

ALGORITMO DE SEGMENTACIÓN CELULAR DE IMÁGENES  
DE PAPANICOLAOU PARA EL APOYO DIAGNÓSTICO DE  
CÁNCER DE CUELLO UTERINO



Jean Carlos Betancourt Nazareno  
Yeiner Alexander Imbachi Gómez

UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERIA ELECTRÓNICA Y  
TELECOMUNICACIONES

JUNIO 2023

ALGORITMO DE SEGMENTACIÓN CELULAR DE IMÁGENES  
DE PAPANICOLAOU PARA EL APOYO DIAGNÓSTICO DE  
CÁNCER DE CUELLO UTERINO



Jean Carlos Betancourt Nazareno

Yeiner Alexander Imbachi Gómez

Documento presentado a  
Programa de Ingeniería en Automática Industrial  
como requisito para optar por el título de

INGENIERO EN AUTOMÁTICA INDUSTRIAL

Director(a):  
MSc. Elena Muñoz España

UNIVERSIDAD DEL CAUCA  
FACULTAD DE INGENIERIA ELECTRÓNICA Y  
TELECOMUNICACIONES

JUNIO 2023

# Resumen

El análisis de frotis cervical, una técnica pionera descrita por Papanicolaou, ha demostrado ser efectiva en la detección temprana de patologías y se utiliza ampliamente en sistemas CAD para el diagnóstico automatizado. En este estudio, se emplearon diversas técnicas de segmentación y clasificación de imágenes PAP con el objetivo de apoyar el diagnóstico del cáncer de cuello uterino. Se utilizaron técnicas de aprendizaje no supervisado, como KMEANS y MEANSHIFT, para llevar a cabo la segmentación del núcleo celular. Estas técnicas se basan en el concepto de agrupamiento para fusionar regiones de superpíxeles y lograr la segmentación deseada. Además, se empleó la red UNET, un modelo de aprendizaje supervisado, para la segmentación tanto del núcleo como del citoplasma. Para la realización de este estudio, se utilizó la base de datos de HERLEV, que consta de 917 imágenes junto con sus correspondientes segmentaciones y etiquetas. Esta base de datos facilitó el proceso de aprendizaje y evaluación de las técnicas utilizadas en la investigación. En cuanto a la clasificación, se emplearon cuatro clasificadores: DT, SVM, KNN y ANN, para resolver problemas de clasificación binaria y de tres clases. Para mejorar el rendimiento de la clasificación, se aplicaron técnicas como PCA y FEATUREWIZ para obtener características significativas de las células. Los resultados obtenidos mostraron que la red UNET fue altamente efectiva en la segmentación tanto del citoplasma como del núcleo, obteniendo índices de DICE de  $0.91 \pm 0.044$  y  $0.90 \pm 0.117$ , respectivamente. En cuanto a la clasificación binaria, el clasificador ANN alcanzó una exactitud del 98 %, mientras que para la clasificación de tres clases, el clasificador KNN obtuvo una precisión del 89 %. Por tanto, este estudio demuestra la eficacia de las técnicas de segmentación y clasificación utilizadas en el análisis de frotis cervical, proporcionando resultados prometedores para el apoyo al diagnóstico de cáncer de cuello uterino.

# Abstract

Cervical smear analysis, a pioneering technique described by Papanicolaou, has proven to be effective in the early detection of pathologies and is widely used in CAD systems for automated diagnosis. In this study, various segmentation and classification techniques for PAP images were employed with the aim of supporting the diagnosis of cervical cancer. Unsupervised learning techniques such as KMEANS and MEANSHIFT were utilized to perform cell nucleus segmentation. These techniques rely on the concept of clustering to merge superpixel regions and achieve the desired segmentation. Additionally, the supervised learning model UNET was employed for both nucleus and cytoplasm segmentation. The HERLEV database, consisting of 917 images along with their corresponding segmentations and labels, was used for this study. This database facilitated the learning and evaluation process of the techniques employed in the research. Regarding classification, four classifiers, namely DT, SVM, KNN, and ANN, were employed to solve binary and three-class classification problems. Techniques such as PCA and FEATUREWIZ were applied to enhance classification performance and extract significant cell features. The obtained results demonstrated that the UNET network was highly effective in segmenting both the cytoplasm and nucleus, achieving DICE indices of  $0.91 \pm 0.044$  and  $0.90 \pm 0.117$ , respectively. In terms of binary classification, the ANN classifier achieved an accuracy of 98%, while for three-class classification, the KNN classifier obtained a precision of 89%. Thus, this study showcases the efficacy of the segmentation and classification techniques used in cervical smear analysis, offering promising results for supporting the diagnosis of cervical cancer.

**Keywords**— Cervical cancer, PAP image processing, machine learning.

## Agradecimientos

El presente trabajo va dirigido como una expresión de gratitud a las personas que estuvieron presentes con su voz de aliento. Expresamos nuestro profundo agradecimiento a nuestra directora de tesis, por su orientación experta, apoyo y dedicación que fueron fundamentales en el desarrollo de este trabajo.

También agradecer a la Universidad del Cauca por brindarnos la oportunidad de llevar a cabo este proyecto y por proporcionar los recursos necesarios para su realización. La calidad de la educación y los conocimientos adquiridos son de gran importancia para nuestra formación personal y profesional.

Asimismo, expresamos nuestra gratitud a familiares y padres por su infinito amor y apoyo incondicional. Su constante apoyo nos dió la fuerza necesaria para superar desafíos y completar esta etapa académica.

A todos los mencionados y a aquellos que han contribuido de alguna manera en este proyecto, les estamos profundamente agradecidos. Su apoyo deja una huella duradera en nuestra trayectoria académica.

# Glosario

**ANN** redes neuronales artificiales. 8, 38, 71, 73, 86, 87, 89–91

**ASM** segundo momento angular. 26, 67

**AUC** área bajo la curva. 98

**CAD** diagnóstico asistido por computador. 1, 2, 5

**CNN** redes neuronales convolucionales. 7, 48, 98, 99, 101

**DBSCAN** agrupamiento espacial de aplicaciones con ruido basado en la intensidad.  
6, 94, 96

**DIC** contraste de interferencia diferencial. 6, 96, 98

**DL** aprendizaje profundo. 7

**DT** árbol de decisión. 6, 71–73

**FCM** C medias difusas. 96

**FN** falsos negativos. 41, 75

**FP** falsos positivos. 41, 42, 75

**GLCM** matriz de coocurrencia de niveles de grises. 7, 24, 25, 67, 70

**HMLS** conjunto de nivel de banda estrecha de mediana jerárquica. 96

**HSIL** lesión intraepitelial escamosa de alto grado. 84, 85, 87, 89, 91, 92

**IA** inteligencia artificial. 94

**KMEANS** k-medias. 17, 18, 53, 76, 77, 79, 96

**KNN** k-vecino más cercano. 6, 8, 31, 32, 71, 84–92, 101

**LSIL** lesión intraepitelial escamosa de bajo grado. 84–87, 89, 91, 92

**MEANSHIFT** agrupamiento por desplazamiento medio. 8, 18, 19, 50, 51, 77, 79, 96

**ML** machine learning. 7

**MRF** campo aleatorio de Markov. 6, 96

**NC** centroide más cercano. 6

**PAP** prueba de Papanicolaou. 2, 3, 7–9, 53, 57, 98, 100

**PCA** análisis de componentes principales. 29, 30, 68

**ReLU** unidad lineal rectificadora. 20, 21, 73

**RGB** rojo, verde, azul. 27

**ROI** región de interés. 6

**SLIC** agrupamiento iterativo lineal simple. 6, 15, 49, 96

**SULOV** búsqueda de lista de variables no correlacionadas. 28

**SVM** máquinas de soporte vectorial. 7, 8, 32, 33, 35, 71, 72, 96, 101

**TWS** segmentación WEKA entrenable. 96

**UNET** segmentación semántica UNET. 20, 57–60

**VN** verdaderos negativos. 41, 75

**VP** verdaderos positivos. 41, 75

# Contenido

	Pag.
Introducción .....	1
1. Planteamiento del problema de investigación.....	2
1.1 Problema de investigación .....	3
1.2 Objetivo general .....	3
1.3 Objetivos específicos .....	4
2. Estado del arte .....	5
3. Marco teórico .....	9
3.1 Aspectos biológicos.....	9
3.1.1 Grado patológico .....	10
3.2 Procesamiento de imágenes .....	12
3.2.1 Preprocesamiento .....	13
3.2.2 Segmentación .....	14
3.3 Descriptores de forma y de textura .....	21
3.3.1 Características de forma .....	21
3.3.2 Características de textura.....	24
3.3.3 Características de color .....	27
3.4 Selección y extracción de características.....	27
3.4.1 Selección basada en correlación.....	28
3.4.2 FEATUREWIZ .....	28
3.4.3 Análisis de componentes principales PCA .....	29
3.5 Aprendizaje automático supervisado .....	31
3.5.1 KNN.....	31
3.5.2 SVM .....	32
3.5.3 Árbol de decisión .....	36
3.5.4 Redes neuronales artificiales ANN.....	38
3.6 Medidas del error .....	41
3.6.1 Medidas estadísticas de error .....	41
3.6.2 Índice de DICE .....	42
3.6.3 Distancia de Jaccard .....	43
4. Desarrollo o solución .....	44
4.1 Imágenes de entrada: conjunto de datos HERLEV .....	45
4.2 Preprocesamiento.....	47

4.3	Segmentación celular .....	50
4.3.1	Solución 1 .....	50
4.3.2	Solución 2 .....	53
4.3.3	Solución 3 .....	55
4.3.4	Segmentación final.....	61
4.4	Extracción de características .....	64
4.5	Selección de características.....	68
4.6	Clasificación.....	71
4.6.1	K vecino más cercano KNN.....	71
4.6.2	Máquinas de soporte vectorial SVM .....	72
4.6.3	Árbol de decisión .....	72
4.6.4	Redes neuronales .....	73
5.	Pruebas y resultados .....	74
5.1	Evaluación de la segmentación .....	74
5.1.1	Métricas de evaluación del algoritmo de segmentación .....	75
5.1.2	Tiempos de ejecución de los algoritmos de segmentación.....	78
5.1.3	Selección del algoritmo de segmentación.....	79
5.2	Evaluación clasificación .....	82
5.2.1	Resultados de clasificación con características del núcleo.....	83
5.2.2	Resultados de clasificación con características núcleo/citoplasma..	86
5.2.3	Selección del modelo de clasificación para dos y tres clases .....	89
5.2.4	Tiempos de ejecución de la clasificación .....	93
6.	Discusión .....	94
6.1	Métodos propuestos.....	94
6.2	Comparación de rendimiento de la segmentación .....	95
6.3	Comparación de rendimiento del clasificador .....	97
7.	Conclusiones y trabajos futuros .....	100
7.1	Conclusiones .....	100
7.2	Trabajos futuros .....	102
8.	Anexos .....	104
8.1	Anexo 1: Detalles del algoritmo de segmentación y clasificación .....	104
8.1.1	Descripción del algoritmo .....	104
8.1.2	Instalación de paquetes para el entorno de desarrollo.....	115
8.2	Prueba de segmentación con imágenes PAP .....	117
	Referencias.....	118

# Indice de figuras

FIGURA	Pag.
3.1 Tinción de muestras [1].....	9
3.2 Ubicación células columnares y células escamosas [2]. .....	10
3.3 Grado patológico [3]. .....	10
3.4 Epitelio escamoso superficial; epitelio escamoso intermedio [2]. .....	11
3.5 Epitelio columnar [2].....	11
3.6 Displasia escamosa no queratinizante leve; displasia escamosa no queratinizante moderada; displasia escamosa no queratinizante severa [2].	12
3.7 Carcinoma in situ [2]. .....	12
3.8 Representación ventana 3x3 y 5x5 [4]. .....	14
3.9 Contornos de superpíxeles generados (fuente: autores).....	16
3.10 Contornos del agrupamiento <i>Watershed</i> (fuente: autores). .....	17
3.11 Ajuste de centroides KMEANS (fuente: autores). .....	17
3.12 Convergencia de centroides (fuente: autores). .....	19
3.13 Estructura UNET [5]. .....	20
3.14 Ángulos cuyo diámetro de Feret es máximo y mínimo [4].....	25
3.15 Ejemplo de características PCA (fuente: autores). .....	30
3.16 Algoritmo de clasificación KNN (fuente: autores). .....	32
3.17 Hiperplano de separación [6]. .....	33
3.18 Problema lineal y no-linealmente separable (fuente:autores).....	36
3.19 Árbol de clasificación a la izquierda y partición del espacio a la derecha [7]. .....	37

3.20	Estructura general de una ANN (fuente: autores). . . . .	38
3.21	Estructura de perceptrón Multicapa (fuente: autores). . . . .	39
3.22	Ejemplo de evaluación de zonas segmentadas [8]. . . . .	41
4.1	Metodología general (fuente: autores). . . . .	44
4.2	Estructura de preprocesamiento y segmentación para las soluciones 1, 2 y 3 (fuente: autores). . . . .	47
4.3	Comparación entre filtros (fuente: autores). . . . .	48
4.4	Fase de preprocesamiento (fuente: autores). . . . .	49
4.5	Agrupamiento por MEANSHIFT (fuente: autores). . . . .	51
4.6	Análisis de intensidad a partir del histograma (fuente: autores). . . . .	52
4.7	Estructura de segmentación usando KMEANS (fuente: autores). . . . .	53
4.8	Análisis de intensidad a partir del histograma con KMEANS (fuente: autores). . . . .	54
4.9	Estructura de segmentación usando UNET (fuente: autores). . . . .	55
4.10	Análisis del balance de clases (fuente: autores). . . . .	56
4.11	Procesamiento de imagen de referencia del núcleo y citoplasma (fuente: autores). . . . .	58
4.12	Segmentación de referencia en la imagen original (fuente: autores). . . . .	59
4.13	Métricas de entrenamiento UNET para núcleo (fuente: autores). . . . .	59
4.14	Métricas de entrenamiento UNET para citoplasma (fuente: autores). . . . .	60
4.15	Segmentación en imágenes de prueba (fuente: autores). . . . .	61
4.16	Ejemplos de pseudonúcleos/pseudocitoplasmas y orificios de la fase de presegmentación (fuente: autores). . . . .	62
4.17	Segmentación de núcleo y citoplasma (fuente: autores). . . . .	62
4.18	Segmentación final (fuente: autores). . . . .	64

4.19	Ejemplo área y perímetro en píxeles. Área = 867, perímetro = 110 (fuente: autores).....	64
4.20	Ejemplo elipse (fuente: autores).....	65
4.21	Ejes mayor y menor de la elipse (fuente: autores). ....	65
4.22	Cuadrado que se ajusta al contorno del objeto (fuente: autores). ....	66
4.23	Relación de aspecto (fuente: autores).....	66
4.24	Diámetro equivalente (fuente: autores). ....	66
4.25	Solidez (fuente: autores). ....	67
4.26	Histograma de niveles de gris solo núcleo (fuente: autores). ....	67
4.27	Espacios de color región segmentada (fuente: autores). ....	68
4.28	Combinación de espacios de características de núcleo para entrenamiento (fuente: autores).....	69
4.29	Combinación de espacios de características de núcleo y citoplasma para entrenamiento (fuente: autores). ....	69
5.1	Imagen de referencia HERLEV [2]. ....	75
5.2	Comparación segmentación KMEANS vs imagen de referencia (fuente: autores). ....	76
5.3	Coeficiente de DICE para segmentación semántica de citoplasma (fuente: autores). ....	80
5.4	Coeficiente DICE para segmentación semantica de núcleo (fuente: autores). 81	
5.5	Distancia de Jaccard para segmentación semántica de núcleo (fuente: autores). ....	81
5.6	Distancia de Jaccard para segmentación semántica de citoplasma (fuente: autores). ....	82
5.7	Matriz de confusión normalizada para 7-PCA con KNN, clasificación binaria (fuente: autores). ....	84
5.8	Matriz de confusión normalizada para PCA y FEATUREWIZ con KNN, tres clases (fuente: autores). ....	86

5.9	Matriz de confusión normalizada para PCA y FEATUREWIZ con KNN, clasificación binaria (N/C) (fuente: autores). . . . .	87
5.10	Matriz de confusión normalizada para KNN con el espacio FEATUREWIZ, 3 clases (N/C) (fuente: autores). . . . .	88
5.11	Curva ROC para clasificadores binarios (fuente: autores). . . . .	90
5.12	Curva ROC para clasificadores de tres clases (fuente: autores). . . . .	92
6.1	Comparación segmentación SAM vs UNET (fuente: autores). . . . .	95
7.1	Pruebas en imágenes de CRIC y SIPaKMeD (fuente: autores). . . . .	103
8.1	Resultados para KMEANS anormalMild50 (fuente: autores). . . . .	107
8.2	Segmentación de núcleo UNET (fuente: autores). . . . .	110
8.3	Segmentación de citoplasma UNET (fuente: autores). . . . .	110
8.4	Prueba de segmentación de imágenes con las 3 técnicas propuestas (fuente: autores). . . . .	117

# Indice de tablas

TABLA	Pag.
3.1 Funciones de activación en redes neuronales artificiales [9]. . . . .	39
4.1 Resumen de clases y tipos de células en el conjunto de datos de HERLEV (fuente: autores). . . . .	46
4.2 Espacio de características para clasificación binaria (fuente: autores). . . . .	70
4.3 Espacios de características para clasificación de 3 clases (fuente: autores). . . . .	70
5.1 Métricas para una sola célula de tipo anormal (fuente: autores). . . . .	76
5.2 Evaluación segmentación (fuente: autores). . . . .	77
5.3 Tiempos globales segmentación (fuente: autores) . . . . .	78
5.4 Rendimiento de clasificación binaria con características de núcleo (fuente: autores). . . . .	83
5.5 Rendimiento de la clasificación de 3 clases con características de núcleo (fuente: autores). . . . .	85
5.6 Rendimiento para clasificación binaria con características de núcleo y citoplasma (fuente: autores). . . . .	86
5.7 Rendimiento de la clasificación 3 clases (N/C) (fuente: autores). . . . .	88
5.8 Comparación de los mejores clasificadores binarios (fuente: autores). . . . .	89
5.9 Comparación de los mejores modelos para 3 clases (fuente: autores). . . . .	91
5.10 Tiempos clasificación (fuente: autores) . . . . .	93
6.1 Comparación de resultados de segmentación (fuente: autores). . . . .	96
6.2 Comparación de los resultados de clasificación con las diferentes propuestas para PAP (fuente: autores). . . . .	98

# Introducción

Los sistemas de diagnóstico asistido por computadora (CAD) son tecnologías que implementan algoritmos y herramientas tecnológicas para analizar e interpretar imágenes médicas. La importancia de estos sistemas radican en la capacidad que tienen para detectar anomalías en los datos o para brindar una segunda opinión a los especialistas. La segmentación es un paso importante en el análisis de células Papanicolaou, ya que logra identificar y aislar las regiones de interés, como el núcleo y el citoplasma de una célula. La segmentación permite obtener mediciones consistentes de las características celulares, como la forma, el tamaño y la textura. Dichas mediciones son utilizadas en el análisis cuantitativo de las células, lo que permite detectar cambios anormales en la morfología celular y finalmente clasificarlas en categorías específicas. Por lo tanto, en esta investigación se desarrollan y evalúan métodos de segmentación y clasificación de células PAP que apoyen a los profesionales de la salud en los procesos de identificación precisa y eficiente de células anormales mediante técnicas de visión por computadora y aprendizaje automático.

Este trabajo de investigación consta de 7 capítulos. En el capítulo 1 se muestra el problema de investigación y el alcance del mismo, a la vez que se hace una breve introducción de los temas más relevantes. El capítulo 2 muestra los trabajos más relevantes que fueron considerados en la revisión del estado del arte. El capítulo 3 provee un marco teórico general entrando en detalle a los temas segmentación, extracción de características y clasificación. El capítulo 4 muestra en detalle la metodología de la solución desarrollada. El capítulo 5 muestra las pruebas y resultados de la segmentación y clasificación. En el capítulo 6 se hace una comparación con propuestas de vanguardia del estado del arte. En el capítulo 7 se presentan las conclusiones obtenidas a partir del estudio y se mencionan las investigaciones futuras.

# 1. Planteamiento del problema de investigación

Los sistemas de diagnóstico asistido por computadora (CAD) a partir de la prueba de Papanicolaou (PAP), son herramientas que permiten analizar automáticamente imágenes de frotis cervical y que brindan apoyo al proceso de detección de cáncer de cuello uterino. Una de las etapas típicas en el análisis automatizado de imágenes PAP es la segmentación, que consiste en la separación de los componentes de interés de la célula, para luego determinar características malignas o anormales, con las cuales se pueda realizar la clasificación. Se han desarrollado numerosas técnicas de segmentación, como las que hacen uso de filtros, detección de bordes y umbrales a partir del histograma de la imagen [10], que logran en su mayoría la segmentación de núcleo, pero presentan problemas de segmentación de citoplasma, dado que muchas veces dicha región tiende a confundirse con el fondo de la imagen, o por su transparencia, es difícil la determinación de un borde o umbral de segmentación del objeto de interés.

Las características ideales de un algoritmo de procesamiento de imágenes PAP, es que pueda hacer el diagnóstico de forma satisfactoria, a partir del análisis de las células, y que sea robusto frente a problemas relacionados con la muestra, como el solapamiento, superposición, moco, y presencia de partículas ajenas a la célula de interés [10]. Es importante tener en cuenta toda la información que se pueda extraer a partir de las imágenes de las células, ya que de ello depende la robustez de un algoritmo de clasificación que realmente pueda apoyar el diagnóstico de dicha enfermedad [11]. La superioridad de un sistema de diagnóstico asistido por computador (CAD) depende en gran medida de los factores que incluyen la precisión del preprocesamiento, la segmentación, y el tipo de conjuntos de datos utilizados [10]. En la actualidad, se han desarrollado algoritmos que facilitan el proceso de diagnóstico imitando la forma en que el especialista realiza el análisis a partir de imágenes de Papanicolau [10][11][12][13]. Generalmente, estos algoritmos se dividen en cuatro fases conocidas como preprocesamiento, segmentación, entrenamiento, clasificación, con los cuales se aborda el problema de detección. Lo ideal es que el algoritmo que se implementa, sea capaz de llevar a cabo cada una de esas etapas de forma satisfactoria.

De acuerdo con [14] [11], uno de los principales retos para el procesamiento de imágenes PAP, es precisamente la dificultad para segmentar correctamente el núcleo y el citoplasma; se encuentran imágenes donde las células presentan solapamiento entre ellas, lo que implícitamente afecta el proceso de obtención de características morfológicas y clasificación. En el mismo sentido, en la mayoría de trabajos enfocados a segmentación, no se consideran las características que se pueden extraer a partir del citoplasma, lo que puede mejorar significativamente la eficiencia para la clasificación. Adicionalmente, uno de los problemas que dificultan el acondicionamiento o preprocesamiento de la imagen se encuentran relacionados con la calidad de las muestras tomadas, las cuales contienen otros elementos diferentes a las células de interés, que dificultan el análisis por computadora [15].

Por tanto, en este trabajo de investigación se propone un algoritmo de procesamiento de imágenes, enfocado en la segmentación de núcleo y citoplasma de las imágenes PAP, haciendo uso del conjunto de datos de HERLEV, que permita apoyar el diagnóstico de cáncer de cuello uterino.

## 1.1 Problema de investigación

Con base en el contexto descrito anteriormente, se formula el problema de investigación.

- ¿Cuáles serán las características de un algoritmo de procesamiento de imágenes PAP que permita la segmentación de células teniendo en cuenta las características morfológicas tanto de citoplasma como del núcleo?.

## 1.2 Objetivo general

- Proponer un algoritmo de procesamiento de imágenes PAP que permita la segmentación de células teniendo en cuenta las características morfológicas tanto de citoplasma y núcleo para el diagnóstico de cáncer de cuello uterino.

### 1.3 Objetivos específicos

- Definir la técnica de segmentación de imágenes que permita separar núcleo y citoplasma celular en imágenes de Papanicolau.
- Establecer el conjunto de características que permita la clasificación del grado patológico de la célula.
- Determinar el rendimiento de la segmentación y clasificación del tipo de célula teniendo en cuenta métricas de evaluación.

## 2. Estado del arte

El cáncer de cuello uterino se produce cuando las células del tejido del cuello uterino crecen y se replican de forma anormal; con ausencia de muerte celular y sin una división celular controlada, las células no pueden llevar a cabo sus funciones principales, conllevando a un tumor maligno [16]. Según estadísticas de la Organización Mundial de la Salud, este tipo de cáncer es el cuarto más común entre las mujeres [17]. En países que cuentan con un índice de desarrollo humano medio-bajo, el cáncer de cuello uterino es el segundo tipo de cáncer más común entre las mujeres y representa gran impacto en la morbilidad y mortalidad de estas [18]. En Colombia, es la primera causa de muerte entre mujeres de 30 a 59 años [19], y en el Cauca presenta una tasa de mortalidad de 10.4 mujeres por cada 100.000 habitantes para el año 2022 [20]. El frotis de Papanicalou ha sido el método más usado durante décadas para detección de cáncer de cuello uterino, donde uno de los objetivos es detectar cualquier alteración potencial precancerosa (displacía cervical) [16]. Esta técnica demostró ser la mejor forma de detectar la patología en sus etapas tempranas, siendo la técnica de detección pionera en el análisis de frotis cervical [21].

En un sistema CAD se identifican las actividades principales que realiza el especialista, para replicarlas en un sistema computacional capaz de analizar y tomar decisiones a partir de unos datos de entrada. En la literatura se encuentran numerosos estudios sobre estos sistemas que, en general, abordan el problema a través de cuatro etapas conocidas como preprocesamiento, segmentación, extracción de características y clasificación, cada una con sus propias técnicas [16].

La umbralización [11] [22] [23] se ha convertido en una de las técnicas más usadas para el proceso de segmentación celular, que generalmente consta de dos etapas, la primera, donde se realiza el acondicionamiento de la imagen de entrada que implica la eliminación de ruido, el suavizado, y mejoras en el contraste de la imagen. La segunda etapa, consiste en establecer un umbral basado en el histograma de intensidad de la imagen que permite realizar la segmentación del objeto de interés. En este paso, se realizan múltiples operaciones morfológicas que permiten la eliminación de partículas mejorando el proceso de segmentación [24].

El estudio realizado en [11], emplea la umbralización a partir de imágenes de contraste de interferencia diferencial (DIC) de cuello uterino, donde utiliza el método de Otsu de valle modificado, haciendo uso de la primera y segunda derivada que ayudan a determinar los valles o los picos de la imagen celular. Dicho estudio se enfoca en la captura de características tanto de núcleo como de citoplasma, mediante la técnica de microscopía cuantitativa sin etiqueta con imágenes DIC, logrando una precisión de clasificación mayor al 90 %, para problemas de dos y tres clases, tomando como prueba las bases de datos de HERLEV y DIC. Sin embargo, el algoritmo no es completamente robusto al no tenerse en cuenta el problema de células superpuestas.

En numerosos trabajos se hace uso de las técnicas de cálculo de superpíxeles para mejorar la caracterización y clasificación celular como se puede comprobar en las siguientes literaturas. En [25] se enfocan principalmente en la detección del núcleo, utilizando los algoritmos agrupamiento iterativo lineal simple (SLIC) y agrupamiento espacial de aplicaciones con ruido basado en la intensidad (DBSCAN) para obtener superpíxeles de la imagen y luego agruparlos, con lo cual se determinan los conjuntos de píxeles que son candidatos para formar núcleos. Igualmente proponen un método de conjunto, en el cual combinan múltiples clasificadores como árbol de decisión (DT), centroide más cercano (NC) y k-vecino más cercano (KNN) resultando en una forma eficiente y robusta con altos valores de desempeño. Sin embargo, este método propuesto requiere de la detección de todos los núcleos de la imagen, de lo contrario, conllevará a un diagnóstico incorrecto. Por otro lado, mantienen una precisión del 98.8 % en la segmentación de núcleos en células superpuestas con una implementación simple, bajo costo computacional y resultados rápidos para la clasificación.

Existen enfoques para segmentar núcleo y citoplasma y otros componentes de la imagen. Por un lado, en [26], se propone el uso de un campo aleatorio de Markov (MRF) basado en superpíxeles para adquirir el núcleo, el citoplasma y el fondo de la imagen, mientras que en [27] se realiza una segmentación de núcleos, grupos celulares y segmentación de citoplasma celulares haciendo uso de una técnica de dos etapas, el cual presenta las siguientes dificultades: grandes grupos de células borrosas con bordes poco claros, que impiden determinar la distancia del citoplasma respecto al núcleo; o que dicho núcleo se encuentre a un lado de la célula, por lo que la determinación de la región de interés (ROI) asociado a los límites de la célula superpuesta sea compleja.

Otra de las técnicas comunes de segmentación celular es por medio del machine learning (ML) [12], cuyos algoritmos permiten la generalización de ciertos comportamientos a partir de una información de entrada que sirven como ejemplo, y luego, esa capacidad de aprendizaje y de anticipación de comportamientos, se utiliza en sistemas que implican clasificación, reconocimiento, y diagnósticos médicos [28]. En ese sentido, el aprendizaje profundo (DL) hace referencia a una característica muy importante de la inteligencia artificial, la cual trata de imitar la forma en que el cerebro humano procesa los datos, asocia patrones, y posteriormente realiza la toma de decisiones [29], como las redes neuronales convolucionales (CNN), que tienen la ventaja de realizar el proceso de detección y clasificación, sin necesidad de realizar acondicionamiento de la imagen de entrada. Sin embargo, este tipo de técnicas requiere un proceso de entrenamiento previo, lo que implica grandes cantidades de datos y complejidad computacional [30].

En [12] se logró realizar la segmentación a través de un algoritmo Weka entrenable, enfocado en la eliminación secuencial de desechos, y que fue probada en la base de datos PAP pública HERLEV. En [31], se propone un método para la extracción de características basado en las capas ocultas de CNN y en matriz de coocurrencia de niveles de grises (GLCM), la cual permite la extracción de características principales, que son fusionadas para el proceso de clasificación por medio de máquinas de soporte vectorial (SVM).

En general, se ha logrado gran avance en el tema de la segmentación celular, y con ello diversas técnicas aplicadas. En [32] se propone un nuevo método de segmentación de núcleos a partir de una imagen completa de PAP con agrupamiento celular, que utiliza directamente la información de profundidad de las imágenes, donde reconoce grupos de células que pueden contener núcleos que se convierten en regiones de interés. Usando la técnica de dominio de árbol se realiza una segmentación de núcleos candidatos, para luego realizar una segmentación fina mediante la eliminación de pseudo-núcleos.

Otra técnica se aplica en [33], donde se aborda el problema de segmentación por el método de corte normalizado mejorado para la segmentación, cuya investigación va enfocada a la detección del límite del citoplasma y el núcleo a partir del análisis de intensidad, y datos espaciales, que posteriormente se convierten en un conjunto de cortes gráficos normalizados que facilitan la detección de cáncer de cuello uterino.

En [13] probaron la segmentación de núcleos usando el algoritmo MEANSHIFT en imágenes PAP. Presentan una solución usando la técnica de morfología matemática flexible, con la cual logran la separación de núcleos superpuestos. En [34] se utiliza un algoritmo de segmentación de núcleos jerárquicos para células cervicales aisladas y superpuestas, basado en el nivel de banda estrecha capaz de segmentar células de bajo contraste de manera eficiente, siendo aplicable en diferentes bases de datos sin requerir una etapa de entrenamiento.

La etapa seguida al proceso de segmentación y extracción de características, es la clasificación. En este punto, las técnicas más usadas implican el aprendizaje de máquina, como lo son las redes neuronales artificiales (ANN), la SVM, y el KNN, los cuales han demostrado tener un buen rendimiento para el problema de la clasificación celular. La literatura [15], desarrolla el proceso de clasificación de frotis de PAP, por medio de técnicas de aprendizaje profundo. El modelo, mejora la robustez al permitir la superposición y plegamiento de diferentes células durante el entrenamiento de la red.

También existen investigaciones que abarcan el proceso completo de diagnóstico, que va desde el preprocesamiento hasta la clasificación como en [35], donde se presenta una herramienta de clasificación automática, basándose solamente en las características del núcleo como el área, brillo, diámetros, redondez, y perímetros usando la base de datos de HERLEV. Se usaron tres algoritmos para la etapa de clasificación: SVM, KNN y ANN, entre ellos, el que presentó mejor desempeño fue KNN, mediante la técnica de la distancia de Manhattan, donde logran resultados de 91 % - 97 % de precisión en la clasificación.

En general, las investigaciones realizadas respecto a los procesos de clasificación, abordan el problema de la segmentación a partir de las características del núcleo. Sin embargo, se ve una gran oportunidad de investigación a partir del análisis del citoplasma, dado que existen varios retos como lo es la correcta segmentación frente a problemas de solapamiento, agrupamiento de células, superposición y partículas que no forman parte de la célula. La extracción de características depende directamente de la segmentación, y, por ende, una buena clasificación depende de las características asociadas. Por otro lado, entre los métodos de clasificación más usados, y que han tenido gran éxito, se encuentran ANN, SVM, y KNN.

### 3. Marco teórico

El presente capítulo proporciona al lector una introducción sobre los principios teóricos que respaldan la utilización conjunta de diversas técnicas para lograr la segmentación y clasificación automatizada de las células PAP. Para ello, se explicará brevemente algunos aspectos fundamentales de la microbiología y el procesamiento digital de imágenes. De este modo, se sentará el contexto necesario para comprender la implementación del sistema.

#### 3.1 Aspectos biológicos

El término prueba de Papanicolaou hace referencia a muestras de células humanas teñidas (Figura 3.1) por el método llamado Papanicolaou. Una muestra de células se esparce en un portaobjetos de vidrio y se tiñe, lo que facilita el examen de las células bajo un microscopio en busca de anomalías que indiquen una etapa precancerosa [2].

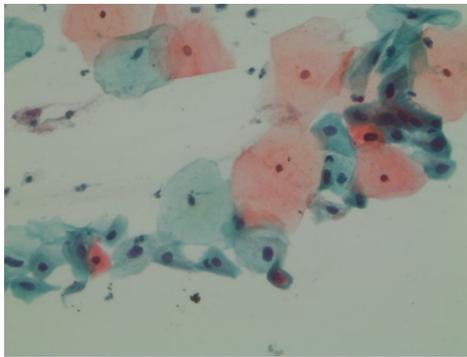


Figura 3.1: Tinción de muestras [1]

Cuando se realiza una prueba de Papanicolaou o citología cervicouterina, idealmente se toman muestras de varias zonas del cuello uterino. Estas muestras suelen contener células de dos tipos de epitelios: el epitelio columnar y el epitelio escamoso [2]. Como se observa en la Figura 3.2, el columnar se encuentra en la parte superior del cuello uterino, mientras que el escamoso reposa en la parte inferior, y en la parte central se ubica la unión escamoso-columnar.

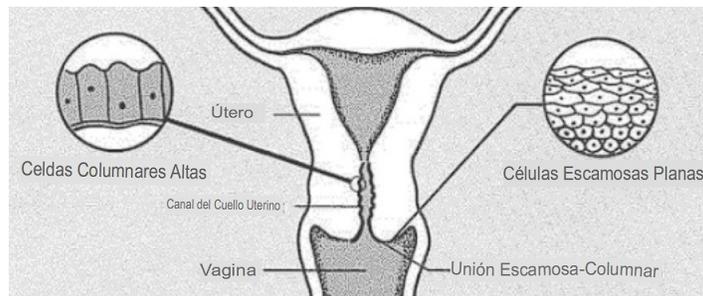


Figura 3.2: Ubicación células columnares y células escamosas [2].

### 3.1.1 Grado patológico

La patogenia del cáncer de cuello uterino, Figura 3.3, se puede evaluar de forma periódica a través del examen de Papanicolau, el cual permite determinar si son células normales, o con algún tipo de displasia leve, moderada, severa (anormales), o si ya han constituido un carcinoma in situ [3].

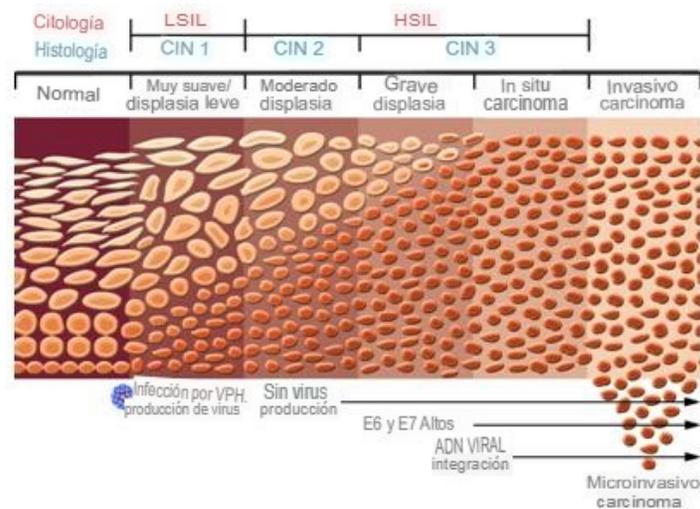


Figura 3.3: Grado patológico [3].

Los cambios morfológicos de las células se pueden caracterizar de la siguiente manera:

#### ■ Normal

- El epitelio escamoso tiene cuatro capas de células. A medida que las células se forman en la capa basal y maduran, ascienden por la capa parabasal,

intermedia y superficial. Cambian de forma, color y otras características a medida que avanzan. Cuando llegan a la capa superficial, son reemplazadas por células nuevas. La capa basal tiene células redondas con núcleos grandes y citoplasma pequeño, pero a medida que maduran, el núcleo se hace más pequeño y el citoplasma más grande. La forma de las células se vuelve menos redonda a medida que maduran [2]. En la Figura 3.4, se observan dos células de tipo normal.

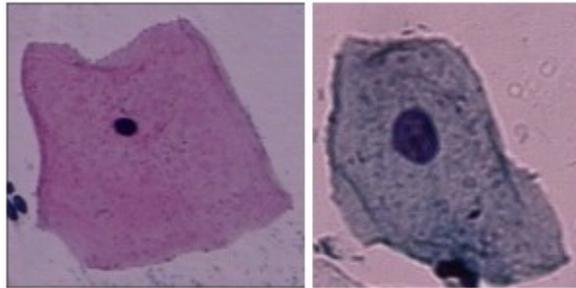


Figura 3.4: Epitelio escamoso superficial; epitelio escamoso intermedio [2].

- El epitelio columnar consta de una sola capa de células, incluyendo células columnares y células de reserva. Las células de reserva se dividen para formar nuevas células de reserva y nuevas células columnares. En las células columnares normales (Figura 3.5), el núcleo se encuentra en la parte inferior del citoplasma, lo que le da una apariencia más grande cuando se ve desde arriba. Desde un lado, el citoplasma parece más grande [2].

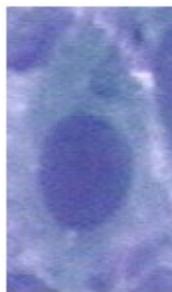


Figura 3.5: Epitelio columnar [2].

#### ■ Anormal

- El epitelio metaplástico se compone de células de reserva del epitelio columnar que, al madurar, se asemejan a las células del epitelio escamoso.

Las células displásicas son células precancerosas que presentan cambios genéticos y una división celular anormal. La displasia se clasifica en leve, moderada y grave (Figura 3.6), dependiendo de la probabilidad de convertirse en células malignas. Las células displásicas escamosas tienen núcleos más grandes y oscuros que tienden a agruparse. En la displasia grave, los núcleos son grandes, con gránulos oscuros y deformados, mientras que el citoplasma es pequeño y oscuro en comparación con los núcleos [2].

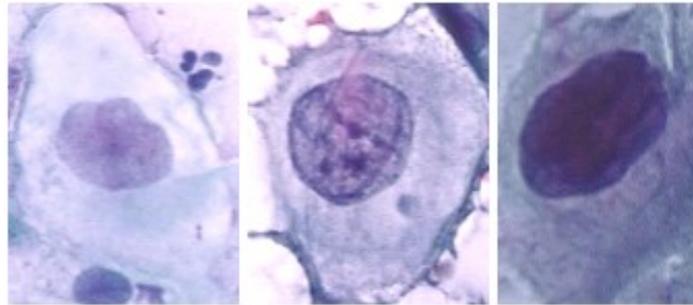


Figura 3.6: Displasia escamosa no queratinizante leve; displasia escamosa no queratinizante moderada; displasia escamosa no queratinizante severa [2].

#### ■ Carcinoma in Situ

El proceso de desarrollo del cáncer es gradual, comenzando con un epitelio normal que se vuelve displásico, luego progresa a carcinoma in situ (Figura 3.7) y finalmente a cáncer invasivo. Debido a la duración de este proceso, existe la posibilidad de un diagnóstico temprano y una cura completa [2].

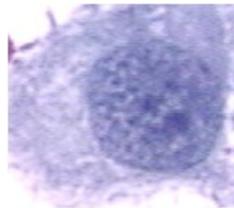


Figura 3.7: Carcinoma in situ [2].

### 3.2 Procesamiento de imágenes

El procesamiento de imágenes es una técnica de gran utilidad que permite la manipulación de imágenes digitales en diferentes etapas a través de la implementación

de algoritmos y técnicas matemáticas. Este proceso incluye la adquisición, preprocesamiento, segmentación, extracción y selección de características, y clasificación de los datos obtenidos [4].

### 3.2.1 Preprocesamiento

Existen diversas formas de ruido que pueden afectar a las imágenes durante la adquisición, como perturbaciones en la transmisión, efectos térmicos o inclusiones extrañas en la imagen. Por tanto, los algoritmos de preprocesamiento tienen la finalidad de modificar la imagen para la eliminación de ruido, transformarla geométricamente, mejorar la intensidad o el contraste [4].

La mayoría de los filtros se implementan en dos dominios: espacial y frecuencial. En el dominio espacial, se manipula directamente la luminancia de los píxeles en función de muestras o píxeles vecinos, mientras que en el dominio frecuencial implica una transformación de la imagen en términos de amplitud y fase para modificar los componentes de frecuencia y luego una transformación inversa para convertirla nuevamente al dominio espacial [4].

- Filtro de Media

Consiste en reemplazar cada píxel por el valor de la media aritmética de los píxeles circundantes, utilizando una ventana de  $N \times N$  que incluye el píxel central, como se vé en la Figura 3.1. Para cada ventana, se suma el valor de todos los píxeles y se calcula la media aritmética, que se utiliza para reemplazar el valor del píxel central en la nueva imagen.  $N$  es conveniente de que sea impar para que el píxel modificado sea el central. El cálculo del valor del nuevo píxel se expresa por la ecuación 3.1:

$$IMB(i, j) = \frac{\sum_{m=-N/2}^{N/2} \sum_{n=-N/2}^{N/2} IMA(i + m, j + n)}{N * N} \quad (3.1)$$

Su resultado es similar a una difuminación de la imagen, lo que produce un suavizado de los gradientes en toda la imagen. Este proceso es eficiente para eliminar el ruido no deseado causado por un muestreo deficiente.

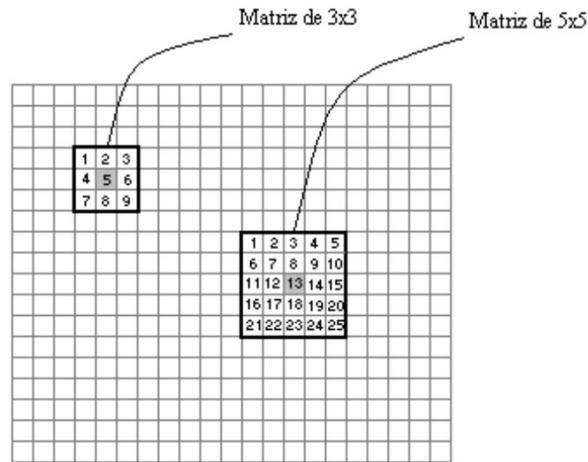


Figura 3.8: Representación ventana 3x3 y 5x5 [4].

- Filtro de Mediana

Obtiene los valores de los píxeles que caen dentro de una ventana, los cuales se organizan, y de ellos, se toma el píxel que se encuentra justo en el medio [4].

- Filtro Gaussiano

El filtro Gaussiano es similar al filtro de media, pero utiliza una ventana con distribución gaussiana. Por tanto, el filtro gaussiano para imágenes tiene la siguiente forma:

$$G(x, y) = \frac{1}{2\pi\sigma} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3.2)$$

Donde  $\sigma$  es la desviación estándar de la distribución gaussiana que determina la forma y la amplitud del filtro, y se asume una mediana igual a cero.

### 3.2.2 Segmentación

La segmentación consiste en seleccionar y separar el objeto de interés de la imagen para su posterior análisis, haciendo uso de técnicas que permitan extraer la región, dejando de lado componentes ajenos a esta. En líneas generales, el objetivo de la segmentación es dividir la imagen en las partes que la constituyen o los objetos que la forman [4].

- SLIC

Un superpíxel es una región pequeña y compacta de la imagen conformada por un conjunto de píxeles vecinos, que tienen características similares en término de color, brillo, textura. Generalmente, estas regiones mantienen la información necesaria para una mayor segmentación de la imagen y no eliminan la información de los límites de los objetos. El agrupamiento iterativo lineal simple (SLIC) realiza una búsqueda de los centros del grupo en una región definida  $2Sx2S$ , donde  $S \approx \sqrt{N/k}$  es el tamaño de la región de búsqueda del superpíxel. Este algoritmo toma dos parámetros [36] que son el número de superpíxeles  $k$  y el valor de compacidad  $m$ . La distancia  $D'$  entre los píxeles en el espacio  $Cielab$  y las coordenadas  $xy$  se calcula como:

$$d_c = \sqrt{(l_i - l_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2} \quad (3.3)$$

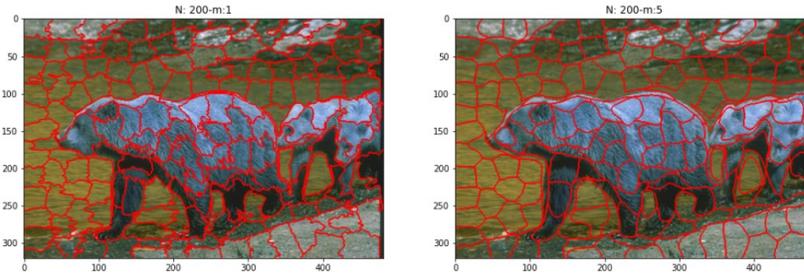
$$d_s = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \quad (3.4)$$

$$D' = \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2} \quad (3.5)$$

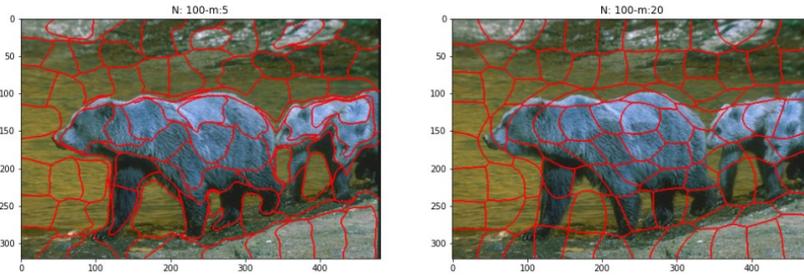
Donde  $N_c$  y  $N_s$  son la distancia máxima de color y distancia espacial, respectivamente.  $N_c$  es difícil de estimar, por esta razón se fija la constante  $m$ .  $D'$  se escala en la práctica como se ve en la expresión 3.6.

$$D = \sqrt{d_c^2 + \left(\frac{d_s}{S}\right)^2 m^2} \quad (3.6)$$

Los valores pequeños de  $m$  generan superpíxeles con mejor adaptación a los límites de color, pero menor compacidad espacial. La compacidad se entiende como la capacidad que tiene el algoritmo de generar píxeles más compactos y regulares que sigan la forma del objeto y su distribución en la imagen. Un ejemplo de esto se puede ver en la figura 3.9, donde se define el número de píxeles y compacidad, mostrando, así, los contornos de los superpíxeles generados.



(a) Número de superpíxeles 200; valores de compacidad 1 y 5 respectivamente.



(b) Número de superpíxeles 100; valores de compacidad 5 y 20 respectivamente.

Figura 3.9: Contornos de superpíxeles generados (fuente: autores).

- Watershed

El método *Watershed* o de cuenca es un algoritmo que emplea la técnica de ascenso de gradiente para producir segmentos delimitados por líneas de *cuenca*, a partir de mínimos locales, mediante la subdivisión de regiones. Sin embargo, es importante tener en cuenta que este algoritmo puede generar superpíxeles irregulares en forma y tamaño [36]. Esta técnica clasifica los píxeles según su proximidad espacial, el gradiente de sus niveles de gris y la homogeneidad de sus texturas. En la figura 3.10, se presentan los resultados del algoritmo *Watershed*, donde se pueden apreciar los segmentos generados. En cada una de las imágenes a, b y c, se ha asignado un valor específico a la variable *markers*: 250, 150 y 400 respectivamente. La variable *markers* determina la cantidad de marcadores utilizados y, por consiguiente, el número de segmentos generados en cada imagen.

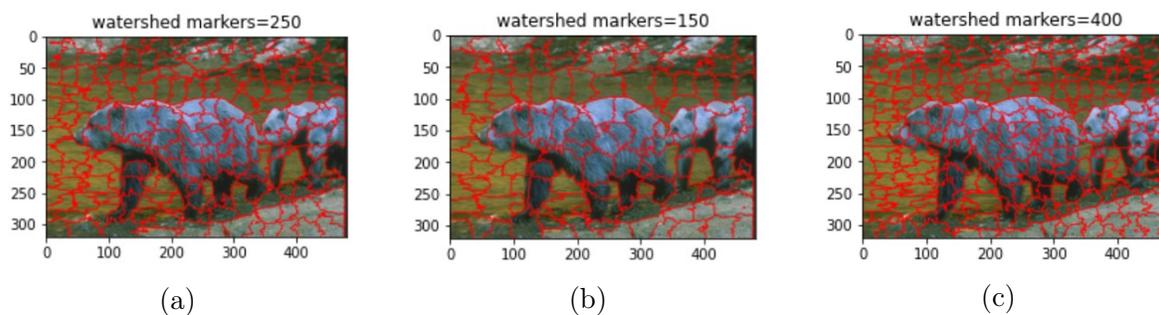


Figura 3.10: Contornos del agrupamiento *Watershed* (fuente: autores).

### ■ KMEANS

Es un algoritmo iterativo utilizado en el aprendizaje no supervisado de máquinas que permite agrupar los elementos de un conjunto de datos no etiquetados [37]. Su objetivo es encontrar grupos o clústeres en los datos los cuales están de forma natural.

En la Figura 3.11, se puede ver el paso a paso del agrupamiento. Es importante establecer el número de grupos que se desean formar, donde la estimación inicial suele basarse en el conocimiento previo que se tenga del conjunto de datos. Se debe elegir adecuadamente el número de grupos, ya que un número muy bajo puede no representar la complejidad del conjunto de datos, mientras que un número muy alto puede conducir a una mala generalización.

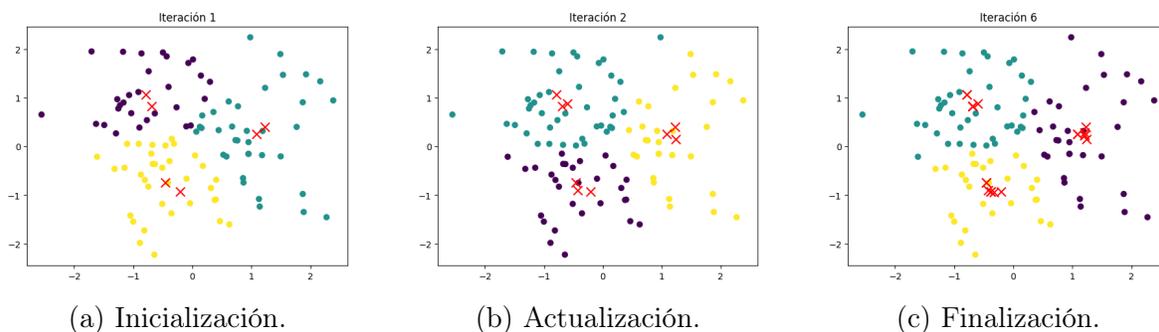


Figura 3.11: Ajuste de centroides KMEANS (fuente: autores).

Como primer paso (Figura 3.11a), se inicializan los centroides en el espacio de datos, donde cada observación es asignada con la etiqueta del centroe más cercano. Matemáticamente se puede expresar de la siguiente manera [38] (ecuación 3.7):

$$\min_s E(u_i) = \min_s \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - u_i\|^2 \quad (3.7)$$

Donde  $S$  es el conjunto de datos que se quiere agrupar, y  $x_j$  son sus correspondientes observaciones.  $u_i$  representa el centroide con índice  $i$  del grupo  $S_i$ . El número de grupos es representado por  $k$  con su centroide  $u_i$ .

La expresión 3.7, indica que se trata de un problema de optimización, donde se buscan los centroides  $u_i$  que minimicen la suma de las distancias euclidianas  $\|x_j - u_i\|^2$  de cada grupo  $S_k$ , de tal forma que minicen la función de costo  $E(u_i)$ .

En el segundo paso (Figura 3.11b), se recalcula la posición de los centroides de acuerdo al promedio de las observaciones asignadas a cada grupo, esto se hace de forma iterativa. La expresión 3.8 indica la actualización de los centroides en cada iteración del algoritmo KMEANS.

$$u_i^{t+1} = \frac{1}{|S_i|} \sum_{x_j \in S_i} x_j \quad (3.8)$$

Es decir que en cada iteración, la nueva posición del centroide  $u_i$  del grupo  $S_i$  se calcula como el promedio de las observaciones  $x_j$  asignadas a dicho grupo.

Finalmente (Figura 3.11c), el criterio de parada ocurre cuando los centroides ya no se mueven más o si se alcanza el número máximo de iteraciones.

#### ■ MEANSHIFT

El algoritmo de desplazamiento medio o ventanas deslizantes [39] consiste en un método de agrupamiento no paramétrico que busca modos o grupos locales de una función de densidad estimada a partir de los datos de entrada [40]. Es decir, que busca los puntos de mayor densidad del conjunto de datos.

Se trata de un método no supervisado basado en centroides, que lo convierte en un algoritmo similar a KMEANS. Ambos son de tipo iterativos y actualizan los centroides en cada ciclo. Sin embargo, MEANSHIFT se diferencia en el enfoque de agrupamiento, ya que se basa en el desplazamiento iterativo de cada centroide hacia el punto de máxima densidad de los datos, y no requiere conocer de antemano el número de grupos presentes en ellos. Matemáticamente se puede expresar así [41]:

Dado un conjunto de datos  $n$  con  $i = 1, 2, 3, \dots, n$ , en un espacio  $d$ -dimensional  $R^d$ , la forma básica del vector MEANSHIFT se define como en la ecuación 3.9:

$$M_h = \frac{1}{k} \sum_{x_i \in S_k} (x_i - x) \quad (3.9)$$

Donde  $S_k$  es una esfera de alta dimensión con radio  $h$  y  $k$  indica que existen  $k$  puntos cayendo dentro de la región  $S_k$  con  $n$  puntos de muestra  $x_i$ . Agregando una función de kernel al vector básico, el algoritmo se transforma en:

$$\hat{f}_{h,k}(x) = \frac{c_{k,d}}{nh^d} \sum_{i=1}^n k\left(\left\|\frac{x-x_i}{h}\right\|^2\right) \quad (3.10)$$

donde  $h$  es el radio, y  $\frac{c_{k,d}}{nh^d}$  es la densidad unitaria.

$$x = \frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} \quad (3.11)$$

La ecuación 3.11, muestra las coordenadas del nuevo centroide, lo que quiere decir el vector MEANSHIFT siempre apunta hacia la dirección donde la densidad aumenta.

En la Figura 3.12, se puede ver la convergencia del centroide del grupo verde. En este caso se tiene un punto en la parte externa del grupo, el cual está encerrado por una circunferencia (ventana) de radio  $r$  cuya área local se encuentra definida por el ancho de banda.

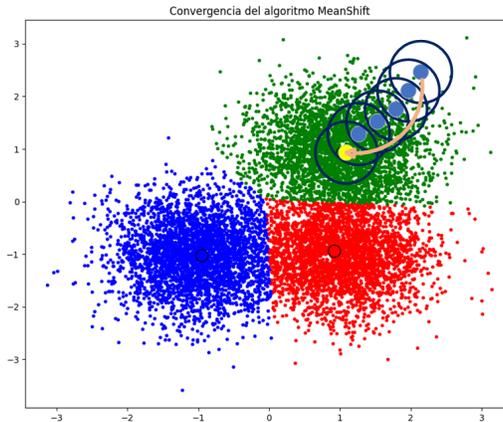


Figura 3.12: Convergencia de centroides (fuente: autores).

El paso siguiente es obtener la posición del nuevo centroide a partir del cálculo de la media de los valores que se encuentran contenidos por la esfera. Seguidamente, la ventana se desplaza hacia el nuevo centroide obtenido, y este proceso se repite varias veces hasta converger, es decir, que los centroides ya no se mueven más puesto que llegaron al punto de densidad máxima. Aquí, las observaciones son etiquetadas o asignadas al centroide más cercano en el espacio de características, formándose así los grupos.

■ **SEGMENTACIÓN SEMÁNTICA UNET**

La segmentación semántica se puede considerar como una técnica de procesamiento de imágenes que tiene como objetivo la clasificación píxel a píxel de una imagen, asignándolos a una categoría predefinida [42]. Por su parte, UNET es una red neuronal convolucional profunda utilizada para segmentación la cual consta de dos partes (Figura 3.13): una de codificación que captura el contexto de la imagen, y otra de decodificación que genera la máscara de segmentación [5].

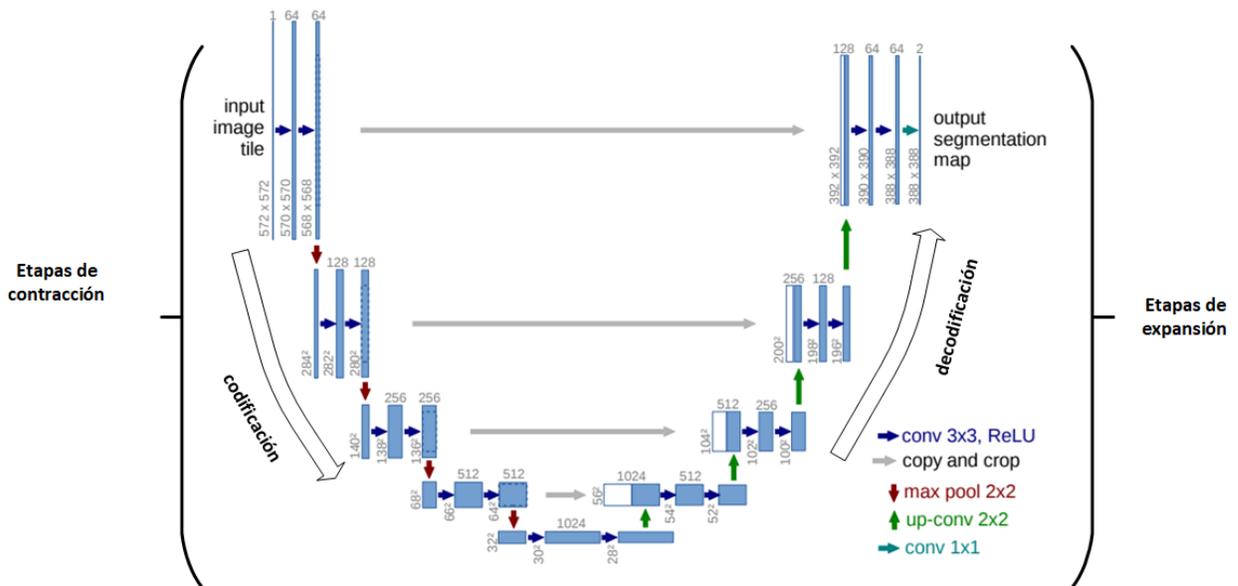


Figura 3.13: Estructura UNET [5].

Como se observa en la Figura 3.13, la ruta de codificación de la parte izquierda sigue la forma típica de una red neuronal convolucional, con cuatro etapas de contracción. Cada etapa tiene dos convoluciones de 3x3 seguidas por una función de activación ReLU, y una operación de *maxpooling* de 2x2 para submuestreo.

ReLU, es una función de activación no lineal, usada en redes neuronales para mejorar la capacidad de aprendizaje, especialmente para tareas de clasificación no lineales. La operación *maxpooling* se utiliza en la etapa de codificación para reducir progresivamente la resolución espacial de una representación de características. En cada etapa de contracción, se duplica la profundidad de los canales, mientras que se reduce la dimensionalidad (ancho y alto) de la imagen.

En la parte derecha, se puede observar la decodificación, que se compone de cuatro etapas de expansión. Cada etapa está acompañada por una operación de convolución hacia arriba de  $2 \times 2$ , que duplica la dimensionalidad de la imagen, una concatenación con el mapa de características recortado de la etapa de compresión, y dos convoluciones de  $3 \times 3$  seguida por una función ReLU. En cada etapa de expansión, se aumenta la resolución de la imagen, mientras que se reduce la profundidad en canales. La concatenación de la capa de contracción con la de expansión permite que la información de alta resolución de la imagen se fusione con la información de baja resolución, lo que permite una segmentación precisa de los objetos de interés. Al final, se obtiene una máscara que corresponde con la imagen segmentada.

La fase de contracción reduce gradualmente la resolución espacial de la imagen mientras aumenta el número de características extraídas, lo que permite obtener una representación semántica de alto nivel sobre los objetos de interés. En la fase de expansión, se aumenta gradualmente la resolución espacial de la imagen mientras se fusionan las características extraídas en la fase de contracción con la imagen de alta resolución generando una máscara de segmentación.

### 3.3 Descriptores de forma y de textura

#### 3.3.1 Características de forma

Las características de forma están relacionadas a medidas numéricas que describen la forma de los objetos presentes en la imagen. Estas características pueden incluir medidas como el área, la longitud entre ejes, la circunferencia, el perímetro, la compacidad, entre otros. A continuación, se presentan algunas:

- Área

Se define como la región encerrada por la línea del contorno, que para una imagen bidimensional está representado por el número de píxeles que son diferentes de cero [43].

Así, el área de una región  $D$  para una imagen binarizada el área se define por la ecuación 3.12:

$$Area(D) = \sum_i \sum_j I(i, j) \quad (3.12)$$

Siendo  $I(i, j)$  el valor del pixel en la  $i$ -ésima fila y  $j$ -ésima columna de la imagen.

- Perímetro

El perímetro es el número total de píxeles en el borde de una región segmentada.

Teniendo en consideración que en plano discreto 2D, donde se conocen las coordenadas  $(x, y)$  de una serie de puntos que forman una curva, el perímetro queda determinado por la ecuación 3.13:

$$L(C) = \sum_{i=1}^N \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (3.13)$$

Donde  $N$  es el número total de puntos que conforman la curva,  $(x_i, y_i)$  son las coordenadas del punto  $i$  de la curva,  $(x_{i-1}, y_{i-1})$  son las coordenadas del punto anterior a  $i$ . Por tanto, el perímetro  $L(C)$  se define como la suma de las distancias euclidianas entre los puntos consecutivos  $(x_{i-1}, y_{i-1})$  y  $(x_i, y_i)$ .

- Centroide

Esta característica representa la posición media del contorno de una figura, la cual se puede calcular de la siguiente manera:

$$C_x = \frac{\sum_x \sum_y xI(x, y)}{\sum_x \sum_y I(x, y)} = \frac{m_{10}}{m_{00}} \quad C_y = \frac{\sum_x \sum_y yI(x, y)}{\sum_x \sum_y I(x, y)} = \frac{m_{01}}{m_{00}} \quad (3.14)$$

Donde  $(C_x, C_y)$  son las coordenadas del centroide de una región, obtenidas mediante el cálculo de promedios ponderados de las coordenadas  $(x, y)$  utilizando sus intensidades como pesos.

- Convexidad

La convexidad es una propiedad geométrica que describe la forma de un objeto o una región. Un conjunto o una figura se considera convexo si, dado cualquier par de puntos dentro de él, la línea que los conecta también está completamente contenida dentro del conjunto.

- Longitud de eje mayor y menor

La longitud del eje mayor y menor se refiere a las dimensiones principales de una elipse.

- Excentricidad

Esta medida establece una relación entre el tamaño de los focos de una elipse y la longitud del eje principal, expresada mediante la ecuación 3.15:

$$Excentricidad = \sqrt{1 + \frac{Longitud\ de\ eje\ menor^2}{Longitud\ de\ eje\ mayor^2}} \quad (3.15)$$

Cuanto menor sea la excentricidad, mayor será la similitud a una forma circular, en consecuencia, se obtiene una elipse menos alargada. Por tanto, indica el grado de desviación respecto a una forma perfectamente circular.

- Solidez

Indica el grado de semejanza de un contorno a una forma convexa. Esto se hace comparando el área del contorno con el área del conjunto convexo más pequeño que lo contiene. Si se toma la colección de puntos  $A$  sobre el contorno, se puede generar el conjunto convexo más pequeño que contiene a  $A$ . Expresado matemáticamente mediante la expresión 3.16, donde  $\alpha_i$  son los pesos asignados a los puntos  $x_i$  para construir el conjunto convexo:

$$Conv(A) = \sum_i \alpha_i x_i \quad \alpha_i \geq 0, \sum_i \alpha_i = 1 \quad (3.16)$$

Por lo tanto, la solidez (ecuación 3.17) se define como la relación que existe entre el área del contorno y el área convexa del casco:

$$Solidez = \frac{Área\ del\ contorno}{Área\ convexa\ del\ casco} \quad (3.17)$$

- Extensión

Esta medida se define como la proporción que existe entre el área del contorno y el área del rectángulo delimitador a este contorno.

$$Extensión = \frac{Área\ del\ Contorno}{Área\ del\ rectángulo\ delimitador} \quad (3.18)$$

- Relación de aspecto

La relación de aspecto es la razón que existe entre el ancho y la altura del contorno del objeto de acuerdo al rectángulo delimitador del contorno.

$$Relación\ de\ Aspecto = \frac{Ancho}{Altura} \quad (3.19)$$

- Diámetro equivalente

Es el diámetro del círculo donde el área de este es equivalente al área del contorno del objeto [43], expresado mediante la siguiente ecuación:

$$Diámetro\ equivalente = \sqrt{\frac{4(Área\ del\ contorno)}{\pi}} \quad (3.20)$$

- Diámetros de Feret

Los diámetros de Feret son una medida comúnmente utilizada para evaluar la geometría de objetos. Esta medida se obtiene midiendo la distancia entre los puntos más alejados en dos direcciones perpendiculares en la imagen del objeto. Hay varios tipos de diámetros de Feret, pero en general, consiste en calcular la máxima longitud para un ángulo en sentido anti horario con el eje  $x$ , y que se define entre  $0$  y  $180^\circ$  [4]. Por lo tanto es posible calcular las dimensiones del eje más largo y más corto del objeto (Figura 3.14).

### 3.3.2 Características de textura

- Matriz de coocurrencia de grises (GLCM)

Esta técnica permite analizar las características de textura de una imagen o región de interés, teniendo en cuenta la frecuencia con la que aparece un nivel de gris con referencia a otro valor de gris dentro de una ventana determinada [44].

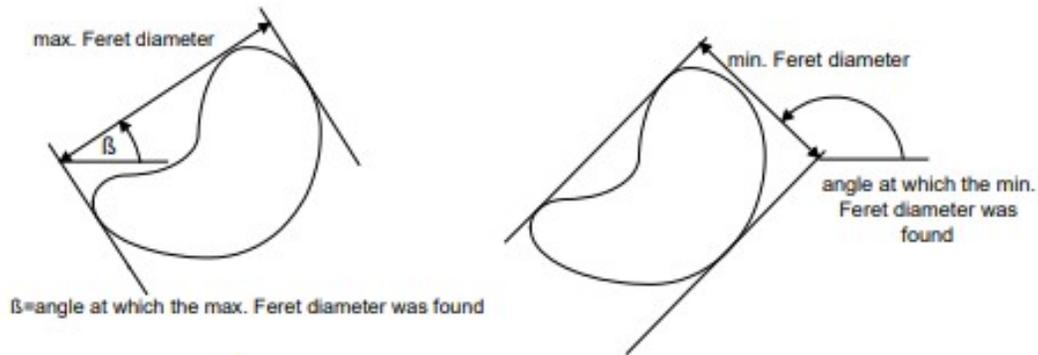


Figura 3.14: Ángulos cuyo diámetro de Feret es máximo y mínimo [4].

De GLCM se pueden derivar varias medidas como la homogeneidad, el contraste, la disimilaridad, correlación y otras más.

- Homogeneidad

Evalúa la uniformidad de los niveles de gris en una imagen. Esta medida puede variar entre 0 y 1, donde un valor más cercano a 1 indica una distribución uniforme de los niveles de gris de la imagen, mientras que un valor cercano a 0 indica una distribución más heterogénea [44].

$$\sum_{i,j=0}^{N-1} P_{i,j}/1 + (i - j)^2 \quad (3.21)$$

Siendo  $P_{i,j}$  la probabilidad de coocurrencia de los valores de gris  $i$  y  $j$ , para una distancia determinada.

- Contraste

El contraste es una medida de la variación local en la imagen, cuyo valor es alto en la escala seleccionada de la ventana cuando el contraste es alto.

$$\sum_{i,j=0}^{N-1} P_{i,j}(i - j)^2 \quad (3.22)$$

- Disimilaridad

Hace referencia a la diferencia entre los valores de gris entre píxeles adyacentes. Al igual que el contraste, la disimilaridad es alta cuando la

región tiene un contraste alto.

$$\sum_{i,j=0}^{N-1} P_{i,j} |i - j| \quad (3.23)$$

- Entropía

Indica el grado de aleatoriedad en la distribución en los niveles de gris. La entropía es alta cuando los elementos de la matriz de coocurrencia tienen valores aproximadamente iguales, y baja cuando los elementos son cercanos a 0 ó 1.

$$\sum_{i,j}^{N-1} -P_{i,j} \ln(P_{i,j}) \quad (3.24)$$

Se asume que  $0 \cdot \ln(0) = 0$ .

- Correlación

Indica la relación lineal de los valores de gris de píxeles adyacentes.

$$\sum_{i,j=0}^{N-1} P_{i,j} \left[ \frac{(i - \mu_i)(j - \mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right] \quad (3.25)$$

Un valor cercano a 1 o -1 indica que existe una relación lineal fuerte entre píxeles adyacentes y cuyos valores de gris tienden a variar juntos en la misma dirección. Un valor cercano a 0 indica ausencia de relación lineal, lo que dificulta la predicción de los valores de gris entre píxeles adyacentes.

- Segundo momento angular (ASM)

El ASM es una medida que ayuda a calcular la homogeneidad local, cuyos valores son altos cuando en la matriz de coocurrencia hay pocas entradas de gran magnitud, y baja cuando hay entradas similares.

$$ASM = \sum_{i,j=0}^{N-1} P_{i,j}^2 \quad (3.26)$$

La raíz cuadrada de esta medida se denomina energía o uniformidad, también usada como medida de textura [44].

### 3.3.3 Características de color

El color se forma mediante la combinación de tres colores básicos, que son el rojo, verde, azul (RGB). Diferentes combinaciones de estos colores dan como resultado la amplia gama de colores. Una imagen se caracteriza por ser un arreglo bidimensional de píxeles con diferentes intensidades luminosas (escala de gris) [45].

- Tono  
Indica si un área se asemeja al rojo, amarillo, verde, azul o una aproximación de estos.
- Luminosidad  
Es el brillo de una zona en relación con otra zona blanca en la imagen.
- Croma  
Indica la coloración del área en relación con el brillo de un blanco de referencia.
- Brillo  
El brillo indica qué tan iluminada está un área de la imagen. El cálculo del brillo  $Y$  de la imagen se realiza promediando la intensidad en cada uno de los espacios de color RGB, multiplicando el promedio de intensidad de cada espacio por los siguientes coeficientes: [0.299, 0.587, 0.114] respectivamente, y luego sumando estos resultados. Este cálculo proporciona una medida objetiva del brillo de la imagen, donde los coeficientes dan una aproximación de la percepción humana del brillo.

$$Y = 0,299 * Ru + 0,587 * Gu + 0,114 * Bu \quad (3.27)$$

Donde  $R_u$ ,  $G_u$  y  $B_u$  son los promedios de intensidad de cada espacio de color.

### 3.4 Selección y extracción de características

La selección de características consiste en evaluar y optimizar el espacio de características que se utilizarán para la clasificación, es decir, aquellas que contengan la mayor información y que permitan la máxima separabilidad entre las clases [46], que sean altamente predictivas y menos correlacionadas entre sí. Esto trae consigo

muchas ventajas, como la reducción de ruido, pues se eliminan características redundantes que no aportan información útil, reducción de la complejidad del modelo en términos de tiempo y espacio, y una interpretación más clara de resultados [47].

A continuación, se presentan algunas técnicas comunes para selección de características:

### 3.4.1 Selección basada en correlación

La correlación de Pearson indica una relación lineal entre dos variables [48]. En el campo del aprendizaje automático, se utiliza para medir la relación entre cada característica y la variable objetivo. Si la correlación es alta, entonces indica que la característica puede ser útil para clasificación. Si bien es cierto, en algunos casos, eliminar características que tienen una correlación similar puede mejorar la clasificación, muchas veces algunas de ellas proporcionan información complementaria que puede ser importante para el modelo de clasificación. Además, cuando el espacio de características es grande, este análisis no es tan sencillo, y puede ser necesario realizar un análisis más complejo y sofisticado. La selección de características para la clasificación es un proceso complejo que requiere considerar múltiples factores, y no se puede reducir únicamente a la evaluación de la correlación entre características y la variable objetivo.

### 3.4.2 FEATUREWIZ

El algoritmo *FEATUREWIZ* [49], permite la selección de características óptimas de un conjunto de datos basándose en dos técnicas: búsqueda de lista de variables no correlacionadas (SULOV) y XgBoost.

El método SULOV se basa en el algoritmo Mínima-Redundancia-Máxima-Relevancia (MRMR) [50] como uno de los mejores métodos de selección de características. SULOV funciona de la siguiente manera:

Paso.1 Búsqueda de pares de características que superan un umbral de correlación lineal de 0,7.

Paso.2 Calcular el puntaje de información mutua basada en la entropía, de cada característica con respecto al valor asignado a la clase (variable objetivo).

Paso.3 Tomar cada par de características correlacionadas y eliminar la que tenga menor puntaje de información mutua. Un puntaje de información mutua alto indica que una característica es altamente relevante para predecir la variable objetivo, mientras que un puntaje bajo indica que la característica no es muy útil para predecir la variable objetivo.

Paso.4 Tomar las características resultantes. Estas características presentan el mayor puntaje de información mutua y la menor correlación entre sí.

Después de obtener las características menos correlacionadas entre sí, se utiliza el algoritmo XgBoost [51] para una búsqueda exhaustiva de las características más relevantes. Este algoritmo utiliza técnicas de inteligencia artificial, generalmente aprendizaje por árboles que permite generar una lista de características. Inicialmente el método XgBoost, toma los datos entregados por SULO V y los divide en varios conjuntos de entrenamiento y prueba. Posteriormente, encuentra las características más representativas y repite este proceso en el siguiente conjunto. Luego de 5 repeticiones, se obtienen 5 conjuntos con las mejores variables, por lo que se combinan y se eliminan las características duplicadas.

### 3.4.3 Análisis de componentes principales PCA

PCA también llamado transformada de Karhunen-Loeve, es un método algebraico que permite sintetizar la información de un conjunto de datos reduciendo su dimensionalidad [52]. Consiste en combinaciones lineales de las variables originales cuyo objetivo es encontrar un conjunto de ejes ortogonales en los cuales la varianza de los datos sea máxima. El primer componente principal registra la mayor cantidad de variabilidad en los datos, seguido por el segundo componente PCA y así sucesivamente. Matemáticamente se puede expresar de la siguiente forma:

$$S_i = \sum_{k=1}^{\bar{d}} a_{k,i} e_{k,i} \bar{d} \leq d \quad (3.28)$$

El método PCA transforma  $n$  número de vectores  $S_1, S_2, S_3, \dots, S_n$  pertenecientes a un espacio dimensional  $d$ , a  $n$  vectores  $S'_1, S'_2, \dots, S'_n$  de un nuevo espacio dimensional  $\bar{d}$ , donde  $e_k$  son los vectores propios y  $\bar{d}$  es el máximo autovalor de la matriz de dispersión [11]. La ecuación 3.29, indica la varianza de los datos, la cual será usada para identificar la dispersión de los datos.

$$Var(x) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{X})^2 \quad (3.29)$$

$$Cov(x, y) = \frac{1}{n-1} \sum_{i=1}^n (x_{i,j} - \mu_{x,j})(y_{i,j} - \mu_{y,j}) \quad (3.30)$$

En el mismo sentido, la covarianza (ecuación 3.30) muestra qué tan relacionadas están dos variables frente a sus cambios individuales [53]. Por tanto, de la matriz de covarianza resultan los vectores propios que son los componentes principales, y de sus autovalores se obtiene la proporción de la varianza explicada por dicho componente. En este punto, se seleccionan los componentes principales que explican la mayor cantidad de varianza en los datos y con ellos se forma un nuevo espacio de datos con los  $k$  componentes principales.

En la Figura 3.15, se muestra un ejemplo de transformación de un espacio con 4 características, al espacio PCA usando el conjunto de datos flor Iris [54].

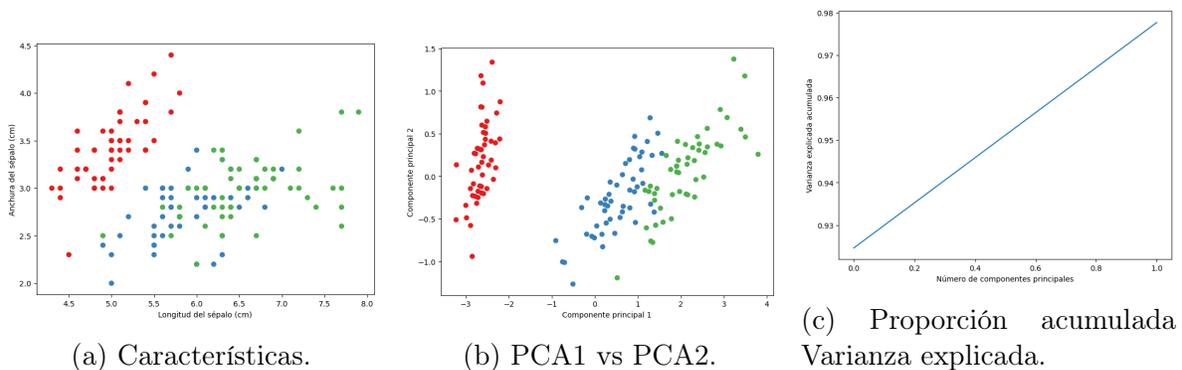


Figura 3.15: Ejemplo de características PCA (fuente: autores).

El conjunto de datos Iris contiene información de tres especies de flor Iris, con cuatro características específicas: longitud del sépalo, anchura del sépalo, longitud del pétalo y anchura del pétalo. La Figura 3.15a, corresponde a un gráfico de dispersión tomando

solo dos características, en las cuales no se logran distinguir las tres clases de flores. Sin embargo, al aplicar componentes principales, como se ve en la Figura 3.15b, con solo dos componentes, se logra una mejor separabilidad entre las clases, con una varianza explicada acumulada superior al 97 % (Figura 3.15c). Por tanto, se puede observar que PCA redujo la dimensionalidad de cuatro a dos dimensiones, los cuales representan mucho mejor los datos en términos de separabilidad.

## 3.5 Aprendizaje automático supervisado

El aprendizaje automático supervisado es aquel proceso en el que un modelo aprende a predecir una salida no etiquetada a partir de datos etiquetados que sirven como referencia. Con dichos datos de referencia, el modelo ajusta sus parámetros internos de tal forma que correspondan con la salida deseada, a este proceso se le conoce como entrenamiento. El objetivo de esto es que el modelo aprenda a mapear una entrada a una salida esperada, de manera que pueda hacer predicciones precisas en nuevos datos no etiquetados.

### 3.5.1 KNN

El algoritmo K-Nearest Neighbors (KNN) es una técnica de aprendizaje automático supervisado que se utiliza para problemas de regresión y clasificación. En este algoritmo, el valor de  $k$  indica el número de observaciones vecinas o próximas que se tienen en cuenta para hacer predicciones [55]. Por lo tanto, si se seleccionan  $k$  valores más cercanos a una instancia de predicción, la salida se determinará por la clase que tenga el mayor número de observaciones entre los  $k$  vecinos seleccionados.

Una de las formas para saber la cercanía de las observaciones es a partir de la distancia euclidiana del elemento a evaluar y las demás observaciones, descrita por la siguiente expresión:

$$Distancia = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (3.31)$$

Donde  $i$  es la  $i$ -ésima característica del total  $n$  de características. El vector  $x$  es una característica del conjunto de datos de aprendizaje, mientras que  $y$  es el nuevo vector de características que se desea clasificar.

En la Figura 3.16, se muestra la idea básica que sigue el algoritmo KNN para clasificación.

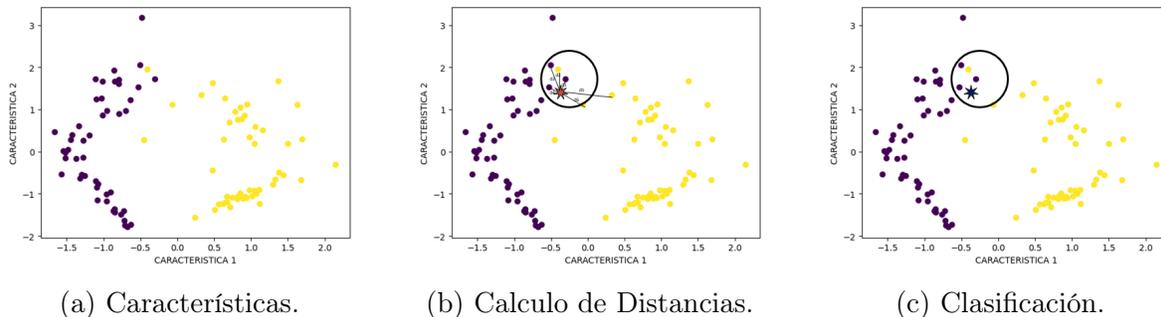


Figura 3.16: Algoritmo de clasificación KNN (fuente: autores).

En la primera Figura 3.16a, se muestra un conjunto de datos ya etiquetados que está dividido en dos clases (amarillo y azul), teniendo en cuenta dos características. En la Figura 3.16b, se quiere clasificar un nuevo dato u observación, por tanto, el primer paso es calcular la distancia del dato con todas las demás observaciones, y de ello seleccionar los  $k$  vecinos o distancias más cercanas a dicho dato. El paso a seguir es realizar la votación de los vecinos más cercanos (Figura 3.16c), es decir, que el nuevo dato se va a etiquetar con la clase que mayores observaciones presente, en este caso, para la clase azul.

### 3.5.2 SVM

Las máquinas de soporte vectorial (SVM), son herramientas aplicadas en problemas tanto para clasificación como para regresión. Las SVM están categorizadas como clasificadores lineales, debido a que estos se distinguen por hacer uso de separadores lineales definidos como hiperplanos (una línea en un espacio 2-D, un plano en un espacio 3-D, etc.). Originalmente las SVM fueron diseñadas para problemas de clasificación binaria, posteriormente se implementan para problemas de regresión, agrupamiento y multclasificación [6]. Al definir los hiperplanos (Figura 3.17), se consideran únicamente los ejemplos de entrenamiento de cada clase que caen en las

fronteras del margen, a estos ejemplos situados justo en las fronteras que delimitan el margen de cada lado del hiperplano de separación, se les llama *vectores soporte*.

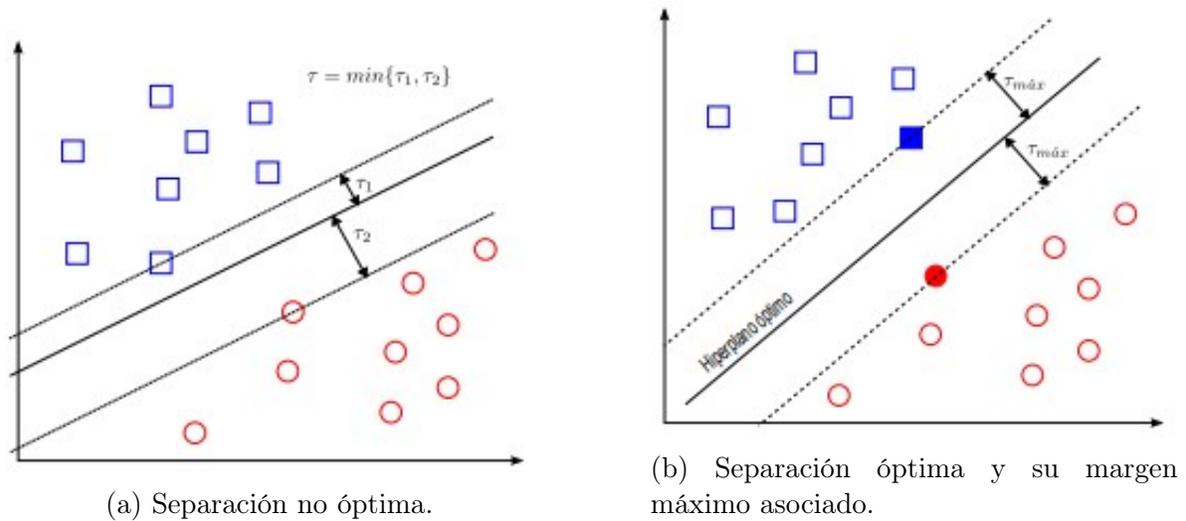


Figura 3.17: Hiperplano de separación [6].

Este algoritmo primero mapea los puntos de entrada a un espacio de características y luego encuentra un hiperplano que los separe y maximice el margen  $m$  entre las clases en este espacio. La SVM encuentra el hiperplano óptimo utilizando el producto punto con funciones en el espacio de características llamadas *kernels*.

Se puede determinar un hiperplano de decisión representado por  $w \cdot x + b = 0$ , donde  $(w \cdot x)$  es el producto punto entre  $w$  y  $x$ .

$$w_1x_1 + w_2x_2 + b = 0 \quad (3.32)$$

Donde  $x \in R^n$  es el espacio de características y  $w \in R^n$  son los parámetros del sistema, siendo las fronteras de decisión:

$$w \cdot x + b > 0 \quad w \cdot x + b < 0 \quad (3.33)$$

SVM propone establecer un margen equidistante al hiperplano quedando la siguiente

frontera de decisión con margen.

$$w \cdot x_+ + b \geq 1 \quad \text{Clase 1} \quad (3.34)$$

$$w \cdot x_- + b \leq -1 \quad \text{Clase -1} \quad (3.35)$$

Así se asegura que ningún elemento esté dentro del margen. A continuación, una expresión equivalente:

$$y_i(w \cdot x + b) \geq 1 \quad \begin{cases} y_i = 1 & x_+ \\ y_i = -1 & x_- \end{cases} \quad (3.36)$$

Para los vectores de soporte se cumple la siguiente igualdad:

$$y_i(w_i x + b) - 1 = 0 \quad (3.37)$$

Ahora lo que plantea esta técnica es que se pueda maximizar el margen teniendo en consideración los vectores de soporte, por lo tanto:

- Longitud del margen

$$d_- = \frac{x_- \cdot w}{\|w\|} \quad d_+ = \frac{x_+ \cdot w}{\|w\|} \quad (3.38)$$

$$d = d_+ - d_- = \frac{1}{\|w\|} (x_+ \cdot w - x_- \cdot w) \quad (3.39)$$

Siendo  $\|w\|$  la norma de  $w$ .

Debido que para los vectores de soporte se cumple la ecuación 3.37, despejando de ella el producto punto  $w \cdot x$ , y reemplazando en la expresión 3.39 se tiene que:

$$d = \frac{1}{\|w\|} (1 - b + 1 + b) \quad d = \frac{2}{\|w\|} \quad (3.40)$$

- Optimización

El problema de maximización y minimización

$$\operatorname{argmax} w = \frac{2}{w^T w} \quad (3.41)$$

$$\operatorname{argmin} w = \frac{1}{2} w^T w \quad (3.42)$$

Teniendo como restricción que los vectores de soporte deben estar sobre los planos de la ecuación 3.37. Resolviéndose con la técnica de los *multiplicadores de Lagrange*, que mediante una función objetivo y unas restricciones de igualdad permite establecer un nuevo problema de optimización donde no se tengan dichas restricciones.

$$L(w, b, \alpha_i) = \frac{1}{2} w^T w - \sum_{i=1}^l \alpha_i [y_i (w \cdot x_i + b) - 1] \quad (3.43)$$

Donde  $\alpha_i$  es el coeficiente de Lagrange y  $l$  el número de vectores soporte. Derivando e igualando a cero se tiene:

$$L(\alpha_i) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \quad (3.44)$$

La expresión 3.44 corresponde a la función de pérdida que optimiza el entrenamiento de una SVM, la cual busca los coeficientes  $\alpha_i, \alpha_j$  óptimos que maximizan el margen de separación entre las clases. Es una forma más compacta de la ecuación 3.43, que se usa solamente para clases linealmente separables.

- Kernel

Para actuar sobre este problema (separabilidad lineal) se hace uso de un *kernel*, permitirá aumentar la dimensionalidad del espacio, y un problema linealmente no separable se convierte en linealmente separable (ecuación 3.45).

$$L(\alpha_i) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad (3.45)$$

En la Figura 3.18 se pueden ver ejemplos de casos en los que los datos son linealmente separables (Figura 3.18a), y casos en los que no es posible la división con un separador lineal (Figura 3.18b), siendo necesario hacer uso del *kernel*.

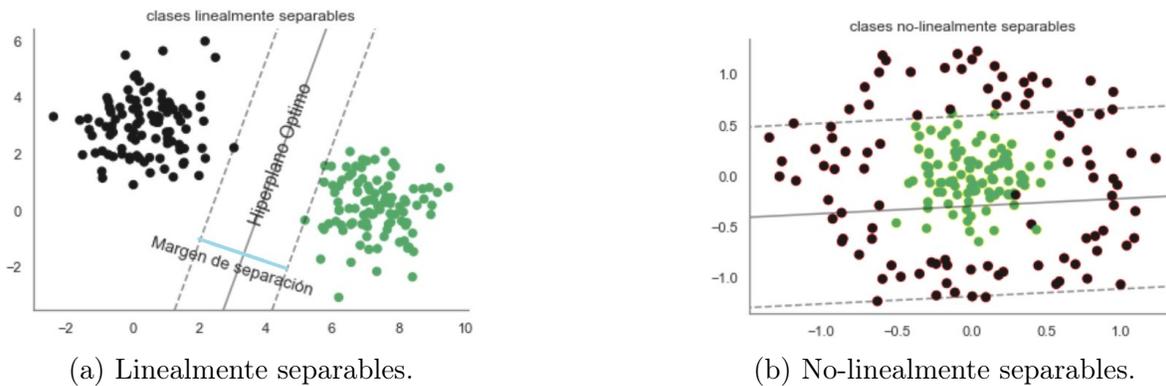


Figura 3.18: Problema lineal y no-linealmente separable (fuente: autores).

### 3.5.3 Árbol de decisión

El árbol de decisión es un modelo de predicción cuyo objetivo principal es el aprendizaje inductivo a través de observaciones y construcciones lógicas, similar a los sistemas de predicción basados en reglas, para la solución de un problema, estos modelos representan y categorizan una serie de condiciones de forma sucesiva.

La representación que se hace es de un árbol, debido al conocimiento obtenido en un proceso de aprendizaje inductivo, este se representa gráficamente por un conjunto de nodos, hojas y ramas. El nodo principal es donde se comienza el proceso de clasificación, los nodos internos representan a cada una de las etapas donde se realizan las preguntas acerca del atributo en particular del problema y cada respuesta de estos nodos internos es un nodo hijo que representa una posible respuesta de estos nodos. Las ramas que salen de estos nodos, están etiquetadas con posibles valores del atributo. Los nodos finales, llamados nodos hojas, corresponden a una decisión, coincidiendo con una variable clase del problema a resolver [56].

El árbol de decisión tiene como objetivo final predecir la variable respuesta  $Y$  en función de covariables, estos están clasificados en dos tipos [7].

- Árboles de regresión donde la variable respuesta es cuantitativa.
- Árboles de clasificación donde la variable respuesta es cualitativa.

En la Figura 3.19 se representa gráficamente el modelo de árbol junto con la partición (fronteras de decisión) del espacio de características tomando en cuenta dos atributos  $x_1, x_2$

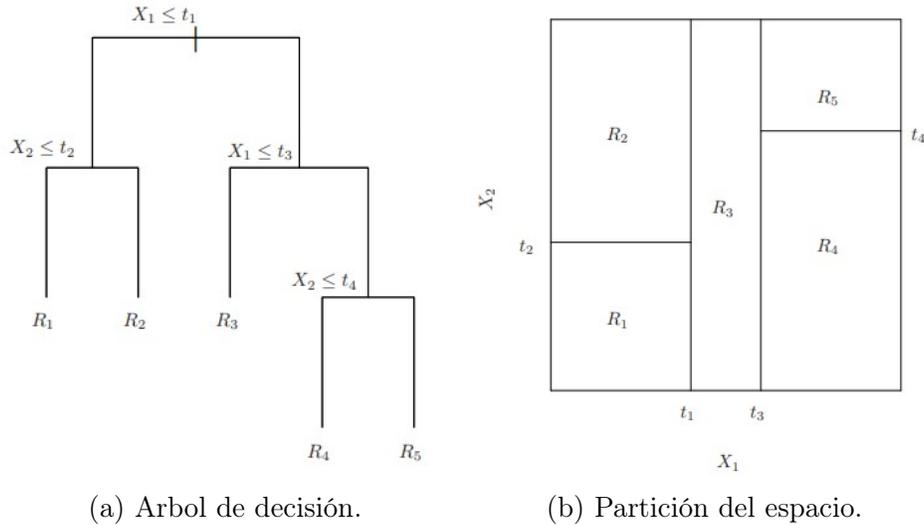


Figura 3.19: Árbol de clasificación a la izquierda y partición del espacio a la derecha [7].

Un árbol de regresión es un modelo que divide el espacio de covariables en hiperrectángulos mediante preguntas del tipo ' $X_k \leq C$ ' para covariables cuantitativas o del tipo ' $X_k \leq nivel_j$ ' para covariables cualitativas. Cada hiperrectángulo representa un espacio donde las observaciones tienen un valor estimado de  $\hat{y}$  similar. Este proceso de división se realiza de manera iterativa buscando las variables y los valores de corte que minimicen la siguiente función de costo:

$$\sum_{i=1}^{i=n} (y_i - \hat{y}_i)^2 \quad (3.46)$$

Esto se logra haciendo preguntas sucesivas para cada variable, eligiendo la variable y el valor de corte que produzcan la mayor reducción en la función de costos. De esta manera, se construye un árbol de regresión que se puede interpretar fácilmente como un conjunto de reglas de decisión [7].

### 3.5.4 Redes neuronales artificiales ANN

Las redes neuronales artificiales son sistemas de procesamiento de información que imitan la estructura neuronal del cerebro con el objetivo de reproducir sus capacidades. Gracias a ello, estas redes son capaces de aprender de la experiencia a través de la interpretación de señales o datos externos [9]. La neurona artificial de la Figura 3.20, se compone de las entradas  $x_j(t)$  multiplicadas por sus pesos sinápticos  $w_{ij}$  que indican el nivel de interacción entre cada neurona *presináptica*  $j$  y *postsináptica*  $i$ , donde el conjunto de todos los pesos se pueden representar mediante la matriz  $W \in R^{m \times n}$ .

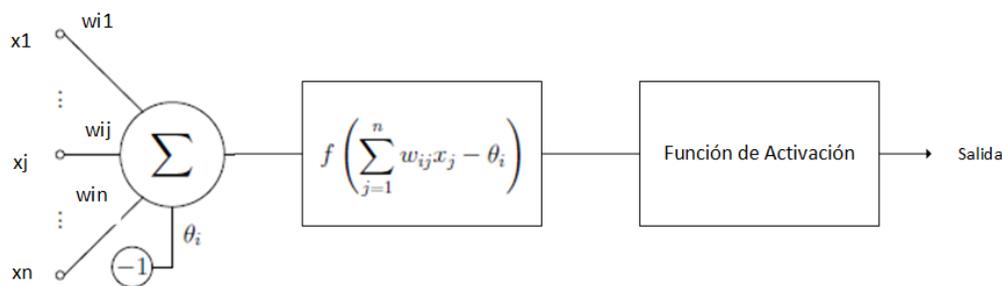


Figura 3.20: Estructura general de una ANN (fuente: autores).

Matemáticamente se puede expresar de la siguiente forma [57]:

$$h_i(t) = \sum_{j=1}^n w_{ij}x_j = w^T \cdot x \quad (3.47)$$

La ecuación 3.47, muestra la regla de propagación lineal que permite expresar la salida postsináptica en función de sus entradas y pesos sinápticos. La suma ponderada de estas entradas se puede expresar como el producto escalar entre los vectores de entrada  $x$  y los pesos  $w^T$ .

La función de activación se le aplica al resultado de la suma ponderada de las entradas y pesos sinápticos, y tiene un parámetro adicional  $\theta$  conocido como *bias*, produciendo así una salida que se utiliza como entrada para otras neuronas en la siguiente capa.

La Tabla 3.1 muestra algunas de las funciones de activación más comunes. Sin embargo, una sola neurona es poco útil para tareas complejas, especialmente tareas de clasificación no lineales. Ello da lugar a las redes neuronales, donde varias neuronas se conectan entre sí formando capas.

<b>Función de Activación</b>	<b>Fórmula</b>
Linear	$f(x) = x$
Sigmoid	$f(x) = \frac{1}{1 + e^{-x}}$
Tangente Hiperbólica (Tanh)	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
ReLU (Rectified Linear Unit)	$f(x) = \text{máx}(0, x)$
Leaky ReLU	$f(x) = \begin{cases} x & \text{si } x > 0 \\ \alpha x & \text{si } x \leq 0 \end{cases}$
Softmax	$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^K e^{x_j}}$

Tabla 3.1: Funciones de activación en redes neuronales artificiales [9].

El perceptrón multicapa (Figura 3.21), ilustra una red neuronal artificial de tipo feedforward, lo que significa que su información fluye en una sola dirección, de la entrada a la salida.

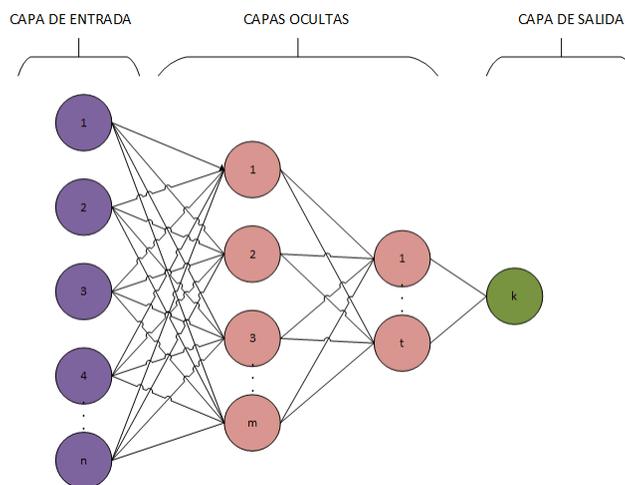


Figura 3.21: Estructura de perceptrón Multicapa (fuente: autores).

La capa de entrada de esta red se conecta directamente a las variables de entrada, mientras que la capa de salida representa la salida deseada de la red. Entre estas

capas se encuentra una o más capas ocultas que procesan la información mediante la aplicación de funciones de activación no lineales a la suma ponderada de las entradas y pesos sinápticos. Esta estructura permite al perceptrón multicapa aproximar funciones complejas y no lineales mediante el aprendizaje a partir de ejemplos. Además, su capacidad de generalización le permite realizar predicciones precisas en datos nunca antes vistos [57].

El proceso de entrenamiento del perceptrón multicapa implica el ajuste iterativo de los pesos sinápticos de la red para minimizar el error de predicción en un conjunto de datos de entrenamiento. Este proceso se lleva a cabo mediante la retropropagación del error, también conocida como *backpropagation*. En primer lugar, se alimenta la red con un conjunto de datos de entrenamiento y se calcula la salida de la red para cada entrada. Luego, se compara la salida de la red con la salida deseada para cada entrada y se calcula el error de predicción. A continuación, se propaga este error hacia atrás a través de la red para actualizar los pesos sinápticos de todas las capas de la red utilizando el algoritmo de optimización de descenso de gradiente. Este proceso se repite iterativamente para todo el conjunto de datos de entrenamiento hasta que el error de predicción es lo suficientemente pequeño. La elección de la función de activación y la arquitectura de la red son importantes para obtener un buen desempeño en la tarea de predicción.

El optimizador Adam, por otro lado, es un algoritmo de optimización utilizado para ajustar los pesos de una red neuronal durante el proceso de entrenamiento. Su objetivo principal es minimizar la función de pérdida o costo de la red neuronal, ajustando los pesos de acuerdo con los gradientes calculados a través del *backpropagation*.

Las funciones de pérdida o de costo, se utilizan para evaluar qué tan bien están prediciendo los valores deseados en comparación con las salidas reales de la red. A continuación, se presentan algunas de las funciones de pérdida más comunes utilizadas en diferentes tipos de problemas:

- Error cuadrático medio (MSE, Mean Squared Error): Es una función de pérdida comúnmente utilizada para problemas de regresión. Calcula el promedio de los errores cuadrados entre las predicciones y los valores verdaderos. El MSE penaliza más fuertemente las predicciones alejadas del valor verdadero.

- Entropía cruzada binaria (binary crossentropy): Es una función de pérdida utilizada en problemas de clasificación binaria. Mide la discrepancia entre las probabilidades predichas y las etiquetas verdaderas. Penaliza los errores de clasificación de manera más significativa y es adecuada cuando hay dos clases posibles.

## 3.6 Medidas del error

### 3.6.1 Medidas estadísticas de error

Las medidas estadísticas de error permiten analizar la frecuencia con la que se clasifican o segmentan de forma errónea los píxeles de la imagen. Esta evaluación se realiza mediante las siguientes cuatro variables:

- Verdaderos positivos (VP): número de píxeles que pertenecen a la región objetivo a segmentar y son clasificados correctamente.
- Verdaderos negativos (VN): número de píxeles que pertenecen al fondo y son clasificados correctamente.
- Falsos positivos (FP): número de píxeles pertenecientes al fondo que son clasificados como parte de la región objetivo.
- Falsos negativos (FN): número de píxeles de la región objetivo que son clasificados como fondo.

En la Figura 3.22, se muestra cada una de las variables teniendo en cuenta la imagen segmentada y la imagen de referencia o ideal:

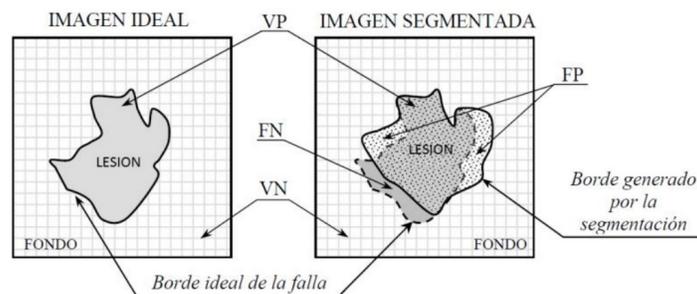


Figura 3.22: Ejemplo de evaluación de zonas segmentadas [8].

La precisión (ecuación 3.48), indica el porcentaje que casos que realmente pertenecen a la región segmentada respecto al total de casos que predichos como positivos. Brinda una idea de la sobresegmentación que existe en el algoritmo propuesto con respecto a la segmentación ideal que debe tener de la región objetivo. La precisión toma un valor de 1 cuando no existen zonas de FP (sobresegmentadas) y valores cercanos a 0 en caso contrario [8].

$$P = \frac{VP}{VP + FP} \quad (3.48)$$

La exactitud (ecuación 3.49), a diferencia de la precisión, obtiene un valor de 1 cuando la segmentación realizada con el algoritmo coincide completamente (la región segmentada es la misma que la ideal) con la realizada por el experto [8].

$$E = \frac{VP + VN}{VP + VN + FP + FN} \quad (3.49)$$

El recall o exhaustividad (ecuación 3.50), da una idea de la subsegmentación que se presenta en la imagen segmentada con respecto a la región ideal. Toma un valor de 1 cuando no existen falsos negativos (subsegmentados) y valores cercanos a cero en caso contrario. Esta medida devuelve el valor máximo en caso de que la segmentación sea idéntica a la región ideal [8].

$$R = \frac{VP}{VP + FN} \quad (3.50)$$

La medida F (ecuación 3.51), indica el rendimiento general de la segmentación en términos de precisión y exhaustividad a partir del cálculo de la media armónica entre las dos medidas.

$$FM = \frac{2 * P * R}{P + R} \quad (3.51)$$

### 3.6.2 Índice de DICE

El índice de DICE, también conocido como índice de *Sorensen*, es una medida que permite evaluar la similitud existente entre dos imágenes binarias [58]. Para un par de

imágenes  $U$  y  $V$  que son las imágenes segmentada y de referencia respectivamente, el índice de DICE tiene la forma de la ecuación 3.52.

$$S = \frac{2|U \cap V|}{|U| + |V|} \quad (3.52)$$

Donde  $|U \cap V|$  es el número de píxeles comunes entre las imágenes  $U$  y  $V$ , indicando que los objetos de interés segmentados en  $U$  se comparan con los objetos de referencia  $V$  mediante la intersección de las imágenes, esto dividido respecto al total de objetos de interés pertenecientes a las dos imágenes.  $S$  obtiene un valor cercano a 1 para una segmentación perfecta, y 0 para una segmentación incorrecta.

### 3.6.3 Distancia de Jaccard

El índice de Jaccard, también conocido como coeficiente de similitud de Jaccard, es una métrica que se utiliza en el contexto de imágenes para comparar los píxeles que pertenecen a clases en relación con su referencia. Para una imagen  $X$  (segmentada) y una imagen  $Y$  (referencia), el índice de Jaccard se calcula de la siguiente manera [58]:

$$J(X, Y) = \frac{X \cap Y}{X \cup Y} \quad (3.53)$$

Donde  $X \cap Y$  son los píxeles que comparten ambas imágenes y  $X \cup Y$  es el número total de píxeles en las imágenes  $X$  y  $Y$ , es decir, unión de  $A$  y  $B$ .

Con lo anterior, la distancia de Jaccard se define como el complemento probabilístico del término anterior. Esta métrica se expresa mediante la ecuación 3.54 y representa la disimilitud entre las dos imágenes.

$$J_\delta(X, Y) = 1 - J(X, Y) \quad (3.54)$$

Por tanto, la mejor segmentación se da cuando el valor del coeficiente es cercano a 0 y la peor cuando se acerca a 1.

## 4. Desarrollo o solución

En este capítulo se describe detalladamente la metodología empleada en el desarrollo del estudio. Se presenta la estructura general del proceso, que abarca desde la adquisición de las imágenes hasta la clasificación de células, pasando por el preprocesamiento, la segmentación y la evaluación de la misma. Se explica en profundidad cada uno de los pasos, justificando la elección de las técnicas y algoritmos empleados en cada uno de ellos, proporcionando así, una visión completa de la metodología utilizada para la realización del estudio, lo que permitirá a los lectores comprender mejor el proceso de investigación y evaluar la validez de los resultados obtenidos.

En la Figura 4.1, se muestra la metodología de trabajo teniendo en cuenta el ciclo completo del procesamiento de imágenes.

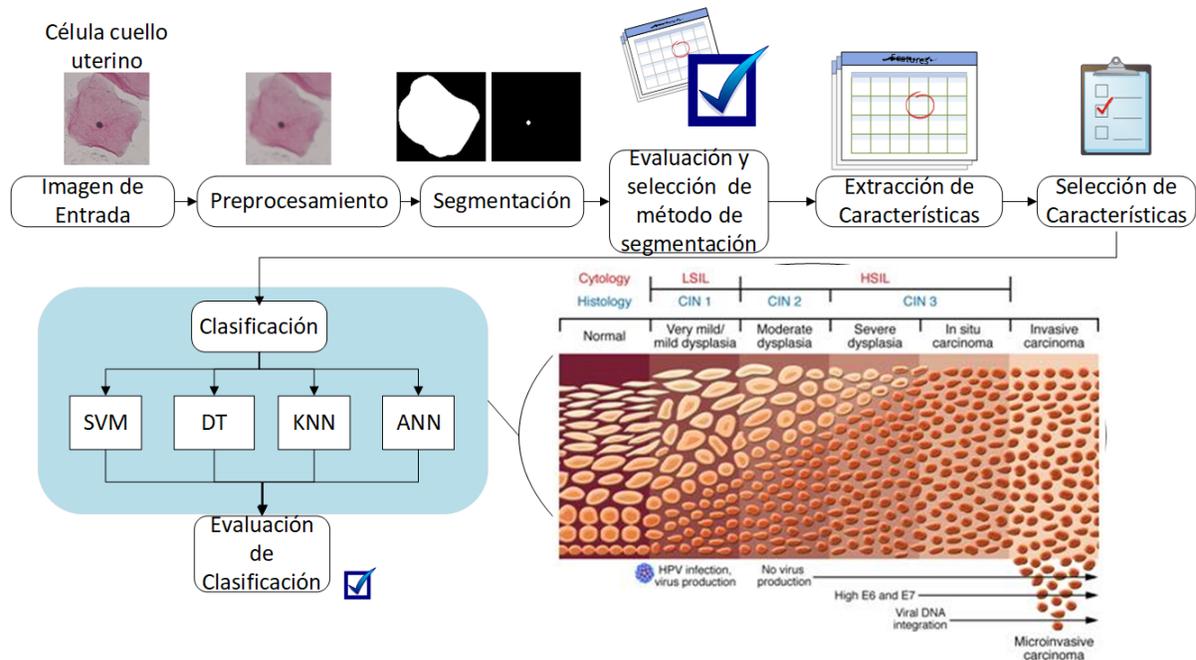


Figura 4.1: Metodología general (fuente: autores).

El preprocesamiento de una imagen de entrada implica realizar operaciones de filtrado que preparan la imagen para segmentar el núcleo y el citoplasma. Se proponen

tres soluciones para abordar el problema de segmentación, donde dos de ellas corresponden a técnicas de aprendizaje no supervisado para el núcleo, mientras que la tercera solución utiliza aprendizaje supervisado tanto para el núcleo como para el citoplasma. Se evalúa y selecciona el método que logre mejores resultados de segmentación con el fin de extraer y seleccionar características representativas. Finalmente, se realiza la clasificación con 4 clasificadores diferentes para tareas binarias y de tres clases, los cuales se evalúan y se seleccionan aquellos con mejor rendimiento.

Por otro lado, cada una de las etapas de procesamiento fue desarrollada con python, principalmente con las librerías de *opencv* para procesamiento de imágenes, *scikit-learn* para modelos de aprendizaje automático, así como *numpy*, *pandas*, y *seaborn* para operaciones matriciales y generación de gráficas, entre otras. Se tienen los siguientes recursos computacionales:

- Procesador intel core i7 5500 U @ 3GHz
- Tarjeta gráfica NVIDIA GEFORCE 940M 2BG, núcleos CUDA 384
- RAM 8 GB
- Sistema operativo Windows 11
- Python v3.10

#### 4.1 Imágenes de entrada: conjunto de datos HERLEV

Se eligió el conjunto de datos de HERLEV debido a que es una base de datos pública que ofrece una amplia variedad de lesiones cervicales, que van desde células normales hasta cáncer cervical, previamente clasificadas por expertos. Esta base de datos se ha utilizado como referencia estándar en la investigación médica, lo que la convierte en una herramienta valiosa para el análisis y la comparación de diferentes tipos de lesiones cervicales. El conjunto de datos de HERLEV de imágenes PAP consta de un total de 917 imágenes de células de la prueba de Papanicolaou. En la Tabla 4.1 se muestra la distribución de las diferentes clases, siendo las clases 1 a 3 aquellas que contienen células

normales y las clases 4 a 7 las que contienen células anormales. La distribución de las imágenes es desigual entre las categorías, con un mayor número de células anormales. Además, cada imagen contiene una única célula, lo que facilita una evaluación individual y detallada de cada una.

Clase	Categoría	Tipo de Célula	Recuento de Células
1	Normal	Epitelio escamoso superficial	74
2	Normal	Epitelio escamoso intermedio	70
3	Normal	Epitelio columnar	98
4	Anormal	Displasia escamosa leve no queratinizante	182
5	Anormal	Displasia escamosa moderada no queratinizante	146
6	Anormal	Displasia escamosa no queratinizante grave	197
7	Anormal	Carcinoma de células escamosas in situ intermedio	150
<b>Subtotal normal</b>			<b>242</b>
<b>Subtotal anormal</b>			<b>675</b>

Tabla 4.1: Resumen de clases y tipos de células en el conjunto de datos de HERLEV (fuente: autores).

Caracterización de las células en cada una de las clases:

- Escamoso superficial:

Núcleo bastante pequeño en comparación con el resto de la célula, forma plana/ovalada, tamaño relativo entre núcleo y citoplasma ( $N/C$ ) muy pequeño.

- Escamoso intermedio:

Forma redonda, núcleo grande,  $N/C$  pequeño.

- Columnar:

Forma alargada y estrecha, núcleo grande,  $N/C$  mediano.

- Displasia leve:

Núcleo claro y grande,  $N/C$  mediano.

- Displasia moderada:

Núcleo grande y oscuro, citoplasma oscuro,  $N/C$  grande.

- Displasia grave:

El núcleo es grande, deforme y oscuro, citoplasma oscuro,  $N/C$  muy grande.

- Carcinoma in Situ:

El núcleo es grande, deforme y oscuro, citoplasma casi inexistente a simple vista,  $N/C$  muy grande.

## 4.2 Preprocesamiento

En la Figura 4.2, se presentan las soluciones para segmentación de núcleo y citoplasma:

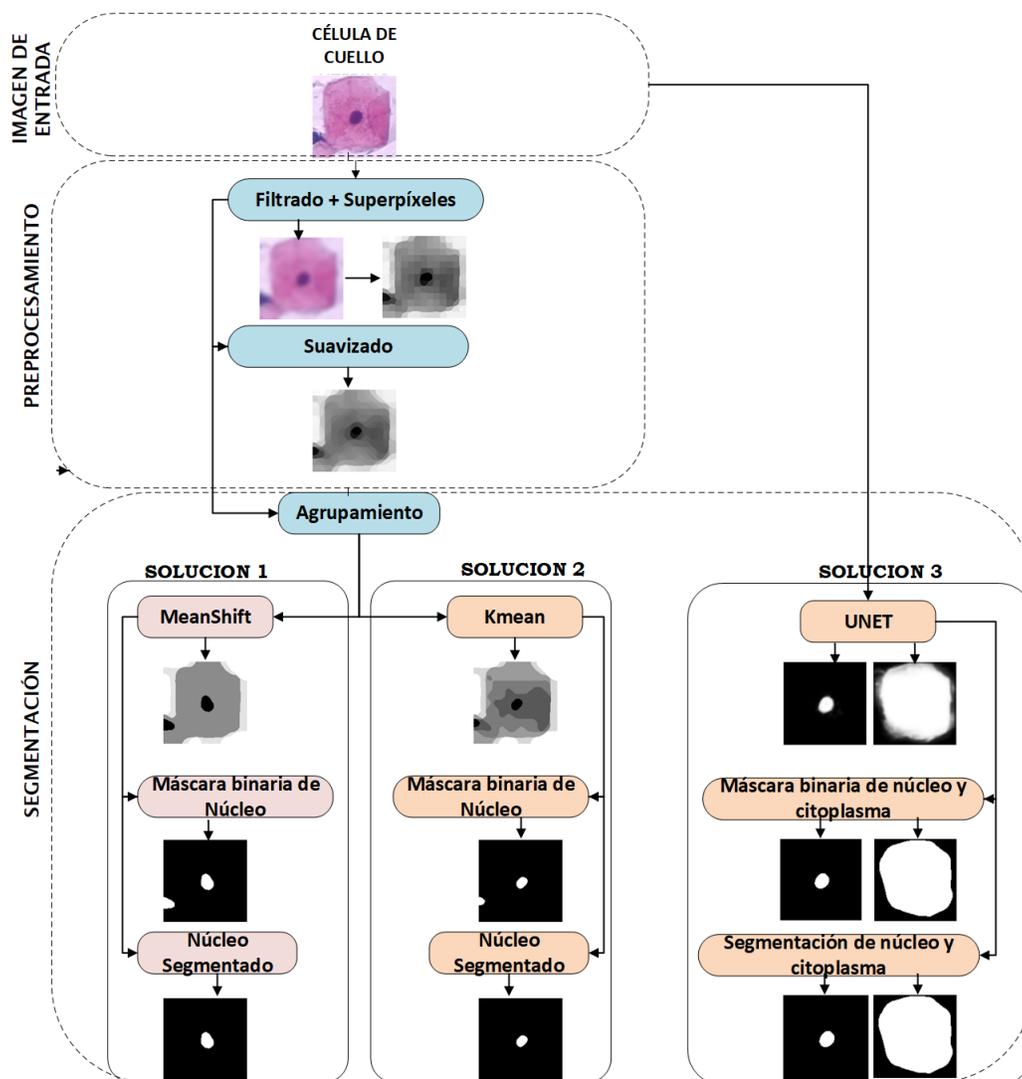


Figura 4.2: Estructura de preprocesamiento y segmentación para las soluciones 1, 2 y 3 (fuente: autores).

Como se puede apreciar en la Figura 4.2, las dos primeras soluciones presentan una característica común en el preprocesamiento de la imagen de entrada, que consiste en la aplicación de operaciones de filtrado para obtener una imagen de salida en forma de superpíxeles. La tercera solución, está basada en segmentación por CNN, la cual no requiere de una etapa previa de preprocesamiento.

En la Figura 4.3, se muestra una comparación entre tres filtros diferentes, cada uno con un *kernel* de  $21 \times 21$ . Los tres filtros son capaces de reducir el ruido en la imagen y mejorar su calidad. Sin embargo, en este tipo de imágenes donde es crucial preservar los bordes, es importante seleccionar cuidadosamente el tipo de filtro a utilizar.

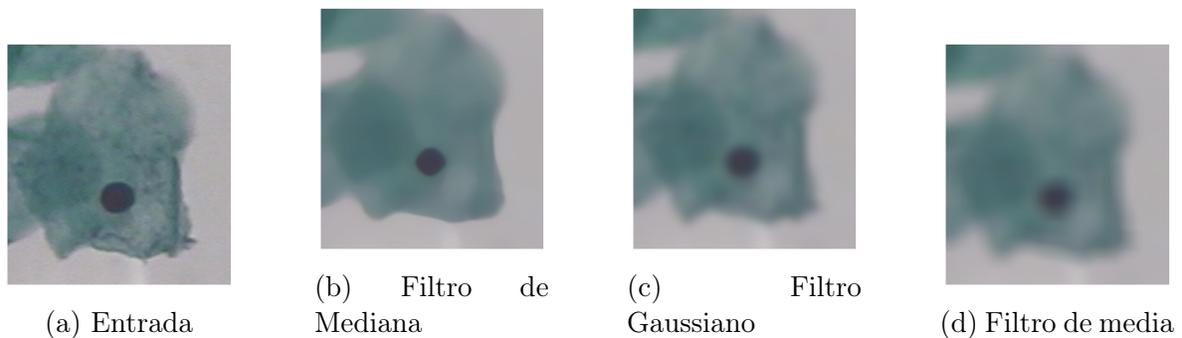


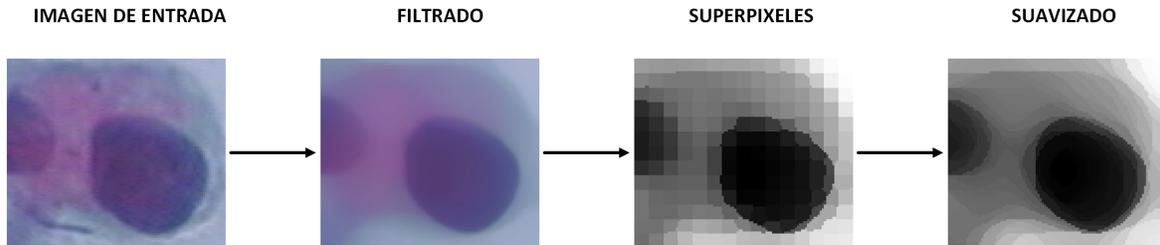
Figura 4.3: Comparación entre filtros (fuente: autores).

En este caso, se optó por utilizar el filtro de mediana debido a que presenta varias ventajas sobre los otros filtros. En primer lugar, el filtro de mediana es capaz de conservar los bordes de la imagen al no difuminar los detalles finos del mismo. Esto es importante para mantener la nitidez y la calidad de la imagen en zonas críticas. Además, el filtro de mediana es capaz de eliminar los puntos blancos y negros (ruido) aleatorios presentes en la imagen, lo que es especialmente útil en zonas de alto contraste.

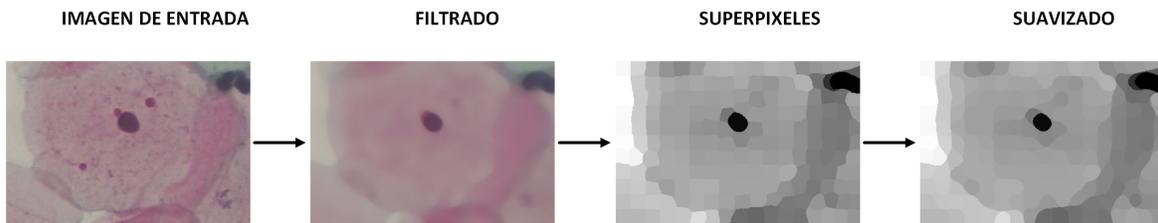
Por otro lado, los superpíxeles permiten simplificar la complejidad de la imagen al agrupar píxeles cercanos con características similares en regiones más grandes y homogéneas, lo que permite una representación más sencilla y fácil de procesar de la imagen.

La Figura 4.4 muestra que, tras el proceso de filtrado, se genera una imagen en escala de grises compuesta por superpíxeles que permiten identificar con mayor precisión las

regiones correspondientes al núcleo, los cuales se destacan mediante un tono oscuro. Al aplicar el suavizado nuevamente con el filtro de mediana a la imagen de superpíxeles, se obtiene una ventaja importante, ya que los superpíxeles con intensidades de color similares se unen para formar regiones aún más grandes. Esto resulta en un núcleo más uniforme, con una menor variación de intensidades.



(a) Superpíxeles y suavizado de célula anormal.



(b) Superpíxeles y suavizado de célula normal.

Figura 4.4: Fase de preprocesamiento (fuente: autores).

Se utilizó el algoritmo agrupamiento iterativo lineal simple (SLIC) [59] para generar una imagen con superpíxeles, donde es importante ajustar los parámetros de *compactness* y *nsegments* adecuadamente, especialmente para células de tipo normal, como se puede observar en la Figura 4.4b. Si los superpíxeles generados son muy grandes, pueden no representar adecuadamente los núcleos celulares, que son más pequeños que el propio superpíxel, lo que puede resultar en una sobresegmentación del núcleo. Mientras que para una célula anormal (Figura 4.4a), no se ve afectada por esta sobresegmentación debido a que, por lo general, estas presentan un núcleo mayor que las clases normales y la información del núcleo sigue estando representada por superpíxeles con intensidades similares.

En este estudio, se generan superpíxeles con una *compactness* de 30, lo que permite que se ajusten mucho mejor a los bordes, y con 185 *nsegments*. Al establecer un valor mayor para *nsegments*, se generan más superpíxeles y, por lo tanto, la segmentación de la imagen es más fina y detallada. Si se establece un valor menor para *nsegments*, se

generan menos superpíxeles y la segmentación es más gruesa y generalizada.

El parámetro *compactness* se utiliza para determinar cuánto se prioriza la proximidad espacial entre los píxeles y la similitud de color al formar los superpíxeles. Un valor más alto de *compactness* hace que los superpíxeles sean más compactos y cuadrados, lo que significa que su tamaño está más influenciado por la proximidad espacial entre los píxeles. Un valor más bajo de *compactness* hace que los superpíxeles sean más irregulares y no cuadrados, y su tamaño está más influenciado por la similitud de color entre los píxeles.

### 4.3 Segmentación celular

En este punto, el desafío consiste en separar el objeto de interés, como el núcleo y el citoplasma de las 7 clases de células presentes en HERLEV. Las primeras dos soluciones abordan este problema de segmentación de núcleo utilizando el concepto de *clustering* o agrupamiento, mediante técnicas de aprendizaje no supervisado. Estas técnicas permiten agrupar los píxeles de la imagen en función de su similitud en términos de color, intensidad, textura y otros atributos, lo que facilita la identificación y separación de los objetos en la imagen.

La tercera solución propuesta utiliza la segmentación semántica basada en redes neuronales convolucionales. Esta técnica se aplica para la segmentación del núcleo y citoplasma de una imagen. En este enfoque, cada píxel de la imagen es analizado y se le asigna una clasificación semántica correspondiente, de manera que se pueda identificar a qué clase de objeto pertenece. De esta forma, se logra obtener una segmentación precisa de los objetos de interés en la imagen.

#### 4.3.1 Solución 1

A partir de una imagen de superpíxeles con suavizado, se busca que el núcleo sea mucho más separable. Se propone el algoritmo de agrupamiento por desplazamiento medio MEANSHIFT, que permite aprovechar la información de alta densidad del núcleo para agruparse en un solo grupo.

En la Figura 4.5 se observa que después de aplicar el preprocesamiento a la imagen y pasarla por el algoritmo de segmentación MEANSHIFT, los píxeles que corresponden al núcleo celular han sido asignados a un único grupo. Esto se puede apreciar en la imagen del medio donde el núcleo aparece representado en un tono oscuro.

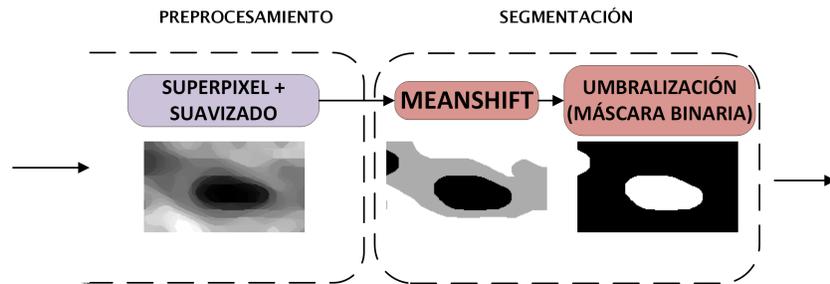


Figura 4.5: Agrupamiento por MEANSHIFT (fuente: autores).

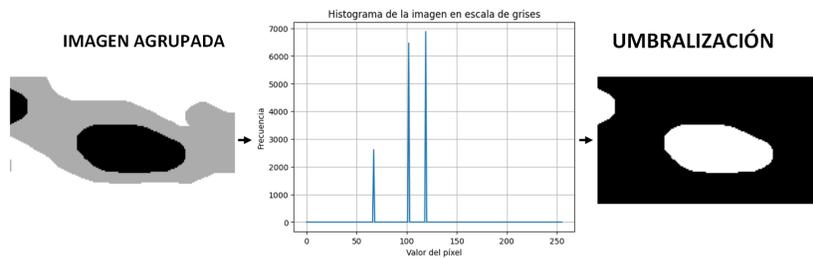
En este caso, se utiliza la biblioteca *scikit-learn* de Python, que permite configurar de forma sencilla los hiperparámetros del algoritmo MEANSHIFT. Una de las ventajas de este método es que no requiere conocer de antemano el número de grupos presentes en la imagen, sino que los determina automáticamente según la información de densidad presente en el espacio de características.

Para el ajuste de los hiperparámetros, se debe tener en cuenta que generalmente los algoritmos de agrupamiento trabajan con datos vectoriales. Por tanto, el primer paso es convertir la imagen obtenida en el preprocesamiento en un vector, donde el valor de intensidad de cada píxel conforma el vector de características. El siguiente paso es determinar el ancho de banda adecuado, el cual determina la escala o distancia máxima en la que se busca la densidad de los datos. Para ello, se utiliza la función *estimate bandwidth*, que permite ajustar el ancho de banda óptimo a partir del vector de características. Con ello, ya se puede instanciar un objeto del algoritmo MEANSHIFT para que realice la operación de agrupamiento. Posteriormente, se calculan los centroides de dichos grupos y cada píxel se designa con la etiqueta del centroide al que pertenece. Finalmente, el vector devuelto con los grupos conformados se convierte nuevamente en una imagen con las dimensiones originales, obteniendo así la imagen central de la Figura 4.5.

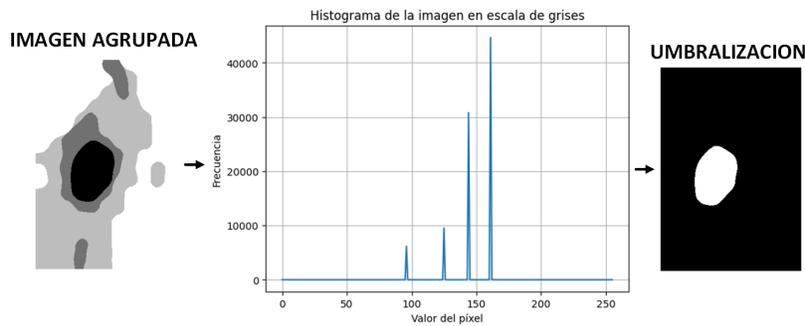
El paso siguiente en el proceso de segmentación consiste en extraer únicamente la parte del núcleo de la imagen, ignorando el resto. Esto se logra mediante la selección de

un valor umbral, donde los píxeles con valores de intensidad iguales o superiores a dicho umbral se representan en blanco, lo que se interpreta como parte del núcleo, mientras que los valores inferiores se representan en negro, generando así una imagen binaria.

Haciendo un análisis a partir de la distribución de intensidad de la imagen agrupada (Figura 4.6), se puede observar que la célula de la Figura 4.6a, se ha dividido en tres partes, donde el citoplasma y el fondo corresponden a las áreas más grandes, seguidas por el núcleo representado en tono oscuro. La imagen adjunta muestra el histograma, donde el eje horizontal representa los valores de los píxeles desde 0 (negro) hasta 255 (blanco), y el eje vertical indica la frecuencia con la que aparece cada valor de píxel.



(a) Célula anormal con tres grupos.



(b) Célula anormal con cuatro grupos.

Figura 4.6: Análisis de intensidad a partir del histograma (fuente: autores).

A partir de lo expuesto anteriormente, se puede ver que en la célula de la Figura 4.6a, el núcleo presenta un tono oscuro que se ubica alrededor de una intensidad de 60. Por lo tanto, al utilizar un umbral de intensidad mayor o igual a 70 se logra obtener una segmentación adecuada. Sin embargo, en el caso de la célula de la Figura 4.6b, el umbral mencionado anteriormente no es el mismo, ya que dicha imagen presenta cuatro grupos y un valor de intensidad de núcleo distinto.

Se puede inferir que el umbral óptimo para la segmentación varía de manera particular para cada imagen. Para solucionar este problema, se debe tener en cuenta que una de las características de estas células PAP es que el núcleo presenta un tono más oscuro que el citoplasma debido a su contenido genético. En el momento del agrupamiento, dicho núcleo se agrupará en un tono más oscuro que el citoplasma. Por lo tanto, la estrategia consiste en tomar el valor umbral mínimo  $u_{min}$  de intensidad de la imagen agrupada y sumarle un valor de seguridad  $b$  adicional para el caso en el que algunas regiones pertenecientes al objeto de interés, se hayan agrupado en un tono ligeramente diferente, siendo la suma de estos dos valores el umbral óptimo  $u_{opt} = u_{min} + b$ . De esta manera, se obtienen las imágenes en blanco y negro de la Figura 4.6 con su máscara binaria. Se puede ver el notebook de Jupyter para MEANSHIFT en el siguiente [enlace](#).

#### 4.3.2 Solución 2

Se propone una técnica de segmentación celular utilizando el algoritmo de KMEANS. Esta técnica permite dividir la imagen en diferentes regiones mediante la asignación de centroides y el etiquetado de píxeles. En el presente caso, se aplica el algoritmo a una imagen previamente filtrada y generada con superpíxeles. Estos superpíxeles facilitan la agrupación en distintas regiones o grupos utilizando el algoritmo de KMEANS. A continuación, se realiza el etiquetado de cada superpíxel para identificar a qué grupo pertenece dentro de las regiones establecidas.

En la Figura 4.7, se muestra la propuesta de segmentación con KMEANS:

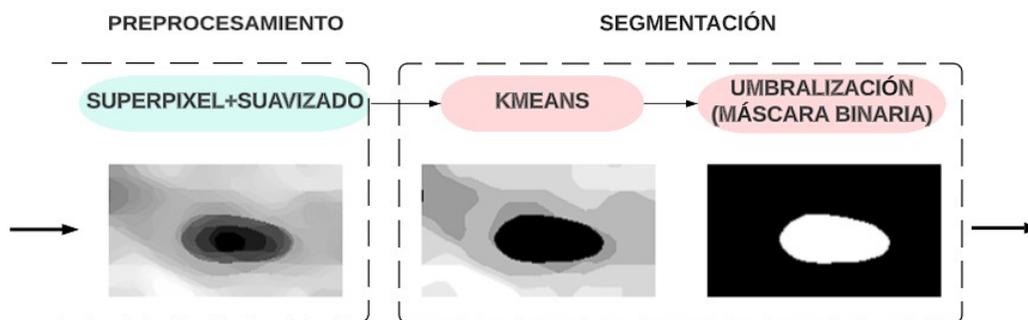
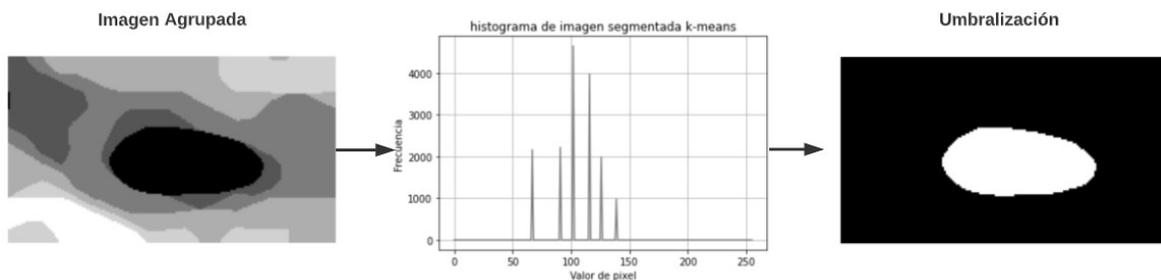


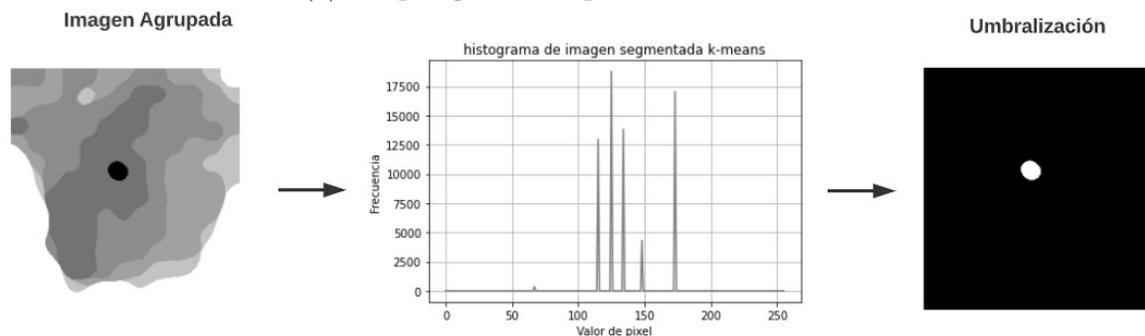
Figura 4.7: Estructura de segmentación usando KMEANS (fuente: autores).

En el proceso de segmentación de imágenes, es necesario contar con información previa sobre el número de regiones o grupos presentes en la imagen. En el caso de las imágenes analizadas en la Figura 4.7, resulta evidente la presencia de tres regiones distintas, como el fondo, el núcleo y el citoplasma. No obstante, en situaciones donde la diferenciación entre el citoplasma y el fondo es poco clara, el resultado generado es una subsegmentación, fusionando regiones que deberían haber sido agrupadas por separado. Se determinó de forma experimental que al utilizar 6 grupos en lugar de los 3 grupos esperados, el algoritmo logró representar de manera más precisa el núcleo celular, asignándolo a un solo grupo. Esta observación se puede apreciar en la imagen intermedia de la Figura 4.7. El siguiente paso es extraer la región perteneciente al núcleo representándolo en una imagen binaria. Del mismo modo que la solución 1, esto se logra aplicando un umbral de intensidad en escala de grises a partir del análisis del histograma de la imagen agrupada.

Para las células de la Figura 4.8, al analizar sus respectivos histogramas, se puede observar la presencia de 6 grupos con diferentes intensidades y número de píxeles.



(a) Grupos generados para célula anormal.



(b) Grupos generados para célula normal.

Figura 4.8: Análisis de intensidad a partir del histograma con KMEANS (fuente: autores).

El histograma representa la distribución de intensidades en escala de grises de la

imagen agrupada, que varía desde 0 (negro) hasta 255 (blanco). Es notable que el núcleo de la célula corresponde al tono más oscuro presente en la imagen.

Con base en esta observación, se puede establecer un umbral para extraer únicamente el núcleo de la célula. Este umbral, al igual que la primera solución, denotado como  $u_{opt} = u_{min} + b$ , se define como el valor mínimo de intensidad presente en la imagen ( $u_{min}$ ) más un valor constante de seguridad ( $b$ ). Al aplicar este umbral, se obtiene una imagen binaria donde únicamente se representa el núcleo de la célula, en contraste con el resto de la imagen. Se puede ver el notebook de Jupyter para KMEANS en el siguiente [enlace](#).

### 4.3.3 Solución 3

En la Figura 4.9, se propone la solución para la segmentación de núcleo y citoplasma mediante segmentación semántica, tomando como base la estructura de UNET. Para la implementación del algoritmo, se utilizará la biblioteca Keras de Python y la estructura dispuesta en [60], que facilita la creación y entrenamiento de modelos de aprendizaje automático.

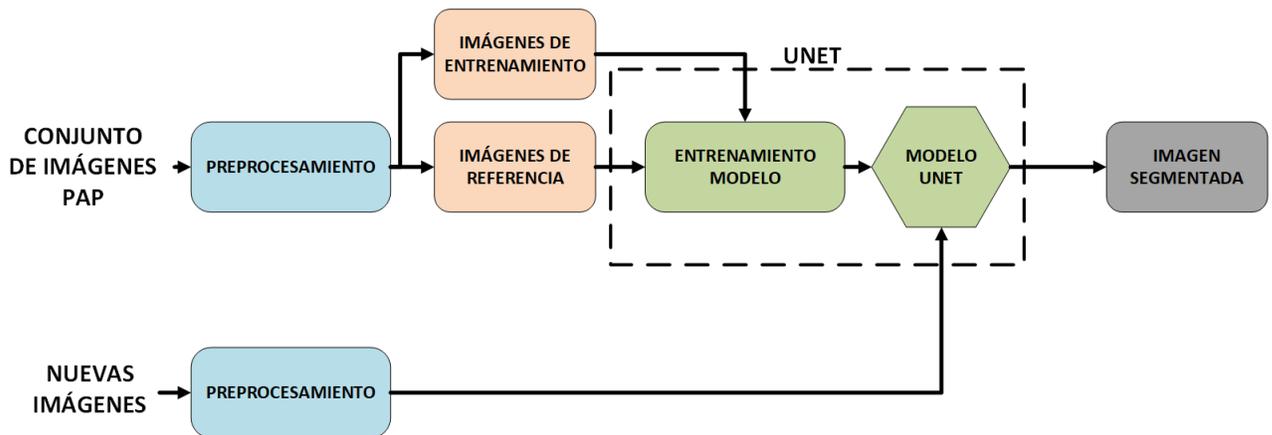
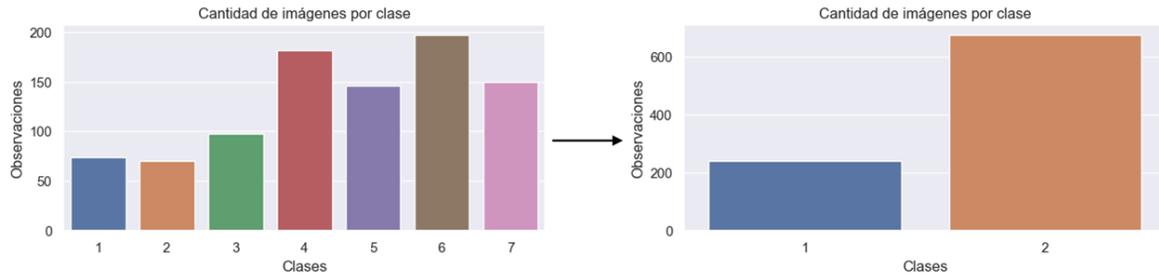


Figura 4.9: Estructura de segmentación usando UNET (fuente: autores).

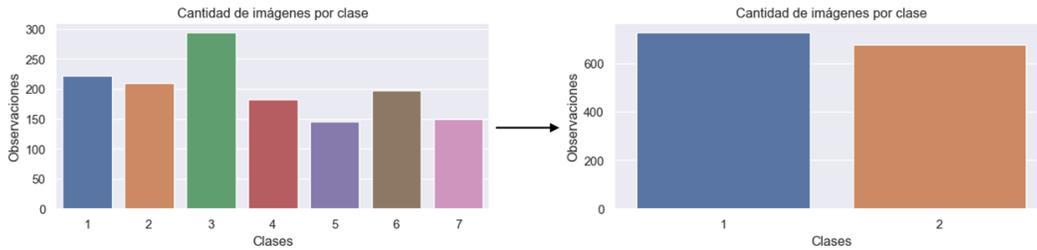
#### 4.3.3.1 Aumento de datos para segmentación UNET

En la Figura 4.10, se puede observar el número de imágenes que contiene cada clase de la base de datos de HERLEV. Las tres primeras clases se pueden agrupar como

células normales, mientras que las clases 4, 5, 6, 7 corresponden a la categoría anormal.



(a) Clases desbalanceadas.



(b) Clases balanceadas.

Figura 4.10: Análisis del balance de clases (fuente: autores).

De acuerdo a la Figura 4.10a, es posible notar que las clases normales en comparación con las anormales están bastante desbalanceadas, lo que puede provocar que el modelo aprenda a segmentar principalmente la clase mayoritaria. Por lo tanto, es necesario realizar un aumento de datos en las tres primeras clases. Para el proceso de balanceo, se aplica un giro de  $90^\circ$  a las imágenes de las clases normales. Esta técnica permite conservar los mismos detalles sin modificar su escala, logrando así un equilibrio entre las clases normales y anormales (Figura 4.10).

#### 4.3.3.2 Creación del conjunto de entrenamiento y prueba

Con el aumento de datos, el conjunto total cuenta ahora con 1159 imágenes, de las cuales el 20% se destinan para prueba y el 80% restante para entrenamiento.

#### 4.3.3.3 Transformación de imágenes

Es importante destacar que la base contiene imágenes PAP, todas ellas etiquetadas y con su correspondiente máscara de segmentación. Sin embargo, para utilizar el algoritmo UNET, es necesario que todas las imágenes de entrada y de referencia tengan las mismas dimensiones. Por lo tanto, se realizó un redimensionamiento a una imagen cuadrada de  $128 \times 128$  píxeles.

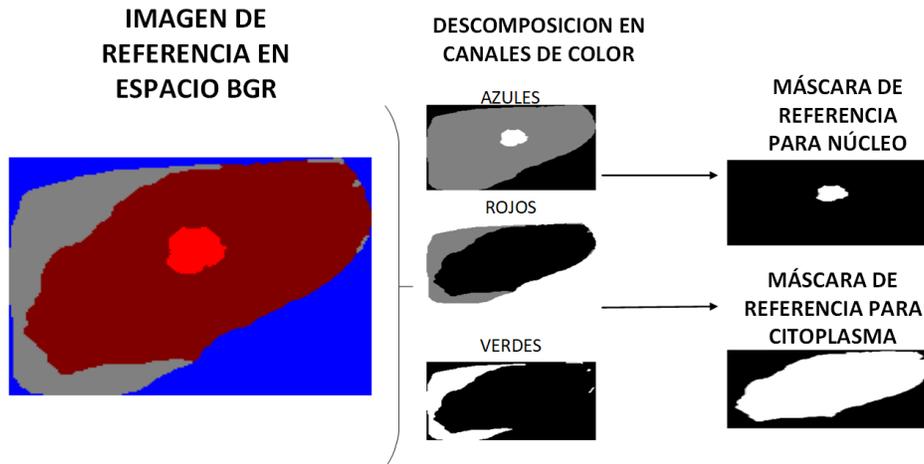
Para el preprocesamiento de las imágenes de referencia del núcleo (Figura 4.11a), partiendo de una imagen en el espacio de color BGR (azul,verde,rojo), el plano que permite la distinción del núcleo es el canal de azules, que convertido a escala de grises se comporta como la imagen central de la Figura 4.11a. Haciendo el análisis de la distribución de color, Figura 4.11b, el núcleo está representado en un tono claro con una intensidad aproximada de 255, por lo que un umbral de 240 en el canal de azules lograría obtener una máscara binaria del núcleo.

Del mismo modo, el plano de rojos es el que representa mejor el citoplasma, el cual está representado por un tono oscuro que abarca la mayor parte de la imagen, con un valor de intensidad de cero. Nuevamente se realiza la umbralización, donde aquellos valores superiores a 100 (fondo y región en gris) se representarán en la máscara como fondo oscuro, mientras que aquellos valores iguales a cero se representarán en blanco, obteniendo así una máscara binaria para el citoplasma.

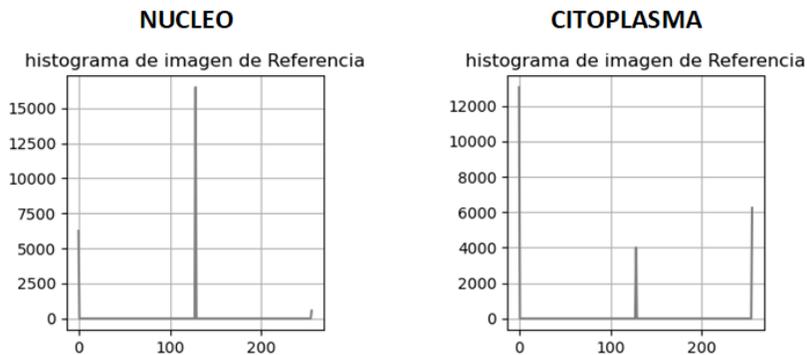
#### 4.3.3.4 Entrenamiento UNET

Dado que UNET es una CNN, es necesario realizar el entrenamiento para que aprenda a segmentar las regiones. El conjunto de datos de entrada corresponde a 927 imágenes con una resolución de  $128 \times 128$  píxeles, con 3 canales de color (BGR), formando así un tensor de cuatro dimensiones (927, 128, 128, 3).

El tensor con las imágenes de referencia tiene una forma de (927, 128, 128, 1), lo que significa que consta de 927 imágenes de tamaño  $128 \times 128$  y un solo canal de color (es decir, máscaras binarias). Por otro lado, el tensor de prueba tiene una forma de (231, 128, 128, 3), lo que significa que consta de 231 imágenes de tamaño  $128 \times 128$  con tres



(a) Máscara de núcleo y citoplasma.



(b) Distribución de color en escala de grises de planos verdes y azul.

Figura 4.11: Procesamiento de imagen de referencia del núcleo y citoplasma (fuente: autores).

canales de color (imágenes en formato BGR).

El modelo UNET entrenado se utilizará para segmentar las imágenes de prueba, lo que significa que se utilizará para identificar las regiones de interés en cada imagen (núcleo y citoplasma) mediante dos algoritmos. Entre los parámetros configurables del entrenamiento, se tiene el optimizador *adam* para el ajuste de pesos de la red neuronal, con la función de pérdida *binary crossentropy* especialmente para clasificación binaria, con función de activación *sigmoid* a la salida de la red que indica la probabilidad de pertenencia a la clase positiva en un rango no lineal de 0 a 1 y finalmente la exactitud como medida de desempeño que indica la proporción de predicciones correctas (verdaderos positivos y verdaderos negativos) sobre el total de muestras de datos. El entrenamiento

se lleva a cabo con 20 épocas y paquetes de validación del 10 %.

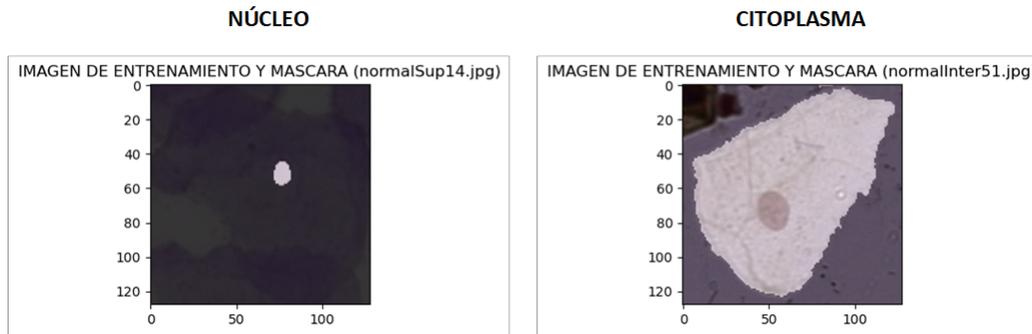


Figura 4.12: Segmentación de referencia en la imagen original (fuente: autores).

En la Figura 4.12, se puede observar en un tono claro la segmentación de referencia sobre la imagen real.

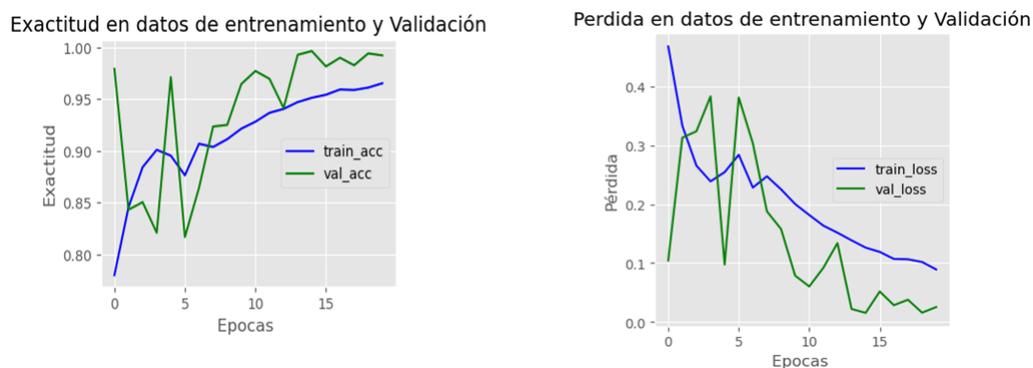


Figura 4.13: Métricas de entrenamiento UNET para núcleo (fuente: autores).

La Figura 4.13 muestra los resultados del entrenamiento para la segmentación del núcleo, donde se observa que la exactitud en los datos de entrenamiento y validación supera el 95%. Es decir, el modelo logró segmentar correctamente las imágenes más del 95% de las veces en relación al total de imágenes. Por otro lado, la función de pérdida tanto en los datos de entrenamiento como en validación muestra una tendencia decreciente a lo largo de las iteraciones, lo que indica que el modelo aprendió a generalizar y a mejorar sus predicciones.

La Figura 4.14 muestra los resultados del entrenamiento para la segmentación del citoplasma. La gráfica de exactitud muestra que tanto en los datos de entrenamiento como en validación se alcanza un 90% después de 20 épocas, lo que indica que acertó

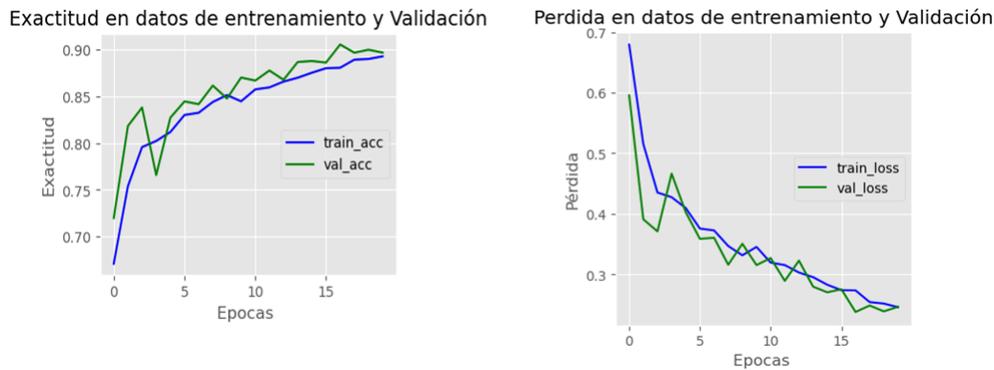


Figura 4.14: Métricas de entrenamiento UNET para citoplasma (fuente: autores).

el 90 % de las veces en relación al total, y se observa que no están muy distantes entre sí, lo que hace que el resultado de la exactitud sea aún más confiable. Del mismo modo, la pérdida a través de las épocas tiende a decrecer, alcanzando un valor de 0.24, y se encuentran bastante juntas, lo que indica que el modelo se está ajustando adecuadamente a los datos. El modelo presenta una buena capacidad de generalización y debería ser capaz de hacer predicciones precisas en nuevos datos.

#### 4.3.3.5 Segmentación en imágenes de Prueba

En la Figura 4.15 se muestran algunas imágenes que fueron segmentadas por los modelos UNET. La columna denominada *segmentación*, corresponde a la imagen de salida de la red Neuronal. Cada píxel blanco de las imágenes en esta columna indica la probabilidad de pertenencia a la clase positiva, mientras que los píxeles negros indican una probabilidad menor. Por tanto, la imagen es en realidad una representación de probabilidades.

Para extraer el objeto de interés, es necesario convertir la imagen de probabilidades, en otra imagen en escala de grises que varíe entre 0 y 255. Posteriormente, se realiza una umbralización utilizando el método de Otsu [61], que consiste en el ajuste óptimo y automático del umbral según la maximización de varianza de los niveles de gris del objeto y fondo, separando de forma efectiva el objeto de interés.

También se muestra una imagen de bordes que encierra el objeto de interés y una vista general de la segmentación en superposición con la imagen de prueba. Se puede

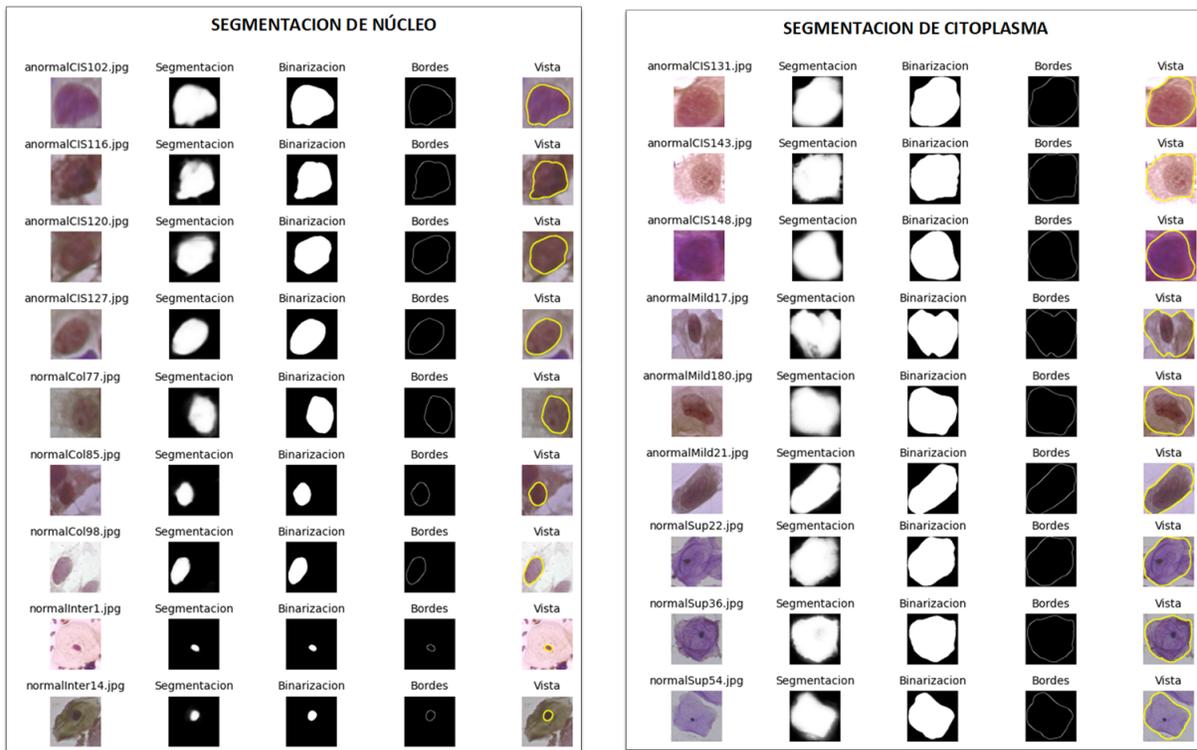


Figura 4.15: Segmentación en imágenes de prueba (fuente: autores).

ver el notebook de Jupyter para UNET en el siguiente [enlace](#).

#### 4.3.4 Segmentación final

La Figura 4.16 muestra que en la segmentación se presentan casos en los que el núcleo y/o el citoplasma contienen orificios en su interior, y/o aparecen más núcleos (pseudonúcleos) debido al proceso de umbralización.

Haciendo un análisis de las imágenes de la base de datos de HERLEV, se puede observar que en las fotografías tomadas el núcleo se encuentra generalmente en la parte central de la imagen, rodeado por el citoplasma.

Con base en esta observación, la Figura 4.17 presenta la estrategia para abordar el problema de los múltiples núcleos y citoplasmas utilizando los centroides, y la presencia de orificios.

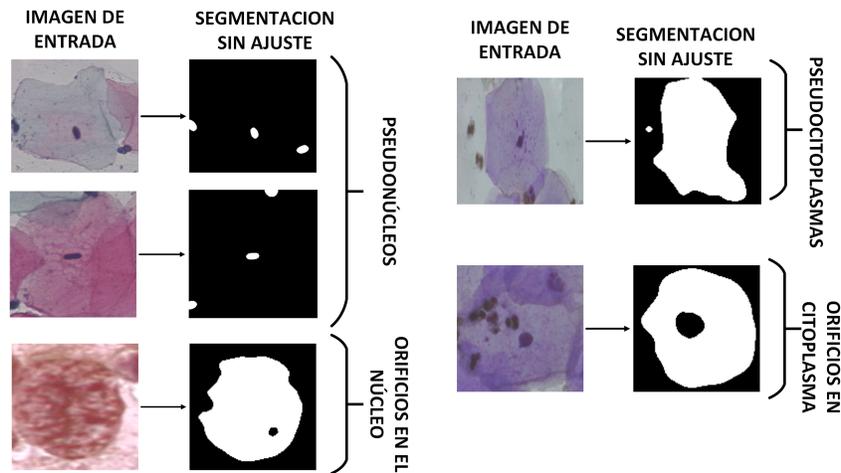
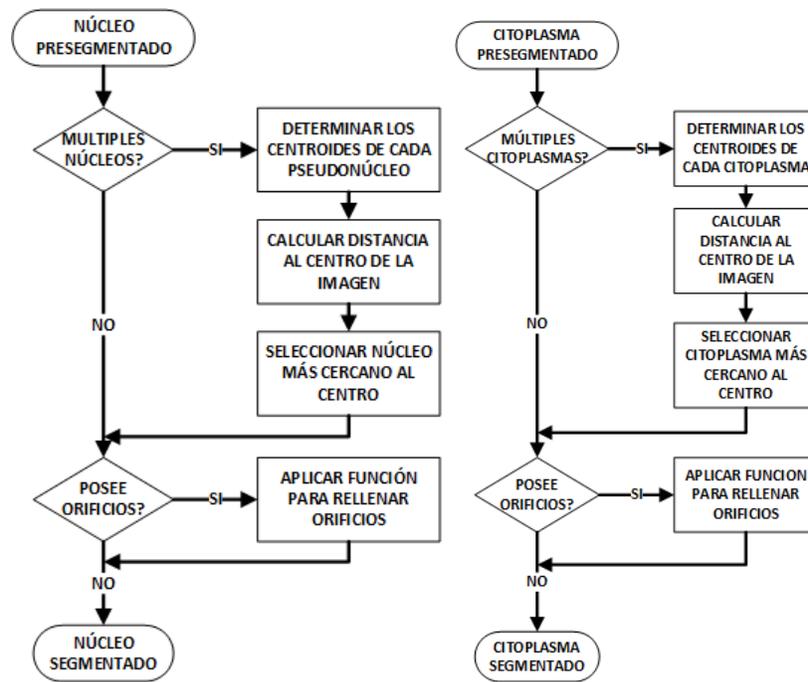


Figura 4.16: Ejemplos de pseudonúcleos/pseudocitoplasmas y orificios de la fase de presegmentación (fuente: autores).



(a) Etapa final segmentación de núcleo. (b) Etapa final segmentación de citoplasma.

Figura 4.17: Segmentación de núcleo y citoplasma (fuente: autores).

Para cada núcleo (Figura 4.17a), se extraen sus coordenadas verticales y horizontales, y se calcula la distancia desde el centro del pseudonúcleo hasta el centro de la imagen. El núcleo con la menor distancia (el que se encuentra más cerca del

centro de la imagen) se selecciona como el núcleo de la célula.

Siguiendo la estructura del diagrama de flujo de la Figura 4.17a, luego de detectar el núcleo correcto, si este presenta orificios dentro de él, se procede a realizar el relleno de regiones. Dicha función de relleno sigue los siguientes pasos:

1. Inversión de color de la imagen original.
2. Etiquetado de las regiones o áreas presentes en la imagen.
3. Encontrar el componente más grande y determinar su etiqueta y su área.
4. Creación de una imagen en blanco con las dimensiones de la imagen original.
5. Rellenar el área del componente más grande en la imagen en blanco. Lo que indica que se pinta todo el interior del agujero.
6. Finalmente, se invierte la imagen resultante de nuevo para que los objetos se conviertan en agujeros llenos, y se devuelve como resultado la imagen con el agujero más grande relleno.

En la Figura 4.17b, se puede observar que la misma estrategia es usada para la segmentación del citoplasma, dado que también puede presentar orificios y/o pseudocitoplasmas, aunque es menos común que en el núcleo.

Utilizando el proceso descrito previamente, se logra obtener la segmentación del objeto de interés. La estrategia implementada permite obtener una segmentación precisa y detallada del objeto de interés, sin dejar múltiples núcleos y/o espacios vacíos en su interior.

Finalmente, en la Figura 4.18, se puede observar que las células ya no presentan ningún tipo de orificio o regiones extra que no correspondan a núcleo y citoplasma, terminando así la fase de segmentación.

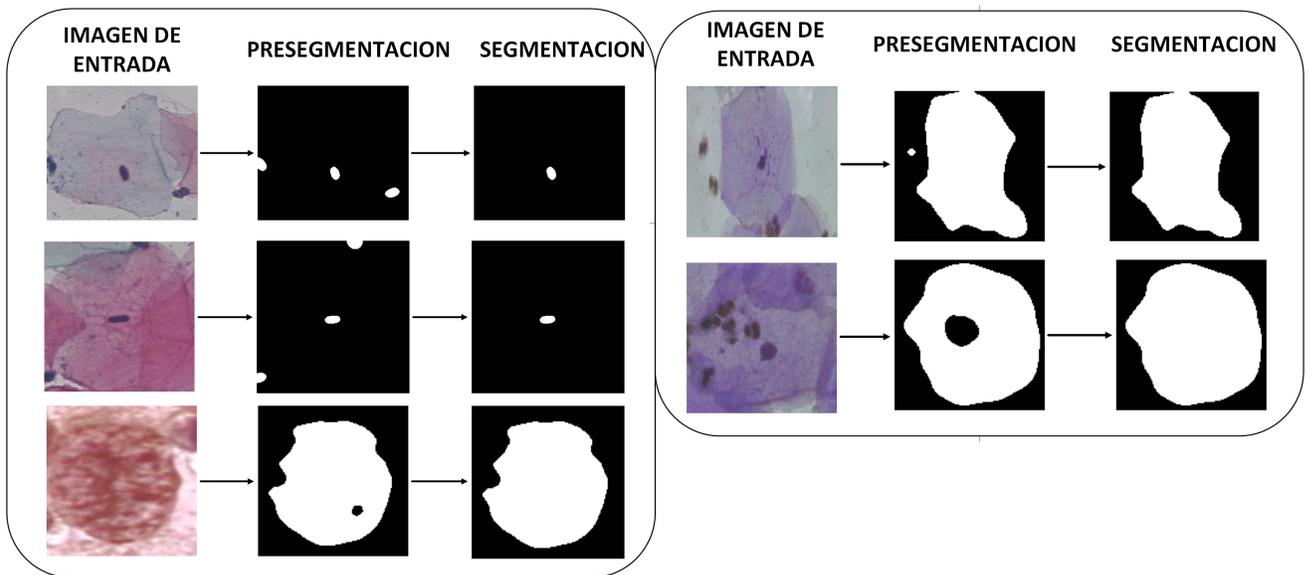


Figura 4.18: Segmentación final (fuente: autores).

#### 4.4 Extracción de características

Una vez obtenidas las regiones de interés mediante la segmentación, es más sencillo medir las características proporcionadas por los componentes de la célula. El objetivo principal de estas mediciones es extraer características numéricas que brinden información clave de las células.

##### 1. Características de forma

A continuación se presentan las características extraídas de las imágenes segmentadas.

- Área y perímetro, en la figura 4.19 se muestra un ejemplo.

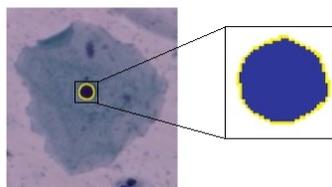


Figura 4.19: Ejemplo área y perímetro en píxeles. Área = 867, perímetro = 110 (fuente: autores).

- Para obtener la característica relacionadas a la elipse inscrita al contorno de la región segmentada (Figura 4.20), se ajusta una elipse al contorno, utilizando la biblioteca *OpenCV*, se obtiene información sobre la posición del centro, los ejes mayor y menor, y el ángulo de rotación.

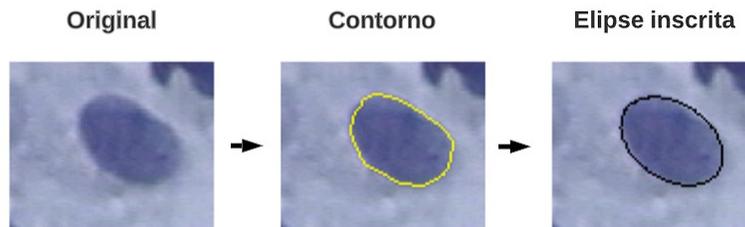


Figura 4.20: Ejemplo elipse (fuente: autores).

- La longitud del eje mayor se determina como la máxima distancia entre dos puntos del contorno en la dirección de la orientación de la elipse ajustada. El eje menor se obtiene como la máxima distancia entre dos puntos del contorno en la dirección perpendicular a la orientación de la elipse ajustada. Esta relación se puede visualizar en la Figura 4.21.

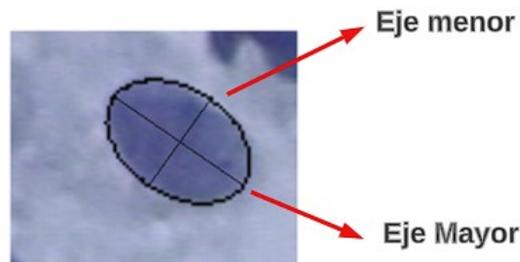


Figura 4.21: Ejes mayor y menor de la elipse (fuente: autores).

- Contando con las medidas de los ejes de la elipse inscrita al contorno de la región segmentada, se puede calcular la excentricidad de la elipse utilizando la relación que existe entre esta medida y la longitud de los ejes.
- La extensión está relacionada con el área del rectángulo que se ajusta al contorno de la región segmentada, como se ilustra en la Figura 4.22.

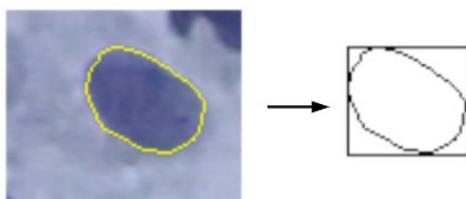


Figura 4.22: Cuadrado que se ajusta al contorno del objeto (fuente: autores).

- La relación de aspecto se calcula al analizar el contorno del objeto y el rectángulo más pequeño que lo rodea, dividiendo la anchura del rectángulo entre su altura. En la Figura 4.23 muestra un ejemplo visual.

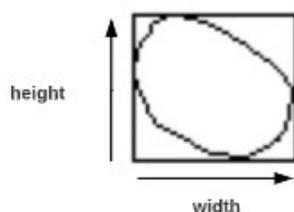


Figura 4.23: Relación de aspecto (fuente: autores).

- Diámetro equivalente. En la Figura 4.24 se observa un ejemplo de un círculo con la misma área que el núcleo:

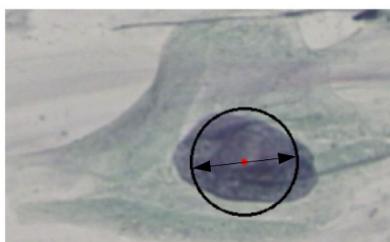


Figura 4.24: Diámetro equivalente (fuente: autores).

- Solidez:

La Figura 4.25 muestra la relación entre el área del contorno del objeto ( $A1$ ) y el área de la envolvente convexa del objeto ( $A2$ ).

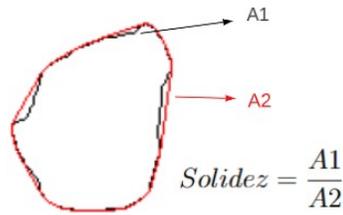


Figura 4.25: Solidez (fuente: autores).

- Diámetros de Feret:

Estos diámetros proporcionan información sobre las longitudes de la estructura celular en diferentes ángulos, lo que es especialmente útil en casos donde el contorno de las células es irregular y no se ajusta adecuadamente a una elipse inscrita o a un rectángulo. Los diámetros de Feret permiten capturar la forma realista de las células al considerar las distancias máximas y mínimas en diferentes orientaciones del objeto. Para ello se utiliza el algoritmo en python expuesto en [62] que permite calcular estas características.

## 2. Características de textura GLCM

La GLCM se configura en las siguientes direcciones para las relaciones de niveles de gris: 0°, 45° y 90°. Además, para capturar la información espacial, se configura una distancia interpíxel de 10, 20 y 30, lo que permite evaluar la variación de la textura a diferentes escalas. Al combinar las medidas de textura en las direcciones y distancias mencionadas, se obtiene una representación completa de la textura contenida en la imagen (Figura 8.1). Las características que se obtuvieron para cada dirección y distancia fueron las siguientes: Correlación, contraste, disimilaridad, homogeneidad, ASM, energía y entropía.

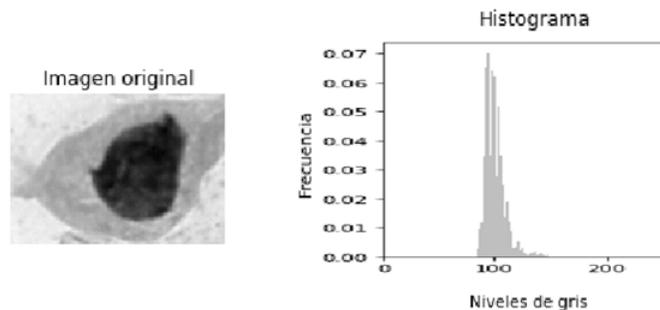


Figura 4.26: Histograma de niveles de gris solo núcleo (fuente: autores).

### 3. Características de color

Para la obtención del brillo de una región segmentada, en primer lugar, se divide la imagen en sus tres canales de color: rojo, verde y azul (RGB), cada uno de los cuales proporciona información sobre la intensidad de la luz en un rango particular de color. Luego, se realiza una combinación ponderada de los planos de color utilizando los coeficientes 0.299, 0.587 y 0.114 para cada canal respectivamente. Esta combinación ponderada proporciona una medida más precisa del brillo de la región de interés teniendo en cuenta los tres canales de color.

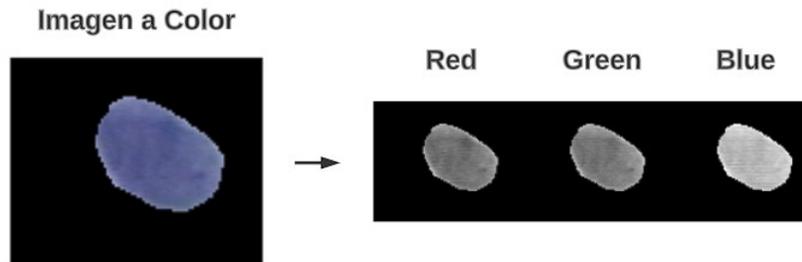


Figura 4.27: Espacios de color región segmentada (fuente: autores).

Con todo lo anterior, al considerar únicamente las características del núcleo, se obtienen 34 variables compuestas por información de forma, textura y color. Mientras que al combinar núcleo y citoplasma, se obtiene un total de 69 características, las cuales se componen de 34 rasgos extraídos del núcleo, 34 del citoplasma, y una característica adicional que relaciona la proporción del área del núcleo con el área del citoplasma.

## 4.5 Selección de características

En la Figura 4.28, se plantean tres métodos diferentes para obtener el espacio de características óptimo utilizando únicamente las características del núcleo para problemas de clasificación de 2 y 3 clases. La primera forma utiliza el algoritmo FEATUREWIZ, que toma todas las características del espacio original y las reduce a las más importantes. La segunda forma parte de las características obtenidas por FEATUREWIZ y las convierte en un nuevo espacio de características llamado PCA. Por último, la tercera forma toma todas las características del espacio original y las convierte directamente al espacio PCA.

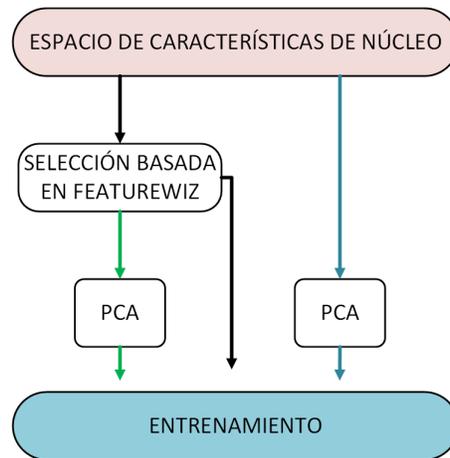


Figura 4.28: Combinación de espacios de características de núcleo para entrenamiento (fuente: autores).

De forma similar, en la Figura 4.29, se proponen dos formas para obtener el mejor espacio de características de núcleo y citoplasma para tareas de clasificación de 2 y 3 clases. La diferencia con respecto a las características solo de núcleo radica en que no existe una tercera forma que obtenga las componentes principales del espacio original, ya que al juntarse núcleo y citoplasma resultaría en demasiadas variables por procesar. Por tanto, el primera forma consiste en seleccionar las características más significativas usando FEATUREWIZ y usarlas para entrenamiento. La segunda forma, toma la características obtenidas por FEATUREWIZ y las convierte al espacio de características PCA el cual será usado para entrenamiento.

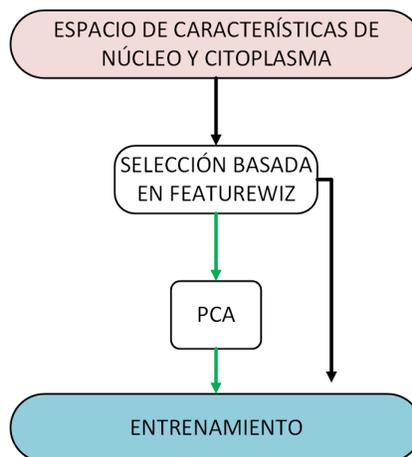


Figura 4.29: Combinación de espacios de características de núcleo y citoplasma para entrenamiento (fuente: autores).

En las Tablas 4.2 y 4.3 se muestran los espacios de características utilizados para la clasificación binaria y de tres clases respectivamente. Cada fila representa un espacio de características específico, identificado por la técnica utilizada. En la columna *Características*, se enumeran las variables seleccionadas para cada espacio.

	Espacio	N° características	Características
Núcleo	FEATUREWIZ	7	CORRELACION3, EXTENSION, RELACION, ENERGIA3, RADIO, SOLIDEZ, RELACIONFERET
	FEATUREWIZ Y PCA	7	PCA1, PCA2, PCA3, PCA4, PCA5, PCA6, PCA7
	PCA	7	PCA1, PCA2, PCA3, PCA4, PCA5, PCA6, PCA7
Núcleo y citoplasma	FEATUREWIZ	13	nRADIO, nENERGIA3, RADIO, BRILLO, DISIMILITUD2, nSOLIDEZ, CORRELACION2, ENTROPIA, nRELACION, nCORRELACION3, DIAMEQUI, ASM2, SOLIDEZ
	FEATUREWIZ Y PCA	13	PCA1, PCA2, PCA3, PCA4, PCA5, PCA6, PCA7, PCA8, PCA9, PCA10, PCA11, PCA12, PCA13

Tabla 4.2: Espacio de características para clasificación binaria (fuente: autores).

	Espacio	N° características	Características
Núcleo	FEATUREWIZ	7	CORRELACION3, EXTENSION, RELACION, ENERGIA3, RADIO, CONTRASTE3, SOLIDEZ, EXCENTRICIDAD
	FEATUREWIZ Y PCA	7	PCA1, PCA2, PCA3, PCA4, PCA5, PCA6, PCA7
	PCA	7	PCA1, PCA2, PCA3, PCA4, PCA5, PCA6, PCA7
Núcleo y citoplasma	FEATUREWIZ	8	NC, nRADIO, nCONTRASTE3, PERIMETRO, BRILLO, nEXTENSION, CONTRASTE3, ASM3
	FEATUREWIZ Y PCA	8	PCA1, PCA2, PCA3, PCA4, PCA5, PCA6, PCA7, PCA8

Tabla 4.3: Espacios de características para clasificación de 3 clases (fuente: autores).

Entre las características se identifican aquellas que comienzan con la letra "n", las cuales pertenecen al núcleo. Por otro lado, aquellas características de textura que finalizan con el número 2, fueron extraídas de una GLCM con orientación de 45° y distancia interpíxel de 20. Características de textura terminadas en 3, parten de una GLCM con orientación de 90° y 30 píxeles de distancia. Del mismo modo, aquellas de textura que no tienen número final, parten de una GLCM de 0° con distancia interpíxel de 10.

## 4.6 Clasificación

Esta sección consiste en la búsqueda de los mejores hiperparámetros que optimizan el desempeño para cuatro clasificadores: DT, SVM, KNN y ANN. La clasificación se divide en dos problemas, el primero, se trata de una clasificación binaria de células normales y anormales. En el segundo problema, se clasifican las células en tres categorías: normal, LSIL y HSIL. En los experimentos de clasificación, se realiza el balance de datos tanto para 2 como para 3 clases y en cada caso se utilizará el 80 % de los datos para entrenamiento y el 20 % restante para probar la capacidad de los esquemas de clasificación en nuevas imágenes.

Para la obtención del mejor hiperparámetro de cada modelo, se realiza la validación cruzada con el 80 % de los datos destinados para entrenamiento, esto con el fin de mitigar el efecto del sesgo y la variabilidad al evaluar un modelo con una única división de datos. Esta técnica consiste en dividir el conjunto de datos de entrenamiento en múltiples subconjuntos, llamados pliegues. En este estudio, se utilizan 5 pliegues, de los cuales, en cada iteración de entrenamiento, se toma un pliegue distinto para prueba mientras que los demás se usan para entrenamiento. Se calcula la exactitud del modelo en cada iteración y, finalmente, se promedian los rendimientos obtenidos para una medida global del desempeño del modelo.

### 4.6.1 K vecino más cercano KNN

En el proceso de evaluación, se han llevado a cabo diferentes pruebas utilizando distintos espacios de características. Estos espacios han sido utilizados tanto para el proceso de entrenamiento como para la evaluación de los datos de prueba. Para la selección de los valores de  $k$  en cada caso, se han considerado las combinaciones de características que mejor optimizan el rendimiento de los clasificadores. En particular, se han seleccionado valores de  $k$  de 30, 31, 40 y 42 para la clasificación de dos clases, mientras que se ha elegido un valor de  $k$  igual a 2 para la clasificación de tres clases. La evaluación cuidadosa de los valores de  $k$  en función de las características de los datos ha permitido obtener los mejores resultados de clasificación para cada caso.

Con el fin de evitar problemas de sobreajuste y de sesgo en la estimación del rendimiento del modelo, se utiliza la técnica de validación cruzada con los datos de entrenamiento para cada valor del hiperparámetro  $k$ .

#### 4.6.2 Máquinas de soporte vectorial SVM

El hiperparámetro  $C$  es crítico en la técnica SVM, ya que controla el compromiso entre maximizar el margen y minimizar el error en la clasificación de los datos de entrenamiento. Un valor de  $C$  grande producirá un margen pequeño y una mayor exactitud en la clasificación de los datos, mientras que un valor de  $C$  pequeño producirá un margen más grande pero puede llevar a una menor exactitud.

Para cada espacio de características se realiza una búsqueda iterativa para encontrar el valor óptimo de  $C$  en el rango de 0.1 a 30, y se selecciona aquel valor de  $C$  que maximice la exactitud. Para evitar el sobreajuste y sesgo en la estimación del rendimiento del modelo, se evalúa el rendimiento del clasificador SVM para los diferentes valores de  $C$  mediante validación cruzada. Los espacios de características mencionados presentan alta dimensionalidad, lo que dificulta trazar fronteras de decisión lineales entre las clases. Para solucionar este problema, se utiliza una técnica conocida como función de kernel. Una de las opciones de función de kernel para SVM es la función polinomial, que utiliza polinomios para trazar fronteras de decisión no lineales y lograr la separabilidad entre las clases.

#### 4.6.3 Árbol de decisión

La profundidad del árbol es un parámetro crítico en el modelo de DT. Un árbol con una profundidad demasiado grande puede generar sobreajuste en los datos de entrenamiento y una mala generalización en imágenes de prueba, resultando en un desempeño deficiente del modelo. Por otro lado, un árbol con una profundidad muy pequeña puede no capturar patrones importantes en los datos, lo que también lleva a un rendimiento deficiente.

En el algoritmo de árboles de decisión (DT), el hiperparámetro  $P$  describe la profundidad máxima del árbol, el cual varía en un rango de 10 a 16. Este parámetro

se determina de forma iterativa, buscando el valor que maximice el *accuracy* en los datos de entrenamiento y de prueba para cada espacio de características. Para evaluar el desempeño del modelo DT con diferentes valores de profundidad, se utiliza la técnica de validación cruzada con 5 pliegues.

#### 4.6.4 Redes neuronales

Las ANN como modelos de clasificación son un poco más complejos que los algoritmos anteriores. En este estudio, se construye una ANN con 3 capas ocultas y una de salida. La capa de salida tiene 1 neurona para el caso de clasificación binaria, y 3 neuronas para el problema de 3 clases. La primer capa oculta consta de 100 neuronas, la segunda presenta 80, y la tercera tiene un valor  $n$  de neuronas para el cual se determinará de forma iterativa en un rango de 5 a 60 neuronas. Dicho valor se seleccionará de tal forma que maximice la exactitud en los datos de entrenamiento y prueba para los distintos espacios de características. De igual forma, se evaluará su rendimiento con validación cruzada, para luego probar el mejor hiperparámetro en el conjunto de prueba.

Para el entrenamiento se usarán 100 épocas, con funciones de activación ReLU que facilita el problema de clasificación no lineal, y *adam* como algoritmo de optimización para el ajuste de pesos sinápticos.

## 5. Pruebas y resultados

En este capítulo se evalúan los métodos de segmentación y clasificación de imágenes PAP tras las diferentes pruebas realizadas. El primer objetivo es seleccionar el mejor método de segmentación de imágenes para las 7 clases presentes en HERLEV, basándose en las métricas de evaluación establecidas en la sección 3.6.

Una vez seleccionado el mejor algoritmo de segmentación, se procede a la extracción de características de las imágenes segmentadas, con el fin de evaluar la clasificación para dos (normal, anormal) y tres clases (normal, LSIL, HSIL) en sus diferentes espacios de características. Se evaluarán 4 clasificadores, eligiendo finalmente los mejores modelos para 2 y 3 clases junto con sus espacios de características.

Los resultados obtenidos en la evaluación de clasificación permitirán determinar el mejor enfoque para la segmentación y clasificación de imágenes.

### 5.1 Evaluación de la segmentación

Aquí se presenta la evaluación de la segmentación de las diferentes propuestas implementadas teniendo en cuenta las siete clases de células presentes en el conjunto de datos. Para ello, se utiliza, como punto de comparación, los datos de la base de HERLEV, la cual cuenta con una imagen de segmentación de referencia para cada célula.

En la Figura 5.1, se pueden apreciar las imágenes de referencia de un grupo de células. Las dos células de la parte superior corresponden a células normales, mientras que la última imagen muestra una célula anormal. En la representación gráfica de las imágenes de referencia en RGB, se observa que la región correspondiente a la célula está coloreada en azul oscuro, mientras que la región del núcleo presenta un tono azul más claro. Cabe aclarar que para efectos de procesamiento en python, se trabajó tanto en un espacio BGR como en un espacio RGB, que al final no afecta el análisis de la imagen de referencia.

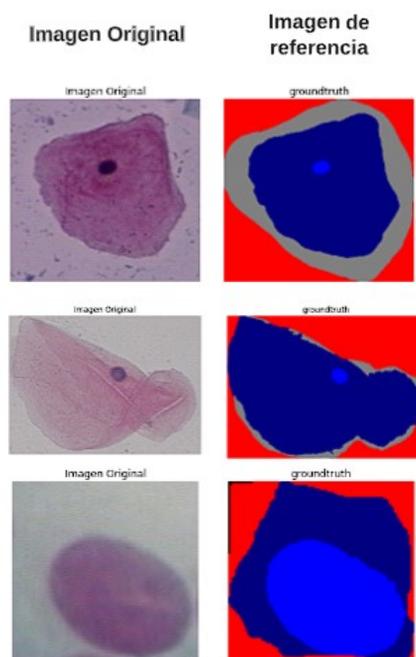


Figura 5.1: Imagen de referencia HERLEV [2].

Para obtener la información del valor de la imagen de referencia y compararla con la máscara obtenida en la segmentación final, se implementa una función que permite calcular las medidas estadísticas de error definidas en la sección 3.6. Este proceso es de gran importancia, ya que permite conocer la precisión y eficacia de los algoritmos de segmentación propuestos.

### 5.1.1 Métricas de evaluación del algoritmo de segmentación

- Evaluación para una sola célula

En la Tabla 5.1 se pueden ver los resultados de las medidas de evaluación para la célula presente en la Figura 5.2, que es una célula anormal categorizada como *light dysplastic* comparada con su referencia.

En la primera parte de la Tabla 5.1 se muestran los resultados de la clasificación de los diferentes píxeles de la imagen segmentada. Se trata de un conteo del total de píxeles que fueron asignados a cada una de las diferentes variables (VP, VN, FP, FN) para dicha célula. En esta tabla se pueden apreciar las medidas calculadas con cada una de estas variables, lo que permite obtener las medidas

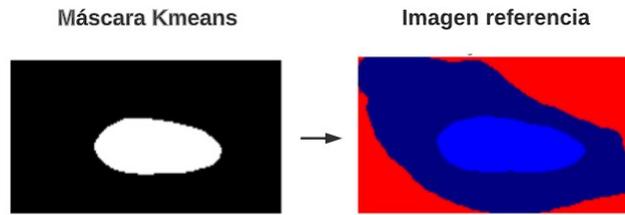


Figura 5.2: Comparación segmentación KMEANS vs imagen de referencia (fuente: autores).

de precisión (P), exactitud (E), exhaustividad (R), índice de DICE y de Jaccard para la segmentación.

En cuanto a la evaluación de la segmentación de la célula, se obtuvo un valor cercano a 0 para la distancia de Jaccard (0.126), lo que indica que la región segmentada del núcleo se asemeja bastante a la región de referencia, teniendo en cuenta que 0 es el valor ideal. Además, el coeficiente de DICE indica que la imagen se asemeja en un 93.28 % a la imagen de referencia. Estos resultados indican que la segmentación de esta célula en particular fue exitosa y precisa, siendo un paso importante para la clasificación.

Medida	Resultados de una sola célula
FN	261
FP	43
VP	2109
VN	13547
P	0.8535
R	0.8509
E	0.8335
Índice de Jaccard	0.8740
Índice de Dice	0.9328

Tabla 5.1: Métricas para una sola célula de tipo anormal (fuente: autores).

- Evaluación para todo el conjunto de células

La Tabla 5.2 proporciona una visión general de los resultados obtenidos en la segmentación de las siete clases de células mediante cada una de las técnicas, evaluando las imágenes segmentadas con las de referencia. Finalmente se muestra el promedio de cada métrica por clase, e igualmente de forma general para todas las células junto con su desviación estándar (*std*).

Se llevó a cabo una propuesta de segmentación de núcleo basada en el algoritmo KMEANS. En promedio, se obtuvo una precisión (P) del 89% y una

exhaustividad (R) del 65.6 % para la región del núcleo. Estos valores indican que el modelo no es muy efectivo en la detección de núcleos, pero cuando los detecta, su confiabilidad es alta, llegando a identificar correctamente el 89 % de ellos. Por medio del coeficiente de DICE, se evaluó la similitud entre las imágenes segmentadas por el modelo y las imágenes de referencia, obteniéndose una semejanza general del 73 %. Estos resultados sugieren que el método de segmentación basado en KMEANS tiene un potencial significativo para la segmentación de imágenes, y puede ser una herramienta valiosa para futuras investigaciones en este campo.

Clase	K-means				Meanshift				U-net							
	Núcleo				Núcleo				Núcleo				Citoplasma			
	P	R	FM	DICE	P	R	FM	DICE	P	R	FM	DICE	P	R	FM	DICE
1	0.850	0.658	0.758	0.706	0.899	0.668	0.749	0.729	0.747	0.866	0.752	0.752	0.921	0.929	0.921	0.921
2	0.945	0.798	0.881	0.856	0.884	0.721	0.874	0.786	0.836	0.917	0.874	0.861	0.933	0.948	0.938	0.938
3	0.850	0.687	0.754	0.731	0.782	0.786	0.775	0.744	0.828	0.951	0.898	0.880	0.895	0.896	0.890	0.890
4	0.901	0.629	0.767	0.712	0.795	0.705	0.839	0.718	0.866	0.963	0.917	0.907	0.922	0.906	0.911	0.911
5	0.913	0.659	0.766	0.740	0.827	0.759	0.826	0.764	0.883	0.969	0.919	0.919	0.914	0.917	0.913	0.913
6	0.902	0.622	0.759	0.716	0.843	0.745	0.807	0.766	0.926	0.954	0.936	0.936	0.928	0.910	0.915	0.915
7	0.859	0.642	0.752	0.712	0.820	0.752	0.806	0.763	0.932	0.952	0.939	0.939	0.943	0.910	0.924	0.924
Promedio	0.890	0.656	0.769	0.730	0.828	0.737	0.812	0.752	0.876	0.947	0.907	0.902	0.923	0.913	0.915	0.915
std	0.246	0.247	0.162	0.232	0.293	0.273	0.163	0.265	0.144	0.105	0.096	0.117	0.069	0.059	0.044	0.044

Tabla 5.2: Evaluación segmentación (fuente: autores).

En el segundo método de segmentación basado en MEANSHIFT, en promedio, se logró una precisión (P) del 82.8 % y una exhaustividad (R) del 73.3 %. En promedio, el modelo logró un 75.2 % de semejanza (índice de DICE) con respecto a las imágenes de referencia superando a la técnica KMEANS con una diferencia de 2.2 % puntos porcentuales. Estos valores sugieren que el modelo es muy selectivo en la detección de la clase de interés, es decir, identifica correctamente gran parte del núcleo. Sin embargo, presenta dificultades para encontrar toda la región correspondiente al núcleo en la imagen.

El método de segmentación semántica UNET, logró una precisión (P) de 87.6 % y una exhaustividad (R) de 94.7 % para segmentación de núcleo. Esto indica que el modelo identificó correctamente el 94.7 % del núcleo presente en las imágenes con una seguridad del 87.6 %. Por otro lado, un índice de DICE de 90.2 % significa que el modelo segmentó los núcleos con una alta similitud con respecto a las imágenes de referencia, lo que sugiere que la segmentación obtenida es muy cercana a la

esperada por un experto.

El método de segmentación semántica UNET aplicado al citoplasma, arrojó valores de precisión (P) y exhaustividad (R) del 92.3% y 91.3% respectivamente. Estos resultados indican que el algoritmo logró identificar correctamente el 92.3% del citoplasma segmentado, es decir, de todo el citoplasma segmentado, el 92.3% correspondía realmente al citoplasma.

Por otro lado, un valor exhaustivo de 91.3% indica que, de todo el citoplasma presente en la imagen real, el algoritmo fue capaz de detectar correctamente el 91.3%. El índice de DICE muestra que, en promedio, los citoplasmas segmentados se asemejan en un 91.5% a la imagen de referencia. Estos resultados son altamente significativos ya que indican que el modelo UNET es altamente preciso y confiable en la segmentación semántica de citoplasma.

### 5.1.2 Tiempos de ejecución de los algoritmos de segmentación

En este apartado, se evalúa el rendimiento de los algoritmos de segmentación utilizando la base de datos de HERLEV con 917 imágenes. Los algoritmos se ejecutan en un equipo con los recursos computacionales mencionados en el capítulo 4. En la Tabla 5.3 se muestran los tiempos globales para la segmentación.

Tiempos de segmentación		
KMEANS	MEANSHIFT	UNET
Núcleo	Núcleo	Núcleo y Citoplasma
21min 23 seg	21 min 38 seg	20 min 57 seg; 15 min 57 seg

Tabla 5.3: Tiempos globales segmentación (fuente: autores)

El tiempo de ejecución promedio para la segmentación de cada imagen fue de 1.39 segundos para KMEANS, 1.41 segundos para MEANSHIFT, y para UNET se tiene 1.37 y 1 segundos para núcleo y citoplasma respectivamente. Cabe destacar que estos tiempos de ejecución pueden variar según el tamaño y la complejidad de las imágenes procesadas. El algoritmo propuesto con UNET logró el mejor tiempo de procesamiento.

### 5.1.3 Selección del algoritmo de segmentación

Comparando los métodos de segmentación de núcleo, se observa que MEANSHIFT y KMEANS obtienen, en promedio, resultados similares, con índices de similitud de 75.2% y 73.0% respectivamente con la imagen de referencia. En contraste, el método de segmentación semántica UNET supera significativamente a estos dos métodos, con una similitud promedio del 90.2% con las imágenes de referencia para el núcleo. Además, la desviación estándar para el método UNET es de 0.117, siendo la más baja en comparación con los otros dos métodos (KMEAN con 0.232, MEANSHIFT con 0.265). Estos resultados sugieren que el método UNET es más consistente y preciso en la segmentación de núcleos en imágenes PAP.

Se observa que el método de segmentación semántica UNET no solo presenta un excelente rendimiento en la segmentación de núcleos, sino también en la segmentación de citoplasma, con una precisión (P) del 92.3% y una exhaustividad (R) del 91.3%, así como un coeficiente de similitud promedio del 91.5% con respecto a la imagen de referencia. Por lo tanto, se puede concluir que UNET es el mejor método de segmentación para ambos componentes celulares, tanto núcleo como citoplasma.

En la Figura 5.3, se muestra de manera visual la distribución de los datos de evaluación con el índice de DICE, para la segmentación de citoplasma utilizando el método UNET.

En el eje horizontal, se presentan los siete tipos de células presentes en la base de datos HERLEV, mientras que en el eje vertical se representan los valores correspondientes de similitud de DICE obtenidos para cada tipo de célula. Se aprecia que en la mayoría de las clases evaluadas presentan una mediana del índice de DICE superior al 90%. Además, se observa que el rango intercuartílico es uniforme y que hay pocos valores atípicos. Estos valores atípicos se encuentran por encima del 60% de similitud, lo que indica que, en el peor de los casos, el método UNET siempre encuentra regiones que realmente pertenecen al citoplasma de las células evaluadas. Estos resultados sugieren que el método UNET es muy efectivo para la segmentación de citoplasma en las imágenes evaluadas.

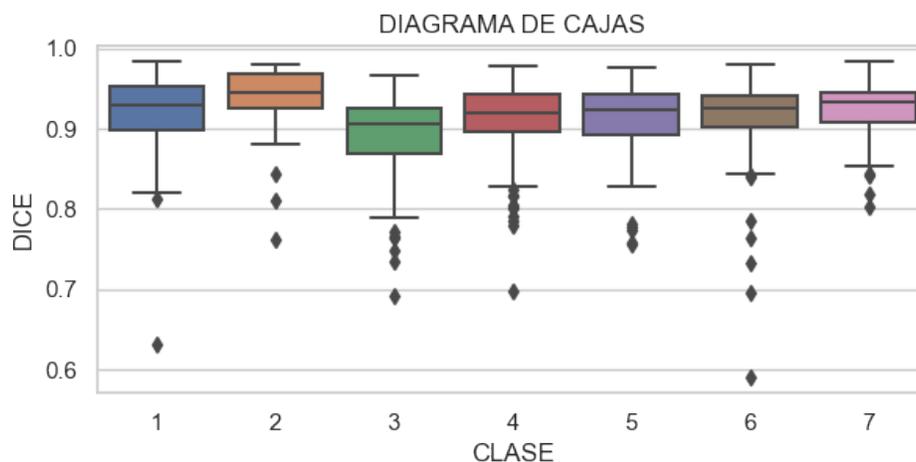


Figura 5.3: Coeficiente de DICE para segmentación semántica de citoplasma (fuente: autores).

En la Figura 5.4, se presenta la distribución de los datos correspondientes a la evaluación de la segmentación de núcleos mediante el método UNET, utilizando el índice de DICE para las siete categorías celulares. En general, se observa que el algoritmo UNET logró una excelente segmentación del núcleo, con una mediana de similitud de DICE superior al 90 % con respecto a la imagen de referencia, a excepción de la primera clase con una similitud mediana del 80 %. Además, se identifican pocas imágenes en las que el algoritmo no logró identificar el núcleo. En relación al rango intercuartílico, se puede observar que en la mayoría de las clases se encuentra bastante compacto, lo que indica que la mayoría de los datos se encuentran concentrados cerca de la mediana, es decir, hay menos variación en los valores de muestra.

La Figura 5.5 presenta la evaluación de la segmentación del núcleo para cada clase, empleando la distancia de Jaccard como medida de similitud. Esta medida varía de 0 a 1, donde 0 es la segmentación perfecta. Se aprecia que el algoritmo UNET tiene un gran desempeño general en la segmentación del núcleo, con una mediana de 0.12. Además, muestra rangos intercuartílicos estrechos para las clases 2,3,4,5,6 y 7, lo que significa que los datos están agrupados cerca de la mediana o segundo cuartil. La única clase que presenta mayor dispersión es la clase 1, correspondiente al tipo *normal Superficial*, que contiene células normales. No obstante, esta clase tiene pocos valores atípicos, indicando que en muy pocos casos, detecta regiones que no corresponden al núcleo.

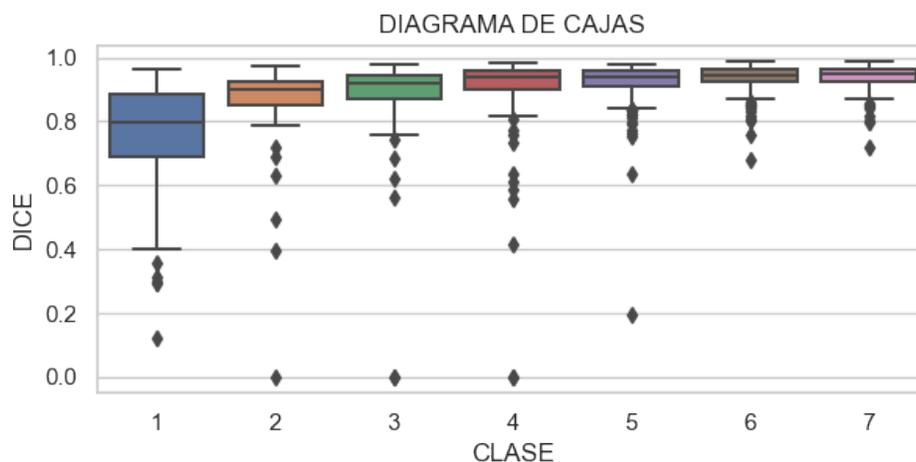


Figura 5.4: Coeficiente DICE para segmentación semántica de núcleo (fuente: autores).

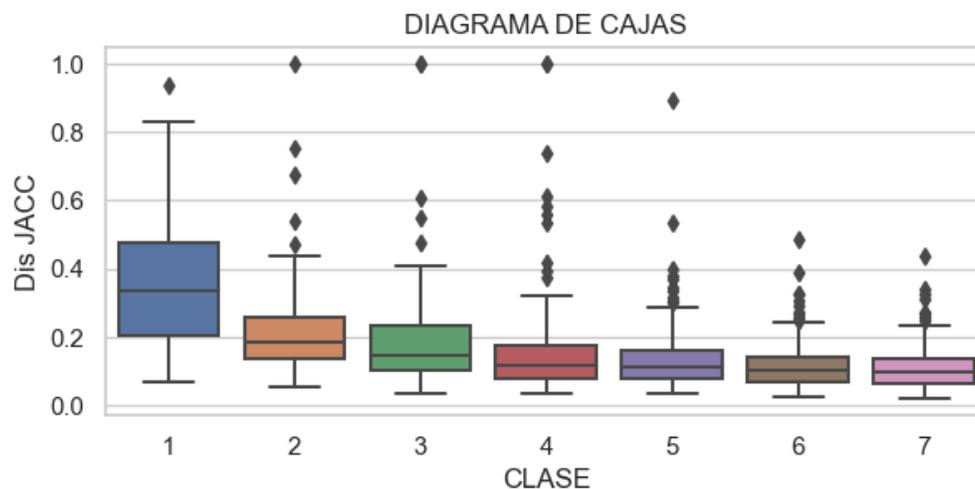


Figura 5.5: Distancia de Jaccard para segmentación semántica de núcleo (fuente: autores).

En la Figura 5.6, se muestra la evaluación de segmentación UNET para el citoplasma según la distancia de Jaccard.

Se puede observar que el algoritmo realiza una segmentación de gran precisión para todas las clases, con una mediana de 0.14 en general. Los valores atípicos se mantienen inferiores a 0.6 indicando que aún en el peor de los casos, el algoritmo detectó una buena proporción del citoplasma presente en la imagen.

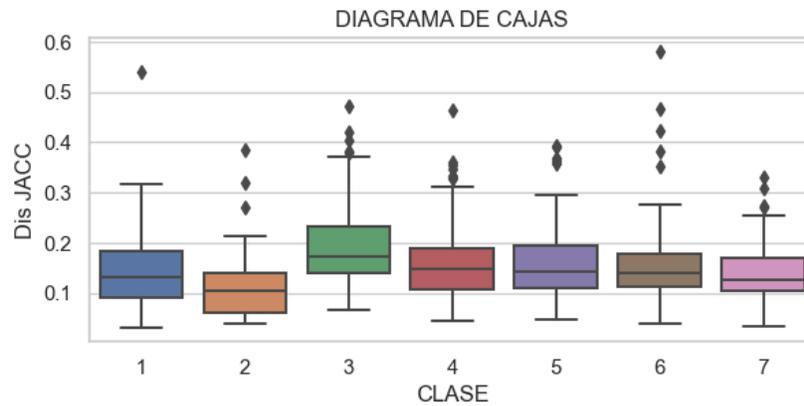


Figura 5.6: Distancia de Jaccard para segmentación semántica de citoplasma (fuente: autores).

Después de evaluar diferentes modelos para la segmentación de núcleo y citoplasma, se puede concluir que el modelo UNET es el mejor en términos de similitud con la imagen de referencia. Con una medida de similitud de DICE del 90.15 % para núcleo y 91.51 % para citoplasma, el modelo UNET logra segmentar de manera precisa ambas estructuras celulares. Por otro lado, utilizando la métrica de coeficiente de Jaccard, se obtiene un 83.54 % para núcleo y 84.63 % para citoplasma. Cabe mencionar que, aunque ambas métricas miden la similitud entre dos conjuntos, el índice de DICE es más optimista porque da más peso a los píxeles correctamente segmentados (verdaderos positivos), lo que lo hace más alto cuando existe una buena intersección entre los conjuntos. Estos resultados muestran que el modelo UNET es una buena opción para la segmentación precisa de núcleo y citoplasma en imágenes de células.

## 5.2 Evaluación clasificación

En esta sección se muestran los resultados obtenidos al aplicar distintos clasificadores para resolver problemas de clasificación binaria (normal, anormal), y de 3 clases (normal, LSIL, HSIL), utilizando los diferentes espacios de características extraídos de los datos a partir de la segmentación UNET. En primera instancia, se evalúa la clasificación de 2 y 3 clases con características del núcleo, y posteriormente la misma tarea de clasificación con características de núcleo/citoplasma. Con estos resultados, se realiza una evaluación comparativa y se seleccionan los clasificadores que ofrecen mejor rendimiento con su correspondiente espacio de características para

las tareas de clasificación. Se realizó el aumento de datos para balancear las clases, manteniendo la relación de 80 % para entrenamiento y 20 % para prueba. Para 2 clases, se tienen un total de 1401 datos, mientras que para 3 clases se tienen 1523.

### 5.2.1 Resultados de clasificación con características del núcleo

#### 5.2.1.1 Clasificación binaria (normal, anormal)

La Tabla 5.4 muestra los resultados del rendimiento de los clasificadores los cuales se evalúan de acuerdo a la precisión (P), la exhaustividad (R), y exactitud (E) para cada espacio de características. Dichas métricas se expresan como el promedio macro, es decir, que primero se obtienen las métricas para cada clase individualmente y luego se calcula la media aritmética de ellas.

En cada clasificador, se detalla el hiperparámetro seleccionado para el modelo y su evaluación con validación cruzada en datos de entrenamiento. También se muestran los resultados de rendimiento en imágenes de prueba, las cuales no han sido vistas por el modelo de clasificación.

Características	Rendimiento	clasificadores							
		DT		SVM		KNN		ANN	
		1	2	1	2	1	2	1	2
7-PCA		P=10		C=21		K=40		n=37	
	P	89,5	-	88	-	97,5	-	93	-
	R	89,5	-	87	-	97	-	93	-
	E	89,67	83,12	86,83	81,69	97,15	95,26	92,88	88,66
FEATUREWIZ		P=13		C=21		K=42		n=57	
	P	94	-	82,5	-	95,5	-	86,5	-
	R	93	-	82	-	95,5	-	86,5	-
	E	93,23	88,03	81,13	78,48	95,37	94	89,67	82,76
FEATUREWIZ y PCA		P=14		C=2		K=31		n=22	
	P	95,5	-	82,5	-	96,5	-	89	-
	R	94,5	-	80	-	96	-	89,5	-
	E	94,66	89,01	79,35	78,75	96,08	94,37	89,32	85

1 = Evaluación con imágenes de Prueba

2 = Validación cruzada en datos de entrenamiento

Tabla 5.4: Rendimiento de clasificación binaria con características de núcleo (fuente: autores).

Los resultados indican que el modelo con el mejor rendimiento para una clasificación binaria es KNN entrenado con los 7 primeros PCA extraídos directamente del espacio de

características originales. Dicho modelo alcanzó una exactitud (E) del 95.26 % evaluado con validación cruzada en 1120 datos de entrenamiento, y un 97.15 % de exactitud (E) en 281 imágenes de prueba no vistas por el modelo, indicando que del total de casos, acertó en 97.15 %. Por otro lado, la exhaustividad (R) indica que de todos los casos que realmente pertenecen a las clases normal (1) y anormal (2), en promedio, tiene una capacidad de detección del 97 %. Por último, la precisión (P), indica que, de todos los casos que el modelo predijo como positivos para las clases, en promedio, el 97.5 % fueron correctamente clasificados.

En la Figura 5.7 se observa la matriz de confusión normalizada para el modelo KNN con el espacio de características 7-PCA. Se puede notar que para la clase normal etiquetada con el número 1, con 146 imágenes, el modelo acertó en el 98 % de los casos y solo falló en el 2.1 % de ellos. Por otro lado, para la clase anormal etiquetada con el número 2, con 135 imágenes, el modelo acertó en el 96 % de los casos, fallando en el 3.7 % de ellos. Estos resultados son indicativos de un modelo altamente preciso y confiable en la tarea de clasificación binaria.

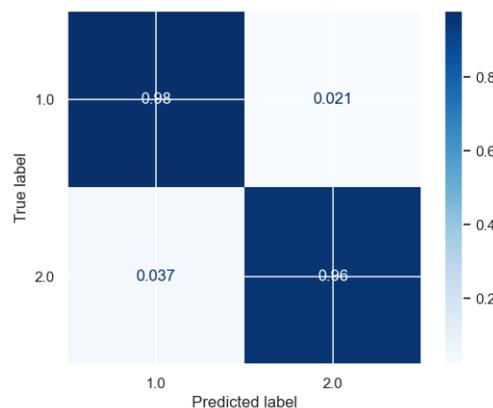


Figura 5.7: Matriz de confusión normalizada para 7-PCA con KNN, clasificación binaria (fuente: autores).

### 5.2.1.2 Clasificación de 3 clases, normal, LSIL, HSIL

En la Tabla 5.5 se presentan las métricas de evaluación macro de los modelos de clasificación en sus diferentes configuraciones para 3 clases conocidas como normal, LSIL, HSIL.

Características	Rendimiento	clasificadores							
		DT		SVM		KNN		ANN	
		1	2	1	2	1	2	1	2
7-PCA	P	P=15		C=26		K=2		n=47	
	R	88	-	79,67	-	90	-	85	-
	E	87	-	75,33	-	88,67	-	83,33	-
FEATUREWIZ	P	87,21	83,25	75,74	71,67	88,85	84,15	83,93	80,37
	R	P=14		C=26		K=32		n=30	
	E	91,33	-	74,67	-	92,67	-	79,33	-
FEATUREWIZ y PCA	P	90	-	70,67	-	90,33	-	78	-
	R	90,16	80,70	71,14	70,44	90,49	83,90	78,36	77,25
	E	P=14		C=16		K=30		n=40	
FEATUREWIZ y PCA	P	86,67	-	79,33	-	91	-	80,67	-
	R	85,67	-	75	-	88,67	-	79,67	-
	E	85,90	83	75,40	71,83	88,85	84,56	80	77,91

1 = Evaluación con imágenes de prueba

2 = Validación cruzada en datos de entrenamiento

Tabla 5.5: Rendimiento de la clasificación de 3 clases con características de núcleo (fuente: autores).

Según los resultados de la Tabla 5.5, el mejor modelo en términos de exactitud (E) para 3 clases es KNN con  $k = 32$  vecinos y el espacio de características FEATUREWIZ, con una exactitud del 90.49% en las imágenes de prueba, con validación cruzada del 83.90% en datos de entrenamiento. Sin embargo, el modelo que guarda la mejor relación entre los datos de prueba y los de entrenamiento, es KNN con  $k = 2$  vecinos, usando el espacio de características 7-PCA extraídas a partir de las características de FEATUREWIZ, logrando así una exactitud promedio de 88.85% en 305 imágenes de prueba y de 84.56% en 1218 imágenes de entrenamiento con validación cruzada.

La Figura 5.8 muestra la matriz de confusión del algoritmo KNN, previamente descrito.

Esta matriz representa de forma porcentual la cantidad de veces que el modelo clasificó correcta o incorrectamente las células, proporcionando una visión general del desempeño del modelo.

Para la clase normal (1), se tienen 97 imágenes, de las cuales el 90% fueron clasificadas de forma correcta con una precisión del 100%. Para la clase anormal LSIL (2), de 109 imágenes el modelo acertó en el 100% con una precisión del 78%. En la tercer clase HSIL (3) se tienen 99 imágenes, de las cuales acertó en el 76% con una

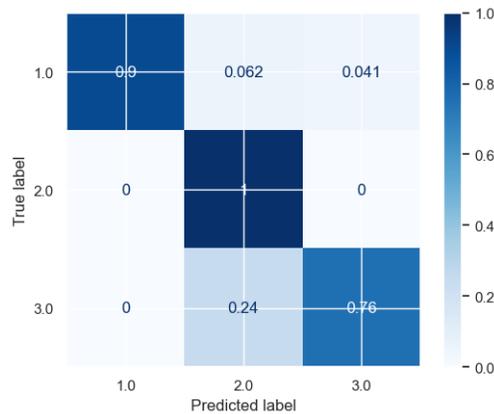


Figura 5.8: Matriz de confusión normalizada para PCA y FEATUREWIZ con KNN, tres clases (fuente: autores).

precisión del 95 % confundiendo el 24 % de ellas con la clase LSIL.

## 5.2.2 Resultados de clasificación con características núcleo/citoplasma

### 5.2.2.1 Clasificación binaria (normal, anormal)

La Tabla 5.6, muestra los resultados para clasificación binaria de los distintos modelos y configuraciones. Las métricas de exactitud (E), precisión (P), y exhaustividad (R) se encuentran expresadas como el promedio macro.

Características	Rendimiento	clasificadores							
		DT		SVM		KNN		ANN	
		1	2	1	2	1	2	1	2
FEATUREWIZ	P	P=13		C=5		K=30		n=43	
	R	94	-	87,5	-	95,5	-	97,5	-
	E	93,59	89,28	86,47	82,23	95,01	92,23	97,5	92,59
FEATUREWIZ y PCA	P	P=14		C=5		K=32		n=54	
	R	94	-	87,5	-	95,5	-	98,5	-
	E	93,23	89,19	86,83	84,19	95,01	91,16	98,57	93,57

1 = Evaluación con imágenes de prueba

2 = Validación cruzada en datos de entrenamiento

Tabla 5.6: Rendimiento para clasificación binaria con características de núcleo y citoplasma (fuente: autores).

Se puede observar que el modelo con el mejor rendimiento es ANN con el espacio

de características PCA y FEATUREWIZ, logrando acertar en un 98.57 % del total de casos de prueba (281 imágenes), con una precisión promedio de 98.5 %.

La Figura 5.9 es una representación visual de la matriz de confusión para el modelo de clasificación basado en ANN con el espacio FEATUREWIZ y PCA.

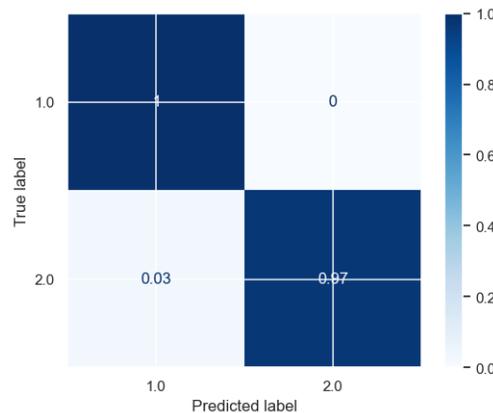


Figura 5.9: Matriz de confusión normalizada para PCA y FEATUREWIZ con KNN, clasificación binaria (N/C) (fuente: autores).

El modelo logró clasificar de forma correcta el 100 % de las imágenes para la clase normal (1) compuesta por 146 imágenes de prueba con una precisión del 97 %. Para la clase anormal (2), del total de casos (135 imágenes de prueba), el modelo detectó de forma correcta el 97 % de ellas con una precisión del 97 %. Se puede concluir que el modelo tiene un desempeño excelente para clasificación de células normales y anormales.

### 5.2.2.2 Clasificación de 3 clases, normal, LSIL, HSIL

En la Tabla 5.7 se muestran los resultados para clasificación de tres clases con los modelos en sus distintas configuraciones. Las métricas de evaluación se expresan como el promedio macro.

En el contexto de la clasificación de 3 clases, se encontró que el modelo con mejor rendimiento en términos de exactitud (E) fue KNN, utilizando el espacio de características FEATUREWIZ obtenidas a partir de todo el espacio de características original.

Características	Rendimiento	clasificadores							
		DT		SVM		KNN		ANN	
		1	2	1	2	1	2	1	2
FEATUREWIZ		P=16		C=15		K=20		n=29	
	P	87,33	-	79,33	-	90,67	-	82	-
	R	87	-	78,33	-	89	-	81,33	-
	E	86,88	82,83	78,68	73,88	89,18	85,21	81,63	80,62
FEATUREWIZ y PCA		P=16		C=13		K=20		n=27	
	P	88	-	77,33	-	90	-	83	-
	R	88	-	76,67	-	88,67	-	82	-
	E	88,19	81,77	77,04	74,62	88,85	85,30	82,29	81,19

1 = Evaluación con imágenes de prueba

2 = Validación cruzada en datos de entrenamiento

Tabla 5.7: Rendimiento de la clasificación 3 clases (N/C) (fuente: autores).

Este modelo logró una tasa de acierto del 89.18 % de un total de 305 casos de prueba, lo que indica que el modelo fue capaz de clasificar correctamente la mayoría de las instancias en el conjunto de datos. Además, el modelo demostró una confianza promedio del 90.67 % al predecir cada clase, lo que sugiere que las clasificaciones obtenidas son altamente confiables. Cabe destacar que, del total de casos reales para cada clase, en promedio, predijo correctamente 89 %.

En la Figura 5.10 se presentan los resultados normalizados de clasificación por clase, los cuales proporcionan información detallada sobre el desempeño del modelo en la clasificación de los datos de prueba:

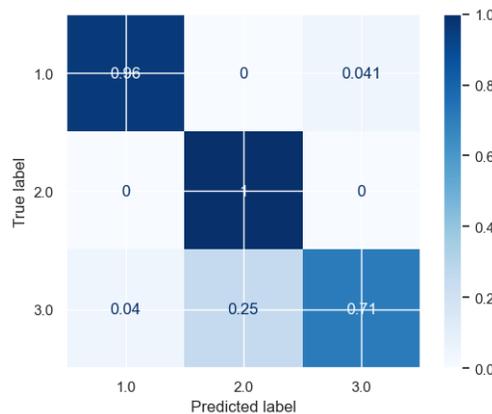


Figura 5.10: Matriz de confusión normalizada para KNN con el espacio FEATUREWIZ, 3 clases (N/C) (fuente: autores).

En la clase normal etiquetada con el número (1), el modelo logró clasificar correctamente el 96 % de las 97 imágenes de prueba, con una precisión del 95.87 %, lo

que indica que es altamente confiable para detectar células normales. En la clase anormal LSIL (2), de un total de 109 imágenes de prueba, el modelo clasificó correctamente el 100 % de ellas con una confianza de detección del 81.34 %. En cuanto a la tercera clase anormal HSIL (3), se dispone de 99 imágenes de prueba, de las cuales acertó en el 71 % con una precisión del 94.59 %. Sin embargo, se observa que el modelo confundió el 25 % de las imágenes de esta clase como células anormales LSIL y un 4 % de ellas como células normales. El modelo muestra alta capacidad de detección de células normales y anormales de la clase LSIL, mientras que en la clase HSIL se observa una mayor tendencia a confundir imágenes de esta clase con las de la clase LSIL y células normales.

### 5.2.3 Selección del modelo de clasificación para dos y tres clases

#### 5.2.3.1 Mejor modelo para clasificación binaria

Usando solo características de núcleo el mejor modelo en términos de exactitud, precisión y exhaustividad es KNN con el espacio PCA, mientras que usando las características de núcleo y citoplasma el mejor modelo es ANN con el espacio de PCA y FEATUREWIZ. En la Tabla 5.8 se muestra una comparación:

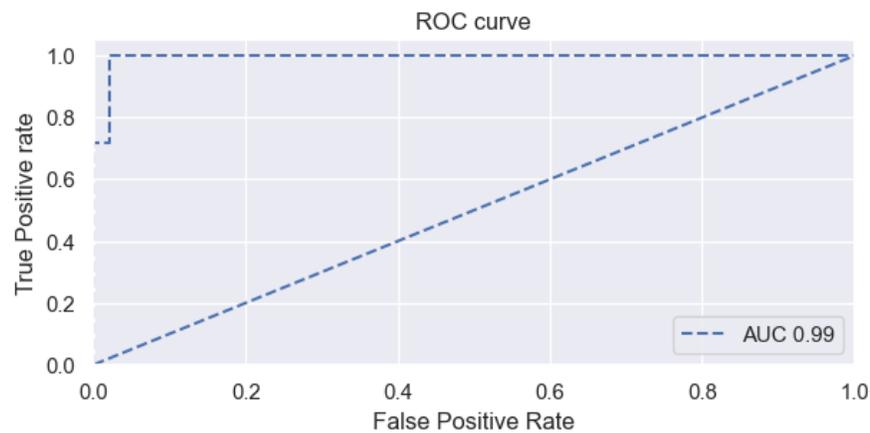
CARACTERÍSTICAS	ESPACIO	CLASIFICADOR	E	P	R	HIPERPARAMETRO
Núcleo	7 PCA	KNN	97,15	97,50	97,00	K=40
Núcleo y citoplasma	FEATUREWIZ Y PCA	ANN	98,57	98,50	98,50	n=54

Tabla 5.8: Comparación de los mejores clasificadores binarios (fuente: autores).

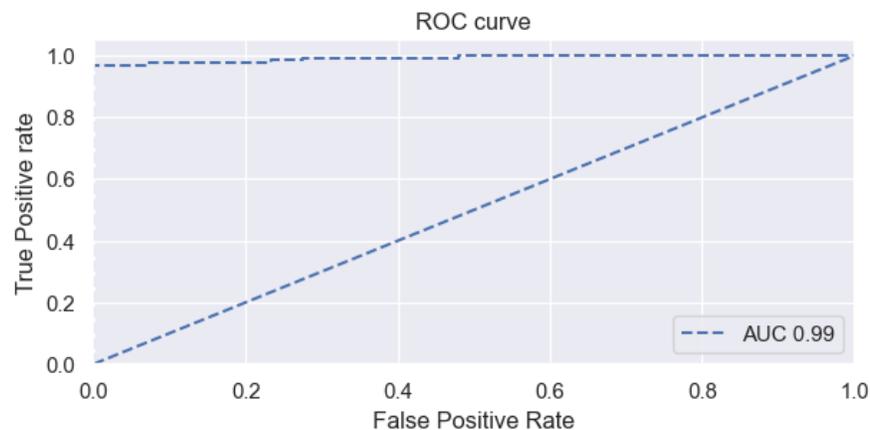
Al analizar los resultados, se puede observar que el modelo ANN con el espacio PCA y FEATUREWIZ, muestra un rendimiento superior en comparación con el modelo KNN. El modelo ANN alcanza una exactitud del 98.57 %, superando al modelo KNN que obtiene un 97.15 %, lo cual representa una mejora de 1.42 puntos porcentuales. Además, el modelo ANN también muestra un mayor índice de precisión y exhaustividad, superando al modelo KNN en 1 y 1.5 puntos respectivamente.

Dado que los resultados están bastante similares, se procede a evaluar con la curva

AUC-ROC, la cual permite determinar la capacidad de discriminación de los modelos entre las dos clases. En la Figura 5.11, se presentan las curvas AUC-ROC para los modelos en cuestión, las cuales se crean trazando la tasa de verdaderos positivos (sensibilidad) en función de la tasa de falsos positivos ( $1 - \text{especificidad}$ ) para diferentes umbrales de probabilidad. La línea diagonal representa un modelo aleatorio, mientras que un modelo perfecto se acercaría a la esquina superior izquierda del gráfico. La área bajo la curva ROC es una medida de la capacidad general de un modelo para distinguir entre clases, la cual varía de 0 a 1, donde 0.5 corresponde a un modelo aleatorio y 1 un modelo perfecto.



(a) Modelo KNN con características del núcleo.



(b) Modelo ANN con características de núcleo y citoplasma.

Figura 5.11: Curva ROC para clasificadores binarios (fuente: autores).

Aunque en ambos casos, el área bajo la curva se encuentra en 0.99, el clasificador ANN (Figura 5.11b) presenta menor tasa de falsos positivos, y mayor tasa de verdaderos

positivos, indicando que se equivoca menos detectando la clase anormal. En contraste, el modelo KNN (Figura 5.11a), presenta menos tasa de verdaderos positivos indicando que se le dificulta un poco más detectar la clase anormal.

Al analizar los resultados obtenidos, se puede concluir que, en la clasificación de células cervicales, la selección de características es un factor crucial para lograr un buen desempeño del modelo. Se encontró que, al utilizar únicamente las características de núcleo para clasificación binaria, el modelo de mejor rendimiento es el KNN con  $k = 40$  vecinos más cercanos, usando el espacio de características 7-PCA, logrando una exactitud de 97.15% en el conjunto de prueba. Sin embargo, al incorporar las características de citoplasma, el modelo más efectivo fue ANN con 54 neuronas en la tercer capa oculta, utilizando el espacio de características PCA y FEATUREWIZ, logrando una exactitud del 98.57% en los datos de prueba.

### 5.2.3.2 Mejor modelo para clasificación de 3 clases

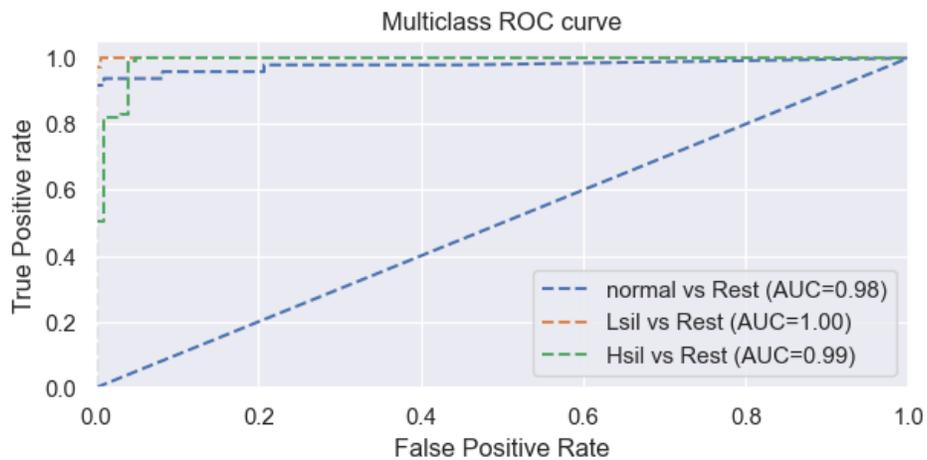
Utilizando solo características de núcleo, el mejor clasificador para la tarea de clasificación de clases (normal, LSIL, HSIL) es KNN con el espacio PCA extraído de FEATUREWIZ. Sin embargo, si se incluyen características de citoplasma, el mejor modelo es KNN con el espacio de FEATUREWIZ. Una comparación detallada de estos modelos con métricas macro se puede encontrar en la Tabla 5.9:

CARACTERÍSTICAS	ESPACIO	CLASIFICADOR	E	P	R	HIPERPARAMETRO
Núcleo	PCA Y FEATUREWIZ	KNN	88.85	91.11	88.67	K=30
Núcleo y citoplasma	FEATUREWIZ	KNN	89.18	90.67	88.86	K=20

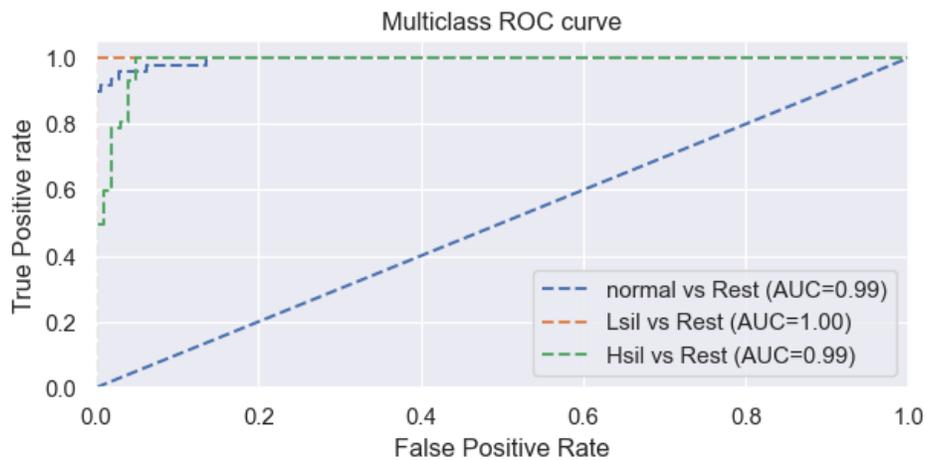
Tabla 5.9: Comparación de los mejores modelos para 3 clases (fuente: autores).

En cuanto a la exactitud de los modelos, se observa que el algoritmo KNN (núcleo) es ligeramente superado por el modelo ANN (núcleo/citoplasma) con una diferencia de 0.33 puntos porcentuales. Sin embargo, en cuanto a la precisión y exhaustividad promedio, ambos modelos están muy parejos. Por tanto, es necesario analizar la curva ROC de la Figura 5.12 para ver qué tan buenos son discriminando las tres clases.

El modelo KNN con características de núcleo de la Figura 5.12a, presenta un AUC de 1 y 0.99 para las clases LSIL y HSIL respectivamente, siendo algo que comparten con el modelo KNN (núcleo/citoplasma) de la Figura 5.12b, indicando así que los dos clasificadores identifican muy bien las clases anormal LSIL y HSIL. Sin embargo, se diferencian en que KNN (núcleo/citoplasma) presenta una mejora ligera pero significativa en la capacidad de detección de células de tipo normal con AUC= 0.99, frente a AUC=0.98 de KNN (núcleo).



(a) Modelo KNN con características de núcleo.



(b) Modelo KNN con características de núcleo y citoplasma.

Figura 5.12: Curva ROC para clasificadores de tres clases (fuente: autores).

Ambos modelos son altamente efectivos para detectar las células anormales LSIL y HSIL, pero el modelo KNN de la Figura 5.12b es mejor para detectar las células normales con características de núcleo/citoplasma (FEATUREWIZ). En promedio,

este modelo con características FEATUREWIZ obtiene una AUC de 0.9933, superando al modelo con espacio PCA y FEATUREWIZ que logra una AUC de 0.9895.

Por tanto, el modelo KNN con  $k = 20$  vecinos más cercanos, y usando características de núcleo/citoplasma obtenidas por FEATUREWIZ puede ser considerado como el mejor modelo para clasificar las células en las categorías de normal, LSIL, HSIL.

#### 5.2.4 Tiempos de ejecución de la clasificación

En este apartado, se muestra el desempeño de los algoritmos de clasificación seleccionados. La implementación del algoritmo se ejecutó en un entorno computacional mencionado en el capítulo 4.

En la tabla 5.10 se muestran los tiempos de clasificación del total de imágenes para los mejores modelos de dos y tres clases.

Tiempos de clasificación					
Clasificación	Espacio de características	Modelo	Hiperparámetro	Tiempo	Total de imágenes de prueba
CLASIFICACIÓN BINARIA	FEATUREWIZ Y PCA	ANN	$n=54$	2,6 seg	281
TRES CLASES	FEATUREWIZ	KNN	$k=20$	0,7 seg	305

Tabla 5.10: Tiempos clasificación (fuente: autores)

El tiempo de ejecución con el clasificador binario ANN fue de 2.6 segundos para 281 imágenes de prueba, logrando una tasa de 9.2 milisegundos por imagen. Del mismo modo, el clasificador KNN para tres clases logró categorizar 305 imágenes de prueba en 0.7 segundos para una tasa de 2.29 milisegundos por imagen.

## 6. Discusión

En el siguiente capítulo se realiza un análisis comparativo de los datos presentados con enfoques y métodos existentes, considerando los resultados de segmentación y clasificación en cada trabajo.

### 6.1 Métodos propuestos

En el presente estudio, se realizaron diferentes análisis utilizando técnicas de segmentación, incluyendo el algoritmo de *Watershed* y DBSCAN. Sin embargo, estos algoritmos no fueron seleccionados como las técnicas principales en comparación con las que se presentaron previamente, ya que estas últimas ofrecieron resultados más significativos para los objetivos del trabajo. El algoritmo de *Watershed*, aunque logra segmentar adecuadamente la región del citoplasma para las primeras clases del conjunto de datos, no arroja buenos resultados para las clases anormales. Por otro lado, el algoritmo DBSCAN demostró ser eficiente en la detección de la región del núcleo en la mayoría de los datos de HERLEV. Sin embargo, su alto costo computacional y la necesidad de ajustar parámetros para la región del citoplasma hicieron que no fuera viable utilizarlo en esta propuesta.

En el ámbito de la investigación, la inteligencia artificial (IA) ha desempeñado un papel fundamental en la resolución de problemas de segmentación de imágenes. En particular, el algoritmo SAM ha emergido como una solución novedosa capaz de segmentar cualquier objeto o elemento característico en una imagen, incluso aquellos objetos que nunca antes había sido visto por el algoritmo [63]. Se trata de una IA que logra la segmentación de cualquier tipo de elemento individual presente en una imagen a partir de la identificación de los píxeles pertenecientes a determinada categoría, y que ha sido entrenada con el conjunto de datos SA-1B [63] que provee más de mil millones de máscaras de segmentación de referencia de mundo abierto que singularizan un objeto por medio de una región delimitadora. En ese sentido, se plantea la realización de pruebas de segmentación de imágenes utilizando la base de datos de HERLEV, y comparar los resultados obtenidos por nuestro modelo especialista en imágenes PAP

contra los de SAM.

En la Figura 6.1 se puede observar una pequeña prueba de segmentación celular con la base de datos HERLEV:

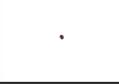
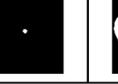
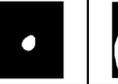
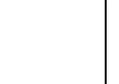
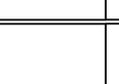
		SAM		UNET		
		ENTRADA	Núcleo	Citoplasma	Núcleo	Citoplasma
NORMAL						
						
						
ANORMAL						
						
						

Figura 6.1: Comparación segmentación SAM vs UNET (fuente: autores).

Como se puede apreciar, el método SAM ha demostrado gran capacidad de segmentación de los núcleos en las clases normales y anormales. No obstante, su desempeño en la detección del citoplasma, especialmente en las clases anormales, no es tan robusto. Este aspecto es precisamente uno de los puntos más fuertes de nuestra propuesta utilizando UNET. Con nuestra propuesta, logramos una detección precisa del citoplasma celular en un amplio rango de clases, tanto normales como anormales. Esta mejora representa una ventaja significativa en comparación con el método SAM y contribuye a una evaluación más completa y precisa de las imágenes de células.

## 6.2 Comparación de rendimiento de la segmentación

En este apartado se realizará una comparación entre los resultados de la técnica de segmentación de imágenes propuesta y otros enfoques reportados en la literatura. Se evaluará y validará la efectividad y rendimiento de la propuesta basada en la red UNET,

que ha mostrado obtener los mejores resultados en la segmentación de las regiones objetivo en comparación con otros enfoques utilizados en este trabajo, como KMEANS y MEANSHIFT. Mediante esta comparación, se busca confirmar la eficacia de la técnica propuesta y su capacidad para mejorar los resultados obtenidos en otros enfoques, lo cual contribuirá al avance en el campo de la segmentación de imágenes médicas.

En la Tabla 6.1 se muestra una comparación de los resultados de segmentación de los trabajos existentes y nuestra propuesta. La precisión ( $P$ ) y la exhaustividad ( $R$ ) son métricas consideradas para evaluar los modelos de segmentación, igualmente se considera el índice de DICE para evaluar la similitud de las regiones segmentadas sea el caso de cada propuesta. El subíndice  $c$  y  $n$  corresponden a las métricas pertenecientes al citoplasma y núcleo respectivamente.

Trabajos Existentes	Técnica	Dataset	Segmentación Citoplasma			Segmentación Núcleo		
			P <sub>c</sub>	R <sub>c</sub>	DICE <sub>c</sub>	P <sub>n</sub>	R <sub>n</sub>	DICE <sub>n</sub>
A. M. Braga [34]	HMLS	ISBF'14	-	-	-	0,93	0,93	0,92 ± 0,04
		EDF	-	-	-	0,95	0,90	0,92 ± 0,05
		Herlev	-	-	-	0,91	0,93	0,90 ± 0,12
P. Wang [13]	Meanshift	Privado	-	-	-	-	0,94	-
T. Wang [32]	FCM y dominio de árbol	ISBF'15	-	-	-	0,95	0,91	0,93 ± 0,04
H. Lee. [23]	Umbralización	ISBF'14	-	-	0,897 ± 0,075	-	-	-
		ISBF'15	-	-	0,879 ± 0,087	-	-	-
L. Zhao. [26]	MRF	Privado	-	-	0,71 ± 0,08	-	-	0,72 ± 0,09
		Herlev	-	-	0,82 ± 0,11	-	-	0,93 ± 0,03
A. Tareef. [27]	superpíxeles-SVM	ISBF'14	-	-	0,89 ± 0,07	0,99	0,94	0,93 ± 0,04
W. Willian [12]	TWS	Herlev	-	-	0,948	-	-	0,973
Diniz. [25]	SLIC-DBSCAN	ISBF'14	-	-	-	0,98	0,99	-
Adhikary. [11]	Otsu valle modificado	DIC	0,91	0,84	-	0,81	0,84	-
		Herlev	0,90	0,83	-	0,86	0,91	-
<b>Nuestra Propuesta</b>	Unet	Herlev	<b>0,92</b>	<b>0,91</b>	<b>0,915 ± 0,044</b>	<b>0,88</b>	<b>0,95</b>	<b>0,902 ± 0,117</b>

Tabla 6.1: Comparación de resultados de segmentación (fuente: autores).

En estos estudios, se utilizan diversas bases de datos, tanto públicas como privadas, de las cuales han surgido distintos enfoques y propuestas. La base de datos pública de HERLEV es ampliamente utilizada en estos estudios. Además, en algunas ocasiones,

se implementa la base de datos de ISBI'14 y ISBI'15 (Overlapping Cervical Cytology Image) en combinación con la base de datos de HERLEV o de manera individual ([23] [25], [27], [32], [34]). Estas bases de datos constan de un conjunto de imágenes generadas artificialmente a partir de células reales.

En [25] se presenta una detección rápida de núcleos, obteniendo valores altos para la precisión y la exhaustividad de (0,98) y (0,99) respectivamente en comparación con nuestro método, con bajo costo computacional. En el método propuesto en [23] se obtienen valores inferiores del índice de DICE indicando que la similitud entre la región del citoplasma, en comparación con nuestro método es poco precisa, pero destaca en que aborda el problema de solapamiento entre múltiples células mediante la técnica de detección de bordes. En nuestro caso, la red UNET logra interpretar regiones del citoplasma que presentan superposición y/o que no son visibles a simple vista logrando concordar con las imágenes de referencia para las diferentes clases presentes en HERLEV.

En [11], [23], [26] [27], y [12] se obtienen regiones segmentadas del citoplasma, siendo [12] la de mayor índice de DICE en núcleo y citoplasma en comparación con nuestra propuesta.

Por tanto, la propuesta de la red UNET para la segmentación de las regiones de interés ha demostrado obtener valores altos que son competitivos y se encuentran a la par con los trabajos existentes en el campo. Esto brinda una oportunidad para obtener características representativas de cada clase a partir de los objetos segmentados, lo que a su vez contribuye a una mejor clasificación de los datos. Es importante destacar que la propuesta de utilizar UNET para la segmentación de imágenes de frotis cervical no es comúnmente implementada, lo que resalta la originalidad y novedad de este enfoque.

### 6.3 Comparación de rendimiento del clasificador

En la Tabla 6.2 se presenta una comparación entre los resultados de clasificación obtenidos en trabajos encontrados en la literatura y nuestra propuesta, lo cual permite evaluar el desempeño de cada enfoque en la clasificación de células de frotis cervical.

Trabajos Existentes	Técnica	Dataset	Clases	Rendimiento			
				exactitud	P	R	AUC
J. Martínez-Más [15]	CaffeNet CNN	Privado	2	0,88	-	0,92	-
A. M. Braga [34]	LSSVM	Herlev	2	0,89	0,91	0,89	0,93
			3	0,77	0,85	0,85	-
P. Wang [13]	SVM	Privado	2	0,93	-	0,95	-
			2	0,97	-	0,96	-
			3	0,96	-	0,92	-
A. D. Jia [31]	CNN-SVM	Herlev	2	0,99	-	98,9	-
			7	0,93	-	0,93	-
		Privado	2	0,94	-	0,93	-
L. Zhang [30]	CNN	Herlev, HEMLBC	7	0,98	-	0,98	0,99
W. Willian [12]	FCM	Herlev	2	0,98	0,99	0,99	0,99
		Dataset 2	2	0,97	0,97	0,98	-
		Dataset 3	2	0,95	0,90	1	-
Adhikary. [11]	SVM	DIC	2	0,97	-	-	0,99
			3	0,92	-	-	0,92
		Herlev	2	0,94	-	-	0,97
			3	0,89	-	-	0,91
<b>Nuestra Propuesta</b>	ANN	Herlev	2	<b>0,98</b>	<b>0,98</b>	<b>0,98</b>	<b>0,99</b>
	KNN		3	<b>0,89</b>	<b>0,91</b>	<b>0,89</b>	<b>0,99</b>

Tabla 6.2: Comparación de los resultados de clasificación con las diferentes propuestas para PAP (fuente: autores).

En nuestra propuesta, se obtuvieron altos valores de precisión; para la clasificación en dos categorías, se alcanzó una exactitud del 98 %, mientras que para la clasificación en tres clases obtuvimos una exactitud del 89 % como ya se mencionó anteriormente. Estos resultados demuestran una notable capacidad de clasificación de células PAP en sus grados patológicos, lo que respalda su efectividad. Además, al comparar estos resultados con los trabajos existentes, se observa que son equivalentes e incluso se supera en algunos casos la clasificación binaria para la base de datos de HERLEV. Esto destaca la eficacia de nuestra propuesta en la clasificación de células de frotis cervical y la relevancia de los resultados obtenidos.

En el trabajo de [30], se presenta un sistema de clasificación celular utilizando CNN,

logrando una exactitud del 98 % en relación al total de los datos. La principal diferencia con nuestra propuesta radica en que en dicho estudio no se realiza la segmentación celular, sino que se determina directamente el grado patológico a partir de una imagen de entrada, obteniendo buenos resultados en la clasificación.

En contraste, en nuestra propuesta utilizamos la red UNET, que también es una CNN, pero se emplea principalmente para la segmentación celular. En lugar de realizar la clasificación directa del grado patológico, clasificamos a nivel de píxel para la segmentación y luego se determina el grado patológico mediante un modelo de clasificación. Aunque los enfoques difieren, los resultados obtenidos son comparables.

Por otro lado, en el trabajo presentado en [33], se introduce un modelo de conjunto de corte de gráfico normalizado para obtener los contornos del núcleo y el citoplasma, logrando una segmentación celular precisa. Este enfoque obtuvo resultados bastante altos, con una exactitud del 99 %, superando los resultados de nuestra propuesta en términos de precisión.

## 7. Conclusiones y trabajos futuros

### 7.1 Conclusiones

Durante este estudio, se presentaron tres enfoques alternativos para la segmentación del núcleo y el citoplasma celular, utilizando tanto métodos de aprendizaje no supervisado para la detección de núcleos como métodos supervisados para el núcleo y el citoplasma. Una de las desventajas de los métodos de agrupamiento sin supervisión fue la falta de etiquetas o anotaciones previas de las regiones de interés, lo cual afectó la calidad y precisión de la segmentación resultante. Además, la selección adecuada de los parámetros del algoritmo fue crucial y requería ajustes y pruebas repetidas para obtener una segmentación óptima, lo cual resultó ser un proceso iterativo y laborioso. Asimismo, la segmentación no supervisada enfrentó dificultades para separar objetos complejos y superpuestos en las imágenes, lo que condujo a fusiones erróneas de regiones.

En contraste, UNET logró segmentar las imágenes, incluyendo aquellas que presentaban superposición con otras células, logrando un índice de similitud DICE del 90.15 % para el núcleo y de 91.51 % para el citoplasma con respecto a las imágenes de referencia. Los métodos de KMEANS y MEANSHIFT lograron un índice de similitud DICE de 73 % y 75 % respectivamente para segmentación de núcleo. Estos resultados resaltan la eficacia de UNET como una herramienta poderosa para la segmentación de células PAP. Este estudio representa una investigación pionera en el uso de la arquitectura UNET para la segmentación de células PAP, y los resultados obtenidos respaldan su potencial para futuros avances en esta área.

La combinación de técnicas de selección de características y algoritmos de clasificación supervisada demostraron ser efectivas para identificar y clasificar con precisión las células PAP en categorías patológicas, tanto en 2 clases (normal, anormal) como en 3 clases (normal, LSIL, HSIL) con una exactitud de clasificación de 98.57 % y 89.18 % respectivamente con características de núcleo y citoplasma. El análisis de componentes principales (PCA) y la herramienta FEATUREWIZ, así como su combinación, permitieron establecer un conjunto de características cruciales para la

clasificación precisa del grado patológico de células PAP en los formatos de 2 y 3 clases.

En este estudio, se han definido un conjunto de medidas para evaluar tanto el proceso de segmentación como el de clasificación, con el objetivo de determinar la técnica o algoritmo que proporciona el rendimiento más preciso en cada caso. En particular, el índice de DICE ha sido fundamental en la evaluación y selección de la técnica de segmentación más adecuada. Este índice ha permitido medir la similitud entre la segmentación obtenida y la segmentación de referencia, ofreciendo una medida cuantitativa de la calidad de la segmentación. Aunque el índice de Jaccard también mide la similitud entre dos imágenes, se tomó el índice DICE, dado que otorga mayor peso a los verdaderos positivos de la imagen segmentada, es decir, a los píxeles del objeto de interés correctamente segmentados. Los resultados obtenidos a través de la métrica DICE han demostrado la calidad y la comparabilidad de los resultados de clasificación logrados a partir de la segmentación seleccionada UNET. Esto indica que se ha logrado una segmentación precisa, lo cual es fundamental para obtener resultados de clasificación de alta calidad y estar a la par con el estado del arte.

En los trabajos consultados en el estado del arte se han encontrado algunas propuestas que realizan la segmentación de la región del citoplasma haciendo uso de diferentes conjuntos de datos y técnicas variadas para lograr la segmentación de esta región objetivo. Sin embargo, nuestra propuesta demostró una mayor capacidad para segmentar dicha región, ya que obtuvo un índice de similitud DICE del 91,5 %, como se puede apreciar en la tabla 6.1 de comparación de la segmentación con otros estudios. Además, se logró mantener una precisión y exhaustividad significativamente altas del 92 % y 91 % respectivamente, al utilizar la estructura CNN de UNET, que generalmente no se ha aplicado en la etapa de segmentación de este tipo de pruebas, es decir, mayoritariamente se usan CNN para clasificación pero no para segmentación. Por otro lado, en la clasificación del tipo de célula, en la mayoría de estudios, las SVM son ampliamente utilizadas en este tipo de problemas de clasificación. Sin embargo, en las pruebas realizadas en nuestra propuesta con el modelo KNN resultó ser sencillo de configurar y el que presentaba valores de exactitud sobresalientes en cada una de ellas.

En las diferentes combinaciones de espacio de características evaluadas, el clasificador KNN demostró consistentemente buenos resultados, destacándose como el

más estable en la mayoría de las pruebas realizadas. En el caso de la clasificación binaria, KNN obtuvo una exactitud del 97.15% usando características de núcleo, mostrando un rendimiento satisfactorio. No obstante, fue superado por el clasificador ANN, que alcanzó una exactitud del 98.57% al utilizar características de núcleo y citoplasma seleccionadas mediante PCA y FEATUREWIZ. Estos resultados resaltan la efectividad de estas técnicas en la clasificación de las células PAP en dos clases. En el contexto de problemas de tres clases, el clasificador KNN demostró un rendimiento sobresaliente, logrando una exactitud del 89.18% al utilizar características de núcleo y citoplasma seleccionadas por FEATUREWIZ. Este resultado destaca la capacidad de KNN para manejar clasificaciones multiclase de manera efectiva.

La inclusión de características del citoplasma en la clasificación de células de frotis cervical resulta crucial para obtener una alta precisión en comparación con el uso exclusivo de características nucleares. Sin embargo, también se observó que solo utilizando características del núcleo se lograron resultados de clasificación satisfactorios. A pesar de esto, es importante considerar tanto las características nucleares como las del citoplasma para obtener una clasificación más precisa y detallada de las células. La combinación de ambas características contribuye a una caracterización más completa y a una mejor comprensión del grado patológico de las células.

## 7.2 Trabajos futuros

Se sugiere mejorar el algoritmo de segmentación para optimizar su rendimiento computacional. Aunque no se consideró como un aspecto crítico en este trabajo, se reconoce que una mayor eficiencia en los cálculos podría resultar en beneficios significativos en términos de tiempo y recursos. Mediante técnicas de optimización y exploración de enfoques novedosos, se busca aumentar la velocidad y eficacia del algoritmo de segmentación, permitiendo su aplicación en conjuntos de datos más extensos. Además, se busca mejorar y fortalecer la propuesta de este estudio evaluando la robustez del algoritmo con diversas bases de datos. Entre ellas, se incluyen los conjuntos de datos de CRIC y SIPaKMeD, que contienen imágenes de células capturadas por diferentes tipos de microscopios. Estas bases de datos presentan desafíos adicionales, como variaciones en la iluminación, imágenes ruidosas

y una amplia diversidad de tamaños y formas celulares. La evaluación del rendimiento del algoritmo en estos escenarios desafiantes permitirá afinar aún más los métodos y asegurar su eficacia y precisión en una variedad de situaciones clínicas. La Figura 7.1 muestra algunas pruebas de segmentación realizadas en imágenes de las bases de datos CRIC y SIPaKMeD, las cuales no fueron utilizadas durante el entrenamiento del algoritmo.

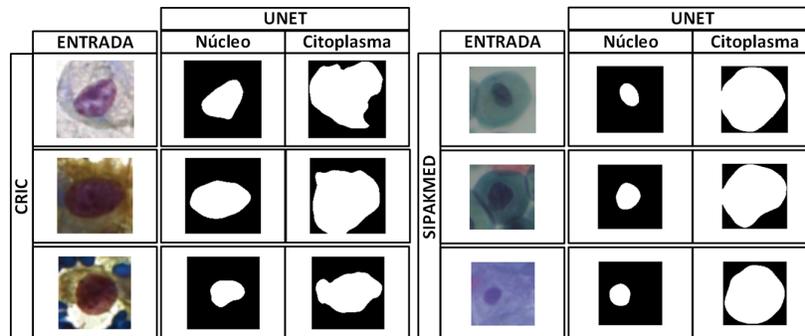


Figura 7.1: Pruebas en imágenes de CRIC y SIPaKMeD (fuente: autores).

Estas pruebas preliminares brindan un vistazo inicial de cómo se desempeña el algoritmo en estas nuevas bases de datos, lo que motiva a realizar un análisis más exhaustivo en futuras investigaciones. Estos trabajos futuros llevarán a un paso más cerca del objetivo de ofrecer una herramienta completa y efectiva para la segmentación y clasificación de células, brindando resultados precisos y confiables en diversas situaciones clínicas.

## 8. Anexos

### 8.1 Anexo 1: Detalles del algoritmo de segmentación y clasificación

Para llevar a cabo el desarrollo y funcionamiento del algoritmo, se deben instalar y descargar los complementos necesarios. En esta sección se describe el algoritmo y se presentan las diferentes herramientas software y paquetes utilizados con sus respectivas versiones.

- Python

Es un lenguaje de programación interpretado, de alto nivel multiparadigma. Creado por Guido van Rossum y lanzado por primera vez en 1991. Caracterizado por su sintaxis clara y legible, junto con su amplia variedad de funciones y módulos, y una gran biblioteca que amplía las capacidades del lenguaje.

Versión usada: python 3.10.3

- Jupyter

Plataforma de código abierto que permite crear y compartir documentos interactivos (*notebook*) que contienen código, visualización, texto y otros elementos. Facilita la programación en lenguajes como Python, R y Julia, aunque admite una gran variedad de lenguajes de programación.

#### 8.1.1 Descripción del algoritmo

El proyecto completo se encuentra en el siguiente enlace de GitHub (enlace: [github/segmentacion-pap](#)). El algoritmo de segmentación se divide en las tres soluciones propuestas KMEANS, MEANSHIFT y UNET.

A continuación se describe brevemente las etapas de los algoritmos KMEANS Y MEANSHIFT:

- Se cargan la imagen de la célula. Se convierte la imagen a RGB, se aplica una transformación a escala de grises

```

1  # RUTA DE LA IMAGEN
2  dirCarpeta = 'DatasetNormalizado/4_light_dysplastic/
3              anormalMild50.jpg'
4
5  # BGR
6  imagen = cv2.imread(dirCarpeta)
7  #RGB
8  imagen = cv2.cvtColor(imagen, cv2.COLOR_BGR2RGB)
9  #GRIS
10 gris = cv2.cvtColor(imagen, cv2.COLOR_RGB2GRAY)
11 img = imagen[:, :, 1]

```

- Se crea una la imagen con superpíxeles y se suaviza con filtro de mediana

```

1  #aplicar superpíxeles -----
2  slic = 1
3  if slic == 1:
4      img = imagen.copy()
5      img = cv2.medianBlur(img, 21)
6      imgborrar = img.copy()
7      img = superSlic(img)
8      titleSlic = 'slic'
9  else:
10     img = img.copy()
11     titleSlic = 'sin_Slic'
12 # filtrado -----
13 opcion = 1
14 if opcion == 1:
15     mediana = cv2.medianBlur(img, 17)
16     filtro = mediana
17     tituloFiltro = 'MEDIANA'
18     ...
19     ...
20 if opcion == 0:

```

```

21     filtro = img
22     tituloFiltro = 'SIN_FILTRO'

```

- Se realiza la segmentación celular, con el algoritmo de agrupamiento según la opción seleccionada (KMEANS o MEANSHIFT), y posteriormente se extrae al región de núcleo mediante umbralización.

```

1
2     #agrupamiento de regiones -----
3
4     opcionAgrup = 2
5
6     if opcionAgrup == 1:
7         segmentacionImagen = shift2(pixelsValues, filtro)
8         tituloSeg = 'meanShift'
9
10    elif opcionAgrup == 2:
11        segmentacionImagen = kmean(pixelsValues, 6, filtro)
12        tituloSeg = 'kmean'
13
14    else:
15        segmentacionImagen = filtro
16        tituloSeg = 'sin_Agrupamiento'
17
18    # nucleos
19    umbral = np.min(segmentacionImagen)
20    umbral = umbral + 10
21    imagen2 = imagen.copy()
22
23    _, thresh = cv2.threshold(segmentacionImagen, umbral, 255,
24                               cv2.THRESH_BINARY_INV)
25    #operaciones morfologicas
26    kernel = cv2.getStructuringElement(MORPH_ELLIPSE, ksize
27                                       =(5,5))
28    thresh = cv2.erode(thresh, kernel, iterations=1)
29    thresh = cv2.dilate(thresh, kernel, iterations=1)

```

La selección de contornos y regiones.

- Selección de la region del nucleo

```

1     contours, hierarchy = cv2.findContours(thresh, cv2.
2         RETR_EXTERNAL, cv2.CHAIN_APPROX_NONE)
3     contorno, cx, cy = selContorno(contours, thresh)
4     cv2.drawContours(imagen2, contours, contorno,
5         (255,255,0), 2);
6
7     #seleccionador de regiones
8     area = cv2.contourArea(contours[contorno])
9     _, label, _, centroids = cv2.
10    connectedComponentsWithStats(thresh)
11    imagenReg = selRegion(label, area)
12    imagenReg[imagenReg > 200] = 255
13    imagenReg[imagenReg != 255] = 0

```

- Segmentación final, representación gráfica de los resultados.

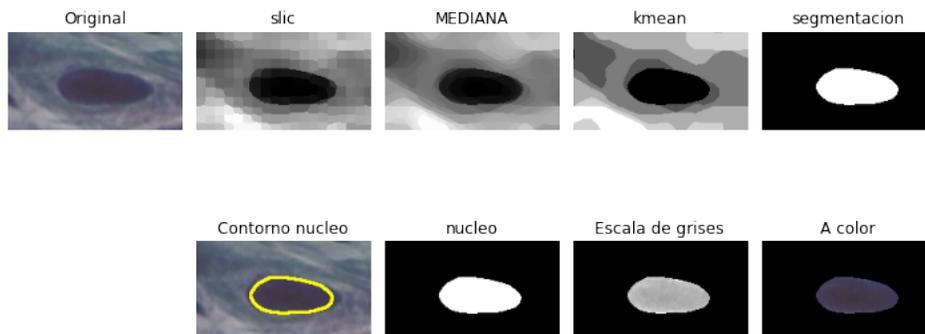


Figura 8.1: Resultados para KMEANS anormalMild50 (fuente: autores)

- Finalmente, se presentan los archivos .CSV del procesamiento de las células PAP, en los cuales se aloja la información de las características junto con las métricas de evaluación.

El algoritmo de segmentación y clasificación con UNET se compone de las siguientes etapas. Se requiere que el modelo y los pesos pre-entrenados se descarguen para un correcto funcionamiento del algoritmo. A continuación se describe brevemente las etapas del algoritmo de segmentación con UNET:

- Se cargan los datos de la célula de la carpeta especificada y se realiza el preprocesamiento, que incluye la redimensión de la imagen y la creación de una copia.

```
1 # RUTA DE LA IMAGEN
2 dirCarpeta = 'DatasetNormalizado/1_normal_superficial/
3             normalSup2.jpg'
4
5 # PREPROCESAMIENTO
6 # BGR
7 imagen = cv2.imread(dirCarpeta)[:,:,:3]
8 imagen = cv2.resize(imagen, dsize=(128,128))
9 imagen2 = imagen.copy()
10 imagenT = np.array([imagen2, imagen2])
```

- Se realiza la segmentación de la imagen utilizando la red UNET. Se selecciona el caso de segmentación (1 o 2) según sea para segmentar núcleo o citoplasmas.

```
1 case= 1
2 if case == 1:
3     segmentacion = model1.predict(imagenT)
4 elif case == 2:
5     segmentacion = model2.predict(imagenT)
```

- Se aplican operaciones de filtrado y binarización a la salida de la segmentación para obtener una imagen binaria que destaque los núcleos celulares

```
1 imgIn = imagen.copy()
2 seg = segmentacion[0].reshape(128,128)
3 seg = Lambda(lambda x: x*255)(seg)
4 seg = np.asarray(seg, dtype=np.uint8)
5 # Aplicar filtro de mediana con kernel de 5x5
```

```

6   gauss = cv2.medianBlur(seg,11)
7   #binarizacion por el metodo de otsu
8   ret3, binarizar = cv2.threshold(gauss,10,255,cv2.
      THRESH_BINARY+cv2.THRESH_OTSU)
9   #operacion morfologica
10  ele = cv2.getStructuringElement(MORPH_ELLIPSE, ksize
      =(3,3))
11  binarizar = cv2.erode(binarizar,ele,iterations=2)
12  binarizar = cv2.dilate(binarizar,ele,iterations=2)

```

- Se buscan los contornos de los núcleos en la imagen binaria y se selecciona el contorno que se encuentre hacia el centro de la imagen. Esto soluciona el problema de múltiples núcleos o citoplasmas en caso de que los haya.

```

1   # Buscamos los contornos
2   (contornos,_) = cv2.findContours(binarizar.copy(), cv2.
      RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
3   print("He encontrado {} posibles núcleos".format(len(
      contornos)))
4   contorno,cx,cy = selContorno(contornos,binarizar)
5   cv2.drawContours(imgIn,contornos,-1,(255,255,0), 2)

```

- Se toma la región correspondiente al núcleo seleccionado en la parte anterior y se aplica relleno de orificios. Esto soluciona el problema de orificios presentes en el núcleo y/o citoplasma.

```

1   #seleccionador de regiones
2   area = cv2.contourArea(contornos[contorno])
3   _, label, _,centroids = cv2.connectedComponentsWithStats(
      binarizar)
4   imagenReg = selRegion(label,area)
5   imagenReg[imagenReg > 200] = 255
6   imagenReg[imagenReg != 255] = 0

```

- Se detectan los bordes en la imagen de los núcleos utilizando el algoritmo Canny.

```

1   # deteccion de bordes
2   bordes = cv2.Canny(imagenReg, 10, 150)

```

- Se convierte la imagen original y la imagen de los núcleos a escala de grises. Se crea una imagen a color en la que se mantienen solo los píxeles correspondientes a los núcleos celulares.

```

1  #imagen en gris y color
2  imgColor = imagen.copy()
3  gris = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)
4  for fila in range (gris.shape[0]):
5      for columna in range (gris.shape[1]):
6          if imagenReg[fila][columna] == 0:
7              gris[fila][columna] = 0
8              imgColor[fila][columna] = 0
9          else:
10             pass

```

- En las Figuras 8.2 y 8.3, se muestran los resultados de la etapa de segmentación de núcleo y citoplasma de una célula con UNET.

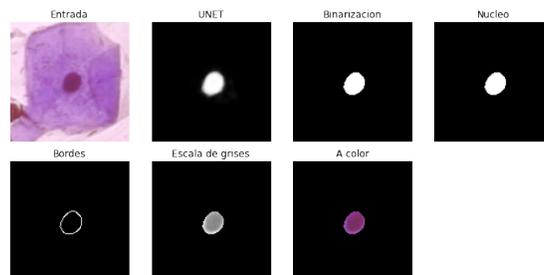


Figura 8.2: Segmentación de núcleo UNET (fuente: autores)

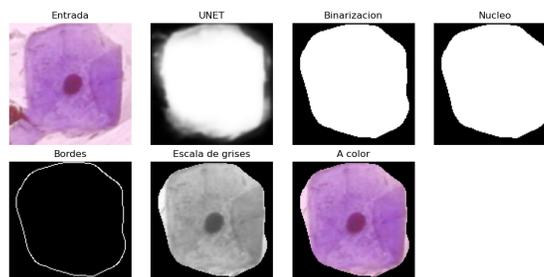


Figura 8.3: Segmentación de citoplasma UNET (fuente: autores)

A continuación se describe brevemente las etapas del algoritmo de clasificación:

- Se cargan los datos de las características de núcleos y citoplasmas y se combinan en un dataframe. Los datos se encuentran balanceados según el tipo de clasificación de 2 o 3 clases. Se realiza la estandarización de las características.

```
1 # Datos Balanceados
2 caractNucleo = pd.read_csv('Archivos_CSV/
   caractNucleoUnetBinBal2.csv', sep=';', index_col=0)
3 caractCito = pd.read_csv('Archivos_CSV/
   caractCitoUnetBinBal2.csv', sep=';', index_col=0)
4
5 #Etiquetado características del nucleo
6 for i,e in enumerate(caractNucleo.columns):
7     nombre = 'n'+e
8     caractNucleo.rename(columns={e:nombre}, inplace=True)
9
10 caract = pd.DataFrame()
11 caract = pd.concat([caractNucleo, caractCito], axis=1)
12
13 caract['NC'] = caractNucleo['nAREA']/caractCito['AREA']
14
15 caract.drop(columns=['nCLASE', 'nCELULA'], inplace=True)
```

- Se calcula la matriz de correlación lineal de cada característica con la variable objetivo (clase).

```
1 matrizCorrelacion = matrizCaract[matrizCaract.columns.
   drop(['HERLEV', 'CYT', 'HIST'])]
2 corr = matrizCorrelacion.corr()
3 plt.figure(figsize=(25,20))
4 sns.heatmap(corr, cbar = True, square = True, annot=True
   , fmt = '.2f', annot_kws={'size': 7},
5           xticklabels= matrizCorrelacion.columns,
6           yticklabels= matrizCorrelacion.columns,
7           cmap= 'coolwarm')
8 plt.xticks(rotation = 90)
```

```

9 plt.yticks(rotation = 0)
10 plt.title('CORRELACION LINEAL')
11 plt.show()

```

- Se seleccionan las características principales ya sea mediante FEATUREWIZ y/o PCA
- Se generan los datos de entrenamiento y de prueba para los diferentes clasificadores. La siguiente proporción código corresponde a un espacio de características por PCA

```

1 dataPCA2 = dataPCA.values
2 X = dataPCA2[:,0:-2]
3 Y = dataPCA2[:, -1]
4 x_train,x_test,y_train,y_test= train_test_split(X,Y,
           test_size=0.2,random_state=0,stratify=Y)

```

En la siguiente fracción de código se muestra la creación de un espacio de características con FEATUREWIZ:

```

1 feat = features[0].copy()
2 feat.append('CYT')
3 matrizCorrelacion2 = matrizCorrelacion[feat].values
4 X = matrizCorrelacion2[:,0:-2]
5 Y = matrizCorrelacion2[:, -1]
6
7 x_train,x_test,y_train,y_test= train_test_split(X,Y,
           test_size=0.2,random_state=0,stratify=Y)

```

En ambos casos se establece un conjunto datos con el 80% para entrenamiento y el 20% restante para prueba. Se mantiene una semilla de 0 para mantener la reproducibilidad y se realiza la estratificación en función de las etiquetas Y para preservar la proporción de las clases en los conjuntos de entrenamiento y prueba.

- Se realiza la clasificación de los datos con diferentes clasificadores como: DT, SVM, KNN y ANN
- Para cada clasificador, en primer lugar, se busca el mejor hiperparámetro de forma iterativa.

```

1         # optimizando hiperpar metros
2 aTest = []
3 aTrain = []
4 valor = np.arange(1,90,1)
5 for p in valor:
6     knn = KNeighborsClassifier(n_neighbors=p,weights='
7         distance').fit(x_train,y_train)
8     y_train_predict = knn.predict(x_train)
9     y_test_predict = knn.predict(x_test)
10    scoreTrain = accuracy_score(y_train_predict,y_train)
11    scoreTest = accuracy_score(y_test_predict,y_test)
12    aTest.append(scoreTest)
13    aTrain.append(scoreTrain)
14    #print('PROF:',p , 'SCORETEST: ',scoreTest,'
15        SCORETRAIN: ',scoreTrain)
16    plt.figure(figsize=(7,3),dpi=98)
17    plt.plot(valor,aTrain,'-o',label='train')
18    plt.plot(valor,aTest,'-o',label='test')
19    plt.tight_layout()
20    plt.legend()
21    #plt.ylim(0,1)
22    plt.xlim(min(valor),max(valor))
23    plt.title('OPTIMIZACION_HIPERPARAMETROS_KNN')
24    plt.xlabel('k_vecinos')
25    plt.ylabel('Accuracy')

```

- Luego, se evalúa el desempeño del modelo con su correspondiente hiperparámetro por medio de validación cruzada en los datos de entrenamiento.

```

1     knn = KNeighborsClassifier(n_neighbors=20,weights='
2         distance')
3     scores = cross_val_score(knn,x_train,y_train,cv=5,scoring
4         ='accuracy')
5     print('RESULTADOS')
6     print('puntaje_por_iteracion:',scores, 'media:',scores.
7         mean())

```

- Se ajusta el modelo de clasificación con el conjunto de entrenamiento y el parámetro seleccionado. Posteriormente, se realizan las predicciones sobre el conjunto de prueba, y se calcula el porcentaje de acierto para cada observación. Finalmente, se muestran los resultados por medio de una matriz de confusión normalizada y su correspondiente informe de clasificación.

```

1     knn = KNeighborsClassifier(n_neighbors=20, weights='
           distance').fit(x_train, y_train)
2     y_train_predict = knn.predict(x_train)
3     y_test_predict = knn.predict(x_test)
4     y_test_predict_prob = knn.predict_proba(x_test)
5     print('porcentaje de aciertos sobre datos de
           entrenamiento', accuracy_score(y_train_predict, y_train)
           )
6     print('porcentaje de aciertos sobre datos de test',
           accuracy_score(y_test_predict, y_test))
7     cm = confusion_matrix(y_true=y_test, y_pred=y_test_predict
           , normalize='true')
8     matriz = ConfusionMatrixDisplay(confusion_matrix=cm,
           display_labels=knn.classes_)
9     matriz.plot(cmap='Blues')
10
11    print('-----')
12    accuracy = accuracy_score(y_test, y_test_predict,)
13    precision = precision_score(y_test, y_test_predict, average
           ='macro')
14    recall = recall_score(y_test, y_test_predict, average='
           macro')
15    f1 = f1_score(y_test, y_test_predict, average='macro')
16    print('EVALUACION SOBRE DATOS DE TEST')
17    print('accuracy:', accuracy, 'precision:', precision, '
           recall:', recall, 'f1:', f1)
18
19    # informe de clasificacion
20    print('-----')
21    target_names = ['clase1', 'clase2', 'clase3']
22    print('INFORME DE CLASIFICACION')

```

23

```
print(classification_report(y_test, y_test_predict,
                             target_names=target_names))
```

- Para finalizar, se muestran las gráficas 'AUC-ROC' para el modelo de clasificación.

### 8.1.2 Instalación de paquetes para el entorno de desarrollo

- OpenCV

OpenCV (Open Source Computer Vision Library) es una biblioteca de software de visión por computadora y aprendizaje automático de código abierto. Proporciona un conjunto de herramientas y funciones que permiten a los desarrolladores procesar imágenes y videos en tiempo real. OpenCV fue inicialmente desarrollado por Intel en 1999 y ha sido mantenido por una comunidad activa desde entonces. La biblioteca ofrece algoritmos y técnicas para diversas tareas de visión por computadora.

OpenCV se puede descargar desde su página principal de internet ([enlace](#)), mediante la línea de comando o terminal de python se puede instalar la biblioteca del software:

```
1 pip install opencv-python
```

la versión de biblioteca: cv2 versión 4.6.0

- Sklearn

Sklearn es una biblioteca de código abierto para Python que se centra en el aprendizaje automático (machine learning). Proporciona una amplia variedad de algoritmos y herramientas para analizar datos y construir modelos predictivos. Los algoritmos incluyen clasificación, regresión y agrupamiento, como máquinas de vectores de soporte, bosques aleatorios, aumento de gradiente, k-medias y DBSCAN.

Sklearn puede descargarse desde su página principal ([enlace](#))

```
1 pip install scikit-learn
```

la versión de biblioteca: scikit-learn versión 1.1.1

- Tensorflow

Es una Biblioteca de aprendizaje automático y de inteligencia artificial desarrollado por Google. Permite la construcción y entrenamiento de redes neuronales profundas. Utiliza un paradigma de programación basado en grafos de flujo de datos y ofrece una amplia gama de herramientas. Se usó esta herramienta para entrenar la red UNET. No es necesario volver a realizar el entrenamiento, dado que sus pesos y estructura ya quedan guardados para usarla para predicciones. Dentro del repositorio se usa la red ya entrenada para hacer segmentación de nucleo y citoplasma.

Para realizar la instalación de esta herramienta se puede seguir el siguiente enlace:(enlace video)

- Intalar los paquetes necesarios

pueden ser instalados mediante el archivo *requirements.txt* del repositorio:

```
1 pip install -r requirements.txt
```

- numpy
- pandas
- matplotlib
- seaborn
- opencv-python
- scikit-learn
- scikit-image
- scipy
- featurewiz
- tqdm
- kneed
- seaborn
- plotly-express
- os-sys

## 8.2 Prueba de segmentación con imágenes PAP

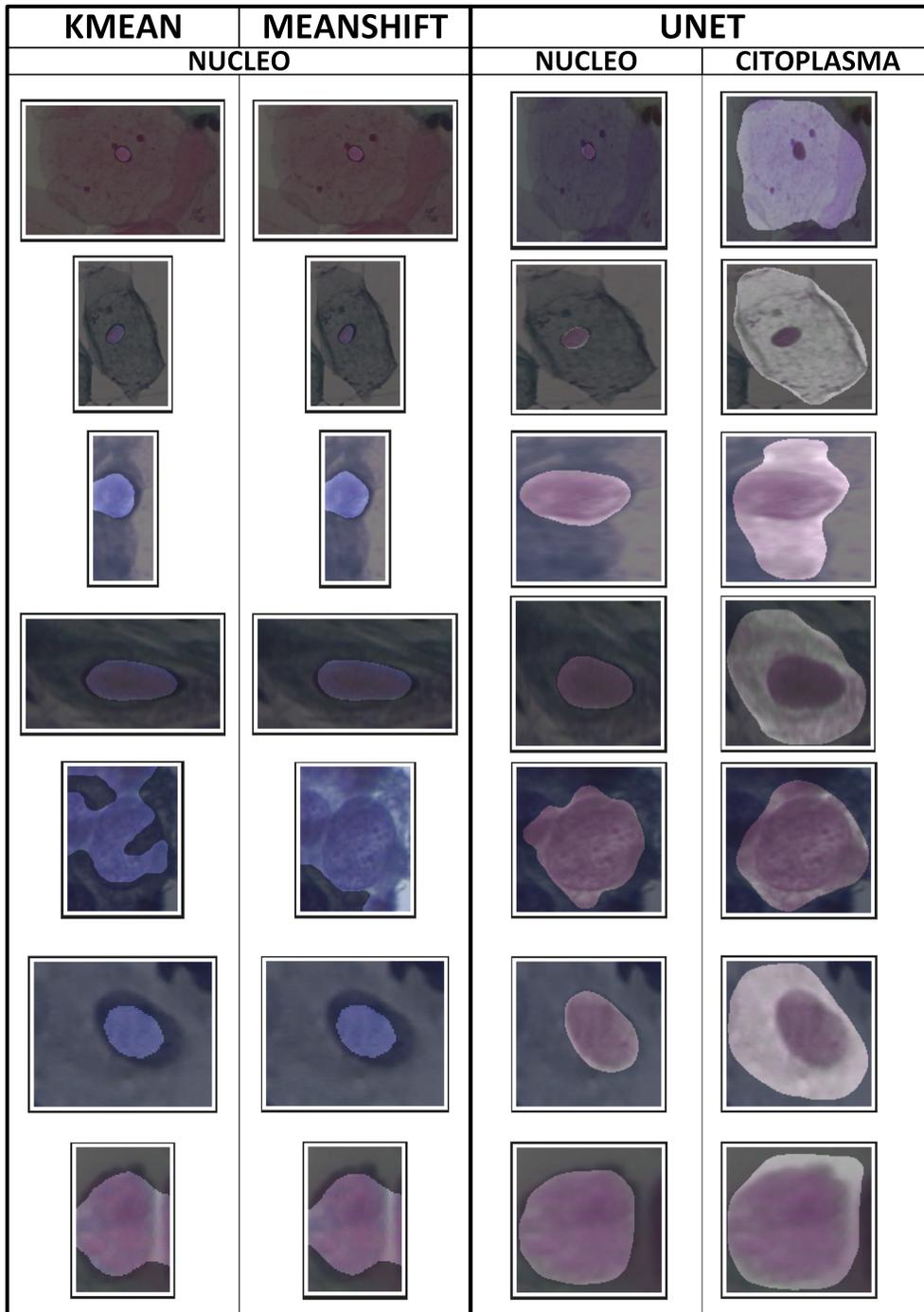


Figura 8.4: Prueba de segmentación de imágenes con las 3 técnicas propuestas (fuente: autores).

## Referencias

- [1] M. E. Plissiti, P. Dimitrakopoulos, G. Sfikas, C. Nikou, O. Krikoni, and A. Charchanti, “SIPAKMED: A new dataset for feature and image based classification of normal and pathological cervical cells in pap smear images,” October 2018.
- [2] J. Jantzen, J. Norup, G. Dounias, and B. Bjerregaard, “Pap-smear benchmark data for pattern classification,” *Nature inspired Smart Information Systems (NiSIS 2005)*, pp. 1–9, 2005.
- [3] J. M. Ojeda, “Virus papiloma humano: de la biología a la clínica,” *Medwave*, vol. 10, 7 2010.
- [4] A. G. Marcos, F. J. M. de Pisón Ascacíbar, A. V. P. Espinoza, F. A. Elías, M. C. Limas, J. B. O. Meré, E. P. V. González, A. G. Marcos, F. J. M. de Pisón Ascacíbar, A. V. P. Espinoza, F. A. Elías, M. C. Limas, J. B. O. Meré, and E. P. V. González, “Técnicas y algoritmos básicos de visión artificial,” *Técnicas y Algoritmos Básicos de Visión Artificial*, 2006.
- [5] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *MICCAI*, 2015.
- [6] E. J. C. Suárez, “Tutorial sobre máquinas de vectores soporte (svm),” *Tutorial sobre Máquinas de Vectores Soporte (SVM)*, vol. 1, pp. 1–12, 2014.
- [7] G. James, D. Witten, T. Hastie, and R. Tibshirani, *Introduction to statistical learning*, vol. 112. Springer, 2013.
- [8] R. H. Rodríguez Espinosa, “Módulo para la validación de la segmentación en imágenes de opacidad de la cápsula posterior provenientes del pentacam,” B.S. thesis, Universidad de las Ciencias Informáticas. Facultad 3. AIRI, Artificial . . . , 2017.
- [9] B. M. Del Brío and A. S. Molina, *Redes neuronales y sistemas borrosos*. Rama, 2001.

- [10] W. William, A. Ware, A. H. Basaza-Ejiri, and J. Obungoloch, “A review of image analysis and machine learning techniques for automated cervical cancer screening from pap-smear images,” *Computer Methods and Programs in Biomedicine*, vol. 164, pp. 15–22, 2018.
- [11] S. Adhikary, S. Seth, S. Das, T. K. Naskar, A. Barui, and S. P. Maity, “Feature assisted cervical cancer screening through DIC cell images,” *Biocybernetics and Biomedical Engineering*, vol. 41, pp. 1162–1181, jul 2021.
- [12] W. William, A. Ware, A. H. Basaza-Ejiri, and J. Obungoloch, “A pap-smear analysis tool (pat) for detection of cervical cancer from pap-smear images,” *BioMedical Engineering Online*, vol. 18, pp. 1–22, 2 2019.
- [13] P. Wang, L. Wang, Y. Li, Q. Song, S. Lv, and X. Hu, “Automatic cell nuclei segmentation and classification of cervical Pap smear images,” *Biomedical Signal Processing and Control*, pp. 93–103.
- [14] B. Sokouti, S. Haghypour, and A. D. Tabrizi, “A framework for diagnosing cervical cancer disease based on feedforward mlp neural network and thinprep histopathological cell image features,” *Neural Computing and Applications*, vol. 24, no. 1, pp. 221–232, 2014.
- [15] J. Martínez-Más, A. Bueno-Crespo, R. Martínez-España, M. Remezal-Solano, A. Ortiz-González, S. Ortiz-Reina, and J. P. Martínez-Cendán, “Classifying Papanicolaou cervical smears through a cell merger approach by deep learning technique,” *Expert Systems with Applications*, vol. 160, p. 113707, dec 2020.
- [16] A. Sarwar, A. A. Sheikh, J. Manhas, and V. Sharma, “Segmentation of cervical cells for automated screening of cervical cancer: a review,” *Artificial Intelligence Review*, vol. 53, pp. 2341–2379, 4 2020.
- [17] World Health Organization, “Cervical cancer.” <https://www.who.int/news-room/fact-sheets/detail/cervical-cancer>.
- [18] J. A. Hernández Vargas, P. X. Ramírez Barbosa, A. M. Valbuena-Garcia, L. Acuña, and J. A. González-Diaz, “Factors associated with delays in time to treatment initiation in Colombian women with cervical cancer: A cross-sectional analysis,” *Gynecologic Oncology Reports*, vol. 35, p. 100697, feb 2021.

- [19] Ministerio de Salud y Protección Social de Colombia, “Cáncer de cuello uterino.” <https://www.minsalud.gov.co/salud/publica/ssr/Paginas/Cancer-de-cuello-uterino.aspx>. Accedido el 7/07/2023.
- [20] Así vamos en Salud, “Mortalidad por cáncer de cuello uterino.” <https://www.asivamosensalud.org/indicadores/enfermedades-cronicas-no-transmisibles/mortalidad-por-cancer-de-cuello-uterino>. Accedido el 7/07/2023.
- [21] G. N. Papanicolaou and H. F. Traut, “The diagnostic value of vaginal smears in carcinoma of the uterus,” *American Journal of Obstetrics and Gynecology*, vol. 42, no. 2, pp. 193–206, 1941.
- [22] K. Santosh, L. Wendling, S. Antani, and G. R. Thoma, “Overlaid arrow detection for labeling regions of interest in biomedical images,” *IEEE Intelligent Systems*, vol. 31, no. 3, pp. 66–75, 2016.
- [23] H. Lee and J. Kim, “Segmentation of overlapping cervical cells in microscopic images with superpixel partitioning and cell-wise contour refinement,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 63–69, 2016.
- [24] C. Li, H. Chen, X. Li, N. Xu, Z. Hu, D. Xue, S. Qi, H. Ma, L. Zhang, and H. Sun, “A review for cervical histopathology image analysis using machine vision approaches,” *Artificial Intelligence Review*, pp. 4821–4862, oct.
- [25] D. N. Diniz, R. F. Vitor, A. G. C. Bianchi, S. Delabrida, C. M. Carneiro, D. M. Ushizima, F. N. S. de Medeiros, and M. J. F. Souza, “An ensemble method for nuclei detection of overlapping cervical cells,” *Expert Systems with Applications*, vol. 185, p. 115642, dec 2021.
- [26] L. Zhao, K. Li, M. Wang, J. Yin, E. Zhu, C. Wu, S. Wang, and C. Zhu, “Automatic cytoplasm and nuclei segmentation for color cervical smear image using an efficient gap-search mrf,” *Computers in Biology and Medicine*, vol. 71, pp. 46–56, 4 2016.
- [27] A. Tareef, Y. Song, W. Cai, H. Huang, H. Chang, Y. Wang, M. Fulham, D. Feng, and M. Chen, “Automatic segmentation of overlapping cervical smear cells based on local distinctive features and guided shape deformation,” *Neurocomputing*, vol. 221, pp. 94–107, 1 2017.

- [28] G. A. González, “Definición de machine learning.” <https://www.definicionabc.com/tecnologia/machine-learning.php>, octubre 2018.
- [29] Á. Artola Moreno, “Clasificación de imágenes usando redes neuronales convolucionales en python,” 2019.
- [30] L. Zhang, L. Lu, I. Noguez, R. M. Summers, S. Liu, and J. Yao, “Deeppap: deep convolutional networks for cervical cell classification,” *IEEE journal of biomedical and health informatics*, vol. 21, no. 6, pp. 1633–1643, 2017.
- [31] A. D. Jia, B. Z. Li, and C. C. Zhang, “Detection of cervical cancer cells based on strong feature cnn-svm network,” *Neurocomputing*, vol. 411, pp. 112–127, 10 2020.
- [32] T. Wang, J. Huang, D. Zheng, and Y. He, “Nucleus Segmentation of Cervical Cytology Images Based on Depth Information,” *IEEE Access*, vol. 8, pp. 75846–75859, 2020.
- [33] C. Rajarao and R. P. Singh, “Improved normalized graph cut with generalized data for enhanced segmentation in cervical cancer detection,” *Evolutionary Intelligence*, vol. 13, pp. 3–8, 3.
- [34] A. M. Braga, R. C. Marques, F. N. Medeiros, J. F. Neto, A. G. Bianchi, C. M. Carneiro, and D. M. Ushizima, “Hierarchical median narrow band for level set segmentation of cervical cell nuclei,” *Measurement*, vol. 176, p. 109232, 5 2021.
- [35] M. S. R. Vázquez, I. A. Vidal, and M. Borges, “Alternative tool for classification of cervical cells using only features of the nucleus,” *Revista Cubana de Informática Médica*, vol. 8, pp. 224–238, 7 2016.
- [36] F. Boemer, E. Ratner, and A. Lendasse, “Parameter-free image segmentation with slic,” *Neurocomputing*, vol. 277, pp. 228–236, 2 2018.
- [37] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.
- [38] J. MacQueen, “Some methods for classification and analysis of multivariate observations,” *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, 1967.

- [39] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. Wiley, 1973.
- [40] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 5, pp. 603–619, 2002.
- [41] X. Wang, H. Liu, and W. Ma, “Sparse least squares support vector machines based on meanshift clustering method,” *IFAC-PapersOnLine*, vol. 51, no. 18, pp. 292–296, 2018. 10th IFAC Symposium on Advanced Control of Chemical Processes ADCHEM 2018.
- [42] P. Sheng, Y. Shi, X. Liu, and H. Jin, “Lsnet: Real-time attention semantic segmentation network with linear complexity,” *Neurocomputing*, vol. 509, pp. 94–101, 2022.
- [43] N. A. Ramírez Pérez *et al.*, “Modelo de clasificación automática de imágenes de resonancia magnética para el diagnóstico del cáncer de próstata.” 2020.
- [44] M. Presutti, “La matriz de co-ocurrencia en la clasificación multispectral: tutorial para la enseñanza de medidas texturales en cursos de grado universitario,” *4<sup>a</sup> Jornada de Educação em Sensoriamento Remoto no Âmbito do Mercosul*, pp. 1–9, 2004.
- [45] M. C. J. Jaime and E. Elizondo, “Fundamentos de procesamiento de imágenes instituto tecnológico de ciudad madero,” 2002.
- [46] W. William, A. Ware, A. H. Basaza-Ejiri, and J. Obungoloch, “Cervical cancer classification from pap-smears using an enhanced fuzzy c-means algorithm,” *Informatics in Medicine Unlocked*, vol. 14, pp. 23–33, 2019.
- [47] G. K. Lakshmi and K. Krishnaveni, “Feature extraction and feature set selection for cervical cancer diagnosis,” *Indian Journal of Science and Technology*, vol. 9, no. 19, pp. 1–7, 2016.
- [48] J. D. H. Lalinde, F. E. Castro, J. E. Rodríguez, J. G. C. Rangel, C. T. Sierra, M. K. A. Torrado, S. M. C. Sierra, and V. J. B. Pirela, “Sobre el uso adecuado del coeficiente de correlación de pearson: definición, propiedades y suposiciones,”

- Archivos venezolanos de Farmacología y Terapéutica*, vol. 37, no. 5, pp. 587–595, 2018.
- [49] S. Ramírez-Gallego, S. García, J. M. Luengo, JuliDinizand Benítez, and F. Herrera, “Featurewiz: Dinizaautomated feature engineering tool,” *Information Sciences*, vol. 45, pp. 1–15, 2018.
- [50] Z. Zhao, R. Anand, and M. Wang, “Maximum relevance and minimum redundancy feature selection methods for a marketing machine learning platform,” *Proceedings - 2019 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2019*, pp. 442–452, 10 2019.
- [51] “Reconocimiento de emociones de la señal eeg usando el algoritmo xgboost,” in *2019 IEEE 16a Conferencia Internacional del Consejo de la India (INDICON)*.
- [52] M. S. Park and J. Y. Choi, “Theoretical analysis on feature extraction capability of class-augmented pca,” *Pattern Recognition*, vol. 42, no. 11, pp. 2353–2362, 2009.
- [53] R. Geetha, S. Sivasubramanian, M. Kaliappan, S. Vimal, and S. Annamalai, “Cervical cancer identification with synthetic minority oversampling technique and pca analysis using random forest classifier,” *Journal of medical systems*, vol. 43, pp. 1–19, 2019.
- [54] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of Eugenics*, vol. 7, no. 2, pp. 179–188, 1936.
- [55] O. Yalçın, *Introduction to Machine Learning*, pp. 33–55. Berkeley, CA: Apress, 2021.
- [56] R. E. B. Martínez, N. C. Ramírez, H. G. A. Mesa, I. R. Suárez, M. Trejo, P. P. León, and S. L. B. Morales, “Árboles de decisión como herramienta en el diagnóstico médico,” *Revista médica de la Universidad Veracruzana*, vol. 9, no. 2, pp. 19–24, 2009.
- [57] H. Vivas, “Optimización en entrenamiento del perceptrón multicapa,” *Universidad del Cauca, Popayán*, 2014.
- [58] D. Marín-Soto, “Segmentación de células mediante técnicas de procesamiento digital de imágenes para el rastreo de células cancerosas,” 2018.

- [59] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, “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.
- [60] D. Revelo Luna, “Semantic segmentation basic u-net.” <https://github.com/DavidReveloLuna/Semantic-Segmentation-Basic-U-Net>, 2023.
- [61] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [62] VolkerH, “Example gist.” <https://gist.github.com/VolkerH/0d07d05d5cb189b56362e8ee41882abf>, 2023.
- [63] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, “Segment anything,” 2023.

# Algoritmo de segmentación semántica UNET y clasificación del grado patológico de imágenes PAP

1<sup>st</sup> Jean Betancourt

Ingeniería Automática Industrial  
Universidad del Cauca  
Popayán, Colombia  
jeancarlos@unicauca.edu.co

2<sup>nd</sup> Yeiner Imbachi

Ingeniería Automática Industrial  
Universidad del Cauca  
Popayán, Colombia  
yeinerimbachi@unicauca.edu.co

3<sup>rd</sup> Elena Muñoz

Depto Electrónica, Instrumentación y Control  
Universidad del Cauca  
Popayán, Colombia  
elenam@unicauca.edu.co

**Abstract**—Cervical cancer is a common and serious disease that affects many women worldwide. However, when detected in early or precancerous stages, it can be successfully treated. In the field of medicine, computer-assisted detection tools have been developed, which play a fundamental role in the screening process of cervical cancer cells. These tools are used to improve the precision and efficiency of the detection process. In this particular study, a PAP cervical image segmentation method is presented using the UNET semantic segmentation technique. This method enables precise segmentation of regions of interest, such as the nucleus and cytoplasm of cells. Segmentation provides detailed information about the shape, size, texture, and color of these regions, which were used in classifiers such as ANN, KNN, SVM, DT, by combining different feature spaces using PCA and a new feature selection method applied to the PAP image domain called FEATUREWIZ. These classifiers are used to determine the pathological grade of cells, classifying them into categories such as normal and abnormal, or into three categories: normal, LSIL (low-grade squamous intraepithelial lesions), and HSIL (high-grade squamous intraepithelial lesions). The study achieved high precision and quality results in the segmentation and determination of the pathological grade of cells.

**Index Terms**—Cervical cancer, PAP image processing, machine learning.

## I. INTRODUCCIÓN

Los sistemas de diagnóstico asistido por computadora (CAD) a partir de la Imágenes microscópicas de células teñidas de Papanicolaou (PAP), son herramientas que permiten analizar automáticamente imágenes de frotis cervical y que brindan apoyo al proceso de detección de cáncer de cuello uterino. Una de las etapas típicas en el análisis automatizado de imágenes PAP es la segmentación, que consiste en la separación de los componentes de interés de la célula, para luego determinar características malignas o anormales, con las cuales se pueda realizar la clasificación. Se han desarrollado numerosas técnicas de segmentación, como las que hacen uso de filtros, detección de bordes y umbrales a partir del histograma de la imagen [1].

Las características ideales de un algoritmo de procesamiento de imágenes PAP, es que pueda hacer el diagnóstico de

forma satisfactoria, a partir del análisis de las células, y que sea robusto frente a problemas relacionados con la muestra, como el solapamiento, superposición, moco, y presencia de partículas ajenas a la célula de interés [1]. Las características ideales de un algoritmo de procesamiento de imágenes PAP, es que pueda hacer el diagnóstico de forma satisfactoria, a partir del análisis de las células, y que sea robusto frente a problemas relacionados con la muestra, como el solapamiento, superposición, moco, y presencia de partículas ajenas a la célula de interés [1]. Es importante tener en cuenta toda la información que se pueda extraer a partir de las imágenes de las células, ya que de ello depende la robustez de un algoritmo de clasificación que realmente pueda apoyar el diagnóstico de dicha enfermedad [2]. La superioridad de un sistema de diagnóstico asistido por computador (CAD) depende en gran medida de los factores que incluyen la precisión del preprocesamiento, la segmentación, y el tipo de conjuntos de datos utilizados [1].

En la actualidad, se han desarrollado algoritmos que facilitan el proceso de diagnóstico imitando la forma en que el especialista realiza el análisis a partir de imágenes de Papanicolaou. Generalmente, estos algoritmos se dividen en cuatro fases conocidas como preprocesamiento, segmentación, entrenamiento, clasificación, con los cuales se aborda el problema de detección. Lo ideal es que el algoritmo que se implementa, sea capaz de llevar a cabo cada una de esas etapas de forma satisfactoria.

De acuerdo con [3] [2], uno de los principales retos para el procesamiento de imágenes PAP, es precisamente la dificultad para segmentar correctamente el núcleo y el citoplasma; se encuentran imágenes donde las células presentan solapamiento entre ellas, lo que implícitamente afecta el proceso de obtención de características morfológicas y clasificación. En el mismo sentido, en la mayoría de trabajos enfocados a segmentación, no se consideran las características que se pueden extraer a partir del citoplasma, lo que puede mejorar significativamente la eficiencia para la clasificación. Adicionalmente, uno de los problemas que dificultan el acondicionamiento o preprocesamiento de la imagen se encuentran relacionados con

la calidad de las muestras tomadas, las cuales contienen otros elementos diferentes a las células de interés, que dificultan el análisis por computadora [4].

En las siguientes secciones se propone un algoritmo de segmentación semántica para imágenes PAP haciendo uso del conjunto de datos HERLEV. Se trata de una CNN conocida como UNET, para identificación de núcleo y citoplasma celular. Posteriormente se extraen y prueban distintos espacios de características en cuatro clasificadores, para seleccionar el mejor en tareas de clasificación binaria y de tres clases.

## II. ESQUEMA METODOLÓGICO

Para la realización de este proyecto de investigación, se utilizó una metodología basada en fases y actividades ordenadas de forma secuencial, que permita definir el alcance de cada fase en los tiempos determinados. En la Figura 1, se presentan las etapas de desarrollo:

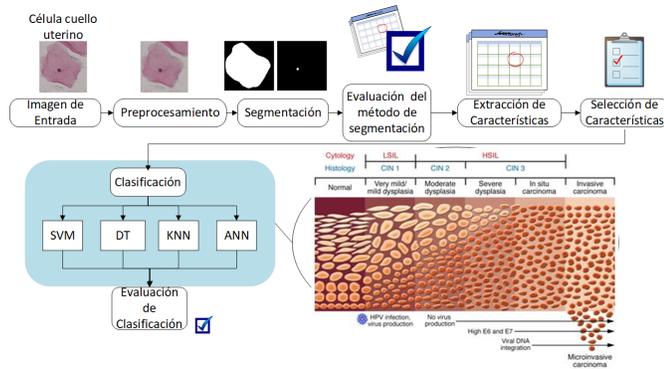


Figura 1: Metodología general (fuente: autores)

Se presenta un esquema para la segmentación y clasificación de imágenes de frotis cervical, constituido por los siguientes apartados: preprocesamiento, segmentación, extracción/selección de características y por último clasificación. En la última sección se presentan los resultados y conclusiones de este estudio.

Por otro lado, cada una de las etapas de procesamiento fue desarrollada con Python, principalmente con las librerías de *OpenCV* para procesamiento de imágenes, *scikit-learn* para modelos de aprendizaje automático, así como *Numpy*, *Pandas*, y *Seaborn* para operaciones matriciales y generación de gráficas, entre otras. Se tienen los siguientes recursos computacionales:

- Procesador intel core i7 5500 U @ 3GHz
- Tarjeta gráfica NVIDIA GEFORCE 940M 2BG
- Núcleos CUDA 384
- RAM 8 GB
- Sistema operativo Windows 11
- Python v3.10

## III. ADQUISICIÓN DE IMÁGENES

Se utilizaron imágenes de frotis cervical de la prueba de Papanicolaou (PAP) de la base de datos de HERLEV, es una base de datos pública que contiene una amplia variedad

de lesiones cervicales previamente clasificadas por expertos [5]. Este conjunto de datos se ha utilizado como referencia estándar en la investigación médica, lo que la convierte en una herramienta valiosa para analizar y comparar diferentes tipos de lesiones cervicales. El conjunto de datos de HERLEV de imágenes PAP consta de un total de 917 imágenes de células de la prueba de Papanicolaou. La tabla I muestra un resumen del tipo de células presentes y la cantidad de imágenes en cada categoría:

Clase	Categoría	Tipo de Célula	Recuento de Células
1	Normal	Superficial squamous epithelial	74
2	Normal	Intermediate squamous epithelial	70
3	Normal	Columnar epithelial	98
4	Anormal	Mild squamous non-keratinizing dysplasia	182
5	Anormal	Moderate squamous non-keratinizing dysplasia	146
6	Anormal	Severe squamous non-keratinizing dysplasia	197
7	Anormal	Squamous cell carcinoma in situ intermediate	150
<b>Subtotal normal</b>			<b>242</b>
<b>Subtotal anormal</b>			<b>675</b>

Tabla I: Resumen de clases y tipos de células en el conjunto de datos de HERLEV (fuente: autores).

Estas imágenes se dividen en diferentes clases, con las clases 1 a 3 que contienen células normales y las clases 4 a 7 que contienen células anormales como se ve en la Figura 2. Es importante tener en cuenta que la distribución las imágenes no es uniforme entre las categorías, con una predominancia de células anormales. Además, cada imagen contiene una sola célula, lo que permite evaluar individualmente y en detalle cada una de ellas. El dataset cuenta igualmente con un conjunto de imágenes de referencia para cada una de las células, presentando en diferentes tonos las regiones que componen la imagen.

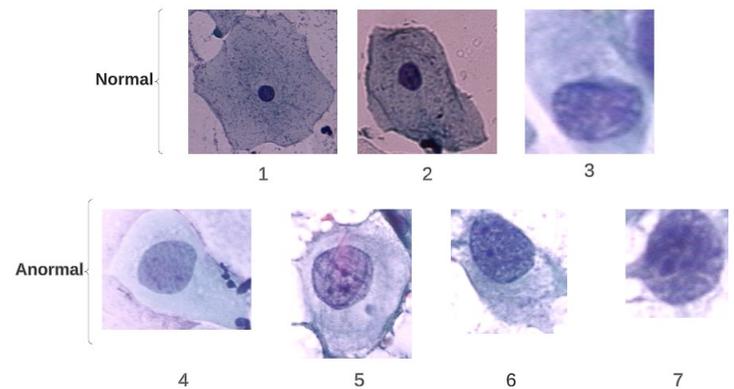


Figura 2: Células del dataset

## IV. MEDIDAS DE EVALUACIÓN

Las medidas estadísticas de error permiten analizar la frecuencia con la que se clasifican o segmentan de forma errónea los datos de una imagen. Esta evaluación se realiza mediante las siguientes cuatro variables [6]:

- Verdaderos positivos (VP): número de píxeles que pertenecen a la región objetivo a segmentar y son clasificados correctamente.
- Verdaderos negativos (VN): número de píxeles que pertenecen al fondo y son clasificados correctamente.

- Falsos positivos (FP): número de píxeles pertenecientes al fondo que son clasificados como parte de la región objetivo.
- Falsos negativos (FN): número de píxeles de la región objetivo que son clasificados como fondo.

La métricas de evaluación utilizadas para informar los resultados son:

$$Precisión = \frac{VP}{VP + FP}, Recall = \frac{VP}{VP + FN}$$

El índice de DICE es una medida que permite evaluar la similitud existente entre dos imágenes binarias [7], Para un par de imágenes  $U$  y  $V$  que son las imágenes segmentada y de referencia respectivamente, el índice de DICE tiene la forma de la siguiente ecuación.

$$S = \frac{2|U \cap V|}{|U| + |V|}$$

## V. SEGMENTACIÓN

En esta etapa, se busca separar el objeto de interés presente en la célula, como el núcleo y citoplasma para los siete tipos de células presentes en la base de datos de HERLEV.

En la Figura 3, se propone la solución para la segmentación de núcleo y citoplasma mediante segmentación semántica, tomando como base la estructura de UNET. Para la implementación del algoritmo, se utilizará la biblioteca Keras de Python y la estructura dispuesta en [8], que facilita la creación y entrenamiento de modelos de aprendizaje automático.

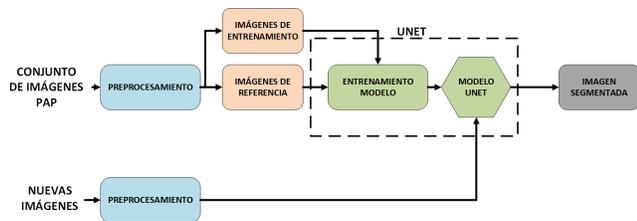


Figura 3: Estructura de segmentación usando UNET (fuente: autores).

### V-A. Creación del conjunto de entrenamiento y prueba

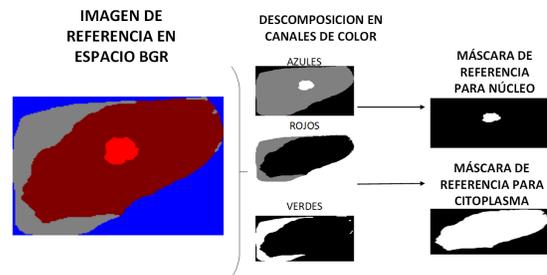
Se aumentaron los datos para las primeras tres clases normales, realizando una inversión de giro de  $90^\circ$  a cada imagen. Con lo anterior, el conjunto total cuenta ahora con 1159 imágenes, de las cuales el 20% se destinan para prueba y el 80% restante para entrenamiento UNET.

### V-B. Transformación de imágenes

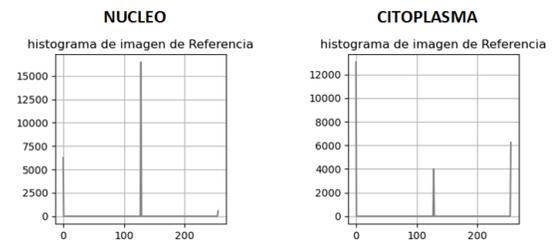
Para hacer uso del algoritmo UNET, es necesario que todas las imágenes de entrada y de referencia tengan las mismas dimensiones. Por lo tanto, cada imagen se redimensionó a  $128 \times 128$  píxeles.

Para la transformación de las imágenes de referencia del núcleo (Figura 4a), partiendo de una imagen en el espacio de color BGR (azul, verde, rojo), el plano que permite la distinción del núcleo es el canal de azules, que convertido a escala de

grises se comporta como la imagen central de la Figura 4a. Haciendo el análisis de la distribución de color, Figura 4b, el núcleo está representado en un tono claro con una intensidad aproximada de 255, por lo que un umbral de 240 en el canal de azules logra obtener una máscara binaria del núcleo.



(a) Máscara de núcleo y citoplasma.



(b) Distribución de color en escala de grises de planos verdes y azul.

Figura 4: Procesamiento de imagen de referencia del núcleo y citoplasma (fuente: autores).

Del mismo modo, el plano de rojos es el que representa mejor el citoplasma, representado por un tono oscuro que abarca la mayor parte de la imagen, con un valor de intensidad de cero. Nuevamente se realiza la umbralización, donde aquellos valores superiores a 100 (fondo y región en gris) se representarán en la máscara como fondo oscuro, mientras que aquellos valores iguales a cero se representarán en blanco, obteniendo así una máscara binaria para el citoplasma.

### V-C. Entrenamiento UNET

Dado que UNET es una CNN, es necesario realizar el entrenamiento para que aprenda a segmentar las regiones. El conjunto de datos de entrada corresponde a 927 imágenes con una resolución de  $128 \times 128$  píxeles, con 3 canales de color (BGR), formando así un tensor de cuatro dimensiones (927, 128, 128, 3).

El tensor con las imágenes de referencia tiene una forma de (927, 128, 128, 1), con 927 imágenes de tamaño  $128 \times 128$  y un solo canal de color con las máscaras binarias. Por otro lado, el tensor de prueba tiene una forma de (231, 128, 128, 3), con 231 imágenes de tamaño  $128 \times 128$  y tres canales de color (imágenes en formato BGR).

El modelo UNET entrenado se utilizará para segmentar las imágenes de prueba, lo que significa que se utilizará para identificar las regiones de interés en cada imagen (núcleo y citoplasma) mediante dos algoritmos. Entre los parámetros

configurables del entrenamiento, se tiene el optimizador *adam* para el ajuste de pesos de la red neuronal, con la función de pérdida *binary\_crossentropy* especialmente para clasificación binaria, con función de activación *sigmoid* a la salida de la red que indica la probabilidad de pertenencia a la clase positiva en un rango no lineal de 0 a 1 y finalmente el *accuracy* como medida de desempeño que indica la proporción de predicciones correctas (verdaderos positivos y verdaderos negativos) sobre el total de muestras de datos. El entrenamiento se lleva a cabo con 20 épocas y paquetes de validación del 10 %.

En la Figura 5, se puede observar en un tono claro la segmentación de referencia sobre la imagen real.

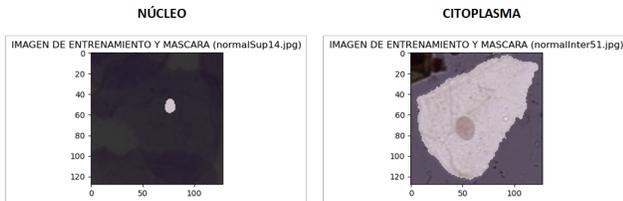


Figura 5: Segmentación de referencia en la imagen original (fuente: autores).

La Figura 6 muestra los resultados de entrenamiento para la segmentación del núcleo, donde se observa que el modelo logró segmentar correctamente las imágenes más del 95 % de las veces en relación al total de imágenes. Por otro lado, la

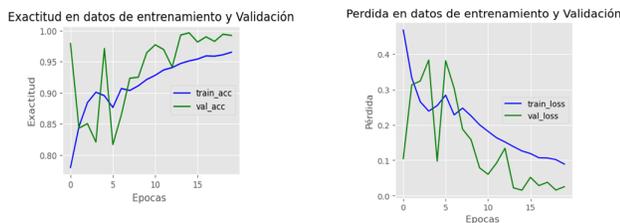


Figura 6: Métricas de entrenamiento UNET para núcleo (fuente: autores).

función de pérdida tanto en los datos de entrenamiento como en validación muestra una tendencia decreciente a lo largo de las iteraciones, lo que indica que el modelo aprendió a generalizar y a mejorar sus predicciones.

La Figura 7 muestra los resultados del entrenamiento para la segmentación del citoplasma.

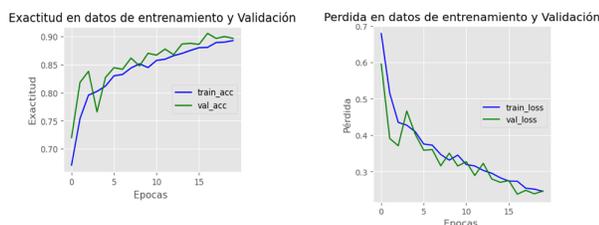


Figura 7: Métricas de entrenamiento UNET para citoplasma (fuente: autores).

La gráfica de exactitud muestra que tanto en los datos de entrenamiento como en validación acertó el 90 % de las veces en relación al total de imágenes, y se observa que no están muy distantes entre sí, lo que hace que el resultado de la exactitud sea aún más confiable. Del mismo modo, la pérdida a través de las épocas tiende a decrecer, alcanzando un valor de 0.24, indicando que el modelo se está ajustando adecuadamente a los datos. El modelo presenta una buena capacidad de generalización y debería ser capaz de hacer predicciones precisas en nuevos datos.

#### V-D. Segmentación en imágenes de Prueba

En la Figura 8 se muestran algunas imágenes que fueron segmentadas por los modelos UNET. La columna denominada *segmentación*, corresponde a la imagen de salida de la red neuronal. Cada punto blanco en esta columna indica la probabilidad de pertenencia a la clase positiva, mientras que los valores negros indican una probabilidad menor de pertenencia. En este caso, la imagen es en realidad una representación de probabilidades.

Por tanto, es necesario convertir la imagen de probabilidades, en otra imagen en escala de grises que varíe entre 0 y 255. Posteriormente, se realiza una umbralización utilizando el método de Otsu, que ajusta de manera óptima y automática el umbral según la maximización de varianza de los niveles de gris del objeto y fondo, separando de forma efectiva el objeto de interés. En la casos donde se presentan pseudonúcleos (múltiples núcleos), se elige aquel que se encuentre más cerca al centro de la imagen. Se aplica relleno de regiones en los casos en los que el núcleo/citoplasma presenta orificios en su interior.

También se muestra una imagen de bordes que encierra el objeto de interés y una vista general de la segmentación en superposición con la imagen de prueba.

## VI. EXTRACCIÓN Y SELECCIÓN DE CARACTERÍSTICAS

### ■ Características de forma

Se extrajeron características morfológicas las cuales relacionan la geometría, tamaño, forma y orientación del objeto en el espacio [9].

- Área
- Perímetro
- Circularidad
- Radio
- Eje mayor
- Eje menor
- Relación de aspecto
- Extensión
- Solidez
- Diámetro equivalente
- excentricidad
- Relación de aspecto con diámetros de Feret [10]
- Relación del área núcleo/citoplasma

### ■ Características de textura

Por otro lado, se utiliza la matriz de coocurrencia de grises GLCM para analizar las características de textura de una región de interés, teniendo en cuenta la frecuencia con la que aparece un nivel de gris con referencia a otro valor de gris dentro de una ventana determinada [11].

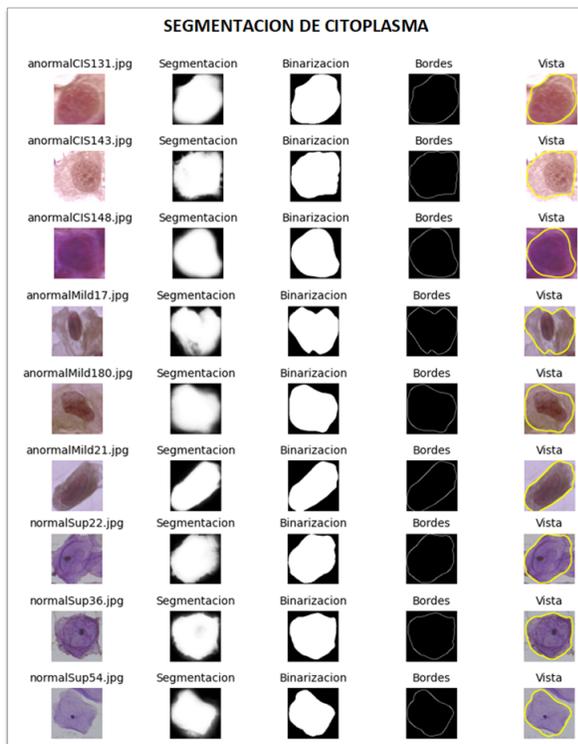
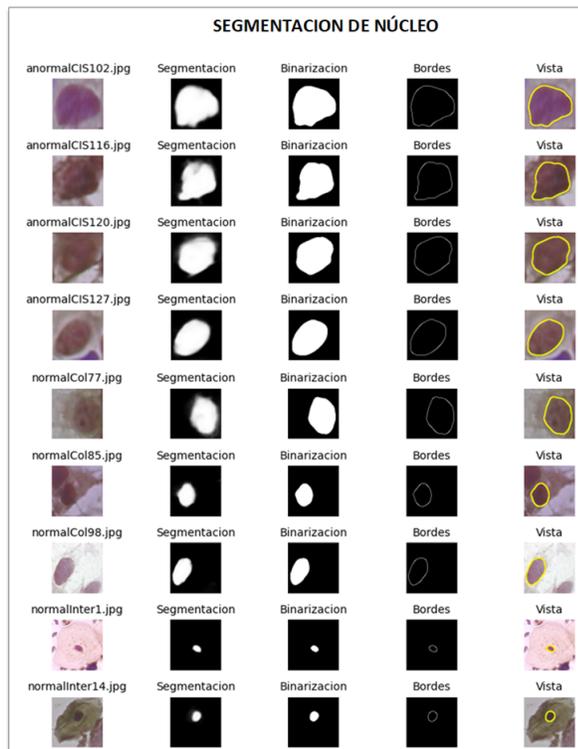


Figura 8: Segmentación en imágenes de prueba (fuente: autores).

- Correlación
- Constraste
- Disimilitud
- Homogeneidad
- Energía
- Entropía

■ Características de color

El brillo es una medida de la iluminación en una imagen y se calcula promediando la intensidad de los espacios de color RGB. Se utilizan coeficientes [0.299, 0.587, 0.114] para ponderar el promedio de intensidad de cada espacio de color, y luego se suman estos resultados. Esta fórmula proporciona una medida objetiva del brillo de la imagen, teniendo en cuenta la percepción humana del brillo [12].

$$Y = 0,299 * Ru + 0,587 * Gu + 0,114 * Bu \quad (1)$$

Donde  $R_u$ ,  $G_u$  y  $B_u$  son los promedios de intensidad de cada espacio de color.

Para la selección de características, se usaron las técnicas de FEATUREWIZ [13], PCA [14] y la combinación de ambas, para seleccionar las mejores características tanto de núcleo como de núcleo/citoplasma para tareas de clasificación binaria y de tres clases. FEATUREWIZ permite la selección de características óptimas de un conjunto de datos basado en dos técnicas, SULOV y XgBoost. SULOV selecciona pares de características más correlacionadas entre sí, y elimina aquella que brinda la menor información con respecto a la variable objetivo (clase). De estas variables, se hace un análisis exhaustivo con XgBoost que selecciona a la más importantes.

El método algebraico PCA permite sintetizar la información de un conjunto de datos reduciendo su dimensionalidad [14]. Consiste en combinaciones lineales de las variables originales cuyo objetivo es encontrar un conjunto de ejes ortogonales en los cuales la varianza de los datos sea máxima. El primer componente principal registra la mayor cantidad de variabilidad en los datos, seguido por el segundo componente PCA y así sucesivamente.

## VII. CLASIFICACIÓN

En esta sección, se realizarán pruebas de rendimiento en cuatro clasificadores: DT, SVM, KNN y ANN. El objetivo es analizar la capacidad de estos algoritmos para clasificar correctamente datos de prueba. El proceso de clasificación total se divide en dos problemas distintos, con dos y tres clases respectivamente. En el primer problema, se clasifican los datos en dos clases: normal y anormal. En el segundo problema, se clasifican los datos en tres clases: normal, LSIL y HSIL. Esta división permite analizar la sensibilidad del proceso de extracción de características.

En los experimentos, se realiza el balanceo de los datos tanto para 2 clases como para 3 y en cada caso se utilizará el 80% de los datos para entrenamiento y el 20% restante para probar la capacidad de los tres esquemas de clasificación en nuevas imágenes. Además, se comparará el rendimiento de los cuatro algoritmos con una línea base presentada en [2]. De esta forma, se podrá evaluar el rendimiento de la clasificación y determinar cuál de los clasificadores es el más adecuado para el conjunto de datos en cuestión.

Los algoritmos de clasificación DT, SVM, KNN y ANN, fueron implementados en Python utilizando la biblioteca Scikit-learn (sklearn). La biblioteca Scikit-learn es una herramienta ampliamente utilizada en el aprendizaje automático

debido a su facilidad de uso, eficiencia y flexibilidad en la implementación de modelos de aprendizaje automático. Se utilizó esta biblioteca para entrenar, validar y evaluar los modelos de clasificación.

### VII-A. Redes neuronales

Las ANN como modelos de clasificación son un poco más complejos que los algoritmos anteriores. En este estudio, se construye una ANN con 3 capas ocultas, con 100 neuronas para la primera capa oculta, 80 para la segunda, y un valor  $n$  de neuronas para la tercer capa oculta el cual se determinará de forma iterativa en un rango de 5 a 60 neuronas. Dicho valor se seleccionará de tal forma que maximice el *accuracy* en los datos de entrenamiento y prueba para los distintos espacios de características. De igual forma, se evaluará su rendimiento con validación cruzada, para luego probar los mejores hiperparámetros en el conjunto de prueba.

Para el entrenamiento se usaran 100 épocas, con funciones de activación ReLU que facilita el problema de clasificación no lineal, y *adam* como algoritmo de optimización para el ajuste de pesos sinápticos.

## VIII. EXPERIMENTOS Y RESULTADOS

### VIII-A. Resultados segmentación

En la Tabla II, se muestran los resultados obtenidos en la segmentación de las siete clases de células mediante UNET, evaluando las imágenes segmentadas con las de referencia. Finalmente se muestra el promedio de cada métrica por clase, e igualmente de forma general para todas las células junto con su desviación estándar (std). El método propuesto logró una precisión (P) de 87,6% y una exhaustividad (R) de 94,7% para segmentación de núcleo. Esto indica que el modelo identificó correctamente el 94,7% del núcleo presente en las imágenes con una seguridad del 87,6%.

Por otro lado, un índice de DICE de 90,2% significa que el modelo segmentó los núcleos con una alta similitud con respecto a las imágenes de referencia, lo que sugiere que la segmentación obtenida es muy cercana a la esperada.

Del mismo modo, la segmentación para el citoplasma arrojó valores de precisión (P) y exhaustividad (R) del 92,3% y 91,3% respectivamente, mientras que para el índice de DICE, en promedio, el 91,5% de la región segmentada se asemeja a la imagen de referencia.

Estos resultados son altamente significativos ya que indican que el modelo UNET es altamente preciso y confiable en la segmentación de las regiones objetivo de imágenes que presentan una célula.

En la Tabla III, se muestra una comparación de los resultados de segmentación de trabajos existentes con nuestra propuesta. El subíndice  $c$  y  $n$  corresponden a las métricas de citoplasma y núcleo respectivamente.

### VIII-B. Resultados clasificación

Utilizando el espacio de características de núcleo y citoplasma, las mejores técnicas de clasificación para dos clases y tres clases fueron ANN y KNN respectivamente. Para

Clase	U-net							
	Núcleo				Citoplasma			
	P	R	FM	DICE	P	R	FM	DICE
1	0.747	0.866	0.752	0.752	0.921	0.929	0.921	0.921
2	0.836	0.917	0.874	0.861	0.933	0.948	0.938	0.938
3	0.828	0.951	0.898	0.880	0.895	0.896	0.890	0.890
4	0.866	0.963	0.917	0.907	0.922	0.906	0.911	0.911
5	0.883	0.969	0.919	0.919	0.914	0.917	0.913	0.913
6	0.926	0.954	0.936	0.936	0.928	0.910	0.915	0.915
7	0.932	0.952	0.939	0.939	0.943	0.910	0.924	0.924
Promedio	0.876	0.947	0.907	0.902	0.923	0.913	0.915	0.915
std	0.144	0.105	0.096	0.117	0.069	0.059	0.044	0.044

Tabla II: Evaluación segmentación (fuente: autores).

Trabajos Existentes	Dataset	Segmentación Citoplasma			Segmentación Núcleo		
		Pc	Re	DICE c	Pn	Rn	DICE n
A. M. Braga [15]	ISBI'14	-	-	-	0,93	0,93	0,92 ± 0,04
	EDF	-	-	-	0,95	0,90	0,92 ± 0,05
	Herlev	-	-	-	0,93	0,95	0,93 ± 0,05
P. Wang [16]	Privado	-	-	-	-	0,94	-
T. Wang [17]	ISBI'15	-	-	-	0,95	0,91	0,93 ± 0,04
H. Lee. [18]	ISBI'14	-	-	0,897 ± 0,075	-	-	-
	ISBI'15	-	-	0,879 ± 0,087	-	-	-
L. Zhao. [19]	Privado	-	-	0,71 ± 0,08	-	-	0,72 ± 0,09
	Herlev	-	-	0,82 ± 0,11	-	-	0,93 ± 0,03
A. Tareef. [20]	ISBI'14	-	-	0,89 ± 0,07	0,99	0,94	0,93 ± 0,04
W. Willian [21]	Herlev	-	-	0,948	-	-	0,973
	-	-	-	-	-	-	-
Diniz. [22]	ISBI'14	-	-	-	0,98	0,99	-
Adhikary. [2]	DIC	0,91	0,84	-	0,81	0,84	-
	Herlev	0,90	0,83	-	0,86	0,91	-
<b>Nuestra Propuesta</b>	Herlev	<b>0,92</b>	<b>0,91</b>	<b>0,915 ± 0,044</b>	<b>0,88</b>	<b>0,95</b>	<b>0,902 ± 0,117</b>

Tabla III: Rendimiento de segmentación (fuente: autores).

clasificación binaria, el modelo de mejor desempeño fue ANN con características de núcleo y citoplasma seleccionadas con FEATUREWIZ:

- nRADIO
- nENERGIA3
- RADIO
- BRILLO
- DISIMILITUD2
- nSOLIDEZ
- CORRELACION2
- ENTROPIA
- nRELACION
- nCORRELACION3
- DIAMEQUI
- ASM2
- SOLIDEZ

Para clasificación de tres clases, el modelo de mejor desem-

peño fue ANN usando características de núcleo y citoplasma. En este caso, el espacio que permite el mejor rendimiento es PCA en combinación con FEATUREWIZ. Es decir, primero se extrajeron las características más importantes con FEATU-REWIZ, y con ellas se formó un nuevo espacio mediante el método PCA que finalmente se usó para entrenamiento. A continuación se muestran las características de núcleo/citoplasma seleccionadas por FEATUREWIZ:

- NC
- nRADIO
- nCONTRASTE3
- PERIMETRO
- BRILLO
- nEXTENSION
- CONTRASTE3
- ASM3

Las características que empiezan con *n* corresponden al núcleo. Las características de textura que terminan en 2 provienen de una GLCM con orientación de 45° y distancia interpíxel de 20. Aquellas que terminan en 3 se extraen de una GLCM con orientación de 90° y distancia de 30 píxeles. Las características de textura sin número final se derivan de una GLCM de 0° con distancia interpíxel de 10.

En la Tabla IV, se especifican los resultados de desempeño de los mejores modelos de clasificación para 2 y 3 clases.

CLASE	ESPACIO	CLASIFICADOR	E	P	R	HIPERPARAMETRO
BINARIA	FEATUREWIZ Y PCA	ANN	98.57	98.50	98.50	n=54
TRES CLASES	FEATUREWIZ	KNN	89.18	90.67	88.86	K=20

Tabla IV: Comparación de los clasificadores ANN y KNN (fuente: autores).

Para el caso binario con ANN, se logró una exactitud de 98.57% en 281 imágenes de prueba con una precisión promedio de 98.50% por clase.

Para tres clases, el modelo KNN logró una exactitud de 89.18% en 305 imágenes de prueba con una precisión promedio de 90.67% por clase. También se observó gran capacidad de detección para las clases normal y LSIL. Sin embargo, se observó que el modelo confundió el 25% de células pertenecientes a la tercera clase HSIL con la clase LSIL.

En la Tabla V, se presenta una comparación de los resultados de clasificación de nuestra propuesta frente a los trabajos encontrados en la literatura. Nuestra propuesta logra valores de exactitud de 0.98 y 0.89 para la clasificación de dos y tres clases, respectivamente. Estos valores son comparables con los de la literatura, y demuestran una precisión y exhaustividad considerables para una clasificación efectiva de los datos, observando que en algunos casos, los resultados de nuestra propuesta son equivalentes o incluso superiores, especialmente en tareas de clasificación binaria con los datos de HERLEV.

## IX. CONCLUSIONES

Durante este estudio se desarrolló un enfoque de segmentación semántica de núcleo y citoplasma celular usando

Trabajos Existentes	Dataset	Clases	Rendimiento			
			exactitud	P	R	AUC
J. Martínez-Más [4]	Privado	2	0,88	-	0,92	-
A. M. Braga [15]	Herlev	2	0,89	0,91	0,89	0,93
		3	0,77	0,85	0,85	-
P. Wang [16]	Privado	2	0,93	-	0,95	-
		2	0,97	-	0,96	-
		3	0,96	-	0,92	-
C. Rajarao [23]	Herlev	-	0,99	-	0,98	-
		-	-	-	-	-
A. D. Jia [24]	Herlev	2	0,99	-	98,9	-
		7	0,93	-	0,93	-
	Privado	2	0,94	-	0,93	-
L. Zhang [25]	Herlev, HEMLBC	7	0,98	-	0,98	0,99
		-	-	-	-	-
W. Willian [21]	Herlev	2	0,98	0,99	0,99	0,99
	Dataset 2	2	0,97	0,97	0,98	-
	Dataset 3	2	0,95	0,90	1	-
Adhikary. [2]	DIC	2	0,97	-	-	0,99
		3	0,92	-	-	0,92
	Herlev	2	0,94	-	-	0,97
		3	0,89	-	-	0,91
Nuestra Propuesta	Herlev	2	<b>0,98</b>	<b>0,98</b>	<b>0,98</b>	<b>0,99</b>
		3	<b>0,89</b>	<b>0,91</b>	<b>0,89</b>	<b>0,99</b>

Tabla V: Comparación de los resultados de clasificación con las diferentes propuestas para PAP (fuente: autores).

la arquitectura UNET, el cual logró segmentar las regiones, enfrentándose con gran éxito a problemas superposición con otras células, obteniendo un índice de similitud DICE del 90.15% para el núcleo y de 91.51% para el citoplasma con respecto a las imágenes de referencia. Estos resultados resaltan la eficacia de UNET como una herramienta poderosa para la segmentación de células PAP. Este estudio representa una investigación pionera en el uso de la arquitectura UNET para la segmentación de células PAP, y los resultados obtenidos respaldan su potencial para futuros avances en esta área.

La combinación de técnicas de selección de características y algoritmos de clasificación supervisada demostraron ser altamente efectivas para identificar y clasificar con precisión las células PAP en categorías patológicas, tanto en 2 clases (normal, anormal) como en 3 clases (normal, LSIL, HSIL). El análisis de componentes principales (PCA) y la herramienta FEATUREWIZ, así como su combinación, permitieron

establecer un conjunto de características cruciales para la clasificación precisa del grado patológico de las células PAP en las clases mencionadas

La inclusión de características del citoplasma en la clasificación de células de frotis cervical resulta crucial para obtener una alta precisión en comparación con el uso exclusivo de características nucleares. Sin embargo, también se observó que solo utilizando características del núcleo se lograron resultados de clasificación satisfactorios. A pesar de esto, es importante considerar tanto las características nucleares como las del citoplasma para obtener una clasificación más precisa y detallada de las células. La combinación de ambas características contribuye a una caracterización más completa y a una mejor comprensión del grado patológico de las células.

## X. AGRADECIMIENTOS

Agradecemos en primer lugar a la Universidad del Cauca y docentes que hacen parte de esta investigación por su participación activa y orientación en cada una de las actividades planteadas. También a la Facultad de Electrónica y Telecomunicaciones, por brindar los espacios y recursos educativos.

## REFERENCIAS

- [1] W. William, A. Ware, A. H. Basaza-Ejiri, and J. Obungoloch, "A review of image analysis and machine learning techniques for automated cervical cancer screening from pap-smear images," *Computer Methods and Programs in Biomedicine*, vol. 164, pp. 15–22, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0169260717307459>
- [2] S. Adhikary, S. Seth, S. Das, T. K. Naskar, A. Barui, and S. P. Maity, "Feature assisted cervical cancer screening through DIC cell images," *Biocybernetics and Biomedical Engineering*, vol. 41, no. 3, pp. 1162–1181, jul 2021.
- [3] B. Sokouti, S. Haghypour, and A. D. Tabrizi, "A framework for diagnosing cervical cancer disease based on feedforward mlp neural network and thinprep histopathological cell image features," *Neural Computing and Applications*, vol. 24, no. 1, pp. 221–232, 2014.
- [4] J. Martínez-Más, A. Bueno-Crespo, R. Martínez-España, M. Remezal-Solano, A. Ortiz-González, S. Ortiz-Reina, and J. P. Martínez-Cendán, "Classifying Papanicolaou cervical smears through a cell merger approach by deep learning technique," *Expert Systems with Applications*, vol. 160, p. 113707, dec 2020.
- [5] J. Jantzen, J. Norup, G. Dounias, and B. Bjerregaard, "Pap-smear benchmark data for pattern classification," *Nature inspired Smart Information Systems (NiSIS 2005)*, pp. 1–9, 2005.
- [6] R. H. Rodríguez Espinosa, "Módulo para la validación de la segmentación en imágenes de opacidad de la cápsula posterior provenientes del pentacam," B.S. thesis, Universidad de las Ciencias Informáticas. Facultad 3. AIRI, Artificial . . . , 2017.
- [7] D. Marín-Soto, "Segmentación de células mediante técnicas de procesamiento digital de imágenes para el rastreo de células cancerosas," 2018. [Online]. Available: <https://repositoriotec.tec.ac.cr/handle/2238/10383>
- [8] D. Revelo Luna, "Semantic segmentation basic u-net," <https://github.com/DavidReveloLuna/Semantic-Segmentation-Basic-U-Net>, 2023.
- [9] N. A. Ramírez Pérez *et al.*, "Modelo de clasificación automática de imágenes de resonancia magnética para el diagnóstico del cáncer de próstata." 2020.
- [10] A. G. Marcos, F. J. M. de Pisón Ascacibar, A. V. P. Espinoza, F. A. Elías, M. C. Limas, J. B. O. Meré, E. P. V. González, A. G. Marcos, F. J. M. de Pisón Ascacibar, A. V. P. Espinoza, F. A. Elías, M. C. Limas, J. B. O. Meré, and E. P. V. González, "Técnicas y algoritmos básicos de visión artificial," *Técnicas y Algoritmos Básicos de Visión Artificial*, 2006.
- [11] M. Presutti, "La matriz de co-ocurrencia en la clasificación multiespectral: tutorial para la enseñanza de medidas texturales en cursos de grado universitario," *4ª Jornada de Educação em Sensoriamento Remoto no Âmbito do Mercosul*, pp. 1–9, 2004.
- [12] M. C. J. Jaime and E. Elizondo, "Fundamentos de procesamiento de imágenes instituto tecnológico de ciudad madero," 2002.
- [13] S. Ramírez-Gallego, S. García, J. M. Luengo, JuliDinizand Benítez, and F. Herrera, "Featurewiz: Dinizautomated feature engineering tool," *Information Sciences*, vol. 45, pp. 1–15, 2018.
- [14] M. S. Park and J. Y. Choi, "Theoretical analysis on feature extraction capability of class-augmented pca," *Pattern Recognition*, vol. 42, no. 11, pp. 2353–2362, 2009. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320309001526>
- [15] A. M. Braga, R. C. Marques, F. N. Medeiros, J. F. Neto, A. G. Bianchi, C. M. Carneiro, and D. M. Ushizima, "Hierarchical median narrow band for level set segmentation of cervical cell nuclei," *Measurement*, vol. 176, p. 109232, 5 2021.
- [16] P. Wang, L. Wang, Y. Li, Q. Song, S. Lv, and X. Hu, "Automatic cell nuclei segmentation and classification of cervical Pap smear images," *Biomedical Signal Processing and Control*, pp. 93–103.
- [17] T. Wang, J. Huang, D. Zheng, and Y. He, "Nucleus Segmentation of Cervical Cytology Images Based on Depth Information," *IEEE Access*, vol. 8, pp. 75 846–75 859, 2020.
- [18] H. Lee and J. Kim, "Segmentation of overlapping cervical cells in microscopic images with superpixel partitioning and cell-wise contour refinement," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2016, pp. 63–69.
- [19] L. Zhao, K. Li, M. Wang, J. Yin, E. Zhu, C. Wu, S. Wang, and C. Zhu, "Automatic cytoplasm and nuclei segmentation for color cervical smear image using an efficient gap-search mrf," *Computers in Biology and Medicine*, vol. 71, pp. 46–56, 4 2016.
- [20] A. Tareef, Y. Song, W. Cai, H. Huang, H. Chang, Y. Wang, M. Fulham, D. Feng, and M. Chen, "Automatic segmentation of overlapping cervical smear cells based on local distinctive features and guided shape deformation," *Neurocomputing*, vol. 221, pp. 94–107, 1 2017.
- [21] W. William, A. Ware, A. H. Basaza-Ejiri, and J. Obungoloch, "A pap-smear analysis tool (pat) for detection of cervical cancer from pap-smear images," *BioMedical Engineering Online*, vol. 18, pp. 1–22, 2 2019.
- [22] D. N. Diniz, R. F. Vitor, A. G. C. Bianchi, S. Delabrada, C. M. Carneiro, D. M. Ushizima, F. N. S. de Medeiros, and M. J. F. Souza, "An ensemble method for nuclei detection of overlapping cervical cells," *Expert Systems with Applications*, vol. 185, p. 115642, dec 2021.
- [23] C. Rajarao and R. P. Singh, "Improved normalized graph cut with generalized data for enhanced segmentation in cervical cancer detection," *Evolutionary Intelligence*, vol. 13, pp. 3–8, 3.
- [24] A. D. Jia, B. Z. Li, and C. C. Zhang, "Detection of cervical cancer cells based on strong feature cnn-svm network," *Neurocomputing*, vol. 411, pp. 112–127, 10 2020.
- [25] L. Zhang, L. Lu, I. Noguez, R. M. Summers, S. Liu, and J. Yao, "DeepPap: deep convolutional networks for cervical cell classification," *IEEE journal of biomedical and health informatics*, vol. 21, no. 6, pp. 1633–1643, 2017.