



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®

**INSTITUTO TECNOLÓGICO DE
DURANGO**

**INSTITUTO TECNOLÓGICO
DEL VALLE DEL GUADIANA**

**DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN**



**“Análisis de calidad del semen de ganado vacuno con fines
reproductivos, con visión artificial”**

TESIS

Que como parte de los requisitos para obtener el grado de

Maestría en Ingeniería

Presenta:

Christian Eduardo Morales Silva

Director de tesis:

Dr. Rubén Guerrero Rivera

Co-Director:

M.C. Aurelio Castillo Liñán

Durango, Dgo. México, agosto, 2024.





Análisis de calidad del semen de ganado vacuno con fines reproductivos, con visión artificial

Presenta:

Christian Eduardo Morales Silva

COMITÉ TUTORIAL

Dr. Rubén Guerrero Rivera

Director

Firma

M.C. Aurelio Castillo Liñán

Codirector

Firma

M.C. Norma Alicia García Vidaña

Asesor

Firma

Dr. Francisco Javier Godínez García

Asesor

Firma

M.C. Norma Alicia García Vidaña

Coordinadora del programa de la Maestría en Ingeniería.

Dr. Francisco Javier Godínez García

Jefe de la División de Estudios de Posgrado e Investigación

Durango, Dgo. México

agosto de 2024



Victoria de Durango, Dgo., a **10 / Septiembre / 2024.**

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
DEPI / C / 365 / 24.

ASUNTO: Autorización de Tema de Tesis de Maestría.

C. CHRISTIAN EDUARDO MORALES SILVA
No. DE CONTROL G14040926
P R E S E N T E .

Con base en el Reglamento en vigor y teniendo en cuenta el dictamen emitido por el Jurado que le fue asignado, se le autoriza a desarrollar el tema de tesis para obtener el **Grado de Maestro en Ingeniería** cuyo título es:

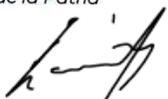
“ANÁLISIS DE CALIDAD DEL SEMEN DE GANADO VACUNO CON FINES REPRODUCTIVOS, CON VISIÓN ARTIFICIAL”

CONTENIDO:

	RESUMEN
CAPÍTULO I	NATURALEZA Y DIMENSIÓN DEL ESTUDIO
CAPÍTULO II	MARCO TEÓRICO
CAPÍTULO III	MATERIALES Y CASO
CAPÍTULO IV	RESULTADOS Y DISCUSIÓN
CAPÍTULO V	CONCLUSIONES Y RECOMENDACIONES
	REFERENCIAS
	ANEXOS

ATENTAMENTE.

Excelencia en Educación Tecnológica®
"La Técnica al Servicio de la Patria"



C. FRANCISCO JAVIER GODÍNEZ GARCÍA
JEFE DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN



FJGG'ammc.





Victoria de Durango, Dgo., a **10 / Septiembre / 2024.**

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN
DEPI / C / 366 / 24.

ASUNTO: Autorización de Impresión de Tesis de Maestría.

C. CHRISTIAN EDUARDO MORALES SILVA
No. DE CONTROL G14040926
P R E S E N T E .

De acuerdo al reglamento en vigor y tomando en cuenta el dictamen emitido por el jurado que le fue asignado para la revisión de su trabajo de tesis para obtener el **Grado de Maestro en Ingeniería**, esta División de Estudios de Posgrado e Investigación le autoriza la impresión del mismo, cuyo título es:

"ANÁLISIS DE CALIDAD DEL SEMEN DE GANADO VACUNO CON FINES REPRODUCTIVOS, CON VISIÓN ARTIFICIAL"

Sin otro particular de momento, quedo de Usted.

ATENTAMENTE.

Excelencia en Educación Tecnológica®
"La Técnica al Servicio de la Patria"

C. FRANCISCO JAVIER GODÍNEZ GARCÍA
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN



FJGG'ammc.





Agradecimientos

En primero y, sobre todo, agradezco a Dios, mi Señor y Salvador, quien me ha dado la fuerza, la sabiduría y la inteligencia para superar cada obstáculo y alcanzar la meta de concluir esta tesis. Sin su guía y apoyo, nada de esto habría sido posible.

A mi amada esposa, Judith Elizabeth Pope Cortez, quien ha sido mi compañera incondicional en este viaje. Gracias por tu amor, paciencia y apoyo en cada momento. Tu presencia ha sido un faro de esperanza y motivación en los días más difíciles. También agradezco a mi madre, Lidia Isela Silva Díaz, por su amor, enseñanzas y por ser siempre una fuente de inspiración. A mi madre, Lidia Isela Silva Díaz, por tu amor, enseñanzas y por ser siempre una fuente de inspiración.

A mi padrastro, Daniel González Cossio, gracias por tu apoyo constante y por ser una figura paterna excepcional en mi vida.

A mis abuelos, Roberto Silva Martínez y María Eugenia Silva Diaz, quienes han sido pilares de amor y sabiduría. Gracias por enseñarme el valor del esfuerzo y la dedicación.

A mis hermanas, Daniela Morales Silva y Lidia Morales Silva, por su cariño y apoyo incondicional. Su confianza en mí ha sido una motivación constante.

A mis asesores, el Dr. Rubén Guerrero Rivera, el Dr. Aurelio Castillo Liñán, el Dr. Francisco Javier Godínez García y la Dra. Norma Alicia García Vidaña, quienes con su orientación y conocimientos han sido fundamentales para el desarrollo de esta investigación. Su paciencia y sus consejos han sido invalorable.

Finalmente, agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) el apoyo financiero y la confianza depositada en mí para llevar a cabo este proyecto. Sin su respaldo, no hubiera podido terminar esta tesis.

A todos ustedes, mi más sincero agradecimiento.





Resumen

Este artículo presenta un software desarrollado para mejorar la confiabilidad en la evaluación de la viabilidad espermática en bovinos, reduciendo los errores en los análisis tradicionales. El sistema emplea procesamiento avanzado de imágenes digitales y redes neuronales para clasificar morfológicamente los espermatozoides bovinos, utilizando técnicas como la segmentación por regiones y la umbralización de imágenes. Gracias a la utilización de Deep Learning, el sistema se adapta con el tiempo, alcanzando una precisión del 84 % en sus predicciones. Esta herramienta representa una mejora significativa para los programas de reproducción y mejora genética bovina, ya que permite realizar un análisis más eficiente y fiable. Además, su implementación se adapta a las condiciones de los centros ganaderos, lo que lo convierte en una solución accesible y rentable.





Abstract

This paper presents software developed to improve the reliability of sperm viability assessment in bovines, reducing errors in traditional analyses. The system employs advanced digital image processing and neural networks to classify bovine sperm morphologically, using techniques such as region segmentation and image thresholding. Thanks to the use of Deep Learning, the system adapts over time, achieving 84% accuracy in its predictions. This tool represents a significant improvement for bovine breeding and genetic improvement programmes, as it allows for more efficient and reliable analysis. In addition, its implementation can be adapted to the conditions of livestock farms, making it an accessible and cost-effective solution.





Contenido

Capítulo 1 Introducción	1
Planteamiento del problema	3
Objetivos	3
Objetivo general	3
Objetivos específicos.....	3
Justificación	3
Delimitación del problema.....	4
Estado del arte	4
Capítulo 2 Capítulo Marco teórico	6
Historia de la Ganadería en México	6
Economía de la Ganadería en México	7
Ganadería en el estado de Durango	7
Evaluación de la calidad del semen bovino	8
Morfología Espermática	9
Sistema de captura de imagen	10
Visión artificial.....	11
Detección del objeto	12
Segmentación y Umbralización	12
Obtención de parámetros	13
Deep Learning	13
Capítulo 3 Materiales y métodos.....	14
3.1 Diseño del algoritmo	14
Desarrollo de funciones.....	16
Imagen	16
Segmentación y umbralización.....	17
Detección del objeto	19
Obtención de los distintos parámetros	21
3.2 Deep Learning	26
3.2.1 Preparación del ambiente de trabajo	26
3.2.2 Data set de imágenes	26





3.2.3 Construcción del modelo.....	27
3.2.4 Fase convolución	27
3.2.5 Fase del clasificador.....	28
3.2.6 Preprocesamiento de datos	28
3.2.7 Arquitectura del modelo	29
3.2.8 Clasificador	30
3.2.9 Entrenamiento del modelo.....	31
Capítulo 4 Resultados y discusión.....	34
4.1 Resultados medidos	34
4.1.1 Planteamiento del análisis.....	35
4.1.2 Prueba de hipótesis para el área y perímetro	35
4.1.3 Intervalos de confianza.....	36
4.1.4 Análisis del comportamiento ya segmentado	39
4.2 Evaluación del modelo de aprendizaje profundo	40
4.2.1 Realizando pruebas	41
4.2.2 Resultados	42
Capítulo 5 Conclusiones y recomendaciones.....	43
Conclusión.....	43
Mejoras	44
Referencias	44
Anexos	47
1.1.1 Código Pycharm.....	47
1.1.2 Código Deep Learning.....	61





Índice de figuras

Figura 2.1 Revolucionario mexicano con su ganado.....	6
Figura 2.2 Producto interno bruto (PIB) del sector pecuario en México del primer trimestre de 2017 al cuarto trimestre de 2023	7
Figura 2.3 Principales entidades federativas productoras de ganado bovino (2019)	8
Figura 2.4 Ganado bovino	8
Figura 2.5 Anomalías en cabezas espermáticas	9
Figura 2.6 Microscopio de contraste de fase.....	10
Figura 2.7 Visión artificial aplicada	11
Figura 3.1 Pantalla principal del software	15
Figura 3.2 Cabeza de espermatozoide normal	17
Figura 3.3 Pantalla del software	18
Figura 3.4 Histograma de una cabeza de espermatozoide.....	19
Figura 3.5 Histograma y detección de la cabeza espermática.	20
Figura 3.6 Método Convex Hull	21
Figura 3.7 Segmentación de una cabeza espermática.....	22
Figura 3.8 Histograma de los canales rojo, verde y azul.....	23
Figura 3.9 Pantalla del software en funcionamiento con cabeza normal.....	24
Figura 3.10 Cabeza anormal segmentada.	25
Figura 3.11 Pantalla del software en funcionamiento con cabeza normal.	25
Figura 3.12 Interacción de Google Colaboratory con el Drive de datos.	26
Figura 3.13 Data augmentation.....	29
Figura 3.14 Red Neuronal.....	29
Figura 3.15 Arquitectura del clasificador.....	30
Figura 3.16 Arquitectura de la red neuronal convolucional	31
Figura 3.17 Entrenamiento de la red neuronal.	32
Figura 3.18 Evolución del error.	33
Figura 3.19 Evolución de la exactitud.....	33
Figura 4.1 Histograma de los valores de área.....	37
Figura 4.2 Intervalo de confianza para el área.	37
Figura 4.3 Histograma de la prueba de normalidad.	38
Figura 4.4 Intervalo de confianza para el perímetro.	39
Figura 4.5 Matriz de confusión.....	41
Figura 4.6 Valores obtenidos del entramiento de la red neuronal.....	41
Figura 4.7 Cabeza anormal espermática.	42
Figura 4.8 Predicción de la red neuronal.....	42





Índice de tablas

Tabla 2.1 Tabla de concentración espermática	10
Tabla 4.1 Muestra de datos de área y perímetro.	34
Tabla 4.2 Datos generales de la muestra.	35
Tabla 4.3 Valores t y p de la prueba de hipótesis para el área.	35
Tabla 4.4 Valores t y p de la prueba de hipótesis para el perímetro.	36
Tabla 4.5 Valores de área y perímetro con su respectivo intervalo.	38
Tabla 4.6 Valores t y p de la prueba de hipótesis para el área después de cambiar los datos.	39
Tabla 4.7 Valores t y p de la prueba de hipótesis para el perímetro después de cambiar los datos.	40

Índice de ecuaciones

Ecuación 3.1 Cálculo de probabilidad de cada nivel de intensidad	17
Ecuación 3.2 Cálculo del vector gradiente para cada píxel	20
Ecuación 3.3 Cálculo de la operación de convolución en dos dimensiones.	27
Ecuación 3.4 Cálculo de la unidad de la red neuronal.	28
Ecuación 3.5 Función de Costo/Loss (Binary Cross-Entropy).....	31
Ecuación 3.6 Ecuación de optimización.....	32
Ecuación 4.1 Calculo del valor z	36
Ecuación 4.2 Calculo del valor Anderson-Darling	36
Ecuación 4.3 Calculo del intervalo de confianza t de Student.	38





Capítulo 1 Introducción

La producción de ganado bovino en México es una de las actividades más importantes para el país pues esta proporciona el abasto de carne para el consumo de sus habitantes que también resulta significativa en la cuestión económica.

Su impacto es tal que en términos del producto interno bruto (PIB) se ha estimado que la ganadería es el 1.1% del PIB agroalimentario del país.

Para el estado de Durango, este se encuentra en el tercer lugar a nivel nacional en la exportación de ganado Instituto Mexicano de la Carne. Además, cuenta con más de seis millones de hectáreas de agostaderos y alrededor de 50 mil familias se dedican a ello, de ahí que la actividad del sector ganadero sea un pilar para la economía en Durango.

Debido a esto no sólo en México se empezaron a implementar técnicas que buscaban mejorar la genética y producción de la especie. Esto se remonta a inicios de la década de 1940 ya que, a su vez hubo un gran avance en el empleo de la inseminación artificial que facilitó la distribución de una genética superior sin requerir la presencia del toro. Pues antes la reproducción del ganado bovino era principalmente con el apareamiento natural algo que conllevaba algunas desventajas significativas como riesgos de enfermedades venéreas y otras patologías, también existía cierta limitación pues la vaca solo estaba fértil durante una temporada y el toro nada más servía para una cantidad limitada de ellas. Esto solo era el comienzo pues en este tipo de apareamiento también hay costos elevados tanto por su alimentación, cuidados y manejo especializado, por ende, no se podía garantizar el control de las características específicas para la herencia, resultando así que no se podía mejorar o seleccionar rasgos específicos dentro de un rebaño.

Como parte de los avances tecnológicos nació el análisis espermático que tenían el fin de una mejora genética del ganado y lograr una eficiencia en la reproducción de este. Ahora, este es un examen de laboratorio que revisa la salud reproductiva del semen, donde se busca medir ciertos aspectos de los espermatozoides y del semen, tales como la concentración, el volumen, la movilidad, morfología, etcétera.

Y este resultado muy significativo, pues daba un diagnóstico de las posibles causas de infertilidad en los machos, dando información útil que después se utilizaría para tratamientos que mejoraran la fertilidad y no menos importante conseguir un alto valor genético, que es vital para maximizar la mejora genética y el éxito reproductivo.

Obteniendo mejoras en las siguientes generaciones de ganado, como la calidad de la carne, una mayor resistencia a enfermedades, mayor producción de leche. A su vez, el seleccionar muestras seminales que contaban con una alta movilidad y morfología adecuada se mejoró la tasa de fertilidad, incluso para especies que se encontraban en peligro de extinción. También se demostró que hay una reducción en los costos y la transmisión de enfermedades hereditarias.





En el caso de la industria ganadera en el país, debido a la falta de recursos y bajos niveles de educación, no se cuentan con las disposiciones para la implementación de equipo computacional para el proceso de la revisión de muestras seminales. Entonces esta actividad de observación y evaluación no se emplea en absoluto o es realizada por un especialista en el área, dando como resultado ciertas deficiencias a la hora de obtener una conclusión realista y confiable, ya que, la observación puede ser subjetiva, consumir mucho tiempo y ser agotadora para el trabajador. Esto resulta en una problemática para el sector pecuario pues la calidad del semen es una pieza fundamental para que la tecnología reproductiva de inseminación artificial y criogenización sea funcional en las especies ganaderas.

Entonces el propósito de esta investigación es mejorar el proceso de revisión de muestras seminales utilizando la automatización que otorga la visión artificial, complementada con técnicas de Deep Learning y así simplificar el proceso y mejora del diagnóstico, se reducen las deficiencias que traía consigo el utilizar a un trabajador para esto.

Para el caso de la unidad ganadera en Durango cuentan ya con un especialista y microscopio para realizar este proceso. Pero, aunque los softwares actuales de sistemas CASA tiene análisis muy completos y fiables, tienen algunos defectos como su alto costo, mantenimiento y altos requerimientos de hardware y software. De ahí que este trabajo sea una alternativa más redituable para la ganadería en México.

El peso del software desarrollado será adaptable a las condiciones de imagen y resolución que se tengan disponibles en el centro ganadero, así como las muestras de semen que se pueden obtener de ahí, y esto se emplea para la especie a la cual se desea mejorar su vigorosidad. Y en la búsqueda de tener un proceso más avanzado y preciso el uso de técnicas de Deep Learning mejorará la identificación y clasificación de los componentes espermáticos, trayendo la oportunidad de que el software sea adaptable a diferentes tipos de imágenes sin necesidad de ajustes manuales y una mejora constante.

Por tanto, la utilización del procesamiento de software tendrá su aplicación para los microscopios que se tenga ya a disposición, tampoco se podrá emplear en el análisis espermático de otras especies ajenas del bovino y que las condiciones del microscopio se encuentren correctamente ajustadas para las mediciones.

La calidad de la muestra seminal está dada por varios factores, tales como: morfología, viabilidad, concentración y motilidad. Para validar el área y el perímetro obtenidos mediante el procesador de imágenes, se ha recurrido a la bibliografía que incluye datos experimentales recogidos por otros investigadores. Se encontró que los valores que determinamos se sitúan dentro de un rango similar al reportado en estos estudios. Esta correlación refuerza la precisión y la fiabilidad de nuestras mediciones realizadas con el procesador de imágenes, demostrando su alineación con los hallazgos ya documentados en este campo de estudio. Para esta revisión es necesario la adquisición de imágenes de la muestra, después se le aplican diferentes procesamientos con el fin de segmentar la





información requerida y a partir de imágenes procesadas extraer los datos de las cabezas de espermatozoides, ingresar el data set en la red neuronal y por último conjuntar eso en un software operable por cualquier persona.

Planteamiento del problema

Objetivos

Objetivo general

Determinar la calidad del semen de ganado vacuno con fines reproductivos, mediante imágenes digitales.

Objetivos específicos

- Investigar las características morfológicas que definen la calidad del semen bovino para confeccionar el software.
- Seleccionar una metodología para el preprocesamiento de las imágenes que mejore la eficiencia del modelo de visión artificial y del aprendizaje profundo.
- Definir un algoritmo para la clasificación y caracterización detallada de los espermatozoides.
- Evaluar y cuantificar los resultados obtenidos por el software a través de análisis estadísticos rigurosos.

Justificación

La industria ganadera históricamente ha jugado un importante papel en la economía y sociedad de México. Una eficiencia reproductiva es fundamental para la sostenibilidad y rentabilidad de esta industria representada por sus componentes en la seguridad alimentaria que aportan una producción de carne y de leche que alimenta a miles de mexicanos.

Sin embargo, actualmente siguen sin realizarse técnicas avanzadas para la evaluación de la calidad del semen, en distintas partes del país todavía se siguen empleando métodos tradicionales para la reproducción del ganado lo que lleva a limitaciones en la mejora reproductiva, que enfrentan limitaciones en términos de precisión, eficiencia y subjetividad en la interpretación de resultados, que podrían llevar a un decaimiento de la reproducción animal lo que conllevaría una transmisión de enfermedades y baja genética.

Por lo que la implementación de nuevas tecnologías tales como visión artificial y aprendizaje profundo pueden aumentar la precisión de evaluaciones y también facilitar los procesos obteniendo resultados reproducibles, superando así los desafíos de los enfoques convencionales.





Este trabajo de tesis, por lo tanto, no solo contribuiría al campo académico y científico proporcionando una solución tecnológica avanzada, sino que también respondería a necesidades prácticas de la industria ganadera, promoviendo avances significativos en programas de inseminación artificial y mejoramiento genético.

Delimitación del problema

- Disponibilidad y calidad de los equipos de microscopía en las unidades ganaderas.
- Costo inicial elevado de implementación de la tecnología, incluyendo adquisición de equipos y software.
- Dependencia de la calidad de las imágenes capturadas para el análisis, que puede verse afectada por factores ambientales y técnicos.
- Nivel de capacitación y adaptación del personal ganadero a las nuevas tecnologías.
- Variabilidad en la precisión de los resultados comparados con los métodos tradicionales de evaluación.

Estado del arte

El campo del análisis de la calidad del semen bovino no es algo nuevo para la ciencia. Esta práctica ha sido implementada en beneficio del sector ganadero, y los avances tecnológicos tales como; los sistemas computacionales, cámaras digitales y microscopios también han influenciado en esta técnica, ya que en las últimas décadas se han realizado distintos softwares comúnmente llamados C.A.S.A en el afán de este proceso sea más preciso y fiable.

(Calderón, 2013) Diseñó un sistema C.A.S.A. a través del uso de procesamiento de imágenes para el centro agropecuario "La granja", este contaba con un microscopio que proporcionaba imágenes detalladas a grandes ampliaciones (40x). Para el dimensionamiento se emplearon las características más significativas de la calidad del semen (motilidad y concentración), intervalos más cortos para la medición, y la nula utilización de químicos. Debido a las imágenes otorgadas por el microscopio, él selecciono a MATLAB R2011 como software para diseñar el código encargado de la adquisición, análisis de características y sección gráfica de operación. Observó que las propiedades de motilidad y el tipo de movimiento se relacionan directamente, sin ser de forma cuantitativa inicialmente pues se les brinda un valor para ser interpretadas, dificultando su medición de manera precisa, además de que los análisis y la determinación de las propiedades espermográficas a través del método antiguo tienen mayor error al uso de imágenes digitales.

(Montaña, 2016) Elaboró un sistema C.A.S.A. primeramente empleo modelos del tipo Gaussiano que detectaban de manera precisa el movimiento de los espermatozoides, después para segmentar cada uno de los espermatozoides aplico el discriminante de Fisher que es una técnica estadística que separa las características de interés. Para procesar las imágenes empleo operadores morfológicos que filtraron y redefinieron las imágenes a valores binarios que mejoraron la claridad





y precisión de los datos. Por último, con el fin de conseguir las trayectorias de los espermatozoides utilizo el filtro de Kalman y el algoritmo de Munkres. Como consecuencia demostró que existe una relación satisfactoria en la comprobación del análisis seminal usando el software y el análisis realizado por un experto tanto para la motilidad, mediante la comprobación Bland-Altman con una relación de 0.966 en un intervalo de confianza del 95% de (3,058- 3,19).

(Pascual-Gaspar, y otros, 2008) Utilizaron técnicas de visión computarizadas para diseñar un software CASA que sea capaz de generar resultados acordes al estándar de andrología de la OMS. Haciendo la imagen a escalas de grises aplicaron un filtro para obtener una imagen binaria con el apoyo de técnicas de segmentación para eliminar impurezas. Después por medio de la segmentación cinemática utiliza un módulo de seguimiento para cada esperma. Así con la imagen del principio y la binaria hicieron una comparación para identificar los cuerpos móviles entre diferentes tramos. Como parte final los autores dividen las imágenes en 4 secciones más pequeñas que son evaluadas cada una como forma de rastreo de espermatozoides. Obteniendo como resultados una clasificación de motilidades dependiendo de las progresiones de los espermatozoides y con un porcentaje de precisión del 93%.

(Llanos Gualtero & Beltrán Diamante, 2016) Trabajaron en el desarrollo de una herramienta computacional para el análisis seminal de concentración y viabilidad espermática en bovinos. A través de técnicas de procesamiento de imágenes, aplicaron filtros de mediana, average y Gaussianas a las imágenes obtenidas por el microscopio Olympus CX31 con el propósito de hacerlas más homogéneas. Y para obtener un análisis de la viabilidad utilizaron el método Fisher de extracción de espermatozoides vivos y muertos se empleó método de clustering. En el caso de la concentración de espermatozoides se utilizó el método canny que extrae los espermatozoides de las imágenes. Como resultados obtuvieron que los análisis realizados por el software en comparación con los que da el experto tuvieron una exactitud del 78% y una precisión del 83%.

(Farhad, M., Masud, M. M., Beg, A., Ahmad, A., & Ahmed, L. (2023). Emplearon redes neuronales convolucionales (CNN) y redes de memoria a largo y corto plazo viendo cómo estas aplicaciones podrían mejorar la precisión y eficiencia del diagnóstico en usos médicos como ultrasonidos y endoscopias. Este estudio concluye que el aprendizaje profundo presenta mejoras en la automatización del análisis, en la precisión de diagnósticos y en la eficiencia pues de ellos se puede extraer información relevante que mejora la predicción y clasificación de historias clínicas.

(Huang, L., Subramoniam, S., Challis, E., & otros autores. (2023). Consiguieron una predicción y clasificación efectiva para las etapas del Alzheimer. Con el fin de mejorar la capacidad de diagnóstico de la enfermedad emplearon principalmente imágenes de resonancia magnética con el uso de red neuronal como ResNet y métodos de clasificación consiguieron demostrar que el aprendizaje profundo ofrece una identificación significativamente mayor en las etapas del Alzheimer, lo que podría ayudar en una detección temprana de la misma.





Capítulo 2 Capítulo Marco teórico

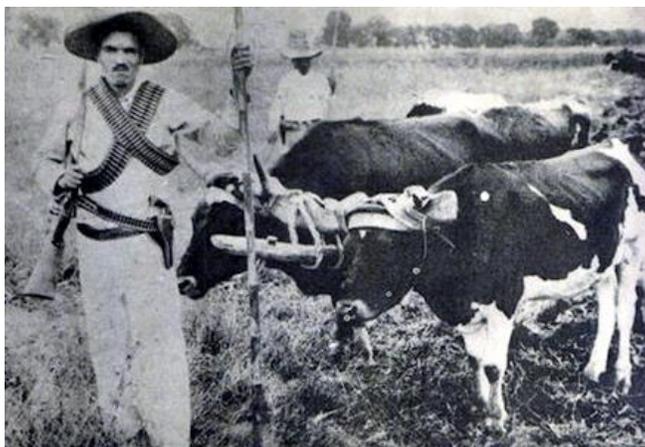
Historia de la Ganadería en México

La ganadería en México siempre ha sido importante, el gran impacto que tiene no sólo es económico sino también social, históricamente fue uno de los principales pilares que iniciaron la revolución de 1910.

El contexto es que previo a la revolución mexicana, existían unas cuantas familias poderosas que eran propietarias de vastas extensiones de tierra, lo que resultaba en una desigualdad, ya que la mayoría de los mexicanos eran arrendatarios o peones que vivían en pobreza, lo que resultó en tensiones sociales que eventualmente promulgarían la revolución.

Tiempo después, terminada la guerra, vino uno de los principales avances de la ganadería en México, que surgió de la promulgación de la constitución de 1917, en esa época este sector ya destacaba por sus exportaciones a Estados Unidos. Y a través de reformas agrarias, se logró un mejor reparto de tierras lo que consolidó un apoyo a los pequeños y medianos productores ganaderos, pues se corrigieron desigualdades extremas en la tenencia de la tierra, lo que fomentó una clase de campesinos con mayor estabilidad económica, lo que se conoce como sistema ejidal, donde la tierra era trabajada de manera colectiva y está no se podía vender.

Figura 2.1 Revolucionario mexicano con su ganado.



Fuente: Breve Historia de la Revolución Mexicana. Libros Maravillosos

Pese a que estas reformas consiguieron equilibrar la distribución de tierras y quitar la explotación del pueblo, a falta de inversión, capacitación de los trabajadores, una corta infraestructura y pocas conexiones al mercado los ejidos presentaban una menor producción que haciendas comerciales por lo que por mucho tiempo no eran viables económicamente.





Teniendo en cuenta su aporte, posteriormente se buscó a través de regulaciones más estrictas que la industria ganadera se modernizara, que apelara por técnicas para la mejora de crías, animales más sanos y un cuidado en la genética de estos.

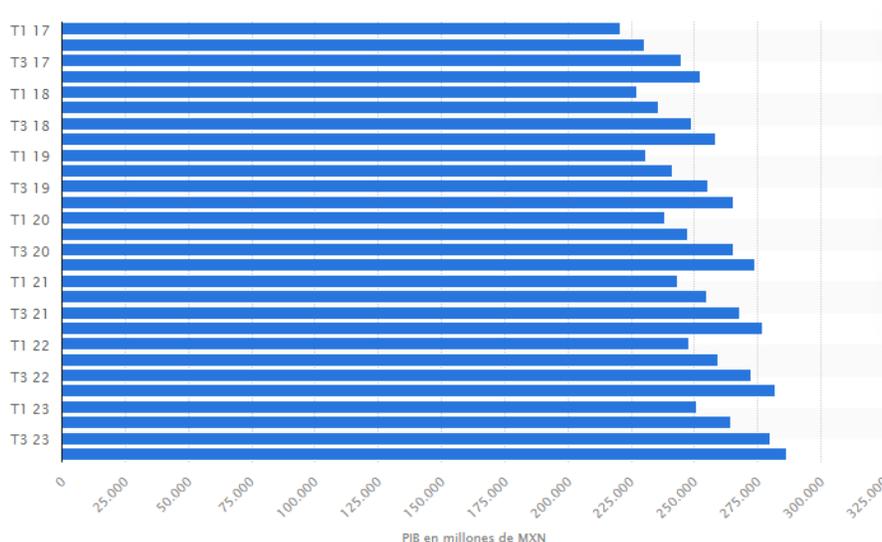
Economía de la Ganadería en México

La contribución de la ganadería en nuestro país hablando con respecto al PIB en oposición a sectores terciarios y secundarios, solo representan una fracción menor de este, pero dado su papel en la seguridad alimentaria proporcionando carne y productos lácteos, además por los insumos que provee a otras industrias se considera como una de las más importantes.

Esta industria también se caracteriza por su contribución en la generación de empleo, sobre todo en zonas rurales.

En cuestión de porcentajes el sector agrícola que incluye la ganadería contribuyó con el 4.03% al producto interno bruto en México (Statista, 2022).

Figura 2.2 Producto interno bruto (PIB) del sector pecuario en México del primer trimestre de 2017 al cuarto trimestre de 2023



Fuente: Statista(www.es.statista.com)

Ganadería en el estado de Durango

Hablando de Durango hay varios datos que respaldan la contribución de este sector para Durango. Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación (SAGARPA). (2019)

- Una producción de 370,529 toneladas de carne, situándose como el 7° lugar a nivel nacional.
- En cuanto a la leche se alcanzaron 1,110,435 mil litros, siendo este el 3° lugar nacional.



- El valor de la producción pecuaria fue de 21,869 millones de peso siendo el 70.1% del valor para el estado.
- Con respecto a la legión de la laguna Durango apporto con 8,710 millones de pesos, siendo el 84% de la producción pecuaria del estado.

Por lo que se puede constatar que la contribución economía agropecuaria de Durango y también de su significativo aporte alimentario tanto en carne como en leche.

Figura 2.3 Principales entidades federativas productoras de ganado bovino (2019)

Entidad federativa	Número de cabezas
Baja California	282,583
Baja California Sur	155,195
Campeche	650,403
Coahuila de Zaragoza	772,705
Colima	240,680
Chiapas	2,072,083
Chihuahua	1,845,995
Durango	1,931,043
Guanajuato	1,140,685
Hidalgo	524,057

Fuente: Cuéntame *INEGI*(inegi.org.mx)

Evaluación de la calidad del semen bovino

En recientes fechas la inseminación artificial ha sido una técnica reproductiva que ha impulsado el avance genético en las especies de ganado, no obstante, la efectividad de esta técnica se encuentra condicionada a la calidad del semen utilizado.

Figura 2.4 Ganado bovino



Fuente: Cuéntame *INEGI*((inegi.org.mx)

La capacidad que tiene una muestra de fertilizar se relaciona con que tenga una cantidad de espermatozoides viables, una morfología normal y una cantidad suficiente que puedan llegar al oviducto, fertilizar el óvulo y comenzar el desarrollo embrionario.

Hoy es importante mencionar que para la evaluación adecuada del semen es necesario contar con un microscopio de alta calidad. Según Bart, es fundamental que al menos 3 parámetros en la evaluación del semen congelado se cumplan, que son: la morfología, la viabilidad y el porcentaje de espermatozoides vivos por dosis.

Morfología Espermática

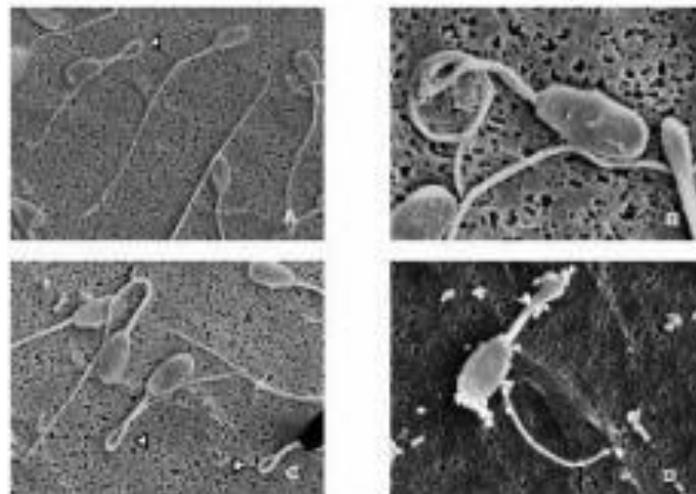
La morfología espermática se refiere a la forma y el estado del acrosoma del espermatozoide, entonces a las anomalías o deformaciones que le puede conocer como defectos primarios y secundarios.

Las deformidades se pueden clasificar en:

1. Presencia de gota citoplasmática proximal
2. Alteraciones de la cabeza, cola.
3. Presencia de gota citoplasmática distal.
4. Colas en látigo.

Se deben medir diversas dimensiones de la cabeza del espermatozoide, como la longitud [L], el ancho [W], el área (expresada en número total de píxeles) [A], y el perímetro [P]. Estas medidas permiten así un análisis detallado de cada cabeza espermática.

Figura 2.5 Anomalías en cabezas espermáticas



Fuente: *Avances sobre nutrición y fertilidad en ganado lechero*





2.3.2 Concentración

Con respecto a la concentración de espermatozoides se considera que existe una relación significativa entre esta variable y la fertilidad del toro, hoy por lo que el número de espermatozoides que se mantenga en lo que las características se consideran como normal pues mejoraría la probabilidad de una fertilización

Tabla 2.1 Tabla de concentración espermática

Clasificación	Descripción
Muy buena (MB)	Semen cremoso, granular con 750 a 1,000 millones de espermatozoides/ml o más.
Buena (B)	Semen lechoso con 400 a 750 millones de espermatozoides/ml
Suficiente (S)	Semen semejante a leche descremada con 250 a 400 millones de espermatozoides/ml
Pobre (P)	Semen translucido con menos de 250 millones de espermatozoides/ml

Fuente: Advances in Cryopreservation of Bull Sperm.

Sistema de captura de imagen

Un factor para considerar es el método de captura de las imágenes digitales, que sería la cámara de vídeo. Ya que para evaluar las capacidades de un sistema y la calidad de los datos que proporciona es necesario que la cámara cuente con una resolución de captura de imágenes de alta calidad. Se recomienda el uso de cámaras de al menos 700 68 * 576 frames. También es importante tener en cuenta el microscopio utilizar, en la cuestión de observación y análisis de espermatozoides de ganado la recomendación es utilizar un microscopio de contraste de fase que tenga una resolución óptica y sea capaz de manejar aumentos de al menos de 1000x.

Figura 2.6 Microscopio de contraste de fase





Fuente: *Nuevas tecnologías aplicadas al procesado y evaluación del semen bovino en centros de inseminación artificial*

Visión artificial

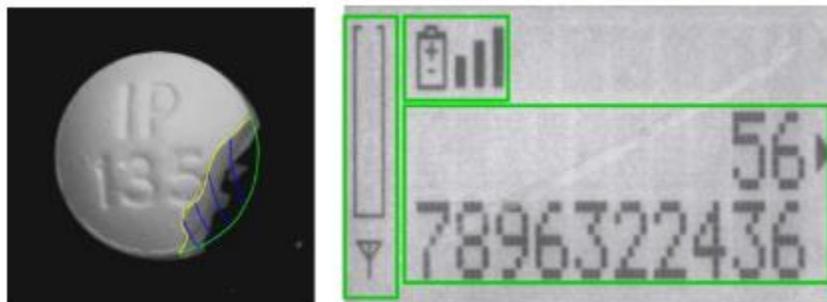
La visión artificial se entiende como una tecnología y técnicas utilizadas para emular la visión humana en diferentes aplicaciones. Esta debe abarcar la detección, el procesamiento y la interpretación de las imágenes, que realizarán tareas específicas dependiendo del entorno y su necesidad. La visión artificial se destaca por su capacidad de ejecutar tareas repetitivas con alta velocidad y precisión, superando limitaciones que tiene el ojo humano en términos de fatiga y subjetividad.

Estas técnicas han sido útiles para diferentes aplicaciones como en la robótica para la manipulación y posicionamiento preciso de objetos, también el reconocimiento de piezas mediante código de barras y otros marcadores, se destaca también en su capacidad para evaluar dimensiones y verificar la conformidad con las especificaciones de algún diseño y también puede detectar defectos o anomalías en productos.

Los componentes principales para un sistema de esta índole parten desde la iluminación donde este elemento resulta crucial para obtener resultados satisfactorios, las cámaras, los sensores de imágenes y los procesadores que analizan las imágenes y van tomando decisiones basadas en los algoritmos.

Figura 2.7 Visión artificial aplicada





Fuente: *Introducción a la visión artificial*

El uso de esta técnica proporciona una mejora para la calidad de algún producto, reducción del desperdicio y una optimización del rendimiento del proceso.

Por ende aplicar la visión artificial es indispensable en nuestros tiempos ya que tiene una capacidad para mejorar significativamente la eficiencia y la calidad en los procesos.

Detección del objeto

Hoy para lograr la detección de los bordes de un objeto se emplea un gradiente en la imagen que identifica y separa el componente principal de otras partículas o contaminantes. Es necesario calcular las diferencias de intensidad entre los píxeles adyacentes en la imagen, pues donde hubo un cambio en la intensidad esto indicaría la presencia de un borde. Este vector se debe calcular para cada píxel y se puede representar gráficamente a través de ilustrar la dirección y la magnitud de estos cambios de intensidad.

Segmentación y Umbralización

Para lograr la segmentación y umbralización dadas sus características y aplicaciones se optó por utilizar el método de Otsu que es una técnica para segmentar de manera automática imágenes en función de la intensidad de los píxeles. La idea detrás de este método es que se pueda encontrar el umbral óptimo para la intensidad que divide una imagen en 2 clases, generalmente el objeto de interés y el fondo, básicamente busca encontrar el umbral que maximiza la separabilidad entre 2 clases.

El algoritmo comienza a través de un histograma que representa la diferente distribución en las intensidades de los píxeles de la imagen, el eje x del histograma representa las intensidades y el eje y presenta la cantidad con cada intensidad. Después se debe normalizar el histograma dividiendo cada valor de frecuencia por el número total de elementos para convertirlo hoy en una función de densidad de probabilidad de esta manera se pueden inicializar las variables para calcular la varianza, después se va iterando a través de todas las posibilidades para las intensidades del umbral calculando la varianza una vez más encontrando el umbral que minimiza la varianza entre los valores y después aplica ese umbral encontrado a la imagen original consiguiendo separar el objeto de interés del fondo lo que resulta en una respuesta binaria segmentada.





Horrible este método resulta fácil y eficiente para implementar teniendo en cuenta que su rendimiento puede verse afectado por factores como ruidos o la presencia de múltiples objetos.

Obtención de parámetros

Un concepto fundamental para la geometría y la teoría de convexidad es el método conocido como casco convexo éste se refiere a la envoltura convexa más pequeña que tiene un conjunto dado de puntos en un espacio, lo que quiere decir que es el conjunto de puntos es la forma más pequeña y convexa que puede contener todos esos puntos. El algoritmo ordena los puntos en un orden lexicográfico después se encuentra el punto más alto y el punto más bajo en términos de coordenadas, a continuación divide los puntos en 2 conjuntos los que están por encima de la línea formada por los: encontrados en el paso anterior, para cada conjunto es necesario ordenar los puntos en función de su ángulo polar en relación con el punto más bajo a través de la función de tangente inversa, desarrolla desde el punto más lejano y va recorriendo los puntos agregándolos a la envoltura convexa a medida que avanza.

Deep Learning

Para conseguir el aprendizaje profundo es necesario la utilización de redes neuronales convolucionales este tipo de redes están diseñadas para procesar datos como imágenes por lo que resulta útil para el caso de la visión por computadora.

Las redes neuronales convolucionales aplican una serie de filtros a la entrada para ir diseñando un contexto general de las características a analizar, dichos filtros detectan patrones espaciales en los datos, cómo texturas bordes y las formas en la imagen. Se puede separar una red neuronal a partir de la capa de convolución hoy que es donde se aplican diferentes filtros para reducir la dimensión de los datos y permitir que la red distinga las características de manera jerárquica. De la capa convolucional se aplica una función de activación para no introducir linealidades en el modelo, por lo que se consigue aprender patrones más complejos. Después se aplica en capas de agrupación y de normalización donde se reduce el tamaño de las características entrantes y se normalizan las entradas en cada capa. El último paso suele terminar con lo que se conoce como capas densas que éstas son quienes realizarán la clasificación en base a las características extraídas con el fin de que sean combinadas para poder formar predicciones.

Con el fin de mejorar los pesos de los filtros se aplican algoritmos de optimización durante el proceso de entrenamiento, como el descenso del gradiente estocástico que emplea una función de pérdida donde el objetivo es minimizar esta función ajustando los pesos y reducir la diferencia entre las predicciones del modelo y los valores reales.





Capítulo 3 Materiales y métodos

3.1 Diseño del algoritmo

Para el desarrollo del programa se empleó el uso del programa conocido como Pycharm con el fin de trabajar el área de la programación por medio del lenguaje de programación Python. El código hace uso extensivo de bibliotecas externas para su funcionamiento. OpenCV se emplea para el procesamiento de imágenes, Pillow (PIL) para la manipulación de imágenes, imutils para redimensionar imágenes, numpy para operaciones numéricas, matplotlib para crear gráficos y scipy para el procesamiento de datos.

Primero se creó la Interfaz Gráfica del Usuario (GUI), empleando principalmente la biblioteca Tkinter. A través de esta aplicación, los usuarios pueden llevar a cabo una variedad de tareas relacionadas con el procesamiento de imágenes y el análisis de datos. Para entender mejor su funcionamiento, se proporciona una explicación detallada de las diferentes partes principales del código:

En primera instancia, se crea la ventana principal denominada "ventana" utilizando Tkinter. Esta ventana se caracteriza por tener unas dimensiones mínimas de 700x500 píxeles, posee un título personalizado y se le asigna un icono particular mediante la línea de código `ventana.iconbitmap('codigo-binario.ico')`. Además, su fondo se configura en blanco, proporcionando así un lienzo limpio para la interfaz gráfica.

```
ventana = Tk()  
ventana.minsize(700, 500)  
ventana.title("Bovine Sperm Quality")  
ventana.resizable(1, 1)  
ventana.iconbitmap('codigo-binario.ico')  
ventana.config(bg='white')
```

El código implementa varias funciones que representan diferentes pantallas dentro de la aplicación, tales como home, formología, y motilidad. Estas funciones son cruciales para la organización de la interfaz de usuario y la gestión de la visibilidad de los elementos según la pantalla actual. Por ejemplo, la función `home()` se encarga de mostrar la pantalla de inicio, mientras que `formologia()` muestra la pantalla relacionada con el procesamiento de imágenes.





python

Copy code

```
def home():
```

```
    # Configuración de elementos en la pantalla de inicio
```

```
    ...
```

```
    return True
```

```
def formologia():
```

```
    # Configuración de elementos en la pantalla de procesamiento de  
    imágenes
```

```
    ...
```

```
    return True
```

En la pantalla inicial (home), se presentan etiquetas de texto y campos de entrada para recopilar información. Además, se incluyen botones para guardar y abrir archivos de texto, permitiendo a los usuarios almacenar y recuperar datos según sea necesario.

Figura 3.1 Pantalla principal del software

Bovine Sperm Quality

Inicio Formologia Metilidad Salir

Bienvenido

Centro de mejoramiento Genético

Ingresar datos de la muestra:

Identificación:

Línea Genética:

Fecha y Hora:

Técnico:

Guardar

Abrir



Fuente: Elaboración propia

El código incorpora la función `elegir_imagen` que posibilita a los usuarios seleccionar una imagen de sus sistemas de archivos. La imagen seleccionada se muestra en la interfaz de usuario, lo que facilita la visualización y el posterior procesamiento de imágenes.

La aplicación dispone de una pantalla dedicada al procesamiento de imágenes (formología). Aquí, los usuarios tienen la capacidad de llevar a cabo diversas operaciones de procesamiento de imágenes, como umbralización, detección de contornos, y cálculo de áreas y perímetros en la imagen seleccionada. Los resultados de estas operaciones, como el área, el perímetro y un porcentaje normalizado del objeto en la imagen, se presentan en la interfaz gráfica para su análisis.

La aplicación permite a los usuarios guardar la información ingresada en campos de texto en un archivo de texto mediante el botón "Guardar". Además, se proporciona la funcionalidad de abrir y mostrar el contenido de archivos guardados previamente.

Desarrollo de funciones.

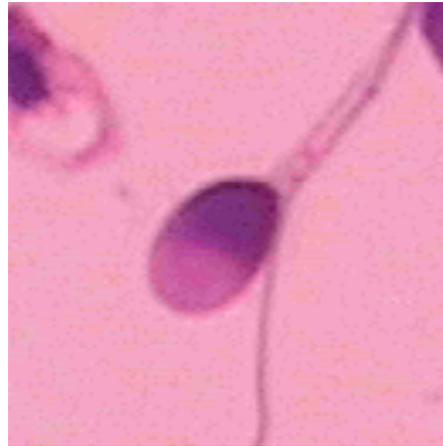
El procedimiento general parte de la obtención de las imágenes, después el primer paso es ubicar lo que se desea medir, a esto se le llama detección del objeto, pues en las imágenes pueden existir algunos otros organismos indeseables o cabezas de espermatozoides que se cruzan entre ellos. A partir de la detección se comienza el método Otsu donde se segmenta e umbraliza la información que previamente se detectó, quitando el fondo y dejando únicamente lo que se va a analizar, por último, se contempla la obtención de los valores, esto a través del `Convex Hull`.

Imagen

Las imágenes empleadas en este documento se obtuvieron a partir de un `image date set` que es un compendio de cabezas de espermatozoides de alguna subpoblación de una muestra seminal. Todas las imágenes se obtienen con un microscopio que cuenta con una cámara digital para la captura de estas, y cada una de las cabezas cuenta con características morfológicas diferentes y medibles.



Figura 3.2 Cabeza de espermatozoide normal



Fuente: *Bull Sperm Head Morphology dataset*

Las imágenes se pueden trabajar en función de dos variables, sus valores por fila por los valores de columna obteniendo un valor, que se puede acomodar en un arreglo o matriz. Y a partir de eso se puede modificar con respecto a los datos requeridos a través de distintos procesos.

En el código, se desarrolló algunas variables globales, como `img`, `path_image`, `segColor`, `p_label` y `a_label`, con el propósito de que estas variables fueran accesibles y modificables en cualquier parte de la función. Luego, la función verificó si la opción "Tratamiento" había sido seleccionada en la interfaz de usuario. Esto se hizo comprobando el valor de la variable `selected`, que se estableció en 1 cuando se seleccionó la opción "Tratamiento".

Segmentación y umbralización

Se empleó el método Otsu que consiste en extraer el fondo de la imagen, para este caso es necesario establecer un umbral T para los valores que van a ser considerados como fondo de los que no. El método Otsu es una forma en la se obtiene un parámetro adecuado para el umbral.

Se parte de una imagen gris con N píxeles y L posibles niveles diferentes y con la probabilidad de ocurrencia del nivel de gris i en la imagen y la frecuencia de repetición del nivel de gris.

Ecuación 3.1 Cálculo de probabilidad de cada nivel de intensidad

$$P_i = \frac{f_i}{N}$$

P_i = Probabilidad de cada nivel de intensidad

f_i = Frecuencia de niveles con cierta intensidad

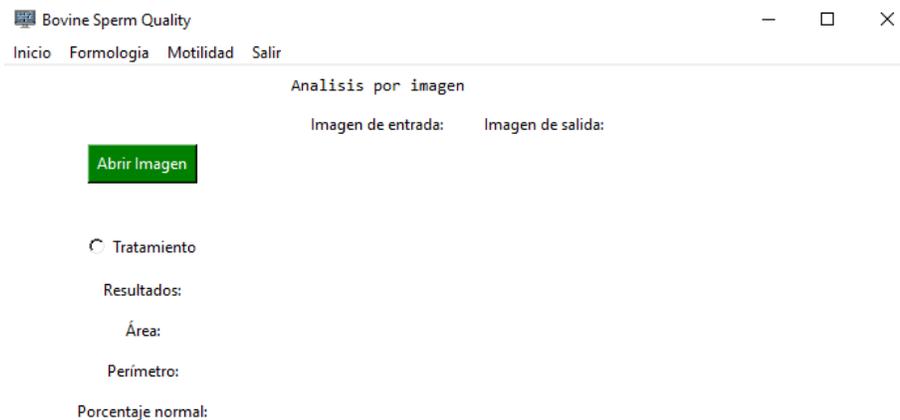
N = Número de total de píxeles en la imagen





Si la opción "Tratamiento" estaba seleccionada, se procede a realizar una serie de operaciones en la imagen que había sido cargada y que estaba almacenada en la variable `img`. En primer lugar, se cambió el espacio de color de la imagen a HSV empleando la función `cv2.cvtColor()`. Esto se hizo para facilitar ciertos análisis de color.

Figura 3.3 Pantalla del software

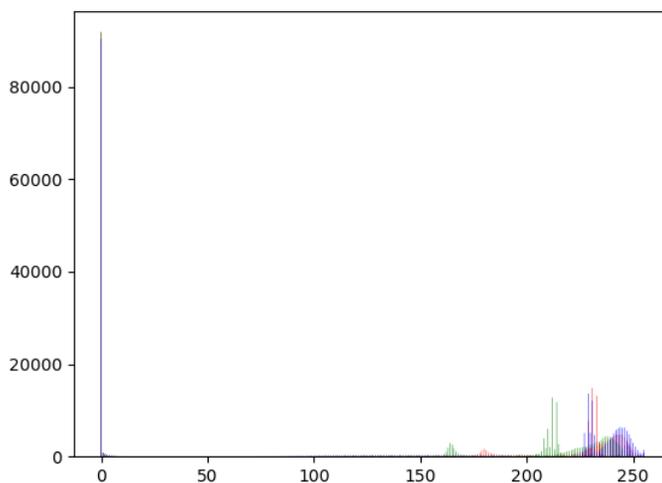


Fuente: Elaboración propia

La segmentación de una imagen es un proceso que separa el fondo del objeto que se desea analizar, eliminando así toda información no esencial y como se escribió previamente se emplea el método de Otsu. De la segmentación se obtiene el histograma, como se muestra en la figura de los valores que proporcionan los diferentes píxeles y así poder diferenciar el fondo del objeto. En este caso los objetos de interés que se buscan aislar son las cabezas de los espermatozoides. Este proceso es un complemento para que la umbralización tenga los valores adecuados.



Figura 3.4 Histograma de una cabeza de espermatozoide.



Fuente: Elaboración propia

Entonces, se convierte la imagen a escala de grises (I) utilizando nuevamente `cv2.cvtColor()`. La conversión a escala de grises simplificó el análisis de intensidades de píxeles. Luego, se calcula el umbral de la imagen en escala de grises utilizando el método de umbralización de Otsu (`cv2.THRESH_OTSU`) para separar el objeto de interés del fondo. El umbral calculado se almacenó en la variable `umbral`.

A través de la umbralización, se puede obtener una nueva imagen con dos colores a partir de una imagen en escala de grises, donde el objeto se separa del fondo. La umbralización se basa en el principio de que los píxeles de un objeto determinado tienen niveles de grises similares. Para ello se busca el intervalo de escala de grises al que pertenece el píxel del objeto.

Detección del objeto

La detección del espermatozoide sirve para caracterizar cada uno de ellos, y separarlos del resto o de cualquier contaminación que pueda existir en ella. En este procedimiento se detecta los cambios de intensidad en la imagen (los cambios de intensidad son los bordes de la imagen) entonces se toma un valor de intensidad para la primera fila y todas las columnas y la diferencia entre un valor de intensidad y su vecino. Entonces, si la intensidad no cambia la diferencia es cero, si la intensidad crece la diferencia será mayor de cero y por último si la intensidad decrece será menor que cero y a partir del vector de gradiente discreto se obtiene información de estos cambios, ya que sólo tomará valores enteros y así detectará los cambios en la intensidad de la imagen tanto horizontales como verticales



Ecuación 3.2 Cálculo del vector gradiente para cada píxel

$$\nabla f(m, n) = V(m, n + 1) - V(m, n), (m + 1, n) - V(m, n)$$

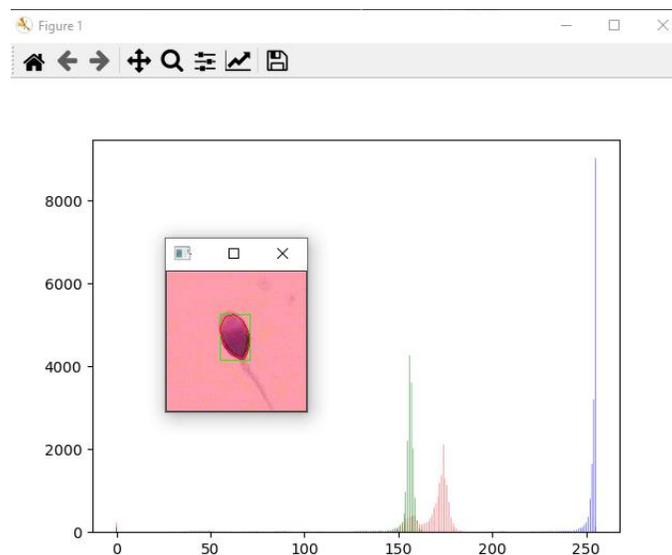
$V(m, n)$ = Intensidad por posición

$V(m + 1, n)$ = Valor de intensidad inmediatamente debajo

$V(m, n + 1)$ = Valor de intensidad inmediatamente a la derecha

Entonces, se creó la máscara binaria (mascara) que identifica las regiones del objeto de interés en la imagen. Los píxeles del objeto se establecieron en 255, y los demás píxeles se establecieron en 0. Esta máscara fue esencial para aislar el objeto del fondo en la imagen. Se utilizó `cv2.connectedComponentsWithStats()` para contar y analizar los componentes conectados en la máscara. `cantObj` almacenó el número de componentes y `label` almacenó las etiquetas de los componentes. `stats` contuvo estadísticas como la posición y el tamaño de cada componente, lo que fue útil para identificar el componente más grande, que generalmente fue el objeto de interés principal.

Figura 3.5 Histograma y detección de la cabeza espermática.



Fuente: Elaboración propia

A partir de las estadísticas, se creó una máscara final (también llamada mascara) basada en el componente más grande, asegurando que solo se analizara el objeto de interés principal en la imagen. Se empleó la función `ndimage.binary_fill_holes()` para rellenar los agujeros en la máscara binaria, garantizando que se consideraran todas las regiones del objeto de interés. Luego,



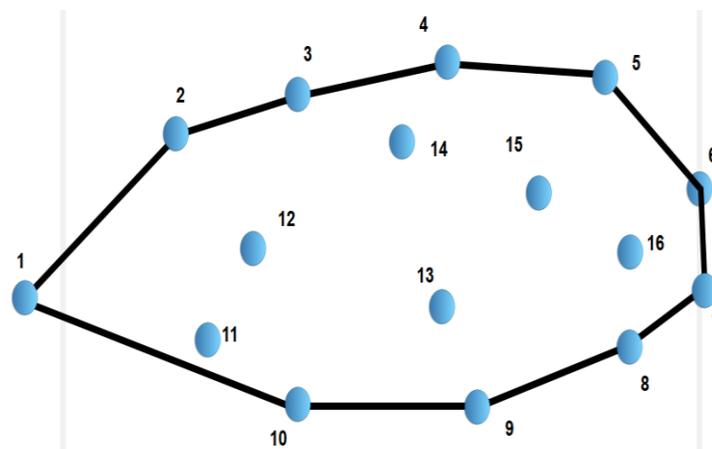
se obtuvo el contorno del objeto de interés en la máscara utilizando `cv2.findContours()`, y se guardó este contorno en la variable `cnt`.

Obtención de los distintos parámetros

La morfología del espermatozoide es una de las más significativas variables de la calidad espermática, entonces a partir del método `Convex Hull` se realiza la obtención de esas características, mencionando que para esta variable se tomó en cuenta las siguientes medidas que son la referencia de un espermatozoide: El área ($36.20\mu\text{m}^2$), el perímetro ($25.36\mu\text{m}$). Se considera viable con al menos $\geq 30\%$ de la población normal.

El `Convex Hull` funciona a través del análisis de los diferentes conjuntos de puntos que se forman a través de píxeles. Siendo `S` un conjunto de puntos este es el polígono convexo más pequeño que incluirá a los puntos de `S`.

Figura 3.6 Método `Convex Hull`



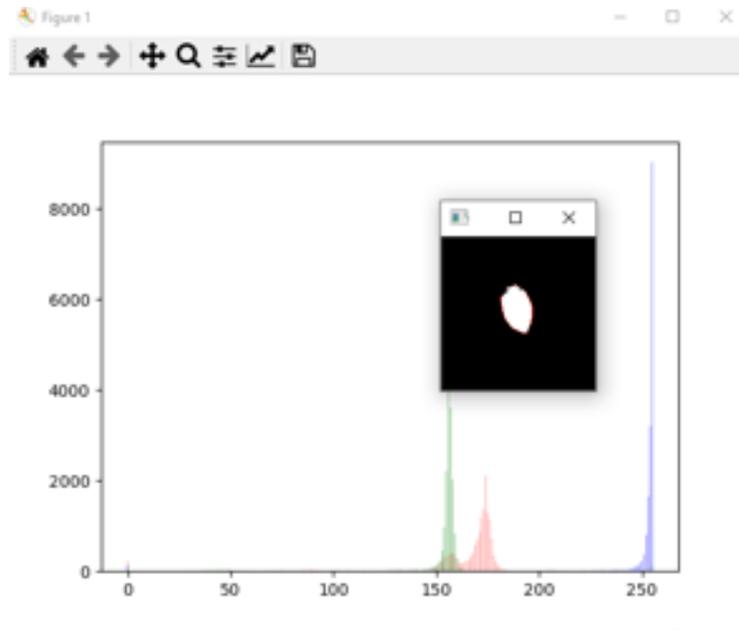
Fuente: Elaboración propia

Entonces se calculó el perímetro (`P`) y el área (`A`) del objeto de interés utilizando `cv2.arcLength()` y `cv2.contourArea()`, respectivamente. Estos valores fueron características importantes para el análisis del objeto. También se calculó `A1`, que fue la suma de píxeles del objeto en la máscara original, proporcionando una medida adicional del tamaño del objeto.

Se obtuvo un porcentaje normalizado (`Por`) empleando una fórmula específica que proporciona información sobre la forma del objeto.



Figura 3.7 Segmentación de una cabeza espermática.



Fuente: Elaboración propia

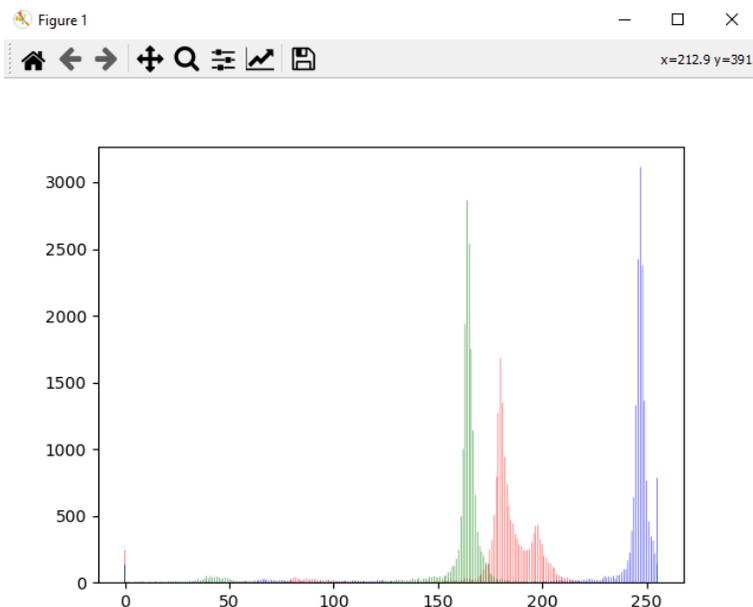
Se desarrolló el casco convexo del objeto de interés (`mascaraConvex`) utilizando `cv2.convexHull()`, lo que fue útil para describir la forma general del objeto. También creo un rectángulo rotado alrededor del objeto (`mascaraRect`) utilizando `cv2.minAreaRect()`, lo que proporcionó información sobre la orientación del objeto.

Además, se empleó un rectángulo recto alrededor del objeto utilizando `cv2.boundingRect()` que se colocó en la imagen original (`img`). Esto proporcionó información sobre la ubicación y el tamaño del objeto. Se dibujaron los contornos del casco convexo en la imagen original (`img`) para resaltar la forma del objeto.

Se utilizaron histogramas de los canales de color (rojo, verde y azul) de la imagen original utilizando `plt.hist()`, lo que proporcionó información sobre la distribución de colores en el objeto. Así se segmentó el objeto de interés en la imagen original y se almacenó en `segColor`. Esta fue una versión de la imagen original que resaltó el objeto de interés.



Figura 3.8 Histograma de los canales rojo, verde y azul.



Fuente: Elaboración propia

También se creó una versión en escala de grises del objeto de interés y se metió en la variable `segGrey`, lo que podría ser útil para análisis adicionales. También se redimensiono la imagen resultante y se convirtió a formato Tkinter para mostrarla en la interfaz de usuario (`lblOutputImage`).

Finalmente, se crearon etiquetas (`a_label` y `p_label`) que mostraron el área y el perímetro calculados en la interfaz de usuario, para que los usuarios pudieran ver los resultados de las operaciones de análisis en la imagen.



Figura 3.9 Pantalla del software en funcionamiento con cabeza normal.



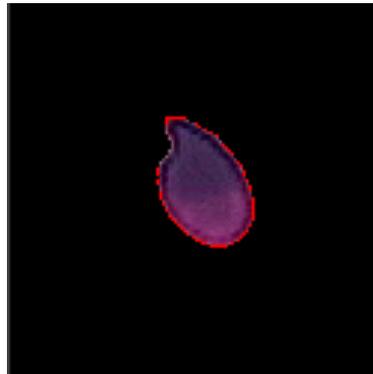
Fuente: Elaboración propia

Se dibujaron los contornos del casco convexo en la imagen original (`img`) para resaltar la forma del objeto. Y se generaron histogramas de los canales de color (rojo, verde y azul) de la imagen original utilizando `plt.hist()`, lo que proporcionó información sobre la distribución de colores en el objeto.

Se segmentó el objeto de interés en la imagen original y se archivó en `segColor`. Esta ha sido una versión de la imagen original que ha resaltado el objeto de interés. También se creó una versión en escala de grises del objeto de interés que se guardó en `segGrey`, lo que ha sido útil para análisis adicionales.



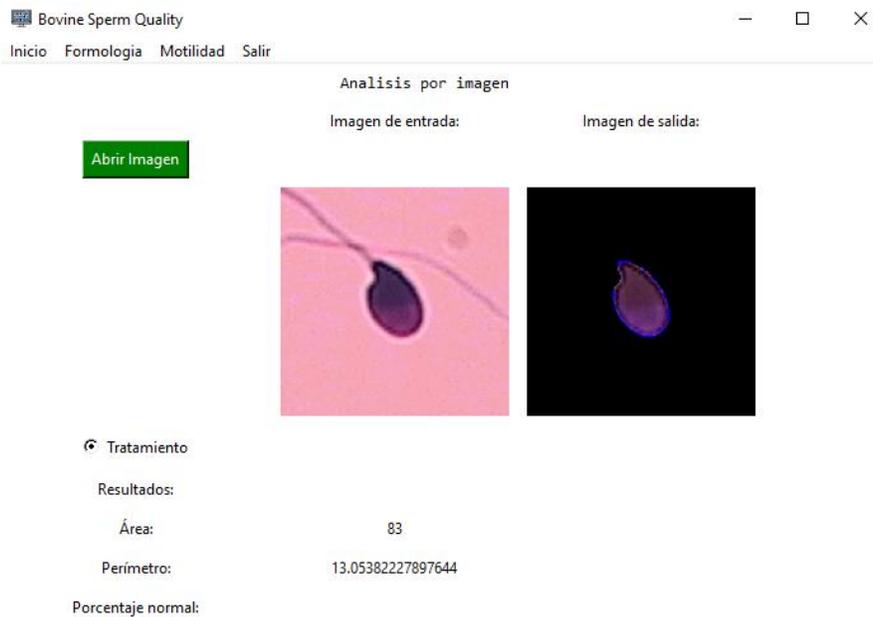
Figura 3.10 Cabeza anormal segmentada.



Fuente: Elaboración propia

A partir del *data set* “*Bull Sperm Head Morphology dataset*” (www.paperswithcode.com/dataset/bushem) que contiene 106 imágenes de la cabezas de espermatozoides se basan los siguientes resultados que se acompañan con el análisis estadístico.

Figura 3.11 Pantalla del software en funcionamiento con cabeza normal.



Fuente: Elaboración propia



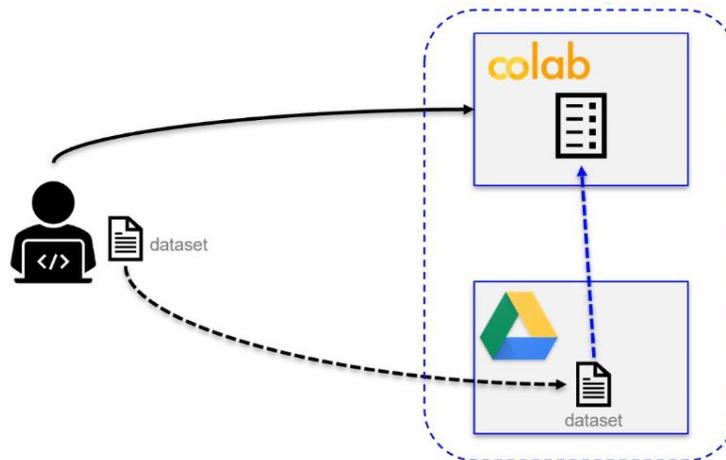
3.2 Deep Learning

Debido a la evolución constante de los métodos de extracción de semen, las técnicas de análisis celular y los procesos de crio preservación, resulta complicado asegurar una medición estándar que sirva como punto de referencia para futuras evaluaciones. Para promover una mejora continua del software, se ha implementado el aprendizaje supervisado. Así, a medida que se utiliza el programa, este se perfecciona en su capacidad para identificar espermatozoides con morfología saludable, procesando la información y estableciendo patrones de estas células.

3.2.1 Preparación del ambiente de trabajo

Se selecciona un ambiente en la nube con la finalidad de evitar cualquier tipo de contratiempo en la instalación y compatibilidad de este, siendo el único inconveniente la necesidad de una red de internet, el IDE a emplear será Google Colaboratory, para tomar las imágenes se enlazará con Google Drive donde quedará almacenado nuestro data set y al que se podrá recurrir en caso de ser necesitado.

Figura 3.12 Interacción de Google Colaboratory con el Drive de datos.



Fuente: Elaboración propia

3.2.2 Data set de imágenes

Como se mencionó previamente se cuenta a disposición el data set de imágenes de 54 elementos con cabezas normales y para este caso 73 elementos considerados como cabezas amorfas o dañadas de alguna forma. Para el entrenamiento se tomará el 90 % de cada set de imágenes y corresponderá un 10% para realizar pruebas.

- 50 imágenes de cabezas normales.
- 64 imágenes de cabezas anormales amorfas.

} Entrenamiento





- 5 imágenes de cabezas normales
- 8 imágenes de cabezas amorfas



Pruebas

Todo este conjunto se acomoda en un archivo Zip con el acomodo de carpetas previamente mencionado.

3.2.3 Construcción del modelo

El modelo de Deep Learning al fin de cuentas resulta en un modelo de Machine Learning por lo que se constituirá de dos fases primordialmente que es la fase de entrenamiento del algoritmo donde aprenderá a reconocer los patrones en una serie de iteraciones de las imágenes que previamente se segmentaron como espermatozoides sanos o anormales. Después se realizará pruebas con imágenes nuevas y analizaremos el pronóstico dado.

Dada la índole del proyecto se decidió emplear una red neuronal convolucional que se divide por dos componentes la parte de convolución y el clasificador.

3.2.4 Fase convolución

Para la convolución se emplea el filtro de Kernel, que se emplea para cada imagen de entrada y en cada posición realiza una operación matemática elemento por elemento entre el filtro y la parte de la imagen que se encuentra. Para pasar los números por la función de activación pues se busca no introducir linealidades en el modelo. Esto resalta las características específicas de la imagen que el filtro está diseñado a detectar.

3.2.4.1 Ecuación de la convolución

Si denotamos I como la imagen de entrada y F como el filtro, entonces el proceso de convolución (sin considerar la función de activación por ahora) puede expresarse como:

Ecuación 3.3 Cálculo de la operación de convolución en dos dimensiones.

$$(I * F)(x, y) = \sum_{i=-k}^k \sum_{j=-k}^k I(x + i, y + j) \cdot F(i, j)$$

I = Es la imagen de entrada a la que se le aplicará el filtro.

F = Es el filtro de kernel que se aplicará a la imagen.

(x, y) = Coordenadas de un píxel en la imagen.

$I(x + i, y + j)$ = Representa el valor de la posición del píxel.

$F(i, j)$ = El valor del filtro en la posición específica.

k = El tamaño del filtro.





Entonces permite a la red aprender a identificar características visuales importantes de manera eficiente y efectiva

3.2.5 Fase del clasificador

Después de pasar por la convolución los datos que eran imágenes se han convertido en datos de alto nivel, una representación matricial numérica, cuyas características el clasificador detectará para determinar a qué categoría pertenece la imagen.

Los componentes del clasificador son las capas Densas, la función de activación, la capa de salida y la función de activación que este caso corresponde una sigmoide ya que es de la índole binaria.

La salida de esta ecuación se calcula como la suma ponderada de sus entradas, seguida de la función de activación.

Ecuación 3.4 Cálculo de la unidad de la red neuronal.

$$y = f \left(\sum_{i=1}^n w_i x_i + b \right)$$

y = Salida del clasificador

f = Función de activación

$\sum_{i=1}^n w_i x_i$ = Es el producto entre los vectores de pesos y entradas

b = Es el término que ajusta la salida del modelo

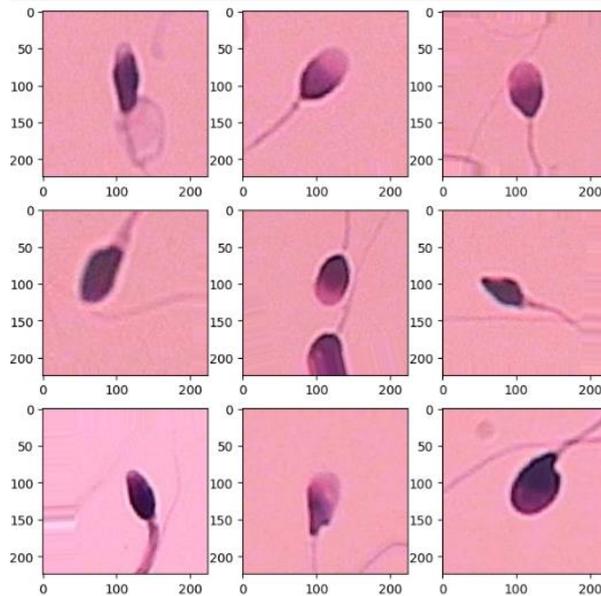
3.2.6 Preprocesamiento de datos

Con el fin de mejorar el rendimiento y velocidad del modelo, al conjunto de imágenes se procede a realizar una normalización de los valores de los píxeles de las imágenes, de un rango de 0-255 a un rango de 0-1 para reducir también la distorsión en los datos.

Dado que la cantidad de imágenes a emplear son muy pocas, también se realiza un proceso conocido como data augmentation donde se puede girar las imágenes en varias direcciones, mover la imagen y también hacerle zoom para de una imagen conseguir hasta 6.



Figura 3.13 Data augmentation

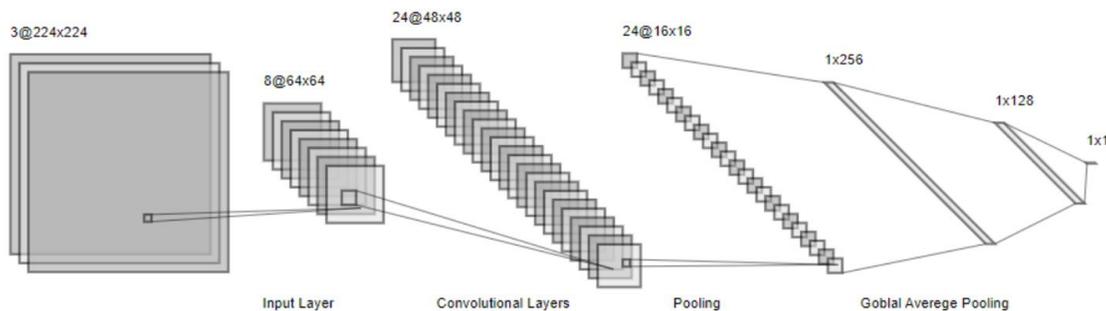


Fuente: Elaboración propia en Google Colaboratory

3.2.7 Arquitectura del modelo

Se utilizará una red del tipo DenseNet con la parte de convolución y su clasificador, dado que la red ya viene pre entrenada de sus capas de convolución y los pesos finales óptimos que se obtuvieron en el proceso de entrenamiento. Por lo que de esta arquitectura se removerá la clasificación y se propondrá un modelo de clasificador diferente que es el que se entrenaría.

Figura 3.14 Red Neuronal



Fuente: *Elaboración propia*





3.2.8 Clasificador

Entonces una vez que el modelo de convolución con los pesos pre entrenados se procede a la etapa de la clasificación. El modelo de clasificación necesita una capa de entrada donde van a estar entrando las matrices del modelo, obteniendo un promedio de éstas. Después se define la capa que va a contener los nodos que se asignan a cada valor de x_n

En estos modelos se debe cuida no sobre ajustarlo por lo que se añade una desactivación de un porcentaje de neuronas de manera aleatoria y se añade la última capa que contendría el último nodo que es quien daría la salida y_n

Figura 3.15 Arquitectura del clasificador.

```

Arquitectura final:
Model: "sequential"

```

Layer (type)	Output Shape	Param #
densenet201 (Functional)	(None, 7, 7, 1920)	18321984
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1920)	0
dense (Dense)	(None, 1000)	1921000
dropout (Dropout)	(None, 1000)	0
dense_1 (Dense)	(None, 1)	1001

```

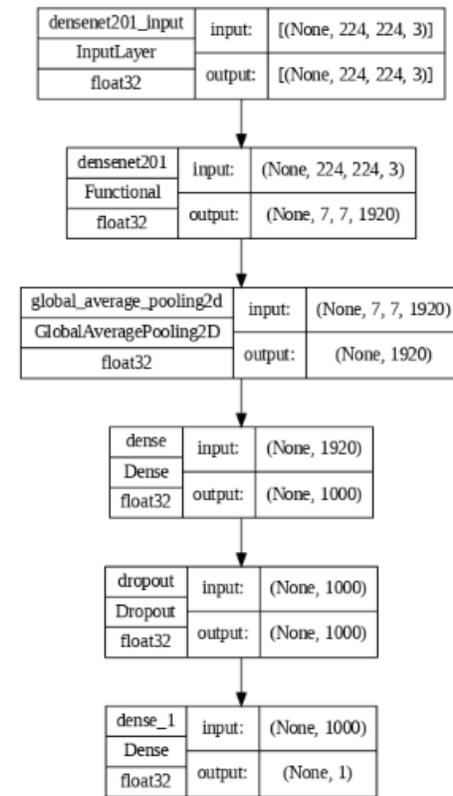
=====
Total params: 20243985 (77.22 MB)
Trainable params: 1922001 (7.33 MB)
Non-trainable params: 18321984 (69.89 MB)
=====

```

Fuente: Elaboración propia en Google Colaboratory



Figura 3.16 Arquitectura de la red neuronal convolucional



Fuente: Elaboración propia en Google Colaboratory

3.2.9 Entrenamiento del modelo

Consiste en exponer al algoritmo a muchas imágenes y se supervisa su aprendizaje, para que este asocie las imágenes con la categoría correspondiente, en este caso sería una cabeza normal y una anormal. Se realiza a través de una función de costo de entropía cruzada binaria para medir el error y se usa un método de optimización conocido como descenso de gradiente que ayuda al modelo a contrarrestar la pérdida ajustando los pesos de la red neuronal.

Ecuación 3.5 Función de Costo/Loss (Binary Cross-Entropy)

$$H_p(\mathbf{q}) = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

$H_p(\mathbf{q})$ = Entropía entre la distribución real y la distribución estimada

N = Es el número total de datos

y_i = El valor real de i-ésimo termino

$P(y_i)$ = Es la probabilidad de que el i-ésimo elemento pertenezca a una clase según el modelo





Ecuación 3.6 Ecuación de optimización

$$\theta_{nuevo} = \theta_{viejo} - \eta \cdot \nabla_{\theta} J(\theta; x^i, y^i)$$

θ_{nuevo} = Es el valor actualizado del parámetro.

θ_{viejo} = El valor anterior del parámetro

η = es la tasa de aprendizaje

$\nabla_{\theta} J(\theta; x^i, y^i)$ = Es el gradiente de costo con respecto al parámetro.

Por lo que se le proporcionan las imágenes etiquetadas en lotes en lugar de una por una, para que empiece a realizar el entrenamiento a través de varias épocas o iteraciones.

Figura 3.17 Entrenamiento de la red neuronal.

```

Epoch 1/20
3/3 [=====] - 55s 15s/step - loss: 0.7107 - accuracy: 0.5545 - val_loss: 0.5591 - val_accuracy: 0.5833
Epoch 2/20
3/3 [=====] - 36s 14s/step - loss: 0.5910 - accuracy: 0.7273 - val_loss: 0.5232 - val_accuracy: 0.7500
Epoch 3/20
3/3 [=====] - 42s 17s/step - loss: 0.4358 - accuracy: 0.8636 - val_loss: 0.4621 - val_accuracy: 0.8333
Epoch 4/20
3/3 [=====] - 35s 11s/step - loss: 0.3522 - accuracy: 0.9000 - val_loss: 0.4751 - val_accuracy: 0.6667
Epoch 5/20
3/3 [=====] - 36s 12s/step - loss: 0.2841 - accuracy: 0.9273 - val_loss: 0.4086 - val_accuracy: 0.7500
Epoch 6/20
3/3 [=====] - 36s 12s/step - loss: 0.2537 - accuracy: 0.8909 - val_loss: 0.3248 - val_accuracy: 0.8333
Epoch 7/20

```

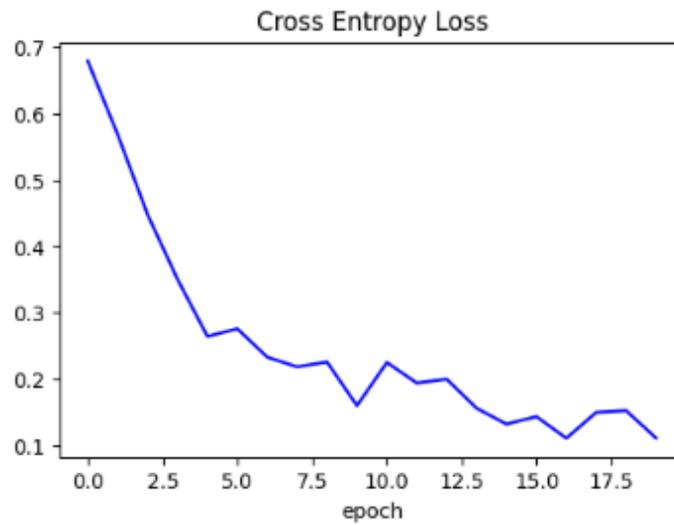
Fuente: Elaboración propia en Google Colaboratory

La finalidad es que se ajusten los pesos internos previamente mencionados para que pueda clasificar con mayor precisión basándose en el aprendizaje adquirido en el entrenamiento, reduciendo el error de la clasificación según mejore su capacidad para predecir la categoría correcta.

Para este punto ya se tiene entrenado el modelo, es decir ya tiene los pesos ideales para que pueda categorizar de manera precisa.

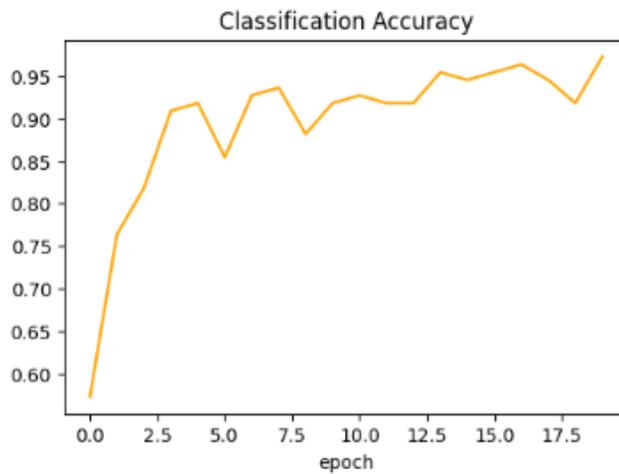


Figura 3.18 Evolución del error.



Fuente: Elaboración propia en Google Colaboratory

Figura 3.19 Evolución de la exactitud.



Fuente: Elaboración propia en Google Colaboratory

Se puede observar como el error va disminuyendo y la exactitud incrementa su valor conforme el modelo se fue entrenando.





Capítulo 4 Resultados y discusión

4.1 Resultados medidos

Se presenta a continuación un resume de algunas de las mediciones morfológicas de áreas y perímetros para la muestra empleada.

Tabla 4.1 Muestra de datos de área y perímetro.

Área μm^2	perímetro μm
32.444	28.014
31.552	20.013
29.352	30.312
31.294	22.872
30.433	29.366
32.416	25.95
31.795	20.556
29.757	21.393
30.190	18.872
30.811	21.194
31.397	23.582
30.583	17.964
31.225	24.563
32.285	17.835
30.056	30.243
32.335	18.282
30.957	27.708
32.071	24.404

Fuente: *Elaboración propia*





4.1.1 Planteamiento del análisis

Para el análisis de resultados se emplearon distintos métodos estadísticos, para dar una validación a los datos obtenidos. En las condiciones actuales donde tenemos una muestra, se buscará que a partir de los estándares morfológicos presentados anteriormente se pueda plantear una *hipótesis* sobre los valores que sirvan para dar una representación estadísticamente significativa de la muestra a la población de la cual no se cuenta acceso. Dando así una conclusión a la calidad morfológica de la misma y por ende una mejor fiabilidad para la toma de decisiones. Entonces se obtienen los siguientes resultados y se plantea la siguiente hipótesis para el conjunto de datos del área:

Tabla 4.2 Datos generales de la muestra.

	Media	Desv. Est.	Error estándar	IC de 95% para μ
A	48.80	27.235	2.33	(43.555,54.054)
P	24.259	9.541	0.927	(22.422,26.096)

Fuente: *Elaboración propia*

4.1.2 Prueba de hipótesis para el área y perímetro

En el contexto del problema resulta ser adecuado el uso de la **distribución t** de *student*, pues estamos trabajando con una muestra pequeña y sobre todo se desconoce la desviación estándar poblacional.

Para un valor de α de 0.05:

$$\text{Hipótesis nula } H_0 : \mu = 36.20$$

$$\text{Hipótesis alterna } H_1 : \mu \neq 36.20$$

Tabla 4.3 Valores t y p de la prueba de hipótesis para el área.

Valor t	Valor p
4.74	0.000

Fuente: *Elaboración propia*

$$\text{Hipótesis nula } H_0 : \mu = 25.36$$

$$\text{Hipótesis alterna } H_1 : \mu \neq 25.36$$





Tabla 4.4 Valores t y p de la prueba de hipótesis para el perímetro.

Valor t	Valor p
-1.19	0.237

Fuente: *Elaboración propia*

Con los datos anteriores se puede afirmar que el *Valor p* es menor al valor de α para el área se rechaza la *hipótesis nula* por que la muestra no entraría en promedio como una muestra *de área y en perímetro* como el *Valor p* es mayor a α por ende no hay suficiente evidencia para rechazar la *hipótesis nula*.

Con lo anterior se procede a revisar el comportamiento de la muestra tanto para área y perímetro en búsqueda de una caracterización de los valores obtenidos y así plantear de nuevo la hipótesis para aquellos valores que si entran dentro del valor establecido.

4.1.3 Intervalos de confianza

Para obtener el *IC* lo primero será demostrar que los datos obtenidos tienen un comportamiento del tipo normal para realizar las pruebas. Ya que según el teorema del límite central si $x_1, x_2, x_3, \dots, x_n$, es una muestra de n variables aleatorias e independientes con media \bar{x} , extraídas de una población μ con media y varianza σ^2 , entonces existe una variable aleatoria z que tiene a la normalidad cuando $n \rightarrow \infty$

Ecuación 4.1 Calculo del valor z

$$z = \frac{\bar{x} - \mu}{\sigma / \sqrt{n}}$$

\bar{x} = Es la media muestral

μ = Es la media poblacional

σ = Es la desviación estándar poblacional

n = Es el tamaño de la muestra

Como estas pruebas son aplicables cuando los datos son distribuidos normalmente se emplea el método de Anderson – Darling que a través de una prueba de bondad de ajuste evalúa si los datos muestreados vienen de una distribución específica.

Ecuación 4.2 Calculo del valor Anderson-Darling

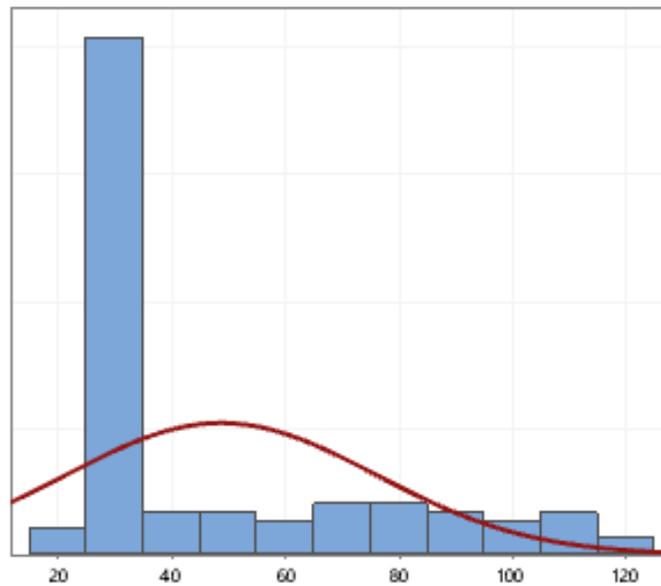
$$A^2 = -n - \frac{1}{N} \sum_{i=1}^n (2i - 1) [\ln(F(y_i)) + \ln(1 - F(y_{n+1-i}))]$$



n = Es el número de observaciones en el conjunto
 $F(y)$ = Es la función de distribución acumulativa
 $2i - 1$ = Es el factor de variación

Obteniendo lo siguiente:

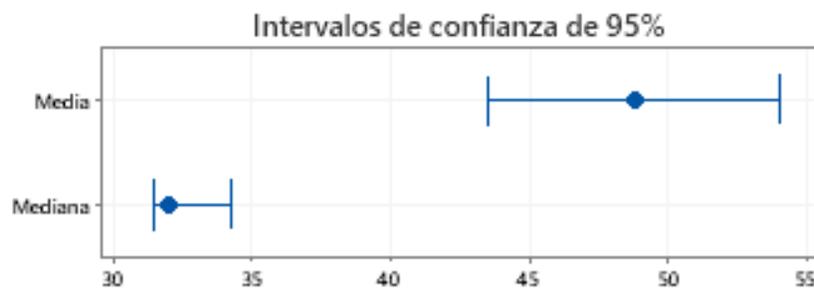
Figura 4.1 Histograma de los valores de área.



Fuente: *Elaboración propia en Mini Tab*

Con un valor de A^2 de 11.66 se aleja del comportamiento normal, aunque en el intervalo de confianza al 95% y en el histograma se observa un rango de valores que podrían ser los espermatozoides de la muestra que cumplen con las características morfológicas.

Figura 4.2 Intervalo de confianza para el área.





Fuente: *Elaboración propia en Mini Tab*

Por lo que se procede a segmentar la información de las áreas y se hace un análisis idéntico para el perímetro, dónde se da un comportamiento similar. Por lo que se define un rango aceptable a partir de los valores para el área (36.20m²) y el perímetro (25.36μm) de la ecuación:

Ecuación 4.3 Calculo del intervalo de confianza t de Student.

$$\bar{x} \pm t_{\alpha/2} \frac{s}{\sqrt{n}}$$

x = Es la media muestral

t_{α/2} = Es el valor crítico de ti

s = Es la desviación estándar muestral

n = Es el tamaño de la muestra

Obteniendo así los siguientes intervalos para el área y el perímetro con un 95% de confianza:

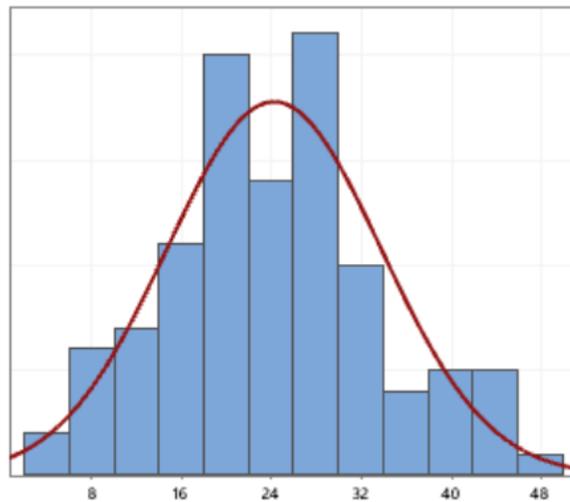
Tabla 4.5 Valores de área y perímetro con su respectivo intervalo.

$$A = 36.20 \pm 4.624$$

$$P = 25.36 \pm 1.838$$

Para la prueba de normalidad de los perímetros de obtiene lo siguiente:

Figura 4.3 Histograma de la prueba de normalidad.

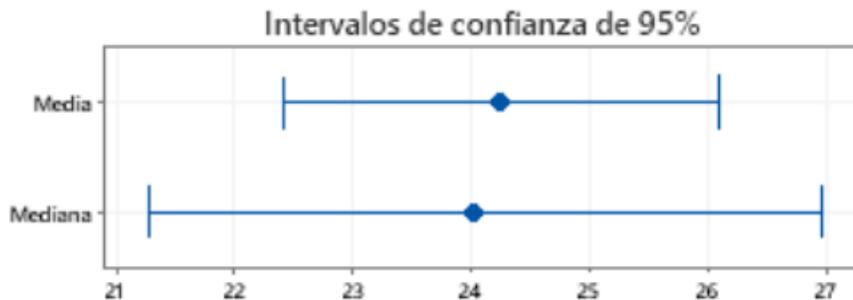


Fuente: *Elaboración propia en Mini Tab*



Con un valor de A^2 de 0.49 se acerca al comportamiento normal por lo que se procede a segmentar los valores que entren en el estándar morfológico.

Figura 4.4 Intervalo de confianza para el perímetro.



Fuente: *Elaboración propia en Mini Tab*

Después de la segmentación de la información por rangos, se procederá a analizar de manera independiente los valores que se acercan al intervalo de confianza. Aunque dadas las condiciones del problema dónde no conocemos la varianza poblacional es necesario el uso de la ya mencionada distribución en t.

4.1.4 Análisis del comportamiento ya segmentado

A partir del IC anterior se procede a segmentar los datos que entrar dentro del mismo tanto para perímetro y como para el área, repitiendo la hipótesis anterior para afirmar el comportamiento de este.

Para un valor de α de 0.05:

Hipótesis nula $H_0: \mu = 36.20$

Hipótesis alterna $H_1: \mu \neq 36.20$

Tabla 4.6 Valores t y p de la prueba de hipótesis para el área después de cambiar los datos.

Valor t	Valor p
1.26	0.212

Fuente: *Elaboración propia*





Hipótesis nula $H_0 : \mu = 25.36$

Hipótesis alterna $H_1 : \mu \neq 25.36$

Tabla 4.7 Valores t y p de la prueba de hipótesis para el perímetro después de cambiar los datos.

Valor t	Valor p
-1.64	0.106

Fuente: *Elaboración propia*

De los datos obtenidos en ambos casos con 95% de probabilidad no se puede rechazar la hipótesis nula pues el Valor p es mayor al valor de α en cuanto al área de la prueba estadística se demuestra que hay una diferencia entre las cabezas sanas y anormales, con respecto al perímetro no se encuentra una diferencia significativa entre las cabezas anormales y sanas.

4.2 Evaluación del modelo de aprendizaje profundo

Se debe entonces evaluar la capacidad de hacer predicciones correctas con datos nuevos, que es el conjunto de prueba.

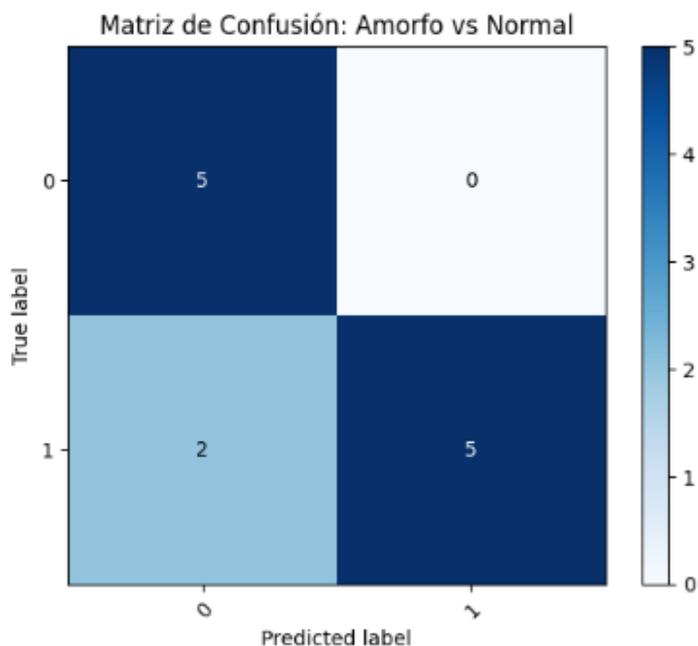
Por lo que se alimenta a la red neuronal con estas imágenes y se recogen las predicciones, estas se almacenan en un vector con 88 valores, cada uno corresponde a la predicción de la imagen, donde "1" indica un resultado normal y "0" indica un resultado anormal.

Para demostrar la efectividad del modelo se compara las predicciones con el resultado real conocido para determinar si el modelo ha clasificado de manera correcta la imagen y se procede a comparar las predicciones correctas con el número total de predicciones.

Se emplea una matriz de confusión para resumir los resultados de las predicciones del modelo, mostrando verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos.



Figura 4.5 Matriz de confusión



Fuente: Elaboración propia en Google Colaboratory

Por último, se muestra la exactitud en porcentaje total de predicciones correctas, la sensibilidad para identificar correctamente las cabezas normales y la especificidad para identificar correctamente los casos negativos.

Figura 4.6 Valores obtenidos del entramiento de la red neuronal.

Métricas:

Exactitud: 83.33%
Sensibilidad: 71.43%
Especificidad: 100.00%

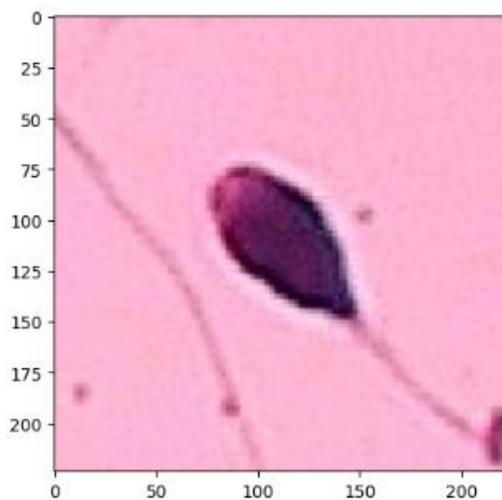
Fuente: Elaboración propia en Google Colaboratory

4.2.1 Realizando pruebas

Para las predicciones del modelo se procede a añadir imágenes completamente nuevas que no provengan del entrenamiento ni de las pruebas. Por lo que primeramente se procesa la imagen y se procede a realizar el pronóstico.



Figura 4.7 Cabeza anormal espermática.



Fuente: Elaboración propia en Google Colaboratory

4.2.2 Resultados

Para la clasificación se obtiene el resultado de la función sigmoide que proporciona valores entre 0 y 1, con un umbral de 0.5 todos los valores por encima de esa cantidad se consideraran como 1 y por debajo 0, para la imagen previa se obtuvieron los siguientes resultados:

Figura 4.8 Predicción de la red neuronal.

Pedicción: Amorfo
Prob: 99.30%

Fuente: Elaboración propia en Google Colaboratory





Capítulo 5 Conclusiones y recomendaciones

Conclusión

En el trabajo actual, se desarrolló con éxito una herramienta que utiliza técnicas de visión artificial y procesamiento de imágenes para analizar la calidad del semen en ganado utilizando el entorno de desarrollo pacharán y el lenguaje de programación python sí se diseñó e implemento un algoritmo funcional.

Integrando bibliotecas especializadas cómo open CV, Pillow, lmutils, numpy, matplotlib y scify para superar los desafíos previamente mencionados con el procesamiento y análisis de datos.

Esta herramienta cuenta con una interfaz gráfica del usuario que se desarrolló mediante la biblioteca de quintero, lo que facilita la interacción de los usuarios con dicha aplicación y la ejecución de las tareas complejas del procesamiento de imágenes y análisis de datos. Dicha interfaz permite la selección de imágenes, la visualización de resultados morfológicos y la arquitectura principal del programa proporciona una forma sólida para el estudio del semen bovino.

De datos que incluía las imágenes de cabezas de espermatozoides para comparar los resultados obtenidos con la categorización preestablecida de las cabezas normales y anormales, con esta validación se demostró que más del 50% de la muestra tiene una morfología acorde a los estándares y a su vez que era estadísticamente representativa de la población seminal.

Además, se ha demostrado que la herramienta también es capaz de realizar evaluaciones morfológicas con una precisión del 84% para la clasificación de las cabezas de los espermatozoides bovinos. El uso correcto de técnicas de Deep Learning y procesamiento digital mejora significativamente las limitaciones de los métodos tradicionales, lo que asegura avances notables en la eficiencia y confiabilidad de los programas de mejora genética bovina. Estas capacidades mejoran el uso continuo y su capacidad de adaptarse mediante las técnicas de aprendizaje profundo garantizan que la industria ganadera se mantenga a la vanguardia de innovaciones tecnológicas.

En conclusión, el presente no sólo ofrece una metodología estandarizada para la evaluación de la calidad espermática, sino que también abre oportunidades para futuras investigaciones, expandiendo la base de datos a otras especies, modificar el código de aprendizaje y añadir más imágenes a la base de datos.





Mejoras

Para seguir mejorando y validando la herramienta desarrollada, es recomendable:

1. Para conseguir un mejor modelo de aprendizaje profundo es importante incrementar la cantidad de imágenes disponibles en la base de datos.
2. También se pueden añadir otras técnicas de aprendizaje supervisado o de visión artificial y analizar los cambios.
3. Buscar la colaboración hoy de laboratorios para realizar pruebas de campo y verificar la efectividad del software en entornos reales y comparar los resultados con los métodos convencionales.

Contemplar la posibilidad de alargar el método de categorización para que sea para que sea posible emplearlo en otras especies de animales.

Referencias

Aguero, G. (2012). *Evaluación de las Características Seminales de Sementales Bovinos mediante el Analizador Seminal Computarizado (CASA)*. Caracas: UNIVERSIDAD CENTRAL DE VENEZUELA .

Secretaría de Información Agroalimentaria y Pesquera. (2018). La ganadería: símbolo de fortaleza del campo mexicano. Gobierno de México. Recuperado de <https://www.gob.mx/siap/articulos/la-ganaderia-simbolo-de-fortaleza-del-campo-mexicano>

Alejandro, Vicente, Pedro, Javier, Oscar, Sergio, & Carlos. (2009). *Sistemas de visión artificial*.

Calderón, C. D. (2013). *SOFTWARE C.A.S.A. PARA BOVINO*. Ibagué, Colombia : Univerdad de Ibagué.

Catena, M., & Cabodevila, J. (1999). *EVALUACIÓN DE SEMEN BOVINO CONGELADO* . Buenos Aires: UNICEN.

Flores, J. M. (2004). *Propuesta de un Sistema de Información para la toma de decisiones en la ganadería*. Cholula: Universidad de las Américas de Puebla.

Gomez, R. G. (2008). *Enciclopedia Bovina*. DF: Universidad Nacional Autónoma de México .

Llanos Gualtero, M. A., & Beltrán Diamante, J. F. (2016). *Diseño e implementación de una herramienta computacional para el análisis asistido de los parámetros de concentración y viabilidad espermática en bovino mediante técnicas de procesamiento de imágenes* . Cundinamarca: Universidad de Cundinamarca





Montaña, D. F. (2016). *Diseño e implementación de una herramienta computacional para el análisis asistido de motilidad espermática en bovinos mediante técnicas de visión artificial*. Cundinamarca: Universidad de cundinamarca.

Muiño, R., Fernández, M., Areán, H., Viana, J., López, M., Fernández, A., & Peña, A. (2005). *Nuevas tecnologías aplicadas al procesado y evaluación del semen bovino en centros de inseminación artificial*. Galicia: Universidad de Santiago de Compostela.

Alejandro, Vicente, Pedro, Javier, Oscar, Sergio, & Carlos. (2009). *Sistemas de visión artificial*.

Morota, G., Ventura, R., Silva, F., Koyama, M., & Fernando, S. (2018). *The livestock farming digital transformation: implementation of new and emerging technologies using artificial intelligence*. *Animal Health Research Reviews*. Cambridge University Press. Recuperado de <https://www.cambridge.org/core/journals/animal-health-research-reviews/article/abs/livestock-farming-digital-transformation-implementation-of-new-and-emerging-technologies-using-artificial-intelligence>

Farhad, M., Masud, M. M., Beg, A., Ahmad, A., & Ahmed, L. (2023). *A review of medical diagnostic video analysis using deep Learning techniques*. *Applied Sciences*, 13(11), 6582. <https://doi.org/10.3390/app13116582>

Huang, L., Subramoniam, S., Challis, E., & otros autores. (2023). *A systematic review on machine Learning and deep Learning techniques in the effective diagnosis of Alzheimer's disease*. *Brain Informatics*. <https://braininformatics.springeropen.com/articles/10.1186/s40708-023-00127-x>

Frontiers. (s.f.). *Advances in Cryopreservation of Bull Sperm*. Recuperado de <https://www.frontiersin.org/articles/10.3389/fvets.2020.00123/full>

MSD Veterinary Manual. (s.f.). *Breeding Programs in Cattle Reproduction - Management and Nutrition*. Recuperado de <https://www.msdsvetmanual.com/>

World Health Organization. (2021). *WHO laboratory manual for the examination and processing of human semen* (6th ed.). World Health Organization.

Zuidema, D., Kerns, K., & Sutovsky, P. (2021). *An Exploration of Current and Perspective Semen Analysis and Sperm Selection for Livestock Artificial Insemination*. *Animals*, 11(12), 3563.

Kambadur, R., Sharma, M., Smith, T. P., & Bass, J. J. (1997). *Mutations in myostatin (GDF8) in double-muscled Belgian Blue and Piedmontese cattle*. *Genome Research*, 7(9), 910-916. <https://doi.org/10.1101/gr.7.9.910>

Bishop, S. C., & Van Eenennaam, A. L. (2020). *Synergistic power of genomic selection, assisted reproductive technologies, and gene editing to drive genetic improvement of cattle*. *CABI Agriculture and Bioscience*. <https://doi.org/10.1186/s43170-021-00014-x>





Castro-Samano, Martha A., García-Mata, Roberto, Parra-Insunza, Filemón, Portillo-Vázquez, Marcos, Márquez-Sánchez, Isaac, & García-Sánchez, Roberto C.. (2019). *El mercado de la carne de bovino en México, considerados los factores externos. Agricultura, sociedad y desarrollo*, 16(1), 85-103. Recuperado en 28 de abril de 2024, de http://www.scielo.org.mx/scielo.php?script=sci_arttext&pid=S1870-54722019000100085&lng=es&tIng=es.

Silva Herzog, J. (s.f.). *Breve historia de la Revolución Mexicana*. Recuperado de <http://www.librosmaravillosos.com/brevehistoriadelaolucionmexicana/index.html>

Sosa Urrutia, M. E., Martínez Castañeda, F. E., Espinosa García, J. A., & Buendía Rodríguez, G. (2017). *Contribución del sector pecuario a la economía mexicana: Un análisis desde la Matriz Insumo Producto. Revista Mexicana de Ciencias Pecuarias*, 8(1), 31-41. <https://dx.doi.org/10.22319/rmcp.v8i1.4308>

Statista. (2024). *Contribución del sector agrícola al producto interno bruto (PIB) en México desde 2010 hasta 2022*. Recuperado de <https://www.statista.com/statistics/764175/agriculture-share-of-gdp-in-mexico/>

Secretaría de Agricultura, Ganadería, Desarrollo Rural, Pesca y Alimentación (SAGARPA). (2019). Programa sectorial de SAGARPA 2019.

Instituto Nacional de Estadística y Geografía (INEGI). (s.f.). *Ganadería*. Recuperado de <https://cuentame.inegi.org.mx/economia/primarias/gana/default.aspx?tema=e>

Meléndez, Pedro, & Bartolomé, Julián. (2017). *Avances sobre nutrición y fertilidad en ganado lechero: Revisión. Revista mexicana de ciencias pecuarias*, 8(4), 407-417. <https://doi.org/10.22319/rmcp.v8i4.4160>

Cognex Corporation. (2018). *Introducción a la visión artificial*. Cognex Corporation.

Gonzalez, Rafael; Woods, Richard. *Digital Image Processing*. 3rd ed., Pearson Education, Inc., 2008.

Otero, R. M. (2008). *EVALUACIÓN DE LA MOTILIDAD Y VIABILIDAD DEL SEMEN BOVINO MEDIANTE EL USO DE SISTEMAS CASA Y CITOMETRÍA DE FLUJO: IDENTIFICACIÓN DE SUBPOBLACIONES ESPERMÁTICAS*. Santiago de compostela: Unversidad de Santiago de Compostela.

Pascual-Gaspar, J., Olmedo, H., Exposito, A., Exposito, A., Bermudez, G., & Finat, J. (2008). *A simple and effective systema for computer-assisted semen analysis*. Valladolid: Univer okio School. (n.d.).





Deep Learning: A Practitioners Approach. Tokio School. Más información disponible en <https://www.tokioschool.com> sidad de Valladolid.

Anexos

1.1.1 Código Pycharm

```
from Tkinter import *
from Tkinter import ttk
from PIL import Image, ImageTk
from Tkinter import scrolledtext as st
from Tkinter import filedialog
import cv2
import imutils
import numpy as np
import matplotlib.pyplot as plt
from scipy import ndimage
# Definir ventana
ventana = Tk()
ventana.minsize(700, 500)
ventana.title("Bovine Sperm Quality")
ventana.resizable(1, 1)
ventana.iconbitmap('codigo-binario.ico')
ventana.config(bg='white')

#Pantallas
def home():
    inicio_label.config(fg='black', bg='white', padx=5, pady=5,
font=('Consolas', 10))
    inicio_label.grid(row=0,column=0)
```





```
inicio_label2.config(fg='CadetBlue', bg='white', font=('Consolas',  
10), padx=5,pady=5) # ,height=10,bg='blue',cursor='arrow')
```

```
inicio_label2.grid(row=1,column=0)
```

```
inicio_label3.config(fg='Black', bg='white', font=('Consolas', 10),  
padx=5,pady=5)
```

```
inicio_label3.grid(row=2,column=0)
```

```
#####Formulario
```

```
# Campos del formulario
```

```
add_frame.grid(row=4,column=0,padx=5,pady=5)
```

```
add_identi_label.grid(row=1, column=0, padx=5, pady=5, sticky=E)
```

```
add_identi_entry.grid(row=1, column=1, padx=5, pady=5, sticky=W)
```

```
add_linea_label.grid(row=2, column=0, padx=5, pady=5, sticky=E)
```

```
add_linea_entry.grid(row=2, column=1, padx=5, pady=5, sticky=W)
```

```
add_fecha_label.grid(row=3, column=0, padx=5, pady=5, sticky=E)
```

```
add_fecha_entry.grid(row=3, column=1, padx=5, pady=5, sticky=W)
```

```
add_tecnico_label.grid(row=4, column=0, padx=5, pady=5, sticky=E)
```

```
add_tecnico_entry.grid(row=4, column=1, padx=5, pady=5, sticky=W)
```

```
add_separator.grid(row=5)
```

```
boton.grid(row=6, column=1, sticky=E)
```

```
boton.config(padx=5,pady=5,bg="green", fg="white")
```

```
botonA.grid(row=7, column=1, sticky=E)
```

```
botonA.config(padx=5, pady=5, bg="green", fg="white")
```





```
abrirArchivo.grid(row=3, column=6, padx=10, pady=10, sticky=E)
```

```
#Ocultar pantallas
form_label.grid_remove()
mot_label.grid_remove()
lblInputImage.grid_remove()
lblOutputImage.grid_remove()
rad.grid_remove()
lblInfo1.grid_remove()
lblInfo3.grid_remove()
botonImagen.grid_remove()
resultado_label.grid_remove()
areal.grid_remove()
peril.grid_remove()
btnTratamiento.grid_remove()
lblVideo.grid_remove()
lblVideo1.grid_remove()
lblInfoVideoPath.grid_remove()
return True

def formologia():
    form_label.config(fg='black', bg='white', padx=5, pady=5,
font=('Consolas', 10))
    form_label.grid(row=0, column=1, sticky=E)
    add_separator2.grid(row=5)
    lblInfo1.grid(column=1, row=1, padx=5, pady=5)
    lblInfo3.grid(column=2, row=1, padx=5, pady=5)
```





```
botonImagen.config(padx=3, pady=3, bg="green", fg="white")
botonImagen.grid(row=3, column=0, sticky=N)

lblInputImage.grid(column=1, row=4, padx=5, pady=5)

lblOutputImage.grid(column=2, row=4, padx=5, pady=5)

rad.grid(column=0, row=8, padx=5, pady=5, sticky= S)
resultado_label.grid(column=0, row=9, padx=5, pady=5)
areal.grid(column=0, row=10, padx=5, pady=5)
peril.grid(column=0, row=11, padx=5, pady=5)
poril.grid(column=0, row=12, padx=5, pady=5)

# Ocultar pantallas
inicio_label.grid_remove()
inicio_label2.grid_remove()
inicio_label3.grid_remove()
mot_label.grid_remove()
add_frame.grid_remove()
btnTratamiento.grid_remove()
lblVideo.grid_remove()
lblVideo1.grid_remove()
lblInfoVideoPath.grid_remove()
return True

def motilidad():
    global p_label
    global a_label
    mot_label.config(fg='black', bg='white', padx=5, pady=5,
font=('Consolas', 10))
    mot_label.grid(column=0, row=0)
```





```
btnTratamiento.grid(column=0, row=2, padx=5, pady=5)
lblVideo.grid(column=0, row=3)
lblInfoVideoPath.grid(column=0, row=4)
lblVideo1.grid(column=0, row=5)
```

```
# Ocultar pantallas
```

```
inicio_label.grid_remove()
inicio_label2.grid_remove()
inicio_label3.grid_remove()
form_label.grid_remove()
add_frame.grid_remove()
lblInputImage.grid_remove()
lblOutputImage.grid_remove()
rad.grid_remove()
lblInfo1.grid_remove()
lblInfo3.grid_remove()
botonImagen.grid_remove()
resultado_label.grid_remove()
areal.grid_remove()
peril.grid_remove()
a_label.grid_remove()
p_label.grid_remove()
```

```
return True
```

```
def guardar():
```

```
    global identi_data
    global linea_data
    global fecha_data
    global tecnico_data
    f=open('Muestra.txt','w',encoding='UTF-8')
```





```
f.write(identi_data.get()+'\n'+linea_data.get()+'\n'+fecha_data.get()+'\n'+tecnico_data.get())

f.close()

def abrir():

    f = open('muestra.txt', 'r', encoding='UTF-8')
    contenido = f.read()
    f.close()
    abrirArchivo.delete('1.0', END)
    abrirArchivo.insert('1.0', contenido)

def elegir_imagen():

    global path_image

    # Especificar los tipos de archivos, para elegir solo a las imágenes
    path_image = filedialog.askopenfilename(filetypes = [

        ("image", ".jpeg"),
        ("image", ".png"),
        ("image", ".jpg"),
        ('image', '.BMP')])

    if len(path_image) > 0:

        global image

        # Leer la imagen de entrada y la redimensionamos
        image = cv2.imread(path_image)

        image= imutils.resize(image, height=380)

        # Para visualizar la imagen de entrada en la GUI
        imageToShow= imutils.resize(image, width=180)

        imageToShow = cv2.cvtColor(imageToShow, cv2.COLOR_BGR2RGB)

        im = Image.fromarray(imageToShow )

        img = ImageTk.PhotoImage(image=im)

        lblInputImage.configure(image=img)

        lblInputImage.image = img
```





```
def tratamientoImagen():
    global img
    global path_image
    global segColor
    global p_label
    global a_label
    if selected.get() == 1:
        img = cv2.imread(path_image)
        hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
        I = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        umbral, _ = cv2.threshold(I, 0, 255, cv2.THRESH_OTSU)

        mascara = np.uint8(I < umbral) * 255
        output = cv2.connectedComponentsWithStats(mascara, 4, cv2.CV_32S)
        cantObj = output[0]
        label = output[1]
        stats = output[2]
        mascara = (np.argmax(stats[:, 4][1:]) + 1 == label)
        mascara = ndimage.binary_fill_holes(mascara).astype(int)

        mascara1 = np.uint8(mascara * 255)

        _contours_, hierarchy = cv2.findContours(mascara1, cv2.RETR_TREE,
        cv2.CHAIN_APPROX_SIMPLE)
        cnt = _contours_[0]
        P = cv2.arcLength(cnt, True) - 114
        A = cv2.contourArea(cnt)
        A1 = np.sum(mascara) - 1035
```





```
#Por=(29/103)*100
# CONVEX HULL
hull = cv2.convexHull(cnt)
puntosConvex = hull[:, 0, :]
m, n = mascara1.shape
ar = np.zeros((m, n))
mascaraConvex = np.uint8(cv2.fillConvexPoly(ar, puntosConvex, 1))
# Bounding box rotado
rect = cv2.minAreaRect(cnt)
box = np.int0(cv2.boxPoints(rect))
m, n = mascara1.shape
ar = np.zeros((m, n))
mascaraRect = np.uint8(cv2.fillConvexPoly(ar, box, 1))

# Bounding box recto
x, y, w, h = cv2.boundingRect(cnt)
cv2.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 1)

    _contours_, hierarchy = cv2.findContours(mascaraConvex,
cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
    cv2.drawContours(img, _contours_, -1, (0, 0, 255), 1)
    # _contours_, hierarchy = cv2.findContours(mascaraRect,
cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)

# cv2.drawContours(img,_contours_,-1,(255,0,0),1)
dato = I.flatten()
rojo = img[:, :, 0].flatten()
verde = img[:, :, 1].flatten()
azul = img[:, :, 2].flatten()
plt.hist(rojo, bins=1000, histtype='stepfilled', color='red')
```





```
plt.hist(verde, bins=1000, histtype='stepfilled', color='green')
plt.hist(azul, bins=1000, histtype='stepfilled', color='blue')
# Segmentación del objeto de interés
segColor = np.zeros((m, n, 3)).astype('uint8')
segColor[:, :, 0] = np.uint8(img[:, :, 0] * mascara)
segColor[:, :, 1] = np.uint8(img[:, :, 1] * mascara)
segColor[:, :, 2] = np.uint8(img[:, :, 2] * mascara)
segGrey = np.zeros((m, n))
segGrey[:, :] = np.uint8(I * mascara)
imageToShow = imutils.resize(segColor, width=180)
im = Image.fromarray(imageToShow)
img = ImageTk.PhotoImage(image=im)
lblOutputImage.configure(image=img)
lblOutputImage.image = img

a_entrada= IntVar(value=A1)
a_label = Label(ventana,bg='white',textvariable=a_entrada)
a_label.grid(column=1, row=10, padx=5, pady=5)
p_entrada = DoubleVar(value=P)
p_label = Label(ventana, bg='white', textvariable=p_entrada)
p_label.grid(column=1, row=11, padx=5, pady=5)

#p_por = DoubleVar(value=Por)
#p_label = Label(ventana, bg='white', textvariable=p_por)
#p_label.grid(column=1, row=12, padx=5, pady=5)
```

```
def elegir_visualizar_video():
    global cap
    global video_path
```





```
if cap is not None:
    lblVideo.image = ""
    cap.release()
    cap = None
video_path = filedialog.askopenfilename(filetypes = [
    ("all video format", ".mp4"),
    ("all video format", ".avi")])
if len(video_path) > 0:
    lblInfoVideoPath.configure(text=video_path)
    cap = cv2.VideoCapture(video_path)
    tratamiento_video()
    #visualizar_video()
else:
    lblInfoVideoPath.configure(text="Aún no se ha seleccionado un
video")
def tratamiento_video():
    global frameI
    global video_path
    global cap
    global mostrar
    cap = cv2.VideoCapture(video_path)
    length = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
    video = []
    ret, frame1 = cap.read()
    for i in range(0, length):

        ret, frame = cap.read()
        if ret:
            video.append(frame)
    suma = np.zeros(frame1.shape)
```





```
for frameI in video:
    suma = suma + frameI
modeloFondo = np.uint8(suma / len(video))
for frameI in video:
    # frameI=video[50]
    diferencia = cv2.absdiff(frameI, modeloFondo).astype('uint8')

    diferencia = cv2.GaussianBlur(diferencia, (3, 3), 0)
    diferencia = cv2.cvtColor(diferencia, cv2.COLOR_BGR2GRAY)
    u, _ = cv2.threshold(diferencia, 0, 255, cv2.THRESH_OTSU)
    umbral = 14
    # umbral=u
    bina = 255 * np.uint8(diferencia > 1 * umbral)
    bina = 255 * np.uint8(ndimage.binary_fill_holes(bina))
    contours, hierarchy = cv2.findContours(bina, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)
    for i in contours:
        rect = cv2.boundingRect(i)
        x, y, w, h = rect
        if cv2.contourArea(i) > 1:
            cv2.rectangle(frameI, (x, y), (x + w, y + h), (0, 255,
0), 2)

    # break
    imageToShow = imutils.resize(frameI, width=640)
    im = Image.fromarray(imageToShow)
    img = ImageTk.PhotoImage(image=im)
    lblVideo.configure(image=img)
    lblVideo.image = img
    #mostrar= cv2.imshow('Conteo', frameI)
    if cv2.waitKey(1) & 0xFF == ord("q"):
```





```
break

def visualizar_video():
    global frameI
    global cap
    global video_path
    if cap is not None:
        ret, frameI = cap.read()
        if ret == True:
            frameI = imutils.resize(frameI, width=640)
            im = Image.fromarray(frameI)
            img = ImageTk.PhotoImage(image=im)
            lblVideo1.configure(image=img)
            lblVideo1.image = img
            lblVideo1.after(10, visualizar_video)
        else:
            lblInfoVideoPath.configure(text="Aún no se ha seleccionado un
video")
            lblVideo1.image = ""
            cap.release()

#Definir campos de pantallas
#####
#####Inicio
inicio_label = Label(ventana, text='Bienvenido ')
inicio_label2 = Label(ventana, text='Centro de mejoramiento Genético')
inicio_label3= Label(ventana,text='Ingresar datos de la muestra:')

#Variables
identi_data = StringVar()
linea_data = StringVar()
fecha_data= StringVar()
```





```
tecnico_data= StringVar()
cap=None
#Campos del formulario
add_frame = Frame(ventana,bg = 'white')

add_identi_label = Label(add_frame, text="Identificación: ", bg='white')
add_identi_entry = Entry(add_frame, textvariable=identi_data)
add_linea_label = Label(add_frame, text="Línea Genética: ", bg='white')
add_linea_entry = Entry(add_frame, textvariable=linea_data)
add_fecha_label = Label(add_frame, text="Fecha y Hora:", bg='white')
add_fecha_entry = Entry(add_frame,textvariable=fecha_data)
add_tecnico_label = Label(add_frame, text="Técnico: ", bg='white')
add_tecnico_entry = Entry(add_frame,textvariable=tecnico_data)
add_separator = Label(add_frame,bg='white')
boton = Button(add_frame, text="Guardar", bg='white',command=guardar)
botonA= Button(add_frame,text='Abrir',bg='white',command=abrir)
abrirArchivo=st.ScrolledText(add_frame,width=30,height=5
#####
#####Formologia
form_label = Label(ventana, text='Análisis por imagen')
add_separator2 = Label(add_frame,bg='white')
botonImagen= Button(ventana,text='Abrir
Imagen',bg='white',command=elegir_imagen)
# Label donde se presentará la imagen de entrada
lblInputImage = Label(ventana,bg='white')
# Label donde se presentará la imagen de salida
lblOutputImage = Label(ventana,bg='white')
# Creamos los radio buttons y la ubicación que estos ocuparán
selected = IntVar()
rad = Radiobutton(ventana, text='Tratamiento', width=25,value=1,
variable=selected,bg='white', command= tratamientoImagen)
```





```
# Label IMAGEN DE ENTRADA
lblInfo1 = Label(ventana, text="Imagen de entrada:",bg='white')
# Al momento que leemos la imagen de entrada, vaciamos
# la imagen de salida y se limpia la selección de los
# radiobutton
lblOutputImage.image = ""
selected.set(0)
# Label IMAGEN DE SALIDA
lblInfo3 = Label(ventana, text="Imagen de salida:",bg='white')
#Resultados de área y perimetro
resultado_label= Label(ventana,text='Resultados:',bg='white')
areal=Label(ventana,text='Área:',bg='white')
peril=Label(ventana,text='Perímetro:',bg='white')
poril=Label(ventana,text='Porcentaje normal:',bg='white')
#####
#####Motilidad
mot_label = Label(ventana, text='Análisis por video ')
btnTratamiento = Button(ventana, text="Elegir y visualizar video",
command=elegir_visualizar_video,bg='white')
lblVideo = Label(ventana, text="Video de entrada:",bg='white')
lblInfoVideoPath = Label(ventana, text="Aún no se ha seleccionado un
video",bg='white')
lblVideo1 = Label(ventana,bg='white')
#####
#cargar pantalla de inicio
home()
#Menu superior
menu_superior=Menu(ventana)
menu_superior.add_command(label='Inicio',command=home)
menu_superior.add_command(label='Formología',command=formologia)
menu_superior.add_command(label='Motilidad',command=motilidad)
```





```
menu_superior.add_command(label='Salir', command=ventana.quit)
#Cargar el menu
ventana.config(menu=menu_superior)
ventana.mainloop()
```

1.1.2 Código Deep Learning

RED NEURONAL CONVOLUCIONAL (MODELO DEEP LEARNING)

Análisis de cabezas de espermatozoides vía clasificación de imágenes aplicando Deep Learning

```
# TensorFlow
```

```
import tensorflow as tf
print('Tensorflow: ',tf.__version__)
```

```
import keras
print('Keras:', keras.__version__)
```

```
#from tensorflow import keras
#print('TF Keras:', tf.keras.__version__)
#print('TF Keras: ',keras.__version__)
```

```
# Python
import platform
print('Python:',platform.python_version())
```

```
from google.colab import drive
drive.mount('/content/drive',force_remount=True)
%%bash
cd "/content/drive/MyDrive/Deeplerning/esperma_proyecto/dataset"
```





```
# validar si dataset (zip) existe y descargar
FILE=Data_set_esperma.zip
if [ ! -f "$FILE" ]; then
    echo "Archivo $FILE no existe --> Descargando ..."

    gdown --id 1ohGQU6g79aWB6WEf-QQUfPGf1sE4favu
fi

# borrar carpetas train y test antes de descomprimir
rm -rf train test

# descomprimir dataset (zip) en carpetas train y test
unzip -o -qq Data_set_esperma.zip

import os
import shutil
import glob
import numpy as np
from random import randrange

from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt

# Folder del proyecto
project_folder =
"/content/drive/MyDrive/Deeplerning/esperma_proyecto/dataset"

files_train_amorfo =
glob.glob(project_folder+"/Data_set_esperma/train/Amorfo/*.jpg")

files_test_amorfo =
glob.glob(project_folder+"/Data_set_esperma/test/Amorfo/*.jpg")
```





```
files_train_normal =
glob.glob(project_folder+"/Data_set_esperma/train/Normal/*.jpg")

files_test_normal =
glob.glob(project_folder+"/Data_set_esperma/train/Normal/*.jpg")

# Obteniendo imágenes al azar de cada folder

file_train_amorfo =
files_train_amorfo[randrange(len(files_train_amorfo))]

image_train_amorfo = image.load_img(file_train_amorfo)

file_test_amorfo = files_test_amorfo[randrange(len(files_test_amorfo))]
image_test_amorfo = image.load_img(file_test_amorfo)

file_train_normal =
files_train_normal[randrange(len(files_train_normal))]

image_train_normal = image.load_img(file_train_normal)

file_test_normal = files_test_normal[randrange(len(files_test_normal))]
image_test_normal = image.load_img(file_test_normal)

# Visualizando 4 imágenes

fig = plt.gcf()
fig.set_size_inches(8, 8)

plt.subplot(2,2,1)
plt.imshow(image_train_amorfo)
plt.title(file_train_amorfo.split("/")[-1])

plt.subplot(2,2,2)
plt.imshow(image_test_amorfo)
plt.title(file_test_amorfo.split("/")[-1])
```





```
plt.subplot(2,2,3)
plt.imshow(image_train_normal)
plt.title(file_train_normal.split("/")[-1])

plt.subplot(2,2,4)
plt.imshow(image_test_normal)
plt.title(file_test_normal.split("/")[-1])

plt.show()
from tensorflow.keras.preprocessing.image import ImageDataGenerator

# Dimensión de las imgs a procesar
img_width = 224
img_height = 224
batch_size = 40

# Data Augmentation and Normalization
datagen_train = ImageDataGenerator(rescale=1.0/255.0,      # Normalizar
los valores al rango [0-1]

                                horizontal_flip=True,    # Giro
horizontal

                                rotation_range=15,        # Giro
aleatorio (clockwise) entre 0 y 15 grados

                                width_shift_range=0.15,   # Mover la
img horizontalmente 15%

                                height_shift_range=0.15, # Mover la
img verticalmente 15%

                                zoom_range=0.2)          # Zoom in /
Zoom out aleatorio de 20% => 80% - 120%

# Leer imagenes del folder dataset/train
```





```
training_set_imgs =
datagen_train.flow_from_directory(project_folder+"/Data_set_esperma/train
/",

                                                                    target_size =
                                                                    class_mode =
                                                                    classes =
                                                                    batch_size =

(img_width, img_height),

'binary',

['Normal', 'Amorfo'],

batch_size)
# Mostrando resultados
num_imgs_training = len(training_set_imgs.files)

print("\n")
print("Núm. de imgs ENTRENAMIENTO:", num_imgs_training)
print("Classes:", training_set_imgs.class_indices)
print("Núm. Classes [0]:", np.sum(training_set_imgs.labels == 0, axis=0))
print("Núm. Classes [1]:", np.sum(training_set_imgs.labels == 1, axis=0))
import copy

# Visualizando data augmentation para 1 imagen
fig = plt.gcf()
fig.set_size_inches(8, 8)

it = copy.copy(training_set_imgs)

for i in range(9):
    plt.subplot(3,3,1 + i)
    batch = it.next()[0]
    imagenp = (batch[0]*255).astype('uint8')
    plt.imshow(imagenp)
```





```
plt.show()

# Data Normalization
datagen_test = ImageDataGenerator(rescale=1.0/255.0)

# Leer imagenes del folder dataset/train
testing_set_imgs =
datagen_test.flow_from_directory(project_folder+"/Data_set_esperma/test/"
,
target_size =
(img_width, img_height),
class_mode =
'binary',
classes =
['Normal', 'Amorfo'],
batch_size =
shuffle=False)

# Mostrando resultados
num_imgs_testing = len(testing_set_imgs filenames)

print("\n")
print("Núm. de imgs TEST:", num_imgs_testing)
print("Classes:", testing_set_imgs.class_indices)
print("Núm. Classes [0]:", np.sum(testing_set_imgs.labels == 0, axis=0))
print("Núm. Classes [1]:", np.sum(testing_set_imgs.labels == 1, axis=0))
# DenseNet https://keras.io/api/applications/densenet/
# https://towardsdatascience.com/paper-review-densenet-densely-connected-convolutional-networks-acf9065dfefb
from tensorflow.keras.applications.densenet import DenseNet201
# ResNet https://keras.io/api/applications/resnet/
# from keras.applications import ResNet152V2
```





```
# Cargando modelo DenseNet

pretrained_model = DenseNet201(weights='imagenet', include_top=False,
input_shape=(img_width, img_height, 3))

pretrained_model.summary();
for layer in pretrained_model.layers:
    layer.trainable = False

pretrained_model.summary()
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import GlobalAveragePooling2D
from keras.layers import Flatten
from keras.layers import Dropout

# Definiendo una Red Neuronal vacía
model = Sequential()

# Agregando la parte convolucional (base)
model.add(pretrained_model)          # Modelo base

# Clasificador propio
model.add(GlobalAveragePooling2D())  # GlobalAveragePooling2D
https://adventuresinmachinelearning.com/global-average-pooling-convolutional-neural-networks/
model.add(Dense(1000, activation='relu'))
model.add(Dropout(rate=0.2))
model.add(Dense(1, activation='sigmoid'))

print("Arquitectura final:")
model.summary()
```





```
from tensorflow.keras.utils import plot_model

plot_model(model, show_dtype=True, show_shapes=True,
show_layer_names=True, rankdir='TB',to_file=project_folder
+'/output/model_plot.png',dpi=80)

from tensorflow.keras.optimizers import SGD
from tensorflow.keras.optimizers import Adam

opt = SGD(Learning_rate=0.008, momentum=0.9)
# opt = Adam(lr=0.008, beta_1=0.9, beta_2=0.999)

# COMPILANDO la Red Neuronal Convolutacional
model.compile(optimizer=opt, loss='binary_crossentropy',
metrics=['accuracy'])

%%time

epochs=20

# Entrenar
history = model.fit(training_set_imgs,
                    epochs=epochs,

steps_per_epoch=np.ceil(num_imgs_training/batch_size),
                    validation_data=testing_set_imgs,

validation_steps=np.ceil(num_imgs_testing/batch_size))

# GRAFICANDO resultados
import matplotlib.pyplot as plt
fig = plt.gcf()
fig.set_size_inches(12, 8)

# plot loss
```





```
plt.subplot(2,2,1)
plt.title('Cross Entropy Loss')
plt.xlabel("epoch")
plt.plot(history.history['loss'], color='blue', label='train')

# plot accuracy
plt.subplot(2,2,2)
plt.title('Classification Accuracy')
plt.xlabel("epoch")
plt.plot(history.history['accuracy'], color='orange', label='train')

# Accuracy del Test (Usar este valor como referencial solamente
https://github.com/keras-team/keras/issues/6499)
eval = model.evaluate(testing_set_imgs,
steps=np.ceil(num_imgs_testing/batch_size))

print('\nValidación en Test:')
print("Loss: {:.4}".format(eval[0]))
print("Accuracy: {:.2%}".format(eval[1]))

# Matriz de Confusión
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_auc_score

THRESHOLD=0.5
testing_set_imgs.reset() #batch_index=0

Y_pred = model.predict(testing_set_imgs,
steps=np.ceil(num_imgs_testing/batch_size))
y_pred = np.where(Y_pred >= THRESHOLD, 1, 0)

# print(Y_pred)
# print(y_pred)
```





```
cm = confusion_matrix(testing_set_imgs.classes, y_pred)
print("\nMatriz de Confusión:\n")
print(cm)

# print('\nClassification Report:')
# classes_names = ['Normal', 'Amorfo']
# print(classification_report(testing_set_imgs.classes, y_pred,
# target_names=classes_names))

from sklearn.utils.multiclass import unique_labels
import seaborn as sns

def plot_confusion_matrix(y_true, y_pred,
                          normalize=False,
                          title=None):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
    if not title:
        if normalize:
            title = 'Matriz de Confusión Normalizada'
        else:
            title = 'Matriz de Confusión sin Normalizar'

    # Compute confusion matrix
    cm = confusion_matrix(y_true, y_pred)
    # Only use the labels that appear in the data
    classes = unique_labels(y_true, y_pred)
    if normalize:
```





```
cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
print("Matriz de Confusión Normalizada")
else:
    print('Matriz de Confusión sin Normalizar')

# print(cm)

fig, ax = plt.subplots()
im = ax.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
ax.figure.colorbar(im, ax=ax)
ax.grid(linewidth=.0)
# We want to show all ticks...
ax.set(xticks=np.arange(cm.shape[1]),
        yticks=np.arange(cm.shape[0]),
        # ... and label them with the respective list entries
        xticklabels=classes, yticklabels=classes,
        title=title,
        ylabel='True label',
        xlabel='Predicted label')

# Rotate the tick labels and set their alignment.
plt.setp(ax.get_xticklabels(), rotation=45, ha="right",
rotation_mode="anchor")

# Loop over data dimensions and create text annotations.
fmt = '.2f' if normalize else 'd'
thresh = cm.max() / 2.
for i in range(cm.shape[0]):
    for j in range(cm.shape[1]):
        ax.text(j, i, format(cm[i, j], fmt),
```





```
ha="center", va="center",
color="white" if cm[i, j] > thresh else "black")

#fig.tight_layout()

plt.show()

return ax

plot_confusion_matrix(testing_set_imgs.classes, y_pred,
normalize=False,title="Matriz de Confusión: Amorfo vs Normal")

# Métricas (https://en.wikipedia.org/wiki/Confusion_matrix)
TN, FP, FN, TP = cm.ravel()

exactitud = (TP + TN)/(TP + TN + FN + FP)
sensitividad = TP / (TP + FN)
especificidad = TN / (TN + FP)
prevalencia =
np.sum(testing_set_imgs.classes)/len(testing_set_imgs.classes)

print("\nMétricas:\n")
print("Exactitud: {:.2%}".format(exactitud))
print("Sensitividad: {:.2%}".format(sensitividad))
print("Especificidad: {:.2%}".format(especificidad))
print("Prevalencia: {:.2%}".format(prevalencia))

# ROC Curve
AUC = roc_auc_score(testing_set_imgs.classes, y_pred)
print("AUCROC: {:.2%}".format(AUC))

# Cargando modelo desde el disco
from keras.models import load_model

# load model
```





```
loaded_model = load_model(project_folder +
"/output/esperma_model_full.h5")

# summarize model

loaded_model.summary()

from tensorflow.keras.preprocessing import image
import matplotlib.pyplot as plt

# Ruta de la imagen en tu drive

image_test =
'/content/drive/MyDrive/Deeplerning/esperma_proyecto/dataset/Data_set_esp
erma/test/Amorfo/image_045.jpg'

# Dimensiones deseadas para la imagen

img_width, img_height = 224, 224

# Cargar la imagen en formato PIL

image_to_predict = image.load_img(image_test, target_size=(img_width,
img_height))

print('Image type:', type(image_to_predict))
print('Image size:', image_to_predict.size)

# Mostrar la imagen

plt.imshow(image_to_predict)

plt.show()

# Procesando la imagen

test_image = image.img_to_array(image_to_predict) # Imagen como NumPy
array

test_image = np.expand_dims(test_image, axis = 0) # Reshaping de (w,h,c)
--> (1,w,h,c)

test_image = test_image.astype('float32') # Valores de [0-255] --
> [0.0-255.0]

test_image /= 255 # Valores en [0.0-1.0]
```





```
# Prediction
output = loaded_model.predict(test_image)[0][0]
print(output)
# Resultados
prediction = 1 if (output >= THRESHOLD) else 0

CLASSES = ['Normal', 'Amorfo']

ClassPred = CLASSES[prediction]
ClassProb = output

print("Pedicción:", ClassPred)
print("Prob: {:.2%}".format(ClassProb))
```

