

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica



**Segmentación de células mediante
técnicas de Procesamiento Digital de
Imágenes para el rastreo de células
cancerosas**

Informe de Proyecto de Graduación para optar por el título de
Ingeniero en Electrónica con el grado académico de Licenciatura

David Marín Soto

Cartago, Costa Rica
Agosto 2018

INSTITUTO TECNOLÓGICO DE COSTA RICA

ESCUELA DE INGENIERÍA ELECTRÓNICA

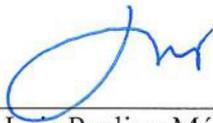
PROYECTO DE GRADUACIÓN

ACTA DE APROBACIÓN

**Defensa de Proyecto de Graduación
Requisito para optar por el título de Ingeniero en Electrónica
Grado Académico de Licenciatura
Instituto Tecnológico de Costa Rica**

El Tribunal Evaluador aprueba la defensa del proyecto de graduación denominado Segmentación de células mediante técnicas de Procesamiento Digital de Imágenes para el rastreo de células cancerosas, realizado por el señor David Marín Soto y, hace constar que cumple con las normas establecidas por la Escuela de Ingeniería Electrónica del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal Evaluador



Ing. Luis Paulino Méndez Badilla

Profesor lector



Ing. Renato Rímolo Donadio

Profesor lector



Ing. Sergio Arriola Valverde

Profesor asesor

Cartago, 17 de agosto, 2018

Declaratoria de autenticidad

Declaro que el presente proyecto de graduación ha sido realizado, en su totalidad, por mi persona y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos que he utilizado material bibliográfico, he procedido a indicar las fuentes mediante citas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.



David Marín Soto

Cédula: 1-1545-0507

Viernes 17 de agosto, 2018

Resumen

Actualmente en el área microbiológica, las metodologías utilizadas para la identificación de células cancerígenas han ido evolucionando debido a la inserción tecnológica en cuanto a equipamiento y el empleo de técnicas de procesamiento de imágenes y visión por computador.

En relación a lo anterior en Costa Rica, mediante un proyecto se ha direccionado un interés importante en llevar a cabo el análisis y rastreo de células cancerígenas, mediante la colaboración de varias instituciones y universidades públicas, entre ellas el Instituto Tecnológico de Costa Rica. No obstante, debido a la ausencia de métodos para el rastreo de células cancerígenas, se presenta este proyecto de graduación que busca generar un primer acercamiento que permita establecer una ruta de trabajo a seguir.

Finalmente, este proyecto evalúa el impacto del procesamiento de las imágenes suministradas por microbiólogos las cuales serán utilizadas por una arquitectura de aprendizaje automático, que se encarga de segmentar semánticamente las imágenes pre-procesadas para luego detectar células cancerígenas, esto mediante la utilización de técnicas de visión por computador y *deep learning*.

Palabras clave: Aprendizaje profundo, cancerígenas, células, imágenes digitales, microbiología, procesamiento, segmentación.

Abstract

In recent years, the methodologies for cancerous cells identification in the microbiological scientific field, have been improving due to technological equipment insertion and the use of digital image processing and computer vision techniques.

For such reason in Costa Rica, there is a project with important interest in cancerous cells tracking and analysis involving several institutions and public universities, including *Instituto Tecnológico de Costa Rica*. Even though, there are absence of methods for cell image recognition; therefore, this graduation project is presented to be a first approach to allow further research and become a path for microbiologists goals.

Finally, this project assess the impact of digital image processing applied as input for an automatic learning architecture, which is responsible for semantic segmentation of cancerous cells using computer vision techniques and deep learning.

Keywords: Cancerous, cells, deep learning, digital images, microbiology, processing, segmentation.

Dedicatoria

Dedicado a todas las personas que han encontrado en el estudio una forma de recreación y crecimiento personal, que han demostrado al mundo que con pasión y esfuerzo se pueden lograr grandes cosas, sin importar clases sociales.

A todos los amigos y amistades (profesores y estudiantes) forjadas a lo largo del hermoso camino de la universidad, que de una u otra manera dieron su aporte para llegar a estas instancias.

A mi sobrina Isabella quien me da razones para despertar cada día, buscar ser una mejor persona, ejemplo y apoyo para ella en su futuro desarrollo personal y profesional. A mi hermana por su gran amistad y mentoría de vida. A mi madre quien con su esfuerzo me dió la oportunidad de estudiar y crecer académicamente

Agradecimientos

Agradezco de manera profunda al grupo de investigación PARMA por haberme permitido formar parte del equipo durante más de un año. Ha sido un tiempo de valiosa formación personal y académica, donde he conocido excelentes profesionales que colaboraron para que este proyecto se llevara a cabo. Sobre todo al MSc. Saúl Calderón Ramírez, director de PARMA-Group por su mentoría en el tema del procesamiento digital de imágenes y aprendizaje profundo, temas que complementaron mi formación académica.

Se agradece a Jordina Torrents Barrena, del Departamento de Ingeniería Informática y Matemáticas de la Universidad Rovira i Virgili, de España; por compartir sus conocimientos en el área de arquitecturas de aprendizaje profundo, específicamente en la red U-Net.

También extiendo mi gratitud al señor Ángel García Pedrero, del Sustainable Forest Management Research Institute, de la Universidad de Valladolid INIA de Soria, España; por sus comentarios y recomendaciones a medida que se obtuvieron resultados y Daniel Riccio del departamento de Ingeniería Eléctrica de la Università degli Studi di Napoli Federico II, Nápoles, Italia; por su contribución en el cálculo de las métricas para precisión de segmentación.

Esta investigación contó con el apoyo de una asignación computacional en la supercomputadora Kabré del Centro Nacional de Alta Tecnología de Costa Rica.

Índice general

1. Introducción	6
1.1. Objetivos y estructura del documento	6
Introducción	7
2. Marco Teórico	8
2.1. Aspectos Biológicos	8
2.2. Procesamiento digital de imágenes	8
2.2.1. Imágenes digitales	9
2.2.2. Modelo de ruido en imágenes	9
2.2.3. Filtrado no lineal de imágenes	10
2.3. Espacio de abstracción de imágenes	11
2.3.1. Filtro de Promediado No Local (NLM)	11
2.3.2. Función Unsharpping Mask (USM)	13
2.3.3. Deceived Non Local Means (DNLM)	15
2.4. Segmentación Semántica	16
2.5. Arquitecturas de Aprendizaje Automático	17
2.5.1. Machine Learning	17
2.5.2. Redes Neuronales	19
2.6. Arquitecturas de aprendizaje profundo	20
2.6.1. Redes Neuronales Convolucionales (CNN)	20
2.6.2. Arquitecturas Deep Learning para segmentación	23
2.6.3. Fully Convolutional Networks (FCN)	25
2.6.4. Red convolucional U-Net	26
2.7. Métricas	28
2.7.1. Índice de Dice	28
2.7.2. F Measure	29
2.7.3. Distancia de Jaccard	30
3. Diseño e implementación del sistema de segmentación de células	31
3.1. Análisis de requerimientos	31
3.2. Análisis y selección de la solución	31
3.2.1. Selección de la solución	32
3.2.2. Desarrollo de la solución	34
3.3. Módulo de filtrado No lineal	35
3.4. Arquitectura de aprendizaje profundo: U-Net	38
3.4.1. Set de datos de entrenamiento para red U-Net	39
3.4.2. Red convolucional U-Net	41

4. Análisis y validación	43
4.1. Set de imágenes iniciales	43
4.2. Clúster Kabré	43
4.3. Procesamiento de imágenes con filtrado No lineal	44
4.4. Generación de groundtruth	48
4.5. Aumentado de datos	49
4.6. Entrenamiento de red U-Net	50
4.7. Validación de Indicadores	51
4.7.1. Precisión de segmentación con U-Net versus técnicas clásicas	55
4.7.2. Validación de resultados mediante pruebas de concepto . . .	56
4.7.3. Validación de resultados con treinta imágenes	59
Conclusiones	63
Recomendaciones y trabajo futuro	64

Índice de figuras

Figura 2.1.	Efecto del ruido Gaussiano sobre una imagen digital	10
Figura 2.2.	Núcleo Gaussiano y su efecto en una imagen.	12
Figura 2.3.	Aproximaciones de la primera y segunda derivada.	15
Figura 2.4.	Métodos de segmentación	17
Figura 2.5.	Modelo general del aprendizaje supervisado	18
Figura 2.6.	Imágenes correspondientes al set de datos de entrenamiento	18
Figura 2.7.	Representación básica de una red neuronal de dos capas . . .	19
Figura 2.8.	Reducción de dimensionalidad al aplicar convolución 2D . . .	21
Figura 2.9.	Mapas de activación en función de la cantidad de filtros . . .	22
Figura 2.10.	Capas de una CNN convencional	22
Figura 2.11.	Red convolucional para clasificación de imágenes de perros . .	23
Figura 2.12.	Reconocimiento de objetos con técnicas de segmentación . . .	24
Figura 2.13.	Representación de distintos métodos para segmentación . . .	25
Figura 2.14.	Salida visualizada como mapa de calor	26
Figura 2.15.	Estrategia de reconstrucción mediante reflejo de la imagen . .	26
Figura 2.16.	Traslape de células	27
Figura 2.17.	Arquitectura de U-Net	27
Figura 2.18.	Conceptos básicos para métricas	29
Figura 3.1.	Diagrama de alto nivel de la solución a implementar	35
Figura 3.2.	Diagrama de segundo nivel	35
Figura 3.3.	Diagrama de relación de módulos para algoritmo DNLM . . .	36
Figura 3.4.	Diagrama de flujo de algoritmo DNLM	37
Figura 3.5.	Diagrama de segundo nivel del módulo	38
Figura 3.6.	Segundo nivel del módulo set de datos de entrenamiento . . .	39
Figura 3.7.	Imágenes obtenidas a partir de una original	40
Figura 3.8.	Diagrama de flujo de algoritmo de U-Net	41
Figura 4.1.	Imagen original	43
Figura 4.2.	Arquitectura del clúster Kabré	44
Figura 4.3.	Disminución de ruido al aumentar h con intervalo de 2 . . .	45
Figura 4.4.	Intensidad en bordes al aumentar λ con intervalo de 3 . . .	45
Figura 4.5.	Imagen filtrada, $h = 6 \lambda = 0$	46
Figura 4.6.	Imagen filtrada, $h = 0 \lambda = 6$	47
Figura 4.7.	Imagen filtrada, $h = 6 \lambda = 6$	47
Figura 4.8.	Groundtruth: a) escala de grises, b) binario	48
Figura 4.9.	Proceso para generar Groundtruth	48
Figura 4.10.	Groundtruth binario: a) sin erosión, b) con erosión	49
Figura 4.11.	Transformaciones sobre imágenes para aumentado de datos . .	49
Figura 4.12.	Groundtruth e imágenes predecidas por la red U-Net	52
Figura 4.13.	Groundtruth e imágenes predecidas por la red U-Net	54
Figura 4.14.	Groundtruth comparado con imagen segmentada	55

Figura 4.15. Imagen de entrada del sistema y salida de módulo de filtrado	57
Figura 4.16. Imagen de entrada del sistema y salidas de módulo U-Net .	57
Figura 4.17. Imagen de entrada original y salida de módulo U-Net	58
Figura 4.18. Precisión de segmentación con métrica F measure	60
Figura 4.19. Precisión de segmentación con métrica distancia de Jaccard	61

Índice de tablas

3.1. Requerimientos primordiales del sistema	31
3.2. Pesos para parámetros de selección	33
4.1. Recursos de Hardware del GPU K40	44
4.2. Combinaciones de parámetros establecidos para experimentación . .	46
4.3. Índice de Dice para combinaciones de parámetros evaluadas	50
4.4. Índice de Dice promedio para conjuntos de parámetros seleccionados	51
4.5. Métricas para imágenes segmentadas sin filtrado	53
4.6. Métricas para imágenes segmentadas con filtrado	53
4.7. Métricas de segmentación para técnica clásica de DIP	56
4.8. Métricas para red U-Net con imágenes preprocesadas	58
4.9. Métrica F Measure para red U-Net con imágenes sin preprocesar . .	59

1. Introducción

Las imágenes han jugado un papel predominante a lo largo de la historia, se les ha utilizado para diversos fines, desde el arte hasta la documentación de sucesos importantes; y con el desarrollo tecnológico actual, permiten la implementación de nuevas aplicaciones en beneficio de la humanidad, por ejemplo, entre los campos de estudio científicos que se han beneficiado del creciente aporte del procesamiento digital de imágenes, se encuentra el sector agropecuario [1], la geografía [2], meteorología [3], biología [4], y como será el tema del presente trabajo, el campo biomédico.

En [5], se brinda un panorama en cuanto a las técnicas médicas que se benefician del uso de imágenes, tales como la resonancia magnética, tomografía por computadora, entre otras. Para el presente proyecto, se trata el caso de la investigación del cáncer, donde la segmentación precisa de células y su respectiva clasificación como cancerígenas juega un papel determinante, por ejemplo, en [6] se estudia la evolución de tumores para determinar la resistencia a medicamentos. El objetivo de los especialistas en microbiología es identificar y rastrear individualmente células con daño en su ADN debido a procesos terapéuticos, ya que su reproducción y proliferación pueden aumenatar la severidad del cáncer. Este es un aporte innovador y de alto impacto en la comunidad científica puesto que busca ser el primero en presentar resultados concluyentes en el tema a nivel mundial.

El primer paso para el rastreo, es el reconocimiento de células, el cual es una tarea que hasta la fecha se realiza por un especialista del área, consume tiempo y lleva implícita la variabilidad de la percepción humana. Por tal motivo, el presente proyecto de graduación pretende contribuir a la investigación de los microbiólogos haciendo posible una segmentación precisa automática de células que facilite un posterior rastreo.

Este proyecto se realiza como parte de la red de cooperación de CONARE, cuenta con presupuesto del Fondo Especial para la Educación Superior (FEES) e involucra la Escuela de Microbiología e Ingeniería Eléctrica de la Universidad de Costa Rica junto con PARMA-Group (PAttern Recognition and MACHine Learning Group), grupo de investigación de la Escuela de Computación del Instituto Tecnológico de Costa Rica y cuenta con el apoyo del Centro Nacional de Alta Tecnología (CeNAT).

1.1. Objetivos y estructura del documento

El presente trabajo tiene como objetivo general, implementar un algoritmo computacional que segmente de forma automática las células en imágenes digitales de microscopía por fluorescencia; en otras palabras, para una imagen digital de entrada en el sistema, su salida será otra imagen donde se destaquen las regiones

correspondientes a células.

Para lograr lo anterior se implementará un filtro no lineal mediante desarrollo de software, el cual mejora las características espaciales a partir de un conjunto de imágenes de glioblastoma. Seguidamente, se deberá implementar una técnica de aprendizaje profundo, basada en redes convolucionales para la segmentación semántica. Finalmente, se evaluará el impacto del preprocesamiento de imágenes biomédicas en la segmentación semántica mediante pruebas de concepto sobre una red convolucional, empleando indicadores, los cuales son las métricas de segmentación Medida F y distancia de Jaccard, los cuales en todos los casos, deben mejorar el índice de segmentación cuando se utiliza el preprocesamiento propuesto como parte de la solución.

El documento se estructura de manera que en el capítulo dos, se inicia con el planteamiento de los fundamentos teóricos que justifican las técnicas que se utilizan como parte del desarrollo de la solución. En el tercer capítulo se explica la solución propuesta y su proceso de diseño e implementación. Luego, en el cuarto capítulo, se validan y analizan los resultados obtenidos según el diseño planteado; por último, en el quinto capítulo se resumen las conclusiones relevantes del proyecto y recomendaciones para trabajos a futuro.

2. Marco Teórico

Este capítulo pone en contexto al lector con los fundamentos teóricos, que sustentan la aplicabilidad de cada una de las técnicas que se utilizan de forma combinada, para crear un sistema capaz de realizar segmentación semántica de forma automática, para un conjunto de imágenes de glioblastoma. A continuación se brindará un recorrido por aspectos generales microbiológicos y del procesamiento digital de señales e imágenes, que a su vez se relacionan con los conceptos básicos del aprendizaje automático, específicamente al enfoque de aprendizaje profundo.

2.1. Aspectos Biológicos

Los especialistas en microbiología pretenden estudiar redes de proliferación de células en respuesta a quimioterapia genotóxica. Los agentes genotóxicos son aquellos capaces de dañar el ADN de un organismo vivo, pueden ser generados tanto por el mismo sistema (origen endógeno) como por agentes externos (origen exógeno). En cuanto al origen exógeno, uno de los principales causantes de daño son la gran cantidad de quimioterapéuticos que se utilizan al brindar tratamiento contra el cáncer y la radiación que se recibe en clínicas y hospitales mediante técnicas terapéuticas y de diagnóstico [7].

Muchas de las células que sobreviven a procesos de quimioterapia se reproducen con rapidez y se propagan en el organismo, lo cual puede causar la transformación maligna de células no tumorales e incluso favorecen la evolución del cáncer a versiones más agresivas que desarrollan resistencia a tratamientos, esto se conoce como multiresistencia. Una vez que una célula desarrolle multiresistencia, las opciones de terapia para el paciente que las porta pasan de ser de carácter curativo a paliativo, y le dará poca esperanza de vida. Basta que un pequeño porcentaje de células sobrevivientes a procesos de daño se reproduzcan, para que se forme un nuevo tumor, por lo cual se resalta la importancia de conocer esta población de células dañadas y su proliferación [8].

El tipo de cáncer que se considera en este proyecto se denomina glioblastoma, el cual es el tumor cerebral más agresivo y maligno en adultos, cuyo tratamiento requiere radiación y quimioterapia [9]. Por lo tanto, el sistema que se implementará en este trabajo servirá para la identificación de este tipo de células cancerígenas cerebrales.

2.2. Procesamiento digital de imágenes

Para realizar la segmentación de células, se recurre al procesamiento digital de imágenes, el cual puede entenderse como una extracción de información de interés en una imagen [10]. Para adentrarse en las técnicas que se utilizan en secciones

posteriores, es de importancia realizar un repaso de conceptos básicos y claves que se aplican a lo largo del documento.

2.2.1. Imágenes digitales

Es posible definir una imagen como una función $F(x, y)$ con dos variables dependientes, pertenecientes a un mismo plano, que representan coordenadas espaciales. La amplitud de dicha función se conoce como *Intensidad*, o escala de grises de la imagen. Si a esta definición se le agrega a las coordenadas espaciales X , Y y la *Intensidad* las características de ser finitas, se llega al concepto de *Imagen Digital* [10].

Las imágenes suelen representarse en notación matricial, para una imagen de dimensiones $M \times N$ se tiene:

$$f(x, y) = \begin{bmatrix} f(0, 0) & f(0, 1) & \cdots & f(0, N - 1) \\ f(1, 0) & f(1, 1) & \cdots & f(1, N - 1) \\ \vdots & \vdots & \cdots & \vdots \\ f(M - 1, 0) & f(M - 1, 1) & \cdots & f(M - 1, N - 1) \end{bmatrix}$$

Donde a cada elemento del arreglo se le denomina píxel, el cual es el término más utilizado para referirse a cada ubicación espacial dentro del dominio de una imagen.

2.2.2. Modelo de ruido en imágenes

En [11] se define el ruido como una distorsión aleatoria en una imagen digital, la cual es inevitable y puede deberse a diversos factores presentes en el proceso de formación y adquisición de una imagen. El tipo de ruido más común y abundante es denominado *ruido Gaussiano* (GN, por sus siglas del inglés), debido a que el histograma relacionado sigue la forma de campana de Gauss. La figura 2.1 muestra el efecto de este tipo de ruido sobre una imagen digital. El GN se puede modelar matemáticamente mediante la ecuación 2.1.

$$U = X + V \tag{2.1}$$

En la cual, la imagen con ruido U es equivalente a la adición de la componente de ruido V sobre la imagen original X , donde el filtrado del de este tipo de ruido se fundamenta en métodos en el dominio espacial de la imagen.

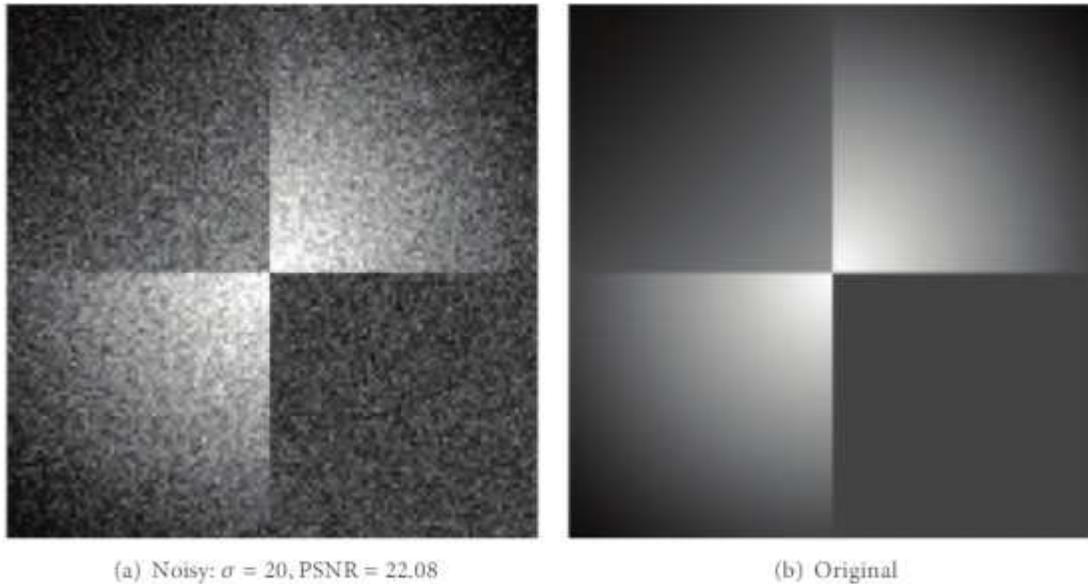


Figura 2.1: Efecto del ruido Gaussiano sobre una imagen digital tomado de [12], σ representa el ancho de la campana de Gauss, y PSNR la relación señal a ruido

2.2.3. Filtrado no lineal de imágenes

La explicación para concepto de filtrado involucra la interacción de tres elementos principales: un vecindario, una operación aritmética y un kernel. Se entiende por vecindario de un píxel, a aquellos puntos espaciales dentro de una imagen, que limitan en las partes laterales, superiores, inferiores y diagonales del píxel central; usualmente se utilizan ventanas de 3×3 para definir el vecindario de un píxel. En cuanto al término kernel, puede relacionarse a una matriz de coeficientes del mismo tamaño del vecindario los cuales definen el comportamiento del tipo de filtrado (paso alto, paso bajo, entre otros). Por lo tanto, el filtrado calcula nuevos valores para los píxeles del vecindario relacionando sus coeficientes con los del kernel, mediante una operación aritmética [10].

El agregar el término *no lineal*, tiene dos implicaciones principales, la primera es que su expresión matemática no cumple el principio de linealidad y la segunda es que se logran efectos más sofisticados que en los filtros lineales, tales como evitar la degradación de bordes en filtros paso bajos o que se amplifique el ruido cuando se aplica un filtrado de tipo paso alto [13].

A continuación se presentan los filtros en el dominio espacial relacionados al trabajo, esto quiere decir que el filtrado se realiza directamente sobre los píxeles de la imagen, las cuales pueden ser sobre un solo píxel o un vecindario, incluso transformaciones geométricas [10].

2.3. Espacio de abstracción de imágenes

Se puede definir *abstracción de imágenes* como la reducción de características irrelevantes, que debido al contexto donde se encuentran, no son necesarias y conllevan a una simplificación de la información en la imagen [14].

La primera aproximación para este concepto se presenta en [15] donde se introduce el modelo denominado WAF (*Weighted Average Filter*) mediante la ecuación 2.2. La cual proporciona una imagen de salida con ruido disminuido para una entrada original U , promediando los pesos para los píxeles p pertenecientes a una subregión Ω , y los píxeles q pertenecientes a un plano que equivale a la región espacial total de la imagen; además, la función ψ_{YL} se conoce como el filtro de Yarolasvsky.

$$\hat{X}(p) = \left(\sum_{q \in \Omega} \psi_{YL}(U, p, q) \right)^{-1} \left(\sum_{q \in \Omega} \psi_{YL}(U, p, q) U(p) \right) \quad (2.2)$$

Se han propuesto nuevos enfoques para mejorar la eliminación de ruido y preservación de bordes, manteniendo el modelo de [15] descrito en la ecuación 2.2, variando el filtro promediador de pesos de las regiones y subregiones de la imagen. En [16] se analizan los filtros:

- Filtro Bilateral, incluyendo algunas variantes.
- Filtro Non Local Means

Para el caso del filtro bilateral (FB), se ha documentado que no tiene buen desempeño cuando la componente de ruido gaussiano se incrementa. Además en [16] se estudian variaciones para FB tales como filtro bilateral escalado y filtro bilateral cruzado. Aún así, en términos de rendimiento y tolerancia a ruido en el dominio espacial, es el filtro de Promediado No Local (NLM) el que presenta un mejor comportamiento.

Otra aproximación para la abstracción de imágenes se plantea en [16] con lo que se denomina **Deceived Weighted Averaging Filter Framework** (DeWAFF, por su abreviación del inglés), el cual incorpora los efectos de definición de bordes y disminución de ruido. Para esto, hace uso de dos métodos principales: el filtro de promediado no local (NLM) y la función unsharpping mask.

2.3.1. Filtro de Promediado No Local (NLM)

La mayoría de filtros supresores de ruido en imágenes, logran este efecto mediante el promediado de píxeles, sin embargo, el filtro de promediado no local presenta un rasgo característico, al darle importancia no sólo a la similitud en la intensidad de los píxeles, sino también a la similitud de los vecindarios involucrados [17]. Así se utiliza el contexto espacial del píxel para determinar la similitud

entre píxeles en función de su vecindario, y se evita la pérdida de información que pueden causar otros filtros cuando abunda la presencia de ruido.

Si se representa v como una imagen con ruido, se puede obtener su valor promedio de acuerdo a la ecuación 2.3. Donde $w(i, j)$ representa los pesos (valor probabilístico, dado que $0 \leq w(i, j) \leq 1$ y $\sum_j w(i, j) = 1$) en función de la similitud de la escala de grises de los píxeles i y j dentro de los vectores $v(N_i)$ y $v(N_j)$ [17], el subíndice de estos vectores significa que el vector está centrado en dicho píxel, y además involucra el recuadro de dimensiones $w \times w$.

$$NLM[v](i) = \sum_{j \in I} w(i, j)v(j), \quad (2.3)$$

Debe existir alguna métrica para poder determinar la similitud, para este algoritmo se utiliza la distancia Euclideana, la cual se describe en la ecuación 2.4, donde el subíndice a representa la desviación estándar de un núcleo gaussiano, cuyo comportamiento se describe en la figura 2.2, donde se muestra el suavizado de una imagen y causa un efecto de desenfoque.

$$\|v(N_i) - v(N_j)\|_{2,a}^2, \quad (2.4)$$

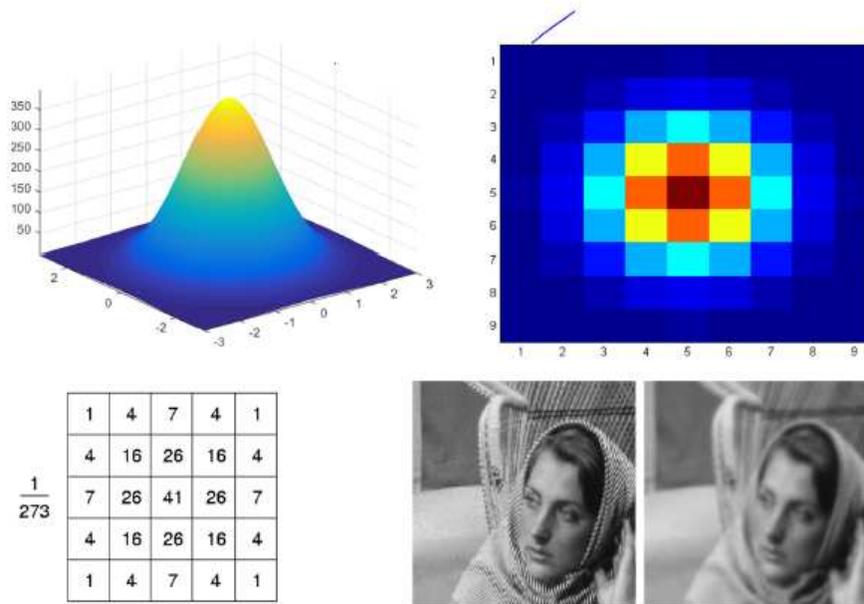


Figura 2.2: Núcleo Gaussiano y su efecto en una imagen. Representación de la función de densidad Gaussiana desde distintas perspectivas y el resultado de convolucionar una imagen contra un núcleo Gaussiano, tomado de [13].

Los pesos, o valores que se relacionan a un píxel y sirven para determinar su similitud con respecto a otros píxeles y vecindarios, se pueden calcular mediante la ecuación 2.5, donde el término $Z(i)$ representa una constante de normalización y la variable h es el factor de suavizado del filtro según la ecuación 2.6.

$$w(i, j) = \frac{1}{Z(i)} e^{-\frac{\|v(N_i) - v(N_j)\|_{2,\alpha}^2}{h^2}}, \quad (2.5)$$

$$Z(i) = \sum_j e^{-\frac{\|v(N_i) - v(N_j)\|_{2,\alpha}^2}{h^2}}, \quad (2.6)$$

Si se representa los vectores $v(N_{i,j})$, el cual está centrado sobre un píxel i o j y se involucran sus vecindarios de tamaño $w \times w$, mediante su descomposición en componentes $\vec{v}(w, i)$, $\vec{v}(w, j)$ se puede caracterizar el filtro de Promediado No Local (NLM) según la ecuación 2.7, donde h representa el factor de suavizado del filtro.

$$\psi_{NLM(i,j)} = \exp\left(-\frac{\|\vec{v}(w, j) - \vec{v}(w, i)\|^2}{h}\right), \quad (2.7)$$

2.3.2. Función Unsharpping Mask (USM)

La traducción al español del término *Unsharpping Mask* corresponde a máscara de desenfoque. Esta es una función que a través del modelo de difuminación, permite realizar mejora de contraste y detección de bordes.

En [16] se define contraste como el radio de intensidad que hace posible la distinción de los píxeles correspondientes a *foreground* y *background*, por su parte, el mejoramiento de bordes hace referencia a la pendiente de las secciones correspondientes a bordes de la imagen. *Foreground* hace referencia a los píxeles que se destacan de una imagen y *background* aquellos que quedan al fondo sin información de importancia.

El efecto perceptual de desenfoque se puede presentar por el movimiento de algún objeto o de la cámara en el proceso de formación de una imagen, se puede modelar mediante la operación de convolución de una imagen contra un filtro, conocido como *Point Spread Function (PSF)* [18]. La ecuación 2.8 se ajusta a la descripción antes planteada, una imagen desenfocada B corresponde a convolucionar el PSF representado mediante H contra la imagen de entrada I , además el parámetro ϕ sugiere la presencia de ruido aditivo en la imagen.

$$B = H * I + \phi, \quad (2.8)$$

De forma conocida, por la propiedad de linealidad, la ecuación 2.8 se puede representar en el dominio de la frecuencia, de la siguiente manera:

$$B_f = H_f I_f + \phi_f, \quad (2.9)$$

Existen dos vertientes para clasificar técnicas en PSF: los métodos explícitos e implícitos. Dentro de la rama de los métodos implícitos se halla la *función Unsharpping Mask*. En el dominio de la frecuencia, se encuentra la mayor densidad de energía en las discontinuidades, este es un principio que utilizan los métodos detectores de contornos. La función Unsharpping Mask aprovecha este factor para detectar los bordes de las imágenes y amplificar su intensidad de acuerdo a un parámetro de ganancia. Según el propósito de este trabajo, se trabaja en el dominio de la frecuencia donde, la función USM (F_{USM}) se representa en la ecuación 2.10, donde el término Υ se descompone en la ecuación 2.11.

$$F_{USM} = G + \lambda \cdot \Upsilon, \quad (2.10)$$

$$\Upsilon = l * G, \quad (2.11)$$

La interpretación de las dos ecuaciones anteriores, es que a una imagen G se le añade el aporte de un filtro paso alto para detección de bordes, en este caso, el filtro Laplaciano, expresado mediante el término Υ .

La variable λ es un coeficiente de ganancia que permite ajustar el nivel de energía que se le otorga a los bordes, y por último el término l equivale a una máscara laplaciana.

Una máscara Laplaciana es utilizada en el procesamiento digital de imágenes para obtener una aproximación a la segunda derivada de una función, dando como resultado la detección de bordes. En el dominio espectral de una función, la mayor energía según la transformada de Fourier se concentra en las discontinuidades.

En relación con lo anterior, en la figura 2.3 se muestra como la primera derivada aproxima un cambio de intensidad de la imagen, tanto de un tono oscuro a uno claro y viceversa. La segunda derivada permite identificar la discontinuidad, sin darle tanto peso al cambio en la tonalidad de la imagen.

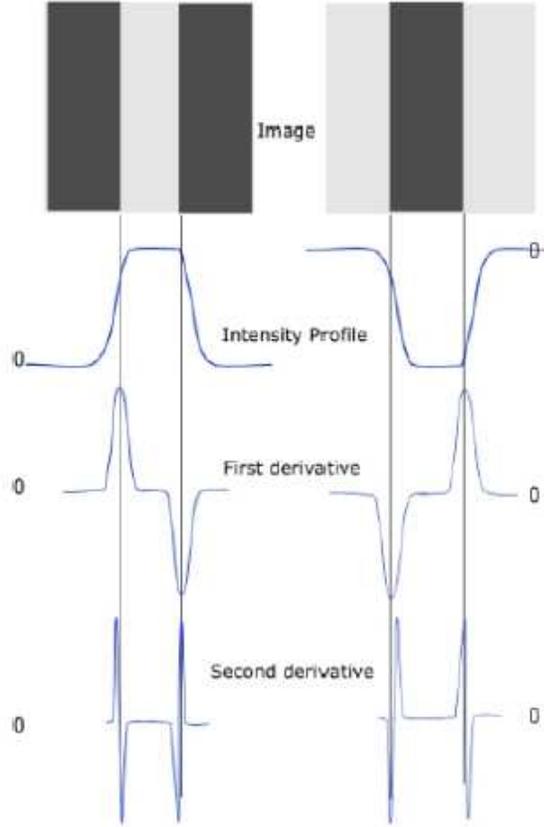


Figura 2.3: Aproximaciones de la primera y segunda derivada.
Tomado de [13].

2.3.3. Deceived Non Local Means (DNLM)

El término *Deceived Non Local Means* es el nombre que recibe el filtro de promediado no local dentro del espacio de abstracción de imágenes *Deceived Weighted Average Filter Framework*. El cual, junto con el aporte de la función Unsharpening Mask, permite realizar disminución de ruido, mejora de contraste y demarcación de bordes; con base a lo anterior, la ecuación 2.12 describe matemáticamente el filtro.

$$Y(p) = \left(\sum_{m \in \Omega} \psi_{\text{NLM}}(U, p, m) \right)^{-1} \left(\sum_{m \in \Omega} \psi_{\text{NLM}}(U, p, m) F_{\text{USM}} \right), \quad (2.12)$$

$$\psi_{\text{NLM}}(p, m) = \exp \left(- \frac{\| \vec{v}(\omega_a, m) - \vec{v}(\omega_a, p) \|^2}{h} \right), \quad (2.13)$$

2.4. Segmentación Semántica

En [10] se define el término segmentación como la división de una imagen en sus regiones u objetos, en la cual el nivel de detalle dependerá de la aplicación donde se utilice y una vez que se haya logrado identificar una región de interés, la segmentación se considera concluida. Las expresiones matemáticas que definen este concepto se expresan a continuación:

$$\bigcup_{i=1}^n R_i = I \quad (2.14)$$

En la ecuación 2.14 se describe que la unión de todas las regiones segmentadas dan como resultado el espacio de la imagen original.

$$Q(R_i) = True \quad (2.15)$$

La condición presentada en la ecuación 2.15, trata acerca de la condición que los píxeles agrupados dentro una región R_i deben cumplir para tener en común al descriptor $Q(\cdot)$, el cual podría ser, por ejemplo, la intensidad o alguna otra propiedad.

$$R_i \cap R_j = \emptyset \quad (2.16)$$

$$Q(R_i \cup R_j) = False \quad (2.17)$$

Las condiciones de las ecuaciones 2.16 y 2.17 denotan que no existe intersección ni unión para regiones que son segmentadas bajo distintos criterios. Dicho en otras palabras, distintos criterios de segmentación aplicados en una misma imagen, dan como resultado objetos de interés que pertenecen a clases diferentes.

En cuanto a las técnicas para segmentación, se tienen dos grandes grupos, segmentación basada en capa y en bloque. La segmentación en bloque aprovecha las características espaciales de la imagen, tales como bordes, límites y texturas. Según la revisión de métodos planteada en [19], se encuentran dos subgrupos basados en el enfoque de segmentación por bloque: métodos basados en región y aquellos basados en bordes y límites. Para el primer subgrupo, la propiedad de la imagen que se busca es la discontinuidad y para el segundo se busca la similitud de píxeles. Si se toma en cuenta la similitud como criterio para segmentación, se pueden derivar métodos implementados mediante algoritmos manuales (*handcrafted*) o técnicas que se realizan mediante aproximaciones computacionales, entre los cuales, se tienen las redes neuronales. En la figura 2.4 se muestra la distribución jerárquica descrita anteriormente.

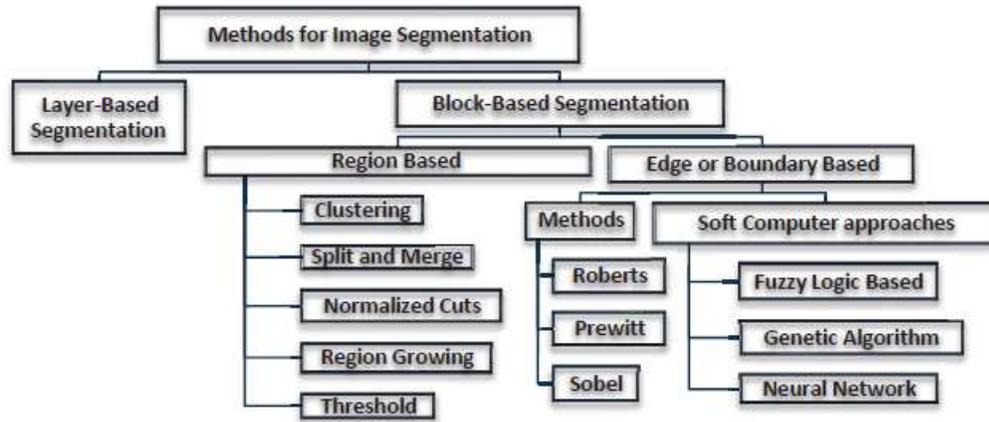


Figura 2.4: Métodos de segmentación tomado de [19]

2.5. Arquitecturas de Aprendizaje Automático

2.5.1. Machine Learning

No hay consenso en cuanto a una definición formal para el aprendizaje de máquina o *Machine Learning (ML)* como se le conoce a esta rama de las ciencias de la computación que se ha convertido en tendencia mundial, con crecimiento marcado y cada vez más impacto científico. Una aproximación para su definición serían máquinas que aprenden a realizar una tarea específica sin haber sido programada explícitamente para ello.

Dentro del aprendizaje automático se tienen dos enfoques principales, en los cuales se pueden clasificar las técnicas:

- Aprendizaje supervisado.
- Aprendizaje no supervisado.

Cuando se tiene poco o nula claridad de cómo se comportará la salida de un sistema que se desea aproximar, o no se tienen etiquetas de las muestras, se puede recurrir al aprendizaje no supervisado, en el cual destacan algunas técnicas como *clustering* y *non-clustering*.

En el aprendizaje supervisado, se cuenta con un conjunto de datos debidamente etiquetados, el cual relaciona cada muestra de entrada con su respectiva salida esperada; de esta forma la máquina aprende a predecir muestras futuras.

El proceso de aprendizaje se denomina entrenamiento y en la figura 2.5 se muestra un diagrama de alto nivel que describe el mecanismo del aprendizaje supervisado.

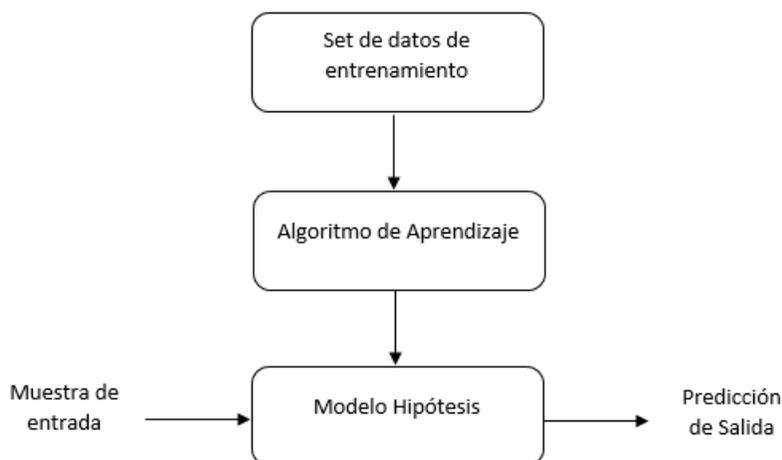


Figura 2.5: Modelo general del aprendizaje supervisado

Con base a la figura anterior, el set de datos de entrenamiento, es analizado por un algoritmo de aprendizaje el cual sirve para crear un modelo hipótesis, que proporciona una predicción para cada muestra de entrada del sistema.

Usualmente, se utiliza aprendizaje supervisado (SL) para **regresión** y **clasificación**. Para regresión se busca una salida continua, es decir, una predicción dentro de los números reales como salida de una función cuando se le aplica una entrada determinada. En el caso de la clasificación, se obtiene una salida discreta, que indica la pertenencia de la entrada del sistema a una clase establecida por el usuario; no obstante, también es posible realizar **segmentación**.

El set de datos de entrenamiento tiene dos componentes: Imagen representativa de la aplicación por trabajar y el **Groundtruth**. Por Groundtruth se entiende, la salida esperada que tendría el sistema para cada una de las imágenes representativas que se usan en el entrenamiento. La figura 2.6 brinda una explicación gráfica.

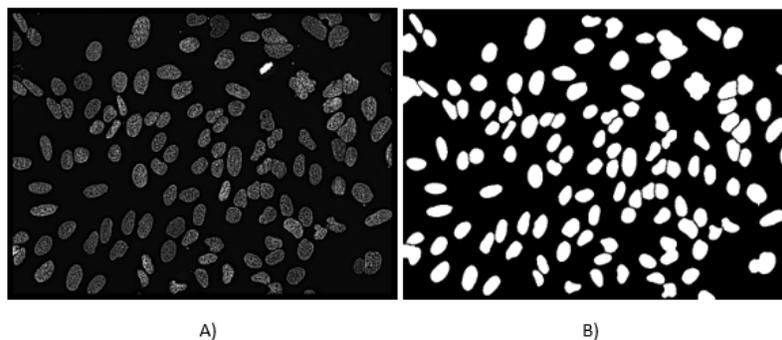


Figura 2.6: Imágenes correspondientes al set de datos de entrenamiento
 a) imagen representativa de células, b) groundtruth de segmentación de células

2.5.2. Redes Neuronales

El término *Redes Neuronales (Neural Networks, NN)*, se maneja desde la década de los cuarenta, cuando diferentes investigadores buscaron representar modelos biológicos de forma matemática [20]. Aún así, con el aumento de la capacidad computacional que se tiene en la actualidad, las redes neuronales se han convertido en el estado del arte en numerosas aplicaciones. Podría definirse una red neuronal como un número definido de funciones básicas con parámetros personalizables que pueden cambiar durante el entrenamiento.

Conforme se avanza hacia las técnicas que se utilizan para el diseño de la solución, deben cubrirse los conceptos básicos de redes neuronales que se utilizarán.

Los modelos que se construyen mediante redes neuronales están conformados por combinaciones lineales de funciones ϕ_j , que se relacionan con un vector de características \vec{x} para las cuales se evalúa el modelo, y conllevan un vector \vec{w} de pesos (coeficientes que acompañan las funciones ϕ_j). La ecuación 2.18 lo describe de forma matemática, siendo la función f la *función de activación* y varía según la aplicación (regresión, clasificación o segmentación).

$$y(\vec{x}, \vec{w}) = f\left(\sum_{j=0}^M w_j \phi_j(\vec{x})\right), \quad (2.18)$$

Se dice que una red neuronal es un grafo de tres capas principales: capa de entrada con x_D nodos, capa oculta con z_M y capa de salida con y_K nodos. En la figura 2.7 se muestra un diagrama de la representación habitual de una red convolucional de dos capas.

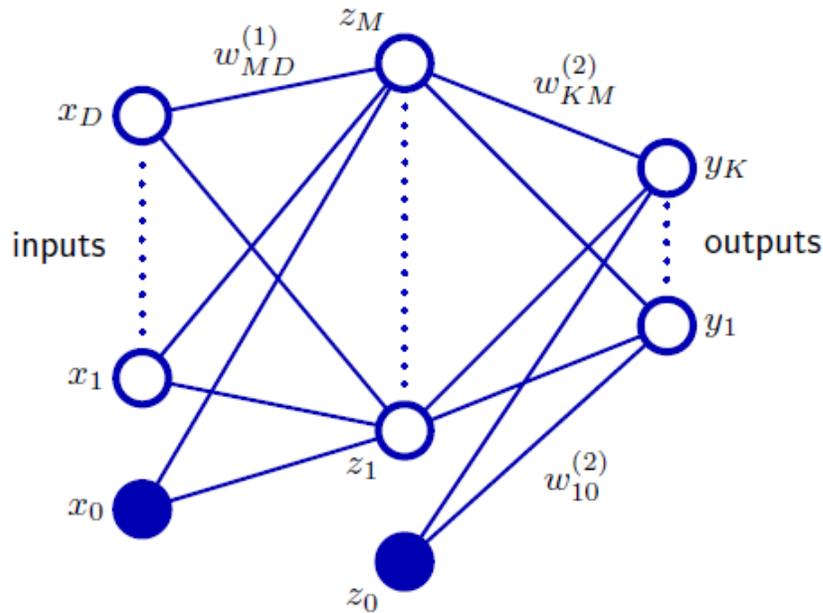


Figura 2.7: Representación básica de una red neuronal de dos capas tomado de [20]

Es importante notar que la red se compone por dos matrices de pesos W^o que une la capa de entrada con la oculta y W^s relaciona la capa oculta con la de salida. Una expresión que unifica ambas matrices se denomina **peso neto** o **coeficiente de activación**, p_m^o , la cual se describe en la ecuación 2.19.

$$p_m^o(\vec{x}, W^o) = \sum_{d=1}^D W_{d,m}^o x_d + W_{0,m}^o, \quad (2.19)$$

Para poner en términos de una sola ecuación que involucre todas las capas del grafo, se presenta la ecuación 2.20. Donde la función g^o es la función de activación no lineal y diferenciable de la capa oculta.

$$y_m^o(\vec{x}, W^o) = g^o(p_m^o(\vec{x}, W^o)), \quad (2.20)$$

El entrenamiento tiene como objetivo iterar hasta que el vector de pesos logre optimizar una función de costo, la cual es variable dependiendo de la aplicación, algunos ejemplos son: error cuadrático medio, descenso de gradiente, o para el caso de segmentación, índice de Dice.

Una red neuronal de dos capas como la que se describió, corresponde a una técnica de aprendizaje de máquina, conforme se avanza en el número de capas ocultas, se aproxima a la concepción del aprendizaje profundo o deep learning.

2.6. Arquitecturas de aprendizaje profundo

En [21] se explica el concepto de aprendizaje de máquina e inteligencia artificial como sistemas computacionales que aprenden de las experiencias y son capaces de entender el mundo de forma jerárquica. Donde cada objeto que vemos y forma parte de nuestro entorno, se relaciona con variables más simples que lo componen; si se hiciese un grafo de esa jerarquía, este poseería muchos niveles de profundidad, por lo cual a estos paradigmas se les denomina *Deep learning*.

2.6.1. Redes Neuronales Convolucionales (CNN)

Las imágenes naturales tienen la propiedad de poseer características estadísticas que son robustas ante traslaciones, según [21], por ejemplo, una imagen de un gato seguirá siendo la misma aunque se desplace un píxel o se rote.

Las redes convolucionales sacan el mayor provecho de la propiedad anterior. Por lo tanto, en el contexto biomédico, si se entrena una red para segmentar células similares a las de glioblastoma con alta certeza, por concepto, la red será capaz de predecir acertadamente la ubicación de las células de las imágenes de glioblastoma cuando estas se alimenten a la red.

Se puede definir las CNN como un tipo especial de red neuronal, de múltiples capas, que utiliza el operador de convolución, la cual muestrea periódicamente una imagen obteniendo mapas de características de la imagen en dos dimensiones.

La convolución de una función $x(t)$ con otra $w(t)$ suele representarse con el operador $*$, según la ecuación 2.21.

$$s(t) = x(t) * w(t), \quad (2.21)$$

Es importante recordar que la convolución es un operador lineal, el término x se conoce como entrada, a w se le conoce como *kernel*, cuya traducción al español podría ser núcleo o filtro, y la salida s se denomina **mapa de características**. En imágenes, las aplicaciones de aprendizaje profundo, la entrada y el kernel suelen ser arreglos multidimensionales, donde los pesos del segundo, son variados en el proceso de entrenamiento. La convolución en 2-D, se asocia con el concepto de ventana deslizante, donde la cantidad de píxeles que se desplaza la ventana (en el eje x y y) se denomina *stride*.

En estas redes, la importancia central de la convolución está en la propiedad de reducción de resolución, y extracción de características tales como patrones, bordes, figuras geométricas y más. En la figura 2.8, se ilustra la reducción de la resolución de una imagen de dimensiones de 4×3 aplicando el operador de convolución contra un kernel de 2×2 . se logra reducir la resolución de la imagen de entrada a un mapa de características de 3×2 , valor dependiente del *stride*.

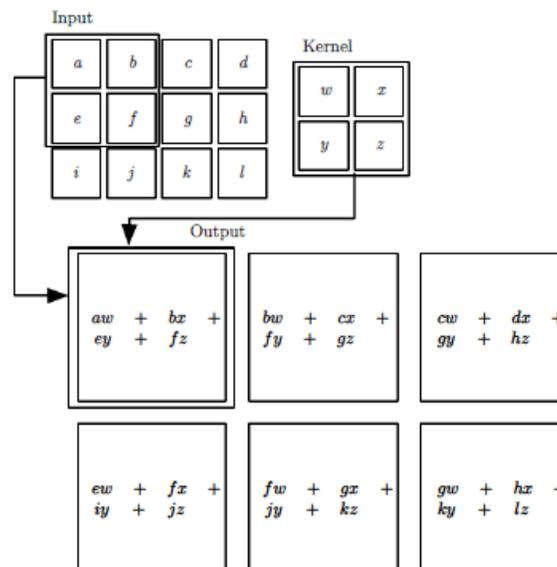


Figura 2.8: Reducción de dimensionalidad al aplicar convolución 2D a una entrada de 4×3 contra un kernel de 2×2 y un $\text{stride} = 1$ tomado de [21]

Si se incrementan la cantidad de canales o filtros, se pueden extraer más características, tal como se muestra en la figura 2.9.

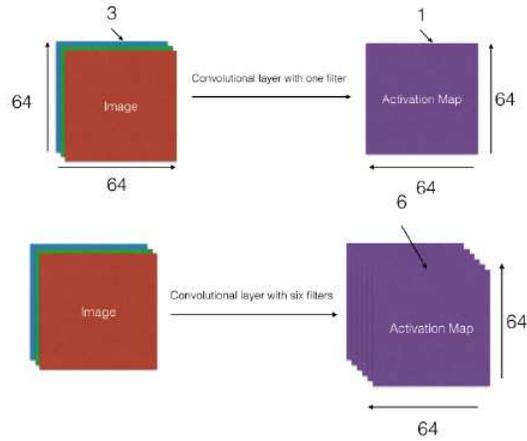


Figura 2.9: Mapas de activación en función de la cantidad de filtros tomado de [21]

A las matrices resultantes de la operación de convolución se les denomina capas de reducción, o **ReLU** como se les encuentra con frecuencia en la literatura. Esto indica que aplicar convoluciones sobre los mapas de características, o mapas de activación, se reduce la resolución de la imagen con el objetivo de extraer características. Otra capa utilizada en las redes convolucionales convencionales es la capa de *pooling*, o también denominada, *capa de submuestreo*. En la figura 2.10 se muestran las capas de una CNN.

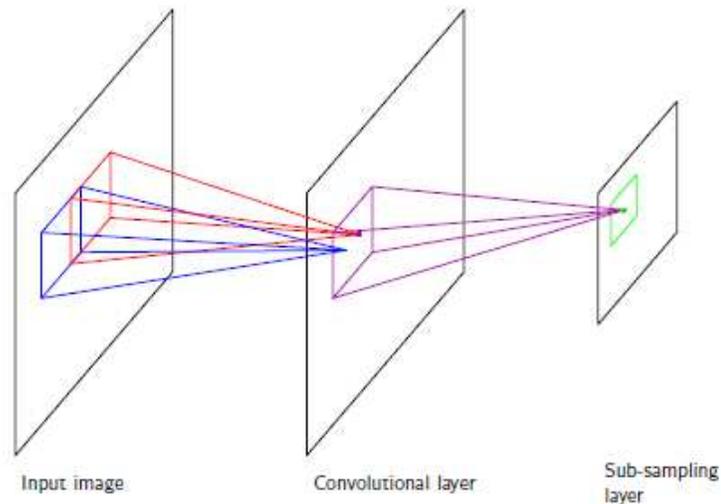


Figura 2.10: Capas de una CNN convencional tomado de [20]

En relación a la figura 2.10, la capa de submuestreo, usualmente es una función que toma ventanas de tamaño $n \times n$ y realizan una operación entre los valores de dicha imagen, entre ellas pueden estar: obtención del promedio de los valores de la

ventana, el máximo, entre otras. Con esto se logra reducir aún más las dimensiones de las características de las imágenes que se obtienen en los mapas de activación. Para completar la descripción de una arquitectura tradicional completa de una red convolucional, hace falta hablar de la capa denominada **Fully Connected Network** o, en español, **capa completamente conectada**, la cual es especialmente útil para problemas de clasificación, debido a que cada nodo de esta capa correspondería a cada una de las clases en las que se podrían clasificar las características presentes en la imagen de entrada de la red. Esta capa, usualmente es una red neuronal sencilla de dos capas, como la que se explicó en la sección anterior.

En la figura 2.11 se muestra un diagrama completo de una CNN para clasificación, donde las capas que se muestran en color celeste representan capas ReLu, obtenidas de aplicar convoluciones con kernels de tamaño 3x3 que se aprenden automáticamente en el proceso de entrenamiento. Las capas representadas en color morado indican los mapas de activación (o de características) correspondientes a la capa de submuestreo o pooling. La dinámica se repite para otra unidad oculta, donde se puede apreciar la reducción de resolución de la imagen de entrada, que pasó de ser de 64x64 píxeles a 16x16 con distintos mapas de activación que poseen las características de la imagen. Por último en la capa completamente conectada, se da el proceso de clasificación binaria, donde se decide si la imagen de entrada corresponde a un perro o no.

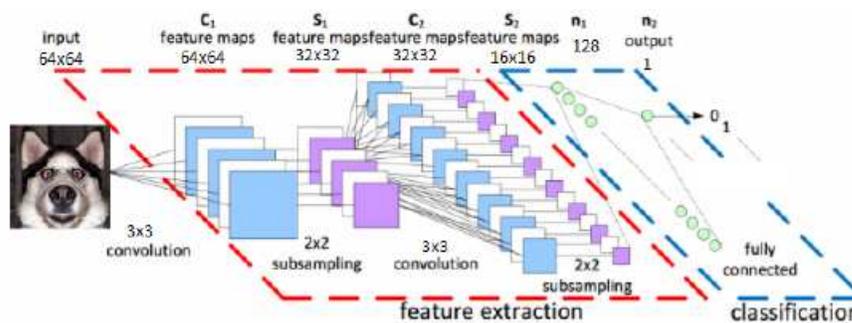


Figura 2.11: Red convolucional para clasificación de imágenes de perros tomado de [13]

2.6.2. Arquitecturas Deep Learning para segmentación

Los usos habituales de las CNN convencionales son la regresión y clasificación, sin embargo, es posible utilizar los enfoques de aprendizaje profundo para realizar aplicaciones tales como la segmentación. En las tareas de clasificación, se pretende asociar un objeto a una clase determinada, mediante un identificador binario su pertenencia o no al conjunto multiclase existente. La segmentación tiene sus bases en la clasificación, pero va más allá puesto que involucra localización y detección de cada píxel en cuanto a sus propiedades espaciales [22].

Existen dos vertientes en el tema de segmentación con similitudes pero rasgos que los diferencia, se tiene la segmentación semántica y de instancias. La figura 2.12 proporciona una panorámica clara acerca de las diferencias entre ambas. La segmentación semántica brinda etiquetas iguales para los elementos de una clase, y la de instancias brinda etiquetas individuales para elementos que pertenecen a una misma clase. Nótese el ejemplo con los cubos de la figura 2.12.

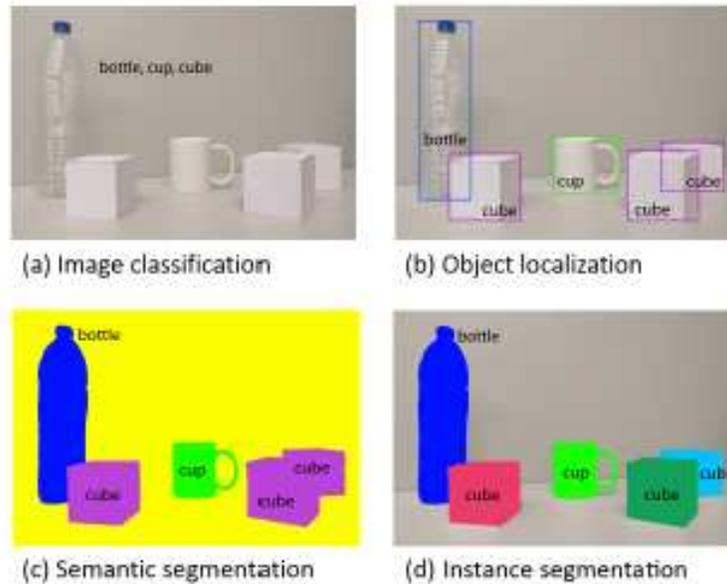


Figura 2.12: Reconocimiento de objetos con técnicas de segmentación tomado de [22]

Las arquitecturas de aprendizaje profundo para segmentación, utilizan como base una red para clasificación, sin embargo, usualmente, reemplazan la *Fully Connected Network* y la sustituyen por otra red que utiliza técnicas para reconstruir la imagen que había sido submuestreada en reiteradas ocasiones y reducida a mapas de características de baja resolución. A la parte de la red que realiza submuestreo, se le denomina *encoder* y la que realiza un *upsampling* o reconstrucción, se llama *decoder*. Un tipo de *decoder* que según [22] ha dado los mejores resultados del estado del arte, es una red convolucional nuevamente, lo que sustenta el nombre de la arquitectura completa (encoder más decoder) el cual se encuentra en la literatura como *Fully Convolutional Network*.

El tema de segmentación es muy amplio, por lo que se recomienda al lector referirse a [22] para ver con detenimiento los distintos enfoques tanto clásicos como de aprendizaje automático y profundo. Para el contexto actual, la figura 2.13 resalta los métodos del estado del arte en los que se desempeña este trabajo.

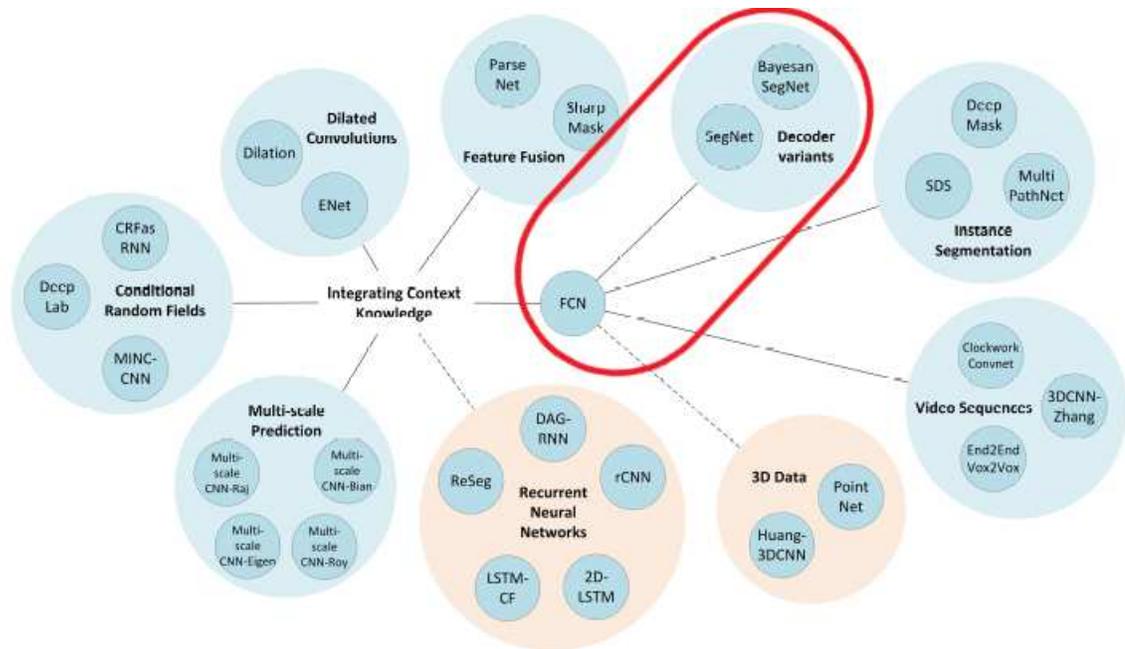


Figura 2.13: Representación de distintos métodos para segmentación tomado de [22]

2.6.3. Fully Convolutional Networks (FCN)

La segmentación semántica responde a la pregunta de qué representa un píxel, para ello las redes completamente convolucionales aprenden y se entrenan utilizando la información de toda la imagen. En las redes comunes para clasificación, se implementaba una función no lineal, en las FCN se implementa un filtro no lineal que se le conoce como *filtro profundo* [23]. Un rasgo característico de este tipo de redes, es que pueden trabajar con imágenes de cualquier tamaño.

El simple hecho de sustituir la capa completamente conectada de una red tradicional por capas convolucionales, permite transformar la señal de salida de una etiqueta que indica la pertenencia de un objeto a una clase a una imagen de mapa de características. Las FCN se les agregan convoluciones con *stride* las cuales se denominan **deconvoluciones** las cuales pueden realizar predicciones densas para reconstruir y etiquetar cada píxel de una imagen. En la figura 2.14 se muestran los píxeles que corresponden a una clase particular.

Esta red presenta algunas desventajas, las cuales son: No se toma en cuenta información útil del contexto de la imagen, le falta mucha optimización para poder ejecutarse en tiempo real y no está pensada para modelos en tres dimensiones.

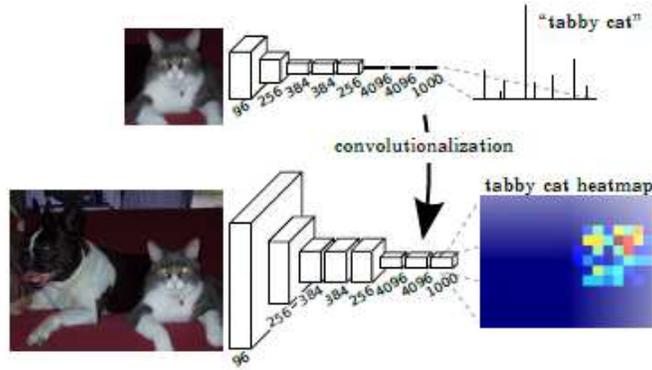


Figura 2.14: Salida visualizada como mapa de calor tomado de [23], el mapa de calor representa la posición donde se ubica el gato en la imagen original

2.6.4. Red convolucional U-Net

U-Net es una arquitectura basada en la *Fully Convolutional Network*, pensada para trabajar con menor cantidad de imágenes de entrada y a su vez alcanzar mejores resultados de segmentación. Uno de los rasgos que justifica su elección como la arquitectura que sustenta su utilización en este trabajo, es que desde que se presentó en [24], se utilizó para imágenes biomédicas.

Algunas de las modificaciones principales que presenta esta red en comparación con la FCN, es que se incrementa la cantidad de filtros o canales, lo que le permite a la red aprender más elementos en los mapas de características de mayor resolución, las cuales se propagan a niveles de menor resolución producto del submuestreo. En esta arquitectura, las fases de codificación (encoder) y decodificación (decoder) son casi simétricas por lo que se representa con la forma de U.

Es importante acotar que esta red solo puede reconstruir las secciones donde se encuentra disponible el contexto completo de la imagen, en partes de la imagen como los bordes, se recurre al reflejado para completar la predicción, véase la figura 2.15.

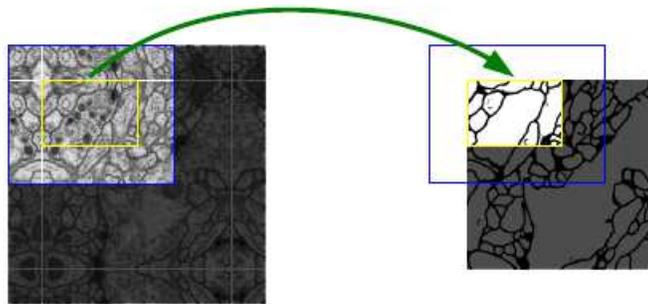


Figura 2.15: Estrategia de reconstrucción mediante reflejo de la imagen tomado de [24]

Además, en el campo de imágenes biomédicas está presente el desafío del reconocimiento de objetos que se traslapan, en nuestro contexto son células, tal y como se muestra en la figura 2.16. Para tales efectos, la red se entrena mediante una función de costo que tome en cuenta ese efecto.



Figura 2.16: Traslape de células representado con cambios de intensidad en escala de grises

Considere el diagrama que se muestra en la figura 2.17, el cual corresponde a la arquitectura de la red U-Net.

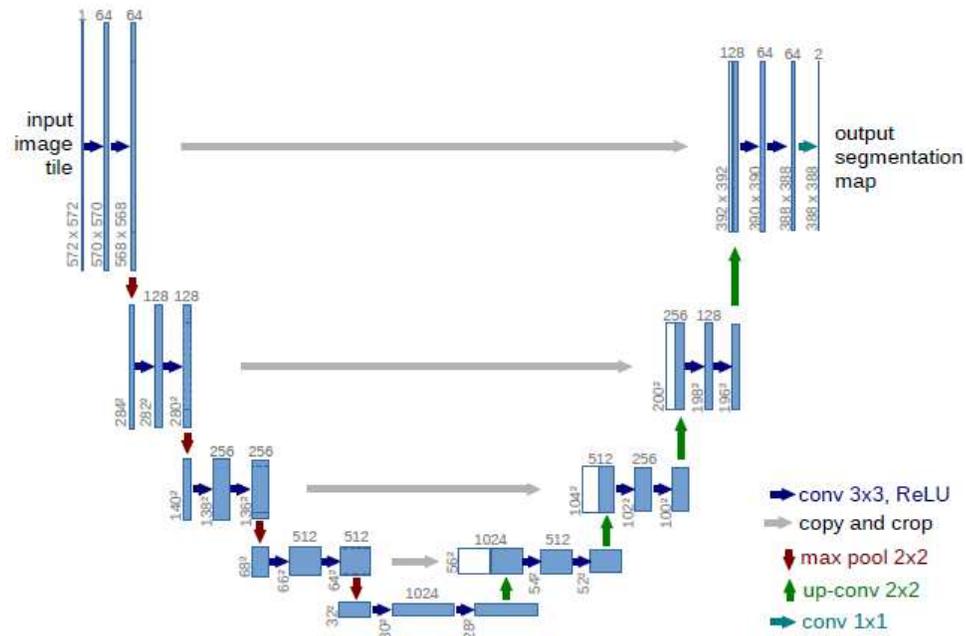


Figura 2.17: Arquitectura de U-Net tomado de [24]

Con base en la figura 2.17, de forma descendente hasta el punto mínimo de la forma de U, en la parte de submuestreo se tienen repetidas convoluciones con un

kernel de tamaño 3x3, seguidas de una ReLU y una operación de max pooling con una ventana de 2x2 que se desplaza dos píxeles en los ejes x y y . Por cada vez que se extraen características, se duplican la cantidad de filtros, lo que permite la propagación de características de las capas de mayor resolución. Para la ruta de reconstrucción de la imagen, *decoder*, se tienen convoluciones con kernels de tamaño 2x2 que reducen a la mitad la cantidad de canales que se habían creado. Un elemento muy importante y característico de U-Net, es que realimenta los mapas de características obtenidos en el *downsampling*, a la etapa de *upsampling*, para concatenar los mapas de activación que se van obteniendo. Se realizan convoluciones de 3x3 seguidas de capas ReLU. Por último, una convolución de 1x1 es aplicada para mapear los mapas de características hacia los píxeles de la imagen reconstruida.

2.7. Métricas

Las métricas descritas en esta sección se utilizan con dos motivos diferentes: Como función de costo para entrenar la red neuronal convolucional U-Net, y como funciones que permiten evaluar la precisión de la segmentación semántica hecha por la red.

2.7.1. Índice de Dice

El proceso de entrenamiento de una red neuronal o alguna arquitectura de aprendizaje automático se realiza mediante la optimización de una función mediante la iteración. Para el caso de U-Net, se utiliza el Índice de Dice, el cual, se utiliza en imágenes binarias para evaluar sus diferencias y también se le conoce como índice de *Sorensen*.

Para un par de imágenes $U \in R^{m \times n}$ con su par ordenado $U[i, j]$ que equivale a la imagen segmentada por la CNN y $V \in R^{m \times n}$, con $V[i, j]$, que equivale al ground-truth de la imagen; con base en lo anterior, el índice de Dice se describe mediante la ecuación 2.22.

$$S = \frac{2 \left(\sum_j^m \sum_i^n U[i, j] V[i, j] \right)}{\left(\sum_j^m \sum_i^n U[i, j] \right) + \left(\sum_j^m \sum_i^n V[i, j] \right)} \quad (2.22)$$

En notación simplificada se presenta en la ecuación 2.23.

$$S = \frac{2 |U \cap V|}{|U| + |V|} \quad (2.23)$$

Donde se deduce que los objetos de interés segmentados de U se comparan con los de la referencia V mediante la intersección de las imágenes, esto es dividido entre la cantidad de píxeles correspondientes a objetos de interés de ambas imágenes. La mejor segmentación será cuando S tienda a “1”, mientras que la peor se da con valores de S cercanos a “0”.

2.7.2. F Measure

Para esta métrica es necesario entender los conceptos de falsos/verdaderos positivos y negativos, precisión y exhaustividad. La figura 2.18 muestra gráficamente los conceptos de: Falsos positivos, falsos negativos, verdaderos positivos y verdaderos negativos.

		Clase Real	
		C1	No C1
Clase estimada	C1	Verdaderos Positivos (VP)	Falsos Positivos (FP)
	No C1	Falsos Negativos (FN)	Verdaderos Negativos (VN)

Figura 2.18: Conceptos básicos para métricas

Estos conceptos pueden ser aplicables a múltiples clases, sin embargo, en el contexto actual se utilizarán solamente dos, píxeles pertenecientes a la clase C1, y no pertenecientes a C1; esto se extiende de la siguiente manera:

- **Verdaderos positivos:** Se predijo que un píxel corresponde a la clase C1, lo cual es correcto.
- **Falsos Positivos:** Se predice que un píxel pertenece a C1, sin embargo, no es así.
- **Falsos negativos:** Se estima que un píxel no pertenece a la clase C1 cuando realmente sí forma parte.
- **Verdaderos negativos:** Se estima correctamente que un píxel está fuera de la clase C1.

Se define **exhaustividad** como la tasa de positivos que se clasifican correctamente tanto como pertenecientes a C1 o no, y está dado por la ecuación 2.24. Para la nomenclatura de la ecuación se tiene que TVP representa la exhaustividad, VP los verdaderos positivos y FN los falsos negativos.

$$TVP = \frac{VP}{FN + VP} \quad (2.24)$$

El término **precisión** o **valor predictivo positivo** hace referencia a la tasa de predicciones acertadas sobre el total de estimaciones que se realizaron y esta dado por la ecuación 2.25, donde VPP hace referencia al valor predictivo positivo, VP a los verdaderos positivos y FP a los falsos positivos.

$$VPP = \frac{VP}{VP + FP} \quad (2.25)$$

Con los conceptos anteriores, se define la métrica **F measure** según la ecuación 2.26.

$$F = \frac{2 \cdot TVP \cdot VPP}{TVP + VPP} \quad (2.26)$$

2.7.3. Distancia de Jaccard

El índice de Jaccard o también conocido como el coeficiente de similitud de Jaccard, en el contexto de imágenes, es una métrica que se utiliza para la comparación de píxeles pertenecientes a clases con respecto a su groundtruth. Para una imagen estimada X y una imagen etiquetada de referencia (groundtruth) Y se calcula el índice de Jaccard según la ecuación 2.27.

$$J(X, Y) = \frac{X \cap Y}{X \cup Y} \quad (2.27)$$

Ahora bien, se presenta la distancia de Jaccard como el complementario probabilístico del término anterior. La métrica está dada por la ecuación 2.28 y representa la desemejanza entre dos imágenes, en el contexto de imágenes esto se refiere a una imagen con segmentación estimada automáticamente contra su groundtruth.

$$J_\delta(X, Y) = 1 - J(X, Y) \quad (2.28)$$

Para la distancia de Jaccard, se trabajan con coeficientes en el rango de 0 a 1, donde la mejor segmentación se da cuando el coeficiente es cercano al cero, y la peor cuando se acerca a uno.

3. Diseño e implementación del sistema de segmentación de células

En este capítulo se muestra la selección de la solución que permite el diseño e implementación del sistema de segmentación de células, utilizando un flujo de diseño de ingeniería.

3.1. Análisis de requerimientos

Para llevar a cabo el proceso de diseño e implementación de la solución, se definieron los requerimientos mínimos que debe contemplar el sistema de segmentación de células. En la tabla 3.1 se muestran los requerimientos mínimos que deberá tener el sistema de segmentación semántico.

Tabla 3.1: Requerimientos primordiales del sistema

Requerimientos del sistema de segmentación de células
1. Ser capaz de reconocer objetos a partir de imágenes con bajo contraste
2. Clasificar los píxeles de la imagen en dos clases distintas: background o foreground
3. Debe ser reproducible para distintos sets de datos de glioblastoma

3.2. Análisis y selección de la solución

Es importante valorar las opciones que se tienen para la implementación de un sistema de segmentación de células, sobre todo para asegurarse que los recursos que se poseen son suficientes para cumplir a cabalidad con los requerimientos y especificaciones. A continuación se proponen tres soluciones para el diseño e implementación del sistema de segmentación de imágenes de células tomando en cuenta los criterios de la tabla 3.1.

- Solución 1: Técnicas clásicas del procesamiento digital de imágenes.
- Solución 2: Enfoque de aprendizaje automático (Machine Learning, ML).
- Solución 3: Enfoque de aprendizaje profundo (Deep learning, DL).

Solución 1

Para llevar a cabo esta solución se necesitaría realizar un procesamiento previo a las imágenes, utilizando una ecualización de histogramas para mejorar el contraste. La identificación de células se realizaría tomando en cuenta un tipo de

biblioteca de alto nivel llamado OpenCV utilizando un lenguaje de programación C++ o Python.

Como técnica de detección de células mediante OpenCV, se utilizaría en primera instancia el operador morfológico de erosión en conjunto con un detector de bordes para encontrar los límites de las estructuras que correspondan a células; posteriormente, se haría una binarización mediante el método de Otsu, el cual permite definir un umbral en escala de grises, permitiendo segmentar la imagen.

Solución 2

En el caso del aprendizaje automático (ML), se podría implementar la solución mediante la técnica de *máquina de soporte vectorial (SVM)*, como se ha hecho en [25], las cuales son las predecesoras históricas de las redes neuronales. Las SVM se basan en buscar un tipo de región de decisión que maximice el margen de separación de un píxel perteneciente a una clase con respecto a otro de clase distinta. La selección de esta solución implica experimentar con el pre-procesamiento de las imágenes de entrada de la red para evaluar su impacto. También se debe validar el modelo en aspectos de ajuste de sus predicciones, para determinar si se ajusta acertadamente a las muestras etiquetadas. Además, la cantidad de datos que se disponga podría ser un factor que afecte su desempeño.

Solución 3

En el caso del aprendizaje profundo (DL), se propone un modelo de redes neuronales convolucionales, especializadas para segmentación de imágenes biomédicas [24] [26]. El cual a partir de aprendizaje supervisado, se entrena para segmentar nuevas imágenes de células que se apliquen a su entrada.

Para construir el modelo de segmentación con una red convolucional, se aplicaría una etapa de preprocesamiento en la entrada, con el fin de mejorar la precisión de la segmentación semántica.

También se realizaría aumento de datos, el cual consiste en aplicar transformaciones espaciales sobre el set de datos de entrenamiento, para cubrir posibles variantes que tengan las muestras futuras que se quieran predecir.

3.2.1. Selección de la solución

Con base en las soluciones planteadas en la sección anterior, para seleccionar la opción que se ajusta a los requerimientos establecidos, se evaluarán las propuestas de acuerdo a las siguientes métricas:

- Funcionalidad.
- Precisión.
- Tiempo de ejecución.

- Costo computacional.
- Reproducibilidad.

Se realiza una ponderación numérica para cada una de las métricas establecidas anteriormente, la cual se muestra en la tabla 3.2, donde los valores representan los siguientes criterios:

- 1 es malo.
- 2 es medio.
- 3 es bueno

Tabla 3.2: Pesos para parámetros de selección

Parámetro	Clásico	ML	DL
Funcionalidad	3	3	3
Precisión	1	2	3
Tiempo de Ejecución	1	2	1
Costo computacional	3	2	1
Reproducibilidad	1	1	3
Total	9	11	13

Para el parámetro de funcionalidad, se tiene que las tres técnicas cumplen con el cometido de segmentar una imagen para determinar poblaciones de células, si bien es cierto algunas técnicas lo harán con mayor esfuerzo que otras, esto no es considerado dentro de este criterio. En cuanto a la precisión, el enfoque de aprendizaje profundo (DL) ha ganado todas las competencias de segmentación de la comunidad de visión por computador, esto causa que el enfoque clásico y de aprendizaje automático (ML) queden en segundo y tercer plano [22].

En el tiempo de ejecución de un algoritmo clásico, se debe evaluar si lo que se requiere hacer está considerado dentro de alguna biblioteca de software, que ejecute las tareas de forma paramétrica, si no es así, se deberá recurrir a ciclos que se desplacen por las coordenadas x,y de la imagen y realicen las transformaciones espaciales que conllevan los métodos; si las dimensiones de la imagen son muy grandes, entonces se tendrán tiempos de ejecución considerables. Para el enfoque de ML, los métodos están integrados en ambientes de software, por lo tanto se ejecutan de forma paramétrica y se tienen tiempos de ejecución medios. En cuanto a la solución por aprendizaje profundo, se habla de múltiples iteraciones, operaciones complejas tales como la convolución, muchos datos de entrada, almacenamiento de memoria elevado, y otros conceptos, los cuales hacen que se tarde considerable cantidad de tiempo en ejecutar un modelo.

El costo computacional es un factor clave para la selección, solo las primeras dos soluciones son capaces de ejecutarse en computadores convencionales. Una aplicación de aprendizaje profundo, usualmente requiere de GPU's o supercomputadores con una gran cantidad de núcleos.

La reproducibilidad, es otro de los parámetros clave en los cuales las técnicas de aprendizaje profundo sacan ventaja sobre los demás enfoques. Las redes convolucionales son robustas ante traslaciones, es decir, un modelo entrenado para segmentación de células es capaz de ajustarse a sets de datos similares, no así las otras dos técnicas, las cuales implican repetir el proceso de diseño para todos los sets de datos de entrada que se deseen probar.

Según la tabla 3.2, se muestra un panorama numérico en el cual se puede basar la comparación de la conveniencia de la solución por seleccionar. Al realizar una sumatoria de pesos, se obtiene que la técnica de aprendizaje profundo presenta un mayor promedio con 13 puntos, teniendo en cuenta que la precisión es quizá el parámetro fundamental para la aplicación requerida, por lo tanto, se elige el enfoque *Deep Learning* para diseñar e implementar el sistema de segmentación automática.

3.2.2. Desarrollo de la solución

Debido al balance entre costos y beneficios planteado anteriormente, se elige la tercera solución como el sistema de segmentación que se implementará.

Ante el problema planteado de las características obtenidas por microscopía, que no permiten identificar las células de una imagen con claridad, se pretende implementar un algoritmo computacional que segmente de forma automática las células de una imagen digital.

Se seguirá una metodología de descomposición de bloques para el desarrollo del sistema, partiendo de alto nivel, se visualizarán los bloques como cajas negras, en las cuales se definen sus entradas y salidas. Según se avanza en el desarrollo de la solución, se obtendrán versiones más detalladas del funcionamiento de cada módulo.

La implementación en alto nivel del sistema se ilustra en la figura 3.1; donde una imagen de entrada, al ser procesada genera una de salida con las características de interés segmentadas semánticamente.



Figura 3.1: Diagrama de alto nivel de la solución a implementar

En un segundo nivel, el sistema se divide en dos módulos: filtrado no lineal y arquitectura de aprendizaje profundo. Es conocido que las imágenes obtenidas mediante microscopía por fluorescencia poseen bajo contraste y es tarea complicada reconocer patrones visualmente, por lo tanto, el bloque de filtrado no lineal mejorará las características espaciales de la imagen en cuanto contraste, definición de bordes y disminución de ruido. El bloque de aprendizaje automático se entrenará mediante aprendizaje supervisado para reconocer los píxeles que correspondan a células de las imágenes que entren al sistema. En la figura 3.2 se muestra el diagrama de segundo nivel del sistema de segmentación semántica.

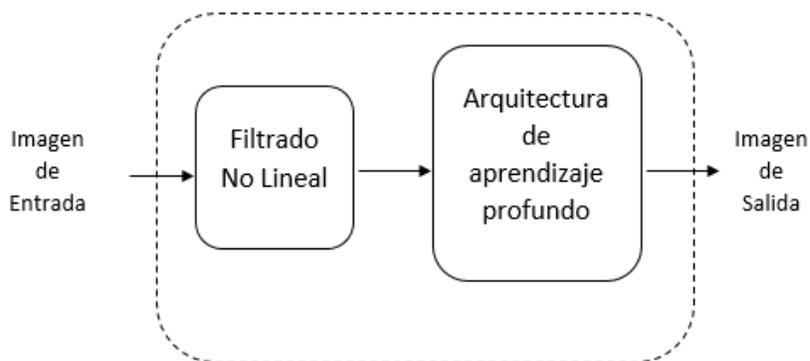


Figura 3.2: Diagrama de segundo nivel

3.3. Módulo de filtrado No lineal

Este es el primer módulo que se verá de forma minuciosa según la metodología propuesta para la solución. Una imagen de entrada, de pobres características en cuanto a contraste y reconocimiento de objetos, deberá exhibir características perceptibles visualmente a la salida del bloque.

Si se baja el nivel de abstracción de la figura 3.2 en el bloque de filtrado no lineal, se llega a módulos que forman parte del filtro DNLM; los cuales, si se combinan de la forma que indica la figura 3.3, pueden ser implementados mediante software para crear el filtro no lineal.

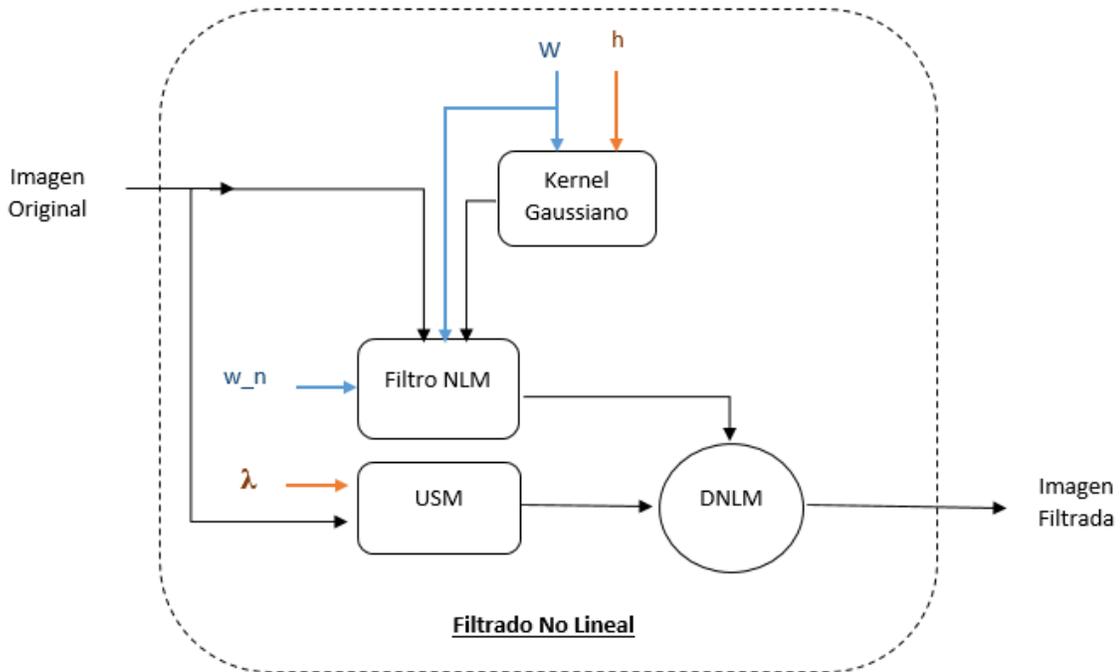


Figura 3.3: Diagrama de relación de módulos para algoritmo DNLM

Nótese que la figura 3.3 es un nivel más bajo de lo que antes fue una caja negra llamada *Filtrado No Lineal*, las entradas y salidas del bloque se mantienen iguales:

- Imagen original: imagen digital obtenida mediante microscopía por fluorescencia, imagen con pobre contraste.
- Imagen filtrada: imagen digital con patrones perceptibles visualmente, debido a la mejora del contraste, definición de bordes y supresión de ruido.

Ahora se tienen cuatro parámetros internos, llamados w , w_n , h , λ . Los cuales están identificados mediante dos colores distintos, aquellos en azul son parámetros estándar, utilizados frecuentemente en el procesamiento de imágenes; los que se identifican en naranja, son los que se varían para lograr distintas combinaciones para mejoras de contraste, demarcación de bordes y eliminación de ruido. A continuación se explica cada parámetro:

- w : tamaño de una ventana cuadrada de búsqueda para el algoritmo de promediado no local (NLM), que se tomará como referencia para promediar los valores de los vecindarios locales de cada píxel.

- w_n : tamaño del vecindario de cada píxel.
- h : Factor de suavizado del filtro DNLM.
- λ : Factor de ganancia de kernel laplaciano, determina la definición de bordes y mejora de contraste.

En relación con lo anterior, para una mejor comprensión del funcionamiento de este módulo, en la figura 3.4 se muestra el diagrama de flujo del algoritmo a implementar.

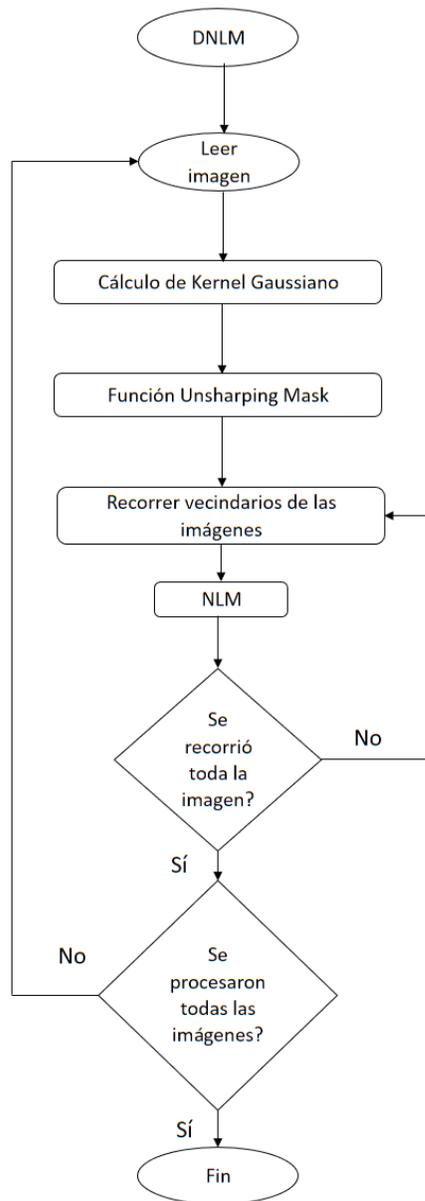


Figura 3.4: Diagrama de flujo de algoritmo DNLM

El algoritmo es implementado en Matlab, utilizando métodos propios del paquete de procesamiento digital de imágenes, el cual permite hacer la lectura desde un directorio donde se almacenan los archivos que se desean preprocesar.

Las funciones contenidas en la biblioteca de procesamiento digital de imágenes, brindan ayuda en cuanto a métodos específicos como cálculo de kernels gaussianos, laplacianos, entre otros; sin embargo, el flujo de diseño planteado y la programación del algoritmo es parte del aporte en el diseño de software proyectado para este módulo.

Según lo mostrado en la figura 3.3, el parámetro del tamaño de ventana que se ingresa como entrada, determinará el núcleo gaussiano que se utiliza en el filtro de promediado no local (NLM).

Una vez calculados esos valores estáticos, se procede a ejecutar la función Unsharpening Mask (USM), que toma como entrada el parámetro λ , el cual brinda la ganancia del filtro laplaciano que se encarga de la detección de bordes del algoritmo. De acuerdo a los tamaños de ventana que se le indiquen al algoritmo, se recorrerá la imagen mediante ciclos para realizar cálculos de suma y ponderación de vecindarios, esto corresponde al algoritmo promediador no local (NLM) y se hará así hasta haber procesado la totalidad de las imágenes en el directorio de fuente de donde se toman las muestras.

3.4. Arquitectura de aprendizaje profundo: U-Net

En esta sección se describe el proceso de implementación de una arquitectura de aprendizaje profundo, específicamente U-Net. En la figura 3.5 se muestra una descomposición a bajo nivel del bloque de aprendizaje profundo de la figura 3.2.

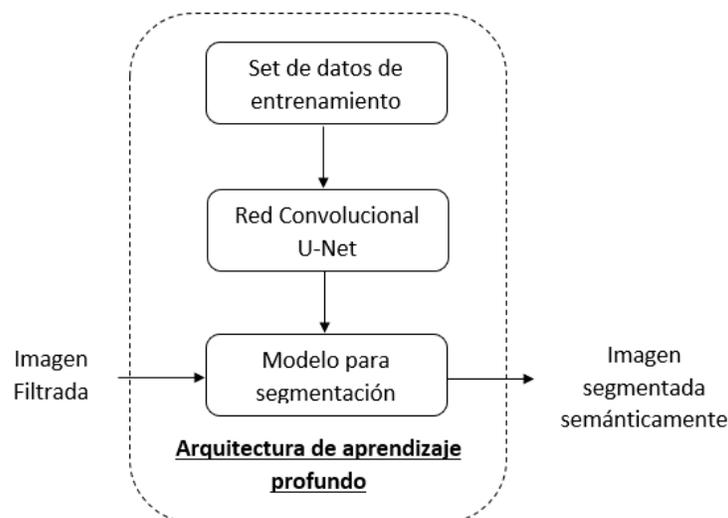


Figura 3.5: Diagrama de segundo nivel del módulo de aprendizaje automático

La entrada de este módulo es la imagen con filtrado no lineal y la salida corresponde a la segmentación semántica de células. Las arquitecturas de aprendizaje automático deben entrenarse para realizar su tarea específica; la red convolucional U-Net, se entrenará mediante aprendizaje supervisado, por lo cual resulta primordial preparar el set de datos de entrenamiento de la forma que se explica a continuación.

3.4.1. Set de datos de entrenamiento para red U-Net

Todas las arquitectura de aprendizaje supervisado poseen etapas de entrenamiento y pruebas, es en la primera de ellas donde se construye el modelo para segmentación.

El módulo de set de datos de entrenamiento de la figura 3.5 se visualiza en un nivel más profundo, en la figura 3.6.

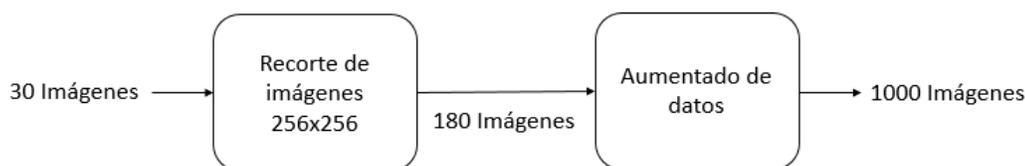


Figura 3.6: Segundo nivel del módulo set de datos de entrenamiento

Es de importancia referirse a los recursos disponibles que se tuvieron al iniciar el proyecto, entre ellos estaban tan sólo 30 imágenes de dimensiones de 696×520 píxeles debidamente referenciadas con su groundtruth. Dichas imágenes fueron las que se obtuvieron gracias a la recomendación de los microbiólogos.

Existen dos motivos principales por los cuales resulta necesario ampliar la cantidad de datos de entrenamiento: por concepto las aplicaciones de aprendizaje profundo mejoran su rendimiento con más datos y el recurso computacional disponible no es suficiente para trabajar con las dimensiones de las imágenes originales.

El recurso computacional del proyecto es provisto por el supercomputador kabré del Centro Nacional de Alta Tecnología (CeNAT), en el cual se ejecuta la implementación de U-Net.

Debido a que U-Net es una red completamente convolucional, y la convolución es una operación que requiere el uso de múltiples cálculos tales como adición y multiplicación, la memoria del clúster Kabré no puede operar sobre imágenes de dimensiones tan amplias debido a insuficiencia de memoria.

Se probó que las dimensiones máximas para ejecutar la red U-Net sin tener problemas de memoria en el sistema computacional en el que se ejecuta, son de 256×256 ; por lo tanto, de una imagen con dimensiones originales, se extraen seis

recortes de 256×256 píxeles, lo cual expande el set de datos de treinta imágenes a ciento ochenta.

Para realizar los recortes de recuadros de dimensiones menores a partir de las imágenes originales, se tomó en cuenta el criterio planteado en [24] donde se demuestra que la red U-Net es sensible a la redundancia de datos, esto quiere decir información repetida en las imágenes, por ende se sigue la subdivisión planteada en la figura 3.7

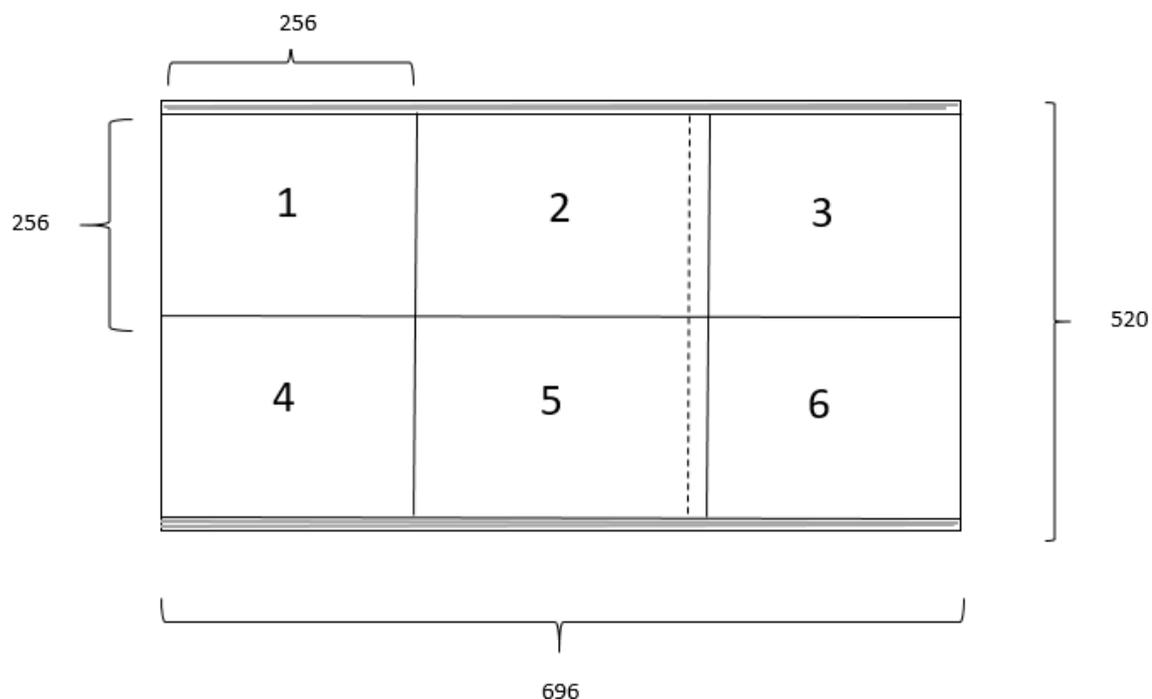


Figura 3.7: Imágenes obtenidas a partir de una original

De la figura 3.7 debe entenderse que los recuadros enumerados del 1 al 6 son las nuevas imágenes que se extraen de la original (de dimensiones 696×520), esta distribución es la que otorga menor redundancia de información, la cual existe únicamente entre las líneas punteada y continua entre los recuadros [2, 3; 5, 6].

Ahora bien, para efectos de rendimiento de la arquitectura de aprendizaje profundo, se utiliza aumentado de datos para incrementar la cantidad de imágenes de entrenamiento. El aumentado de datos consiste en realizar transformaciones al set de datos disponible para enseñar variabilidad a la red.

Mediante la biblioteca de Python llamada Augmentor, se aplican transformaciones de rotación y reflejo sobre las 180 imágenes de 256×256 , y se obtienen 1000 nuevas con las mismas dimensiones.

3.4.2. Red convolucional U-Net

La implementación de la arquitectura U-Net se realiza en Python, utilizando la biblioteca *Keras*, la cual es especializada en aprendizaje profundo, y sirve como interfaz de desarrollo para redes convolucionales. En la figura 3.8, se muestra un diagrama de flujo que explica la implementación del algoritmo.

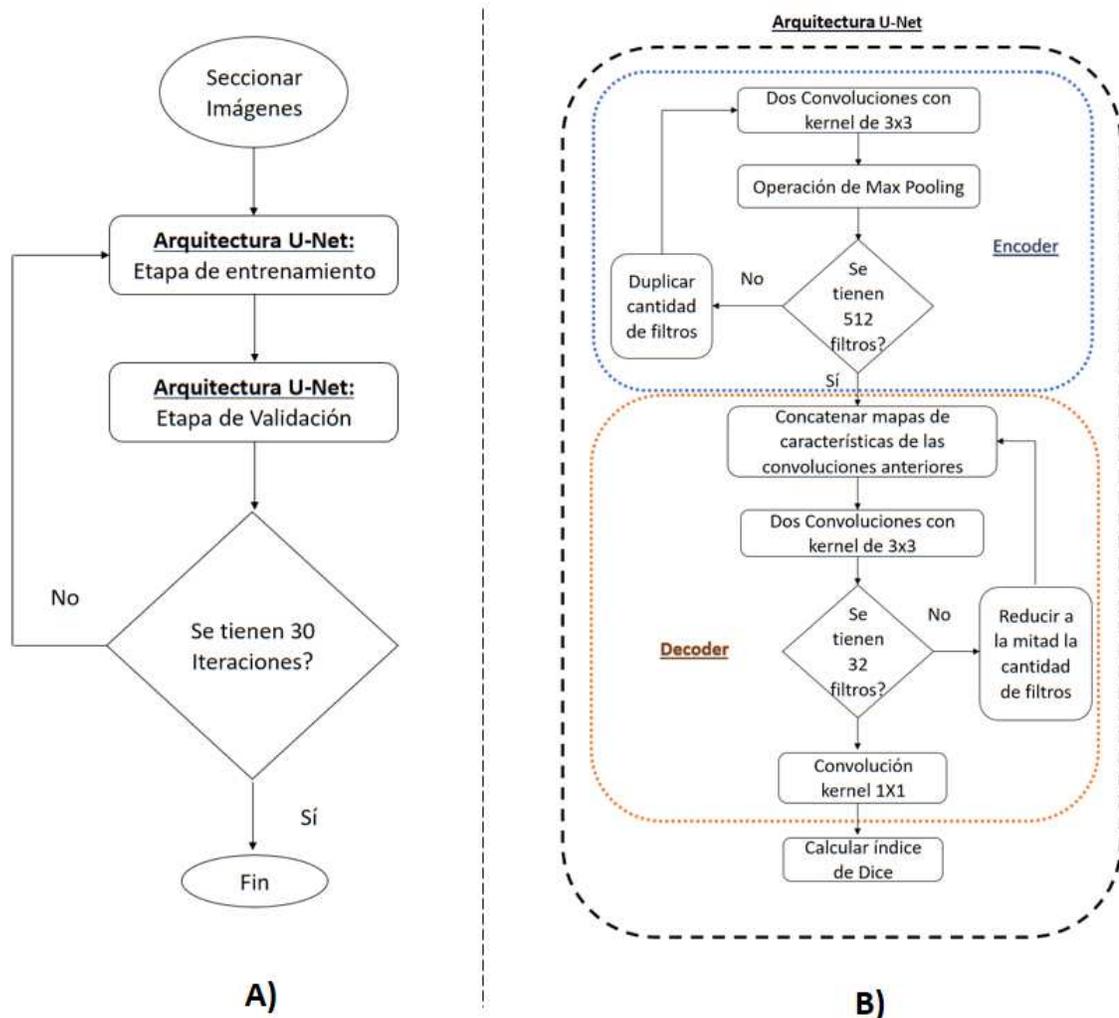


Figura 3.8: Diagrama de flujo de algoritmo de U-Net
 A) estructura del algoritmo, B) diagrama de segundo nivel del bloque de arquitectura U-Net

Cuando se tiene listo el set de datos de entrenamiento, se puede hacer una partición del lote total, de la forma 70% para entrenamiento y 30% para verificación, que son proporciones que se utilizan convencionalmente en aplicaciones de aprendizaje automático [27].

En relación con la figura 3.8.A, mediante el proceso de entrenamiento se aprenden los coeficientes de los kernels contra los cuales se aplican las operaciones de

convolución. Para la etapa de verificación, se prueban los kernels aprendidos y se realizan predicciones sobre el set de datos de 30% que se destinó para ese fin. Este es un proceso de treinta iteraciones que tiene como propósito optimizar la función de costo de la red, la cual es el índice de Dice y debe incrementarse con cada repetición.

Con base en la figura 3.8.B, la arquitectura U-Net consta de dos secciones principales: submuestreo o *encoder* (que va desde el bloque *iniciar entrenamiento* hasta el punto de control de 512 filtros en la figura 3.8) y reconstrucción o *decoder* (que va desde el punto de control de 512 filtros hasta el bloque *convolución kernel 1x1* en la figura 3.8). Para la etapa de *encoder*, para la cantidad de canales (filtros) que se tienen en cada nivel, la red ejecuta dos convoluciones contra kernels de 3x3 y luego la operación de *maxpooling*; se inicia con 32 filtros, los cuales se duplican hasta alcanzar el último nivel con 512 filtros.

En la etapa de *decoder*, se sigue un proceso inverso, el cual inicia con 512 filtros y se va reduciendo a la mitad periódicamente en cada nivel, hasta alcanzar 32 filtros. Cada ocasión que se incrementa la cantidad de canales, hay un proceso de concatenación de mapas de características del nivel correspondiente de la etapa de *encoder*. Además, debe notarse que U-Net es derivado de una red completamente convolucional, donde las operaciones de *maxpooling* para la etapa de reconstrucción se sustituyen con convoluciones.

El último paso para la arquitectura es ejecutar una convolución contra un kernel de 1x1, la cual asigna una probabilidad de pertenencia a células para todos los píxeles, utilizando una función de activación tipo sigmoide.

4. Análisis y validación

Este capítulo presenta los resultados obtenidos según el diseño e implementación del sistema de segmentación semántica de células. Todos los resultados se dieron a partir del set de datos recomendado por los microbiólogos.

4.1. Set de imágenes iniciales

Se comienza por observar las imágenes de entrada, para conocer sus características perceptuales. Por recomendación de los especialistas en microbiología, se recomienda utilizar una selección de imágenes de 30 muestras del set de datos *BBBC006v1*, del **Broad Bioimage Benchmark Collection** [Ljosa et al., Nature Methods, 2012].

La figura 4.1 muestra una imagen original, sin alterar, la cual es representativa del conjunto de imágenes de células que se utilizan en este trabajo, nótese que debido a sus propiedades de bajo contraste es imposible apreciar alguna estructura a simple vista.



Figura 4.1: Imagen original

4.2. Clúster Kabré

Los procesos que involucran el uso del lenguaje de programación Python (aumentado de datos y arquitectura de aprendizaje profundo) fueron ejecutadas en el supercomputador Kabré propiedad del Centro Nacional de Alta Tecnología. Cada servidor que compone el clúster se denomina nodo, para este trabajo se utiliza el nodo Tule. En la figura 4.2 se muestra un diagrama de su distribución.

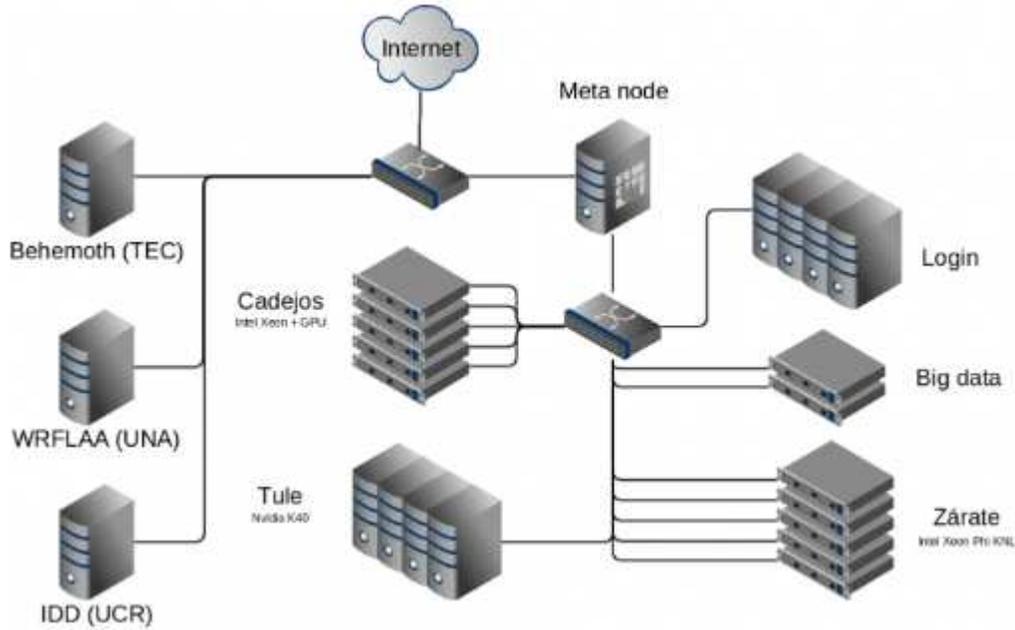


Figura 4.2: Arquitectura del clúster Kabré
 Imagen tomada de <http://cluster.cenat.ac.cr/guia>

Para ejecutar la red neuronal convolucional U-Net dentro del nodo Tule, se utilizó la GPU NVIDIA Tesla K40, cuyas propiedades se muestran en la tabla 4.1.

Tabla 4.1: Recursos de Hardware del GPU K40

1. GPU para procesamiento paralelo acelerado
2. 745 MHz GPU
3. 2880 núcleos CUDA
4. 12GB GDDR5 RAM

4.3. Procesamiento de imágenes con filtrado No lineal

Se procedió a analizar el comportamiento del procesamiento digital de imágenes realizado sobre el set de datos inicial mediante el módulo de filtrado no lineal, el cual consiste en el filtro DNLM, donde se realizaron transformaciones en el dominio espacial de cada muestra para modificar sus características perceptuales.

Para el algoritmo DNLM, se tienen dos parámetros principales, de los cuales depende el grado de mejora de contraste, definición de bordes y disminución de ruido; ellos son h que representa el factor de suavizado del filtro y λ la ganancia que permite la detección de bordes.

Para establecer las combinaciones para estos parámetros (h y λ), se siguieron los siguientes criterios:

- El valor mínimo por defecto para los parámetros es cero.
- El valor máximo para h es aquel en donde la imagen no luzca desenfocada debido al kernel gaussiano.
- Para el caso de λ , su máximo será la ganancia donde la intensidad de los píxeles en los bordes de las células permita una clara identificación.
- El intervalo o *step size* para los valores de un mismo parámetro será aquel donde se logre apreciar un cambio significativo en la imagen.

Para verificar los criterios de forma visual, considere la figura 4.3.a, donde se resalta con un recuadro rojo el objeto de interés que permite validar la disminución de ruido, nótese un tipo de constelación que rodea el elemento de mayor intensidad de píxeles (representa el ruido en la imagen), la cual se disminuye conforme se aumenta el valor del parámetro h según las figuras 4.3.b y 4.3.c.

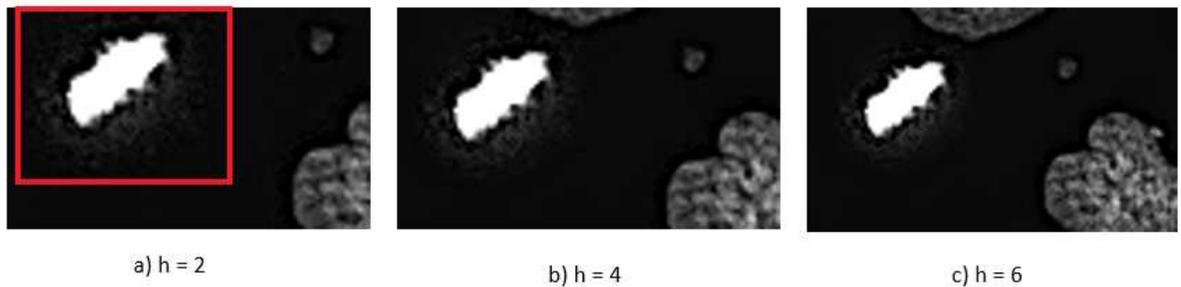


Figura 4.3: Disminución de ruido al aumentar h con intervalo de 2

Para el caso del parámetro λ , en la figura 4.4 se observa que conforme se incrementa el parámetro cada tres unidades, se le da mayor ganancia a los píxeles de los bordes de la célula, esto se debe a que se amplifican los valores de la máscara Laplaciana que detecta bordes y contornos en la imagen.

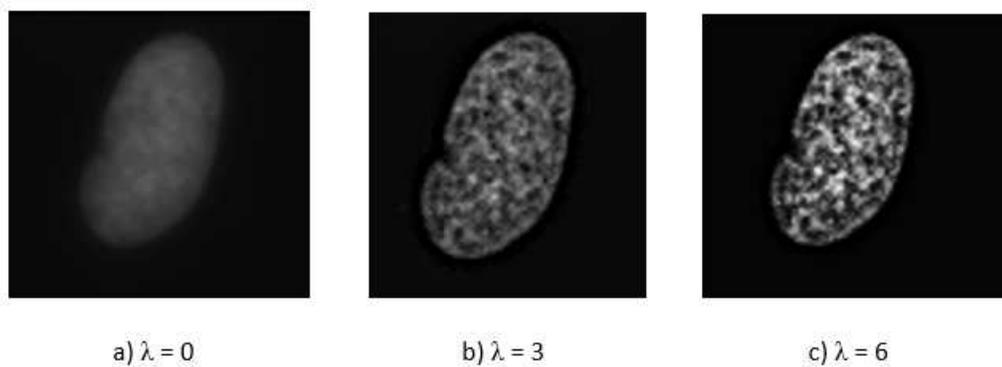


Figura 4.4: Intensidad en bordes al aumentar λ con intervalo de 3

Tomando en cuenta los criterios descritos y la verificación visual para la selección de las combinaciones de parámetros pertinentes para el filtro DNLM, se proponen las configuraciones que se muestran en la tabla 4.2.

Tabla 4.2: Combinaciones de parámetros establecidos para experimentación

h	λ
0	0
0	3
0	6
2	0
2	3
2	6
4	0
4	3
4	6
6	0
6	3
6	6

Se ejemplificaran algunas de estas combinaciones, mediante figuras que muestran la salida del bloque de filtrado no lineal, bajo los parámetros indicados. Partiendo del peor de los casos, donde ambos parámetros son cero, se tiene por consecuencia la imagen original sin alteración, debido a que no hay reducción de ruido y la máscara laplaciana es nula.

Ahora bien, si se realiza solamente filtrado de ruido, estableciendo λ , que es la ganancia de la función USM, igual a cero, se obtiene una imagen difuminada. En la figura 4.5 se muestra la imagen resultante al utilizar un valor de $\lambda = 0$.

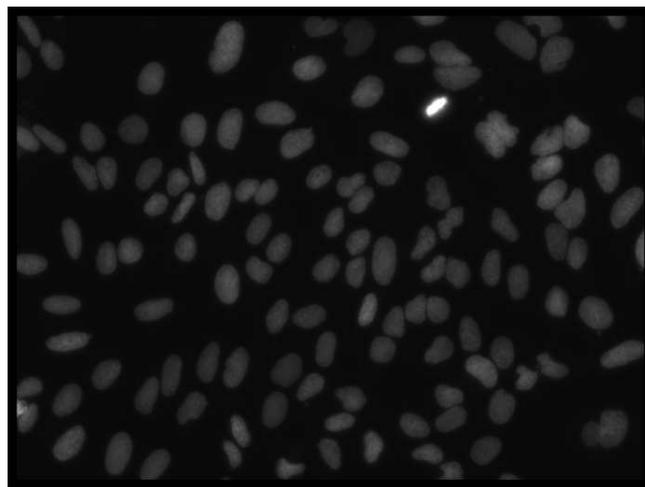


Figura 4.5: Imagen filtrada, $h = 6$ $\lambda = 0$

En el otro extremo, si se deja un factor h de suavizado del kernel gaussiano

muy cercano a cero, se aplicaría solamente la función USM, la cual está definida por una máscara laplaciana para detección de bordes con un factor de ganancia λ que amplifica la intensidad de los píxeles detectados pertenecientes a contornos de células. Esto realizaría mejora de contraste y detección de bordes, sin embargo, la imagen resultante carecería de disminución de ruido gaussiano. En la figura 4.6 se muestra la imagen resultante al hacer el valor de $h = 0$.

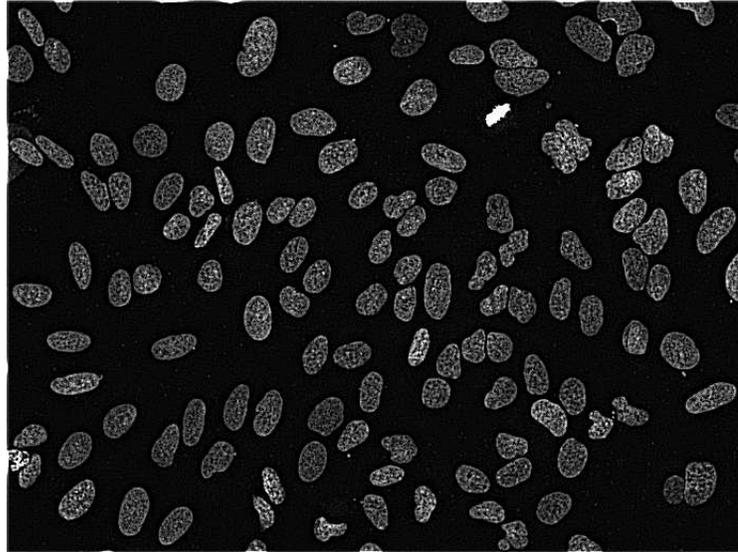


Figura 4.6: Imagen filtrada, $h = 0$ $\lambda = 6$

En un punto medio, si se seleccionan los parámetros con valores diferentes a cero, entonces se logran los efectos del filtro DNLM. La figura 4.7 muestra una imagen con mejor contraste junto con definición de bordes, gracias al aporte del parámetro λ y considerable disminución de ruido debido a la contribución de h .

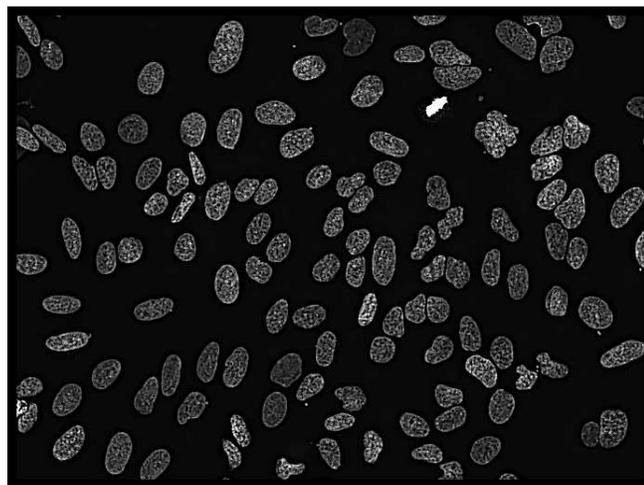


Figura 4.7: Imagen filtrada, $h = 6$ $\lambda = 6$

4.4. Generación de groundtruth

El groundtruth que se proporciona por el set de datos BBBC006v1, no se puede usar directamente en la red convolucional U-Net, debido a que está en escala de grises, como se muestra en la figura 4.8 a) y U-Net requiere imágenes binarias. Por ello, se aplicó mejora de contraste mediante ecualización de histogramas y se agregó un algoritmo de umbralización adaptativo con un filtro detector de bordes y método de Otsu, en la figura 4.8 b) se presenta el resultado.

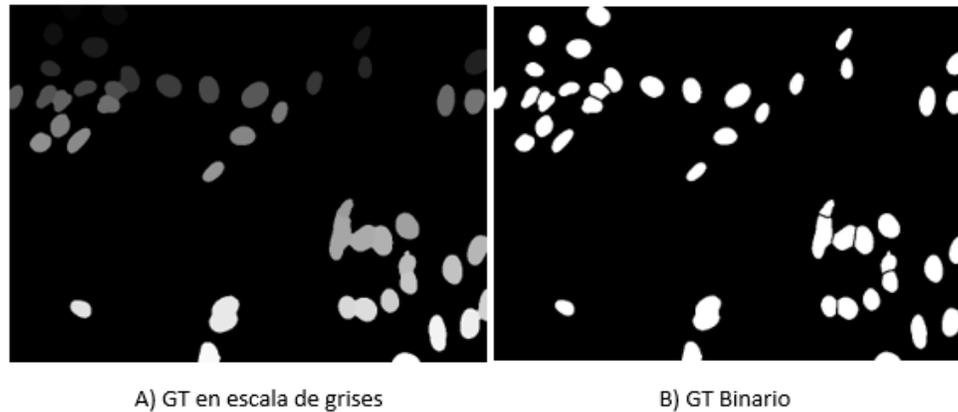


Figura 4.8: Groundtruth: a) escala de grises, b) binario

Para la generación del groundtruth, se procesó una imagen mediante una ecualización de histograma (figura 4.9 a); no obstante, para obtener un mejor detalle de las células se utilizó un filtro tipo Sobel (figura 4.9 b), y para binarizar la imagen fue necesario realizar procesamiento en Matlab (ver figura 4.9 c). Finalmente, para demarcar los bordes, se recorrió la máscara binaria, donde si un píxel posee un valor igual “1”, el píxel correspondiente en la imagen tendrá un valor de “0”.

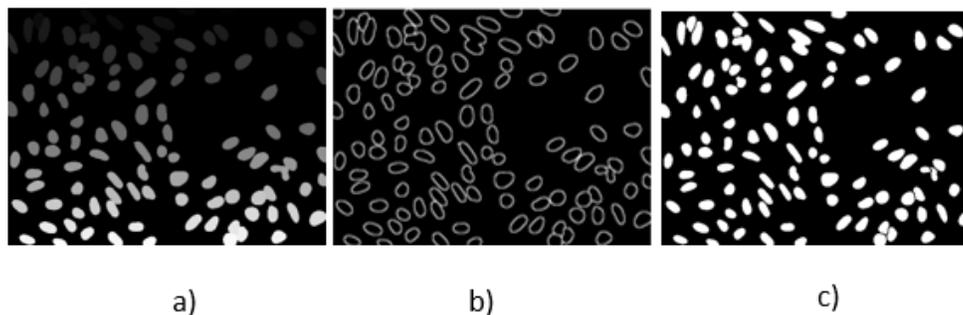


Figura 4.9: Proceso para generar Groundtruth
a) imagen original, b) máscara obtenida mediante detección de bordes, c) resultado de la binarización

Para tratar el desafío de la detección de células traslapadas se pensó en hacer más grueso (aumentar la cantidad de píxeles que pertenecen a los bordes) para

ello se aplicó un operador morfológico de erosión. En la figura 4.10 se muestra el resultado de aplicar el operador morfológico de erosión.

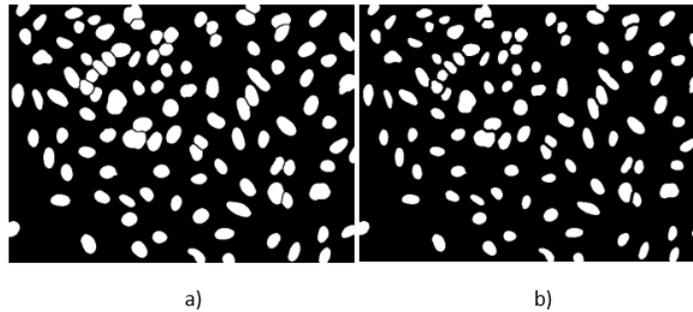


Figura 4.10: Groundtruth binario: a) sin erosión, b) con erosión

4.5. Aumentado de datos

El aumentado de datos se aplicó al set de datos de entrenamiento, mediante una programación en Python con la biblioteca Augmentor, donde se aplicaron las transformaciones que se enlistan a continuación debido a que estas no deforman las imágenes de células.

- Rotaciones de 90° , 180° y 270° .
- Rotaciones entre ángulos de -5° y 5° .
- Reflejado lateral.
- Reflejado vertical.

En la figura 4.11, se muestran las transformaciones de rotación de 90° , reflejado lateral y rotación de 180° para una imagen del groundtruth tomando como referencia las células encerradas en un círculo rojo.

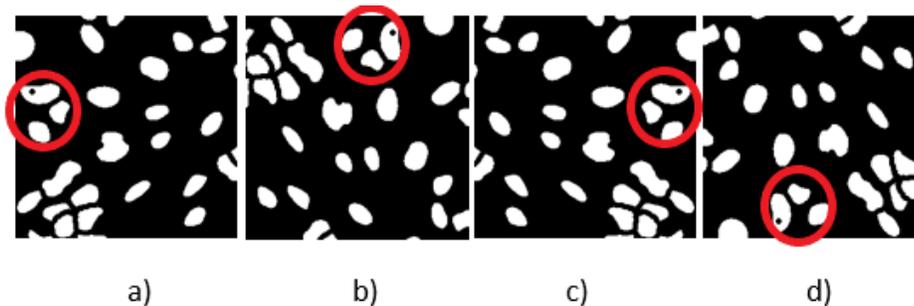


Figura 4.11: Transformaciones sobre imágenes para aumentado de datos
a) Imagen original, b) rotación de 90° , c) reflejado lateral, d) rotación de 180°

Para la implementación en Python de la biblioteca Augmentor, se tienen dos parámetros obligatorios: la cantidad de imágenes de salida que se desea obtener y la probabilidad de aplicar una transformación a una imagen. El valor probabilístico se representa mediante un coeficiente entre 0 y 1, el cual determina la aleatoriedad con la que se realizan las transformaciones. El carácter aleatorio de la biblioteca garantiza que para cada valor distinto entre 0 y 1 que se ingrese como parámetro, siempre se tendrán sets de datos diferentes.

4.6. Entrenamiento de red U-Net

Para el aprendizaje supervisado, una vez que se generó el set de datos de entrenamiento y la implementación de la red U-Net, se iniciaron las etapas de entrenamiento en la cual se realiza el cálculo del índice de Dice, como función de optimización de la red; mediante software se establece la opción para visualizar el índice en cada iteración. Para la arquitectura U-Net se probaron preliminarmente cada una de las combinaciones de los parámetros h y λ con una probabilidad de aumento de datos del 70 %. En relación con lo anterior, en la tabla 4.3 se muestran los resultados obtenidos para el Índice de Dice.

Tabla 4.3: Índice de Dice para combinaciones de parámetros evaluadas

Parámetros		Probabilidad
h	λ	70 %
0	0	0.8979
0	3	0.9345
0	6	0.9275
2	0	0.9165
2	3	0.9139
2	6	0.9103
4	0	0.9164
4	3	0.9187
4	6	0.9111
6	0	0.9174
6	3	0.9163
6	6	0.9088

Dado que, según el índice de Dice, la mejor segmentación se da cuando el coeficiente es cercano a 1, se tomaron los tres parámetros más altos de la tabla 4.3 para realizar más ejecuciones sobre el clúster, con el fin de verificar estadísticamente, que los valores no son producto del azar o la casualidad. Para realizar dicha verificación, se probó nuevamente la red con los mismos parámetros de filtrado de imágenes, pero con sets de datos distintos.

Se aprovecha la propiedad probabilística de la biblioteca Augmentors para procesar sets de datos diferentes, al variar la probabilidad de aplicar las operaciones de rotación y reflejado a las imágenes de entrada, según los rangos (de 10 % a 95 %) que se muestran en la tabla 4.4, en el cual también se muestran los resultados, en términos del índice de Dice para cada ejecución.

Tabla 4.4: Índice de Dice promedio para conjuntos de parámetros seleccionados

Parámetros		Probabilidades										Promedio	Desv. Est
h	λ	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	95 %		
0	0	0,8930	0,8815	0,9040	0,8934	0,8962	0,8941	0,9000	0,9145	0,9048	0,9001	0,8982	0,008816046
0	3	0,9313	0,9405	0,9328	0,9305	0,9295	0,9397	0,9350	0,9304	0,9401	0,9343	0,9344	0,004288862
0	6	0,9345	0,9269	0,9315	0,9256	0,9254	0,9292	0,9275	0,9305	0,9255	0,9300	0,9287	0,003019639
4	3	0,9170	0,9141	0,9151	0,9066	0,9164	0,9135	0,9151	0,9135	0,9126	0,9125	0,9136	0,002893748

Se realizó un mínimo de diez ejecuciones para cada combinación de parámetros (h y λ) para analizar la varianza de los resultados que se obtienen y verificar si es confiable estadísticamente, lo cual se comprueba en la tabla 4.4, en la columna *Desv. Est* que muestra la desviación estándar de los índices de Dice que se calculan en cada ejecución, la magnitud de sus valores demuestra que se tiene repetibilidad en las ejecuciones con poca variación.

No obstante, se mantiene la tónica entre los sets de datos, el mayor coeficiente de Dice es de 0,9344 y se obtiene cuando se preprocesan las imágenes de entrada de la CNN con $h = 0$ y $\lambda = 3$, representa la mejor segmentación debido a que es el valor más cercano a 1. Por lo tanto, se escogen estos parámetros de filtrado para el sistema de segmentación de células.

La red consta de dos etapas, entrenamiento y validación; en la etapa de validación se retorna como salida las predicciones que realiza el sistema, estas están identificadas apropiadamente, de tal forma que es posible compararlas con su groundtruth y establecer por medio de las métricas distancia de Jaccard y F Measure, la precisión de la segmentación. El set de datos que consta de 1000 imágenes se divide en 30 % para la etapa de validación de la red. Las cuales están numeradas desde la muestra 700 hasta la 1000.

4.7. Validación de Indicadores

Según se establece en el primer objetivo del proyecto, se debe comparar el beneficio de utilizar filtrado no lineal sobre las imágenes sin preprocesamiento alguno.

Entrenamiento de U-Net sin filtrado no lineal

Se entrena una red U-Net con las imágenes originales, sin aplicar filtrado y se analizan las predicciones que realiza la red, los resultados se muestran en la figura 4.12 se visualizan los resultados y se comparan con su respectivo groundtruth; nótese que la segmentación falla al predecir muchas células, donde se aprecia en las pequeñas manchas blancas que no tienen sentido semántico señaladas con recuadros en figura 4.12.

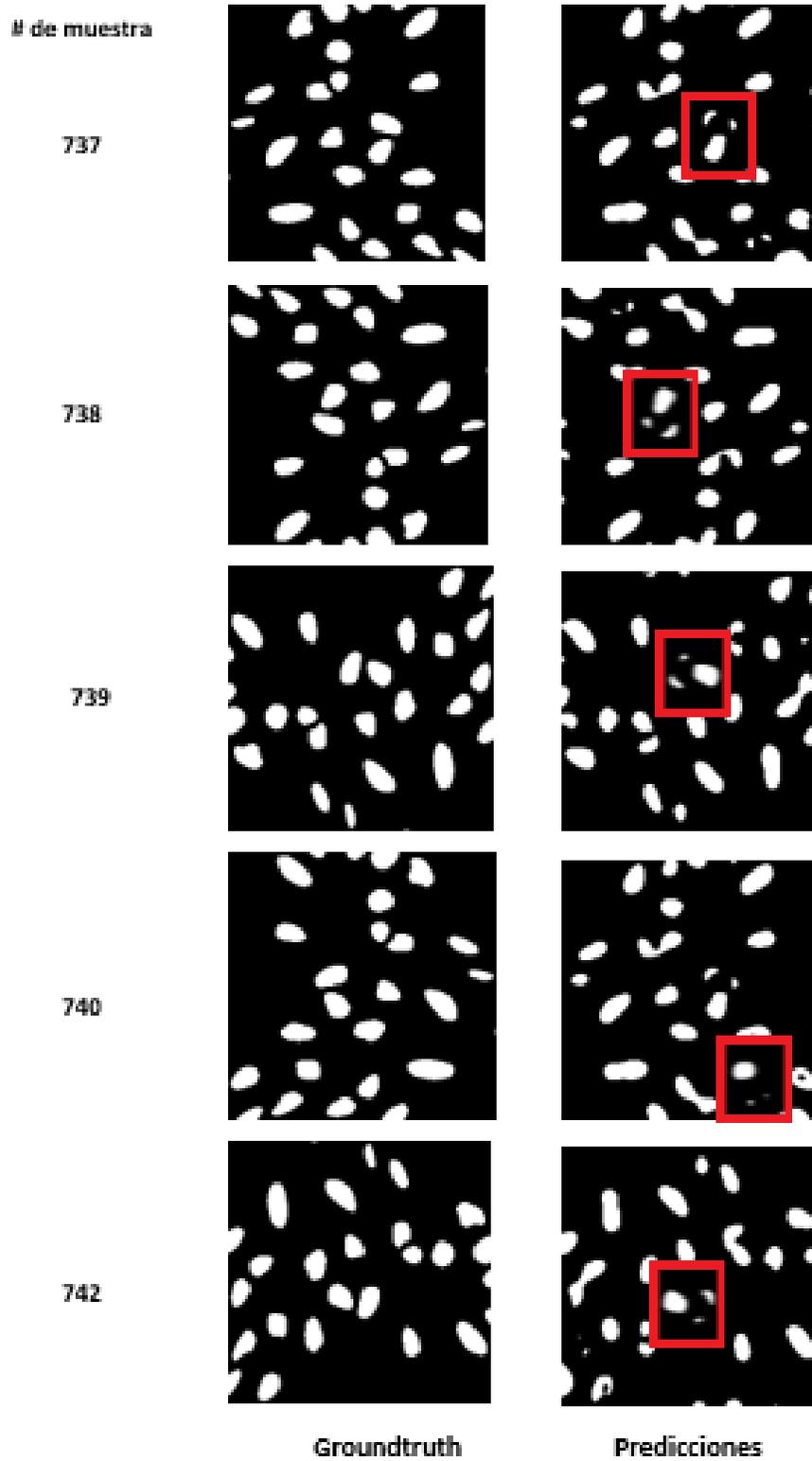


Figura 4.12: Groundtruth e imágenes predecidas por la red U-Net

Además, para las mismas cinco imágenes que se tomaron para la inspección visual, se analizan las métricas para precisión de la segmentación. En la tabla 4.5 se muestran los resultados obtenidos para el set de datos de imágenes sin filtrado.

Tabla 4.5: Métricas para imágenes segmentadas sin filtrado

Número de muestra	Distancia de Jaccard	F Measure
737	0.2880	0.8318
738	0.2919	0.8291
739	0.3018	0.8223
740	0.3010	0.8228
742	0.3085	0.8176
Promedio	0,2982	0,82472
Desviación E.	0,00822	0,005691

Entrenamiento de U-Net con filtrado no lineal

Por otra parte, para imágenes de entrada con filtrado no lineal, con los parámetros que se seleccionaron, $h = 0$ y $\lambda = 3$, se ejecutó el modelo de aprendizaje automático y se obtuvieron los resultados de la figura 4.13, la cual muestra predicciones que se ajustan al contenido semántico del groundtruth. En los recuadros en verde se hizo énfasis en estructuras mejor definidas que representan las ubicaciones espaciales de las células dentro de las imágenes. Para este caso, también se tomaron las mismas cinco imágenes para analizar las métricas de precisión de segmentación. Los resultados se visualizan en la tabla 4.6 los cuales fueron calculados mediante un script en Matlab. Debe considerarse que la métrica F measure representa la similitud de una predicción con respecto a su etiqueta o groundtruth y la distancia de Jaccard mide la disimilitud de las predicciones con respecto a su referencia.

Tabla 4.6: Métricas para imágenes segmentadas con filtrado

Número de muestra	Distancia de Jaccard	F Measure
737	0.0636	0,9672
738	0.0841	0,9561
739	0.0843	0,9560
740	0.0841	0,9561
742	0.0838	0,9563
Promedio	0,0800	0,9583
Desviación E.	0,00916	0,00495

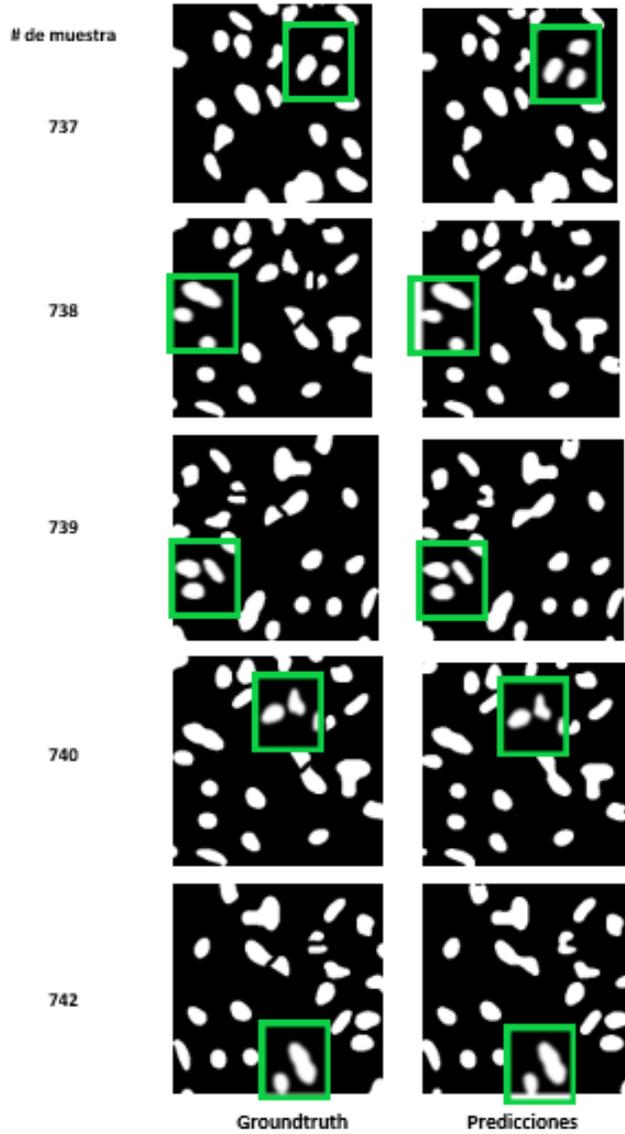


Figura 4.13: Groundtruth e imágenes predecidas por la red U-Net utilizando filtrado

Análisis del impacto del preprocesamiento en la red U-Net

Al comparar las tablas 4.5 y 4.6 se pueden establecer comparaciones en términos de las métricas de precisión de segmentación, F Measure y distancia de Jaccard. Debe apreciarse que para ambos casos, la desviación estándar siempre es menor al 1%, lo cual brinda confiabilidad en los datos que se obtuvieron.

Para relacionar las métricas de segmentación de F Measure y poder realizar una comparación entre prescindir y utilizar el filtrado lineal, se implementó la ecuación 4.1. Donde la variable $índiceIF$ corresponde al valor de la métrica F Measure para las predicciones donde se utiliza filtrado no lineal, e $índiceISF$ donde no.

$$\%mejoramiento = 100 * \frac{\text{índice}IF - \text{índice}ISF}{\text{índice}IF} \quad (4.1)$$

Al evaluarlo numéricamente, con los valores promedios de la métrica F measure, se determina que se aumenta la similitud de las estimaciones de células con respecto al groundtruth en aproximadamente 14 % cuando se usa imágenes filtradas en vez de las originales. Dicho en otras palabras, se puede afirmar que el uso de filtrado no lineal incrementa la precisión de la segmentación semántica de la red U-Net en un 14 %.

Tomando como referencia los valores promedios de la distancia de Jaccard de las tablas 4.5 y 4.6, la diferencia entre las predicciones y el groundtruth a la salida de la CNN con imágenes de entrada sin preprocesar, es de 29.82 %.

Por otra parte, con el filtrado se pasa a un valor de 8 %; realizando una simple operación de diferencia, se obtuvo que el uso de preprocesamiento reduce la disimilitud de las predicciones con su referencia en un 21.82 %.

4.7.1. Precisión de segmentación con U-Net versus técnicas clásicas

Otro de los puntos fundamentales que se deben comprobar en el proyecto, es que el enfoque de aprendizaje profundo supera los resultados en cuanto a precisión que se pueden alcanzar con técnicas clásicas de procesamiento digital de imágenes (DIP). Debido a que las comparaciones se realizaron con métricas para cinco imágenes de muestra, se obtendrá la misma cantidad de imágenes con dimensiones 256×256 , pero procesadas con el enfoque clásico, específicamente ecualización de histogramas para mejora de contraste y umbralización con la técnica de Otsu para obtener una imagen binaria. El resultado de dicho proceso se muestra en la figura 4.14, donde a su vez, se compara con su groundtruth y denota poca fineza en la segmentación, sobre todo en la demarcación de bordes.

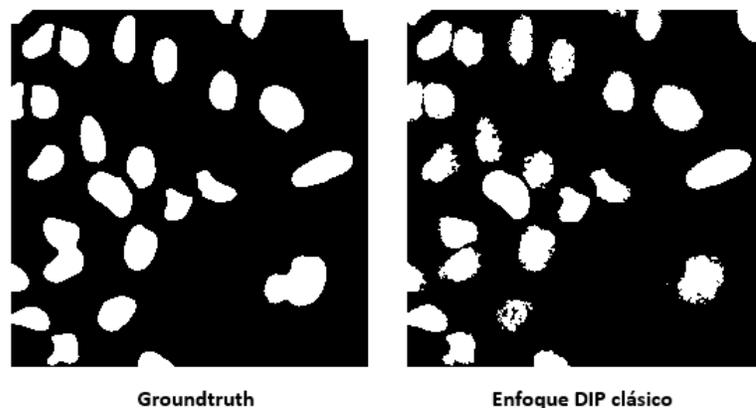


Figura 4.14: Groundtruth comparado con imagen segmentada mediante un enfoque clásico de procesamiento digital de imágenes.

A las cinco imágenes procesadas de forma clásica, se les calcularon las métricas para precisión de segmentación, F measure y distancia de Jaccard, los cuales se muestran en la tabla 4.7.

Tabla 4.7: Métricas de segmentación para técnica clásica de DIP

Número de muestra	Distancia de Jaccard	F Measure
1	0,2110	0,8821
2	0,2325	0,8684
3	0,2512	0,8564
4	0,1813	0,9003
5	0,1990	0,8895
Promedio	0,2150	0,87934
Desviación E.	0,027497	0,0172923

En la subsección anterior se habían encontrado coeficientes de 0.9583 y 0.08 para las métricas F measure y distancia de Jaccard respectivamente. Para el procesamiento digital de imágenes clásico, en la tabla 4.7, se obtuvieron 0.87934 y 0.2150 para F measure y Jaccard, en ese orden. Interpretando la métrica F measure, que indica similitud entre datos, se observa que el enfoque clásico presenta menos similitud. Para cuantificar el porcentaje de mejoramiento del enfoque de aprendizaje profundo sobre la técnica clásica, se plantea la ecuación 4.2. Las variables denominadas *índiceEP* significan la métrica del enfoque de aprendizaje profundo y la variable *índiceEC* corresponde al enfoque clásico.

$$\%mejora = 100 * \frac{\text{índiceEP} - \text{índiceEC}}{\text{índiceEP}} \quad (4.2)$$

Al sustituir las variables por los valores promedio de F measure, se obtiene para este set de datos, el enfoque de arquitectura de aprendizaje profundo mejora la estimación de células en un 8.24% en comparación con una técnica clásica de procesamiento digital de imágenes.

Al analizar la métrica de distancia de Jaccard, realizando una operación de diferencia, se afirma que la arquitectura U-Net, reduce la disimilitud de las estimaciones con respecto a su groundtruth, en un 13.5% sobre los resultados que se obtienen con el estilo clásico.

4.7.2. Validación de resultados mediante pruebas de concepto

Para las pruebas de concepto se utilizará una misma imagen para validar tanto el filtrado como el no uso de él. Se procede a realizar pruebas de concepto sobre la red U-Net, se tiene una imagen de entrada con pocas características perceptibles, que sale del bloque de filtrado no lineal, con mejora de contraste, definición de bordes y supresión de ruido. En la figura 4.15 se muestra el resultado de llevar a cabo un filtrado no lineal en U-Net.

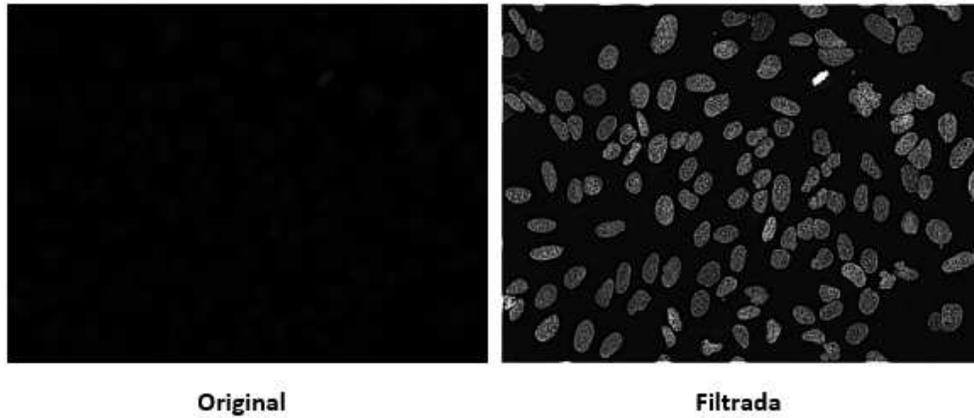


Figura 4.15: Imagen de entrada del sistema y salida de módulo de filtrado

Se subdivide la imagen filtrada en seis imágenes de dimensiones 256×256 . Luego, se entrena un modelo en la red U-Net que aprenda a segmentar imágenes de células de forma automática, la cual dará como resultado imágenes con predicciones que si se combinan de la forma en que fueron divididas, dan como resultado la imagen segmentada original. En la figura 4.16 se muestra la subdivisión de la imagen original en 6 divisiones.

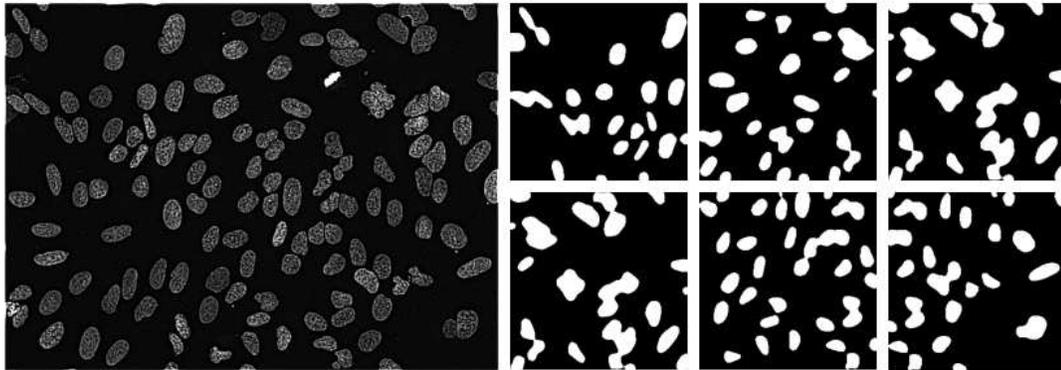


Figura 4.16: Imagen de entrada del sistema y salidas de módulo U-Net
Imagen filtrada con parámetros $h = 0$ y $\lambda = 3$

Se repite el proceso para la misma imagen pero sin filtrado y se muestran los resultados obtenidos en la figura 4.17, nótese las deficiencias perceptuales en las estimaciones realizadas por la red.

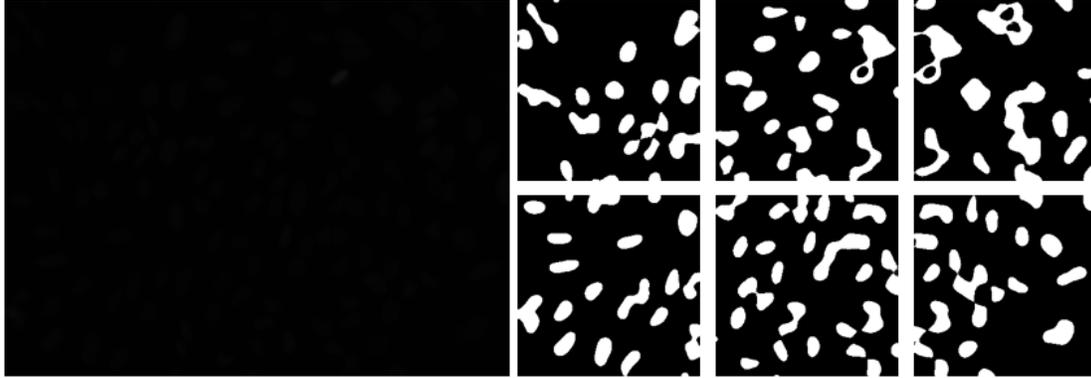


Figura 4.17: Imagen de entrada original y salida de módulo U-Net

Métricas para resultados obtenidos con imágenes de entrada filtradas

Para las seis imágenes binairas que corresponden a predicciones realizadas por la red U-Net cuando se hace filtrado a las imágenes de entrada, que se muestran en la figura 4.16, se calcularon las métricas de segmentación mediante software y se recopilieron los datos de la tabla 4.8.

Tabla 4.8: Métricas para red U-Net con imágenes preprocesadas

Número de muestra	F Measure	Distancia Jaccard
1	0,9288	0,1329
2	0,9102	0,1647
3	0,8799	0,2144
4	0,9552	0,0857
5	0,9434	0,1072
6	0,9444	0,1054
Promedio	0,926983	0,13505
Desviación E.	0,027824	0,047477

Métricas para resultados obtenidos con imágenes sin filtrar

Para el caso de prescindir del filtrado en la red U-Net, cuyas estimaciones se presentan en la figura 4.17, se recopilieron los datos de la tabla 4.9 con los cálculos de sus métricas.

Para comprobar el funcionamiento del sistema de segmentación semántico de células y validar que el uso de filtrado no lineal mejora los resultados de la precisión de segmentación, se comparan los resultados obtenidos en las tablas 4.8 y 4.9, para interpretar las métricas F measure y distancia de Jaccard.

Para el coeficiente promedio de la métrica F measure cuando se utiliza preprocesamiento, se tiene un valor de 0.9270. En el caso contrario, donde no se utiliza filtrado para entrenar la red U-Net, se tiene un coeficiente de 0.8935. Si sustituimos

Tabla 4.9: Métrica F Measure para red U-Net con imágenes sin preprocesar

Número de muestra	F Measure	Distancia Jaccard
1	0,9008	0,1805
2	0,8795	0,215
3	0,8480	0,264
4	0,9176	0,1523
5	0,9011	0,18
6	0,9138	0,1586
Promedio	0,8935	0,1917
Desviación E.	0,025962	0,041666

estos valores en la ecuación 4.3, donde las variables índiceIF corresponde al valor del coeficiente de F measure para la imagen filtrada y índiceISF para la imagen de entrada sin filtrar, encontramos el porcentaje de mejoramiento que tuvo la red para estimar una imagen individual, cuyo valor sería 3.6% de superioridad en el ajuste de la predicción con respecto a su groundtruth.

$$\% \text{mejoramiento} = 100 * \frac{\text{índiceIF} - \text{índiceISF}}{\text{índiceIF}} \quad (4.3)$$

Con las distancias de Jaccard, cuando se utiliza filtrado se tiene un coeficiente de 0.1350, en el caso contrario, se tiene el valor de 0.1917. Esta métrica indica que tan distintas son las estimaciones de la red con respecto a su referencia, y si se aplica una operación de diferencia sencilla, se tiene que el uso de filtrado reduce la disimilitud de las predicciones de células en 5.67% en una imagen individual.

4.7.3. Validación de resultados con treinta imágenes

En este apartado se aumenta la cantidad de imágenes a las cuales se les aplican las métricas de precisión de segmentación de cinco imágenes a treinta, lo cual corresponde a un 10% del set de imágenes de la etapa de validación de la red U-Net.

Para las predicciones hechas por U-Net para la segmentación de células, con treinta imágenes con filtrado y utilizando la métrica F measure, se obtuvo una precisión de 94.91% contra 89,08% cuando se prescinde del procesamiento previo. Bajo la metodología para calcular el porcentaje de mejora que se ha aplicado, el uso de procesamiento en las imágenes de entrada de una etapa de aprendizaje profundo mejora la precisión de los resultados en un 6.14%.

Por su parte, con la métrica de distancia de Jaccard se obtuvo una disimilitud de las predicciones hechas por U-Net de 0.0967% cuando se utilizó filtrado, y un valor de 0.1965 cuando no se usó. Esto corrobora que el uso del filtro DNLM mejora la precisión de la segmentación semántica de células aproximadamente 10% bajo esta métrica.

El coeficiente de F measure para treinta imágenes procesadas con un enfoque tradicional fue de 83.7%. Si se calcula el porcentaje de mejora del enfoque de aprendizaje profundo sobre el enfoque clásico se obtiene una mejoría del 11.81% en cuanto a la precisión de segmentación al utilizar el sistema propuesto.

La distancia de Jaccard para las imágenes segmentadas con procesamiento digital de imágenes tradicional muestran un coeficiente 0.2775, lo cual indica un 27.75% de diferencia entre las predicciones y su referencia esperada, por otra parte se tiene 18% al utilizar deep learning e indica que las predicciones tienen mejor ajuste al groundtruth.

Para una visualización clara de la precisión alcanzada por el sistema implementado, usando como referencia la métrica F measure, se presenta la figura 4.18, en la cual se aprecia que el sistema U-Net con filtrado de imágenes obtiene los mejores resultados al acercarse más al caso de ideal de segmentación perfecta para todas las muestras.

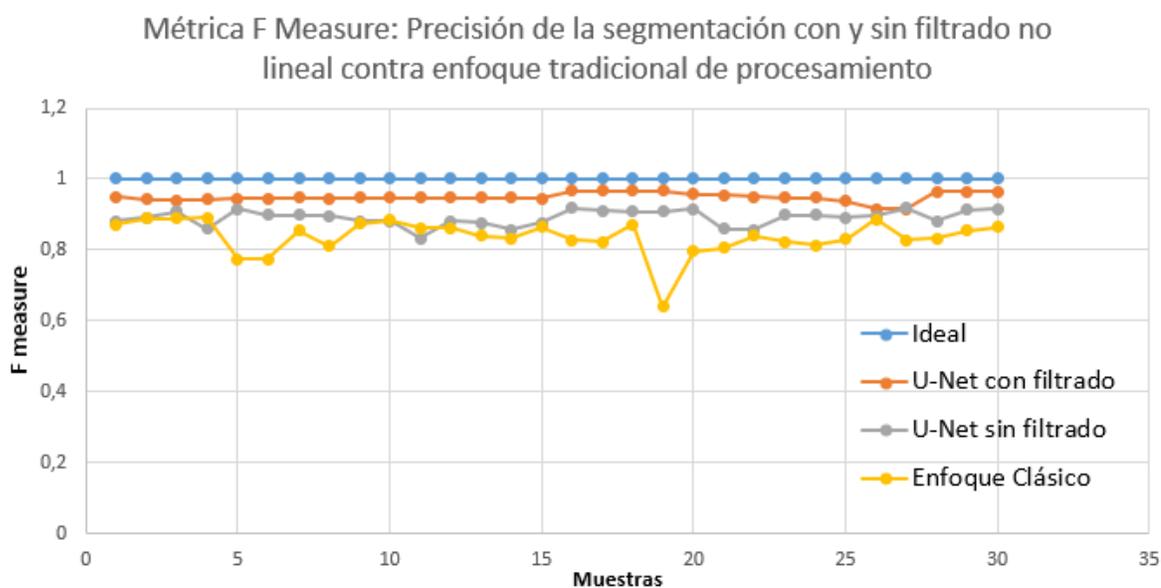


Figura 4.18: Precisión de segmentación con métrica F measure

Para la distancia de Jaccard, el caso ideal sería una diferencia nula entre las predicciones y su groundtruth, en la figura 4.19 se muestra que el sistema propuesto tiene menor disimilitud y se evidencian los pobres resultados que se obtuvo la segmentación de células mediante enfoques clásicos de procesamiento digital de imágenes.

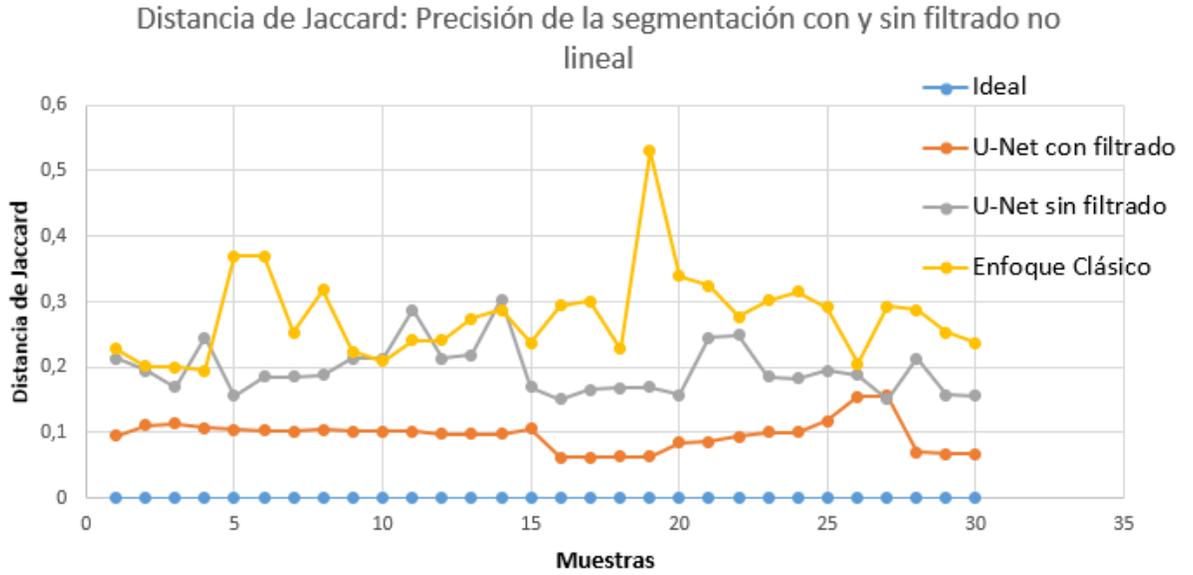


Figura 4.19: Precisión de segmentación con métrica distancia de Jaccard

Conclusiones

La solución seleccionada para el diseño e implementación del sistema de segmentación de células, compuesta de filtrado no lineal y una arquitectura de aprendizaje profundo, fue exitosa por cuanto cumplió los requerimientos mínimos planteados: es capaz de reconocer objetos a partir de imágenes de bajo contraste, clasifica correctamente los píxeles correspondientes a células y es reproducible con confiabilidad estadística para distintos sets de datos de células.

Se comprobó que el uso de pre-procesamiento en las imágenes de entrada de la arquitectura U-Net aumenta la precisión de la segmentación semántica con respecto al set de datos de entrada sin modificaciones en sus características perceptuales espaciales. Esto se verificó, al ser evaluada la precisión de la segmentación mediante las métricas F measure y distancia de Jaccard.

Para la métrica F measure, se obtuvo un porcentaje de mejora del 14 % cuando se comparó la segmentación de la red U-Net utilizando 5 imágenes con y sin filtrado. Cuando se expandió el número de muestras para comparación de 5 a 30, se obtuvo una mejoría cercana al 6 %.

En cuanto a la distancia de Jaccard, al realizar la comparación con 5 imágenes, se mejoró la precisión en aproximadamente 22 % y para muestras de 30 imágenes, la mejoría fue del 10 %.

Al comparar la precisión de segmentación semántica de un método clásico de procesamiento digital de imágenes, con respecto al enfoque de aprendizaje profundo, utilizando la métrica F measure para 5 imágenes bien identificadas con su respectivo groundtruth, se obtuvo un porcentaje de mejora del 8 % a favor de la red U-Net; además, al aumentar la cantidad de imágenes de comparación a 30 unidades, la mejoría fue cercana al 12 %. En relación a lo anterior, se confirma el hecho que las técnicas de *deep learning* dominan el estado del arte en el tema de segmentación superando los resultados que se pueden obtener mediante enfoques tradicionales.

Por otra parte, las pruebas de concepto realizadas sobre el sistema el de segmentación celular propuesto e implementado, confirman mediante las métricas F measure y distancia de Jaccard, que inclusive para una sola imagen particular, se tiene mejor precisión de segmentación al utilizar filtrado no lineal en las imágenes de entrada de la arquitectura U-Net.

Otra de las conclusiones relevantes del trabajo es que según los parámetros h y λ del algoritmo DNLM que se escogieron para preprocesar las imágenes de entrada, el aporte predominante que da mejores resultados de segmentación de células, es la demarcación de bordes provisto por la función Unsharpening Mask; el ruido Gaussiano, en principio, parece no afectar la red U-Net.

Además, el desafío de la identificación de traslapes de células no se resolvió a pesar de las modificaciones que se realizaron sobre el groundtruth. El índice de Dice como función de optimización para la red U-Net, se concentra en la predicción de contornos demarcados con respecto al background.

Recomendaciones y trabajo futuro

A raíz del análisis de las observaciones realizadas como parte de este trabajo, se plantean criterios de recomendación para la reproducibilidad del sistema implementado.

Se recomienda aumentar el rango de los parámetros del filtro DNLM (λ y h), y experimentar con motivo de determinar si existe una combinación que incremente los resultados de la segmentación de células que no haya sido considerada en este proyecto.

Se sugiere realizar un análisis de varianza (ANOVA) para todas las combinaciones de parámetros del filtro DNLM que se plantearon, para determinar de que forma es afectado el índice de Dice por las interacciones de los parámetros. Esto no fue pudo ser realizado en este trabajo debido al acceso restringido que se tuvo al clúster Kabré en etapas finales del proyecto, debido a trabajos de mantenimiento sobre el sistema computacional por parte del CeNAT.

Es recomendable experimentar con otras redes neuronales convolucionales para segmentación, tales como Segnet o MIMO-Net, para establecer comparaciones y relaciones de costo-beneficio con respecto a la aquitectura U-Net.

Como trabajo futuro, se planea incursionar en una técnica de pre-procesamiento previa a la arquitectura de aprendizaje automático, pero posterior al filtrado, la cual se denomina *superpíxeles* [28] y consiste en un nuevo espacio de representación para la imagen, basado en la técnica de clústers, en la cual se agrupan los píxeles que poseen una intensidad similar. De forma tal, que se forma una sobre-segmentación de estructuras que pretende romper la rigidez de la estrucutra clásica de ubicación de píxeles en dos coordenadas. En artículos recientes [29], se ha demostrado que la utilización de esta técnica incrementa la precisión de la segmentación y aumenta la eficiencia computacional del sistema donde se aplica.

Además, se pretende sustituir la función de optimización del índice de Dice por una métrica donde se le de más peso a la idenficación de los traslapos de células.

Bibliografía

- [1] A. Singh, “Review article digital change detection techniques using remotely-sensed data,” *International journal of remote sensing*, vol. 10, no. 6, pp. 989–1003, 1989.
- [2] S. J. Walsh, D. R. Butler, and G. P. Malanson, “An overview of scale, pattern, process relationships in geomorphology: a remote sensing and gis perspective,” *Geomorphology*, vol. 21, no. 3-4, pp. 183–205, 1998.
- [3] S. Hay, “An overview of remote sensing and geodesy for epidemiology and public health application,” *Advances in parasitology*, vol. 47, pp. 1–35, 2000.
- [4] C. Soeller and M. Cannell, “Examination of the transverse tubular system in living cardiac rat myocytes by 2-photon microscopy and digital image-processing techniques,” *Circulation research*, vol. 84, no. 3, pp. 266–275, 1999.
- [5] M. Mahesh, “Fundamentals of medical imaging,” *Medical Physics*, vol. 38, no. 3, pp. 1735–1735, 2011.
- [6] N. McGranahan and C. Swanton, “Clonal heterogeneity and tumor evolution: past, present, and the future,” *Cell*, vol. 168, no. 4, pp. 613–628, 2017.
- [7] C. J. Lord and A. Ashworth, “The dna damage response and cancer therapy,” *Nature*, vol. 481, no. 7381, p. 287, 2012.
- [8] S. P. Jackson and J. Bartek, “The dna-damage response in human biology and disease,” *Nature*, vol. 461, no. 7267, p. 1071, 2009.
- [9] M. E. Davis, “Glioblastoma: overview of disease and treatment,” *Clinical journal of oncology nursing*, vol. 20, no. 5, p. S2, 2016.
- [10] R. C. Gonzalez and R. E. Woods, “Digital image processing,” 2012.
- [11] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.
- [12] X. Ji, D. Zhang, Z. Guo, and B. Wu, “Image denoising via nonlinear hybrid diffusion,” *Mathematical Problems in Engineering*, vol. 2013, 2013.
- [13] S. Calderón, “Introducción al reconocimiento de patrones : Preprocesamiento de señales,” 2017.
- [14] J. E. Kyprianidis, “Image and video abstraction by multi-scale anisotropic kuwahara filtering,” in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering*, pp. 55–64, ACM, 2011.

- [15] L. P. Yaroslavsky, *Digital picture processing: an introduction*, vol. 9. Springer Science & Business Media, 2012.
- [16] S. Calderón, A. Sáenz, R. Mora, F. Siles, I. Orozco, and M. Buemi, “Dewaff: A novel image abstraction approach to improve the performance of a cell tracking system,” in *Bioinspired Intelligence (IWOB), 2015 4th International Work Conference on*, pp. 81–88, IEEE, 2015.
- [17] A. Buades, B. Coll, and J.-M. Morel, “A non-local algorithm for image denoising,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 60–65, IEEE, 2005.
- [18] A. Levin, “Blind motion deblurring using image statistics,” in *Advances in Neural Information Processing Systems*, pp. 841–848, 2007.
- [19] N. M. Zaitoun and M. J. Aqel, “Survey on image segmentation techniques,” *Procedia Computer Science*, vol. 65, pp. 797–806, 2015.
- [20] M. B. Christopher, *PATTERN RECOGNITION AND MACHINE LEARNING*. Springer-Verlag New York, 2016.
- [21] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.
- [22] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, “A review on deep learning techniques applied to semantic segmentation,” *arXiv preprint arXiv:1704.06857*, 2017.
- [23] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.
- [24] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.
- [25] A. Lucchi, K. Smith, R. Achanta, V. Lepetit, and P. Fua, “A fully automated approach to segmentation of irregularly shaped cellular structures in em images,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 463–471, Springer, 2010.
- [26] Z. Yao, Z. Zhang, and L.-Q. Xu, “Convolutional neural network for retinal blood vessel segmentation,” in *Computational Intelligence and Design (IS-CID), 2016 9th International Symposium on*, vol. 1, pp. 406–409, IEEE, 2016.
- [27] M. F. Akay, “Support vector machines combined with feature selection for breast cancer diagnosis,” *Expert systems with applications*, vol. 36, no. 2, pp. 3240–3247, 2009.

- [28] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, S. Ssstrunk, *et al.*, “Slic superpixels compared to state-of-the-art superpixel methods,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.
- [29] R. Gadde, V. Jampani, M. Kiefel, D. Kappler, and P. V. Gehler, “Superpixel convolutional networks using bilateral inceptions,” in *European Conference on Computer Vision*, pp. 597–613, Springer, 2016.