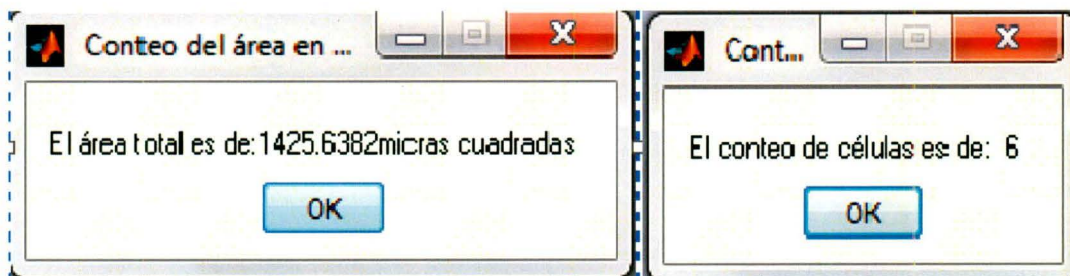


Figura 25: Despliegue de la información al final el proceso de imagen.

6 Resultados:

El presente proyecto busca separar de forma automática los tres canales del formato RGB, lo cual se pudo hacer de manera simple y rápida sin la necesidad de tener que seleccionar un comando; el programa permite realizarlo automáticamente al seleccionar a la imagen.

Un proceso que ayudó a la mejora del conteo celular es la selección manual de la región de interés (ROI), ya que el cliente presentaba un problema para seleccionar una ROI en el programa que usa, ImageJ, haciendo una herramienta más adecuada al usuario logrando la eliminación de estructuras no deseadas en el conteo.

Al implementar el detector Canny aumentó la calidad al encontrar los contornos de las células haciendo mucho más visibles las estructuras de las células, esto es una ventaja por que el programa ImageJ no cuenta con este tipo de detector.

Para verificar el algoritmo se comprobaron las cantidades arrojadas por el programa con una lista de cinco diferentes imágenes de células cancerígenas, que tiene las cantidades de células que hay en cada una. A continuación se presenta la Tabla 3 que tiene tres columnas las cuales son: conteo del usuario manualmente, conteo del presente algoritmo y conteo del usuario con el programa ImageJ.

Tabla 3: Tabla de resultados del programa creado

Imagen	Conteo del usuario manual	Conteo con el presente algoritmo*	Conteo del usuario con ImageJ
Wt 25xA	20	20	20
Wt 25xJi	42	46	52
Wt 25xB	26	22	27
Wt 25xC	28	30	25
Wt 25xD	40	33	47

***El conteo puede variar por como seleccionar la región de interés.**

El rango de error que tiene el programa está dentro del diez por ciento ($\pm 10\%$), el cual se estableció como objetivo, así como la mejora del tiempo en el procesamiento de imágenes, el cual es alrededor de dos minutos por sección de estudio, lo cual baja significativamente el tiempo de conteo por que el usuario se tomaba dos horas mínimo en analizar manualmente solamente una imagen.

7 Conclusiones:

Con este estudio se utilizó el análisis de procesamientos de imágenes para poder encontrar un algoritmo que pueda contar de manera eficiente las células presentes en imágenes de tejidos de animales, lo cual es un indicador para diagnosticar el crecimiento del cáncer en un paciente.

Para mejorar el conteo de células se observó en el trayecto del estudio que hay que mejorar la obtención de la imagen, ya que los niveles de intensidad son bajos y se podrían confundir con otras estructuras celulares.

En un futuro, el siguiente paso es realizar el mismo programa pero ahora en un lenguaje en Java por las licencias gratuitas que maneja Sun Systems.

8 Bibliografía.

1. Aizawa, K., Sakaue, K., & Suenaga, Y. (2004). *Image processing technologies: Algorithms, sensor and applications*. (Primera ed., p. 282). New York: Marcel Dekker.
2. Bai, X, C Sun, and F Zhou. (2008) "Touching cells splitting ellipse fitting by using concave points." Ed. IEEE: Dicta Digital Image. Canberra: IEEE Computer Society, pp 271-278.
3. Canny, J (1974) "A computational approach for edge detection". *IEEE Trans. Pattern Anal. Machine Intell.*, vol 8, no 6, pp 678-698
4. Carl Zeiss (2012) *Filter set 09*. Recuperado el 19 de abril del 2012, de https://www.micro-shop.zeiss.com/us/us_en/spektral-info.php?i=488009-9901-000
5. Carl Zeiss (2012) *Filter set 43*. Recuperado el 19 de abril del 2012, de https://www.micro-shop.zeiss.com/us/us_en/spektral-info.php?i=000000-1114-101
6. Chaudry, Q, S Raza, Y Sharma, et al. (2008.) "Improving renal cell carcinoma classification by automatic region of interest selection." Ed. IEEE International Symposium. Athens: System, Man, & Cybernetics Society, pp 1-6.
7. Chucherd, S, and S Makhanov. (2011) "Spare phase portrait analysis for preprocessing and segmentation of ultrasound images of cancer." *Internacional Journal of Computer Science*. 38. pp 146-159.

8. González, R., & Woods, R. (2007). *Digital image processing*. (Tercera ed., p. 954). New Jersey: Pearson.
9. Harandi, N, S Sadri, and N Moghaddam. (2009) "An automated method for segmentation of epithelial cervical cells in images of thinprep." *J Med Syst*. 34.2008: 1043-1058.
10. Ilea, Dana E. and Whelan, Paul F. (2006) *Color image segmentation using a spatial k-means clustering algorithm*. In: IMVIP 2006 - 10th International Machine Vision and Image Processing Conference, 30 August - 1 September 2006, Dublin, Ireland.
11. Iuliana, D. (2008) "Medical diagnosis based on image analysis and processing." *Seria Technica*. 61.2009 pp 383-388.
12. Kothari, S, Q Chaudry, and M Wang. (2009) "Automated cell counting and cluster segmentation using concavity detection and ellipse fitting techniques." *Biomedical Imaging: From Nano to Macro*. 28/abril/2009. Ed. IEEE International Symposium. Atlanta: pp 795-798.
13. Lakowicz, J. (2006). *Principles of fluorescence spectroscopy*. (Tercera ed., p. 954). Baltimore: Springer.
14. Life Technologies (2012) *Alexa Fluor® Dye-Across the Spectrum*. Recuperado el 19 de abril del 2012, de <http://es-mx.invitrogen.com/site/mx/es/home/brands/Molecular-Probes/Key-Molecular-Probes-Products/alexa-fluor/Alexa-Fluor-Dyes-Across-the-Spectrum.html>
15. MatWorks. (2010). *Image processing toolbox 7 user's guide*. (p. 1284). New York: MatWorks.
16. McAndrew, A. (2004). *Introduction to digital image processing with matlab®*. (Primera ed., p. 510). Melbourne: Thomson.

17. Murphy, D. (2001). *Fundamentals of light microscopy and electronic imaging*. (p. 638). Florida: Wiley-Liss.
18. Rosenbaum, T, R Vidaltamayo, M Sánchez, A Zentella, and M Hiriart.(1998) "Pancreatic β cells synthesize and secrete nerve growth factor." *Proc. Natl. Acad. Sci. USA*. 99.1998 pp 7784-7788.
19. Russ, J. (2002). *The image processing handbook*. (Cuarta ed.). Florida: CRC PRESS.
20. Tirodkar, A. (2011) "A multi-stage algorithm for enhanced X-ray image segmentation." *Internacional Journal of Engineering Science and Technology*. 3. (2011): pp 7056-7065.
21. University of Western Australia, P. K. (2011, Diciembre 12). *Matlab and octave functions for computer vision and image processing*. Recuperado el 3 de abril del 2012 de, <http://www.csse.uwa.edu.au/~pk/Research/MatlabFns/>
22. Vidaltamayo, R, M Sánchez, T Rosenbaum, T Martínez-Merlos, and H Marcia. (1997) "Neuron-like phenotype changes in pancreatic β -cells induced by NGF, FGF and dbcAMP." *Endocrine*. 4.: pp 19-26.

9 Anexo

9.1 Programa de conteo de imágenes

```
close all
clear all
clc
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[file,path] = uiputfile('* .tif', 'Save Workspace As');
I = imread(file);
[G BB nq] = ROI(I);
n = 0;
figure,imshow(I)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Ciclos%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
prompt1 = 'Cuántas secciones tiene la imagen';
dlg_title1 = 'Confirmación de segmentos';
options.Resize='on';
options.WindowStyle='normal';
options.Interpreter='tex';
num_lines=1;
resp1 = inputdlg(prompt1,dlg_title1,num_lines);
c1 = resp1(1);
n1 =str2double(c1);
scount = ones(n1,1);
sarea = ones(n1,1);
close all
for i=1:n1
    prompt = 'Teclee 1 si ya termino de seleccionar la máscara';
    dlg_title = 'Confirmación de máscara';
    options.Resize='on';
    options.WindowStyle='normal';
```

```

options.Interpreter='tex';
num_lines=1;
while (n == 0)
    figure,imshow(I)
    [X Y] = getpts;
    roi = roipoly(G,X,Y);
    resp = inputdlg(prompt,dlg_title,num_lines);
    c= resp(1);
    n = str2double(c);
end
f1 = fspecial('gaussian',[5 5],1.2);
%%%%%%%%Verde%%%%%%%%
rt = abs(1-roi);
bw = G-im2uint8(rt);
areass =bwarea(bw);
sarea(i,1) = areass;
J_bw = bw>100;
[muG maskG] = kmeans(J_bw,2);
[gradientG orG] = Canny(J_bw,1.4);
%figure(1),imshow(gradientG);
%%%%%%%%Azul%%%%%%%%
bwB = BB-im2uint8(rt);
[muB maskB] = kmeans(bwB,10);
[gradientB orB] = Canny(maskB,1.4);
bw_lfB = filter2(f1,gradientB);
%%%%%%%%Proceso de contornos%%%%%%%%
bw_lf = filter2(f1,gradientG);
lf_bw = im2bw(bw_lf,.15);
%figure(2),imshow(lf_bw);
area = bwareaopen(lf_bw,45,4);
%figure(3),imshow(area);
[Co,h] = contour(area);

```

```

bw_limpia = area-rt;
bw_peri = bwperim(bw_limpia);
Segout = bwB;
Segout(bw_peri) =255;
%figure(4),imshow(Segout)
e = strel('line',1,0);
bw_erode = imerode(bw_limpia,e);
[LL count] = bwlabel(bw_erode);
[B,L,N,A] = bwboundaries(bw_erode);
figure, imshow(bwB); hold on;
colors=['b' 'g' 'r' 'c' 'm' 'y' 'k'];
for k=1:length(B)
    boundary = B{k};
    cidx = mod(k,length(colors))+1;
    plot(boundary(:,2), boundary(:,1),...
        colors(cidx),'LineWidth',2);
    %randomize text position for better visibility
    rndRow = ceil(length(boundary)/(mod(rand*k,7)+1));
    col = boundary(rndRow,2); row = boundary(rndRow,1);
    h = text(col+1, row-1, num2str(LL(row,col)));
    set(h,'Color',colors(cidx),...
        'FontSize',14,'FontWeight','bold');
end
figure; spy(A);
scount(i,1) = count;
n =0;
end
count = sum(scount(:))-(n1-1);
areasT = sum(sarea(:));
disp('El área total es de:');
disp(num2str(areasT));
disp('El conteo de células son');

```

```
disp(num2str(count));
```


9.2 Programa ROI

```
function [G,B,bw] = ROI(I)
```

```
%La entrada es una imagen RGB
```

```
% La salida del método nos regresa los canales divididos del RGB
```

```
G = I(:,:,2);
```

```
B = I(:,:,3);
```

```
c = [300 900 900 300];
```

```
r = [58 58 231 231];
```

```
roi = roipoly(I,c,r);
```

```
Y = im2uint8(roi);
```

```
bb = Y-B;
```

```
bw=abs(bb-1);
```

9.3 Programa K-means

(University of Western Australia, 2011)

```
function [mu,mask]=kmeans(ima,k)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% kmeans image segmentation
%
% Input:
%   ima: grey color image
%   k: Number of classes
% Output:
%   mu: vector of class means
%   mask: clasification image mask
%
% Author: Jose Vicente Manjon Herrera
% Email: jmanjon@fis.upv.es
% Date: 27-08-2005
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% check image
ima=double(ima);
copy=ima;    % make a copy
ima=ima(:);  % vectorize ima
mi=min(ima); % deal with negative
ima=ima-mi+1; % and zero values

s=length(ima);

% create image histogram
```

```

m=max(ima)+1;
h=zeros(1,m);
hc=zeros(1,m);

for i=1:s
    if(ima(i)>0) h(ima(i))=h(ima(i))+1;end;
end
ind=find(h);
hl=length(ind);

% initiate centroids

mu=(1:k)*m/(k+1);

% start process

while(true)

    oldmu=mu;
    % current classification

    for i=1:hl
        c=abs(ind(i)-mu);
        cc=find(c==min(c));
        hc(ind(i))=cc(1);
    end

    %recalculation of means

    for i=1:k,
        a=find(hc==i);

```

```

    mu(i)=sum(a.*h(a))/sum(h(a));
end

if(mu==oldmu) break;end;

end

% calculate mask
s=size(copy);
mask=zeros(s);
for i=1:s(1),
for j=1:s(2),
    c=abs(copy(i,j)-mu);
    a=find(c==min(c));
    mask(i,j)=a(1);
end
end

mu=mu+mi-1; % recover real range

```

9.4 Programa: Detección de contorno Canny

(University of Western Australia, 2011)

```
%CANNY - Canny edge detection
%
% Function to perform Canny edge detection.
%
% Usage: [gradient or] = canny(im, sigma)
%
% Arguments: im - image to be processed
%             sigma - standard deviation of Gaussian smoothing filter
%                  (typically 1)
%
% Returns: gradient - edge strength image (gradient amplitude)
%          or - orientation image (in degrees 0-180, positive
%          anti-clockwise)
%
% See also: NONMAXSUP, HYSTHRESH, DERIVATIVE5

% Copyright (c) 1999-2010 Peter Kovesi
% Centre for Exploration Targeting
% The University of Western Australia
% http://www.csse.uwa.edu.au/~pk/research/matlabfns/
%
% Permission is hereby granted, free of charge, to any person obtaining a copy
% of this software and associated documentation files (the "Software"), to deal
% in the Software without restriction, subject to the following conditions:
%
% The above copyright notice and this permission notice shall be included in all
% copies or substantial portions of the Software.
%
% The Software is provided "as is", without warranty of any kind.
```

```
% April 1999 Original version
% January 2003 Error in calculation of d2 corrected
% August 2010 Changed to use derivatives computed using Farid and
% Simoncelli's filters. Cleaned up
```

```
function [gradient, or] = canny(im, sigma)
```

```
    assert(ndims(im) == 2, 'Image must be greyscale');
```

```
    % If needed convert im to double
```

```
    if ~strcmp(class(im), 'double')
```

```
        im = double(im);
```

```
    end
```

```
    im = gaussfilt(im, sigma);      % Smooth the image.
```

```
    [lx, ly] = derivative5(im, 'x', 'y'); % Get derivatives.
```

```
    gradient = sqrt(lx.*lx + ly.*ly); % Gradient magnitude.
```

```
    or = atan2(-ly, lx);           % Angles -pi to + pi.
```

```
    or(or<0) = or(or<0)+pi;       % Map angles to 0-pi.
```

```
    or = or*180/pi;               % Convert to degrees.
```

9.5 Programa: Filtro Gaussiano

(University of Western Australia, 2011)

```
% GAUSSFILT - Small wrapper function for convenient Gaussian filtering
%
% Usage: smim = gaussfilt(im, sigma)
%
% Arguments: im - Image to be smoothed.
%           sigma - Standard deviation of Gaussian filter.
%
% Returns: smim - Smoothed image.
%
% See also: INTEGGAUSSFILT

% Peter Kovesi
% Centre for Exploration Targeting
% The University of Western Australia
% http://www.csse.uwa.edu.au/~pk/research/matlabfns/

% March 2010

function smim = gaussfilt(im, sigma)

    assert(ndims(im) == 2, 'Image must be greyscale');

    % If needed convert im to double
    if ~strcmp(class(im), 'double')
        im = double(im);
    end

    size = ceil(6*sigma);
    if ~mod(size,2) % Ensure filter size is odd
        size = size+1;
```

```
end
size = max(size,1); % and make sure it is at least 1

h = fspecial('gaussian', [size size], sigma);

smim = filter2(h, im);
```


9.6 Programa: Derivadas

(University of Western Australia, 2011)

```
%DERIVATIVE5 - 5-Tap 1st and 2nd discrete derivatives
%
% This function computes 1st and 2nd derivatives of an image using the 5-tap
% coefficients given by Farid and Simoncelli. The results are significantly
% more accurate than MATLAB's GRADIENT function on edges that are at angles
% other than vertical or horizontal. This in turn improves gradient orientation
% estimation enormously. If you are after extreme accuracy try using
DERIVATIVE7.
%
% Usage: [gx, gy, gxx, gyy, gxy] = derivative5(im, derivative specifiers)
%
% Arguments:
%         im - Image to compute derivatives from.
%         derivative specifiers - A comma separated list of character strings
%         that can be any of 'x', 'y', 'xx', 'yy' or 'xy'
%         These can be in any order, the order of the
%         computed output arguments will match the order
%         of the derivative specifier strings.
% Returns:
% Function returns requested derivatives which can be:
%   gx, gy - 1st derivative in x and y
%   gxx, gyy - 2nd derivative in x and y
%   gxy - 1st derivative in y of 1st derivative in x
%
% Examples:
% Just compute 1st derivatives in x and y
% [gx, gy] = derivative5(im, 'x', 'y');
%
% Compute 2nd derivative in x, 1st derivative in y and 2nd derivative in y
% [gxx, gy, gyy] = derivative5(im, 'xx', 'y', 'yy')
```

```

% arguments is longer than 1 char, this implies 2nd derivative needed.
secondDeriv = false;
for n = 1:length(varargin)
    if length(varargin{n}) > 1
        secondDeriv = true;
        break
    end
end

if ~secondDeriv
    % 5 tap 1st derivative coefficients. These are optimal if you are just
    % seeking the 1st derivatives
    p = [0.037659 0.249153 0.426375 0.249153 0.037659];
    d1 = [0.109604 0.276691 0.000000 -0.276691 -0.109604];
else
    % 5-tap 2nd derivative coefficients. The associated 1st derivative
    % coefficients are not quite as optimal as the ones above but are
    % consistent with the 2nd derivative interpolator p and thus are
    % appropriate to use if you are after both 1st and 2nd derivatives.
    p = [0.030320 0.249724 0.439911 0.249724 0.030320];
    d1 = [0.104550 0.292315 0.000000 -0.292315 -0.104550];
    d2 = [0.232905 0.002668 -0.471147 0.002668 0.232905];
end

% Compute derivatives. Note that in the 1st call below MATLAB's conv2
% function performs a 1D convolution down the columns using p then a 1D
% convolution along the rows using d1. etc etc.
gx = false;

for n = 1:length(varargin)
    if strcmpi('x', varargin{n})
        varargout{n} = conv2(p, d1, im, 'same');
    end
end

```

```

    gx = true; % Record that gx is available for gxy if needed
    gxn = n;
elseif strcmpi('y', varargin{n})
    varargout{n} = conv2(d1, p, im, 'same');
elseif strcmpi('xx', varargin{n})
    varargout{n} = conv2(p, d2, im, 'same');
elseif strcmpi('yy', varargin{n})
    varargout{n} = conv2(d2, p, im, 'same');
elseif strcmpi('xy', varargin{n}) | strcmpi('yx', varargin{n})
    if gx
        varargout{n} = conv2(d1, p, varargout{gxn}, 'same');
    else
        gx = conv2(p, d1, im, 'same');
        varargout{n} = conv2(d1, p, gx, 'same');
    end
else
    error(sprintf('"%s" is an unrecognized derivative option', varargin{n}));
end
end
end

```