



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA**

**FACULTAD DE CIENCIAS DE LA ELECTRÓNICA  
MAESTRÍA EN INGENIERÍA ELECTRÓNICA, OPCIÓN  
INSTRUMENTACIÓN ELECTRÓNICA**

**Tesis para obtener el grado de  
MAESTRO EN INGENIERÍA ELECTRÓNICA**

---

**"Identificación del ciclo estral"**

---

**Presenta:**

**Leonardo Delgado Toral\***

**Asesores:**

**Dr. ALDRIN BARRETO FLORES**

**Dra. VERÓNICA EDITH BAUTISTA LÓPEZ**

\*Becario CONACYT

Puebla, Pue., Diciembre 2017

# Agradecimientos

Agradezco al Consejo de Ciencia y Tecnología (CONACYT) y a la Benemérita Universidad Autónoma de Puebla por su apoyo y patrocinio para la realización de este proyecto de tesis.

Agradezco a mis asesores, al personal del Bioterio Claude Bernard especialmente al M.V.Z Francisco Ramos Collazo director del Bioterio y al M.V.Z. César Olvera Pérez quienes me proporcionaron la capacitación para la interpretación de las imágenes utilizadas además de etiquetar la base de datos utilizada en este trabajo, a la Dra. Rosalina Reyes Luna de la Facultad de Ciencias Biológicas por su asesoría en la parte biológica. Al Dr. Juan Humberto Sossa Azuela y a su equipo de trabajo especialmente al Dr. Erik Zamora y al estudiante de doctorado Gerardo Hernández sus retroalimentaciones me permitieron mejora mi trabajo. A mi familia especialmente a mi madre Leticia Delgado Toral. A mis amigos y compañeros de maestría especialmente a Arturo Cocomo y Jorge Mateo.

¿Por qué todos los caballos son iguales, Sofía? A lo mejor piensas que no lo son en absoluto. Pero hay algo que todos los caballos tienen en común, algo que hace que nunca tengamos problemas para distinguir un caballo de cualquier otro animal. El caballo individual «fluye», claro está. Puede ser viejo, cojo, y, con el tiempo, se pondrá enfermo y morirá. Pero el «molde de caballo» es eterno e inmutable.

**El mundo de Sofía, Jostein Gaarder**

---

# Resumen

Este trabajo presenta la clasificación autónoma del ciclo estral a través de la visión por computadora. El ciclo estral es el ciclo reproductor de la rata y consta de cuatro etapas: Estro, Proestro, Diestro y Metaestro. El ciclo dura de 4 a 5 días lo cual lo convierte en un modelo ideal para estudiar los cambios asociados a este. Actualmente la clasificación del ciclo estral es un proceso manual donde el método más eficiente es la observación de la citología de las células. La clasificación del ciclo puede abordarse desde 2 puntos de vista. El primero es cuando el ciclo estral es utilizado para el control de población de los roedores dentro de un bioterio. En este caso es necesario separar las etapas en 2 clases considerando la receptividad a la reproducción que se presenta en cada etapa. El segundo punto de vista es cuando el ciclo estral es utilizado dentro de una investigación donde el objetivo es asociar la reacción de fármacos o condiciones inducidas que el roedor presenta en cada etapa.

Para lograr la clasificación por medio de la visión por computadora se diseñó un mecanismo para la toma de las imágenes de las células. Se diseñaron 3 versiones de clasificadores: utilizando el histograma de la imagen como características, clasificándolas con el algoritmo de los K-vecinos más cercanos (K-Nearest Neighbors (K-NN) K-Nearest Neighbors), utilizando características extraídas con un modelo de red neuronal pulso acoplada (PCNN Pulse Coupled Neuronal Network) clasificándolas con una máquina vector soporte (Support Vector Machine (SVM) support vector machine) lineal y utilizando características de textura - forma clasificándolas con 4 SVM (lineal, la propuesta por Cramer y Singer, de base radial y polinomial) y una red neuronal perceptrón multi capa (Multilayer Perceptron (MLP) Multilayer Perceptron). El mejor desempeño fue obtenido con la red MLP con la cual se desarrolló una aplicación capaz de determinar cada etapa con una exactitud de 87%. Los algoritmos se implementaron en la etapa inicial (versiones: 1 y 2) utilizando la plataforma Matlab 2016b. La versión 3 fue implementada en el lenguaje de programación Python 3. El entorno gráfico se desarrolló en el lenguaje de programación Java utilizando la librería windowbuilder.

---

# Índice general

<b>1. Marco referencial</b>	<b>3</b>
<b>2. Marco teórico</b>	<b>10</b>
2.1. El ciclo estral de la rata . . . . .	10
2.1.1. Observación al microscopio . . . . .	10
2.2. Procesamiento digital de imágenes . . . . .	11
2.2.1. Representación de imágenes . . . . .	12
2.2.2. Umbralización . . . . .	13
2.2.3. K-Means . . . . .	15
2.2.4. Operaciones basadas en objetos . . . . .	17
2.2.5. Rasgos . . . . .	18
2.3. Inteligencia artificial . . . . .	22
2.3.1. Aprendizaje basado en instancia . . . . .	22
2.3.2. Perceptrón . . . . .	24
2.3.3. Descenso de gradiente . . . . .	25
2.3.4. Perceptrón multicapa . . . . .	27
2.3.5. Red neuronal pulso acoplada . . . . .	27
2.3.6. Máquina de soporte vectorial . . . . .	29
2.4. Métricas de desempeño . . . . .	31
2.4.1. Micro y macro $F_1$ . . . . .	31
2.4.2. Validación cruzada . . . . .	32
<b>3. Desarrollo</b>	<b>33</b>
3.1. Medio de adquisición de imágenes . . . . .	34
3.2. Versiones . . . . .	36
3.2.1. Versión 1: Rasgos del histograma clasificados utilizando K-NN . . . . .	38
3.2.2. Versión 2: Rasgos PCNN . . . . .	38
3.2.3. Versión 3: Rasgos de textura - forma . . . . .	41

---

3.3. Aplicación . . . . .	46
<b>4. Resultados</b>	<b>47</b>
4.1. Versiones 1,2 . . . . .	47
4.2. Versión 3: Rasgos de textura - forma . . . . .	51
4.2.1. SVM lineal . . . . .	51
4.2.2. SVM propuesta por Cramer y Singer . . . . .	52
4.2.3. SVM de base radial . . . . .	54
4.2.4. SVM polinomial . . . . .	55
4.2.5. MLP . . . . .	56
4.3. Comparación . . . . .	58
4.4. Aplicación . . . . .	59
4.5. Conclusiones y trabajo futuro. . . . .	62
<b>Bibliografía</b>	<b>63</b>
<b>A. Codigos</b>	<b>68</b>
A.1. Otsu . . . . .	68
A.2. Media Arigmetica . . . . .	70
A.3. Entropía . . . . .	71
A.4. Entropía Cruzada . . . . .	72
A.5. K-Means . . . . .	73
A.6. K-MeansRGB . . . . .	74
A.7. K-MeansP . . . . .	76
A.8. PCNN . . . . .	77
<b>B. Imágenes segmentadas</b>	<b>78</b>
B.1. Media Aritmética . . . . .	79
B.2. Máxima Entropía . . . . .	82
B.3. Entropía Cruzada . . . . .	85
B.4. Otsu . . . . .	88
B.5. K-Means . . . . .	91
<b>C. Producción académica</b>	<b>94</b>
<b>D. Acronimos</b>	<b>100</b>

---

# Índice de figuras

1.1.	Etapas del ciclo estral. Imágenes proporcionadas por el bioterio Claude Bernard	4
1.2.	De izquierda a derecha umbralado por Otsu, umbralado por algoritmo propuesto, umbralado por C-V, umbralado por algoritmo propuesto [1]. . . . .	4
1.3.	De izquierda a derecha las primeras dos imágenes muestran los resultados del algoritmo aplicado a las imágenes sintéticas mientras las últimas dos muestran lo que se obtiene al aplicarlo al caso real [2]. . . . .	5
1.4.	De izquierda a derecha células sin traslape, células con doble traslape y con múltiple traslape[3]. . . . .	5
1.5.	a)Posibles núcleos, c)Detección de núcleos reales [4]. . . . .	6
1.6.	Segmentación usando super-pixeles con corrección de células encimadas [5]. .	7
1.7.	Ejemplo de la segmentación en muestras: a) fuertemente encimadas, b) bajo contraste y tinción no uniforme, c) pico de brillo en rojo [6]. . . . .	7
1.8.	Imagen utilizada en el dataset de [7]. . . . .	8
1.9.	Imagen original e imagen segmentada por [8]. . . . .	8
2.1.	Matrices RGB . . . . .	12
2.2.	Representación del espacio de color CIELAB. Imagen tomada de [9]. . . . .	13
2.3.	Matriz de imagen binaria . . . . .	17
2.4.	Matriz de imagen binaria con 2 objetos . . . . .	17
2.5.	a) 5x5 imagen 4 niveles de gris, b) Forma general para una matriz GLCM para imágenes con valores de gris de 0-3, c)-f) Cálculo de la GLCM para: $(d,\theta)=(1,0),(1,45),(1,90),(1,135)$ . . . . .	20
2.6.	K-NN Con 2 Clases. . . . .	23
2.7.	Distancias entre el conjunto de entrenamiento y la entrada. . . . .	23
2.8.	Perceptrón: $w_0$ es el valor de umbral $x_i$ es la entrada y $w_i$ son los pesos sinápticos.	25
2.9.	Espacio de hipótesis [10] . . . . .	26
2.10.	Neurona de una red neuronal pulso acoplada tomada de [11]. . . . .	29

---

3.1. Diagrama general para clasificación del ciclo estral. . . . .	33
3.2. Modelo del sistema con la cámara de la Raspberry . . . . .	34
3.3. Implementación del sistema con la cámara de la Raspberry . . . . .	34
3.4. Imagen obtenida con la cámara de la Raspberry . . . . .	35
3.5. Sistema de adquisición con cámara c120 . . . . .	35
3.6. Imagen sin corrección de iluminación . . . . .	36
3.7. Imagen con corrección de iluminación . . . . .	36
3.8. Segmentación de la etapa Diestro a partir del espacio de color CIELAB y la escala de grises. . . . .	39
3.9. Imagen normalizada adecuada para utilizarla dentro de la red PCNN. . . . .	40
3.10. Salida de la red PCNN para las iteraciones 2, 5, 7 y 10. . . . .	40
3.11. Imágenes pertenecientes al Dataset. Las imágenes fueron proporcionadas por el bioterio Claude Bernard. . . . .	41
3.12. Imágenes en los espacios de color CIELAB y escala de grises para la etapa Metaestro después de aplicar el filtro promedio . . . . .	42
3.13. Imágenes segmentadas en los espacios de color CIELAB y escala de grises para la etapa Metaestro. . . . .	43
4.1. Tiempo de ejecución en segundos vs tamaño de imagen . . . . .	47
4.2. Series de tiempo para la etapa Proestro. . . . .	48
4.3. Series de tiempo para la etapa Estro. . . . .	49
4.4. Series de tiempo para la etapa Metaestro. . . . .	49
4.5. Series de tiempo para la etapa Diestro. . . . .	50
4.6. Perceptron multicapa: 105 características, 22 neuronas de entrada, 44 neuronas en la capa oculta y 4 clasificadores softmax en la capa de salida. La capa oculta y la capa de entrada usan la función de activación relu. . . . .	56
4.7. Perceptrón multicapa: 105 características, 1 neurona de entrada, 135 neuronas en la primera capa oculta, 116 neuronas en la segunda capa oculta, 139 neu- ronas en la tercera capa oculta y 2 clasificadores softmax. Las capas ocultas y la capa de entrada usan la función de activación relu. . . . .	58
4.8. Aplicación: Menú Abrir. . . . .	60
4.9. Aplicación: Menú Guardar. . . . .	60
4.10. Aplicación: Resultado de procesar una imagen. . . . .	61
B.1. Imágenes segmentadas por Media Aritmética partiendo de la escala de grises, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro. . . . .	79

---

B.2. Imágenes segmentadas por Media Aritmética partiendo del canal A del espacio CIELAB, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro. . . . .	80
B.3. Imágenes segmentadas por Máxima Entropía partiendo de la escala de grises, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro. . . . .	82
B.4. Imágenes segmentadas por Máxima Entropía partiendo del canal A del espacio CIELAB, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro. . . . .	83
B.5. Imágenes segmentadas por Entropía Cruzada partiendo de la escala de grises, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro. . . . .	85
B.6. Imágenes segmentadas por Entropía Cruzada partiendo del canal A del espacio CIELAB, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro. . . . .	86
B.7. Imágenes segmentadas por Otsu partiendo de la escala de grises, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro. . . . .	88
B.8. Imágenes segmentadas por Otsu partiendo del canal A del espacio CIELAB, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro. . . . .	89
B.9. Imágenes segmentadas por K-Means partiendo de la escala de grises, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro. . . . .	91
B.10. Imágenes segmentadas por K-Means partiendo del canal A del espacio CIELAB, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro. . . . .	92
B.11. Imágenes segmentadas por K-Means partiendo del espacio RGB, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro. . . . .	93

---

# Índice de tablas

2.1. Matriz de confusión. C representa la salida del clasificador entrenado. . . . .	31
3.1. Tiempos de ejecución para una sola imagen en escala de grises en el caso de K-means se consideran una k de 5 y solo los últimos tres conjuntos son tomados, en el caso de K-means en RGB se considera una k de 5 y solo se toman los 4 primeros conjuntos. . . . .	42
3.2. Valores de la red MLP . . . . .	45
4.1. Comparativa de los resultados en diferentes resoluciones. . . . .	48
4.2. Distribución de los conjuntos de validación que dieron el mejor resultado para el caso de las 2 clases para los 5 clasificadores. . . . .	51
4.3. Matriz de confusión con el mejor resultado para las 4 etapas del ciclo estral utilizando la SVM lineal . . . . .	51
4.4. Matriz de confusión promedio para las 4 etapas del ciclo estral utilizando la SVM lineal . . . . .	52
4.5. Matriz de confusión con el mejor resultado para el caso de 2 clases utilizando la SVM lineal . . . . .	52
4.6. Matriz de confusión promedio para el caso de 2 clases utilizando la SVM lineal	52
4.7. Matriz de confusión con el mejor resultado para las 4 etapas del ciclo estral utilizando la SVM propuesta por Cramer y Singer . . . . .	53
4.8. Matriz de confusión promedio para las 4 etapas del ciclo estral utilizando la SVM propuesta por Cramer y Singer . . . . .	53
4.9. Matriz de confusión con el mejor resultado para el caso de 2 clases utilizando la SVM propuesta por Cramer y Singer . . . . .	53
4.10. Matriz de confusión promedio para el caso de 2 clases utilizando la SVM propuesta por Cramer y Singer . . . . .	53
4.11. Matriz de confusión con el mejor resultado para las 4 etapas del ciclo estral utilizando la SVM de base radial . . . . .	54

---

4.12. Matriz de confusión promedio para las 4 etapas del ciclo estral utilizando la SVM de base radial . . . . .	54
4.13. Matriz de confusión con el mejor resultado para el caso de 2 clases utilizando la SVM de base radial . . . . .	54
4.14. Matriz de confusión promedio para el caso de 2 clases utilizando la SVM de base radial . . . . .	55
4.15. Matriz de confusión con el mejor resultado para las 4 etapas del ciclo estral utilizando la SVM polinomial . . . . .	55
4.16. Matriz de confusión promedio para las 4 etapas del ciclo estral utilizando la SVM polinomial . . . . .	55
4.17. Matriz de confusión con el mejor resultado para el caso de 2 clases utilizando la SVM polinomial . . . . .	56
4.18. Matriz de confusión promedio para el caso de 2 clases utilizando la SVM polinomial . . . . .	56
4.19. Matriz de confusión con el mejor resultado para las 4 etapas del ciclo estral utilizando la red MLP . . . . .	57
4.20. Matriz de confusión promedio para las 4 etapas del ciclo estral utilizando la red MLP . . . . .	57
4.21. Matriz de confusión con el mejor resultado para el caso de 2 clases utilizando la red MLP . . . . .	57
4.22. Matriz de confusión promedio para el caso de 2 clases utilizando la red MLP	57
4.23. Métricas de validación y desempeño para las cuatro etapas del ciclo estral . .	59
4.24. Micro $F_1$ para el caso de 2 clases . . . . .	59

---

# Introducción

El ciclo estral es el ciclo reproductor de la rata consta de cuatro etapas: Estro, Proestro, Diestro y Metaestro. La importancia de este ciclo en las ratas radica en su uso para determinar el efecto que este tiene en las reacciones de fármacos o condiciones inducidas al roedor. Lo cual permite probar de manera rápida nuevos tratamientos a enfermedades o condiciones. La rapidez de las pruebas se debe al corto periodo del ciclo en los roedores el cual dura entre 4-5 días. Algunos trabajos de esta índole son:

- Brittany [12] partiendo de la premisa de que las mujeres tienden a ser inducidas con mayor facilidad dentro del consumo del alcohol conduce un experimento en 2 sepas de rata Long Evans y Winstar. Sus resultados muestran que el ciclo estral no es un factor que influya de manera directa en el consumo.
- Sang Yoon Lee [13] toma como fundamento las propiedades de las enzimas antioxidante de prevenir enfermedades, encuentra una relación entre estas y el ciclo estral.
- Matthew G [14] muestra la relación entre los niveles de estradiol E2 asociados al ciclo estral con la inhibición latente.
- Ricardo Cossio [15] con sus resultados sugieren una modulación de la respuesta al miedo asociada a los niveles de estrógeno presentes en la etapa metaestro del ciclo estral.

Para la determinación del ciclo estral se han propuesto diversas metodologías de las cuáles la que presenta el mejor desempeño es la observación de la citología de las células. Sin embargo, esta metodología presenta la desventaja de requerir personal capacitado que pueda clasificar la etapa a partir de la observación de las células. Esto conlleva a que este método presente problemas de identificación de las etapas en personas que están iniciando en esta área. Nuestra propuesta es el desarrollo de un software libre que permita la identificación de cada etapa del ciclo a través de imágenes tomadas a las células.

Si bien la clasificación de células no es un trabajo nuevo en el área de la visión por computadora; si lo es la clasificación del ciclo estral. Las células del ciclo estral comparten similitudes con las células utilizadas en las pruebas de Papanicolau (PAP), sin embargo el criterio de clasificación es diferente puesto que las células PAP utilizan criterios basados en la forma del núcleo o citoplasma principalmente de células aisladas, el ciclo estral utiliza el criterio de la cantidad de células de cada tipo presentes en la muestra.

Para alcanzar el objetivo principal *Clasificación de células del ciclo estral* es necesario cubrir ciertos objetivos particulares:

- Proponer un sistema que permita realizar la captura de imágenes digitales.
- Diseñar un algoritmo para determinar el ciclo estral en las muestras basado en análisis de características de las células, así como análisis de la textura y el color.
- Generación de una interface con la base de imágenes y de datos.

La validez del trabajo radica en tres ejes:

- Es el primer trabajo que aborda la clasificación del ciclo estral de manera autónoma a través de la observación de la citología de las células.
- La realización del trabajo conllevó al desarrollo de una base de imágenes la cual puede ser empleada en futuras investigaciones.
- La aplicación en sí misma es una herramienta que puede ser utilizada dentro de bioterios para el control de población.

El resto del trabajo está organizado de la siguiente manera. El capítulo 1 presenta los trabajos previos entorno a la clasificación del ciclo estral asimismo presenta trabajos donde el objetivo es la segmentación y clasificación de células PAP a través del procesamiento de imágenes o la visión por computadora. El capítulo 2 introduce los algoritmos utilizados posteriormente. El capítulo 3 presenta las propuestas que se implementaron, así como los resultados y las conclusiones. El apéndice A presenta los principales algoritmos utilizados en este trabajo, destacando la implementación de la red neuronal PCNN y nuestra propuesta del algoritmo K-Means el cual lleva por nombre K-MeansP. El apéndice B muestra el resultado de los algoritmos de segmentación descritos en el capítulo 2 para la segmentación de las células del ciclo estral. Finalmente, el apéndice C muestra los artículos que se publicaron gracias al desarrollo de este trabajo.

---

# Capítulo 1

## Marco referencial

Para la detección del ciclo estral se han propuesto diversas metodologías manuales las cuales son descritas por Shannon L. Byers [16] donde a su vez presenta un apoyo visual para la detección del ciclo estral a través de observar la citología de los frotis tomados de la rata. Claudia Caligioni [17] describe el proceso que se debe seguir para identificar el ciclo estral a través de la observación de la citología del frotis.

Para la identificación del ciclo estral por medio de su citología se observa la presencia de células escamosas, células epiteliales nucleadas y leucocitos, la proporción de estas células determina la fase en la que se encuentra [16],[17]:

- **Proestro:** Predominan las células epiteliales nucleadas pueden estar en grupos o individuales (Fig 1.1 a).
- **Estro:** Predominan las células epiteliales escamosas aparecen en grupo (Fig. 1.1 b).
- **Metaestro:** Es una mezcla de células donde predominan los leucocitos pero existe presencia de células epiteliales nucleadas o escamosas (Fig. 1.1 c)
- **Diestro:** Predominan los leucocitos (Fig. 1.1 d)

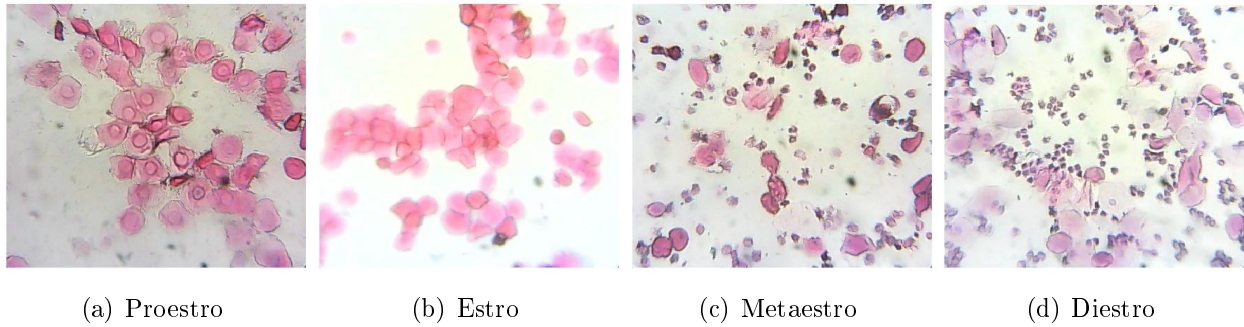


Figura 1.1: Etapas del ciclo estral. Imágenes proporcionadas por el bioterio Claude Bernard

En el trabajo de Shannon se describen una serie de metodologías para la detección del ciclo estral, todas las metodologías son a través de la observación del roedor o de la citología del frotis. Al momento de la redacción no se encontró ningún trabajo que abordara la clasificación del ciclo estral por medio de la visión por computadora y algoritmos inteligentes.

En [1] se propone un esquema de segmentación el cual es aplicado a la segmentación de células en imágenes de alta resolución. Primero utiliza el algoritmo Chan-Vese (C-V) para separar el fondo de las células. Posteriormente propone un filtro de color para confirmar las regiones. Finalmente, un proceso morfológico es usado para refinar las regiones de células extraídas. Estos resultados fueron comparados contra el método Otsu y el método C-V, los resultados se muestran en la figura 1.2.

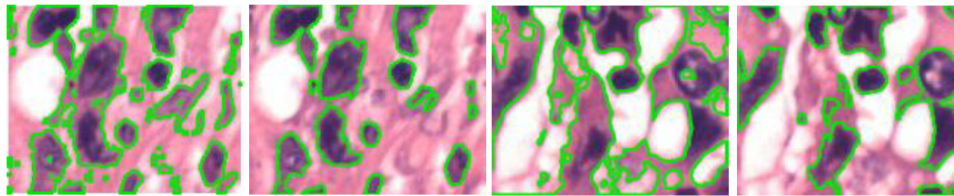


Figura 1.2: De izquierda a derecha umbralado por Otsu, umbralado por algoritmo propuesto, umbralado por C-V, umbralado por algoritmo propuesto [1].

En [2] se muestra un algoritmo que se encarga de la segmentación en imágenes que presentan traslape. Para probar el algoritmo generan una base de imágenes a partir de transformaciones a un conjunto base. En la figura 1.3 se muestran los resultados alcanzados.

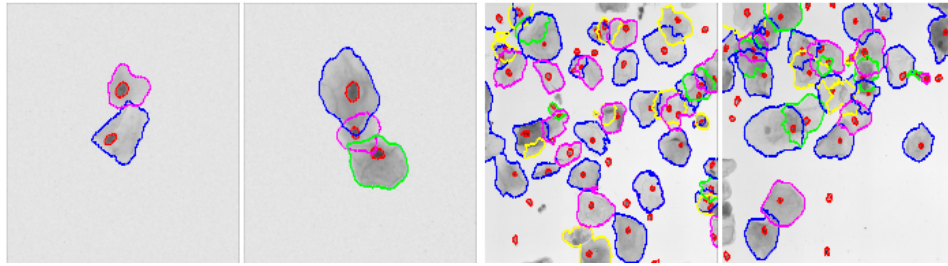


Figura 1.3: De izquierda a derecha las primeras dos imágenes muestran los resultados del algoritmo aplicado a las imágenes sintéticas mientras las últimas dos muestran lo que se obtiene al aplicarlo al caso real [2].

En [3] se implementó un algoritmo para lograr la segmentación de células de cáncer, este partió de la agrupación de super-píxeles para generar un mapa, el mapa se pasó a un algoritmo de umbral adaptativo, después se utilizó un clasificador basado en las características del super-píxel para separar el núcleo del citoplasma, finalmente se utilizó un algoritmo de ejes y regiones para mejorar la segmentación, él método alcanzo un resultado del 94.8 % en la segmentación del citoplasma y de un 98.9 % en la segmentación del núcleo. En la figura 1.4 se muestran los resultados alcanzados.

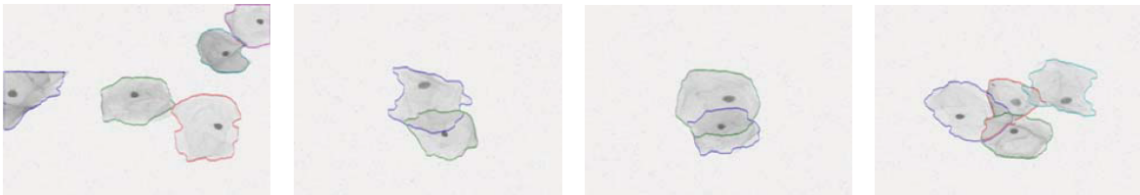


Figura 1.4: De izquierda a derecha células sin traslape, células con doble traslape y con múltiple traslape[3].

En [18] se utiliza una variante de Level Sets como algoritmo de segmentación que es aplicado a la imagen en escala de grises previamente mejorada a través de un filtro Gaussiano y un filtro Wiener. El objetivo es la clasificación de células cancerígenas. Una vez segmentada la imagen se le extraen características geométricas: área, perímetro, excentricidad, radio del área, las cuales son extraídas al citoplasma y al núcleo. Adicionalmente se extraen características de textura: autocorrelación, contraste, correlación, cluster prominence, cluster shade, disimilitud, energía, homogeneidad, probabilidad máxima, varianza de la suma de cuadrados, suma promedio, suma de la varianza, suma de la entropía, diferencia de la entropía, correlación 1, correlación 2, entropía, media de Kurtosis y media de oblicuidad. Para reducir tiempo

de computo se aplica el algoritmo de componentes principales (PCA). La salida del PCA es introducida a 3 Máquinas vector soporte: polinomial con un 95 % de precisión, cuadrática 85 % precisión, Gaussiana 85 % de precisión.

En [4] se utiliza la transformación de línea divisora para la segmentación del núcleo de células. Primero elimina el fondo con una combinación de ecualización de histograma en el RGB y el método Otsu, esto arroja tres matrices binarias que son unidas con una operación  $\circ$  para generar una máscara, a la esta se le aplica un análisis de objetos para eliminar el ruido esto se logra al descartar los objetos con área menor a una célula, finalmente esta máscara se aplica a la imagen original para eliminar el fondo. Al área detectada se le aplica la transformación h mínima para obtener las marcas del núcleo. A la ubicación del núcleo superpuesta con la máscara se le aplica una transformación de distancia para determinar las marcas del citoplasma. La transformación de línea divisora es aplicada considerando las áreas encontradas. Finalmente se aplica C-Means para eliminar falsos positivos. El método es comparado contra el flujo de vector gradiente el cual arroja una sensibilidad mayor pero menor especificidad y armónico medio.

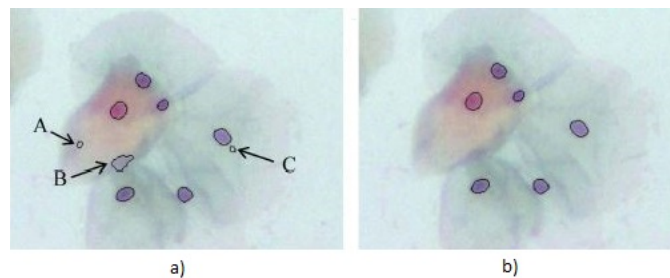


Figura 1.5: a)Posibles núcleos, c)Detección de núcleos reales [4].

En [5] se utiliza el algoritmo de super-píxeles para la segmentación del citoplasma y núcleo así como para la separación de citoplasmas traslapados. La segmentación del núcleo se logra al aplicar K-Means a los super-píxeles definiendo las regiones con un umbral. Para la segmentación del citoplasma se extrajeron características de intensidad del RGB construyendo un vector de características que contiene el centroide del superpíxel, el promedio en rojo, el promedio en verde, el promedio en azul y el promedio general. Este vector de características se usó como entrada de K-Means para la detección de cada citoplasma. Finalmente se determina el área encimada utilizando un algoritmo que evalúa la distancia al centro de cada célula de la región con traslape, considerando además la intensidad para determinar a cuál célula pertenece cada super-píxel.

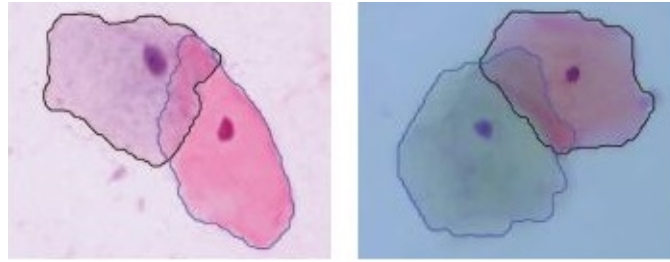


Figura 1.6: Segmentación usando super-píxeles con corrección de células encimadas [5].

En [6] se aborda la segmentación a través del uso de un Otsu de múltiple umbral, así como del uso grafos para el refinamiento, esto se hace en el canal del espacio de color CIELAB. En este artículo se muestra una comparativa de la segmentación por Otsu y por k-means donde el algoritmo Otsu muestra un mejor desempeño. Este algoritmo es utilizado para generar la raíz de los grafos. El algoritmo alcanza un 93 % de precisión comparado contra la segmentación manual.

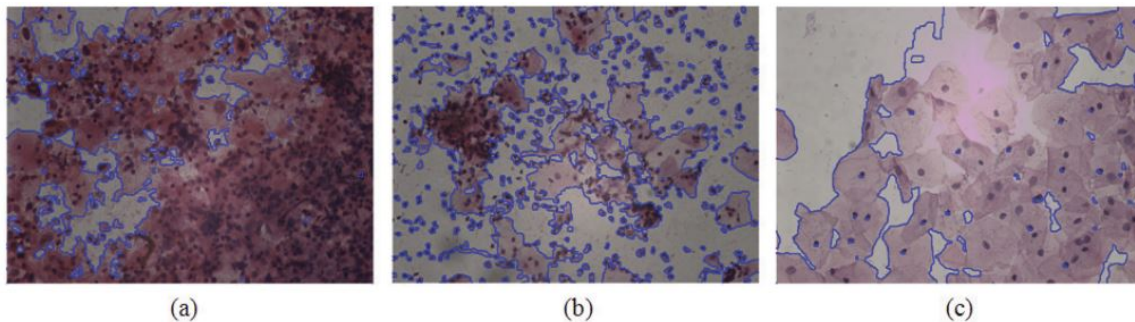


Figura 1.7: Ejemplo de la segmentación en muestras: a) fuertemente encimadas, b) bajo contraste y tinción no uniforme, c) pico de brillo en rojo [6].

En [7] se describe un algoritmo que pretende eliminar la segmentación al proponer un análisis basado en bloques generados de una imagen. Este trabajo considera 12 imágenes que son sub-muestreadas para llevarlas de una resolución de 2592x1944 a una resolución de 800x1000 para finalmente generar bloques de 100x100 píxeles. En el trabajo se utilizan tres espacios de color: el R del RGB, el I del HSI y el gris. Para eliminar el fondo utiliza el algoritmo patrones binarios locales (LBP) en cada bloque generado. A los bloques restantes les extrae características de textura considerando el algoritmo de la matriz de coocurrencia en escala de grises, características de color utilizando el histograma. finalmente reduce el conjunto de características utilizando medidas estadísticas quedándose solo con: Patrones locales binarios uniformes (51-59), concurrencia, energía, correlación, contraste, promedio del histograma, varianza del histograma, oblicuidad del histograma, energía del histograma

(en los tres espacios de color), entropía del histograma (en los tres espacios de color), el radio de la intensidad de color en células epiteliales y el radio en escala de grises para el intervalo entre la célula epitelial y el núcleo. Estos datos se introdujeron a una máquina vector soporte con lo que se obtuvo un 98.98 % de precisión y un 95 % de sensibilidad.

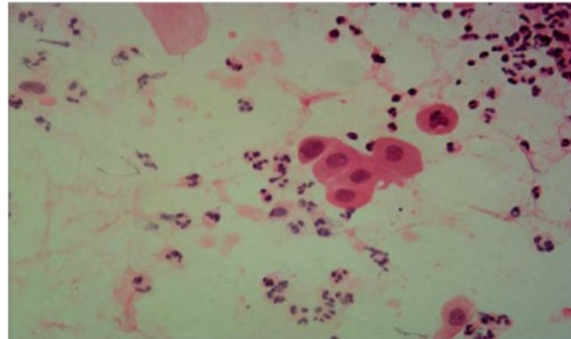


Figura 1.8: Imagen utilizada en el dataset de [7].

En [19] se presenta un algoritmo para la segmentación de los núcleos, El algoritmo utiliza operaciones morfológicas para identificar a los candidatos a núcleo. Las operaciones morfológicas son realizadas en el espacio RGB a cada canal obteniendo la salida a través de una or entre estas. Los posibles núcleos son reducidos utilizando: sus centroides, intensidad y la distancia euclidiana. A la cantidad reducida se le aplica C-Means y una máquina vector soporte para eliminar al resto de falsos núcleos. Los mejores resultados fueron obtenidos por C-Means. Su tiempo de ejecución depende de la cantidad de células en la imagen.

En [8] se presenta una red neuronal convolucional denominada U-Net enfocada para la segmentación de células. El conjunto de imágenes con el que fue entrenada se formó a partir de aplicar deformaciones elásticas a un conjunto reducido. El resultado se muestra en la figura 1.9

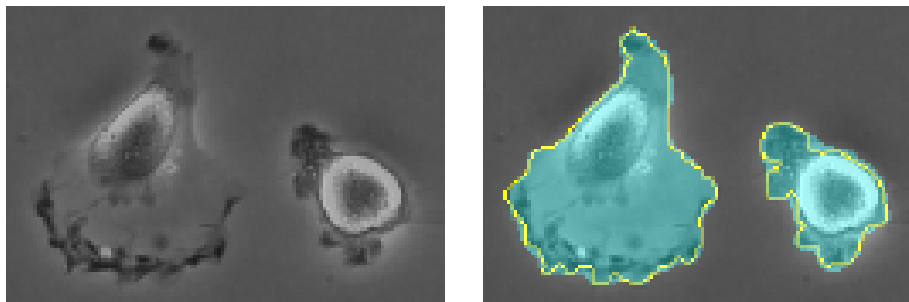


Figura 1.9: Imagen original e imagen segmentada por [8].

---

Se mostraron algoritmos que buscan segmentar o clasificar células, sin embargo aún esta tarea es un problema abierto, una muestra de esto es el ISBI (International Symposium on Biomedical Imaging) reto de seguimiento de células [8], [20] donde cada año se presentan algoritmos que buscan dar la mejor solución a conjuntos de prueba. En el siguiente capítulo se presentan los fundamentos teóricos de los algoritmos utilizados en este trabajo.

---

# Capítulo 2

## Marco teórico

### 2.1. El ciclo estral de la rata

El ciclo estral de la rata dura de 4 a 5 días. Las fases del ciclo son: Proestro, Estro (al final de la cual ocurre la ovulación), Metaestro y Diestro [21].

Hacia el final del Proestro presenta un estado receptivo hacia la copulación el cual llega a su máximo en el Estro en donde aumentan las probabilidades de fecundación. En los estados Metaestro y Diestro no hay receptividad [21].

#### 2.1.1. Observación al microscopio

Las distintas fases del ciclo se pueden diferenciar por el aspecto de los frotis vaginales.

- Proestro: Ocurre un día antes de la ovulación puede durar entre 12 y 18 horas. Abundan las células epiteliales de gran tamaño y con núcleo, no hay leucocitos. Dentro de esta fase se pueden distinguir varias sub-fases según el número de células y tamaño de los núcleos. En Proestro tardío las células epiteliales superficiales empiezan a transformarse en células cornificadas, con aspecto escamoso, anucleadas y transparentes de forma irregular, que aparecerán en el frotis claramente distinguibles de las células nucleadas [21].
- Estro: Al final de él ocurre la ovulación, suele durar 12 horas. Hay gran abundancia de células anucleadas escamosas, que pueden aparecer formando masas blancas e irregulares. Proceden de la descamación del epitelio superficial. La fase ovulatoria y la conducta sexual de la rata tienden a estar sincronizadas por ritmos biológicos circadianos de ma-

nera que estos eventos ocurren en determinadas horas (medianoche y primeras horas de la mañana respectivamente) [21].

- **Metaestro.** Se caracteriza por la presencia de algunos leucocitos además de células cornificadas anucleadas. Los leucocitos se reconocen bien por su pequeño tamaño y la forma irregular de su núcleo, que aparece fuertemente teñido. Es una fase de transición una vez finalizada la fase Estro hacia la siguiente fase de Diestro, dura de 10 a 14 horas [21].
- **Diestro.** Los leucocitos invaden el epitelio y son las células predominantes en los frotis. Es la fase más larga y variable del ciclo (2-3 días). Empiezan a aparecer células epiteliales nucleadas, más abundantes cuanto más avanzado sea el Diestro. Al final del Diestro los folículos ováricos empiezan a crecer[21].

## 2.2. Procesamiento digital de imágenes

El PDI recibe como entrada una imagen y produce como resultado una imagen, se encarga de:

- **Mejorar una imagen:** esta rama del PDI tiene que ver con el aumento del contraste de la imagen de entrada, la eliminación del ruido en dicha imagen, así como el filtrado espacial y de frecuencia entre otros [22].
- **La restauración de una imagen:** esta área del PDI tiene como objeto el estudio de fenómenos tales como el desenfoque, así como las alteraciones sobre una imagen causadas por el movimiento y otro tipo de degradaciones que puedan originarse en el momento de la captura [22].
- **Reconstrucción:** esta técnica es muy empleada en áreas como la medicina y la geología para la obtención de una imagen tridimensional a través de diferentes cortes (imágenes bidimensionales) [22].
- **Compresión de imágenes:** tiene que ver con las formas de codificar imágenes en archivos más pequeños con miras a un mejor almacenamiento y transmisión más eficiente [22].

El primer paso en esta área de relativamente reciente creación fue dado por los laboratorios Jet Propulsion en 1964 quienes procesaron imágenes provenientes de la luna. Desde ese momento se han ido incrementado los desarrollos en esta área, así como las herramientas de desarrollo [23],[22].

En 1999 surge la biblioteca OpenCV desarrollada por Intel la cual hoy en día es soportada por los sistemas Windows, Linux, Mac OS, iOS y Android. Está disponible para C++, C, Python y Java. Fue diseñada para tener una alta eficiencia computacional y un fuerte enfoque hacia aplicaciones de tiempo real. La biblioteca puede tomar ventaja de los procesadores multi-núcleo y de la aceleración por hardware. Es una biblioteca libre para el uso académico y comercial bajo la licencia BSD [24].

### 2.2.1. Representación de imágenes

Las formas más comunes en las que se pueden encontrar las imágenes son RGB por sus siglas en inglés Red Green Blue y escala de grises cada una de estas formas puede encontrarse en una amplia variedad de bits de imagen, la más común es de 8 bits dando 3 matrices con rangos de 0 255 para el caso RGB y una matriz de 0 255 para el caso de la escala de grises. Valores altos en un pixel representan mayor cantidad de luz, siendo el vector  $[255, 255, 255]$  un color blanco en el RGB así como el valor 255 un blanco para la escala de grises.

Existen diferentes formas para convertir una imagen en RGB a escala de grises. La forma más usada es la que se asemeja a la percepción del ojo humano del color. Para esto cada pixel en la escala de grises es representado como la suma ponderada de los valores en RGB dando  $\text{Gray} = 0.2989 * R + 0.5870 * G + 0.1140 * B$ .

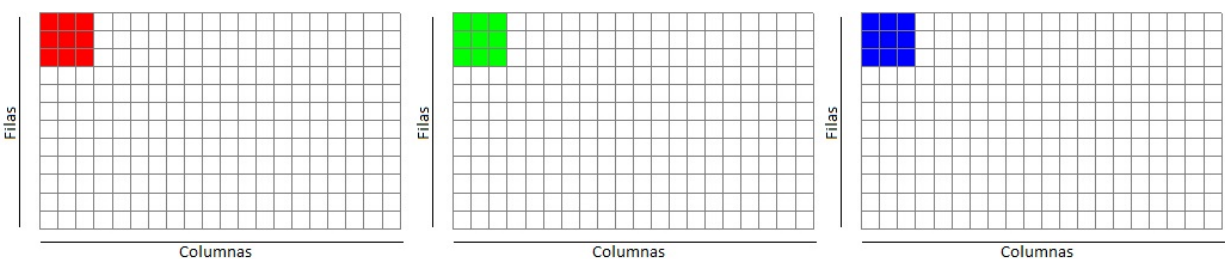


Figura 2.1: Matrices RGB

Adicionalmente al espacio RGB y la escala de grises se utilizará el espacio de color CIE-LAB propuesto por HunterLab [9]. El espacio CIELAB representa a las imágenes en 3 canales el canal L que representa la intensidad de luz, el canal A que representa la variación del color rojo al verde y el canal B que representan la variación del color azul al amarillo.

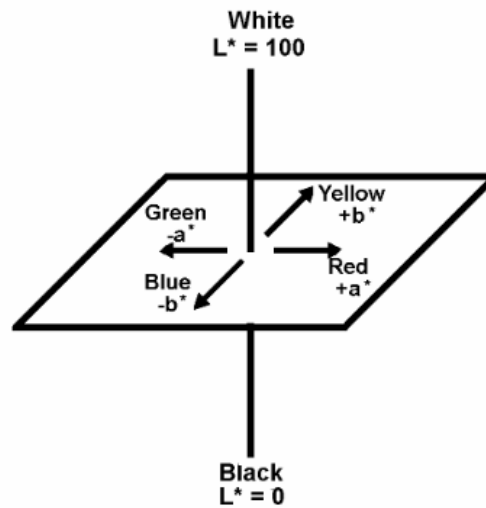


Figura 2.2: Representación del espacio de color CIELAB. Imagen tomada de [9].

### 2.2.2. Umbralización

La umbralización es el proceso por el cual se selecciona un valor o un conjunto de valores dentro del espacio de color, el cual sirve como medida para separar el fondo del conjunto de píxeles deseados. Existen actualmente diferentes algoritmos de umbralización no supervisado como son: La Media Aritmética, Máxima Entropía, Entropía Cruzada, Otsu. Estos algoritmos funcionan a través de una búsqueda exhaustiva del valor en el espacio de color que maximice o minimice su correspondiente función objetivo.

#### Media Aritmética

La expresión matemática para la umbralización por media aritmética está dada por:

$$\mu_{Umbral} = \sum_{n=1}^N L_n P(L_n) \quad (2.1)$$

Donde  $L_n$  es el  $n$ -ésimo valor de gris de la imagen y  $P(L_n)$  es la distribución de probabilidad de  $L_n$  [25].

El umbralado a partir de el algoritmo de la Media Aritmética calcula el valor promedio de la intensidad de una imagen, el cual es utilizado como umbral.

## Máxima Entropía

La expresión matemática para la umbralización por máxima entropía está dada por:

$$H(S) = -p_1 \ln p_1 - p_0 \ln p_0 \quad (2.2)$$

Donde  $p_1$  y  $p_0$  son las densidades de probabilidad de 1 y 0 de la imagen segmentada y  $S$  es el umbral a evaluar, el método selecciona el valor  $S$  que maximice la expresión 2.2 [25].

## Entropía Cruzada

La expresión matemática para la umbralización por Entropía Cruzada está dada por:

$$\begin{aligned} C(P, Q : t) = & \sum_{f=1}^t [fh(f) \ln \frac{f}{\mu_1(t)} + \mu_1 h(f) \ln \frac{\mu_1(t)}{f}] \\ & + \sum_{f=t+1}^F [fh(f) \ln \frac{f}{\mu_2(t)} + \mu_2 h(f) \ln \frac{\mu_2(t)}{f}] \end{aligned} \quad (2.3)$$

$$\mu_1(t) = \frac{1}{\sum_{f=0}^t h(f)} \sum_{f=0}^t fh(f) \quad (2.4)$$

$$\mu_2(t) = \frac{1}{\sum_{f=t+1}^F h(f)} \sum_{f=t+1}^F fh(f) \quad (2.5)$$

Donde  $f$  es valor en el espacio de color,  $F$  es el máximo valor de  $f$ ,  $t$  es el valor del umbral a evaluar,  $h(f)$  es el valor del histograma,  $\mu_1$  y  $\mu_2$  son el promedio del valor en el espacio de color del objeto y el fondo respectivamente. El objetivo es buscar el valor de  $t$  que minimice 2.3 [25]

## Otsu

El algoritmo Otsu recibe su nombre por el investigador que lo propuso en 1979 como un algoritmo de umbralización no supervisado y no paramétrico que maximiza la separabilidad entre clases dentro de una imagen en la escala de grises. El objetivo del algoritmo es encontrar el valor  $K$  que maximice la ecuación 2.6 [26].

$$\sigma_B^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \quad (2.6)$$

$$\mu_T = \sum_{i=1}^L ip_i \quad (2.7)$$

$$\omega_0 = \sum_{i=1}^K p_i \quad (2.8)$$

$$\omega_1 = \sum_{i=K+1}^L p_i \quad (2.9)$$

$$\mu_0 = \sum_{i=1}^K \frac{ip_i}{\omega_0} \quad (2.10)$$

$$\mu_1 = \sum_{i=K+1}^L \frac{ip_i}{\omega_1} \quad (2.11)$$

$$p_i = n_i/N \quad (2.12)$$

Donde  $n_i$  es el valor del histograma para la intensidad  $i$ ,  $N$  es el número total de píxeles,  $L$  es la intensidad máxima y  $K$  el umbral a evaluar.

### 2.2.3. K-Means

El algoritmo K-Means fue publicado inicialmente por Stuart Lloyd en los laboratorios Bell, desde entonces se han presentado mejoras a este algoritmo como es el K-Means++ [27] el cual se diferencia en el momento de seleccionar las semillas iniciales, el K-Means propuesto por Hartigan [28] el cual reduce las iteraciones al funcionar con conjuntos vivos<sup>1</sup>. De manera directa el algoritmo K-Means no es un algoritmo de umbralado si no un algoritmo de agrupación el cual retorna  $K$  conjuntos, donde cada uno está asociado con un centroide. Para aplicar este algoritmo para segmentar se seleccionan los  $K - n$  conjuntos, donde  $n$  es el número de elementos pertenecientes al fondo de la imagen.

#### Algoritmo 1: K-Means

Paso 1 Seleccionar arbitrariamente los centroides  $C_k$ ,  
 $k = 1, 2, 3, \dots, K$ .

Paso 2 Establecer los conjuntos  $Co_k$  con los  $x_i$  elementos más cercanos a cada centroide  $C_k$ .

Paso 3 Para cada  $Co_k$  establecer los nuevos centroides como el promedio de los  $x_i$  pertenecientes.

<sup>1</sup>Se denomina conjunto vivo al conjunto que se mantiene con cambios entre cada iteración del algoritmo K-Means, es decir si el centroide no cambia de una iteración a otra este dejara de ser utilizado.

Paso 4 Si  $C_k$  no cambio terminar si no repetir desde el paso 2.

## K-Means para imágenes en escala de grises

En [29] se presenta un algoritmo de K-Means que itera sobre el histograma de la imagen con lo cual mejora el tiempo de ejecución considerablemente al reducir la cantidad elementos<sup>2</sup> al número de niveles de gris de la imagen<sup>3</sup>.

Algoritmo 2: Transcripción del algoritmo propuesto en [29]

```

Histogram for gray image
INPUT N, Ml, and  $f(p_j^i)$  where  $i=1,2,\dots,M, f(p_j^i)$  is
pixel level function of  $p_j^i$ 
OUTPUT  $h(r)$ , where  $0 \leq r \leq l$ 
Step 1 For  $i=1,2,\dots, N$  do Steps 1-3
Step 2 For  $j=1,2,\dots, M_i$  do Steps 2-3
Step 3 If  $r=f(p_j^i)$ , then  $h(r)=h(r)+1$ 
Step 4 OUTPUT  $h(r)$ 
Step 5 Stop

Fast K-means algorithm for gray image:
Input N, Ml,  $h(r)$ , K,  $n \delta T$ , t and  $C_k^{(0)}$ , where  $i=1,2,\dots,N$ ,
 $0 \leq r \leq l$ ,  $k=1,2,\dots,K$ , t is the tth iteration,  $\delta T$  is the threshold
for continuing training of the cluster center and (0) is the
0th iteration
OUTPUT  $C_k^{op}$ 
Step 1 Set t=0
Step 2 For  $r=1,2,\dots,L$  do Steps 2-5
Step 3 For  $k=1,2,\dots,K$  do Steps 3-5
Step 4 Set  $d(r, C_k^t) = r - C_k^{(t)}$ ,  $C_k^p = C_k^p + r$  and  $C_k^N = C_k^N + h(r)$ ,
where  $k = \operatorname{argmin}[d(r, C_k)]$ ,  $C_k^p$  is
the total gray level of the kth cluster center and  $C_k^N$  is the
total pixel number of the kth cluster center.
Step 5 Set t=t+1
Step 6 For  $k=1,2,\dots,K$  do Steps 7-9
Step 7 Set  $C_k^{(t)} = \frac{C_k^p}{C_k^N}$ 
Step 8 Set  $\delta(C_k^{(t)}, C_k^{(t-1)}) = C_{k,l}^{(t)} - C_{k,l}^{(t-1)}$ 
Step 9 if  $\delta(C_k^{(t)}, C_k^{(t-1)}) \geq \delta T$  do Step 2
Step 10 Set  $C_k^{op} = C_k^{(t)}$ 
Step 11 OUTPUT ( $C_k^{(op)}$ )
Step 12 STOP

```

<sup>2</sup>Si se aplica directamente el algoritmo K-Means a una imagen este iterara sobre cada pixel por lo cual si se tiene una imagen de 100\*100 iteraría sobre 10000 elementos al aplicarlo sobre el histograma iterara sobre la cantidad de niveles de gris.

<sup>3</sup>El algoritmo K-MeansP (Apéndice A.7) inicialmente se había planteado como una contribución propia. Al comenzar a redactar un artículo en torno a él se encontró el artículo [29] el cual presenta la misma propuesta con un error en la descripción de su algoritmo.

### 2.2.4. Operaciones basadas en objetos

En las imágenes binarias se entiende como objeto a un conjunto de píxeles conectados entre sí con valor uno, por ejemplo, considere la imagen de la figura 2.3 en ella se observa un conjunto de píxeles que mantienen una conexión entre ellos. Este tipo de operaciones es útil cuando se desea realizar conteo de objetos o extracción de características inherentes a ellos tales como área y perímetro en píxeles o centro de masa de un objeto [30].

0	0	0	0	0	0
0	0	0	0	0	0
0	1	1	1	0	0
0	1	1	1	0	0
0	1	1	1	0	0
0	0	0	0	0	0

Figura 2.3: Matriz de imagen binaria

#### Conexión de 4 vecindades

La conexión de cuatro vecindades busca etiquetar a cada objeto utilizando los vecinos superiores, inferiores, y ambos laterales. Tomando la imagen de la figura 2.4 y de acuerdo a la conexión de 4 vecindades se puede observar que en dicha figura están presentes 2 objetos.

0	0	0	0	0	0
0	1	1	0	0	0
0	1	1	0	0	0
0	0	0	1	1	0
0	0	0	1	1	0
0	0	0	0	0	0

Figura 2.4: Matriz de imagen binaria con 2 objetos

#### Algoritmo 3: Conexión de 4 vecindades

```

Paso 1 Se obtienen las dimensiones de la imagen F y C
Paso 2 Se inicia en la posición (0,0)
Paso 3 Se obtiene el valor del pixel
Paso 4 Si el pixel es 1 se obtienen los valores superior e izquierdo.
Paso 5 Se revisa que las etiquetas de estos pixeles sean iguales
Paso 6 Si son iguales se le asigna la misma etiqueta al pixel
           presente y se avanza a la siguiente posición.
Paso 7 Si son diferentes se genera una bandera de igualdad
           entre ellas, se asigna cualquiera de las dos al pixel presente

```

y se avanza a la siguiente posición.

Paso 8 Si solo una existe se le asigna esta al pixel presente y se avanza a la siguiente posición.

Paso 9 Si ninguna existe se genera una nueva etiqueta la cual se le asigna al pixel presente y se avanza a la siguiente posición.

Paso 10 Si el pixel es 0 se avanza a la siguiente posición y se realiza desde el paso 3 hasta que se barra toda la imagen.

Paso 11 Se utilizan las banderas de igualdad para renombrar a cada vecindad

Note que cada etiqueta debe ser un número entero

### 2.2.5. Rasgos

Como rasgos o características en visión por computadora se entiende un conjunto de valores numéricos asociados a características inherentes a la imagen. Dentro de los rasgos destacan aquellos que son extraídos de la textura de imágenes y los referentes a formas. De un conjunto de rasgos descriptores se esperan las siguientes características:

- Poder de discriminación. Los rasgos deben permitir discriminar entre los objetos de diferentes clases; tienen que proporcionar valores numéricos diferentes para objetos de clases distintas [30].
- Fiabilidad. Las características deben ser fiables; es decir con cambios numéricos pequeños para objetos de la misma clase [30].
- Incorrelación. Las características deben presentar la menor correlación posible entre ellas.
- Rapidez. Las características deben calcularse en tiempos aceptables. Este es un requisito que puede llegar a ser determinante en ciertas aplicaciones [30].
- Economía. Las características deben obtenerse con captosres económicos [30].

#### Rasgos de textura

Uno de los principales algoritmos para extraer rasgos de textura fue presentado por Robert M. Harlick en 1973 [31]. A este algoritmo se le conoce en la actualidad como la matriz de co-ocurrencia en escala de grises (GLCM Gray Level Cooccurrence Matrix). El procedimiento para obtener los rasgos de textura a partir de la GLCM está basado en asumir que la información de textura de una imagen  $I$  está contenida en la relación espacial promedio entre los tonos de gris. Para calcular la matriz GLCM suponga una imagen rectangular de resolución  $N_x$  horizontal y  $N_y$  vertical. Suponga una resolución de grises  $N_g$  entonces

$L_x = 1, 2, 3, \dots, N_x$  será el dominio horizontal,  $L_y = 1, 2, 3, \dots, N_y$  será el dominio vertical y  $G = 1, 2, 3, \dots, N_g$  es el conjunto cuantizado de tonos de gris. Entonces la imagen  $I$  puede ser representada como una función que asigna un tono de gris en  $GL$  a cada celda o par de coordenadas  $L_x, L_y$ ;  $I : L_x * L_y \rightarrow GL$ . Ahora considere el paso como el salto de pixeles es decir un paso de 1 sería el pixel continuo y un paso de 2 sería considerar el segundo pixel continuo. Ahora considere un ángulo como la dirección del pixel que se considerara, es decir el pixel a 0 grados sería el pixel continuo de lado derecho y a 45 el pixel superior izquierdo. Las expresiones formales para el cálculo a un paso  $d$  son:

$$\begin{aligned} G(i, j, d, 0) &= \#\{(k, l), (m, n) \in (L_y, L_x) \\ (L_y, L_x) \mid k - m = 0, |l - n| = d, I(k, l) = i, I(m, n) = j \} \end{aligned} \quad (2.13)$$

$$\begin{aligned} G(i, j, d, 45) &= \#\{(k, l), (m, n) \in (L_y, L_x) \\ (L_y, L_x) \mid (k - m = d, l - n = -d), \\ o(k - m = -d, l - n = d) I(k, l) = i, I(m, n) = j \} \end{aligned} \quad (2.14)$$

$$\begin{aligned} G(i, j, d, 90) &= \#\{(k, l), (m, n) \in (L_y, L_x) \\ (L_y, L_x) \mid |k - m| = d, l - n = 0, I(k, l) = i, I(m, n) = j \} \end{aligned} \quad (2.15)$$

$$\begin{aligned} G(i, j, d, 135) &= \#\{(k, l), (m, n) \in (L_y, L_x) \\ (L_y, L_x) \mid (k - m = d, l - n = d), \\ o(k - m = -d, l - n = -d), I(k, l) = i, I(m, n) = j \} \end{aligned} \quad (2.16)$$

La figura 2.5 muestra un ejemplo del cálculo.

Las métricas utilizadas en este trabajo calculadas a partir del algoritmo GLCM son [32]:

Contraste:

$$C = \sum_{i,j=0}^{l-1} P_{i,j} (i - j)^2 \quad (2.17)$$

Disimilitud:

$$D = \sum_{i,j=0}^{l-1} P_{i,j} |i - j| \quad (2.18)$$

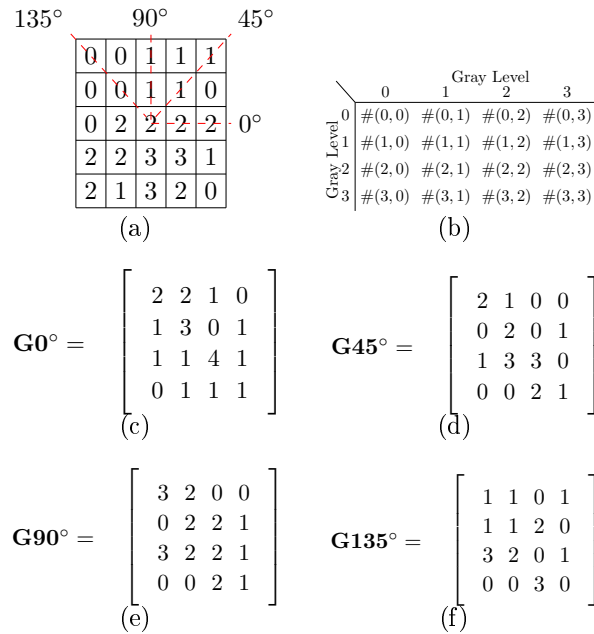


Figura 2.5: a) 5x5 imagen 4 niveles de gris, b) Forma general para una matriz GLCM para imágenes con valores de gris de 0-3, c)-f) Cálculo de la GLCM para:  $(d,\theta)=(1,0),(1,45),(1,90),(1,135)$

Homogeneidad:

$$H = \sum_{i,j=0}^{l-1} \frac{P_{i,j}}{1 + (i - j)^2} \quad (2.19)$$

Energía:

$$E = \sum_{i,j=0}^{l-1} P_{i,j} \quad (2.20)$$

donde  $P_{i,j}$  (la probabilidad de co-ocurrencia entre los niveles de gris  $i$  y  $j$ ) se define como [33]:

$$P_{i,j} = \frac{G_{i,j}}{\sum_{i,j=0}^{l-1} G_{i,j}} \quad (2.21)$$

### Rasgos de forma

El principal algoritmo que se utiliza para extraer rasgos de forma son los momentos invariantes de Hu [30] sin embargo el costo computacional es alto y el momento con mayor discriminación es el primero. En este trabajo se utilizó en su lugar los rasgos a partir del factor de compacidad.

Factor de compacidad:

$$FC = \frac{\text{Área}}{\text{Perímetro}^2} \quad (2.22)$$

Compacidad total:

$$CT = \sum_{i=1}^{Nl} FC_i \quad (2.23)$$

Compacidad promedio:

$$CP = \frac{\sum_{i=1}^{Nl} FC_i}{Nl} \quad (2.24)$$

## 2.3. Inteligencia artificial

La inteligencia artificial tiene como objetivo el desarrollo de algoritmos que sean capaces de aprender<sup>4</sup> a través de la experiencia. Existen dos vertientes: una orientada a sistemas de aprendizaje no supervisados y la otra a sistemas supervisados. Dentro de la primera categoría se encuentran algoritmos como los mapas auto-organizados los cuales separan las entradas mediante características inherentes a ellas. En el caso de los sistemas supervisados se encuentran algoritmos como el K-NN el cual utiliza un conjunto de etiquetas para cada ejemplo de entrenamiento, de esta forma logra la clasificación.

### 2.3.1. Aprendizaje basado en instancia

Una característica que diferencia el aprendizaje basado en instancia es que almacena los ejemplos de entrenamiento, los cuales son utilizados cada vez que se desea realizar una clasificación. Para este tipo de métodos es necesario extraer características.

Se conoce como dimensionalidad a la dimensión del vector de características, El vector de características es el resultado de ordenar dentro de un vector el conjunto de características. Así pues, si se tienen 10 características se poseerá un vector de características de dimensionalidad 10. El problema surge cuando se tienen demasiadas características lo que implica una dimensionalidad alta y gran tiempo de computo. En estos casos se emplean algoritmos como el PCA [34] el cual se encarga de reducir la dimensionalidad al tamaño del conjunto de ejemplos de entrenamiento esto a través de transformaciones lineales.

#### K-NN

El algoritmo K-NN utiliza la distancia euclidiana entre la entrada de prueba y el conjunto de entrenamiento para determinar la clase a la cual pertenece la entrada de prueba, como entrada de prueba se hace referencia a la entrada que debe ser clasificada, a cada entrada se le deben aplicar el mismo proceso de extracción de características que se le aplico al conjunto de entrenamiento [35]. En la figura 2.6 se observan dos conjuntos de entrenamiento con 2 características A, B cada uno y una entrada de test. El algoritmo determinará la clase en base a los k vecinos seleccionados. En la figura 2.7 se observan las distancias entre la entrada y el conjunto de entrenamiento. Para una  $K = 1$  se observa que la clase a la que pertenecería la entrada sería la 1. Un caso particular se presenta en  $K = 2$  en el cual se realiza un empate,

---

<sup>4</sup>El termino de aprender se refiere a generalizar es decir a partir de ejemplos vistos los algoritmos deben predecir ejemplos desconocidos.

en este punto se debe decidir a qué clase se le dará mayor peso.

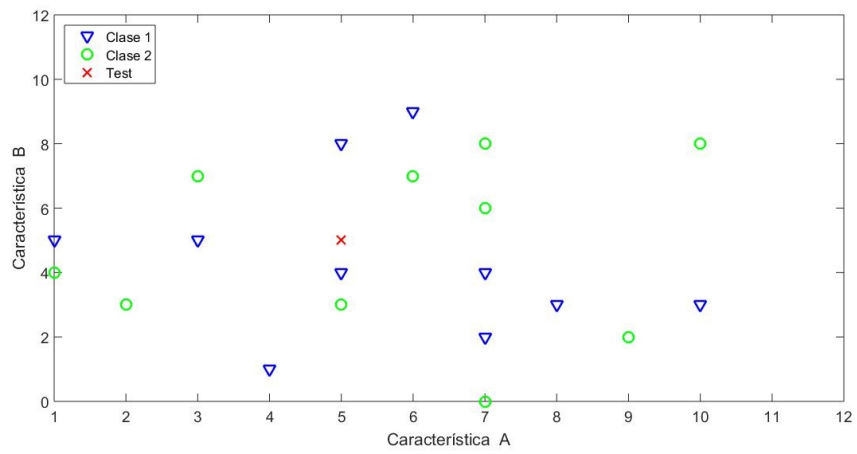


Figura 2.6: K-NN Con 2 Clases.

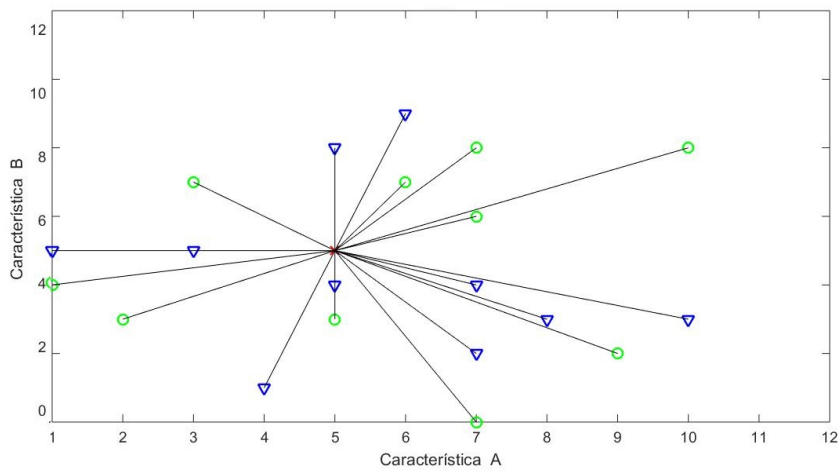


Figura 2.7: Distancias entre el conjunto de entrenamiento y la entrada.

#### Algoritmo 4: K-NN

- Seleccionar los conjuntos de entrenamiento.
  - Ejemplo: Manzanas Gala y Golden.
- Extraer características del conjunto y etiquetar
  - Ejemplo: Características:
    - Color

- PH
- Etiquetas:
  - 1=Manzanas Gala
  - 2=Manzanas Golden
- Seleccionar elemento a evaluar
  - Ejemplo: Manzana de test
- Extraer características del elemento seleccionado
  - Ejemplo: Características
    - [255,45,15]
    - 3.3
- Calcular la distancia euclidiana con el conjunto de entrenamiento
  - Ejemplo: sea A el vector de características de test [255,45,15,3.3] y B un elemento del conjunto de entrenamiento cuyo vector característico es [245,77,10,3.0] entonces la distancia euclidiana se define como  $d = \sqrt{\sum_{i=1}^4 (A_i - B_i)^2}$
- Seleccionar las etiquetas de los k vecinos más cercanos
  - Ejemplo: Sea  $K = 3$  con el vector de distancias euclidianas [2,5,7,58,...] con el vector de etiquetas asociado [1,1,2,1,...] entonces las etiquetas seleccionadas son [1,1,2]
- Obtener la etiqueta con mayor frecuencia
  - Ejemplo: si el vector es [1,1,2] la etiqueta es 1

### 2.3.2. Perceptrón

Un perceptrón toma un vector de entrada de valores reales, calcula una combinación lineal, entonces asigna a la salida un valor de uno si el resultado sobrepasa un valor de umbral y menos uno en otro caso, es decir dada una entrada  $X_i = [x_1, \dots, X_n]$  la salida  $O[x_i, \dots, x_n]$  calculada por el perceptron es:

$$O(x_i, \dots, x_n) = \begin{cases} 1 & \text{si } w_0 + w_1x_1 + \dots + w_nx_n > 0 \\ -1 & \text{Para el resto} \end{cases} \quad (2.25)$$

Donde cada  $w_i$  es un valor constante real o peso que determina la contribución de la entrada  $x_i$  a la salida del perceptrón además  $w_0$  es el valor de umbral el cual debe ser superado por la combinación lineal  $w_i x_i$  con  $i > 0$  para que la salida sea uno.

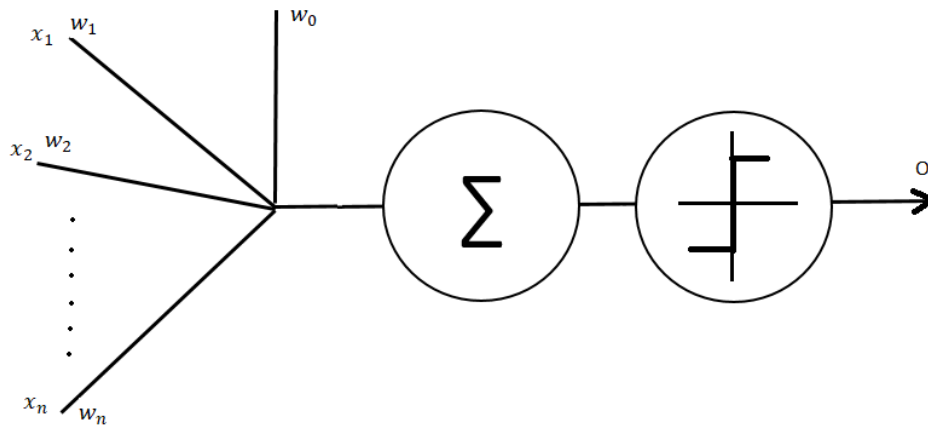


Figura 2.8: Perceptrón:  $w_0$  es el valor de umbral  $x_i$  es la entrada y  $w_i$  son los pesos sinápticos.

### Regla de entrenamiento del perceptrón

Una vez definido el perceptrón es necesario definir una técnica que ajuste los valores de los pesos sinápticos de la neurona. Para el caso de una neurona el método clásico se conoce como regla de entrenamiento del perceptrón, la cual se define como:

$$w_i = w_i + \Delta w_i \quad (2.26)$$

donde  $\Delta w_i = n(t - o)x_i$ ,  $t$  es la salida objetivo,  $o$  es la salida general y  $n$  es una constante positiva llamada tasa de aprendizaje, generalmente  $n$  toma valores pequeños y a veces decae con cada iteración.

### 2.3.3. Descenso de gradiente

El algoritmo de descenso de gradiente es utilizado para entrenar los pesos sinápticos de una neurona. Busca reducir un término denomina error de entrenamiento el cual está definido en términos de la salida deseada de la neurona y la salida obtenida. Para entrar al algoritmo del descenso de gradiente es necesario definir varios términos.

#### Error de entrenamiento del vector de pesos

El error de entrenamiento está definido en términos de la salida objetivo de la neurona y la salida obtenida.

$$E(w) = \sum_{d \in D} (t_d - O_d)^2 \quad (2.27)$$

Donde  $D$  es el conjunto de entrenamiento,  $t_d$  es la salida deseada para el ejemplo  $d$  del conjunto de entrenamiento y  $O_d$  es la salida para el ejemplo  $d$ . De lo anterior  $E(w)$  es la mitad de la diferencia al cuadrado entre la salida objetivo y la real.

### Espacio de hipótesis

Para comprender el descenso de gradiente es necesario visualizar el espacio de hipótesis de los posibles vectores de pesos y su valor asociado  $E$  como se ilustra en la figura 2.9. Los ejes  $w_0$  y  $w_1$  del plano representan los posibles valores para dos pesos sinápticos. El eje vertical indica el error relativo  $E$ . Se busca el conjunto de pesos sinápticos con el mínimo error.

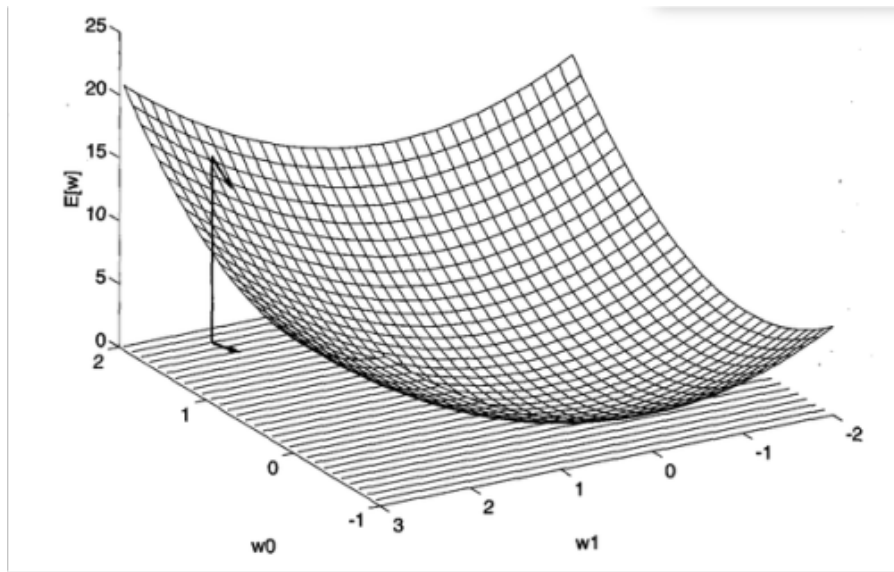


Figura 2.9: Espacio de hipótesis [10]

El descenso de gradiente busca determinar el vector de pesos que minimice  $E$  comenzando con un valor arbitrario del vector de pesos. Para calcular la dirección del descenso del error a lo largo de su superficie se obtiene la derivada de  $E$  con respecto a cada componente del vector  $w$ . Este vector derivado es llamado el gradiente de  $E$  con respecto a  $w$  y se expresa:

$$\nabla E(w) = \left[ \frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_n} \right] \quad (2.28)$$

Con lo anterior la regla de descenso de gradiente se define como:

$$w = w + \Delta w \quad (2.29)$$

donde

$$\Delta w = -n \nabla E(w) \quad (2.30)$$

Donde  $n$  es una constante positiva llamada tasa de entrenamiento que determina el tamaño de búsqueda en el descenso de gradiente. El signo negativo está presente por que se busca el vector de pesos en la dirección de decremento de  $E$ .

### 2.3.4. Perceptrón multicapa

El perceptrón multicapa o MLP (Multi layer perceptron) es una red neuronal compuesta por arreglos de perceptrones ordenados en capas. Para entrenar esta red neuronal se usa una extensión de la regla del descenso de gradiente redefiniendo el termino de error.

### 2.3.5. Red neuronal pulso acoplada

La investigación en la corteza visual ha arrojado modelos matemáticos que describen el potencial en las neuronas, podemos encontrar modelos como los propuestos por: Hodgkin-Huxley, Fitzhugh-Nagumo, Rybank, Eckhorn entre otros. El modelo de Eckhorn se basa en la corteza visual del gato posee dos comportamientos: uno de retroalimentación y otro de enlace ambos se combinan para alcanzar un voltaje  $U_m$  que se compara contra un voltaje de umbral local  $\theta$ . El modelo de Eckhorn esta descrito por:

$$U_{m,k}(t) = F_k(t)[1 + L_k(t)] \quad (2.31)$$

$$F_k(t) = \sum_{i=1}^N [w_{ki}^f Y_i(t) + S_k(t) + N_k(t)] \otimes I(V^a, \tau^a, t) \quad (2.32)$$

$$L_k(t) = \sum_{i=1}^N [w_{ki}^l Y_i(t) + N_k(t)] \otimes I(V^l, \tau^l, t) \quad (2.33)$$

$$Y_k(t) = \begin{cases} 1 & \text{si } U_{m,k}(t) \geq \theta_k(t) \\ 0 & \text{Para el resto} \end{cases} \quad (2.34)$$

$$X(t) = Z(t) \otimes I(v, \tau, t) \quad (2.35)$$

$$X[n] = X[n-1]e^{-\frac{t}{\tau}} + VZ[n] \quad (2.36)$$

Donde  $N$  es el número de neuronas,  $w$  es el peso sinóptico,  $Y$  es la salida y  $S$  es el estímulo externo. Los rangos típicos para las constantes son:  $\tau^a = [10, 15]$ ,  $\tau^l = [0.1, 1]$ ,  $\tau^S = [5, 7]$ ,  $V^a = 0.5$ ,  $V^l = [5, 30]$ ,  $V^S = [50, 70]$ ,  $\theta_0 = [0.5, 1.8]$ .

Para la implementación digital de este modelo es necesario realizar ciertos cambios:

$$F_{ij}[n] = e^{\alpha_F} F[n-1] + S_{ij} + V_F \sum_{kl=1}^N M_{ijkl} Y_{kl}[n-1] \quad (2.37)$$

$$L_{ij}[n] = e^{\alpha_L} L[n-1] + V_L \sum_{kl=1}^N W_{ijkl} Y_{kl}[n-1] \quad (2.38)$$

Donde  $F_{ij}$  es el comportamiento de retroalimentación de la neurona  $(i, j)$  en un arreglo de 2 dimensiones y  $L_{ij}$  es el comportamiento de enlace.  $Y_{kl}$  son las salidas de las neuronas de una interacción previa. Ambos comportamientos poseen memoria la cual decae con el tiempo de manera exponencial a través de los términos  $\alpha_F$  y  $\alpha_L$  [11]. Las constantes  $V_F$  y  $V_L$  son constantes de normalización.

El estado de los 2 comportamientos es combinado para crear un estado interno de la neurona llamado  $U$  este es controlado a través de la constante  $\beta$ . La actividad interna se calcula por:

$$U_{ij}[n] = F_{ij}[n] \{1 + \beta L_{ij}[n]\} \quad (2.39)$$

El estado interno de la neurona es comparado con un umbral dinámico  $\Theta$  para producir la salida  $Y$  por:

$$Y_{ij} = \begin{cases} 1 & \text{si } U_{ij}[n] > \theta_{ij}[N] \\ 0 & \text{Para el resto} \end{cases} \quad (2.40)$$

El umbral dinámico  $\theta$  cuando la neurona es disparada incrementa su valor, el cual decae hasta que la neurona es disparada de nuevo. El proceso es descrito por:

$$\theta_{ij}[n] = e^{\alpha_\theta} \theta_{ij}[n-1] + V_\theta Y_{ij}[n-1] \quad (2.41)$$

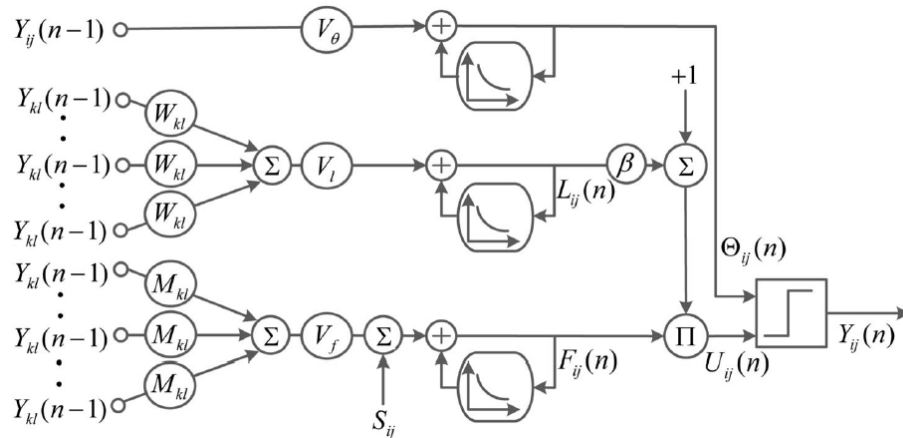


Figura 2.10: Neurona de una red neuronal pulso acoplada tomada de [11].

El modelo clásico de la red neuronal pulso acoplada (PCNN Pulse coupled neural network) tiene 8 parámetros ( $\alpha_F, V_F, \alpha_L, V_L, \alpha_\theta, V_\theta, \beta y N$ ) y 2 matrices de parámetros ( $M$  y  $W$ ) 3 escalares variables ( $n, i, j$ ) y 5 matrices variables ( $F_{ij}, L_{ij}, U_{ij}, Y_{i,j}$  y  $\theta_{ij}$ ) donde  $n$  corre en el intervalo  $[0, N]$  y  $N$  es el total de neuronas. Todas las matrices variables son inicializadas en 0 excepto la matriz de umbral dinámico.

Las matrices  $M$  y  $W$  para este trabajo se toman iguales  $M = W = K$  donde  $K$  esta definido por:

$$K_{ij} = \begin{cases} 0 & \text{si } i=m \text{ y } j=m \\ \frac{1}{r} & \text{Para el resto} \end{cases} \quad (2.42)$$

Donde  $r$  es la distancia al elemento  $ij$  y  $m$  es la mitad de la dimensión del núcleo [36],[25].

### 2.3.6. Máquina de soporte vectorial

La máquina de soporte vectorial (SVM support vector Machine) es un arreglo que busca que la clase correcta para cada entrada tenga un valor mayor que las clases incorrectas por un margen fijo  $\Delta$ . Antes de definir una SVM es necesario definir los siguientes conceptos:

#### Función de puntuación

La función de puntuación mapea los valores de entrada a un puntaje de confianza es decir a puntuaciones de clases. La función más simple de puntuación es un mapeo lineal.

$$f(x_i, w, b) = wx_i + b \quad (2.43)$$

En la ecuación anterior se asume que la entrada  $x_i$  contiene todos los rasgos en un vector columna  $[D, 1]$ . La matriz  $w$  es de tamaño  $[K * D]$  y el vector  $x_i$  es de tamaño  $[K * 1]$ . Los parámetros de  $w$  son llamados pesos y  $b$  es llamado el vector vías por que este influye en puntaje de salida sin interactuar con la entrada. Note que cada fila de  $w$  actúa como un clasificador.

### **Función de costo**

Es una función que mide el desempeño del clasificador, de manera intuitiva la perdida va a ser alta si se realiza una mala clasificación y baja de lo contrario.

### **SVM**

Una máquina de vector soporte busca que la función de puntuación separe las clases por al menos un  $\Delta$ , para esto ocupa la función de costo la cual evalúa si la separación se cumple de lo contrario acumula perdida.

## 2.4. Métricas de desempeño

Métricas de desempeño Una vez entrenado un clasificador es necesario evaluar el comportamiento que este presenta frente a otros. Una de las formas más comunes de hacerlo es utilizando la exactitud del clasificador la cual se define en la ecuación 2.44 [37].

$$exactitud = \frac{\text{número de ejemplos correctamente clasificados}}{\text{Total de ejemplos}} \quad (2.44)$$

### 2.4.1. Micro y macro $F_1$

La métrica  $F_1$  balancea la precisión y la repetitividad en el sentido de que asigna el mismo peso a cada una. Su porcentaje se maximiza cuando los porcentajes de precisión y exactitud son iguales y disminuye en caso contrario [37],[38].

$$F_1 = 2rp/(r + p) \quad (2.45)$$

Donde r representa la repetitividad y p la precisión.

Las métricas precisión y repetitividad se pueden definir de 2 manera estas reciben el nombre de promedio micro y promedio macro [38].

Precisión y repetitividad micro.

$$p = \frac{TP}{TP + FP} \quad (2.46)$$

$$r = \frac{TP}{TP + FN} \quad (2.47)$$

Donde TP, FP y FN se definen a partir de la matriz de confusión mostrada en la tabla 2.1.

Clase i		Juicio Experto	
		1	0
C	1	TP	FP
	0	FN	TN

Tabla 2.1: Matriz de confusión. C representa la salida del clasificador entrenado.

Para calcular la precisión y repetitividad macro se calculan primero las métricas para cada clase individual se suman y se dividen entre el total de clases. Note que si se habla de

solo 2 clases no tiene sentido de hablar de esta métrica pues el valor es el mismo que el micro.

$$P = \frac{\sum_{i=1}^l p_i}{NC} \quad (2.48)$$

$$R = \frac{\sum_{i=1}^l r_i}{NC} \quad (2.49)$$

Donde NC es el número de clases a evaluar.

### 2.4.2. Validación cruzada

La validación cruzada consiste en separar el conjunto de datos en sub-conjuntos que contengan la misma cantidad de elementos donde cada uno de los sub-conjuntos será utilizado para probar el clasificador mientras el resto se utilizaran para entrenarlo de esta manera si se seleccionan 5 conjuntos de validación se entrenara al clasificador 5 veces. La exactitud promedio de los conjuntos de validación nos indican la robustez del clasificador [39].

---

# Capítulo 3

## Desarrollo

El proceso que se sigue para clasificar un objeto a partir de una imagen parte de la adquisición de esta. El sistema de adquisición debe suprimir la mayor cantidad de componentes no deseados de tal forma que el objeto deseado sea fácil de aislar. Se debe obtener un conjunto de imágenes el cual debe ser subdividido en 2 subconjuntos un conjunto de entrenamiento y uno de prueba. El conjunto de entrenamiento es utilizado para sintonizar los parámetros del clasificador. El clasificador utiliza los rasgos extraídos de cada imagen. Para extraer los rasgos es necesario aislar el objeto de interés. Para separar el objeto de interés se pueden aplicar filtros con el objetivo de resaltar el objeto. Una vez entrenado el clasificador, se evalúa utilizando el conjunto de prueba. El diagrama se observa en la figura 3.1.

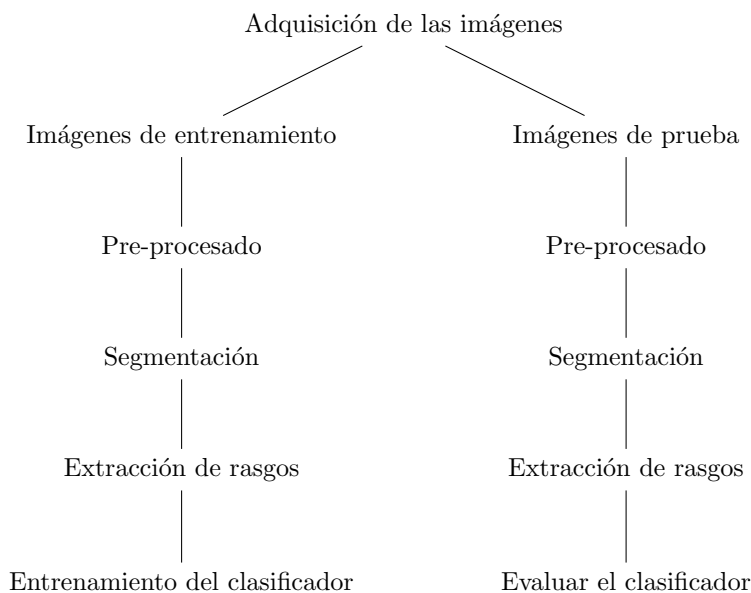


Figura 3.1: Diagrama general para clasificación del ciclo estral.

### 3.1. Medio de adquisición de imágenes

Como medio de adquisición se utilizó en primera instancia un microscopio de la marca NATIONAL en conjunto con la cámara de la Raspberry y una Raspberry 2B. Se diseñó un acoplamiento al ocular del microscopio el cual se implementó utilizando PLA y una impresora 3d, en la figura 3.2 se observa el modelo en SolidWorks 2015, en la figura 3.3 se observa el sistema implementado y en la figura 3.4 se observa una foto capturada con él.

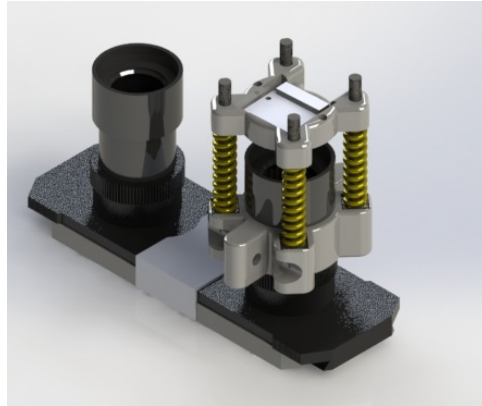


Figura 3.2: Modelo del sistema con la cámara de la Raspberry



Figura 3.3: Implementación del sistema con la cámara de la Raspberry

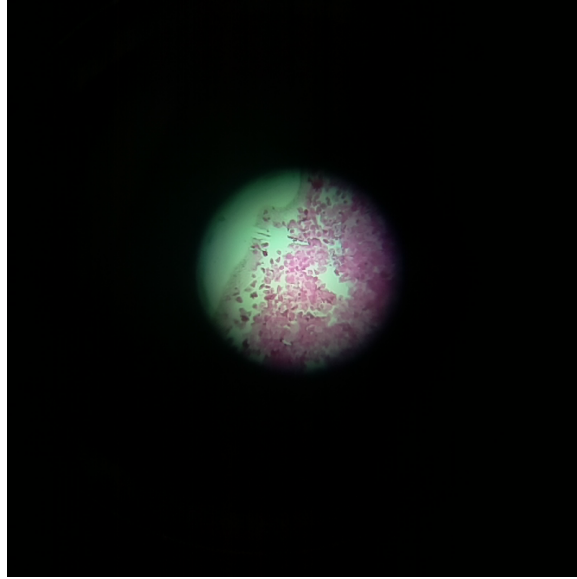


Figura 3.4: Imagen obtenida con la cámara de la Raspberry

Para la segunda versión se conservó la base del sistema previo sustituyendo la cámara por la cámara Logitech C120 puesto que esta cámara es compatible con los sistemas operativos basados en Linux además de poseer una resolución de 5mp mejorada por software, en la figura 3.6 se observa una imagen adquirida por este sistema. El problema que presentaba es que no aislaba la iluminación ambiental entre la cámara y el ocular lo cual provocaba ruido en la imagen y una iluminación no uniforme sin embargo aprovechaba mejor la cantidad de píxeles presentes en la imagen. Se solucionó implementando una extrusión que rodea la cámara hasta el ocular además del control de la apertura de la cámara mediante los drivers propios de esta, con lo cual se logró mejorar la imagen. En la figura 3.5 se observa el sistema actual y en la figura 3.7 una imagen capturada con este sistema.

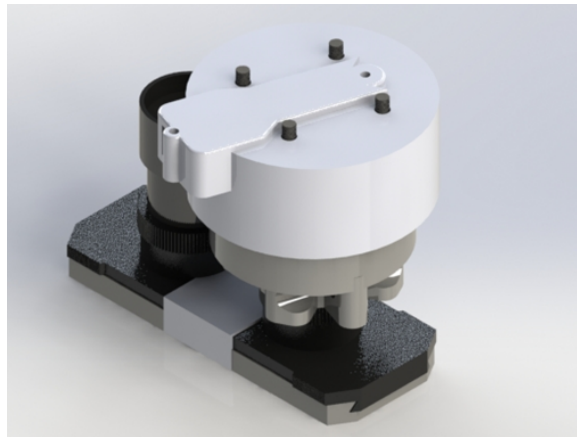


Figura 3.5: Sistema de adquisición con cámara c120

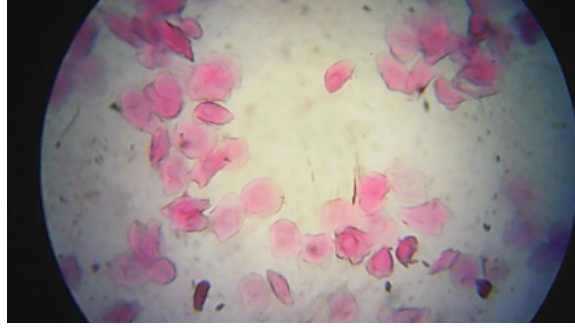


Figura 3.6: Imagen sin corrección de iluminación

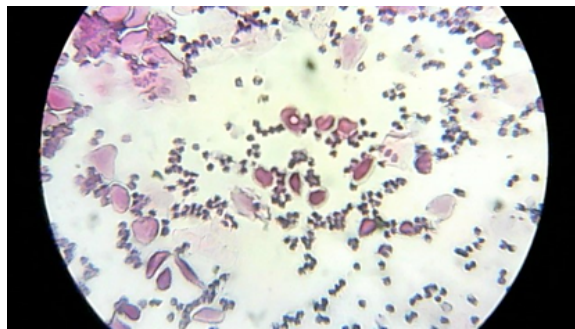


Figura 3.7: Imagen con corrección de iluminación

## 3.2. Versiones

Las versiones presentadas muestran la evolución del trabajo de investigación en torno al ciclo estral, a los algoritmos de procesamiento de imágenes y a los algoritmos de inteligencia artificial. Si bien en las primeras versiones (1,2) no se alcanzaron resultados favorables su implementación permitió comprender conceptos que nos llevarían a mejorar los resultados en la versión 3. La versión 2 motivó el uso de clasificadores sofisticados como lo son las máquinas de vector soporte y la red neuronal perceptrón multicapa, esto debido a que las características arrojadas por la red PCNN presentaban un fuerte traslape entre ellas lo cual complicaba su clasificación por métodos simples. La implementación de la red PCNN nos encauso a buscar formas de paralelizar nuestro código, esto dio pie a investigar las librerías de uso de GPU particularmente CUDA, si bien esto no se implementó para esta red, fue de gran utilidad cuando se abordó la red neuronal MLP utilizando la librería keras la cual realiza el entrenamiento a partir de librerías que manipulan directamente el GPU (Theano, Caffe, Tensorflow, etc). El motivo por el cual se decidió utilizar la red PCNN fue que esta entrega características de forma invariantes a traslación y rotación. La red PCNN también

es capaz de entregar características de textura, pero presenta la desventaja de no entregar las 2 al mismo tiempo. A diferencia de la red MLP la red PCNN no presenta un algoritmo de entrenamiento, esto hace que la sintonización sea un trabajo complicado. Tras descartar a la red PCNN como el algoritmo usado para extraer características, se retomó el trabajo previo entorno a las células PAP y el marco referencial acerca de algoritmos para extraer características. El problema se convirtió primero en encontrar que características comparten las imágenes de cada etapa del ciclo estral entre una misma etapa, pero no entre diferentes. Se descartaron características asociadas al color puesto que la tinción no es homogénea. Se descartaron características del tipo SIFT (Scale-invariant feature transform) esto debido a que nuestro objetivo no es clasificar un objeto que presente una morfología continua. Finalmente se lograron identificar las siguientes características. La primera en identificar fue la forma de la imagen segmentada, si bien la forma de una célula no representa una característica discriminante si lo hace el promedio de las formas segmentadas de cada etapa del ciclo, para representar numéricamente este valor se usó el factor de compacidad promedio. El problema de obtener 2 promedios de formas iguales en etapas diferente se solucionó al usar la suma de los factores de compacidad de cada forma presente en la imagen y el número de elementos encontrados en la imagen segmentada. Los nuevos rasgos permitieron diferenciar las etapas Metaestro y Diestro de las etapas Proestro y Estro por lo cual se optó por incorporar métricas que midieran la textura de las imágenes segmentadas incorporando el algoritmo GLCM.

### 3.2.1. Versión 1: Rasgos del histograma clasificados utilizando K-NN

En esta versión se inició con un proceso de segmentación manual sobre los canales R y B del espacio de color RGB. Una vez segmentada la imagen se transformó a escala de grises utilizando la ecuación 3.1 y se obtuvo el histograma de la imagen. El histograma se utilizó como el vector de rasgos para posteriormente realizar la clasificación utilizando el algoritmo K-NN.

$$G = 0.1 * R + 0.15 * G + 0.75 * B \quad (3.1)$$

#### Algoritmo 1: Histograma y K-NN

```

Entradas: Imágenes de entrenamiento, Imágenes de prueba
Para cada imagen de entrenamiento
    Aplicar el umbral manual al canal R lo cual genera
    la máscara MR
    Aplicar el umbral manual al canal B lo cual genera
    la mascara MB
    Segmentar utilizando la unión de las máscaras
    MR y MB.
    Obtener el Histograma de la imagen
Para cada imagen de prueba
    Aplicar el umbral manual al canal R lo cual genera
    la máscara MR
    Aplicar el umbral manual al canal B lo cual genera
    la mascara MB
    Segmentar utilizando la unión de las máscaras
    MR y MB.
    Obtener el Histograma de la imagen
    Clasificar con K-NN utilizando las imágenes de
    entrenamiento

```

### 3.2.2. Versión 2: Rasgos PCNN

#### Dataset

Se utilizaron 64 imágenes del ciclo estral, 16 por cada etapa, 32 imágenes fueron utilizadas para entrenar el clasificador y 32 para probarlo. Se dividió el conjunto de acuerdo con la receptividad del roedor en cada etapa, formando las siguientes clases:

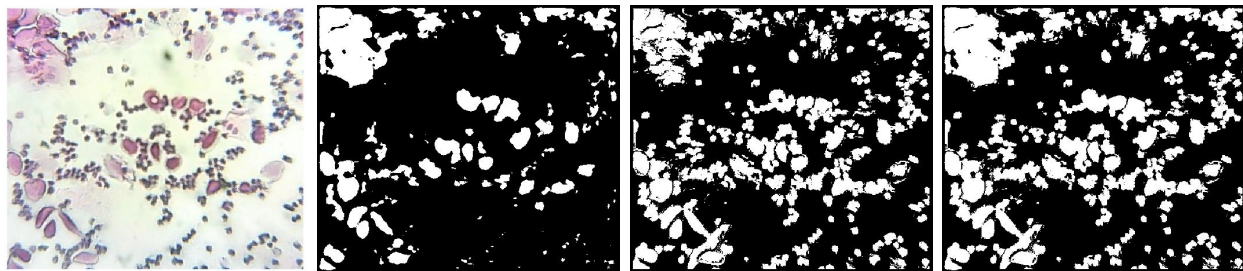
- Clase 1 formada por Proestro, Estro.
- Clase 2 formada por Metaestro, Diestro.

## Pre-procesado

La imagen se convirtió a escala de grises y al espacio de color CIELAB. La imagen en escala de grises se invirtió para dejar el fondo en una tonalidad oscura.

## Segmentación

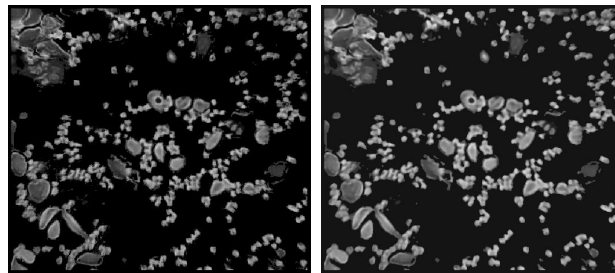
Se aplicó el umbralado por entropía cruzada en el canal A y la escala de grises (Figuras 3.8(b) y 3.8(c)) con la unión de los resultados se generó una máscara que serviría para segmentar la imagen. Note que el umbralado en el canal A no es capaz de segmentar todas las células y el umbralado en la escala de grises no segmenta partes internas de éstas, sin embargo se complementan la una a la otra.



(a) Diestro (b) Diestro umbralado a partir del canal A del espacio CIELAB (c) Diestro umbralado a partir de la escala de grises (d) Mascarera generada con la unión de el umbralado a partir del canal A y la escala de grises.

Figura 3.8: Segmentación de la etapa Diestro a partir del espacio de color CIELAB y la escala de grises.

Las imágenes se normalizaron restándole el promedio de los píxeles que representan a las células y dividiéndola entre el valor máximo de estos. Después se cambió de escala sumándole el valor mínimo de la imagen y multiplicando por 255. Esta adecuación se realizó para facilitar la sintonización de la red PCNN al tener todas las imágenes en los mismos rangos (Fig. 3.9).

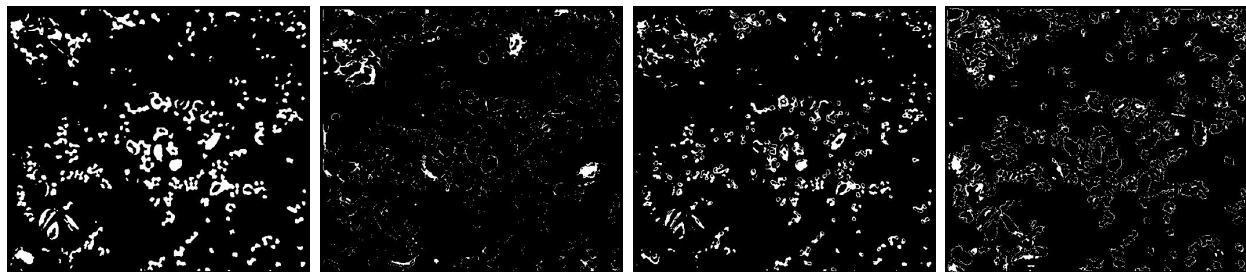


(a) Resultado de la segmentación (b) Resultado de la normalización

Figura 3.9: Imagen normalizada adecuada para utilizarla dentro de la red PCNN.

### Extracción de rasgos

Como rasgos se utilizaron 15 iteraciones de la red neuronal PCNN con las cuales se generó la serie de tiempo. La serie de tiempo se genera a partir de la suma de los pulsos obtenidos (los pulsos son los pixeles con valor de 1) en cada iteración. En este caso antes de obtener la suma se multiplica la salida por la máscara utilizada para segmentar la imagen, esto con la intención de eliminar la influencia de los pulsos externos a las células. Cada rasgo se normaliza dividiéndolo entre la cantidad de pixeles pertenecientes a la máscara con valor de 1, esto se realizó para tener los rasgos en la misma escala.



(a) Iteración 2 (b) Iteración 5 (c) Iteración 7 (d) Iteración 10.

Figura 3.10: Salida de la red PCNN para las iteraciones 2, 5, 7 y 10.

### Clasificación

Se utilizó una SVM lineal<sup>1</sup> variando el parámetro de penalización  $C$  en el rango de  $10^1$  a  $10^6$  siguiendo una distribución logarítmica.

<sup>1</sup>Se denomina lineal porque su función de puntuación también conocida como núcleo es lineal.

### 3.2.3. Versión 3: Rasgos de textura - forma

#### Dataset

Se utilizaron 32 imágenes del ciclo estral, ocho por cada etapa. Las imágenes fueron divididas en 16 sub-imágenes. Las 512 imágenes resultantes fueron etiquetadas por personal del bioterio Claude Bernard resultando un total de 412 imágenes (Fig. 3.11): 112 de la etapa Estro, 89 de la etapa Diestro, 125 de la etapa Metaestro y 86 de la etapa Proestro.

El dataset se utilizó para realizar dos pruebas:

- Primera prueba utilizo cada etapa del ciclo estral como una clase.
- Segunda prueba se dividió el conjunto de acuerdo con la receptividad del roedor en cada etapa, formando las siguientes clases:
  - Clase 1 formada por Proestro, Estro.
  - Clase 2 formada por Metaestro, Diestro.

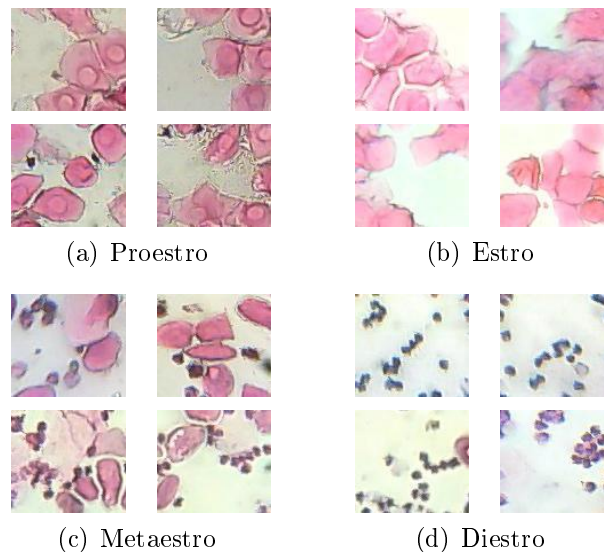
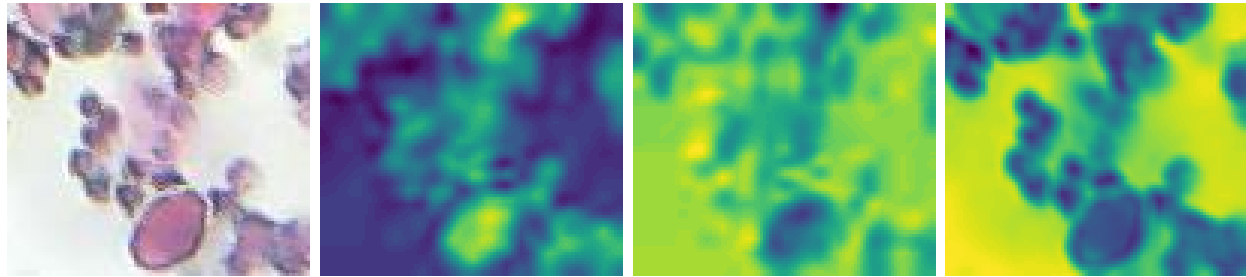


Figura 3.11: Imágenes pertenecientes al Dataset. Las imágenes fueron proporcionadas por el bioterio Claude Bernard.

#### Pre-procesado

Una vez adquirida la imagen es convertida al espacio de color CIELAB y la escala de grises. Del espacio CIELAB se toman los canales A y B. El uso del canal A se debe a que este representa una variación del color rojo al verde lo cual es favorable para las células epiteliales

que están teñidas de una tonalidad rojiza. El canal B es usado ya que representa una variación del color azul al amarillo esto es útil para la segmentación de leucocitos los cuales son teñidos en tonalidades de azul. La escala de grises se usa ya que conserva características tanto de leucocitos como de células epiteliales. Una vez la imagen convertida al espacio de color se le aplica un filtro promedio con una ventana de  $[5,5]$  para eliminar el ruido y suavizar la imagen (Fig. 3.12).



(a) Metaastro (b) Metaastro en el canal A del espacio CIELAB (c) Metaastro en el canal B del espacio CIELAB (d) Metaastro en la escala de grises

Figura 3.12: Imágenes en los espacios de color CIELAB y escala de grises para la etapa Metaastro después de aplicar el filtro promedio

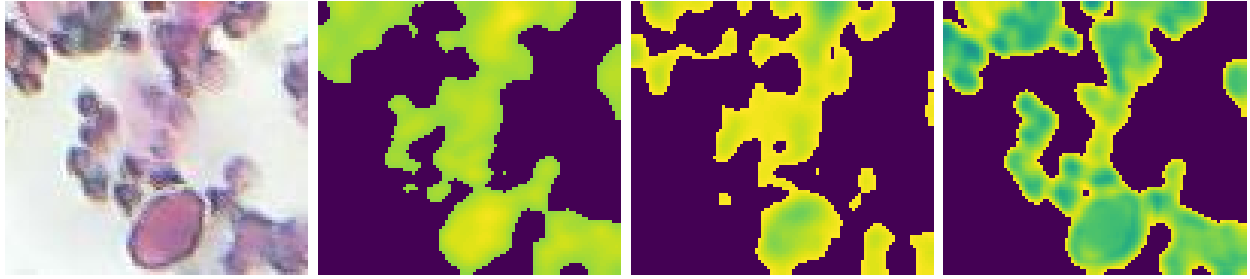
### Segmentación

Para la segmentación se aplicó el algoritmo de umbralado de Otsu esto se realizó para aislar las células del fondo y eliminar el ruido asociado a este. Se optó por Otsu pues de los algoritmos que mostraron buen desempeño Otsu obtuvo el menor tiempo de computo (Tabla 3.1) los resultados de los algoritmos evaluados se pueden observar en el Apéndice B.

Algoritmo	Tiempo (s)
Entropía Cruzada	0.029s
Otsu	0.014s
K-Means	2.171s
K-MeansRGB	106.233s

Tabla 3.1: Tiempos de ejecución para una sola imagen en escala de grises en el caso de K-means se consideran una  $k$  de 5 y solo los últimos tres conjuntos son tomados, en el caso de K-means en RGB se considera una  $k$  de 5 y solo se toman los 4 primeros conjuntos.

El umbralado se aplicó en los canales A y B del espacio de color CIELAB, y la escala de grises una vez aplicado el pre-procesado los resultados se muestran en la figura 3.13.



(a) Metaastro (b) Metaastro segmentado en el canal A del espacio CIELAB (c) Metaastro segmentado en el canal B del espacio CIELAB (d) Metaastro segmentado en la escala de grises

Figura 3.13: Imágenes segmentadas en los espacios de color CIELAB y escala de grises para la etapa Metaastro.

### Extracción de rasgos

Los rasgos de textura se obtuvieron de la matriz de concurrencia a partir de la imagen segmentada para los ángulos  $\theta = [0, 45, 90, 135]$  y los pasos  $\delta = [1, 5]$ :

$$G(i, j, 1, 0) = \#((k, l), (m, n)) \in (L_y, L_x) \quad (3.2)$$

$$(L_y, L_x) | k - m = 0, |l - n| = 1, I(k, l) = i, I(m, n) = j$$

$$G(i, j, 1, 45) = \#((k, l), (m, n)) \in (L_y, L_x) \quad (3.3)$$

$$(L_y, L_x) | (k - m = 1, l - n = -1),$$

$$o(k - m = -1, l - n = 1) I(k, l) = i, I(m, n) = j$$

$$G(i, j, 1, 90) = \#((k, l), (m, n)) \in (L_y, L_x) \quad (3.4)$$

$$(L_y, L_x) | |k - m| = 1, l - n = 0, I(k, l) = i, I(m, n) = j$$

$$G(i, j, 1, 135) = \#((k, l), (m, n)) \in (L_y, L_x) \quad (3.5)$$

$$(L_y, L_x) | (k - m = 1, l - n = 1),$$

$$o(k - m = -1, l - n = -1), I(k, l) = i, I(m, n) = j$$

$$\begin{aligned}
G(i, j, 5, 0) &= \#((k, l), (m, n)) \in (L_y, L_x) \\
(L_y, L_x) &|k - m = 0, |l - n| = 5, I(k, l) = i, I(m, n) = j
\end{aligned} \tag{3.6}$$

$$\begin{aligned}
G(i, j, 5, 45) &= \#((k, l), (m, n)) \in (L_y, L_x) \\
(L_y, L_x) &|(k - m = 5, l - n = -5), \\
o(k - m = -5, l - n = 5) &I(k, l) = i, I(m, n) = j
\end{aligned} \tag{3.7}$$

$$\begin{aligned}
G(i, j, 5, 90) &= \#((k, l), (m, n)) \in (L_y, L_x) \\
(L_y, L_x) &||k - m| = 5, l - n = 0, I(k, l) = i, I(m, n) = j
\end{aligned} \tag{3.8}$$

$$\begin{aligned}
G(i, j, 5, 135) &= \#((k, l), (m, n)) \in (L_y, L_x) \\
(L_y, L_x) &|(k - m = 5, l - n = 5), \\
o(k - m = -5, l - n = -5) &I(k, l) = i, I(m, n) = j
\end{aligned} \tag{3.9}$$

En total se calcularon 8 matrices de concurrencia (Ecuaciones 3.2 a 3.9) donde para cada una se calcularon las métricas: Contraste, Disimilitud, Homogeneidad y Energía (Sección 2.2.5). Dando 32 características por cada canal del espacio de color CIELAB y la escala de grises. Siendo un total de 96 rasgos de textura.

Como rasgos de forma se utilizó el factor de compacidad promedio, la compacidad total (Sección 2.2.5) y el número de elementos (Sección 2.2.4). Dando 3 características por cada canal del espacio de color CIELAB y la escala de grises. Siendo 9 rasgos de forma y un total de 105 rasgos por imagen.

## Clasificación

Para validar la clasificación se utilizó el método de validación cruzada utilizando 5 conjuntos. Se probaron 5 clasificadores: 4 SVM (lineal, polinomial, base radial, y la propuesta por Cramer y Singer) y la red neuronal MLP. Los parámetros de cada clasificador se obtuvieron de la siguiente forma:

- SVM lineal variando el parámetro C en el rango de  $10^1$  a  $10^6$  siguiendo una distribución logarítmica.

- SVM propuesta por Cramer y Singer variando el parámetro C en el rango de  $10^1$  a  $10^6$  siguiendo una distribución logarítmica.
- SVM de base radial variando  $\gamma$  y C obteniendo 10,000 combinaciones donde los valores fueron obtenidos con una distribución logarítmica donde  $\gamma$  obtuvo valores entre  $10^{-2}$  a  $10^1$  y C entre  $10^2$  a  $10^4$ . Por cada valor de  $\gamma$  (100 valores) se usaron 100 valores de C. El núcleo de la SVM se muestra en la ecuación 3.10.

$$-\gamma\|x - x'\|^2 \tag{3.10}$$

- MVZ polinomial variando el grado del polinomio,  $\gamma$  y C obteniendo 2,700 combinaciones. Los valores fueron obtenidos con una distribución logarítmica donde para cada grado del polinomio [1,2,3] se obtuvieron 30 valores de C en el rango de  $10^2$  a  $10^4$ , para cada valor de C se usaron 30 valores de  $\gamma$  en el rango  $10^{-2}$  a  $10^1$ . El núcleo de la SVM se observa en la ecuación 3.11.

$$(\gamma\langle x, x' \rangle)^d \tag{3.11}$$

- Red neuronal perceptrón multicapa. Los parámetros fueron obtenidos de manera aleatoria entrenando 16,000 modelos. Los rangos de los parámetros se muestran en la tabla 4.2.

Híper parámetro	Rango
Tasa de aprendizaje	0.005 a 0.7
Numero de neuronas por capa	1 a 150
Numero de capas	1 a 5
Numero de épocas	600 a 2000

Tabla 3.2: Valores de la red MLP

La normalización se realizó utilizando el conjunto de entrenamiento de la siguiente forma. Considere una matriz de  $n \times m$  donde  $n$  representa el número de filas y cada fila representa los rasgos de una imagen. El índice  $m$  representa el número de columnas y cada columna representa un rasgo. Considere la matriz  $L$  como la matriz de rasgos de los ejemplos de entrenamiento sin normalizar. Considere la matriz  $P$  como una matriz de dimensión  $m \times n$  donde cada elemento de cada columna representa el promedio de la columna de la matriz  $L$ . Considere la matriz  $F$  de dimensiones  $m \times n$  donde cada elemento de cada columna posee el

valor máximo de la columna asociada a la matriz  $L$  entonces la matriz de rasgos normalizada  $D$  se define como:

$$D = (L - P) ./ F \quad (3.12)$$

Donde la división se realiza elemento a elemento.

### 3.3. Aplicación

Las primeras 2 versiones fueron implementadas en la plataforma Matlab 2016b. La versión 3 fue implementada en Python 3 utilizando las librerías Sklearn [40], Skimage [32], Opencv [24], Keras (Para la red MLP) [41] y numpy en el entorno de desarrollo anaconda utilizando el Ide Spyder [42]. La interfaz de usuario se desarrolló utilizando eclipse bajo el lenguaje de programación Java con la librería windowbuilder para el entorno gráfico y Python 3 como el núcleo de procesamiento.

---

# Capítulo 4

## Resultados

### 4.1. Versiones 1,2

Los resultados de las primeras 2 versiones fueron menores al 60% en exactitud. Para la segunda versión los resultados de la segmentación se muestran en el apéndice B. Estos resultados llevaron a proponer una variación del algoritmo K-Means (K-MeansP) basada en el histograma de la imagen. El algoritmo se comparó contra K-Means ++ y K-Means con semillas uniformes (K-MeansU). La figura 4.1 muestra el tiempo de ejecución en segundos contra el tamaño de la imagen, la gráfica muestra una gran diferencia en tiempos de ejecución para imágenes grandes. La tabla 4.1 muestra los centroides alcanzados por cada algoritmo para distintas resoluciones, en esta tabla se observa cómo los 3 algoritmos convergen a centroides similares.

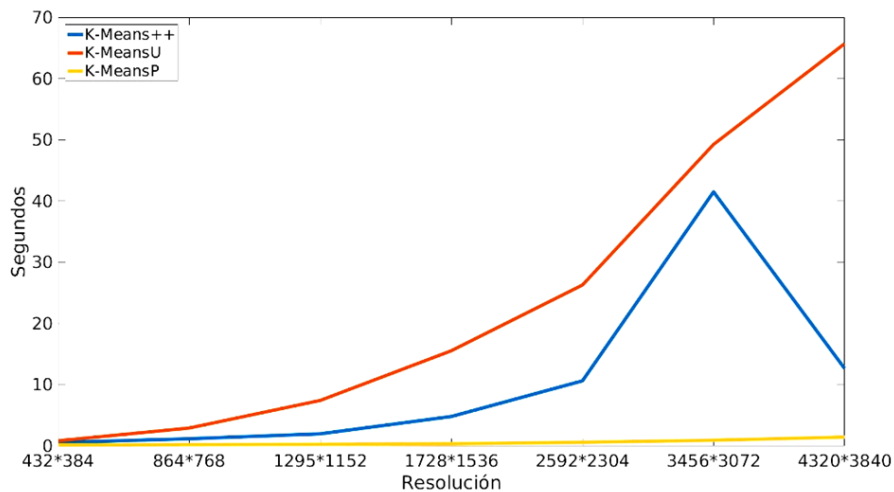


Figura 4.1: Tiempo de ejecución en segundos vs tamaño de imagen

		Resolución								
		4320*3840			2592*2304			432*384		
		K-Means++	K-MeansU	K-MeansP	K-Means++	K-MeansU	K-MeansP	K-Means++	K-MeansU	K-MeansP
Centroides	C1	116.11	116.11	116.01	123.63	115.03	115.03	116.11	116.11	116.11
	C2	157.33	157.33	157.33	170.53	156.91	156.91	157.33	157.33	157.33
	C3	192.38	192.38	192.37	204.25	192.39	192.39	192.38	192.38	192.38
	C4	221.20	221.20	221.25	229.83	221.26	221.26	221.20	221.20	221.20
	C5	244.70	244.70	244.27	246.59	244.32	244.32	244.70	244.70	244.70

Tabla 4.1: Comparativa de los resultados en diferentes resoluciones.

Los rasgos para la versión 2 se obtuvieron a través de las series de tiempo asociadas a la red PCNN las figuras 4.2, 4.3, 4.4 y 4.5 muestran las series de tiempo obtenidas para cada etapa del ciclo estral. Si bien las series de tiempo son favorables para las etapas Diestro y Metaestro, no son adecuados para el resto, pues no presentan una repetitividad entre imágenes de la misma clase. Aun si las imágenes presentaran repetitividad entre las clases Proestro y Estro no sería suficiente para discernir entre las cuatro clases pues las series de tiempo para las etapas Diestro y Metaestro presentan comportamientos similares.

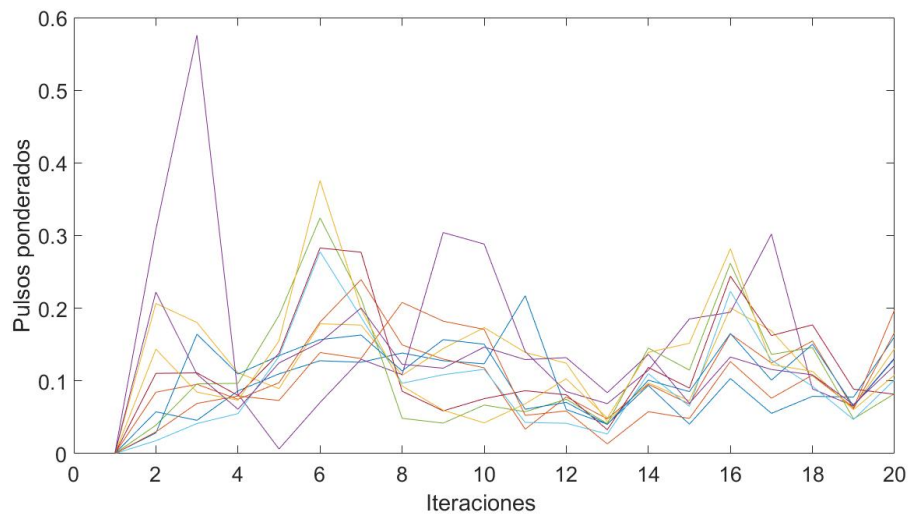


Figura 4.2: Series de tiempo para la etapa Proestro.

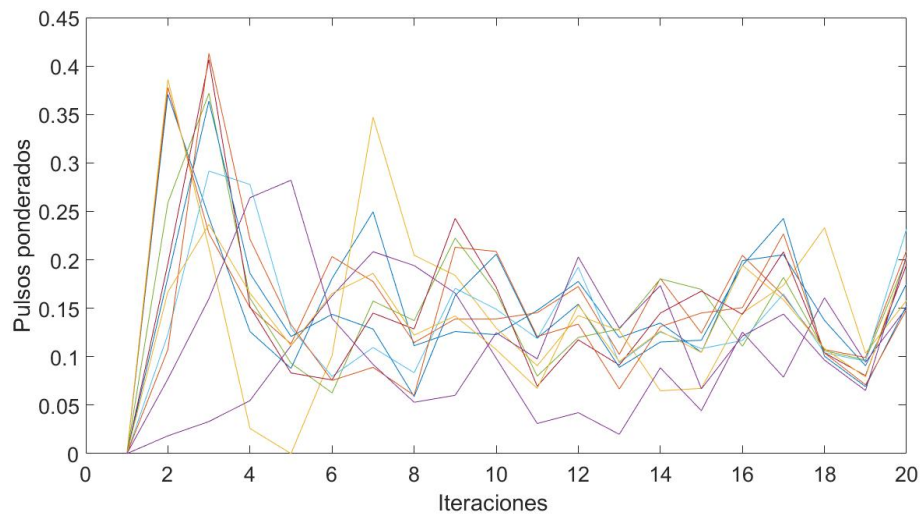


Figura 4.3: Series de tiempo para la etapa Estro.

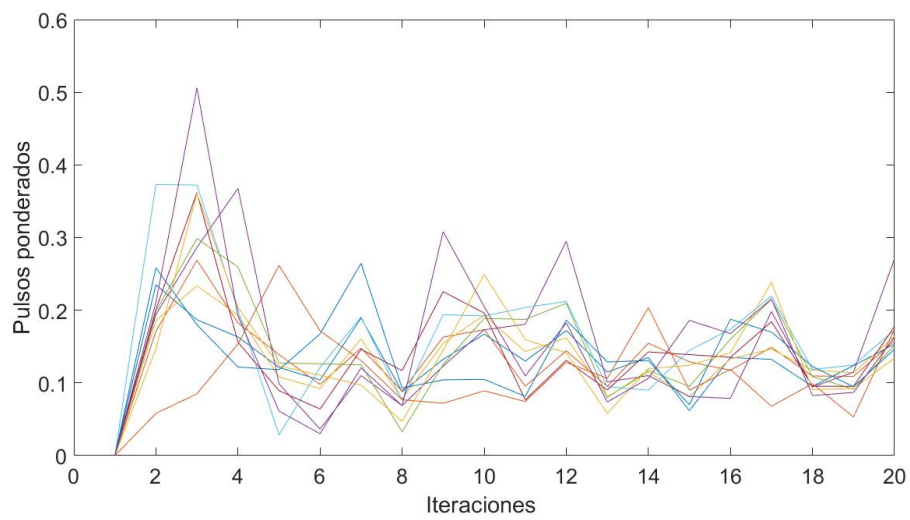


Figura 4.4: Series de tiempo para la etapa Metaestro.

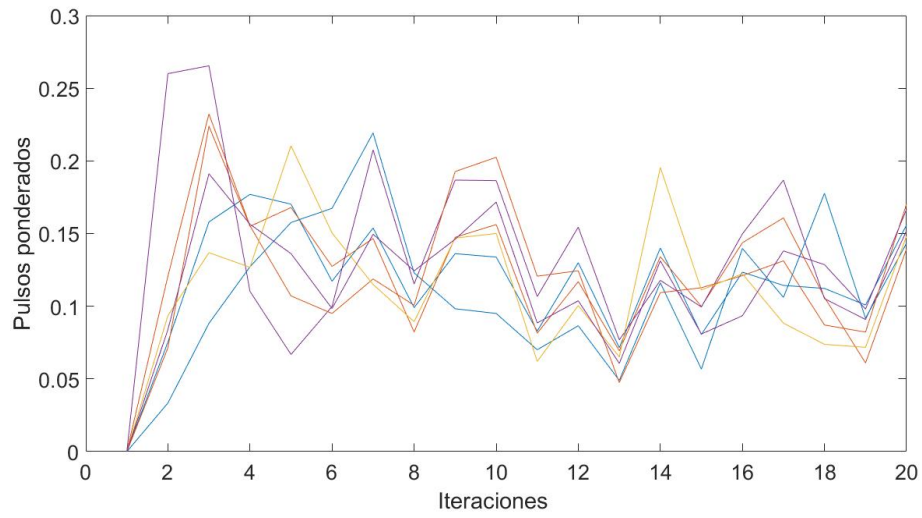


Figura 4.5: Series de tiempo para la etapa Diestro.

## 4.2. Versión 3: Rasgos de textura - forma

Para la versión 3 se cambiaron las características por métricas de textura y forma. El conjunto de validación que dio mejor resultado en la clasificación de las cuatro etapas del ciclo estral estaba conformado por 82 imágenes de prueba y 330 de entrenamiento para los 5 clasificadores. Los conjuntos de prueba y entrenamiento que dieron mejores resultados en la clasificación de las 2 clases definidas a partir de la receptividad del roedor a la reproducción se muestran en la tabla 4.2.

	Imágenes de entrenamiento	Imágenes de prueba
SVM lineal	83	329
SVM propuesta por Cramer y Singer	83	329
SVM de base radial	82	330
SVM polinomial	83	329
MLP	83	329

Tabla 4.2: Distribución de los conjuntos de validación que dieron el mejor resultado para el caso de las 2 clases para los 5 clasificadores.

### 4.2.1. SVM lineal

El valor de C que dio el mejor resultado fue 60.36 con el cual se alcanzó una exactitud de 82.92 % en prueba y 87.88 % en entrenamiento. Los porcentajes de exactitud obtenidos para cada etapa fueron: 64.71 % para la etapa Diestro, 75 % para la etapa Metaestro, 100 % para la etapa Estro y 83.33 % para la etapa Proestro. Los resultados se validaron usando validación cruzada [37] utilizando 5 conjuntos. La tabla 4.3 muestra el mejor resultado de los conjuntos de validación y la tabla 4.4 muestra el promedio.

T/R	Diestro	Metaestro	Estro	Proestro
Diestro	<b>11</b>	6	0	0
Metaestro	5	<b>20</b>	0	0
Estro	0	0	<b>22</b>	0
Proestro	0	3	0	<b>15</b>

Tabla 4.3: Matriz de confusión con el mejor resultado para las 4 etapas del ciclo estral utilizando la SVM lineal

T/R	Diestro	Metaestro	Estro	Proestro
Diestro	<b>11</b>	5.8	0.4	0.6
Metaestro	7.6	<b>16</b>	0	1.4
Estro	0.8	0.2	<b>20.6</b>	0.8
Proestro	0	2.6	1	<b>13.6</b>

Tabla 4.4: Matriz de confusión promedio para las 4 etapas del ciclo estral utilizando la SVM lineal

Para el caso de 2 clases el valor de C que obtuvo el mejor desempeño fue 45.25 con una exactitud de 98.18 % en entrenamiento y 97.59 % en prueba. La exactitud en prueba para la clase 1 fue de 95 % y para la clase 2 del 100 %. La tabla 4.5 muestra el mejor desempeño de los conjuntos de validación y la tabla 4.6 muestra el promedio.

T/R	C2	C1
C2	<b>43</b>	0
C1	2	<b>38</b>

Tabla 4.5: Matriz de confusión con el mejor resultado para el caso de 2 clases utilizando la SVM lineal

T/R	C2	C1
C2	<b>40.8</b>	2
C1	2	<b>37.6</b>

Tabla 4.6: Matriz de confusión promedio para el caso de 2 clases utilizando la SVM lineal

#### 4.2.2. SVM propuesta por Cramer y Singer

El valor de C que dio el mejor resultado fue 93.96 con el cual se alcanzó una exactitud de 80.49 % en prueba y 91.21 % en entrenamiento. Los porcentajes obtenidos para cada etapa fueron: 70.59 % para la etapa Diestro, 72 % para la etapa Metaestro, 95.45 % para la etapa Estro y 83.33 % para la etapa Proestro. Los resultados se validaron usando validación cruzada [37] utilizando 5 conjuntos. La tabla 4.7 muestra el mejor resultado de los conjuntos de validación y la tabla 4.8 muestra el resultado promedio.

T/R	Diestro	Metaestro	Estro	Proestro
Diestro	<b>12</b>	5	0	0
Metaestro	6	<b>18</b>	0	1
Estro	0	0	<b>21</b>	1
Proestro	0	3	0	<b>15</b>

Tabla 4.7: Matriz de confusión con el mejor resultado para las 4 etapas del ciclo estral utilizando la SVM propuesta por Cramer y Singer

T/R	Diestro	Metaestro	Estro	Proestro
Diestro	<b>11.4</b>	5.6	0.2	0.6
Metaestro	6.6	<b>16.6</b>	0.2	1.6
Estro	0	0	<b>20.8</b>	1.6
Proestro	0.2	2	1.2	<b>13.8</b>

Tabla 4.8: Matriz de confusión promedio para las 4 etapas del ciclo estral utilizando la SVM propuesta por Cramer y Singer

Para el caso de 2 clases el valor de C que obtuvo el mejor desempeño fue 22.92. Con una exactitud de 98.18 % en entrenamiento y 97.59 % en prueba. La exactitud en prueba para la clase 1 fue de 95 % y para la clase 2 del 100 %. La tabla 4.9 muestra el mejor desempeño de los conjuntos de validación y la tabla 4.10 muestra el promedio.

T/R	C2	C1
C2	<b>43</b>	0
C1	2	<b>38</b>

Tabla 4.9: Matriz de confusión con el mejor resultado para el caso de 2 clases utilizando la SVM propuesta por Cramer y Singer

T/R	C2	C1
C2	<b>40.8</b>	2
C1	2	<b>37.6</b>

Tabla 4.10: Matriz de confusión promedio para el caso de 2 clases utilizando la SVM propuesta por Cramer y Singer

### 4.2.3. SVM de base radial

Los valores de  $C$  y  $\gamma$  que dieron mejores resultados fueron 104.76 y 0.1 respectivamente. Con los cuales se alcanzó una exactitud de 76.83 % en prueba y 92.42 % en entrenamiento. Los porcentajes obtenidos para cada etapa fueron: 66.7 % para la etapa Diestro, 68 % para la etapa Metaestro, 80 % para la etapa Estro y 82.35 % para la etapa Proestro. Los resultados se validaron usando validación cruzada [37] utilizando 5 conjuntos. La tabla 4.11 muestra el mejor resultado de los conjuntos de validación y la tabla 4.12 muestra el resultado promedio.

T/R	Diestro	Metaestro	Estro	Proestro
Diestro	<b>12</b>	6	0	0
Metaestro	7	<b>17</b>	0	1
Estro	0	0	<b>20</b>	2
Proestro	1	2	0	<b>14</b>

Tabla 4.11: Matriz de confusión con el mejor resultado para las 4 etapas del ciclo estral utilizando la SVM de base radial

T/R	Diestro	Metaestro	Estro	Proestro
Diestro	<b>10.8</b>	6	0.2	0.8
Metaestro	9	<b>14.2</b>	0.2	1.6
Estro	1	0.6	<b>19.2</b>	1.6
Proestro	0.6	2.4	0.8	<b>13.4</b>

Tabla 4.12: Matriz de confusión promedio para las 4 etapas del ciclo estral utilizando la SVM de base radial

Para el caso de 2 clases los valores de  $C$  y  $\gamma$  que dieron mejores resultados fueron 890.21 y 0.1748 respectivamente. Con una exactitud de 100 % en entrenamiento y 96.34 % en prueba. La exactitud en prueba para la clase 1 fue de 94.87 % y para la clase 2 del 97.67 %. La tabla 4.13 muestra el mejor desempeño de los conjuntos de validación y la tabla 4.14 muestra el promedio.

T/R	C2	C1
C2	<b>42</b>	1
C1	2	<b>37</b>

Tabla 4.13: Matriz de confusión con el mejor resultado para el caso de 2 clases utilizando la SVM de base radial

T/R	C2	C1
C2	<b>39</b>	3.8
C1	3.2	<b>36.4</b>

Tabla 4.14: Matriz de confusión promedio para el caso de 2 clases utilizando la SVM de base radial

#### 4.2.4. SVM polinomial

Los valores de  $C$ ,  $\gamma$  y grado del polinomio que dieron mejores resultados fueron 1000, 0.6723 y 1 respectivamente. Con los cuales se alcanzó una exactitud de 80.49% en prueba y 96.67% en entrenamiento. Los porcentajes obtenidos para cada etapa fueron: 66.7% para la etapa Diestro, 80% para la etapa Metaestro, 90.9% para la etapa Estro y 82.35% para la etapa Proestro. Los resultados se validaron usando validación cruzada [37] utilizando 5 conjuntos. La tabla 4.15 muestra el mejor resultado de los conjuntos de validación y la tabla 4.16 muestra el resultado promedio.

T/R	Diestro	Metaestro	Estro	Proestro
Diestro	<b>12</b>	6	0	0
Metaestro	3	<b>20</b>	0	2
Estro	0	0	<b>20</b>	2
Proestro	0	3	0	<b>14</b>

Tabla 4.15: Matriz de confusión con el mejor resultado para las 4 etapas del ciclo estral utilizando la SVM polinomial

T/R	Diestro	Metaestro	Estro	Proestro
Diestro	<b>12.6</b>	5	0.2	0
Metaestro	7.6	<b>16</b>	0.2	1.2
Estro	0.4	0.8	<b>20</b>	1.2
Proestro	0.6	1.4	1.6	<b>13.6</b>

Tabla 4.16: Matriz de confusión promedio para las 4 etapas del ciclo estral utilizando la SVM polinomial

Para el caso de 2 clases (Clase 1 Proestro, Estro; Clase 2 Metaestro, Diestro) los valores de  $C$ ,  $\gamma$  y grado del polinomio que dieron mejores resultados fueron 489.39, 0.1 y 1 respectivamente. Con una exactitud de 98.18% en entrenamiento y 97.59% en prueba. La exactitud en prueba para la Clase 1 fue de 95% y para la clase 2 del 100%. La tabla 4.17 muestra el mejor resultado de los conjuntos de validación y la tabla 4.18 muestra el promedio.

T/R	C2	C1
C2	<b>43</b>	0
C1	2	<b>38</b>

Tabla 4.17: Matriz de confusión con el mejor resultado para el caso de 2 clases utilizando la SVM polinomial

T/R	C2	C1
C2	<b>40.6</b>	2.2
C1	2	<b>37.6</b>

Tabla 4.18: Matriz de confusión promedio para el caso de 2 clases utilizando la SVM polinomial

### 4.2.5. MLP

El modelo de la red se muestra en la figura 4.6, con este modelo se alcanzó una exactitud de 87.80 % en prueba y 93.94 % en entrenamiento. Los porcentajes obtenidos para cada etapa fueron: 77.78 % para la etapa Diestro, 80 % para la etapa Metaestro, 95.45 % para la etapa Estro y 100 % para la etapa Proestro. Los resultados se validaron usando validación cruzada [37] utilizando 5 conjuntos. La tabla 4.19 muestra el mejor resultado de los conjuntos de validación y la tabla 4.20 muestra el resultado promedio.

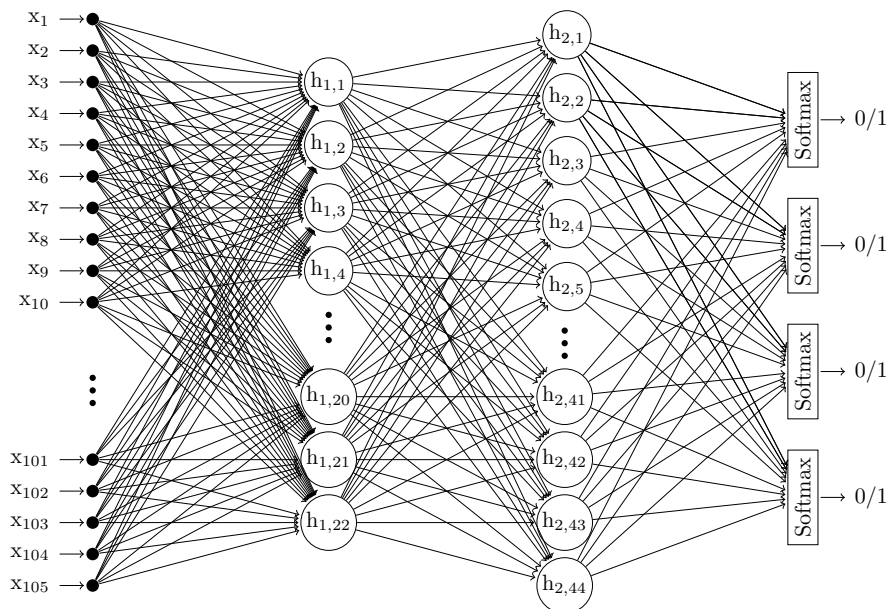


Figura 4.6: Perceptron multicapa: 105 características, 22 neuronas de entrada, 44 neuronas en la capa oculta y 4 clasificadores softmax en la capa de salida. La capa oculta y la capa de entrada usan la función de activación relu.

T/R	Diestro	Metaestro	Estro	Proestro
Diestro	<b>14</b>	4	0	0
Metaestro	5	<b>20</b>	0	0
Estro	0	0	<b>21</b>	1
Proestro	0	0	0	<b>17</b>

Tabla 4.19: Matriz de confusión con el mejor resultado para las 4 etapas del ciclo estral utilizando la red MLP

T/R	Diestro	Metaestro	Estro	Proestro
Diestro	<b>11.2</b>	6	0.2	0.4
Metaestro	3.4	<b>19.8</b>	0.2	1.6
Estro	0.2	0.6	<b>20.2</b>	1.4
Proestro	0.4	1.6	1.4	<b>13.8</b>

Tabla 4.20: Matriz de confusión promedio para las 4 etapas del ciclo estral utilizando la red MLP

Para el caso de 2 clases el modelo de la red que dio mejores resultados se muestra en la figura 4.7. Con una exactitud de 100 % en entrenamiento y en prueba. La tabla 4.21 muestra el mejor resultado de los conjuntos de validación y la tabla 4.22 muestra el promedio.

T/R	C2	C1
C2	<b>43</b>	0
C1	0	<b>40</b>

Tabla 4.21: Matriz de confusión con el mejor resultado para el caso de 2 clases utilizando la red MLP

T/R	C2	C1
C2	<b>39</b>	3.8
C1	3.2	<b>36.4</b>

Tabla 4.22: Matriz de confusión promedio para el caso de 2 clases utilizando la red MLP

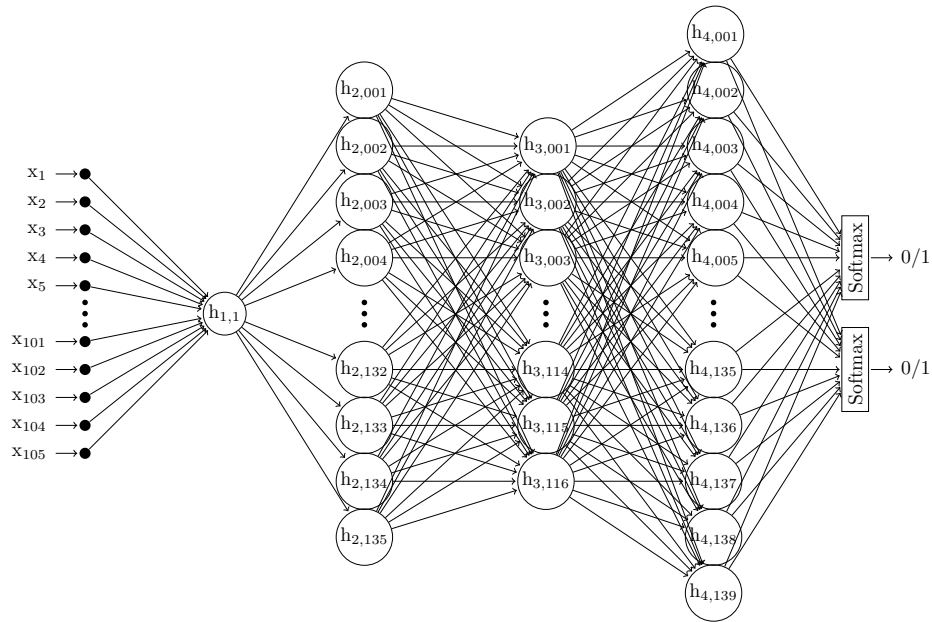


Figura 4.7: Perceptrón multicapa: 105 características, 1 neurona de entrada, 135 neuronas en la primera capa oculta, 116 neuronas en la segunda capa oculta, 139 neuronas en la tercera capa oculta y 2 clasificadores softmax. Las capas ocultas y la capa de entrada usan la función de activación relu.

### 4.3. Comparación

El porcentaje de exactitud alcanzado por la versión 3 para la SVM lineal fue mayor al 80 % esto motivo a mantener los rasgos y variar los clasificadores. En primera instancia se implementaron SVM con diferentes núcleos. Con excepción de la SVM de base radial, los resultados seguían siendo mayores al 80 % en exactitud. Lo anterior motivo a probar la red MLP con la cual se alcanzó el mejor porcentaje alcanzado un 87 % para el caso de las cuatro etapas del ciclo estral y un 100 % para el caso de las 2 clases. Para poder comparar los resultados se optó por utilizar las métrica micro, macro  $F_1$  [37], [38], la exactitud y la exactitud promedio de los conjuntos de validación para el caso de las 4 etapas. Para el caso de las 2 clases se presenta la métrica micro  $F_1$  esto debido a que al ser 2 clases los resultados de las métricas micro y macro son iguales. Se opto por usar las métricas  $F_1$  pues estas representan el comportamiento de manera fiel al considerar la repetitividad que tiene el clasificador y la precisión. La tabla 4.23 muestra los resultados de los 5 clasificadores utilizados con las métricas de textura y forma utilizadas para la clasificación de las cuatro etapas. Como se observa el clasificador que presento el mejor desempeño fue la red MLP. las SVM lineales (núcleo lineal y la propuesta por Cramer y Singer) y la polinomial presentaron porcentajes

aceptables. Todos los clasificadores entraron en una condición de sobre entrenamiento<sup>1</sup> siendo mas marcado en el caso de la SVM de base radial, esto se observa en los valores de exactitud de su conjunto de entrenamiento y prueba. Los valores de exactitud promedio nos indican la fiabilidad de los parámetros de los clasificadores pues es la exactitud que el mismo modelo alcanza con diferentes conjuntos de entrenamiento y prueba. Si bien la fiabilidad de los parámetros no es alta esta puede mejora al incrementar el dataset lo cual disminuiría la condición de sobre entrenamiento. La tabla 4.24 muestra la métrica  $F_1$  para el caso de las 2 clases. En ella se observa como todos los clasificadores alcanzaron valores superiores al 96 % lo cual se debe a que al reducir el número de clases se aumentaron el número de ejemplos de entrenamiento y prueba que se utilizaron por cada clase para entrenar al clasificar. Además de aumentar el número de ejemplos cabe mencionar que las etapas del ciclo pertenecientes a cada clase comparten similitudes.

	micro $F_1$	macro $F_1$	exactitud	exactitud promedio
SVM lineal	82.92 %	83.20 %	82.92 %	74.27 %
SVM propuesta por Cramer y Singer	80.48 %	80.68 %	80.49 %	75.97 %
SVM de base radial	76.82 %	77.28 %	76.83 %	69.90 %
SVM polinomial	80.48 %	80.82 %	80.49 %	75.48 %
MLP	87.80 %	88.08 %	87.80 %	78.88 %

Tabla 4.23: Métricas de validación y desempeño para las cuatro etapas del ciclo estral

	micro $F_1$
SVM lineal	97.59 %
SVM propuesta por Cramer y Singer	97.59 %
SVM de base radial	96.34 %
SVM polinomial	97.59 %
MLP	100 %

Tabla 4.24: Micro  $F_1$  para el caso de 2 clases

## 4.4. Aplicación

La aplicación puede evaluar imágenes independientes o carpetas que contengan imágenes (Fig. 4.8). Para evaluar una imagen independiente, se presiona el menú archivo, se selecciona

<sup>1</sup>El sobre entrenamiento o sobre ajuste ocurre cuando el conjunto de entrenamiento es pequeño, lo cual provoca que el clasificador se ajuste a cada ejemplo de entrenamiento. Se puede decir que existe sobre entrenamiento cuando la exactitud de entrenamiento es mucho mayor a la exactitud de prueba.

el menú abrir, dentro del menú abrir se selecciona la opción Abrir Imagen y se pulsa el botón procesar (Fig. 4.10). Si se desea evaluar un conjunto de imágenes se selecciona la opción Abrir Carpeta y se pulsa el botón procesar (Fig. 4.10). Si se desea guardar los resultados de diferentes evaluaciones (Fig. 4.9), se selecciona el menú archivo, se selecciona el menú Guardar, dentro del menú guardar se selecciona la opción Guardar todo. Si se desea guardar únicamente el último análisis se selecciona la opción Guardar Ultimo (Fig. 4.9). Si se desea eliminar los análisis anteriores se pulsa la opción Nuevo dentro del menú Archivo. La aplicación utiliza el modelo de red neuronal presentado en los resultados de la versión 3.

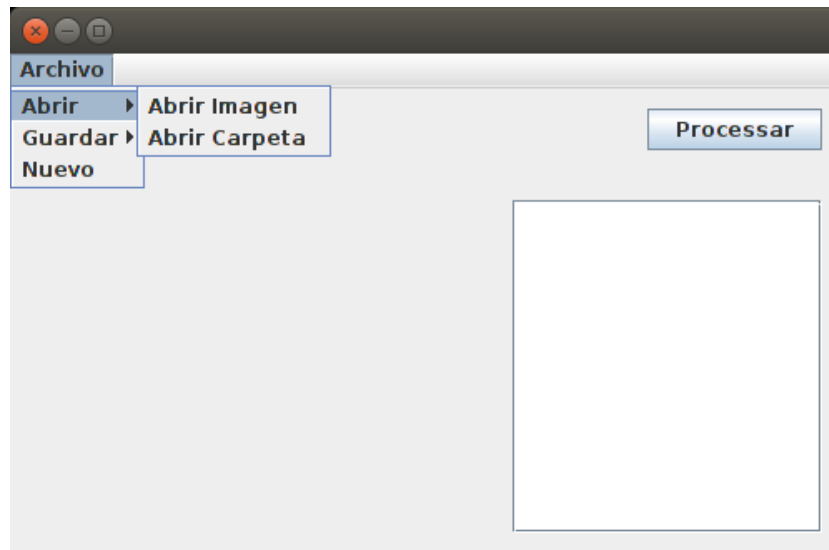


Figura 4.8: Aplicación: Menú Abrir.

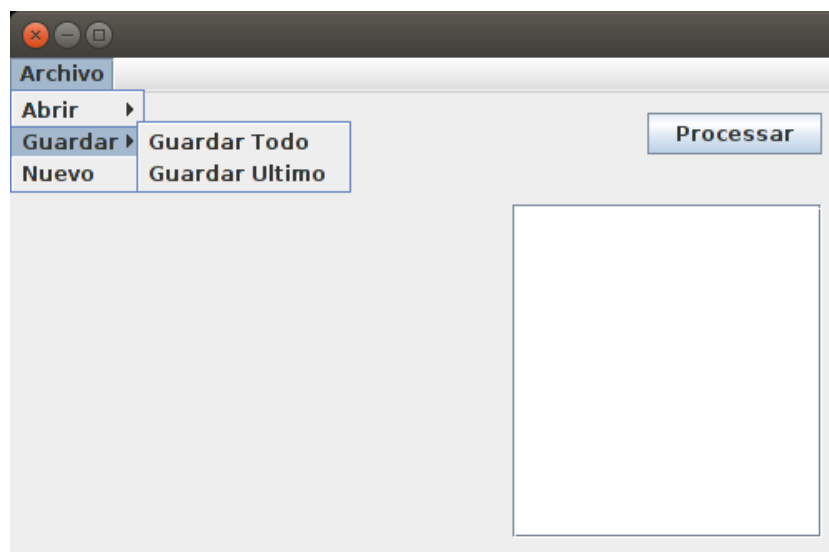


Figura 4.9: Aplicación: Menú Guardar.

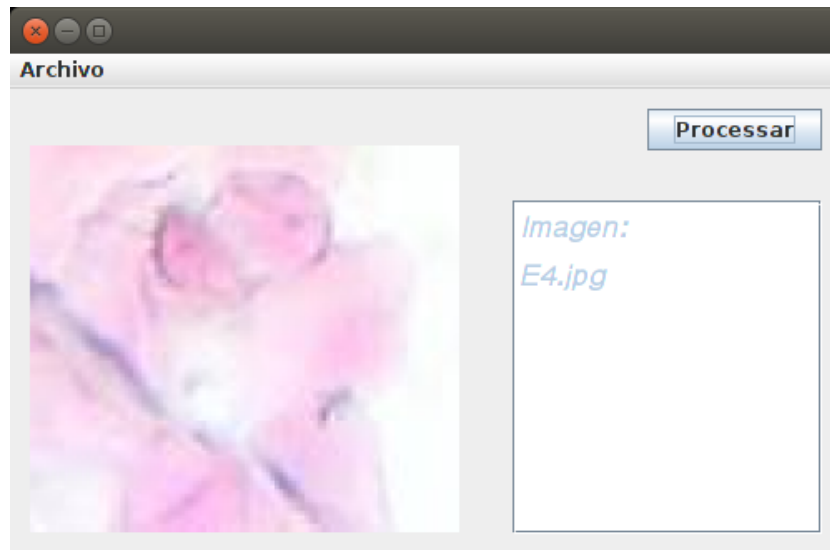


Figura 4.10: Aplicación: Resultado de procesar una imagen.

## 4.5. Conclusiones y trabajo futuro.

La primera versión no produjo resultados adecuados debido a que los histogramas no compartían ni distribuciones ni valores semejantes entre una misma clase. Esto ocurre porque no se controlan la cantidad de células presentes en cada imagen, lo cual provoca que los valores de cada elemento del histograma sean diferentes para diferentes imágenes de una misma clase. La segunda versión presenta un gran problema en el momento de la sintonización manual de la red PCNN pues no existe una metodología manual para realizarla. Esta versión puede ser mejorada implementando métodos de optimización como la evolución diferencial o algoritmos genéticos. El problema se reduciría entonces a determinar una función de ajuste que permita incrementar la distancia euclidiana entre los vectores de las series de tiempo de clases diferentes y reducirla para los de la misma clase. La versión 3 puede ser mejorada de 3 formas. La primera sería expandiendo el conjunto de entrenamiento lo cual ayudaría a reducir el sobre entrenamiento presente en sus clasificadores. La segunda forma es modificar el algoritmo GLCM de tal manera que este omita la condición (0,0) pues esta no proporciona información de la textura de las células lo cual introduce ruido al clasificador. La tercera forma sería implementar un algoritmo de evolución diferencial utilizando los parámetros que mostraron porcentajes de clasificación altos como la población inicial. Sin embargo, esto podría solo incrementar la mejor exactitud sin mejorar su fiabilidad. Note que la fiabilidad está asociada en este caso a la condición de sobre entrenamiento y esta propuesta no modifica ni el dataset ni los rasgos por lo cual seguiría estando presente el sobre entrenamiento.

Con los resultados alcanzados en este trabajo se considera que el problema de la clasificación del ciclo estral para el control de la reproducción está resuelto pues los porcentajes de clasificación son cercanos al 100 %. Sin embargo, el problema de la clasificación de las cuatro etapas del ciclo estral aún debe ser mejorado, lo cual puede hacerse con lo descrito anteriormente o abordando nuevas perspectivas como desarrollar un sistema experto que sea capaz de detectar cada tipo de célula presente en una imagen o por medio de las redes neuronales convolucionales las cuales han mostrado tener un desempeño sobresaliente en problemas de este tipo. En este trabajo no se atacó el problema desde esos puntos de vista pues cada uno presenta sus propios inconvenientes. Para el caso del sistema experto el principal problema a resolver es la segmentación de células entrelazadas el cual aún es un problema abierto. Por su parte las redes neuronales convolucionales requieren de una dataset grande lo cual implica que demandan una mayor potencia de computo.

---

## Bibliografía

- [1] H. Tan, H. Jiang, A. Dong, B. Yang, and L. Zhang, “C-v level set based cell image segmentation using color filter and morphology,” in *2014 International Conference on Information Science, Electronics and Electrical Engineering*, vol. 2, April 2014, pp. 1073–1077.
- [2] Z. Lu, G. Carneiro, and A. P. Bradley, “An improved joint optimization of multiple level set functions for the segmentation of overlapping cervical cells,” *IEEE Transactions on Image Processing*, no. 4, pp. 1261–1272, April 2015.
- [3] A. Tareef, Y. Song, W. Cai, D. D. Feng, and M. Chen, “Automated three-stage nucleus and cytoplasm segmentation of overlapping cells,” in *2014 13th International Conference on Control Automation Robotics Vision (ICARCV)*, Dec 2014, pp. 865–870.
- [4] M. E. Plissiti, C. Nikou, and A. Charchanti, “Watershed-based segmentation of cell nuclei boundaries in pap smear images,” in *Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine*, Nov 2010, pp. 1–4.
- [5] M. E. Plissiti, M. Vrigkas, and C. Nikou, “Segmentation of cell clusters in pap smear images using intensity variation between superpixels,” in *2015 International Conference on Systems, Signals and Image Processing (IWSSIP)*, Sept 2015, pp. 184–187.
- [6] L. Zhang, H. Kong, C. T. Chin, T. Wang, and S. Chen, “Cytoplasm segmentation on cervical cell images using graph cut-based approach,” *Biomed Mater Eng*, vol. 24, no. 1, pp. 1125–1131, 2014.
- [7] M. Zhao, A. Wu, J. Song, X. Sun, and N. Dong, “Automatic screening of cervical cells using block image processing,” *Biomed Eng Online*, vol. 15, p. 14, Feb 2016. [Online]. Available: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4743397>

- [8] O. Ronneberger, P. Fischer, and T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Cham: Springer International Publishing, 2015, pp. 234–241. [Online]. Available: [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)
- [9] HunterLab, “Cie l\*a\*b\* color scale,” *Application Notes*, vol. 8, no. 7, pp. 1–4, Jul. 1996. [Online]. Available: <http://cobra.rdsor.ro/cursuri/cielab.pdf>
- [10] Daniel. (2017, Oct.) Daniel’s homework for pattern-recognition course. <http://cges560809.weebly.com/homeworkreading.html>. [Online]. Available: <http://cges560809.weebly.com/homeworkreading.html>
- [11] K. Zhan, J. Shi, H. Wang, Y. Xie, and Q. Li, “Computational mechanisms of pulse-coupled neural networks: A comprehensive review,” *Archives of Computational Methods in Engineering*, vol. 24, no. 3, pp. 573–588, Jul 2017. [Online]. Available: <https://doi.org/10.1007/s11831-016-9182-3>
- [12] B. M. Priddy, S. A. Carmack, L. C. Thomas, J. C. Vendruscolo, G. F. Koob, and L. F. Vendruscolo, “Sex, strain, and estrous cycle influences on alcohol drinking in rats,” *Pharmacology Biochemistry and Behavior*, vol. 152, pp. 61 – 67, 2017, sex differences on drugs affecting behavior. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0091305716301356>
- [13] S. Y. Lee, J.-Y. Lee, S. J. Oh, H.-C. Kim, and S. K. Kim, “Expression of hepatic and ovarian antioxidant enzymes during estrous cycle in rats,” *Toxicology Letters*, vol. 212, no. 3, pp. 329 – 336, 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0378427412011708>
- [14] M. G. Quinlan, A. Duncan, C. Loiselle, N. Graffe, and W. G. Brake, “Latent inhibition is affected by phase of estrous cycle in female rats,” *Brain and Cognition*, vol. 74, no. 3, pp. 244 – 248, 2010. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0278262610001119>
- [15] R. Cossio, M. B. Carreira, C. E. Vásquez, and G. B. Britton, “Sex differences and estrous cycle effects on foreground contextual fear conditioning,” *Physiology & Behavior*, vol. 163, pp. 305 – 311, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0031938416302621>
- [16] S. L. D. Shannon L. Byers, Michael V. Wiles and R. A. Taft, “Mouse estrous cycle identification tool and images,” *PLoS ONE*, vol. 7, 2012.

- [17] C. S. Caligioni, *Assessing Reproductive Status/Stages in Mice*. John Wiley & Sons, Inc., 2001. [Online]. Available: <http://dx.doi.org/10.1002/0471142301.nsa04is48>
- [18] D. Kashyap, A. Somani, J. Shekhar, A. Bhan, M. K. Dutta, R. Burget, and K. Riha, “Cervical cancer detection and classification using independent level sets and multi svms,” in *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, June 2016, pp. 523–528.
- [19] M. E. Plissiti, C. Nikou, and A. Charchanti, “Automated detection of cell nuclei in pap smear images using morphological reconstruction and clustering,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 2, pp. 233–241, March 2011.
- [20] (2017, Oct.) Cell tracking challenge. <http://biomedicalimaging.org/2015/hardi-reconstruction-challenge/>. <http://biomedicalimaging.org/2015/hardi-reconstruction-challenge/>. [Online]. Available: <http://biomedicalimaging.org/2015/hardi-reconstruction-challenge/>
- [21] R. P. R. R. de la Cruz, “Control endocrino del ciclo reproductor en la rata: Una experiencia docente utilizando un método de caracterización citológico,” *Revista de Enseñanza Universitaria*, pp. 545–554, 1998.
- [22] R. Roberto and S. Juan, *Procesamiento Y Análisis Digital De Imágenes (Spanish Edition)*. Alfaomega Grupo Editor, 2012. [Online]. Available: <https://www.amazon.com/Procesamiento-An%C3%A1lisis-Digital-Im%C3%A1genes-Spanish/dp/6077072230?SubscriptionId=0JYN1NVW651KCA56C102&tag=techkie-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=6077072230>
- [23] I. Pitas, *Digital Image Processing Algorithms and Applications*. JOHN WILEY & SONS INC, 2000. [Online]. Available: [http://www.ebook.de/de/product/3602312/ioannis\\_pitas\\_digital\\_image\\_processing\\_algorithms\\_and\\_applications.html](http://www.ebook.de/de/product/3602312/ioannis_pitas_digital_image_processing_algorithms_and_applications.html)
- [24] O. team. (2017, Sep.) Opencv. <https://opencv.org/>. <https://opencv.org/>. [Online]. Available: <https://opencv.org/>
- [25] Y. Ma, K. Zhan, and Z. Wang, *Applications of Pulse-Coupled Neural Networks*. Springer-Verlag GmbH, 2011. [Online]. Available: [http://www.ebook.de/de/product/11508764/yide\\_ma\\_kun\\_zhan\\_zhaobin\\_wang\\_applications\\_of\\_pulse\\_coupled\\_neural\\_networks.html](http://www.ebook.de/de/product/11508764/yide_ma_kun_zhan_zhaobin_wang_applications_of_pulse_coupled_neural_networks.html)
- [26] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan 1979.

- [27] D. Arthur and S. Vassilvitskii, “K-means++: The advantages of careful seeding,” in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '07. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1283383.1283494>
- [28] 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. [Online]. Available: <http://www.jstor.org/stable/2346830>
- [29] C.-H. Lin, C.-C. Chen, H.-L. Lee, and J.-R. Liao, “Fast k-means algorithm based on a level histogram for image retrieval,” *Expert Systems with Applications*, vol. 41, no. 7, pp. 3276 – 3283, 2014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0957417413009299>
- [30] J. H. S. Azuela, *Visión artificial : rasgos descriptores para el reconocimiento de objetos*. RA-MA S.A. Editorial y Publicaciones, 2014. [Online]. Available: [http://www.ebook.de/de/product/20302851/juan\\_humberto\\_sossa\\_azuela\\_vision\\_artificial\\_rasgos\\_descriptores\\_para\\_el\\_reconocimiento\\_de\\_objetos.html](http://www.ebook.de/de/product/20302851/juan_humberto_sossa_azuela_vision_artificial_rasgos_descriptores_para_el_reconocimiento_de_objetos.html)
- [31] R. M. Haralick, K. Shanmugam, and I. Dinstein, “Textural features for image classification,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, Nov 1973.
- [32] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Guillard, T. Yu, and the scikit-image contributors, “scikit-image: image processing in Python,” *PeerJ*, vol. 2, p. e453, 6 2014. [Online]. Available: <http://dx.doi.org/10.7717/peerj.453>
- [33] D. A. Clausi, “An analysis of co-occurrence texture statistics as a function of grey level quantization,” *Canadian Journal of Remote Sensing*, vol. 28, no. 1, pp. 45–62, 2002. [Online]. Available: <https://doi.org/10.5589/m02-004>
- [34] L. I. Smith. (2017, Oct.) A tutorial on principal components analysis. <http://xurl.es/tunxw>. <http://xurl.es/tunxwf>. [Online]. Available: <http://xurl.es/tunxw>
- [35] T. M. Mitchell, *Machine Learning*. McGraw-Hill Education, 1997.
- [36] J. K. Thomas Lindblad, *Image Processing Using Pulse-Coupled Neural Networks*. Springer Berlin Heidelberg, 2005. [Online]. Available: <http://www.springer.com/9783540237708>

- lable: [http://www.ebook.de/de/product/8897426/thomas\\_lindblad\\_jason\\_kinser\\_image\\_processing\\_using\\_pulse\\_coupled\\_neural\\_networks.html](http://www.ebook.de/de/product/8897426/thomas_lindblad_jason_kinser_image_processing_using_pulse_coupled_neural_networks.html)
- [37] V. Van Asch, “Macro-and micro-averaged evaluation measures [[basic draft]],” 2013. [Online]. Available: <https://pdfs.semanticscholar.org/1d10/6a2730801b6210a67f7622e4d192bb309303.pdf>
- [38] F. Sebastiani, “Machine learning in automated text categorization,” *ACM Comput. Surv.*, vol. 34, no. 1, pp. 1–47, Mar. 2002. [Online]. Available: <http://doi.acm.org/10.1145/505282.505283>
- [39] Stanford. (2017, Oct.) Linear classification. <http://cs231n.github.io/linear-classify/>. <http://cs231n.github.io/linear-classify/>. [Online]. Available: <http://cs231n.github.io/linear-classify/>
- [40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [41] F. Chollet *et al.*, “Keras,” <https://github.com/fchollet/keras>, 2015.
- [42] T. S. P. Contributors. (2017, Sep.) Spyder 3.2.3. <https://spyder-ide.github.io/>. <https://spyder-ide.github.io/>.

---

# Apéndice A

## Codigos

### A.1. Otsu

Algoritmo 1: Otsu

```
function Umbral=OtssuL(Imagen)
[r,c]=size(Imagen);
Histograma=zeros(1,256);
for i=1:c
    for j=1:r
        Intencidad=double(Imagen(j,i))+1;
        Histograma(Intencidad)=Histograma(Intencidad)+1;
    end
end
N=r*c;
Pi=Histograma/N;
W0=zeros(1,255);
W1=zeros(1,255);
Ii=1:1:256;
MT=Pi.*Ii;
for k=2:255
    W0(k)=sum(Pi(1,1:k));
    W1(k)=1-W0(k);
    M0(k)=sum(MT(1,1:k))/W0(k);
    M1(k)=sum(MT(1,k+1:256))/W1(k);
    Vb2(k)=W0(k)*W1(k)*(M1(k)-M0(k))^2;
end
Vb2(isnan(Vb2))=0;
[~,I]=sort(Vb2);
i=1;
Umbral=I(255);
WK=sum(Pi(1,1:Umbral));
while WK==0
    Umbral=I(255-i);
    WK=sum(Pi(1,1:Umbral));
    i=i+1;
end
```

```
end  
Umbral=Umbral -1;
```

## A.2. Media Arigmetica

### Algoritmo 2: Media Arigmetica

```
function Umbral = ArithmeticMean(Imagen)
[r,c]=size(Imagen);
Histograma=zeros(1,256);
for i=1:c
    for j=1:r
        Intencidad=double(Imagen(j,i))+1;
        Histograma(Intencidad)=Histograma(Intencidad)+1;
    end
end
L=1:1:256;
P=Histograma/(r*c);
Umbral=sum(L.*P);
```

### A.3. Entropía

Algoritmo 3: Entropía

```
function Umbral=EntropyT(Imagen)
[r,c]=size(Imagen);
N=r*c;
ValorMinimo=min(min(Imagen));
ValorMaximo=max(max(Imagen));
Umbral=2:1:ValorMaximo-1;
indice=1;
for i=Umbral
    BTem=Imagen>i;
    NumDe1=sum(sum(BTem));
    NumDe0=N-NumDe1;
    P1=NumDe1/N;
    P0=NumDe0/N;
    Entropia(indice)=-P1*log(P1)-P0*log(P0);
    Umbrales(indice)=i;
    indice=indice+1;
end
[~,Indices]=sort(Entropia,'descend');
Umbral=Umbrales(Indices(1))-1;
```

## A.4. Entropía Cruzada

### Algoritmo 4: Entropía Cruzada

```

function [Umbral,CoEntropia]=CoEntropy(Imagen)
[r,c]=size(Imagen);
Histograma=zeros(1,256);
for i=1:c
    for j=1:r
        Intencidad=double(Imagen(j,i))+1;
        Histograma(Intencidad)=Histograma(Intencidad)+1;
    end
end
Histograma=Histograma+1;
f=1:256;
P=f.*Histograma;

for f=1:256
    M1(f)=(1/sum(Histograma(1:f)))*sum(P(1:f));
    M2(f)=(1/sum(Histograma(f+1:256)))*sum(P(f+1:256));
end

for t=1:256
    A(t)=0;
    for f=1:t
        A(t)=f*Histograma(f)*log(f/M1(t))+M1(t)*Histograma(f)*log(M1(t)/f)+A(t);
    end
    B(t)=0;
    for f=t+1:256
        B(t)=f*Histograma(f)*log(f/M2(t))+M2(t)*Histograma(f)*log(M2(t)/f)+B(t);
    end
end

CoEntropia=A+B;
[~,Umbrales]=sort(CoEntropia);
Umbral=Umbrales(1)-1;

```

## A.5. K-Means

### Algoritmo 5: K-Means

```

function [centroides, ClustersB, NI] = KmeansL(Imagen, k, IteracionesMaximas)
Tolerancia=1;
NI=1;
Max=double(max(max(Imagen)));
Min=double(min(min(Imagen)));
centroides=linspace(Min,Max,k);
T=sum(centroides);
while NI<IteracionesMaximas && T>Tolerancia
    NI=NI+1;
    CentroidesA=centroides;
    [centroides, ClustersB]=update(centroides, Imagen);
    T=abs(sum(double(CentroidesA)-double(centroides)));
end

function [Centroides, ClustersB]=update(centroides, Imagen)
[r, c]=size(Imagen);
[~, NC]=size(centroides);
ClustersB=zeros(r, c);
for i=1:c
    for j=1:r
        for N=1:NC
            DistanciaEuclidiana(N)=sqrt(((double(Imagen(j, i))-double(centroides(N)))
                ^2));
            if N>1
                if DistanciaEuclidiana(N-1)>DistanciaEuclidiana(N)
                    ClustersB(j, i)=N;
                end
            else
                ClustersB(j, i)=N;
            end
        end
    end
end
end
for N=1:NC
    Mascara=uint8(ClustersB==N);
    TotalDeElemntos=sum(sum(Mascara));
    Mascara1=sum(sum(Mascara.*Imagen));
    Centroides(N)=Mascara1/TotalDeElemntos;
end
end

```

## A.6. K-MeansRGB

Algoritmo 6: K-MeansRGB

```

function [centroides, ClustersB, NI] = KmeansR(Imagen, k, IteracionesMaximas)
R=Imagen(:,:,1);
G=Imagen(:,:,2);
B=Imagen(:,:,3);
centroides=CentroidesRGV(R,G,B,k);
Tolerancia=0;
T=sum(sum(centroides));
NI=1;
while NI<IteracionesMaximas && T>Tolerancia
    NI=NI+1;
    CentroidesA=centroides;
    [centroides, ClustersB]=update(centroides, Imagen, R, G, B, k);
    T=abs(sum(sum(double(CentroidesA)-double(centroides))));
end

function [Centroides, ClustersB]=update(centroides, Imagen, R, G, B, k)
[r, c]=size(R);
ClustersB=zeros(r, c);
for i=1:c
    for j=1:r
        for N=1:k
            A=double([Imagen(j, i, 1), Imagen(j, i, 2), Imagen(j, i, 3)]);
            BB=double(centroides(N, :));
            DistanciaEuclidiana(N)=sqrt(sum((A-BB).^2));
            if N>1
                if DistanciaEuclidiana(N-1)>DistanciaEuclidiana(N)
                    ClustersB(j, i)=N;
                end
            else
                ClustersB(j, i)=N;
            end
        end
    end
end

for N=1:k
    Mascara=uint8(ClustersB==N);
    TotalDeElemntos=sum(sum(Mascara));
    Mascara1=sum(sum(Mascara.*R));
    CentroidesR=Mascara1/TotalDeElemntos;
    Mascara1=sum(sum(Mascara.*uint8(G)));
    CentroidesG=Mascara1/TotalDeElemntos;
    Mascara1=sum(sum(Mascara.*uint8(B)));
    CentroidesB=Mascara1/TotalDeElemntos;
    Centroides(N, :)=[CentroidesR, CentroidesG, CentroidesB];
end

```

```
function OC=CentroidesRGV(R,G,B,k)
MaxR=double(max(max(R)));
MaxG=double(max(max(G)));
MaxB=double(max(max(B)));
MinR=double(min(min(R)));
MinG=double(min(min(G)));
MinB=double(min(min(B)));
centroidesR=linspace(MinR,MaxR,k);
centroidesG=linspace(MinG,MaxG,k);
centroidesB=linspace(MinB,MaxB,B);
OC=[centroidesR',centroidesG',centroidesB'];
```

## A.7. K-MeansP

### Algoritmo 7: K-MeansP

```

function [Centroides,Mk]=KMeansP(Imagen,k,Iteraciones)
[H,~]=imhist(Imagen);
ValorMinimo=double(min(min(Imagen)));
ValorMaximo=double(max(max(Imagen)));
Centroides=linspace(ValorMinimo,ValorMaximo,k);
H=H';
iteraciones=1;
error=10;
tolerancia=0.001;
while error>tolerancia & Iteraciones>iteraciones
    Hi=zeros(1,k);
    SH=zeros(1,k);
    for i=1:256
        if H(i)>0
            Indice=Distancia(i,Centroides);
            Hi(Indice)= H(i)*i+Hi(Indice);
            SH(Indice)= H(i)+SH(Indice);
        end
    end
    Centroides0=Centroides;
    Centroides = Hi./SH;
    Iteraciones = iteraciones+1;
    error=sum(abs(Centroides0-Centroides));
end
Indice0=1;
cont=2;
vector(1)=0;
for i=1:256
    Indice=Distancia(i,Centroides);
    if Indice0~=Indice
        vector(cont)=i-1;
        Indice0=Indice;
        cont=cont+1;
    end
end
vector(cont)=255; Centroides=Centroides-1;
for i=1:k
    if i==1
Mk = uint8(Imagen>vector(i) & Imagen<=vector(i+1))*i;
    else
Mk = uint8(Imagen>vector(i) & Imagen<=vector(i+1))*i+Mk;
    end
end
function IndiceCentroide=Distancia(i,Centroides)
D=abs(Centroides-i);
[~,ind]=sort(D);
IndiceCentroide=ind(1);

```

## A.8. PCNN

### Algoritmo 8: PCNN

```

function [Y,F,L,U,Theta]=PCNNParaImagnes(Interacciones,Imagen,DimencionKernel,
    ThetaInicial)
[Jj,Ii]=size(Imagen);
AlphaF=0.7;
AlphaL=0.9;
AlphaT=.2;
Betha=0.000005;
VT=500;
S=double(Imagen);
F=zeros(Jj,Ii);
Y=zeros(Jj,Ii);
L=zeros(Jj,Ii);
U=zeros(Jj,Ii);
Theta=ones(Jj,Ii,1)*ThetaInicial;
for i=1:Interacciones
    progressBar(1/(Interacciones/i))
    K = Contribucion(Y(:,:,i),DimencionKernel);
    F(:,:,i+1)=exp(-AlphaF)*F(:,:,i) + S + K ;
    L(:,:,i+1)=exp(-AlphaL)*L(:,:,i) + K;
    U(:,:,i+1)=F(:,:,i+1)+Betha*F(:,:,i+1).*L(:,:,i+1);
    Y(:,:,i+1)=U(:,:,i+1)>Theta(:,:,i);
    Theta(:,:,i+1)=exp(-AlphaT)*Theta(:,:,i)+VT*Y(:,:,i+1);
    progressBar(1/(Interacciones/i))
end
function YC=Contribucion(Y,DKernel)
[ImagenAlto,ImagenAncho,~]=size(Y);
parfor i=1:ImagenAncho
    for j=1:ImagenAlto
        YC(j,i)= KernelPCNN(DKernel,DKernel,ImagenAncho,ImagenAlto,i,j,Y);
    end
end
function W=KernelPCNN(KernelX,KernelY,ImagenAncho,ImagenAlto,iNeurona,jNeurona,y)
W=0;
X=iNeurona-(KernelX-1)/2;
for i=1:KernelX
    Y=jNeurona-(KernelY-1)/2;
    for j=1:KernelY
        if X>0 && Y>0 && ~(Y==jNeurona && X==iNeurona)
            if X<=ImagenAncho && Y<=ImagenAlto
                W=1/(sqrt((X-iNeurona)^2+(Y-jNeurona)^2))*y(Y,X)+W;
            end
        end
        Y=Y+1;
    end
    X=X+1;
end
end

```



# Apéndice B

## Imágenes segmentadas

### B.1. Media Aritmética

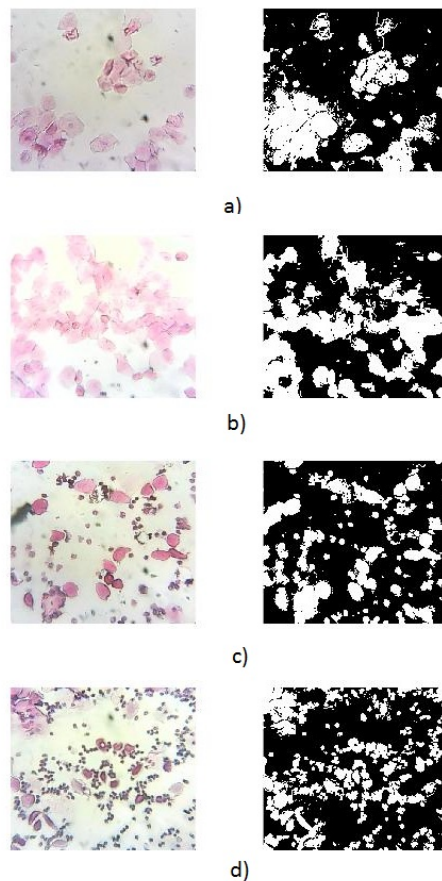


Figura B.1: Imágenes segmentadas por Media Aritmética partiendo de la escala de grises, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro.

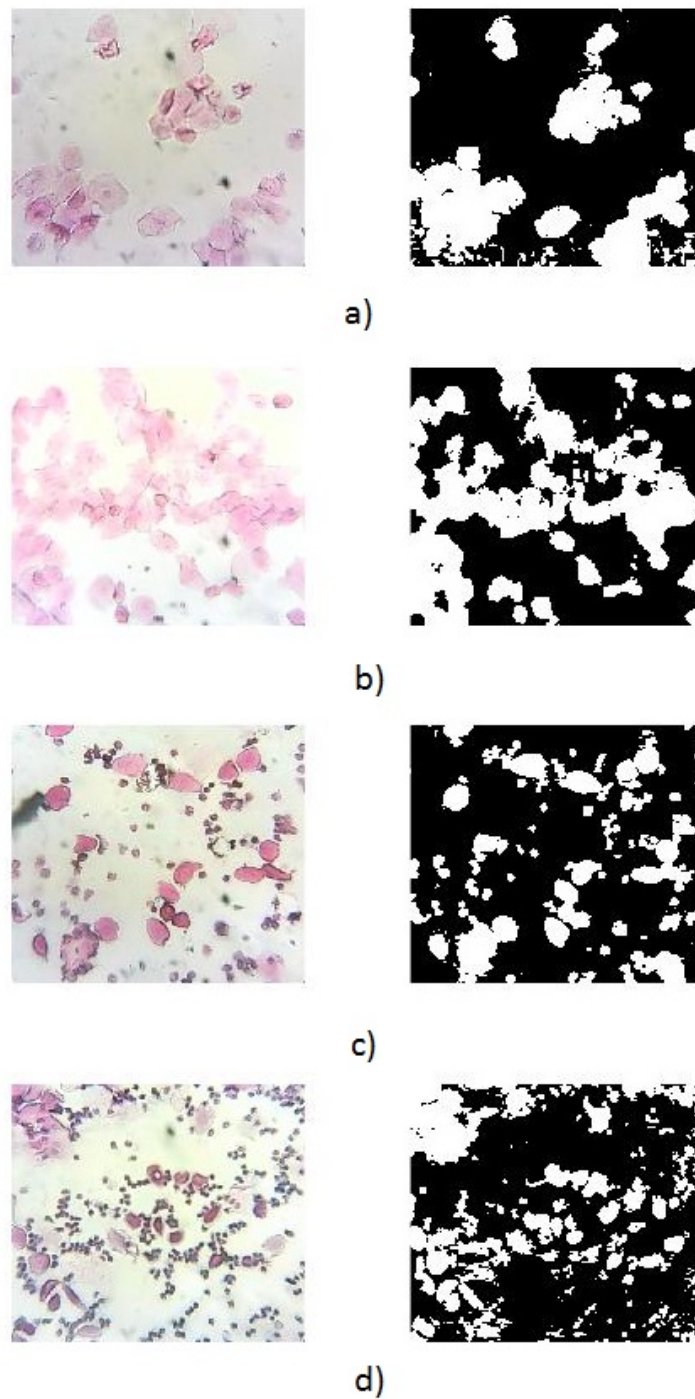


Figura B.2: Imágenes segmentadas por Media Aritmética partiendo del canal A del espacio CIELAB, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro.



## B.2. Máxima Entropía

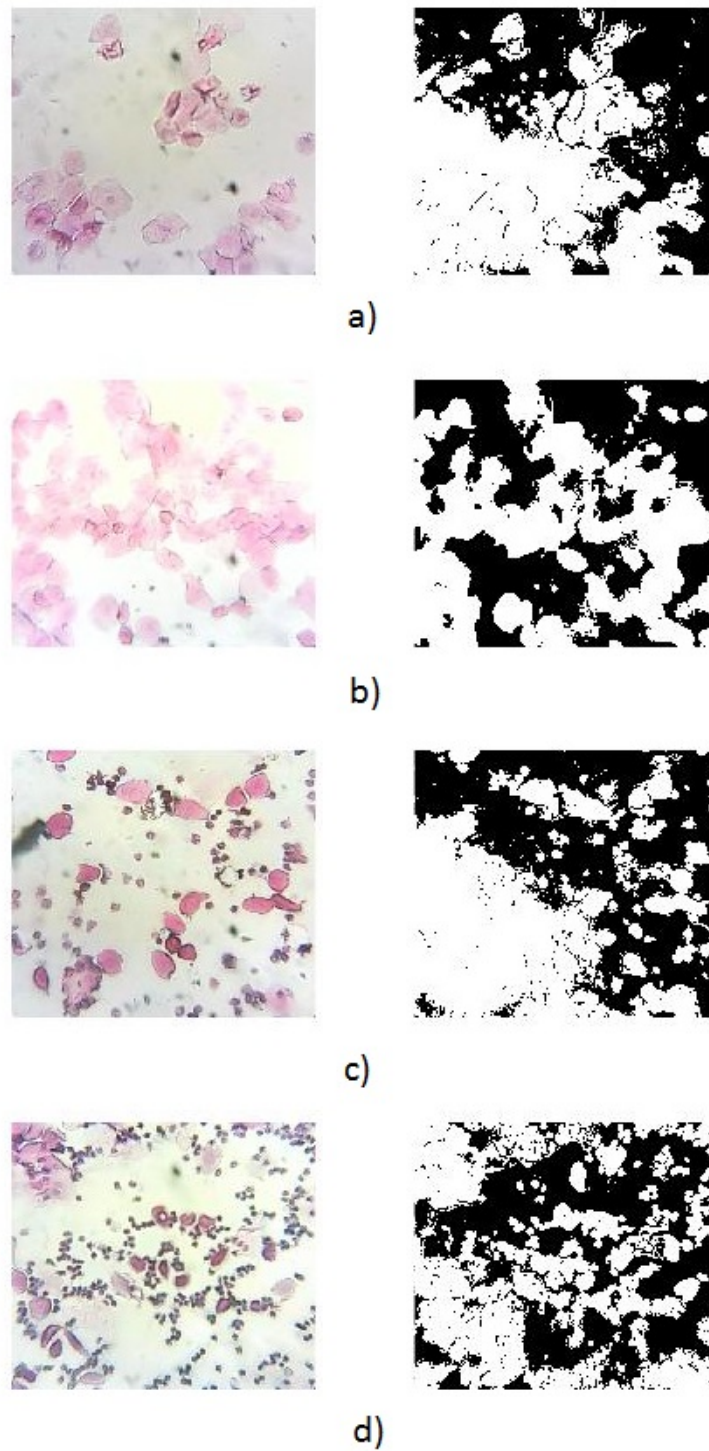


Figura B.3: Imágenes segmentadas por Máxima Entropía partiendo de la escala de grises, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro.

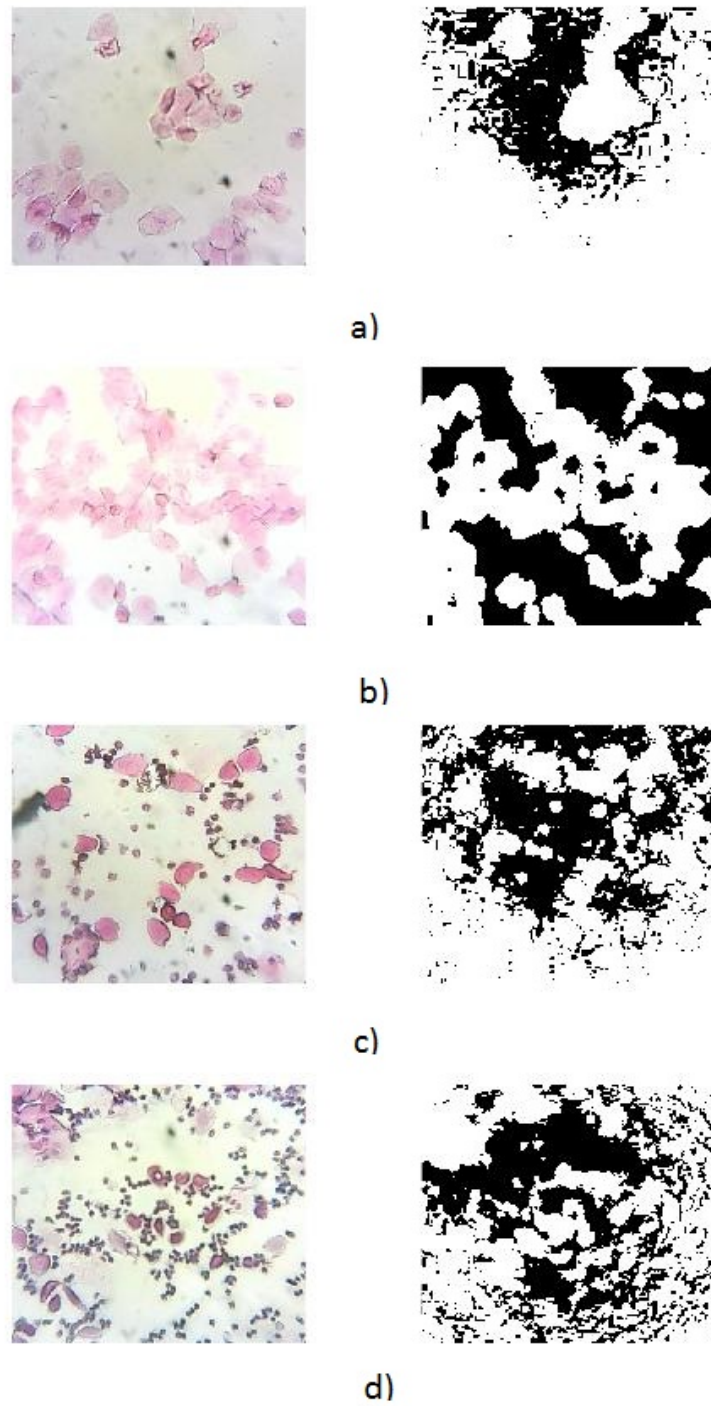


Figura B.4: Imágenes segmentadas por Máxima Entropía partiendo del canal A del espacio CIELAB, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro.



### B.3. Entropía Cruzada

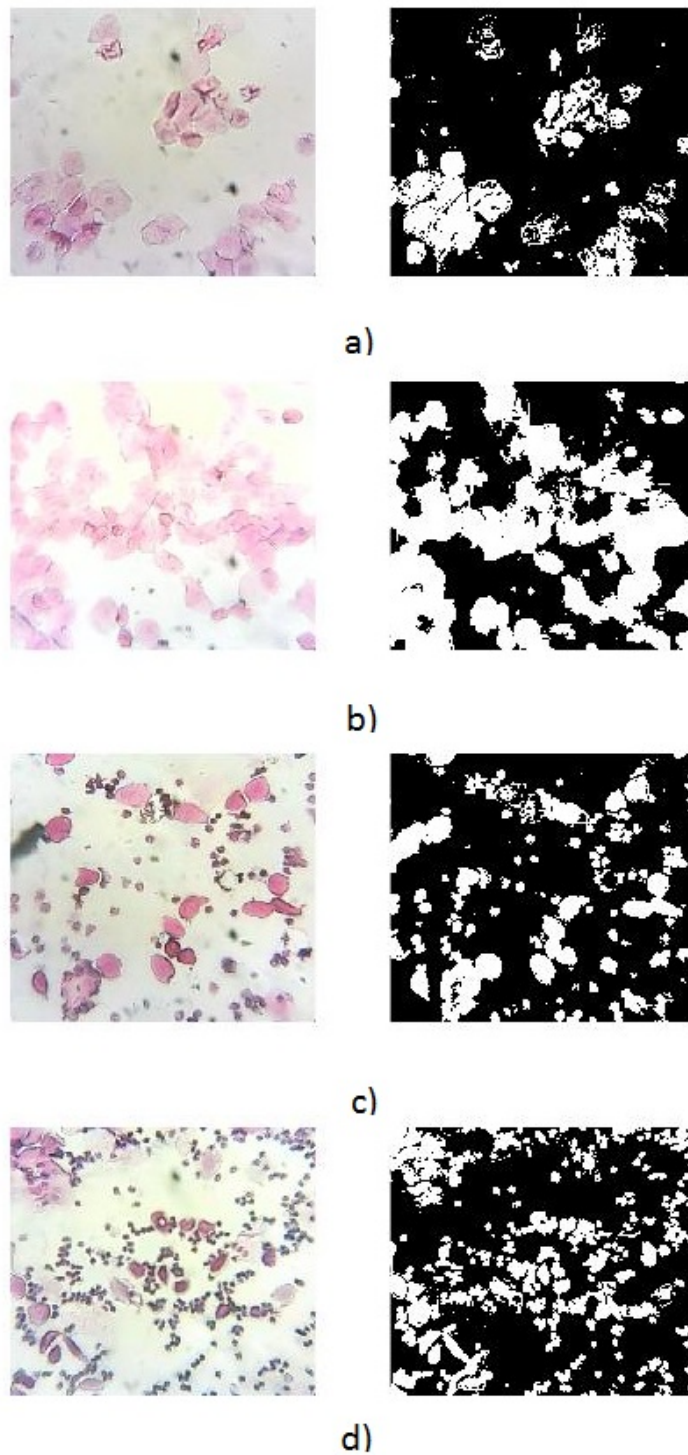


Figura B.5: Imágenes segmentadas por Entropía Cruzada partiendo de la escala de grises, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro.

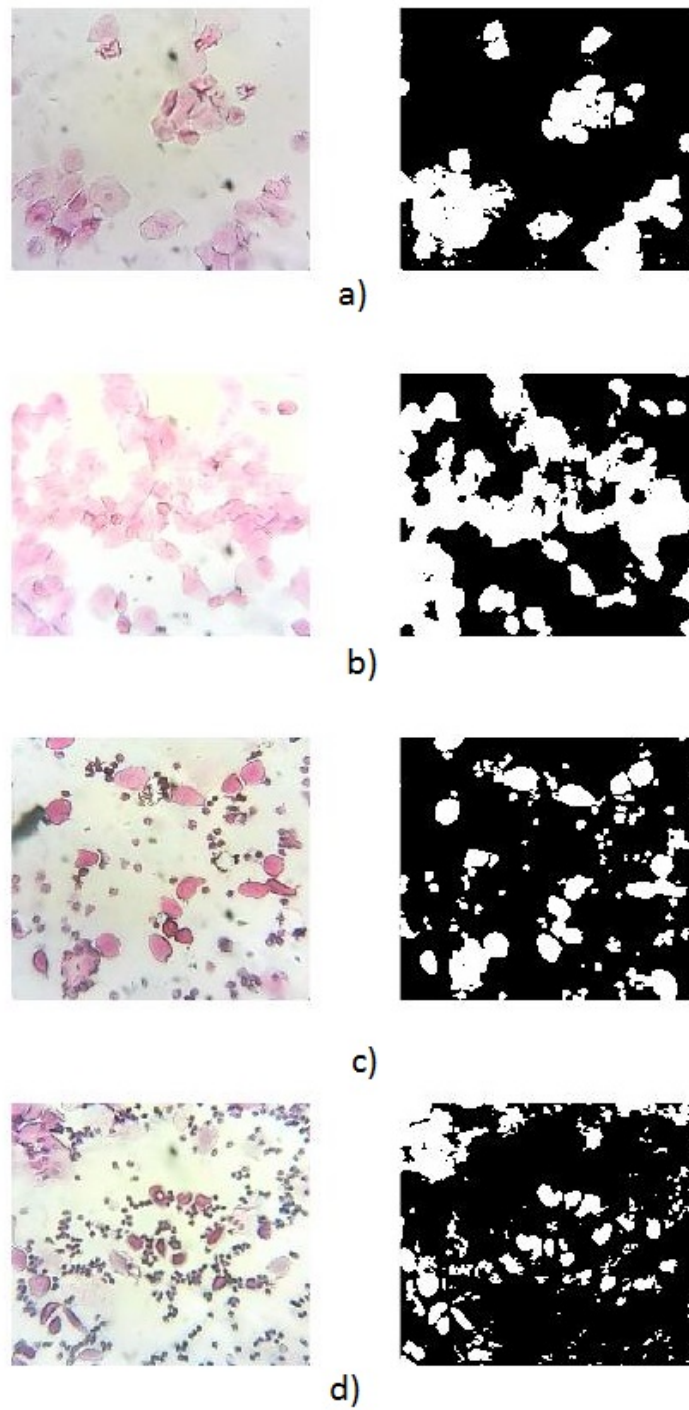


Figura B.6: Imágenes segmentadas por Entropía Cruzada partiendo del canal A del espacio CIELAB, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro.



## B.4. Otsu

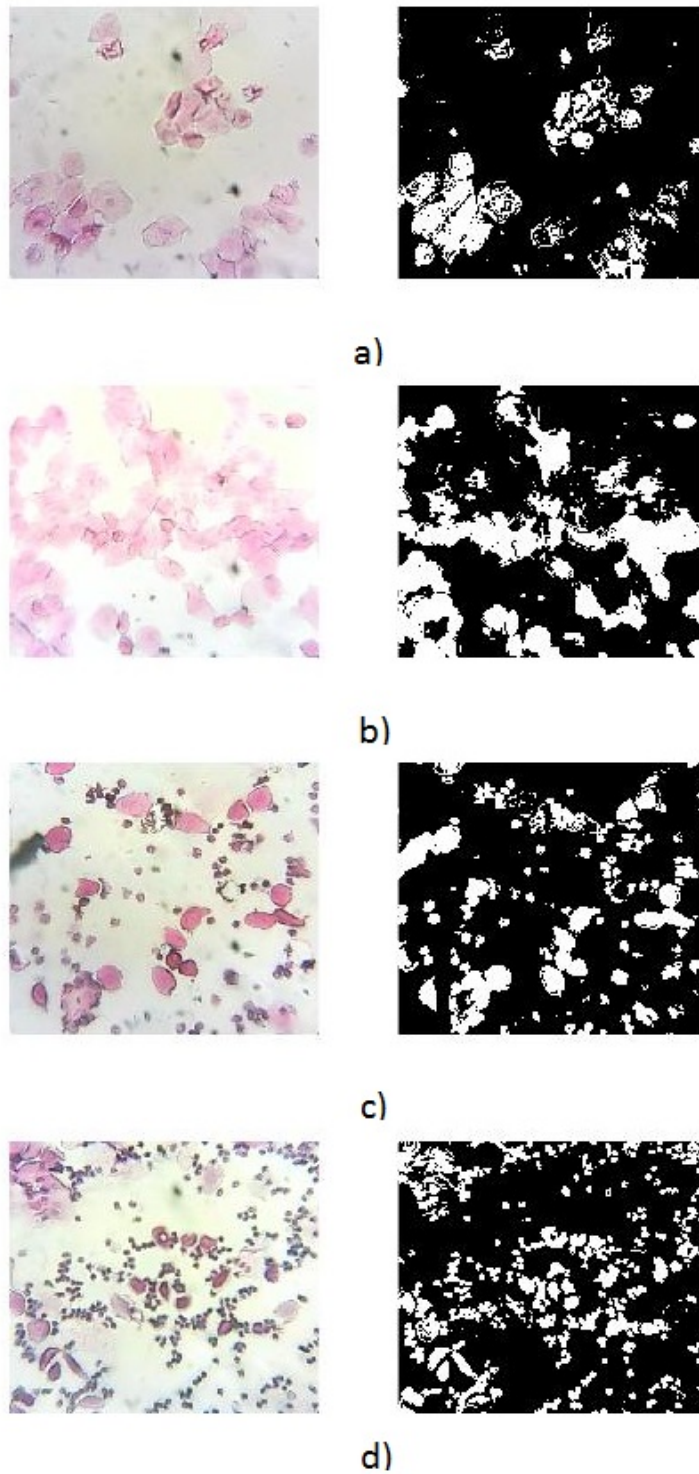


Figura B.7: Imágenes segmentadas por Otsu partiendo de la escala de grises, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro.

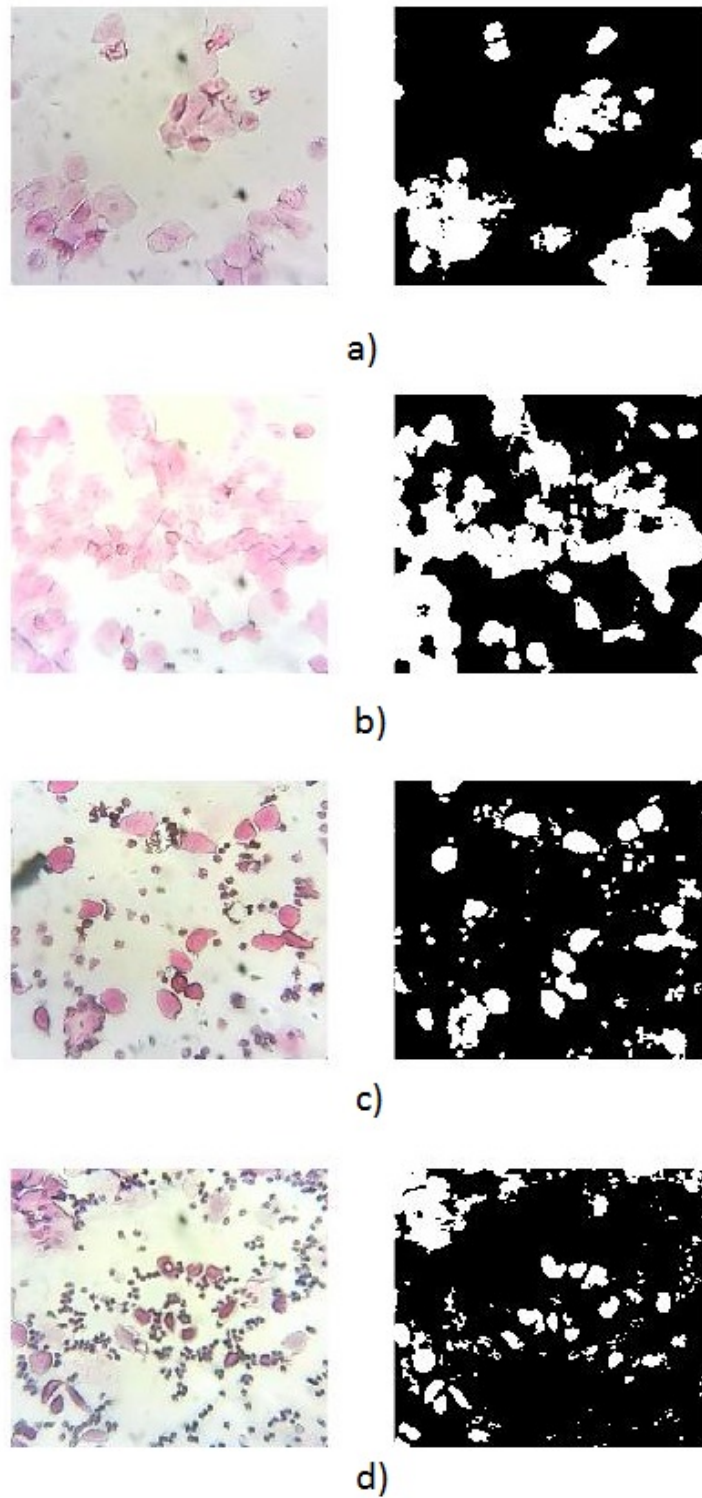


Figura B.8: Imágenes segmentadas por Otsu partiendo del canal A del espacio CIELAB, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro.



## B.5. K-Means

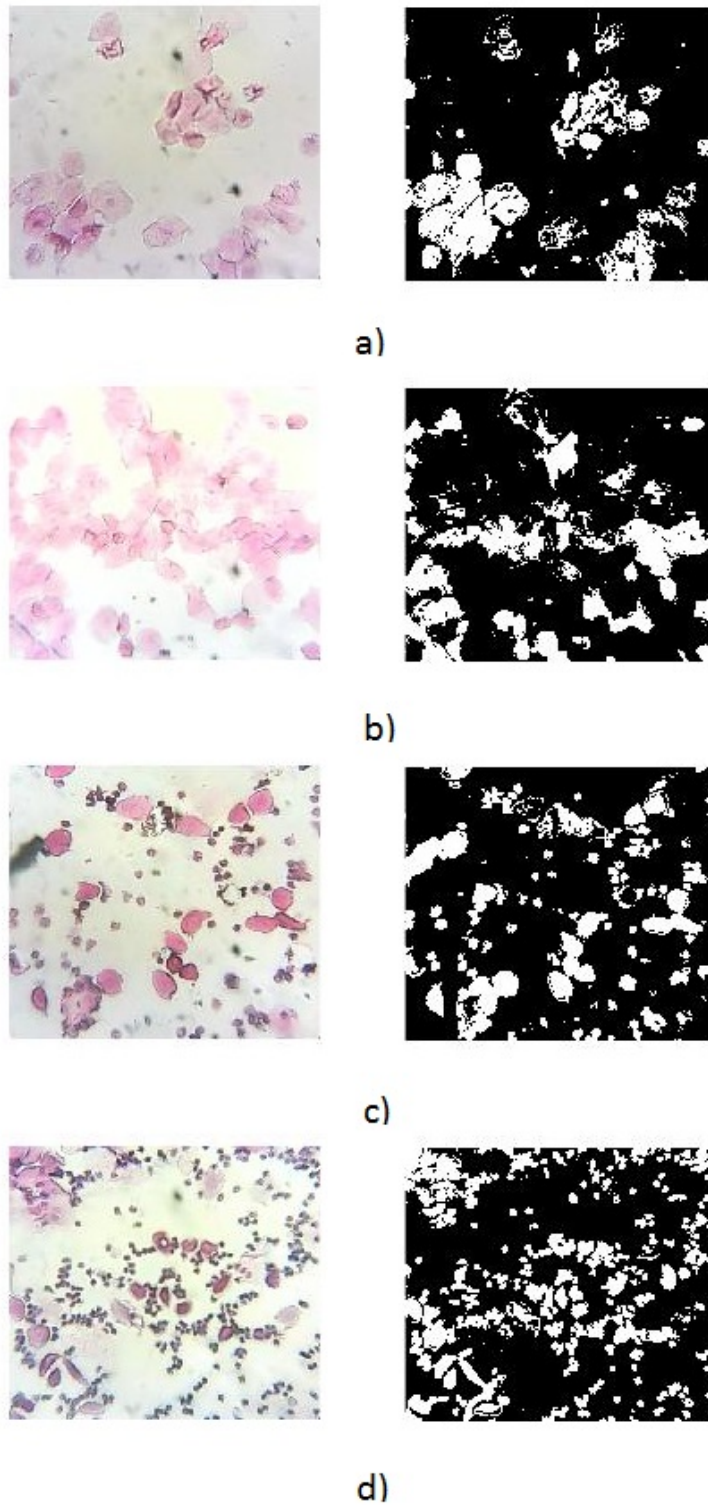


Figura B.9: Imágenes segmentadas por K-Means partiendo de la escala de grises, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro.

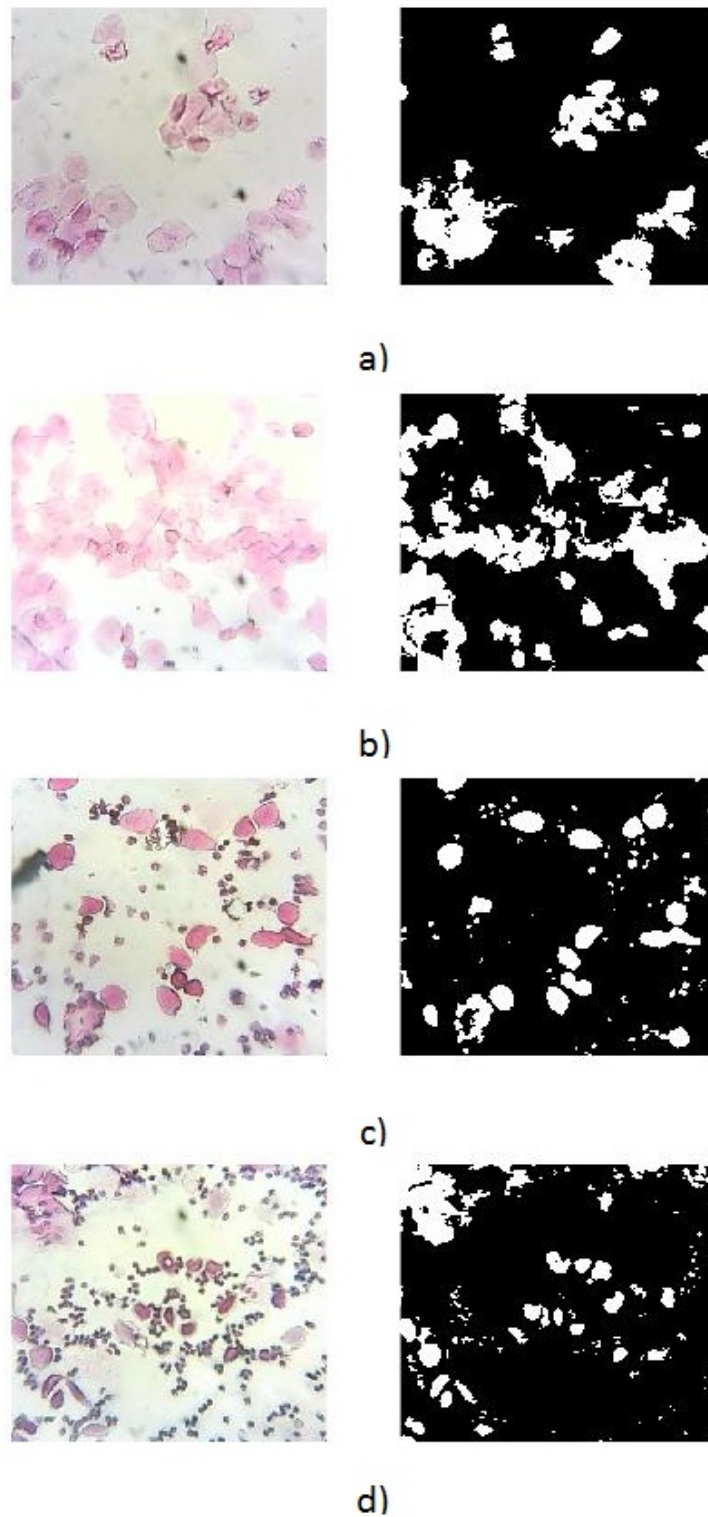


Figura B.10: Imágenes segmentadas por K-Means partiendo del canal A del espacio CIELAB, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro.

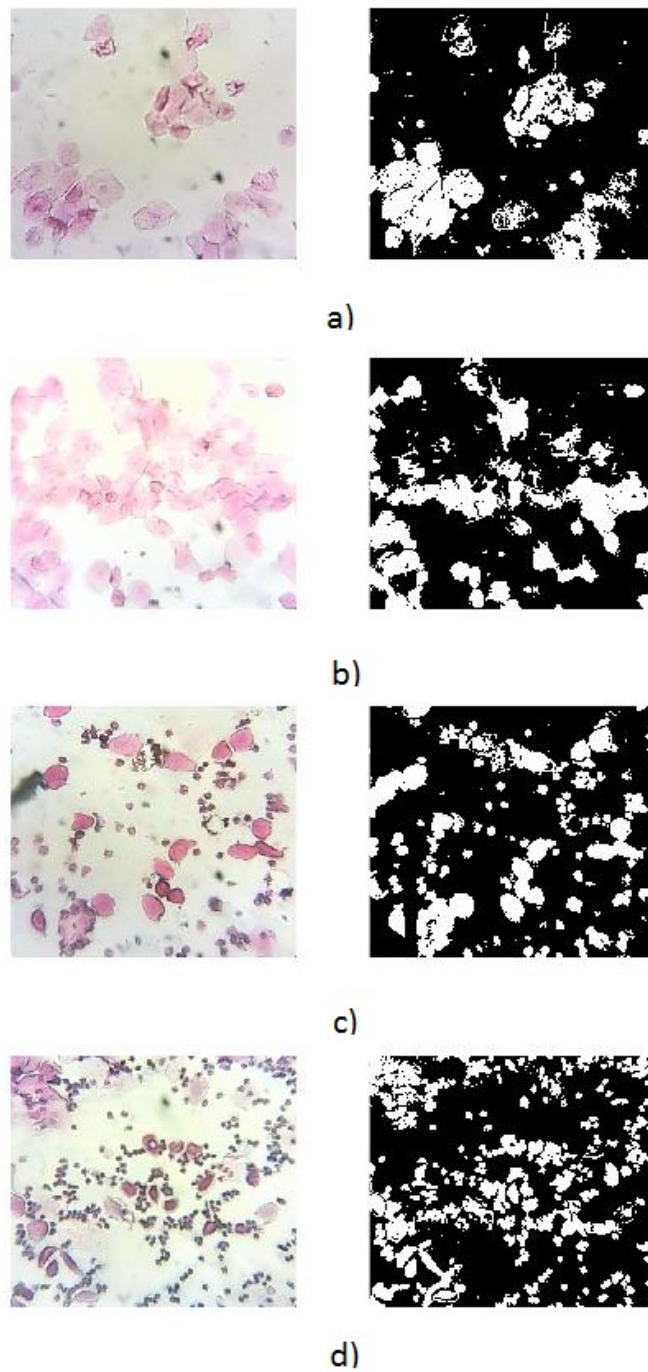


Figura B.11: Imágenes segmentadas por K-Means partiendo del espacio RGB, del lado derecho imagen original y de lado izquierdo imagen segmentada. a) Proestro, b) Estro, c) Metaestro y d) Diestro.

---

# Apéndice C

## Producción académica

Congresos internacionales:

- *Segmentación de células del ciclo estral, Comparativa entre: Otsu, Media Aritmética, Máxima Entropía y Entropía Cruzada, Leonardo Delgado Toral*, Aldrin Barreto Flores, Veronica Edith Bautista Lopez, Salvador E. Ayala Raggi y Francisco Ramos Collazo. Nuevas Perspectivas en Robótica y Computación, 2017, páginas 130-134, ISBN:978-607-97128-5-3.
- *Classification of the Estrous Cycle through Texture and Shape Features, Leonardo Delgado*, Gerardo Hernandez, Erik Zamora, Humberto Sossa, Aldrin Barreto, Francisco Ramos and Rosalina Reyes, IEEE Symposium Series on Computational Intelligence, 2017.

Congreso nacional:

- *Variación del algoritmo K-means para segmentar células del ciclo estral, Leonardo Delgado Toral*, Aldrin Barreto Flores, Veronica Edith Bautista Lopez, Salvador E. Ayala Raggi y Francisco Ramos Collazo. Congreso Nacional de Ciencias de la Computación, 2017.

La Paz, B.C.S., **10/marzo/2017**

**C. Leonardo Delgado Toral  
PRESENTE.**

Nos complace informarle(s) que el comité técnico revisor ha dictaminado que su trabajo titulado: **“Segmentación de células del ciclo estral, Comparativa entre: Otsu, Media Aritmética, Máxima Entropía y Entropía Cruzada.”** sea **ACEPTADO** para su presentación **como ponencia** en el Cuarto Congreso Internacional de Robótica y Computación (CIRC 2017), que se realizará en el Hotel Riu Santa Fe, Cabo San Lucas, Baja California Sur, México, del 2 al 4 de mayo de 2017. Adjunto a esta carta encontrará los comentarios de los revisores.

Para que el trabajo sea incluido en las memorias del congreso, es requisito cumplir con los siguientes puntos:

- Que al menos uno de los autores del artículo se inscriba y asista al evento para su presentación. La fecha límite para la inscripción de autores es el **24 de marzo de 2017**.
- Proporcionar, a más tardar el **24 de marzo de 2017**, los siguientes datos de **todos los autores** para el trámite del ISBN de la memoria del evento:
  - Nombre completo sin abreviaturas
  - Correo electrónico
  - Nacionalidad
- Realizar las modificaciones de los revisores al preparar la versión final del artículo.
- El trabajo deberá tener el formato del congreso que le adjuntamos.
- Enviar la versión final del artículo en versión PDF en alta calidad. El nombre del archivo debe ser el número de referencia que le fue asignado, a más tardar el **7 de abril 2017** al correo [icastro@itlp.edu.mx](mailto:icastro@itlp.edu.mx).

Tan pronto como tengamos disponible el programa detallado del evento, se presentará en la página Web del congreso, para que usted lo consulte y tenga la información precisa de la fecha, hora y lugar de su presentación.

Agradecemos su participación a este evento y esperamos contar con su valiosa presencia.

Atentamente,

**Comité Técnico del CIRC**



# Segmentación de células del ciclo estral, Comparativa entre: Otsu, Media Aritmética, Máxima Entropía y Entropía Cruzada.

Leonardo Delgado Toral\*<sup>§</sup>, Aldrin Barreto Flores\*, Veronica Edith Bautista Lopez<sup>†</sup>,  
Salvador E. Ayala Raggi\* y Francisco Ramos Collazo<sup>‡</sup>

\*Facultad de Ciencias de la Electrónica, Benemérita Universidad Autónoma de Puebla Puebla, México.

<sup>†</sup>Facultad de Ciencias de la Computación, Benemérita Universidad Autónoma de Puebla Puebla, México.

<sup>‡</sup>Bioterio Claude Bernard, Benemérita Universidad Autónoma de Puebla Puebla, México.

<sup>§</sup>Email: leonardo.delgadot@alumno.buap.mx

**Resumen**—El presente trabajo muestra una comparación entre algoritmos clásicos de umbralización (Otsu, Media Aritmética, Máxima Entropía, Entropía Cruzada) aplicados en los espacios de color CIELAB en el canal A, HSI en el canal I y en la escala de grises, con el objetivo de segmentar las células del ciclo estral (epiteliales nucleadas, epiteliales cornificadas y leucocitos) en roedores en muestras teñidas con eosina y hematoxilina. La segmentación se aplica en las cuatro etapas del ciclo: Estro, Proestro, Metaestro y Diestro.

**Index Terms**—Ciclo Estral, CIELAB, Entropía Cruzada, Máxima Entropía, Media Aritmética, Otsu.

**Abstract**—The present work shows a comparison between classical algorithms of thresholding (Otsu, Arithmetic Mean, Maximum Entropy, Cross-Entropy) applied in CIELAB color spaces in channel A, HSI in channel I and grayscale, this for segmented estrous cells (nucleated epithelial, cornified epithelial and leukocyte) in rodents in samples stained with eosin and hematoxylin. The segmentation is applied in the four stages of the cycle: Estro Proestro Metaestro and Diestro.

**Index Terms**—Arithmetic Mean CIELAB, Cross-Entropy, estrous cycle, Maximum Entropy, Otsu.

## I. INTRODUCCIÓN

El ciclo estral es el ciclo reproductor de la rata, dura de 4 a 5 días. Las fases del ciclo son: proestro, estro (al final de la cual ocurre la ovulación), metaestro y diestro [1]. La identificación del ciclo estral es un proceso manual sujeto a errores de interpretación. La importancia de este radica en la programación de la fertilización de los roedores, esto se traduce en un control de población lo cual conlleva a tener roedores de edades controladas para las distintas actividades de investigación o capacitación. Para la identificación del ciclo estral la persona debe realizar observación en el microscopio de las muestras tomadas donde debe observar:

- Proestro. Abundan las células epiteliales de gran tamaño y con núcleo, no hay leucocitos. Dentro de esta fase se pueden distinguir varias subfases según el número de células y tamaño de los núcleos. En proestro tardío las células epiteliales superficiales empiezan a transformarse en células cornificadas, con aspecto escamoso, anucleadas y transparentes de forma irregular, que aparecerán en el

frotis claramente distinguibles de las células nucleadas [1].

- Estro. Hay gran abundancia de células anucleadas escamosas, que pueden aparecer formando masas blancas e irregulares. Proceden de la descamación del epitelio superficial. [1].
- Metaestro. Se caracteriza por la presencia de algunos leucocitos además de células cornificadas anucleadas. Los leucocitos se reconocen bien por su pequeño tamaño y la forma irregular de su núcleo, que aparece fuertemente teñido. [1].
- Diestro. Los leucocitos invaden el epitelio y son las células predominantes en los frotis. Empiezan a aparecer células epiteliales nucleadas, más abundantes cuanto más avanzado sea el diestro [1].

En las figuras 1, 2, 3 y 4 se observan las cuatro etapas del ciclo estral.

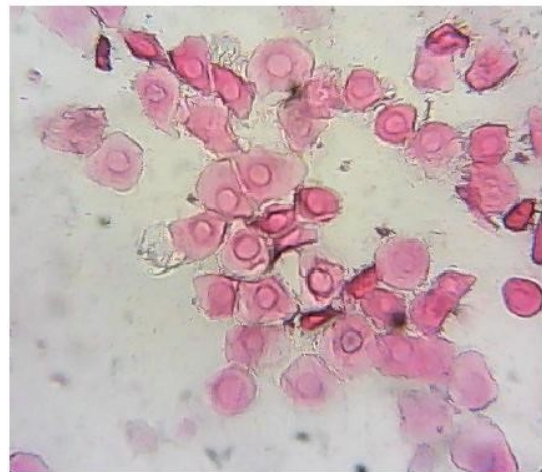


Figura 1. Proestro. Abundan las células epiteliales de gran tamaño y con núcleo, no hay leucocitos. Muestras proporcionadas por el bioterio Claude Bernard de la BUAP. La imagen fue tomada con la cámara Logitech c170 acoplada a un microscopio con un objetivo de 40 y un ocular de 10.

# Classification of the Estrous Cycle Through Texture and Shape Features

Leonardo Delgado <sup>1</sup>, Gerardo Hernandez <sup>2</sup>, Erik Zamora<sup>3</sup>, Humberto Sossa <sup>4</sup>, Aldrin Barreto <sup>5</sup>,  
Francisco Ramos <sup>6</sup>, Rosalina Reyes <sup>7</sup>

<sup>1</sup>Faculty of Electronics Sciences, Benemérita Universidad Autónoma de Puebla, Puebla, México  
Email: leonardo.delgadot@alumno.buap.mx

<sup>2</sup>Instituto Politécnico Nacional (CIC), México, México  
Email: gerardohernandez.hernandez@gmail.com

<sup>3</sup>Instituto Politécnico Nacional (UPIITA), México, México  
Email: ezamora1981@gmail.com

<sup>4</sup>Instituto Politécnico Nacional (CIC), México, México  
Email: hsossa@cic.ipn.mx

<sup>5</sup>Faculty of Electronics Sciences, Benemérita Universidad Autónoma de Puebla, Puebla, México  
Email: aldrin.barreto@correo.buap.mx

<sup>6</sup>Vivarium Claude Bernard, Benemérita Universidad Autónoma de Puebla, Puebla, México  
Email: bio\_paco@yahoo.com.mx

<sup>7</sup>Faculty of Biological Sciences, Benemérita Universidad Autónoma de Puebla, Puebla, México  
Email: rreyesluna@hotmail.com

**Abstract**—We show, for the first time, an autonomous classification of the estrous cycle (the reproductive cycle in rats). This cycle consists of 4 stages: Proestrus, Estrus, Metestrus and Diestrus. The short duration of the cycle in rats makes them an ideal model for research about changes that occur during the reproductive cycle. Classification is based on the cytology shown by vaginal smear. For this reason, we used texture and shape features on the gray level color space and CIELAB color space on channels A and B, which were classified using support vector machines (SVM) and the artificial neural network multilayer perceptron (MLP). As dataset of 412 images of estrous cycle was used. It was divided into two sets. The first contains all four stages, the second contains two classes. The first class is formed by the stages Proestrus and Estrus and the second class is formed by the stages Metestrus and Diestrus. The two sets were formed to solve the main problems, the research of the reproductive cycle and the reproduction control of rodents. For the first set, we obtained an 87% of validation accuracy and 100% of validation accuracy for the second set using the multilayer perceptron. The results were validated through cross validation using 5 sets and  $F_1$  metric.

**Index Terms**—Estrous cycle, GLCM, SVM.

## I. INTRODUCTION

The estrous cycle is the period between two ovulations and defines the receptivity of the female for its reproduction. The short duration of the cycle in rats makes them an ideal model for investigating the changes that occur during the reproductive cycle [1], [2], [3], [4]. The cycle is used for reproduction control of rodents, obtaining high fertility rates, scheduling of production on specific dates, knowing the time of gestation, and the development of embryos at specific ages. The estrous cycle in rats lasts four days and is characterized by the stages: Proestrus, Estrus, Metestrus and Diestrus, which

can be determined according to the cell types observed in the vaginal smear [5]. The objective of this work is provide a tool for automatic determination of estrous cycle stages through the image processing and neuronal networks. We make a comparison between results reached by SVM (linear kernel, Cramer and Singer, polynomial kernel and radial base kernel) and MLP. The gray level co-occurrence matrix (GLCM) and the compacity factor were used as features.

The rest of the paper is organized as follows. Section 2 provides a review of classical methods used for classifying the estrous cycle and shows the similarities between Papanicolaou (PAP) cells and cells of the estrous cycle. Section 3 introduces our approach for classifying the estrous cycle. We present several experiments and results, as well as a short discussion in section 4 to show the effectiveness of our proposal. Finally, section 5 gives our conclusions and directions for future research.

## II. PREVIOUS WORK

Shannon L. Byers [5] describes a variety of methodologies used for classifying the estrous cycle. All of them need specialized people for being performed (table.I). In the same paper is presented with a graphic tool for classifying the estrous cycle (Fig. 1).

Claudia Caligioni [6] describes the process to follow for classifying the estrous cycle through vaginal smear. In her work shows that is not necessary dyed the sample to classify it. According to her work and the work of Marcondes [7] the percentage of each kind of cell present in the vaginal smear must be:

# Variación del Algoritmo K-Means para Segmentar Células del Ciclo Estral

Leonardo Delgado Toral<sup>1</sup>, Aldrin Barreto Flores<sup>1</sup>, Verónica Edith Bautista López<sup>2</sup>, Salvador Eugenio Ayala Raggi<sup>1</sup>, Francisco Ramos Collazo<sup>3</sup>

<sup>1</sup> Facultad de Ciencias de la Electrónica.

<sup>2</sup> Facultad de Ciencias de la Computación.

<sup>3</sup> Bioterio Claude Bernard.

Benemérita Universidad Autónoma de Puebla

Prolongación 24 Sur S/N Ciudad Universitaria, San Manuel, 72570, Puebla, Pue

leonardo.delgadot@alumno.buap.mx

**Resumen.** El trabajo presenta una variación del algoritmo K-Means aplicado para la segmentación de las células del ciclo estral. El ciclo estral es el ciclo reproductor de la rata, se compone por cuatro etapas: Estro, Proestro, Metaestro y Diestro, estas etapas contienen principalmente células: epiteliales nucleadas, epiteliales cornificadas y leucocitos. La proporción de las células en la muestra determina la etapa, por esta razón un primer paso para la clasificación del ciclo estral es la segmentación de estas. Los resultados obtenidos muestran que al algoritmo propuesto le toma menos del 30% del tiempo en ejecutarse que a los algoritmos de K-Means y K-Means++ además de alcanzar centros similares.

**Palabras Clave:** Ciclo estral, K-Means, K-Means++.

## 1 Introducción

El ciclo estral es el ciclo reproductor de la rata, dura de 4 a 5 días. Las fases del ciclo son (Fig. 1): Proestro, Estro, Metaestro y Diestro. Su identificación es un proceso manual sujeto a errores de interpretación. La importancia de este radica en la programación de la fertilización de los roedores para las distintas actividades de investigación o capacitación [1], [2].

Para determinar el ciclo estral existen diversas metodologías las cuales son descritas por [3], dentro de estas la que presenta el mejor desempeño es la observación de las células del epitelio vaginal [2], [3]:

- **Proestro.** Abundan las células epiteliales de gran tamaño y con núcleo, no hay leucocitos [3].
- **Estro.** Abundancia de células anucleadas escamosas, que pueden aparecer formando masas blancas e irregulares [3].
- **Metaestro.** Se caracteriza por la presencia de algunos leucocitos además de células cornificadas anucleadas. Los leucocitos se reconocen bien por su pequeño tamaño y la forma irregular de su núcleo, que aparece fuertemente teñido [3].



"Año del Centenario de la Promulgación de la Constitución Política de los Estados Unidos Mexicanos"  
"60 Aniversario del CECyT 14 Luis Enrique Erro"  
"60 Aniversario del Patronato de Obras e Instalaciones"  
"50 Aniversario de la COFAA-IPN"  
"30 Aniversario de la Unidad Profesional Interdisciplinaria de Biotecnología"

Ciudad de México a 04 de octubre de 2017.

**A QUIEN CORRESPONDA:**

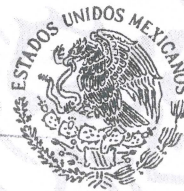
Por medio de este conducto hago constar que el **C. LEONARDO DELGADO TORAL**, estudiante de la **Maestría en Ingeniería Electrónica Opción Instrumentación Electrónica (MIEL)** de la **Benemérita Universidad Autónoma de Puebla**, concluyó satisfactoriamente una estancia bajo mi supervisión, durante el periodo comprendido del 06 de abril al 06 de julio del presente año, participando en el proyecto denominado **"DETECCIÓN DEL CICLO ESTRAL Y CONTEO ESPERMÁTICO"**, desarrollando satisfactoriamente cada una de las actividades encomendadas.

Quedo a sus órdenes para cualquier asunto relacionado a la presente.

ATENTAMENTE

"LA TÉCNICA AL SERVICIO DE LA PATRIA"

**DR. HUMBERTO SOSSA AZUELA**  
**PROFESOR INVESTIGADOR**  
hsossa@cic.ipn.mx



INSTITUTO POLITÉCNICO NACIONAL  
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN  
SUBDIRECCIÓN ACADEMICA

Vo.Bo

**DR. MOISÉS SALINAS ROSALES**  
**SUBDIRECTOR ACADÉMICO**

**"Certificado ISO9001:2008 No. 2013CRE-253"**

---

# Apéndice D

## Acronimos

K-NN:	K-Nearest Neighbors.
PCNN:	Pulse Coupled Neuronal Network.
SVM:	Support Vector Machine.
MLP:	Multilayer Perceptron.
PAP:	Prueba de Papanicolau.
K-Means:	Nombre del algoritmo de agrupación propuesto por Stuart Lloyd.
K-MeansP:	Variación del algoritmo de agrupación propuesto por Stuart Lloyd para segmentar imágenes en escala de grises.
C-V:	Algoritmo de segmentación por contornos activos propuesto por Tony Chan y Luminita Vese (Chan-Vase)
Otsu:	Nombre que se le asigna al algoritmo de umbralado no paramétrico propuesto por Nobuyuki Otsu en 1979.
PCA:	Principal Component Analysis