



**Segmentación de partes de espermatozoides humanos con técnicas de  
Deep Learning usando Deep Cell**

**Nicolás Ignacio Román Marchant**

Profesor guía: Violeta Chang Camacho

Trabajo de titulación en conformidad a los  
requisitos para obtener el título de Ingeniero  
Civil en Informática

Santiago – Chile

2020

© Nicolás Ignacio Román Marchant , 2020



• Algunos derechos reservados. Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-Compartirlgual 3.0. Sus condiciones de uso pueden ser revisadas en: <http://creativecommons.org/licenses/by/3.0/cl/>.

## RESUMEN

Diversos estudios internacionales han mencionado que la calidad del semen es un elemento fundamental a la hora de explicar la infertilidad masculina. Frente a esto, existen diversos métodos para estudiar las muestras de espermatozoides humanos, y realizar segmentaciones de estos, que lleven a conclusiones claras y certeras. Entre esos métodos, un artículo reciente muestra una mejora frente a métodos anteriores, además, propone un *gold standard* para comparar resultados, expandiendo las soluciones a plataformas especializadas en esta labor, que utilizan técnicas de aprendizaje profundo. Con ayuda de la herramienta Deep Cell, se realizaron dos experimentos, involucrando dos sets de datos distintos y dos arquitecturas de segmentación, generando cuatro redes neuronales profundas que segmentan cabezas de espermatozoides humanos. Ninguno de estos modelos fue capaz de mejorar el coeficiente de Dice de la última investigación en el estado del arte.

**Palabras Claves:** Segmentación, Espermatozoides, Deep Learning, Red Neuronal Convolutacional, Deep Cell, Gold Standard, Transfer Learning, Retina Net, Panoptic

*True magic results from courage of the heart  
Boys and girls be ambitious  
One step can change the world*

## **AGRADECIMIENTOS**

Agradecer encarecidamente a mi familia, mis padres, mi hermano, primos, tíos, abuelos, los cuales me ayudaron a seguir mi sueño de estudiar una carrera universitaria, sin preocuparse del aspecto económico, y apoyándome en lo emocional, nunca dejándome afrontar solo los momentos más duros de estos largos 6 años. Muchas gracias por ser una familia comprensiva, sin conflictos importantes, que siempre están para uno.

Además, agradecer a mi profesora guía, Violeta Chang, por apoyarme durante el desarrollo de esta memoria con feedback, consejos, y por el inmenso esfuerzo realizado para encontrar un servidor para utilizar la herramienta, como también a Reza Akbari Movahed, por facilitarme los resultados de su investigación para comparar y ayudarme con algunas dudas.

Finalmente, darle las gracias a mi grupo de amigos de la universidad, con quienes pase los momentos más divertidos de este largo proceso, además de ser un apoyo fundamental en el aspecto académico. Sin ustedes esta memoria no sería posible. ¡Muchas gracias!

# TABLA DE CONTENIDO

<b>1 Introducción</b>	<b>1</b>
1.1 Antecedentes y motivación . . . . .	1
1.2 Descripción del problema científico . . . . .	3
1.2.1 Brecha del conocimiento . . . . .	3
1.2.2 Preguntas de Investigación . . . . .	4
1.3 Planteamiento de Hipótesis . . . . .	5
1.4 Herramienta a utilizar . . . . .	5
1.5 Objetivos y alcance del proyecto . . . . .	6
1.5.1 Objetivo general . . . . .	6
1.5.2 Objetivos específicos . . . . .	6
1.5.3 Alcances . . . . .	7
1.6 Metodología y herramientas . . . . .	8
1.6.1 Metodología . . . . .	8
1.6.2 Herramientas de desarrollo . . . . .	9
1.7 Estructura del Documento . . . . .	10
<b>2 Marco Teórico</b>	<b>12</b>
2.1 Marco conceptual biológico . . . . .	12
2.2 Marco conceptual . . . . .	15
2.2.1 Deep Learning . . . . .	15
2.2.2 Segmentación de imágenes médicas . . . . .	18
2.2.3 Aumento de datos . . . . .	20
2.2.4 Modelos de Redes Neuronales . . . . .	22
2.2.5 Transfer Learning . . . . .	26
2.2.6 Deep Cell . . . . .	27
2.2.7 Métricas de evaluación de resultados . . . . .	31
2.2.8 Validación Cruzada . . . . .	33
<b>3 Estado del arte</b>	<b>34</b>
3.1 Gold Standard . . . . .	34
3.2 Procesamiento de imágenes y segmentación automática de partes de espermatozoides . . . . .	37
<b>4 Diseño e Implementación</b>	<b>42</b>
4.1 Descripción de los datos . . . . .	42
4.2 Ambiente Experimental . . . . .	43
4.3 Procedimiento experimental: Metodología Basado en Objetivos . . . . .	46
4.3.1 Aumento de Datos . . . . .	46
4.3.2 Experimentos . . . . .	51
4.3.3 Entrenamiento de la red . . . . .	52
4.3.4 Análisis y comparación de resultados . . . . .	64
<b>5 Resultados experimentales</b>	<b>66</b>
5.1 Experimento 1: Gold Standard desde cero . . . . .	66
5.1.1 Protocolo de Experimentos . . . . .	66
5.1.2 Segmentación de Cabezas . . . . .	67
5.1.3 Retina Net . . . . .	67
5.1.4 Panoptic . . . . .	69
5.2 Experimento 2: Transfer Learning con Hela_S3 . . . . .	70
5.2.1 Protocolo de Experimentos . . . . .	70

5.2.2	Segmentación de Cabezas . . . . .	71
5.2.3	Retina Net . . . . .	71
5.2.4	Panoptic . . . . .	73
<b>6</b>	<b>Análisis de Resultados</b>	<b>75</b>
6.1	Experimento 1 . . . . .	76
6.2	Experimento 2 . . . . .	78
6.3	Análisis entre experimentos . . . . .	80
<b>7</b>	<b>Conclusiones</b>	<b>83</b>
7.1	Objetivos Específicos . . . . .	83
7.1.1	Incremento de datos . . . . .	83
7.1.2	Entrenamiento de modelos con Deep Cell 2.0 . . . . .	83
7.1.3	Análisis de técnicas y parámetros . . . . .	84
7.2	Experiencia . . . . .	84
7.3	Trabajo Futuro . . . . .	86
7.4	Otras reflexiones . . . . .	87
<b>Glosario</b>		<b>88</b>
<b>Referencias bibliográficas</b>		<b>89</b>
<b>Apéndices</b>		<b>97</b>
<b>A</b>	<b>Tablas</b>	<b>97</b>

# ÍNDICE DE TABLAS

Tabla 1.1	Selección de tareas por objetivo . . . . .	9
Tabla 1.2	Especificaciones de Hardware para baja carga computacional . . . . .	10
Tabla 1.3	Especificaciones de Hardware para alta carga computacional . . . . .	10
Tabla 4.1	Porcentajes y número de imágenes para entrenamiento, testeo y validación de la red neuronal. . . . .	48
Tabla 4.2	Transformaciones escogidas para el proceso de aumento de imágenes. . . . .	49
Tabla 4.3	Parámetros de entrenamiento para modelo RetinaNet con ambos dataset . .	56
Tabla 4.4	Parámetros utilizados en red neuronal Panoptic con computador personal. .	59
Tabla 4.5	Parámetros de entrenamiento para modelo Panoptic. . . . .	63
Tabla 5.1	Promedios y mediana del modelo Retina Net desde cero, en distintas instancias. . . . .	67
Tabla 5.2	Promedios y mediana del modelo Panoptic desde cero, en distintas instancias.	69
Tabla 5.3	Promedio y mediana del modelo Retina net con transfer learning . . . . .	71
Tabla 5.4	Promedios y mediana del modelo Panoptic con transfer learning. . . . .	73
Tabla A.1	Promedios y mediana del método Movahed desde cero, en distintas instancias.	97
Tabla A.2	Tabla comparativa de resultados de pruebas entre bibliotecas que aportan al aumento de datos. La prueba muestra cuantas imágenes por segundo pueden ser procesados en un solo core, con un dataset de 2000 imágenes provenientes de ImageNet con una CPU Intel Xeon Gold 6140. Puntaje mas alto es mejor. . . . .	98

# ÍNDICE DE FIGURAS

Figura 1.1 Logo de la biblioteca computacional deepcell-tf . . . . .	6
Figura 2.1 Estructura del Espermatozoide . . . . .	13
Figura 2.2 Proceso de la espermatogénesis . . . . .	14
Figura 2.3 Defectos asociados a la morfología de los espermatozoides, separados por la parte defectuosa. Ya sea la cabeza ( <b>A</b> ), la pieza media ( <b>B</b> ), la cola ( <b>C</b> ), o exceso de citoplasma ( <b>D</b> ) . . . . .	15
Figura 2.4 Composición básica de una red neuronal de múltiples capas . . . . .	16
Figura 2.5 Arquitectura de una red neuronal convolucional (CNN), incluyendo entrada, convolución y pooling. . . . .	17
Figura 2.6 Ilustración de métodos de convolución y pooling. . . . .	17
Figura 2.7 Radiografía de una rodilla antes y después de un proceso de segmentación	19
Figura 2.8 Ejemplo de transformación fotométrica, cambio de contraste . . . . .	21
Figura 2.9 Proceso de segmentación de células con Deep Learning. Resumen del algoritmo de segmentación de células 2D usando RetinaNet. Las imágenes iniciales entran al modelo Retina Net para crear mapas de probabilidades para las células. Luego, múltiples pasos de post procesamiento más tarde, se inicia la segmentación de células a partir de las muestras generadas por el modelo. . . . .	23
Figura 2.10 Estructura base de un modelo Retina Net. El esqueleto de la arquitectura se basa en una red de extractor de características en forma de pirámide (FPN) ( <b>a</b> ), que genera otra red neuronal FPN que posee las características a reconocer ( <b>b</b> ). Al mismo tiempo, esta última red se encuentra conectada a una red de clasificadores en base a las características ( <b>c</b> ) y otro modelo encargado de analizar los resultados del modelo ( <b>d</b> ). . . . .	24
Figura 2.11 Arquitectura tradicional de una red neuronal Panoptic. Mientras que una red convolucional completa se encarga de realizar la segmentación semántica, y una generadora de máscaras con base Retina Net realiza la segmentación de instancias La unión de ambas permite realizar una segmentación completa tanto del fondo, como de los objetos importantes. . . . .	25
Figura 2.12 Framework y paradigma generado por en modelos Panoptic con imagenes medicas. . . . .	25
Figura 2.13 Diferencias entre un aprendizaje inductivo y la transferencia inductiva. Mientras que el aprendizaje tradicional busca sus resultados (hipótesis) a través del entrenamiento y la función de error, una transferencia inductiva realiza este procedimiento más rápido, debido que la transferencia “guía” al modelo a buscar resultados por una ruta determinada. . . . .	27
Figura 2.14 Arquitectura de Deep Cell 2.0. Las flechas azules representan los movimientos de datos, mientras que las flechas negras representan los movimientos de recursos computacionales. . . . .	29
Figura 2.15 Resultados del artículo de Kromp, donde compara resultados del entrenamiento con dos dataset distintos. En negrita se puede ver el mejor resultado. . .	30
Figura 2.16 Ecuaciones de Jaccard y Dice. Donde R corresponde al conjunto de píxeles de la imagen de referencia, y S al conjunto de píxeles de la imagen obtenida por el algoritmo o herramienta informática. . . . .	32
Figura 2.17 Ecuación de distancia de Hausdorff. Se lee “para cada punto a de A, encuentre su distancia más pequeña a cualquier punto b de B; finalmente, mantenga la menor distancia encontrada entre todos los puntos a . . . . .	32
Figura 2.18 Ejemplo de <i>4-fold cross validation</i> , o validación cruzada de 4 intervalos. El segmento azul representa el conjunto de validación y el blanco el conjunto de entrenamiento por cada iteración de entrenamiento. . . . .	33

Figura 3.1 Diferencias entre muestra real, segmentada, y teórica. En la parte <b>a</b> ), es posible apreciar una imagen de muestra tomada desde un microscopio, luego, en la sección <b>b</b> ), se observa una segmentación manual realizada por un experto, trazando por encima de la imagen, siguiendo las normas establecidas en la teoría, graficadas en la parte <b>c</b> ). . . . .	34
Figura 3.2 Ejemplo de segmentación utilizada usando k-means. En color amarillo se ve la segmentación de cabezas y pieza media luego del cambio en el espacio de color y el uso del método de agrupamiento k-medias. . . . .	35
Figura 3.3 Procedimiento de algoritmo de segmentación de cabezas de espermatozoides. Proceso del algoritmo de segmentación de cabeza utilizando <i>k-means</i> y <i>clustering</i> . A partir de la cabeza obtenida en el primer algoritmo (paso a)), pasando por proceso de k-means y clustering (b), c) y d)) para finalmente retornar la cabeza de la mejor manera (imagen e)) sobretodo comparando con la imagen original (sección f) de la figura 2.14) . . . . .	36
Figura 3.4 Segmentación particular de cabezas de espermatozoides: acrosoma y núcleo. a) Resultado del algoritmo de de segmentación de cabezas con <i>clustering</i> y k-medias. b) Limpieza e binarización de la imagen y detección de ángulo y dirección de la cabeza del espermatozoide. c) Separación de acrosoma. d) Separación de núcleo) . . . . .	36
Figura 3.5 Segmentación de cabezas usando el método Shaker. (a) Imagen original. (b) Detección de bordes usando Canny. (c) Segmentación de cabezas y pieza media con GVF. . . . .	38
Figura 3.6 CNN utilizada por Nissen para evaluar la segmentación de espermatozoides humanos. . . . .	38
Figura 3.7 Representación gráfica del método Movahed. Se distinguen 4 secciones definidas en el método Movahed, el preprocesamiento de las imágenes tanto para las redes convolucionales como la red neuronal de clasificación (sección celeste). Con dos redes convolucionales, se realiza la segmentación de la cabeza, en relación a la parte axial (pieza media y cola) (sección gris). Con esta segmentación inicial, es posible realizar la segmentación de núcleo y acrosoma con técnicas de <i>clustering</i> y <i>k-means</i> (sección verde) y la segmentación de la cola y la pieza media con una red neuronal de clasificación pre entrenada. . . . .	40
Figura 4.1 Directorio de Trabajo. Existe una separación de 3 principales directorios: Scripts propios para el apoyo del trabajo de investigación (sección azul). Archivos locales, específicamente las imágenes utilizadas para entrenar el modelo y Directorios de Deep Cell montado a Docker, que permiten conectar el ordenador utilizado con el contenedor que posee la herramienta. . . . .	44
Figura 4.2 Estructura/Ambiente de Trabajo. Mientras que el ordenador personal (Tabla 1.2) se encarga de la creación de scripts propios de código, en el servidor remoto (Tabla 1.3) se almacena Docker y los drivers necesarios para la utilización de la GPU como principal motor para el entrenamiento. La biblioteca deepcell-tf se encuentra en la imagen de Docker <b>vanhalen:deepcell</b> , y posee una instancia de <i>Jupyter Notebook</i> para facilitar el trabajo . . . . .	45
Figura 4.3 Experimentos realizados para determinar el rendimiento de la segmentación automática de partes de espermatozoides. . . . .	52
Figura 4.4 Error durante entrenamiento en modelo Retina Net según tamaño de lote con el dataset Hela_S3. Estrella indica el valor con menor error. Valores que no se muestran poseen un error muy alto para comparar. . . . .	55
Figura 4.5 Error durante validación en modelo Retina Net según tamaño de lote con el dataset Hela_S3. Estrella indica el valor con menor error. Valores que no se muestran poseen un error muy alto para comparar. . . . .	56
Figura 4.6 Ejemplo de detección de células con modelo Panoptic, usando solo SCIAN-SpermSegGS, con experimento inicial en ordenador personal. . . . .	59

Figura 4.7 Error durante entrenamiento según tamaño de lote con el dataset Hela_S3. Estrella indica el valor escogido. . . . .	60
Figura 4.8 Error durante validación según tamaño de lote con el dataset Hela_S3. Estrella indica el valor escogido. . . . .	61
Figura 4.9 Error durante entrenamiento según tamaño de lote con el dataset SCIAN-SpermSegGS. Estrella indica el valor escogido. . . . .	62
Figura 4.10 Error durante validación según tamaño de lote con el dataset SCIAN-SpermSegGS. Estrella indica el valor escogido. . . . .	63
 Figura 5.1 Resultados del modelo con Retina Net desde cero, sin considerar falsos positivos. Línea central dentro de las cajas indica Promedio (Dice: 0,917458484, Jaccard: 0,852925447, Haussdorf: 0,742010342) . . . . .	68
Figura 5.2 Superposición entre muestras (Imagen RGB), máscaras de referencia (Color Blanco) y máscaras predecidas por modelo Retina Net desde cero (Color Azul) con las células con mayor y menor coeficiente de Dice. El color rojo representa partes que el modelo predijo, pero que no existen en la máscara de referencia. En la primera fila se pueden ver las imágenes originales, y en la segunda fila el proceso de superposición. Los valores de Dice para cada imagen son, de izquierda a derecha: <b>0.9752, 0.981884, 0.98217, 0.6418, 0.60493, 0.1904</b> . . . . .	68
Figura 5.3 Resultados del modelo con Panoptic desde cero, sin considerar células no detectadas. Línea central dentro de las cajas indica Promedio (Dice: 0,80036165, Jaccard: 0,675362142, Haussdorf: 0,742010342) . . . . .	69
Figura 5.4 Superposición entre muestras (Imagen RGB), máscaras de referencia (Color Blanco) y máscaras predecidas por modelo Panoptic desde cero (Color Azul) con las células con mayor y menor coeficiente de Dice. En la primera fila se pueden ver las imágenes originales, y en la segunda fila el proceso de superposición. Los valores de Dice para cada imagen son, de izquierda a derecha: <b>0.90339, 0.9115, 0.95551, 0.3197, 0.4009, 0.4733</b> . . . . .	70
Figura 5.5 Resultados del modelo Retina Net con transfer learning. Línea central dentro de las cajas indica Promedio (Dice: 0.8947, Jaccard: 0.82096, Haussdorf: 0,7414) . . . . .	72
Figura 5.6 Superposición entre muestras (Imagen RGB), máscaras de referencia (Color Blanco) y máscaras predecidas por modelo Retina Net con <i>transfer learning</i> (Color Azul) con las células con mayor y menor coeficiente de Dice. El color rojo representa partes que el modelo predijo, pero que no existen en la máscara de referencia. En la primera fila se pueden ver las imágenes originales, y en la segunda fila el proceso de superposición. Los valores de Dice para cada imagen son, de izquierda a derecha: <b>0.9669, 0.9608, 0.9581, 0.4829, 0.60579, 0.773.</b> . . . . .	72
Figura 5.7 Resultados del modelo Panoptic con transfer learning. Línea central dentro de las cajas indica Promedio (Dice: 0,72809, Jaccard: 0,57412, Haussdorf: 0,78644) . . . . .	73
Figura 5.8 Superposición entre muestras (Imagen RGB), máscaras de referencia (Color Blanco) y máscaras predecidas por modelo Panoptic con <i>transfer learning</i> (Color Azul) con las células con mayor y menor coeficiente de Dice. El color rojo representa partes que el modelo predijo, pero que no existen en la máscara de referencia. En la primera fila se pueden ver las imágenes originales, y en la segunda fila el proceso de superposición. Los valores de Dice para cada imagen son, de izquierda a derecha: <b>0.8216, 0.8383, 0.8347, 0.3814, 0.382, 0.486.</b> . . . . .	74
 Figura 6.1 Ejemplos de células no detectadas (falsos negativos) durante el experimento 1, con ambas arquitecturas. Estas imágenes en el dataset SCIAN-SpermSegGS corresponden a las imágenes 14 y 2, de izquierda a derecha) . . . . .	76
Figura 6.2 Función de densidad de probabilidad para coeficiente de Dice con un modelo Retina Net (en azul), Panoptic (línea roja) y el último método del estado del arte, el método Movahed (línea amarilla) en el experimento 1. . . . .	78

Figura 6.3 Función de densidad de probabilidad para coeficiente de Dice con un modelo Retina Net (en azul), Panoptic (línea roja) y el último método del estado del arte, el método Movahed (línea amarilla) en el experimento 2. . . . .	80
Figura 6.4 Comparación de resultados entre métodos Panoptic De izquierda a derecha: Imagen RGB original. Máscara de referencia, Máscara generada a partir del modelo Panoptic desde cero. Máscara generada a partir del modelo Panoptic con Transfer Learning. Imagen 19 del dataset SCIAN-SpermSegGS. . . . .	81
Figura 6.5 Comparación de resultados entre métodos Retina Net. De izquierda a derecha: Imagen RGB original. Máscara de referencia, Máscara generada a partir del modelo Retina Net desde cero. Máscara generada a partir del modelo Retina Net con <i>transfer learning</i> en primera iteración de validación cruzada .Imagen 15 del dataset SCIAN-SpermSegGS. . . . .	81
Figura 6.6 Comparación entre resultados de modelos generados por Deep Cell y el método Movahed (Movahed et al., 2019) que segmentan automáticamente cabezas de espermatozoides humanos. . . . .	82

# CAPÍTULO 1. INTRODUCCIÓN

## 1.1 ANTECEDENTES Y MOTIVACIÓN

La infertilidad humana consiste en la incapacidad del ser humano para reproducirse debido a problemas asociados a diversos factores. Estudios recientes indican que existe un 15% de hombres infértils en todo el mundo (aproximadamente, unos 574 millones de personas), y de este número, un 30% se debe a problemas relacionados con la calidad del semen (Agarwal, Mulgund, Hamada, & Chyatte, 2015). En otras palabras, un porcentaje considerable de la infertilidad masculina se debe específicamente a la calidad que se posee, principalmente en la forma (también llamado morfología), aspecto, y movimiento, y otras características de su unidad básica, los espermatozoides.

La infertilidad masculina no es solo la incapacidad de reproducirse, sino la responsable de una serie de problemas asociados. Tal como se ha investigado por varias décadas (MacLeod & Gold, 1951), la infertilidad masculina conlleva una serie de trastornos psicológicos a saber, estrés, depresión, problemas en el ambiente social y de pareja, bajo autoestima, entre otros (Domínguez, 2002). Estos autores definen el impacto en un hombre infértil como "una amenaza al proyecto de vida de las personas". Por otro lado, la mala calidad de semen también está vinculada a una serie de síntomas y enfermedades. A modo de ejemplo, estudios hablan sobre la relación entre el tiempo de vida y el semen de un paciente infértil (Jensen, Jacobsen, Christensen, Nielsen, & Bostofte, 2009), o los impactos generados en pacientes con cáncer testicular e, incluso, pacientes con celiaquía (Farthing, Edwards, Rees, & Dawson, 1982), lo cual puede afectar en la salud general de cualquier paciente.

Los métodos tradicionales para evaluar la calidad del semen de un paciente consisten principalmente en la toma de muestras, para un posterior análisis biológico y visual de los espermatozoides. En este último aspecto, y mediante una segmentación de las imágenes de la muestra, es posible encontrar algunas características que pueden determinar la causa de la infertilidad ó de algunas de las patologías mencionadas anteriormente, a saber, anomalías en la forma de la cabeza del espermatozoide, longitud de la cola, entre otras. Todos estos elementos necesitan tiempo y personal capacitado para realizar la segmentación, como también para contrastar con otras opiniones de otros especialistas.

Los avances computacionales permiten generar nuevas soluciones frente a las adversidades antes mencionadas, sobretodo en materias relacionadas a la inteligencia artificial y aplicaciones en imágenes, las cuales, en la actualidad, funcionan como un asistente al especialista, tanto en el área científica (investigando causas, analizando resultados) como en

el área médica (a la hora de realizar diagnósticos y tratamientos). Aportar con herramientas computacionales que permitan el desarrollo de la medicina y la ciencia es esencial para solucionar o explicar los distintos inconvenientes relacionados. En relación a este tema, uno de los principales avances es *Deep Learning* o aprendizaje profundo, que en conjunto con la visión en computador, permite encontrar características o patrones que pueden entregar pistas o conclusiones frente a una imagen. De ahí que esta disciplina sea tan utilizada actualmente en medicina, en temas como revisión de radiografías, muestras de sangre y células en particular, como lo son los espermatozoides.

Nuevas plataformas computacionales se han creado en los últimos años, con el fin de suplir las distintas necesidades de los expertos, sin embargo, no todas las investigaciones, métodos o estándares han sido implementadas en estas, tal como es el caso de la segmentación de partes de espermatozoides utilizando técnicas de aprendizaje profundo, y no existe una comparación entre métodos tradicionales y los últimos avances en la computación en el área especializada en ello.

Este tema de investigación busca ser un aporte generado en el área médica, ya que permite comprobar si el uso de determinados métodos o plataformas son las adecuadas para el procesamiento de imágenes en el análisis y segmentación de espermatozoides, lo que puede disminuir recursos en el diagnóstico y posterior tratamiento de pacientes con infertilidad u otra patología asociada. Además, la importancia de la fertilidad en el mundo entero, ya que es un elemento fundamental en algunas culturas (llegando a ser trascendental en el aspecto social) (Cui, 2010) y en la salud de las personas que sufren esta patología (Domar, Broome, Zuttermeister, Seibel, & Friedman, 1993). Si el tema de investigación comprueba la hipótesis planteada, significa un avance en la investigación de patologías relacionadas, lo cual puede mejorar la calidad de vida de estas personas.

Al mismo tiempo, es importante considerar que la investigación no es suficiente para ser utilizada como diagnóstico clínico, a pesar de llevar muchos años desarrollando métodos, estrategias y herramientas. Esto se debe a los bajos niveles de precisión alcanzados durante la investigación, sobretodo comparando otras aplicaciones similares en el área médica, a saber, mapas de segmentación de sustancia gris, blanca o líquido cefalorraquídeo, o técnicas de segmentación, unidas a la reconstrucción de segmentos del sistema nervioso central (Anwar et al., 2018). Debido a esto, la tendencia en la investigación corresponde mejorar partes esenciales a la hora de realizar un diagnóstico; en este caso, la segmentación de partes automatizadas en espermatozoides humanos. Se suma a esto la conclusión de algunos investigadores en el área, que indican que la utilización de herramientas basadas en Deep Learning para realizar tareas de detección, segmentación, análisis morfológico y clasificación no poseen, en conjunto, resultados trascendentales en la investigación, por lo que una corriente entre los investigadores es la

utilización de estas herramientas para cada área involucrada (Movahed et al., 2019). Este trabajo de investigación busca aportar en esta tendencia, particularmente en el trabajo de segmentar partes de espermatozoides humanos, particularmente en cabezas.

El principal impacto, es lograr un avance considerable en el área médica y científica, utilizando una técnica computacional novedosa, que ha sido utilizada por otros especialistas en otros rubros, con el fin de reducir tiempos y recursos, como también marcar un precedente en la segmentación de espermatozoides humanos. Al mismo tiempo, aportar en la investigación de segmentación automática de espermatozoides, con una nueva instancia para comparar, y analizar el impacto de una red de aprendizaje profundo como esta.

## 1.2 DESCRIPCIÓN DEL PROBLEMA CIENTÍFICO

Tal como se mencionó en la sección anterior, uno de los principales elementos a considerar en el análisis de una muestra de espermatozoides son los especialistas, quienes suelen ser limitados en número debido a su especialidad. Acompañado a esto, un problema importante es la necesidad de “una segunda opinión” de parte de otro especialista, con el fin de discernir sobre alguna muestra en particular, u otros problemas. Frente a esto, el uso de tecnologías informáticas para “asistir” al especialista permite solucionar este problema, y el fuerte impulso que se le ha dado a la inteligencia artificial, principalmente a la segmentación con redes de aprendizaje profundo en la medicina durante los últimos años (Hesamian et al., 2019), forma parte de la pregunta de cómo estas tecnologías pueden aportar a la investigación ya instalada, en la segmentación automática de espermatozoides.

Sin embargo, la utilización de redes neuronales de aprendizaje profundo conlleva una serie de problemas asociados, como la necesidad de poseer una gran cantidad de datos, en este caso, muestras de semen con espermatozoides válidos para comparar, además de una segmentación de referencia (o conocida en el área de la inteligencia artificial como *labeled image*) que permite al modelo entrenar y aprender en base a los datos entregados (Taulli, 2019), como también la definición de un modelo eficaz, con parámetros definidos, para una situación particular.

### 1.2.1 Brecha del conocimiento

Siguiendo con la idea anterior, podemos afirmar que existen dos brechas en el conocimiento, siendo la primera los pocos datos que posee el dataset de imágenes utilizado

para la segmentación automática de espermatozoides. Actualmente, el *Gold Standard* definido por (Chang et al., 2014) posee 20 imágenes con, aproximadamente, 250 espermatozoides para evaluar, lo cual es insuficiente para entrenar cualquier tipo de red neuronal. Según (Nissen et al., 2017), un número de células suficiente para entrenar una red convolucional tradicional, usando un número mínimo de operaciones, es de 38.708. El incremento de las imágenes debe ser acorde a las especificaciones del dataset, como también, evitar errores que puedan afectar a una posterior comparación con estudios realizados anteriormente.

La segunda brecha corresponde a la inexistencia de un método de segmentación automatizada de espermatozoides humanos compuesto en su totalidad por un red neuronal de aprendizaje profundo. Las últimas investigaciones utilizan diversas técnicas de inteligencia artificial, incluyendo redes convoluciones de diversos tipos interconectadas para evaluar la efectividad de la segmentación automática (Movahed et al., 2019), pero en ninguna investigación publicada sobre el tema se realiza una completa implementación usando redes convolucionales profundas. El uso del *deep learning* se ha masificado en los últimos años, mostrando buenos resultados en investigaciones que integran esa rama de la inteligencia artificial, por ejemplo, segmentación de palabras en chino (Li et al., 2019), modificación de vídeos de manera realista como lo son los “*deep fakes*”, y áreas como la seguridad, en relación con la seguridad de cuentas de correos y/o cuentas bancarias.

## 1.2.2 Preguntas de Investigación

Algunas preguntas que surgen a partir de este trabajo de investigación van enfocadas en si el uso de una red neuronal convolucional profunda otorga resultados positivos en la investigación, y, al mismo tiempo, si este tipo de redes neuronales, específicamente utilizando la herramienta Deep Cell 2.0 es una herramienta válida para segmentar automáticamente imágenes con espermatozoides humanos, entregando como resultado la cabeza de cada uno de ellos. Al mismo tiempo, y con el fin de aportar a la investigación de segmentación automática, se deben realizar diversas comparaciones con las últimas publicaciones sobre el tema, como también el uso de otras tecnologías o métodos frente a la segmentación de espermatozoides. Y así evaluar si la hipótesis planteada se cumple o no.

### **1.3 PLANTEAMIENTO DE HIPÓTESIS**

Dentro de las preguntas de investigación, se menciona la necesidad de comprobar si la herramienta computacional Deep Cell, que crea una red neuronal profunda, tiene algún beneficio en la investigación y el proceso de segmentación en general, particularmente en la precisión. A partir de esto, se plantea la siguiente hipótesis:

**La herramienta Deep Cell mejora la precisión en la segmentación automatizada de partes de espermatozoides humanos.**

### **1.4 HERRAMIENTA A UTILIZAR**

La herramienta Deep Cell (Logo en figura 1.1), creada por el Instituto Tecnológico de California (CALTECH), es una de las plataformas más recientes en el área de la creación de modelos de aprendizaje profundo para segmentar y seguir organismos unicelulares. Fue presentada en el año 2018 en un artículo publicado por el grupo Cold Spring Harbor Laboratory (Bannon et al., 2018a), como una alternativa de código abierto a los investigadores y científicos interesados en integrarse al área de la inteligencia artificial, y la posibilidad de crear modelos de aprendizaje profundo de manera fácil y sencilla. En el documento se menciona la importancia de la integración remota (a través de AWS) a especialistas, y la fuerza que las redes neuronales convolucionales tienen en el área médica. En un artículo posterior (Bannon et al., 2018b), se ven diversos avances en la herramienta, integrando la plataforma de Google Cloud, y la publicación de la biblioteca deepcell-tf, la cual entrena y genera modelos de redes neuronales profundas utilizando la GPU como principal motor en un ambiente experimental local. Frente a esta última publicación, diversos investigadores y entusiastas en el área del *Deep Learning* empezaron a utilizar esta nueva herramienta en distintas áreas donde se pueden visualizar organismos unicelulares, principalmente para comparar los métodos de segmentación ya documentados. La hipótesis planteada en este trabajo de titulación consiste en realizar un análisis sobre el rendimiento y precisión obtenidos al utilizar la librería Deep Cell para segmentar la cabeza de espermatozoides humanos, utilizando una red neuronal de aprendizaje profundo en su totalidad, y comparar estos resultados con las diversas investigaciones, que van desde el uso de técnicas del análisis de datos, hasta el uso de herramientas de inteligencia artificial para apoyar diversos modelos (Véase Capítulo 2).



# Deep Learning for Single-Cell Analysis

Figura 1.1: Logo de la biblioteca computacional deepcell-tf

Fuente: (Deep Cell, 2019)

## 1.5 OBJETIVOS Y ALCANCE DEL PROYECTO

### 1.5.1 Objetivo general

El objetivo general es corroborar si el uso de herramientas computacionales como la plataforma Deep Cell 2.0, que permite generar y entrenar redes neuronales profundas, aportan (en la precisión) a la segmentación de partes de espermatozoides humanos, en comparación con las últimas publicaciones en el área, en otras palabras, la comprobación de la hipótesis planteada.

### 1.5.2 Objetivos específicos

- **Aumentar los datos de entrenamiento:** como en cualquier aplicación de redes neuronales profundas, se requieren una gran cantidad de datos para iniciar las sesiones de entrenamiento, por lo que es necesario implementar métodos que permitan aumentar estos datos. Para realizar este objetivo, es importante considerar y comparar las distintas alternativas para realizar el incremento en el número de imágenes, incluyendo técnicas tradicionales, algoritmos específicos para incrementar los datos en organismos unicelulares, o el uso de técnicas relacionadas con la inteligencia artificial, ó específicamente con el *deep learning*, como lo son los *auto-encoders*. Además, debido a que los datos corresponden a muestras médicas, es importante verificar que no se generen imágenes invalidadas por

inverosimilitudes con una muestra real, que pueda afectar en el entrenamiento de la red neuronal.

- **Entrenar modelos neuronales mediante Deep Cell 2.0:** La creación de modelos utilizando la plataforma Deep Cell requiere una serie de actividades, a saber, la modificación de parámetros para mejorar el proceso de preparación y entrenamiento, la elección de modelos, la integración del dataset generado por la finalización del objetivo antes mencionado, la constante revisión del proceso de entrenamiento y preparación, el monitoreo de la carga remota y una etapa de validación correcta. Todo esto para entregar la red neuronal profunda en el mejor estado posible, es decir, que realice una correcta segmentación automática de espermatozoides humanos con una precisión medible.
- **Analizar y discutir técnicas y parámetros de modelos generados:** El análisis de resultados no solo conlleva el paso esencial para afirmar o refutar la hipótesis planteada sino medir el impacto que tienen sobre los resultados una serie de parámetros, técnicas, y arquitecturas, por ejemplo, visualizar el cambio en el rendimiento al utilizar una técnica utilizada en las redes neuronales profundas como *transfer learning*, o una arquitectura de red neuronal profunda distinta, entre otros.

### 1.5.3 Alcances

El alcance de este trabajo de investigación es el desarrollo de red neuronal profunda para la segmentación automática de espermatozoides humanos utilizando la herramienta Deep Cell, y no otra técnica u herramienta de aprendizaje profundo en particular. Esto se realiza principalmente porque distintas universidades o especialistas se encuentran experimentando con diversas técnicas de Deep Learning, como el uso de auto-encoders, SVM, GAN, etc. (Revollo Sarmiento et al., 2020), u otras herramientas que realizan un trabajo similar a Deep Cell, como es el caso particular de U-net (Ronneberger et al., 2015), específicamente en la segmentación automática de espermatozoides humanos. Es decir, este trabajo de investigación busca ser, además de un precedente en caso de que la hipótesis planteada sea veraz, una fuente de comparación con otros métodos creados que se publicarán en un futuro, ya sea con métodos de inteligencia artificial, aprendizaje profundo u otra alternativa.

Además, es importante mencionar que este trabajo de investigación utilizará, como base, el Gold Standard y el dataset SCIAN-SpermSegGS como datos para la evaluación de los modelos sin ningún proceso de preprocesamiento, debido a que son estos datos se han utilizado a lo largo de la investigación, y realizar una comparación con los diversos métodos ya creados,

además de ser un set de datos claro, con un tamaño considerable y con una segmentación de referencia realizada por expertos, y así, medir la precisión este trabajo de investigación.

## 1.6 METODOLOGÍA Y HERRAMIENTAS

Uno de los principales aspectos en cualquier trabajo de investigación es la definición de la metodología, que sea capaz de comprobar la veracidad de la hipótesis planteada. De la misma manera, conocer las herramientas de trabajo utilizadas durante el desarrollo de las distintas tecnologías permite entender el funcionamiento general del software, y en casos específicos, como el desarrollo de redes neuronales afecta a los tiempos de trabajo.

### 1.6.1 Metodología

Uno de los principales elementos a considerar a la hora de trabajar en una memoria es la metodología, la cual define la forma a la hora de trabajar para lograr los diversos objetivos propuestos.

Debido a la naturaleza de esta memoria, enfocada principalmente en el desarrollo de redes neuronales profundas, y los objetivos específicos (Véase capítulo 1) definidos buscan el desarrollo (y posterior análisis) la metodología de trabajo se crea en base a los objetivos específicos. Esto conlleva a la realización enfocada de cada objetivo de manera secuencial: preparación, creación y desarrollo, y posterior análisis de la red neuronal utilizando la herramienta Deep Cell. Esto se apoya con el uso de Diagramas de flujo, una herramienta bastante utilizada durante el diseño de redes de aprendizaje profundo y el diseño de procesos en general (Nooteboom, 2018), que aporta al entendimiento general del sistema en base de distintos sistemas que reciben entradas y salidas.

Al mismo tiempo, cada objetivo específico posee una serie de tareas (visibles en la tabla 1.1) a realizar para considerarse completado, lo cual conlleva a una completitud del objetivo, para iniciar el paso definido según la metodología. Además, debido a la naturaleza del trabajo con redes neuronales, existe la posibilidad de tener que rediseñar determinadas tareas, con el fin de corregir errores durante el desarrollo de un objetivo posterior, sin embargo, la definición de las tareas es atómica y fácilmente modificable con el fin de evitar una gran modificación de otros objetivos, al igual que los mencionados en importantes organizaciones del desarrollo de proyectos (O'Reilly, 2020).

Tabla 1.1: Selección de tareas por objetivo

**Fuente:** Elaboración propia, 2020

Objetivo	Tarea
Aumentar de los datos de entrenamiento	Implementación de proceso de aumento de datos.
	Revisión de imágenes generadas (filtrado manual).
	Separación de dataset por porcentajes establecidos.
	Creación de dataset en formato numpy (npz).
Entrenar modelos neuronales mediante Deep Cell 2.0	Instalación de archivos necesarios en servidor remoto.
	Importación de Dataset ampliado y definición de directorios.
	Definición del tipo de modelo y sus parámetros.
	Entrenamiento del modelo Deep Cell.
Analizar y discutir técnicas y parámetros de modelos generados:	Elección de imágenes específicas a probar en el modelo.
	Evaluación del modelo y exposición
	Análisis de los resultados del modelo
	Comparación y análisis con otros métodos de segmentación automática.

En el capítulo 5 se comenta en detalle cada tarea, además de su importancia en el diseño general de la realización de la memoria, su importancia para verificar la hipótesis planteada, y la forma de implementación de cada tarea.

### 1.6.2 Herramientas de desarrollo

Tal como está definido en la sección 1.5.1, la metodología incluye "dos procesos" con distinta carga computacional, siendo la creación de métodos de incrementos de datos, creación y división del dataset, y técnicas de evaluación de resultados son tareas de bajo costo computacional, por lo que no requiere de un ordenador con grandes características. En caso contrario, el entrenamiento de un red neuronal de *deep learning* requiere una gran capacidad de procesamiento, tanto de CPU como de GPU, y esta necesidad aumenta con la utilización de la herramienta Deep Cell, donde los creadores indican que uso de GPU de la marca NVIDIA posee un mejor rendimiento que el uso de CPU's especificadas, determinada por el puntaje CUDA de la unidad de procesamiento. Frente a esto, se presentan las especificaciones de las herramientas de hardware destinadas al desarrollo con baja carga computacional, en la tabla 1.2, y el servicio remoto para el entrenamiento de la red neural en la tabla 1.3

Tabla 1.2: Especificaciones de Hardware para baja carga computacional

**Fuente:** Elaboración propia, 2020

<b>Modelo</b>	Asus X456U
<b>Sistema Operativo</b>	Manjaro 18.04 Xfce
<b>Procesador</b>	Intel I7-6500U (2 nucleos fisicos, 4 subprocessos, 2,50 GHz)
<b>Memoria Ram</b>	8 GB DDR3L 1600 MHz SDRAM
<b>GPU Integrada</b>	Intel® HD Graphics 520
<b>GPU Dedicada</b>	NVIDIA Geforce 940MX de 2GB de VRAM

Tabla 1.3: Especificaciones de Hardware para alta carga computacional

**Fuente:** Elaboración propia, 2020

<b>Modelo</b>	Servidor Remoto DIINF
<b>Sistema Operativo</b>	Ubuntu 1804 LTS
<b>Procesador</b>	Intel(R) Core (™) i9-9820X CPU @ 3.30Ghz
<b>Memoria Ram</b>	128 GB
<b>GPU Integrada</b>	-
<b>GPU Dedicada</b>	RTX TITAN X (24GB)

## 1.7 ESTRUCTURA DEL DOCUMENTO

En relación al formato de este documento, exceptuando el capítulo actual, correspondiente a la introducción de la memoria, se compone de cinco capítulos generales que abarcan todo el desarrollo de la memoria. El capítulo 2 corresponde al marco teórico, y una explicación general sobre la forma de los espermatozoides, las partes que los componen y su funcionalidad, un marco conceptual sobre conceptos de la inteligencia artificial, dando

énfasis en la importancia de las redes convolucionales, aprendizaje profundo, aplicación en la ciencia y medicina, modelos de redes neuronales, además del funcionamiento computacional de la herramienta Deep Cell. El capítulo 3 corresponde a la investigación hasta la fecha, donde se incluye gran parte de la investigación realizada al día de la entrega de este documento sobre la segmentación de espermatozoides, incluyendo la definición del Gold Standard para la segmentación, el artículo de Nissen (Nissen et al., 2017), que corresponde a la base de las redes neuronales en el área, y las últimas investigaciones publicadas. El capítulo 4 corresponde al diseño e implementación de las herramientas generadas para afirmar o refutar la tesis, donde se explica el proceso de creación y funcionamiento de todas las tareas mencionadas en la metodología, sumado al ambiente experimental, descripción de los datos, y diversos elementos del desarrollo de la memoria. Durante el capítulo 5, se plantean los resultados obtenidos. En el capítulo 6, con todo el proceso de implementación realizado, es posible realizar un Análisis de resultados, donde se discute sobre los resultados en base al modelo de aprendizaje profundo, y la posterior comparación con otros métodos ya publicados sobre la segmentación automática de espermatozoides. Finalmente, el capítulo 7 corresponde a la sección de conclusión, y a partir de lo observado durante el análisis de resultados, es posible entregar conclusiones, y refutar o afirmar la hipótesis planteada.

## CAPÍTULO 2. MARCO TEÓRICO

Poseer una idea o definición sobre las distintas áreas que se exponen en este documento es importante para entenderlo en su totalidad. En este capítulo se busca informar sobre los conceptos en relación a los espermatozoides, inteligencia artificial, aprendizaje profundo y herramientas específicas, como Deep Cell.

### 2.1 MARCO CONCEPTUAL BIOLÓGICO

El espermatozoide (o espermio) es una célula única en el ser humano, y, en conjunto con el óvulo, es responsable de la existencia de la vida. Es la célula sexual encargada de la reproducción de parte del ser humano masculino (también conocido como gameto masculino), y posee la mitad de los cromosomas que posee un humano, incluyendo el cromosoma decisivo para determinar el sexo. Al unirse con el gameto femenino, el óvulo, durante el proceso de reproducción humana, se genera el proceso de la fecundación, generando el cigoto humano, que evoluciona hasta el momento del parto o nacimiento, en el caso promedio (Monroy, 2020).

El espermatozoide se compone de tres partes fundamentales, como se puede ver en la imagen 2.1, la cabeza, la pieza media, y la cola. En la cabeza del espermio se subdivide en otras dos partes, el acrosoma, ubicado en la parte superior, y el núcleo, ubicado en la parte inferior. El núcleo es el lugar donde se almacena los cromosomas del padre, y el acrosoma lo protege, como también es capaz de atravesar el núcleo del óvulo para iniciar el proceso de fecundación, todo esto gracias a su composición de proteínas (MedicalNewsToday, 2020). La pieza media, tal como su nombre lo indica, corresponde a la sección intermedia entre la cabeza y la cola de la célula; en ella se encuentra la mitocondria, conocida también como “la fuente de poder” del espermatozoide, que genera la energía necesaria para desplazarse (Clare & Jill, 2010). Finalmente, la cola o flagelo, realiza un movimiento similar a lo que realiza un nadador bajo el agua, impulsa al espermatozoide en su destino hacia el gameto femenino (MedicalNewsToday, 2020).

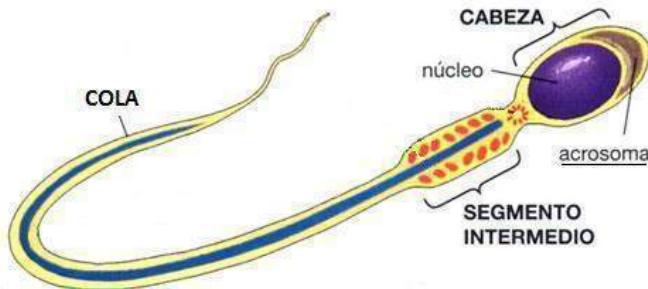


Figura 2.1: Estructura del Espermatozoide

Fuente: (EcuRed, 2015)

Los espermatozoides se generan en un proceso llamado espermatogénesis, que empieza a ocurrir durante la pubertad, tras la regulación de la proteína BMP8B, donde se empieza a generar la espermatogonias (Gilbert, 2000). Luego, inician una serie de procesos celulares, iniciando con una mitosis (proceso donde las células se reproducen, generando dos a partir de una), y una posterior meiosis (unión de células) para generar los espermatozoides generales (Adé-Damilano, 2008). Finalmente, este proceso se va repitiendo cada 62 o 75 días (dependiendo de cada persona) y finaliza con la generación de espermios determinados. El proceso de la espermatogénesis se puede ver en la imagen 2.2

La infertilidad humana masculina tiene, como uno de sus principales factores, al nivel de calidad de semen. Este nivel depende de cinco mediciones principales: La cantidad de espermatozoides generados por la espermatogénesis, el cual depende principalmente de la etnia de una persona, pero un número aproximado para el mundo occidental se encuentra en el rango de los 193 a 398 millones de espermatozoides por día (Amann, 2008). El número de espermatozoides es uno de los principales elementos para determinar la infertilidad humana (Zacà et al., 2020), por lo que la detección, y posterior segmentación de las partes de una muestra son pieza esencial a la hora de dar un diagnóstico certero. El volumen de semen liberado, la concentración de espermatozoides en el líquido seminal, la capacidad de los espermatozoides para moverse de manera independiente, a través de su mitocondria, es palabras específicas, la motilidad del semen.

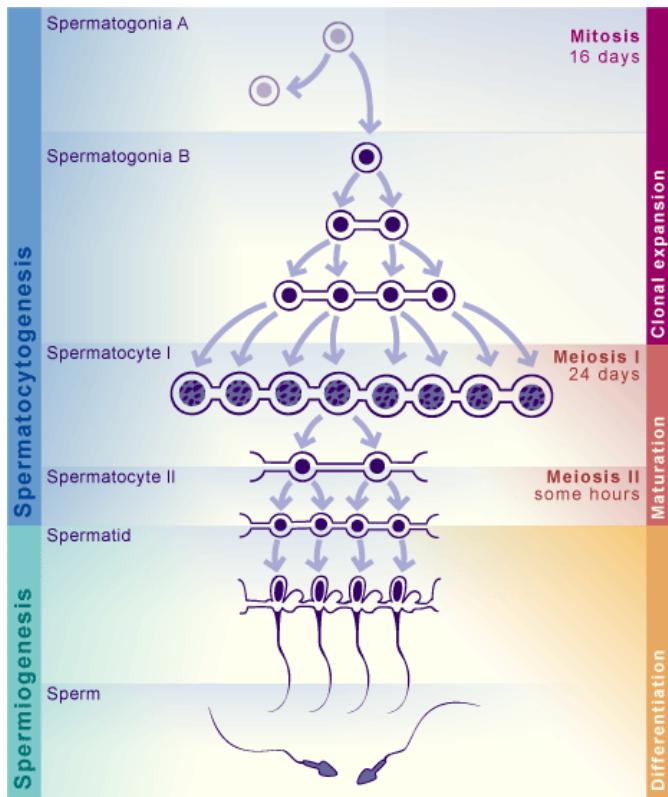


Figura 2.2: Proceso de la espermatogénesis

Fuente: (Adé-Damilano, 2008)

La morfología se define como la forma o estructura de un organismo, objeto, ser vivo, etc (ASALE & RAE, 2020), dentro de la morfología de un organismos unicelular es posible identificar su forma específica (circular, elíptica, etc.), sus partes, y las distintas mutaciones o malformaciones que pueda poseer. Esto es uno de los aspectos que más requieren de un especialista, ya que existen distintos tipos de “problemas morfológicos”, que están asociados a distintas patologías. En la imagen 2.3, se describe los defectos asociados a las partes de los espermios. Estudios recientes han confirmado que la detección de defectos en la cabeza, pieza media y cola pueden tener efecto en la infertilidad masculina, como también en otras patologías (Auger et al., 2016), y, al mismo tiempo, las razones de los defectos, como el artículo que define que una cantidad de la enzima Topoisomerasas de tipo II encontrada en muestra de semen propician los defectos de cabeza del espermio (Netherton et al., 2020).

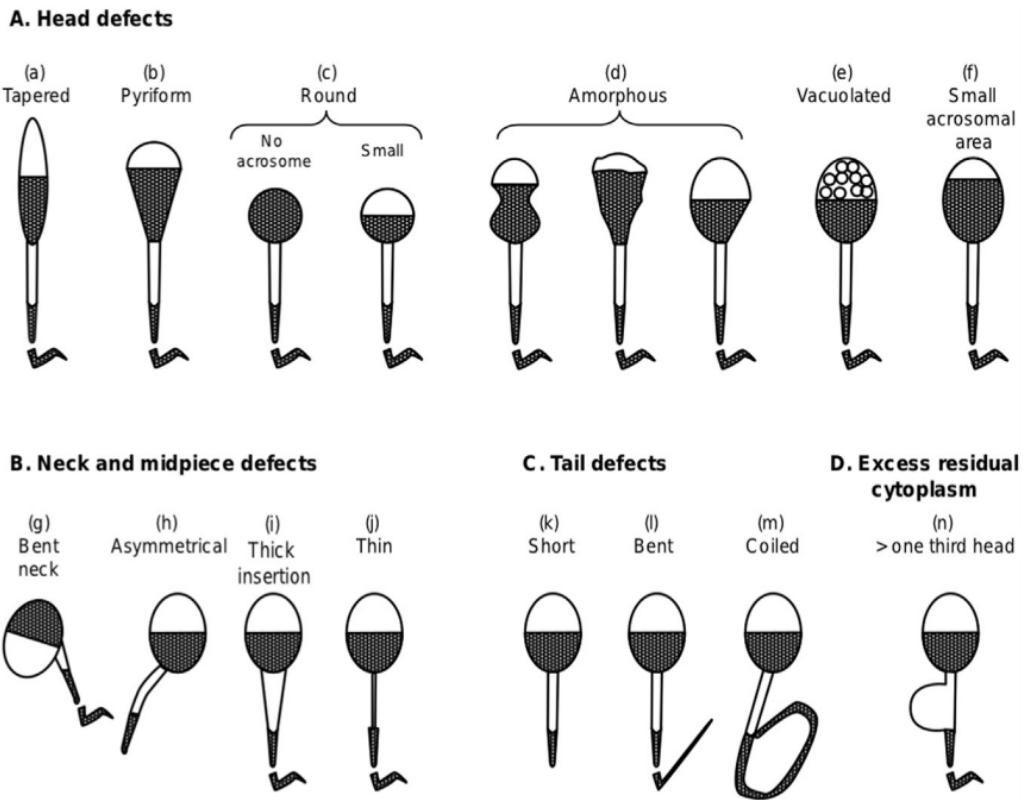


Figura 2.3: Defectos asociados a la morfología de los espermatozoides, separados por la parte defectuosa. Ya sea la cabeza (A), la pieza media (B), la cola (C), o exceso de citoplasma (D)

Fuente: (OMS., 2017)

## 2.2 MARCO CONCEPTUAL

### 2.2.1 Deep Learning

La aplicación del *deep learning* en sistemas computacionales ha abierto un sinfín de posibilidades en múltiples áreas, incluyendo el sector empresarial, logístico, político y médico. Por definición, el *deep learning* son los métodos de la rama de la inteligencia computacional (*o machine-learning*), que mediante modelos de representación en múltiples capas, logran realizar labores de clasificación y regresión con un mayor nivel de abstracción, a partir de datos de diversas entradas, con estructuras de aprendizaje supervisados (con parámetros introducidos por un especialista) o no-supervisados (el mismo método define los parámetros) (Lee et al., 2017). Basándose en las conexiones que realizan las neuronas para enviar información, las

redes neuronales de *deep learning* se dividen en tres tipos de capas: la capa de entrada con, consiguientemente, datos de entrada, la capa de salida, que entrega resultados a partir de los datos de entrada, y las capas ocultas, que se encuentran en medio de las capas ya mencionadas y que mediante decisiones simples (conocidas como funciones de activación) actúan como las neuronas en el cuerpo humano, enviando información y aprendiendo según el nivel de potencia de la información. Estas “neuronas”, interconectadas entre sí, permiten llegar a generar modelos de aprendizaje en diversas áreas, al recibir nuevos datos de entradas. Es importante mencionar la importancia de cumplir con diversos elementos para generar un buen modelo de aprendizaje, siendo uno de los principales los datos que sirven para entrenar el modelo, tanto en calidad, es decir, que los datos representan la intención del modelo, como en cantidad, que sean suficientes para atravesar por todas las neuronas del modelo (Lee et al., 2017). Un ejemplo gráfico de como se ve una red neuronal se ve en la figura 2.4.

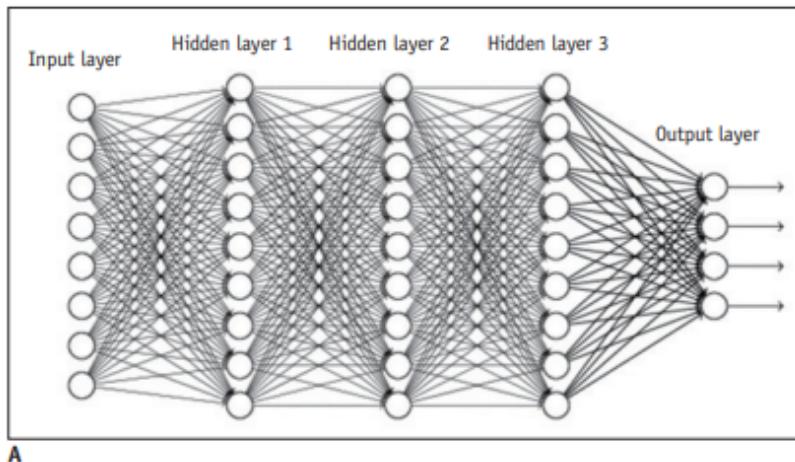


Figura 2.4: Composición básica de una red neuronal de múltiples capas

**Fuente:** (Lee et al., 2017)

Las redes neuronales convolucionales (CNN) permiten aplicar los fundamentos de la inteligencia artificial y las redes neuronales profundas al trabajo con imágenes, logrando tareas de segmentación y clasificación, aportando a cualquiera de sus aplicaciones, incluyendo el área médica y científica. Las redes neuronales convolucionales, tal como dice su nombre, utilizan la operación de convolución, en conjunto con diversos tipos de filtros o máscaras, como también otras herramientas como el *max-pooling* o *average pooling* (elección de valores en base a un criterio, máximo o promedio, respectivamente), para obtener características de las imágenes. Al igual que una red neuronal profunda, es posible aplicar un gran número de capas ocultas en CNN, y así aplicar métodos de *deep learning* logrando tareas de reconocimiento de patrones o características en aplicaciones que no era posible en redes neuronales tradicionales, lo que

amplió en gran medida el espectro donde el área de aprendizaje profundo se extendió, creando una disciplina conocida como la visión por computador (Forsyth & Ponce, 2003), y llegando incluso a la medicina (Lee et al., 2017). En las imágenes 2.5 y 2.6 es posible visualizar algunos de los conceptos de inteligencia artificial y *deep learning* aplicado a imágenes.

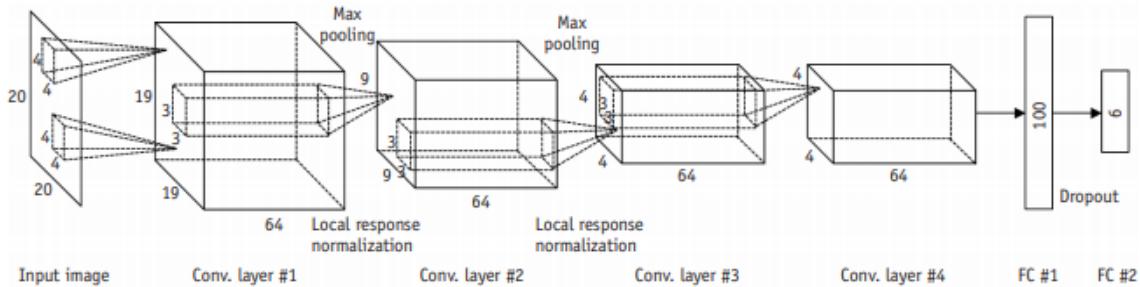


Figura 2.5: Arquitectura de una red neuronal convolucional (CNN), incluyendo entrada, convolución y pooling.

**Fuente:** (Lee et al., 2017)

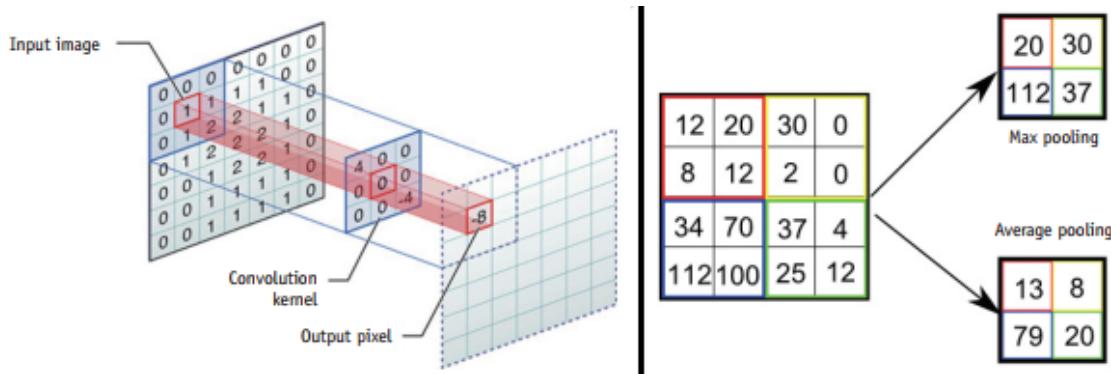


Figura 2.6: Ilustración de métodos de convolución y pooling.

**Fuente:** (Lee et al., 2017)

El uso de modificaciones, alteraciones y técnicas del procesamiento de imágenes, es necesario para incrementar los datos para crear correctos modelos de redes neuronales convolucionales profundas con imágenes. La posibilidad de usar estrategias para generar imágenes artificiales, con el fin de incrementar la calidad, y sobre todo la cantidad de los datos, es una parte esencial de la aplicación de *deep learning* en imágenes. Por ejemplo, en la investigación sobre la segmentación de núcleos celulares inmunofluorescentes (Kromp et al., 2019), los autores, además de utilizar métodos tradicionales para la aplicación de imágenes a redes convolucionales, crearon una estrategia que busca generar muestras artificiales a partir de las muestras originales. Específicamente, este método obtiene cada uno de los núcleos celulares de la primera muestra, para luego posicionarse en una nueva imagen, basándose en una

grilla generada a partir de procesos de procesamiento de imágenes, esto sumado a un número importante de transformaciones que se realiza a cada uno de estos núcleos, como rotaciones, escalamiento, desplazamiento, entre otras. El número de imágenes posibles para segmentar puede incrementar drásticamente, como también, generar casos conocidos por los expertos en el tema, pero que la muestra original no poseía, generando una segmentación que pueda cubrir el mayor espectro de posibilidad encontrables en una muestra real.

## **2.2.2 Segmentación de imágenes médicas**

La llegada de estas herramientas computacionales a la medicina, y particularmente a la segmentación de imágenes, ha logrado avances en procedimientos de exámenes, detección de enfermedades, elección de tratamiento, y avances científicos que aportan a nuevos diagnósticos, como soluciones a enfermedades. A modo de ejemplo, la segmentación de una radiografía de una rodilla, en conjunto con una modificación de un CNN, además de diversas técnicas, tanto computacionales como médicas, permite, en un inicio, mejorar la calidad de la radiografía, entregando una mejor comprensión del paciente para los especialistas, como también, la probabilidad de tener una lesión de esclerosis múltiple, o sufrir una posteriormente (Brosch et al., 2016). Procedimientos como estos pueden ser llevados a una plataforma computacional conocida como CAD, que es una herramienta computacional que permite detectar y diagnosticar una enfermedad a partir de un examen determinado. Todo el proceso de lesión de esclerosis múltiple se puede ver gráficamente en la figura 2.7.

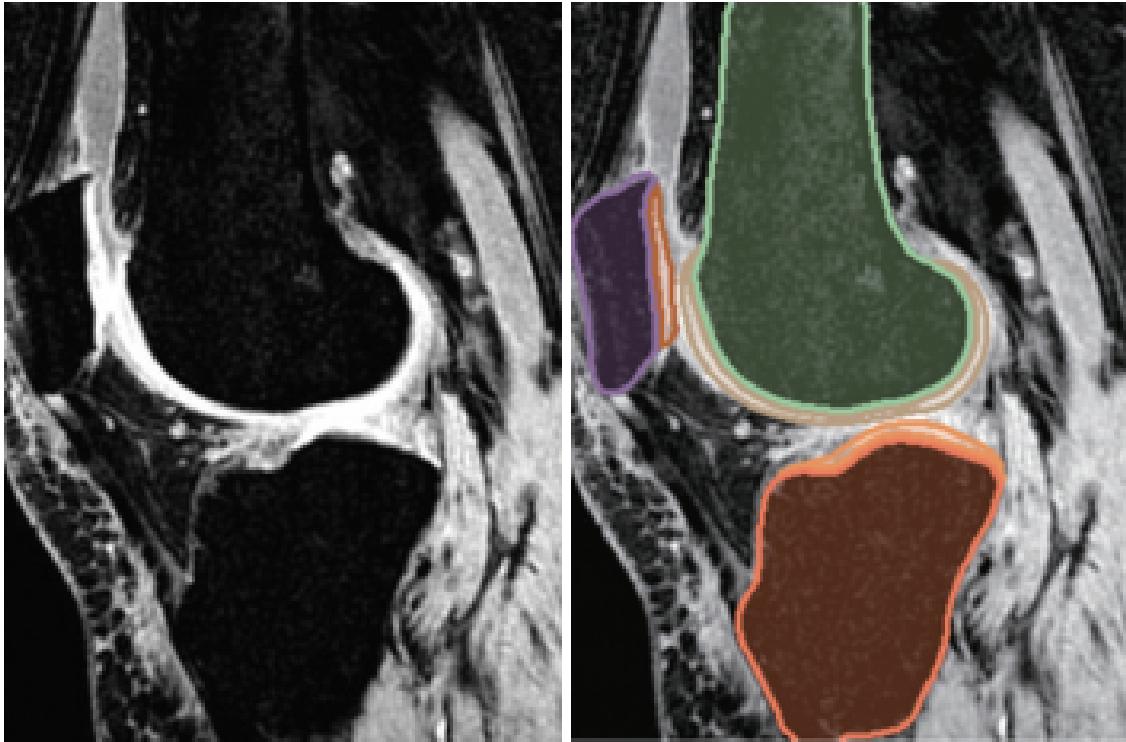


Figura 2.7: Radiografía de una rodilla antes y después de un proceso de segmentación

**Fuente:** (Lee et al., 2017)

Al mismo tiempo, el uso de segmentación de imagen y las CNN ha sido ampliamente utilizado en el área científica-médica, particularmente, ayudando en el estudio de células humanas, relacionadas a alguna enfermedad. Cuando un especialista revisa una muestra de células a través de un microscopio, suele enfocarse en una región de interés, conocida como ROI (Ver Glosario) en particular de cada unidad. A partir de la observación de estas, es posible detectar malformaciones, deficiencias u otros elementos que permiten a un médico realizar un diagnóstico determinado, sin embargo, existen diversos problemas a la hora de realizar este proceso. En algunos casos, la determinación de un número de células que posee una malformación es subjetivo, lo que puede derivar en distintos diagnósticos, y por lo tanto diversos tratamientos. De igual forma, al revisar una muestra de células, existe una gran problemática al observar elementos sobrepuertos, lo cual puede impedir al especialista tomar una decisión frente a ese caso particular (Bajcsy et al., 2015). Además, es imposible olvidar que la segmentación manual o tradicional, es decir, cuando se realiza por el especialista de manera manual, identificando y marcando las partes “sobre un papel”, es inmensamente subjetivo, laborioso, lento, e incluso puede llegar a ser impreciso, como ha explicado Charlene Brazil en su artículo (Brazil, 2010), sobre el análisis de semen y espermatozoides.

Un problema del uso de CNN en conjunto con algoritmos de *deep learning* es la

poca cantidad de datos o imágenes que se posee, frente a esto, la posibilidad de usar estrategias para generar imágenes artificiales, con el fin de incrementar la calidad, y sobre todo la cantidad de los datos, es una parte esencial de la aplicación de *deep learning* en imágenes. Por ejemplo, en la investigación sobre la segmentación de núcleos celulares inmunofluorescentes (Kromp et al., 2019), los autores, además de utilizar métodos tradicionales para la aplicación de imágenes a redes convolucionales, crearon una estrategia que busca generar muestras artificiales a partir de las muestras originales. Específicamente, este método obtiene cada uno de los núcleos celulares de la primera muestra, para luego posicionarse en una nueva imagen, basándose en una grilla generada a partir de procesos de procesamiento de imágenes, esto sumado a un número importante de transformaciones que se realiza a cada uno de estos núcleos, como rotaciones, escalamiento, desplazamiento, entre otras. El número de imágenes posibles para segmentar puede incrementar drásticamente, como también, generar casos conocidos por los expertos en el tema, pero que la muestra original no poseía, generando una segmentación que pueda cubrir el mayor espectro de posibilidades encontrables en una muestra real.

A partir de estos problemas, el uso de la segmentación automática de imágenes y el uso de la inteligencia artificial, en especial técnicas novedosas y eficientes como el aprendizaje profundo, son esenciales para lograr un buenos resultados. Una aplicación concreta de esto es la segmentación de espermatocitos humanos, donde el uso de estas técnicas computacionales busca aumentar la eficiencia en los diagnósticos, como también el grado de exactitud entre los resultados presentados por expertos, y un sistema computacional.

### **2.2.3 Aumento de datos**

Tal como se explicó anteriormente, el número de datos es importante a la hora de realizar una red neuronal, incluso mayor si hablamos de técnicas de *deep learning*. Esta disyuntiva se hace mayor en el sector médico, donde los datos son reducidos en comparación a otras áreas (por ejemplo, publicidad, historial de páginas web, etc.), debido a que suelen ser datos específicos (en una parte del cuerpo, por ejemplo), como también la necesidad de poseer el consentimiento del paciente a la hora de recopilar datos para la creación de un dataset. Frente a esta situación, es necesario aumentar el número de datos usando métodos claros, sencillos, que sean consecuentes con los datos originales, y coherentes con la teoría, este caso, con los espermatocitos humanos. Esta área se conoce como *Data Augmentation* o incremento de datos.

Los modelos de *deep learning* necesitan poseer un número importante de datos para evitar *overfitting*, o sobreajuste, que es un concepto que habla sobre la gran varianza que puede

poseer una red neuronal si no se tienen los datos necesarios, logrando que el modelo no converja, tanto a nivel de calidad como de cantidad. Para evitar el sobreajuste del modelo, es necesario generar datos con otros métodos especializados, que utilizan técnicas del procesamiento de imagen, como también de la inteligencia artificial.

Según una investigación del año 2019 (Shorten & Khoshgoftaar, 2019), existen 2 tipos principales de métodos de *data augmentation*, técnicas relacionadas a la manipulación de imágenes, los cuales se subdividen en transformaciones geométricas (rotación, traslación, reflejo, entre otros), transformaciones fotométricas y las técnicas basadas en *deep learning*, que involucran la creación de redes neuronales para generar nuevos datos, a saber, modificación de espacios de características, el uso de GAN (Ver Glosario), conocidas como redes adversarias, meta aprendizaje, entre otros. Las principales diferencias entre estos dos tipos de procesos son los recursos utilizados, y el ajuste a la red.



Figura 2.8: Ejemplo de transformación fotométrica, cambio de contraste

**Fuente:** (Shorten & Khoshgoftaar, 2019)

Mientras que los métodos de *deep learning* suelen evitar el *overfitting* en la mayoría de los casos, estos suelen usar una gran cantidad de tiempo, recursos computacionales y pre procesamiento, lo cual puede incrementar la carga computacional, en comparación a la red neuronal que se quiere entrenar en un principio. En cambio, los métodos basados en el procesamiento de imágenes pueden no ser tan eficaces en todos los contextos, además de requerir en gran medida un filtrado manual, para determinar qué imágenes sirven luego del incremento de imágenes, sin embargo, la baja carga computacional en relación al uso de *deep learning* justifica su uso en determinados casos (Shorten & Khoshgoftaar, 2019).

Al llevarlo al contexto del área de la imagenología médica, ambas formas de incremento de datos han sido utilizadas, siendo la más común el uso de *deep learning*. A pesar de lo anterior, diversos estudios han demostrado que se necesita una gran cantidad de tiempo, debido principalmente a lo limitado de los datos actuales, siendo el uso de procesamiento de imágenes más eficaz en cuanto a los recursos, y al mismo tiempo, concluyen sobre las mejores y peores transformaciones, tanto geométricas como fotométricas, son más eficaces a la hora de trabajar con imágenes del área médica; siendo el uso de filtros gaussianos y reflección, los que

generan mejores resultados (Hussain et al., 2018).

## 2.2.4 Modelos de Redes Neuronales

Debido al impulso que ha tenido la inteligencia artificial y el *deep learning* diversos especialistas han creado modelos basados en determinadas características que se adecuan a ciertos problemas, en este caso, se evalúan dos modelos que determinan los experimentos a realizar durante este trabajo de investigación, Retina Net y Panoptic.

### 2.2.4.1 *Retina Net*

Retina Net es uno de los modelos de redes neuronales convolucionales más utilizados para segmentación, debido a su facilidad, rendimiento y documentación disponible. Su principal característica es ser una red de detección única, con extracción de características en una red con forma de Pirámide (FNP), como puede verse en la imagen 2.10. Además, poseen una función de pérdida de error durante su entrenamiento de carácter focal de una, mejorando la precisión de los modelos utilizados sin realizar una gran cantidad de recursos computacionales (Lin et al., 2017). En relación a la segmentación en el área médica, es posible encontrar diversos artículos que fomentan el uso de esta red para esta tarea, a saber, su uso en la identificación de masas irregulares, y posteriormente, tumores benignos en mamas humanas (Jung et al., 2018), segmentación de células humanas en 2D usando principios de *deep learning* con Retina Net utilizando un algoritmo especificado en la figura 2.9 (Al-Kofahi et al., 2018), y lo más destacable, es un artículo que muestra diversas mejoras y pruebas con Retina Net, y segmentación de organismos unicelulares utilizando *deep learning* (W. Cui, 2010). Esta última publicación es una de las más importantes de todas, principalmente porque aporta con distintas variables para entrenar la red, además de demostrar el efecto del *transfer learning* en la segmentación de células. Como dato extra, la misma autora es desarrolladora de la herramienta Deep Cell 2.0.

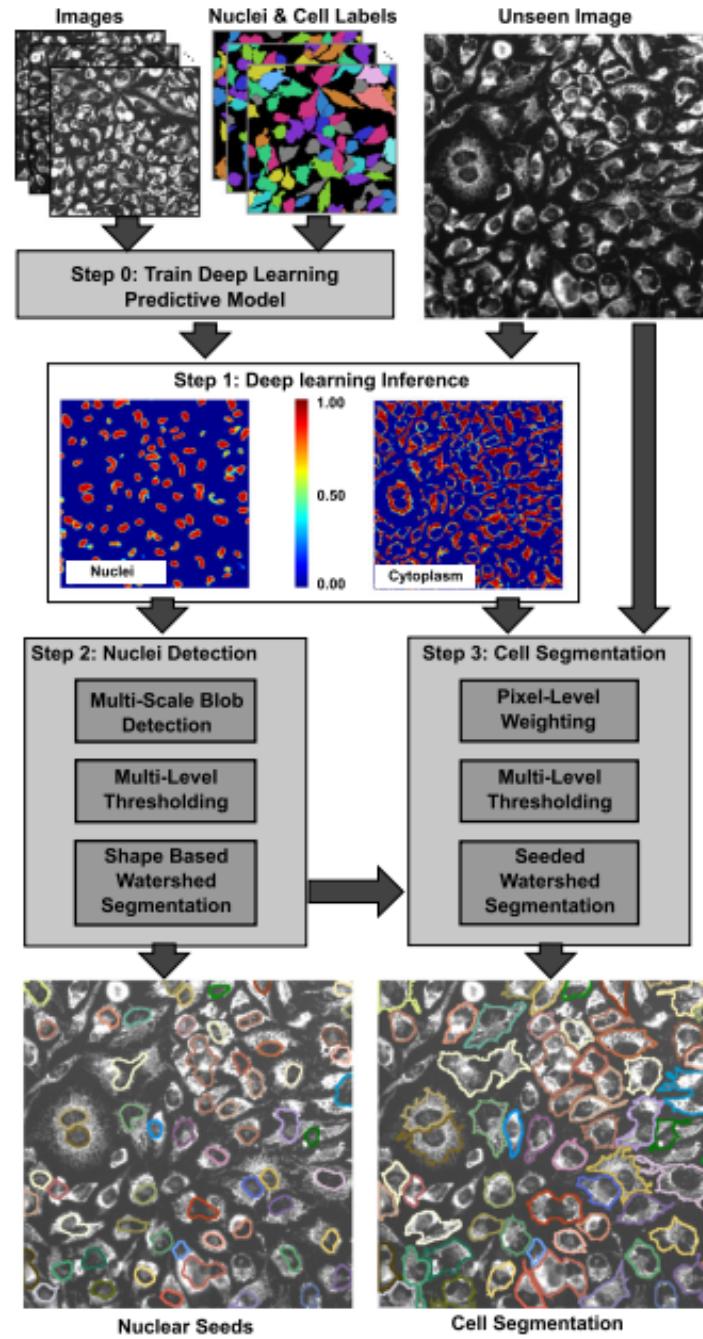


Figura 2.9: Proceso de segmentación de células con Deep Learning. Resumen del algoritmo de segmentación de células 2D usando RetinaNet. Las imágenes iniciales entran al modelo RetinaNet para crear mapas de probabilidades para las células. Luego, múltiples pasos de post procesamiento más tarde, se inicia la segmentación de células a partir de las muestras generadas por el modelo.

**Fuente:** (Al-Kofahi et al., 2018)

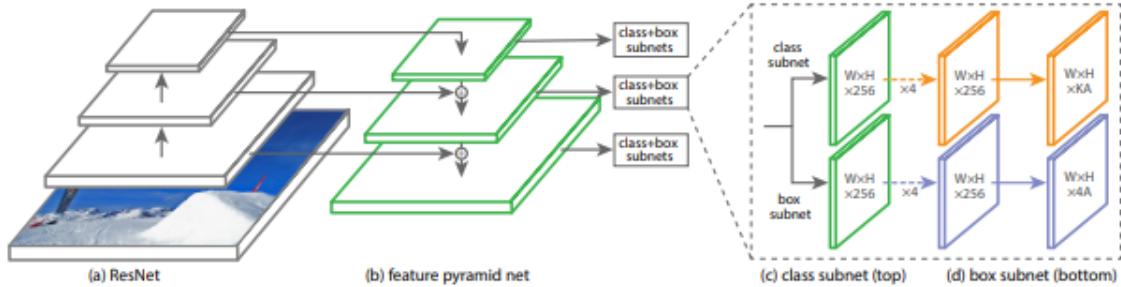


Figura 2.10: Estructura base de un modelo Retina Net. El esqueleto de la arquitectura se basa en una red de extractor de características en forma de pirámide (FPN) (a), que genera otra red neuronal FPN que posee las características a reconocer (b). Al mismo tiempo, esta última red se encuentra conectada a una red de clasificadores en base a las características (c) y otro modelo encargado de analizar los resultados del modelo (d).

Fuente: (Lin et al., 2017)

#### 2.2.4.2 Panoptic

Panoptic, en cambio, es uno de los modelos más novedosos en cuanto a la segmentación en el área médica. Utiliza la misma base de Retina Net, pero actúa con dos niveles de FNP, generando dos segmentaciones, una semántica (compuesta principalmente del fondo) y una de instancia (de forma, con objetos identificables y poco repetibles en una imagen). Esto permite a la red, además de tener una alta precisión, realizar dos reconocimientos, uno a nivel regional (para el fondo) y a nivel de píxel (para la forma) y la oportunidad de generar nuevos datos de prueba (máscaras) a partir de su proceso de entrenamiento (Selman, 2004), lo cual se visualiza en la figura 2.11. Esta última función es muy importante en experimentos donde los datos son limitados, pero en el área médica, debido a la necesidad de poseer datos coherentes con la teoría médica y expertos aprobando los datos, no suele usarse. Al ser una novedad en el área de *deep learning*, existen menos artículos en relación al uso de Retina Net, pero existen algunas excepciones, particularmente con la segmentación en el área médica, por ejemplo, la detección de patologías particulares en radiografías (Zhang et al., 2018), la creación de modelos de segmentación usando Panoptic y bases de *deep learning* (Liu et al., 2019), y la conciliación de un modelo de Panoptic aplicable en imágenes biomédicas en base al artículo anterior, en conjunto con un paradigma para entrenar un modelo similar, cuyo paradigma se ve en la figura 2.12 (Liu et al., 2020).

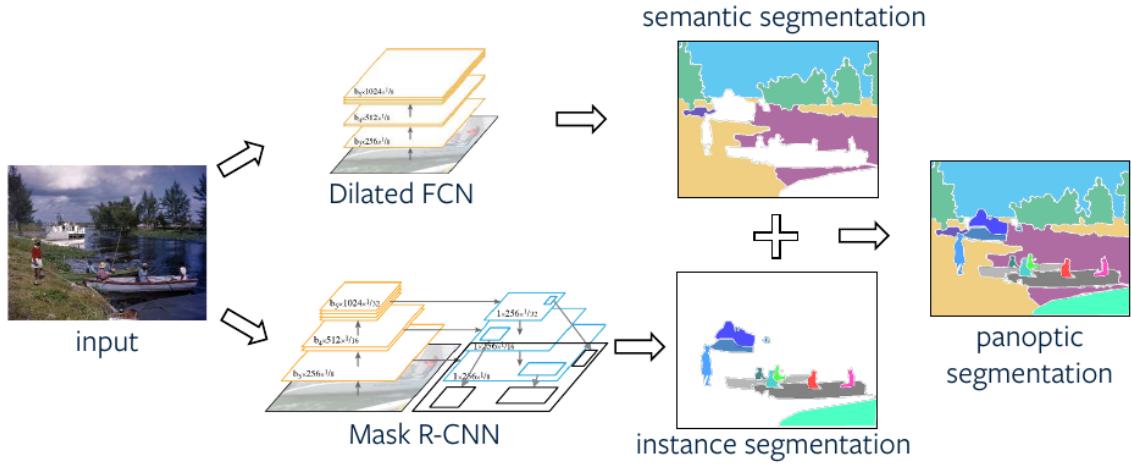


Figura 2.11: Arquitectura tradicional de una red neuronal Panoptic. Mientras que una red convolucional completa se encarga de realizar la segmentación semántica, y una generadora de máscaras con base Retina Net realiza la segmentación de instancias La unión de ambas permite realizar una segmentación completa tanto del fondo, como de los objetos importantes.

**Fuente:** (Kumar, 2019)

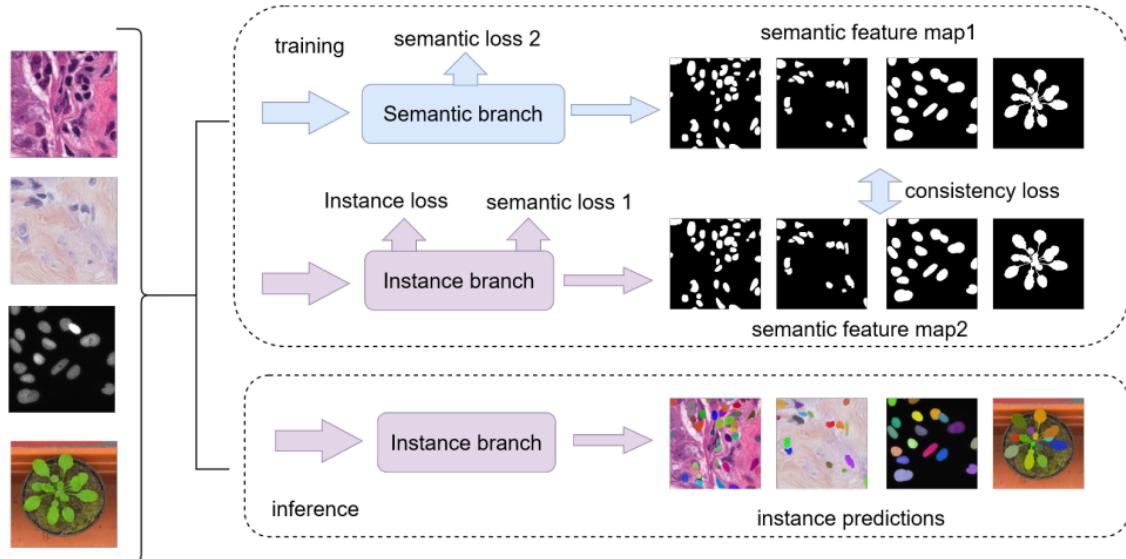


Figura 2.12: Framework y paradigma generado por en modelos Panoptic con imágenes medicas.

**Fuente:** (Liu et al., 2020)

## 2.2.5 Transfer Learning

La evolución en el área de las redes neuronales y la inteligencia artificial ha logrado crear diversos métodos para lograr un mejor entrenamiento de los modelos. Uno de ellos consiste en el *transfer learning*, que corresponde a la transferencia del aprendizaje de un modelo con sus correspondientes parámetros y datos, para servir de base a otro. La técnica del *transfer learning* se define como “el mejoramiento del entrenamiento de un modelo en una nueva tarea, a través de la transferencia de conocimiento desde su entrenamiento en otra tarea relacionada” (Goodfellow et al., 2016). En la práctica, esto se produce al integrar los pesos (que corresponden a los valores de cada conexión entre neuronas de dos capas continuas) de un modelo ya entrenado, en un nuevo modelo con una labor similar. El proceso del *transfer learning* puede ahorrar tiempo y mejorar resultados, debido al principio de la transferencia inductiva, que permite a la red reducir el espectro de búsqueda del error y guiando a la red durante el segundo entrenamiento basándose la historia, los pesos de la red y los resultados del primer entrenamiento (Brownlee, 2017), como puede verse en la imagen 2.13. La segmentación de imágenes es una tarea que usa bastante la técnica del *transfer learning*, debido a la dificultad de poseer un set de datos robusto, y con una cantidad suficiente para realizar una segmentación correcta; utilizando, principalmente, datasets públicos relacionados a la tarea inicial, por ejemplo, la investigación de Cui sobre cómo mejorar la segmentación de células utilizando *transfer learning* con determinados dataset públicos de células (Buslaev et al., 2020), o la creación de una red neuronal de estructura Panoptic universal, usando 3 grandes bases de datos públicas, COCO (imágenes segmentadas de diversos tipos), Cityscapes (imágenes segmentadas para vehículos autónomos) y Mapillary Vistas (imágenes segmentadas de ciudades desde drones) (Cheng et al., 2020).

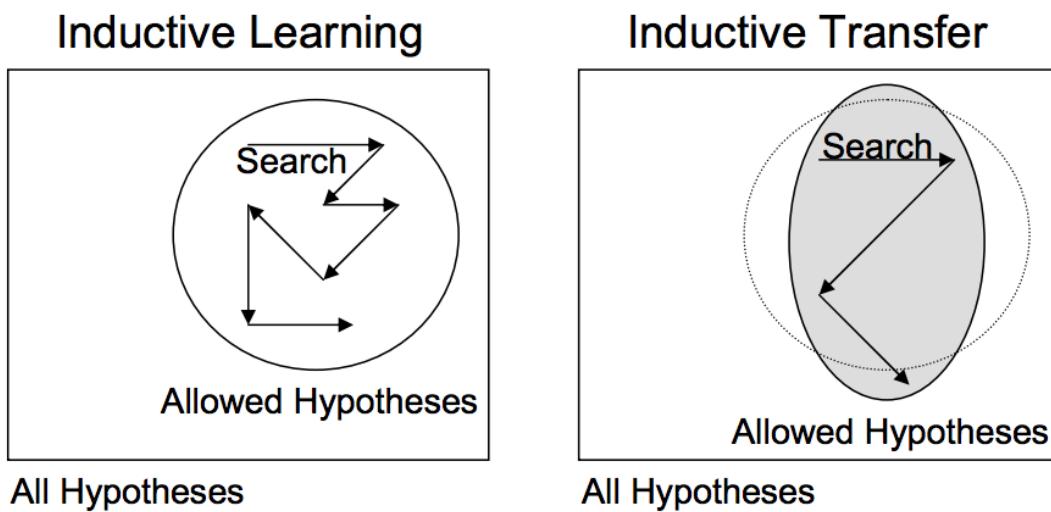


Figura 2.13: Diferencias entre un aprendizaje inductivo y la transferencia inductiva. Mientras que el aprendizaje tradicional busca sus resultados (hipótesis) a través del entrenamiento y la función de error, una transferencia inductiva realiza este procedimiento más rápido, debido que la transferencia “guía” al modelo a buscar resultados por una ruta determinada.

**Fuente:** (Brownlee, 2017)

## 2.2.6 Deep Cell

Deep Cell 2.0 (Bannon et al., 2018c) es una herramienta y biblioteca computacional de código abierto que busca acercar, de manera fácil y remota, técnicas de *deep learning* a científicos o especialistas que trabajan con segmentación automatizada de organismos unicelulares. Creada por especialistas del instituto de tecnología de California (Caltech), se presenta como una plataforma para “crear un modelo de *deep learning* para segmentar imágenes”. Deep Cell busca solucionar diversos problemas asociados al desarrollo de modelos con técnicas del aprendizaje profundo, que se pueden englobar en la infraestructura de software y hardware y el reciente interés de los científicos y especialistas en el trabajo de análisis, detección y segmentación de imágenes celulares.

Para crear un correcto modelo de CNN profundo, que sea eficaz y eficiente, es necesario poseer de una respectiva infraestructura de hardware, que pueda procesar un gran número de imágenes, y un lenguaje y software en general que funcione y aproveche esta infraestructura. Frente a esta disyuntiva, Deep Cell 2.0 se presenta como una plataforma de *Cloud Computing* o computación en la nube, es decir, de manera simplificada, que la demanda computacional en un sitio remoto, y el ordenador del usuario solo se utiliza para labores de

entrada y salida de datos (subir imágenes para el modelo, y recibir las imágenes segmentadas). Al mismo tiempo, la plataforma utiliza, principalmente, *framework* o plataformas habilitadas para funcionar con el lenguaje Python, uno de los más utilizados en el desarrollo de redes neuronales convolucionales, debido al uso de Tensor Flow, la herramienta de Google para generar modelos CNN de manera simple y sencilla.

A pesar de existir un particular interés en los científicos en utilizar herramientas computacionales para automatizar procesos de segmentación, el tiempo, los recursos, y la falta de conocimiento en el área informática suelen ser grandes muros para ellos, a partir de esto, Deep Cell 2.0 se desarrolló como una plataforma pensada para ellos, logrando un funcionamiento simple, e invisible para el científico, sin antes, realizar una configuración adecuada para cada problema correspondiente. Además, al tratarse de un programa con código abierto, es posible modificar funcionalidades, parámetros, dataset utilizados para el entrenamiento, entre otros.

Caltech, en una posterior revisión de su artículo, permitió a los desarrolladores no solamente usar un servidor remoto automatizado, sino la liberación de la biblioteca *deepcell-tf* como código abierto a través de *Github* (Bannon et al., 2018d). Esto permite a muchos desarrolladores adaptar el uso de la herramienta a distintos tipos de células, utilizando otros set de datos para entrenar una red neuronal convolucional, además de la posibilidad de modificar el funcionamiento de la librería de manera más simple y sencilla a través código, lo cual se debe principalmente a la integración de librerías como CUDA y *Jupyter Notebook* al contenedor Docker liberado al área científica.

La arquitectura de software de Deep Cell 2.0 busca el funcionamiento de una plataforma de *Cloud Computing simple*, que sea prácticamente invisible para el usuario, para esto, utiliza diversas tecnologías como Redis, Kubernetes, Tensorflow, entre otras. En la figura 2.14 es posible ver la arquitectura de Deep Cell 2.0 de manera gráfica, donde, en un inicio, se ingresa el set de imágenes a segmentar (también conocidas como imágenes de entrada) a la instancia creada en el navegador web (un ejemplo de esto se encuentra en su página web, [www.deepcell.org](http://www.deepcell.org)), para luego realizar una configuración del modelo CNN a utilizar, donde se pueden modificar variables como el funcionamiento general, parámetros, infraestructura, carga de trabajo, etcétera. Lo cual se realiza mediante un administrador de Kubernetes conocido como Kiosk. Esta configuración se mezcla con las imágenes en una base de datos (BD) disponible con el motor Redis. Luego, un procesador en la nube se comunica con un servicio de Tensorboard, para monitorear los procesos de entrenamiento, y, al mismo tiempo, con la BD, para administrar la carga de trabajo a los distintos nodos disponibles, que realizan la segmentación. El punto medio entre estos elementos es un módulo de auto escalamiento, que regula el funcionamiento de los equipos que realizan el entrenamiento del modelo, según la demanda del proceso.

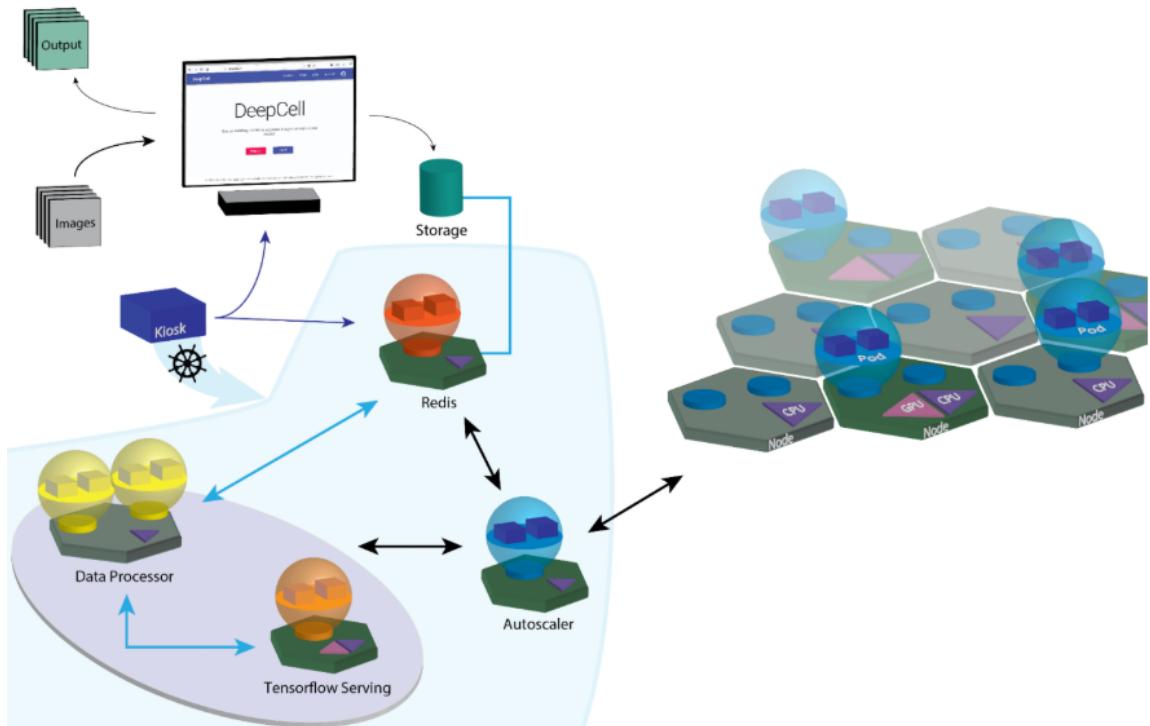


Figura 2.14: Arquitectura de Deep Cell 2.0. Las flechas azules representan los movimientos de datos, mientras que las flechas negras representan los movimientos de recursos computacionales.

**Fuente:** (Bannon et al., 2018)

La efectividad de la herramienta en su aplicabilidad se ha demostrado en otras investigaciones del área biológica, por ejemplo, la segmentación de organismos unicelulares, presentado en el artículo (Caicedo et al., 2019). La aplicación de métodos clásicos del análisis de datos, como la técnica “*Random Forest*”, ó herramientas similares en funcionamiento con Deep Cell 2.0, podemos ver que esta última se encuentra bien posicionada en la segmentación de núcleos de organismos unicelulares, presentando una baja cantidad de fallos y un buen rendimiento. No obstante lo anterior, una diferencia importante entre la presente investigación, y la segmentación de organismos unicelulares, es la naturaleza de las muestras, ya que estas últimas son imágenes fluorescentes, lo que significa que las imágenes tienen una mejor calidad en general, pero conlleva un costo monetario mayor en su obtención, debido a la infraestructura necesaria (microscopios, cámaras y sustancias) para obtener estas muestras.

El uso de la arquitectura Deep Cell 2.0 no garantiza que se entreguen resultados favorables en la segmentación, dependiendo de variables como el tipo de muestra, preprocesamiento y escalamiento de imágenes, número de muestras, entre otros. El artículo (Kromp et al., 2019) realizó una comparación entre las principales arquitecturas que realizan segmentación de imágenes usando técnicas de *deep learning*, entre ellas, Deep Cell 2.0, la investigación bus-

caba, principalmente, evaluar estas arquitecturas en relación a núcleos celulares que poseen propiedades que las diferenciaban de la mayoría, como lo es la incapacidad de modificar el fondo, un número reducido de muestras, o un mala calidad de la toma de muestras en general, lo que depende mucho del aparato científico utilizado para tomarlas. Para realizar la investigación, es necesario un exhaustivo trabajo de preprocesamiento de las muestras, con el fin de obtener un número estable para realizar tanto el entrenamiento, validación y pruebas de los modelos generados por las tecnologías, como también lograr un análisis en la mayor cantidad de casos posibles, incluyendo muestras válidas, como erróneas. Finalmente, los resultados de la investigación demuestran que Deep Cell es la peor evaluada entre las cuatro arquitecturas, llegando incluso a ser descartada para el uso de células fluorescentes de preparación simple, tal como se puede ver en la imagen 2.15.

Metric	U-Net silv/gold	U-Net ResNet silv/gold	MRCNN silv/gold	DeepCell silv/gold
US	<b>0.02/0.04</b>	0.03/0.16	<b>0.02/0.02</b>	0.36/0.42
REC	0.80/0.85	0.75/0.77	<b>0.90/0.87</b>	0.58/0.48
PREC	0.78/0.87	0.79/0.84	<b>0.81/0.91</b>	0.68/0.74
AJI	<b>0.71/0.75</b>	0.71/0.72	0.65/0.69	0.56/0.53

Figura 2.15: Resultados del artículo de Kromp, donde compara resultados del entrenamiento con dos dataset distintos. En negrita se puede ver el mejor resultado.

**Fuente:** (Kromp et al., 2019)

La liberación de la biblioteca *deepcell-tf* llevó la segmentación automatizada de organismos celulares a diversas áreas, por ejemplo, la detección de determinadas células vinculadas a la tuberculosis, y su posterior segmentación (McCaffrey et al., 2020). La tuberculosis es una enfermedad considerada grave, principalmente por la expansión descomunal por el mundo, siendo considerada una pandemia en la actualidad, que se agrava al juntarse con otra pandemia conocida, el VIH. Esta patología ha causado la muerte de 1.5 millones de personas en el mundo según estudios de la Organización Mundial de la Salud (WHO, 2020). El artículo expone la necesidad de reconocer y clasificar granulomas, las cuales se definen como masas o tumores destinados para aislar el actuar de una enfermedad (Steckelberg, 2019), en distintos tipos basado en cualidades como el tamaño, sustancias u otras células detectables, o la etnia de los enfermos por tuberculosis, entre otros elementos. Para lograr esta identificación, es necesario detectar y segmentar células unicelulares encontradas cerca de las granulomas, y los autores decidieron utilizar la herramienta Deep Cell para lograr este objetivo. El estudio concluye que ciertos grupos de personas poseen características especiales en su sistema inmune, particularmente en la sangre y los tejidos afectados, y esto solo se puede verificar luego de la inclusión de herramientas

de redes neuronales, con muestras médicas. (McCaffrey et al., 2020).

## 2.2.7 Métricas de evaluación de resultados

La utilización de métricas reconocidas en la segmentación de células permite medir los resultados realizados por herramientas computacionales, como también una comparación con los otros métodos y herramientas utilizadas en la segmentación de espermatozoides humanos. En una encuesta (Bajcsy et al., 2015) que compara distintos elementos y características de una segmentación automática en el área científica-biológica, como lo es el análisis de entrada, tipo de algoritmos utilizados y evaluación, menciona la importancia de utilizar métricas concisas y adecuadas para cada tipo de célula a segmentar, teniendo en consideración la investigación actual, como la futura. Bajo el contexto de la investigación en la segmentación automática de espermatozoides humanos, el artículo recomienda el uso de métricas que evalúen los fallos en número y posición entre la imagen segmentada de referencia (por un experto) y la imagen segmentada por la plataforma informática. Para esto, las medidas recomendadas son los índices de Dice y Jaccard, y el uso de la distancia de Hausdorff, respectivamente.

La principal función de los índices de Dice y Jaccard es calcular el nivel de error en el número de píxeles de la imagen segmentada con el método automático, en relación con la imagen de referencia. Ambos índices difieren mínimamente en la forma que realizan esta relación, y la elección de uno sobre otro se debe a las aplicaciones en que se ven. Por ejemplo, cuando se desean encontrar relaciones genéticas entre hebras de ADN, el coeficiente de Dice suele usarse más, debido a su equivalente con otro coeficiente que calcula similitud biológica específica en el área genética, el índice de Nei & Li (Applied Maths BioNumerics, 2019). Por contraparte, se ha demostrado que el coeficiente de Jaccard es una buena métrica para encontrar sinónimos o similitudes entre palabras claves en motores de búsqueda como Google o Bing, aportando en la creación de índices, u evaluando otros ya generados. Las ecuaciones específicas pueden verse en la imagen 2.16

$$Jaccard(R, S) = \frac{|R \cap S|}{|R \cup S|}$$

$$Dice(R, S) = \frac{2|R \cap S|}{|R| + |S|}$$

Figura 2.16: Ecuaciones de Jaccard y Dice. Donde R corresponde al conjunto de píxeles de la imagen de referencia, y S al conjunto de píxeles de la imagen obtenida por el algoritmo o herramienta informática.

**Fuente:** (Bajcsy et al., 2015)

Pero el cálculo del número no es suficiente para garantizar que una segmentación es de buena calidad, por lo que el cálculo de las posiciones es importante en la segmentación de cualquier célula, para medir esto, se utiliza la métrica de la distancia de Hausdorff. Por definición, corresponde a “el máximo de los conjuntos de distancias mínimas entre las formas de dos imágenes”, en este caso, la imagen resultante del modelo y la imagen de referencia, lo cual puede verse en la figura 2.17. En la segmentación automática de células, moléculas y, particularmente, en la segmentación automática de partes de espermatozoides humanos mediante técnicas de inteligencia artificial, esta métrica se utiliza mucho, debido a su simpleza, eficiencia, y la posibilidad de realizar una comparación con las investigaciones anteriores, como para entregas posteriores (Bajcsy et al., 2015).

$$D(A, B) = \min_{a \in A} \left\{ \min_{b \in B} \{d(a, b)\} \right\}$$

Figura 2.17: Ecuación de distancia de Hausdorff. Se lee “para cada punto a de A, encuentre su distancia más pequeña a cualquier punto b de B; finalmente, mantenga la menor distancia encontrada entre todos los puntos a”

**Fuente:** (Grégoire & Bouillot, 1998)

## 2.2.8 Validación Cruzada

Con el fin de evitar el sobreajuste, otra técnica conocida en el área de la inteligencia artificial y particularmente las redes neuronales, es el procedimiento de validación cruzada, que se define como “una técnica para evaluar modelos de *machine learning* mediante el entrenamiento de varios modelos de *machine learning* en subconjuntos de los datos de entrada disponibles y evaluarlos con el subconjunto complementario de los datos.” (AWS, 2016). En palabras simples, consiste en separar los datos de entrada en datos de entrenamiento y datos de evaluación, y realizar un entrenamiento en K iteraciones, cambiando la proporción de datos de entrada. Por ejemplo, en la figura 2.16, se puede ver un proceso de validación cruzada de 4 iteraciones, donde cada rectángulo corresponde a la totalidad de los datos de entrada, y se subdivide en 25% de datos de evaluación, y 75% de datos de entrenamiento, y con estos datos, se realiza el entrenamiento. Al finalizar, se realiza el entrenamiento sobre el mismo modelo anterior, pero cambiando los datos de entrenamiento y evaluación. Esto se realiza un número K de iteraciones, con el fin de que el modelo converja, y no posea sobreajuste.

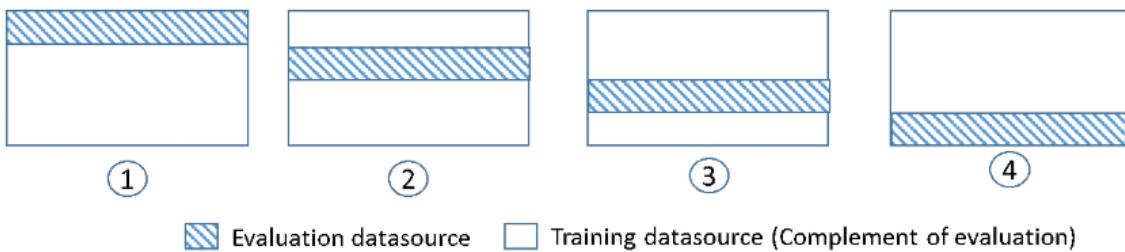


Figura 2.18: Ejemplo de *4-fold cross validation*, o validación cruzada de 4 intervalos. El segmento azul representa el conjunto de validación y el blanco el conjunto de entrenamiento por cada iteración de entrenamiento.

Fuente: (AWS, 2016)

## CAPÍTULO 3. ESTADO DEL ARTE

### 3.1 GOLD STANDARD

La detección y segmentación de espermatozoides, principalmente de la cabeza del espermatozoide, es un proceso esencial a la hora de realizar un correcto análisis de semen, y así la condición del paciente. La manera tradicional es mediante la ayuda de un especialista que conozca la estructura de un espermatozoide tradicional como se puede visualizar en la figura 3.1. Sin embargo, realizar esto de manera manual tiene diversos problemas, como el error humano, tiempo utilizado, recursos asociados, y sobretodo, la subjetividad de la segmentación, debido a que si otro especialista realiza el mismo proceso, es probable que el resultado sea diferente.

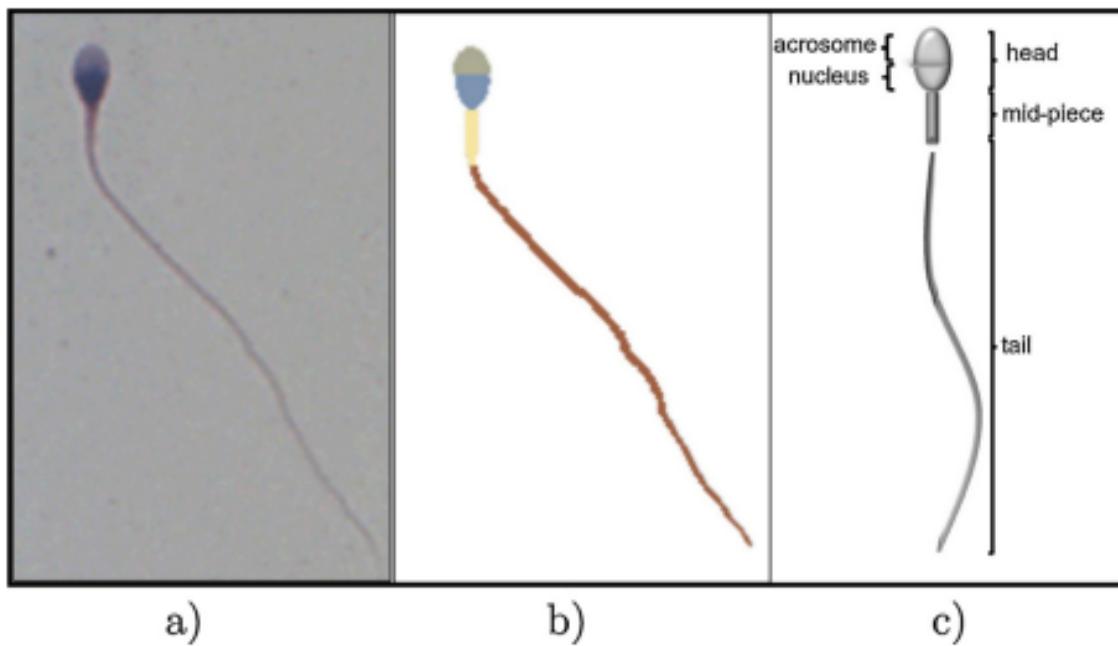


Figura 3.1: Diferencias entre muestra real, segmentada, y teórica. En la parte a), es posible apreciar una imagen de muestra tomada desde un microscopio, luego, en la sección b), se observa una segmentación manual realizada por un experto, trazando por encima de la imagen, siguiendo las normas establecidas en la teoría, graficadas en la parte c).

Fuente: (Chang et al., 2014)

Una de las soluciones es utilizar herramientas computacionales, que automaticen procesos utilizando técnicas del procesamiento de imágenes. Estos métodos se basan en diversos artículos, por ejemplo, el planteado por Henry Carrillo (Carrillo et al., 2007) que menciona diferencias importantes entre ellos, lo que genera el mismo problema que los procedimientos tradicionales, la subjetividad. Sin embargo, un artículo reciente busca resolver estos problemas,

ya que define de manera simple un *Gold Standard* (Véase Glosario), para luego, a partir de esta prueba, crear un método adecuado para la segmentación de las imágenes (Chang et al., 2014).

El método planteado se basa en dos partes, la primera corresponde a la detección de la imagen, la cual mediante modificaciones de los espacios de colores logra distinguir entre un espermatozoide y el fondo de cada imagen. La segunda al método de agrupamiento *k-means* o k-medias (Dhanachandra et al., 2015), permite detectar en las imágenes los espermatozoides correspondientes. El resultado de este proceso puede verse en la imagen 3.2.

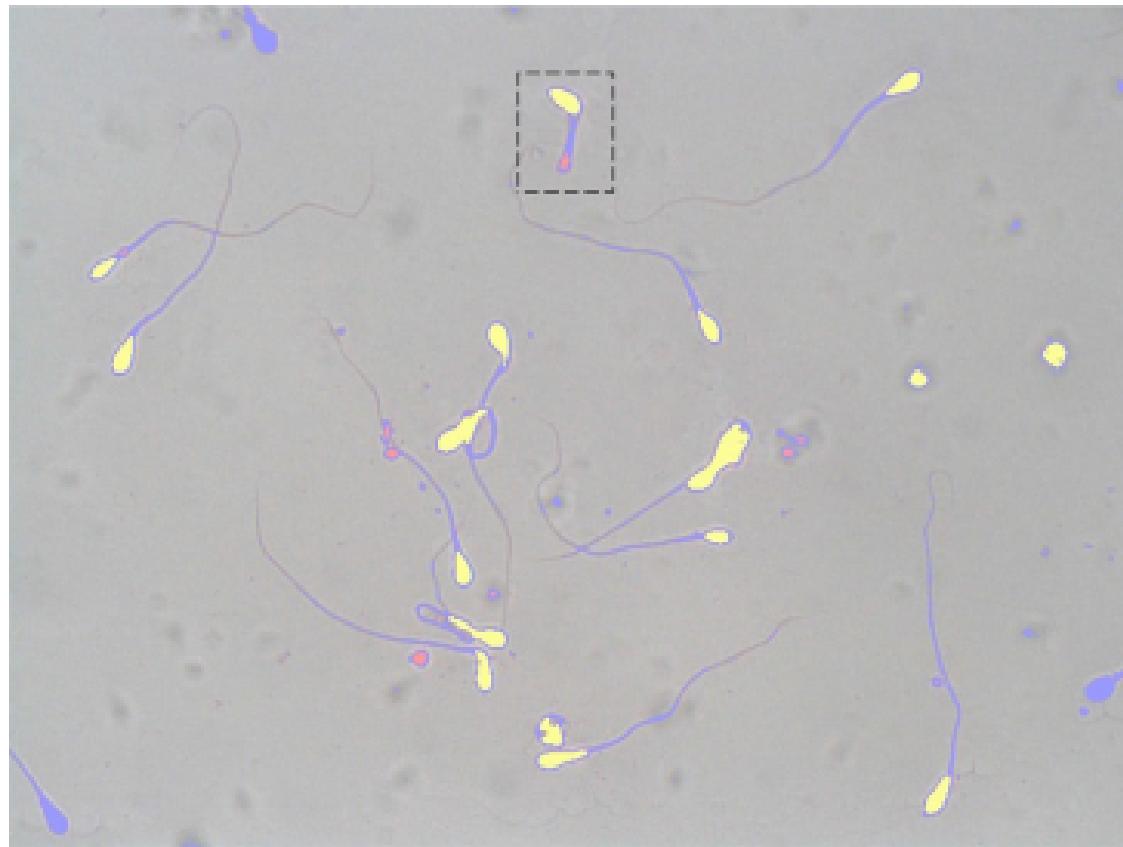


Figura 3.2: Ejemplo de segmentación utilizada usando k-means. En color amarillo se ve la segmentación de cabezas y pieza media luego del cambio en el espacio de color y el uso del método de agrupamiento k-medias.

**Fuente:** (Chang et al., 2014)

Uno de los elementos más importantes de los espermatozoides es su cabeza, el cual permite realizar un análisis más certero sobre las anomalías, de ahí la importancia del proceso de segmentación, por lo que el método procede a eliminar tanto las colas, como los bordes de las células, con el fin de dejar solo la cabeza del esperma para su segmentación. Utilizando la base del primer algoritmo con el método de *K-means*, y utilizando los píxeles determinados de la imagen, permite retornar todas las imágenes de cabezas de espermatozoides de manera limpia,

que permitan identificar su forma, tipo o característica determinada. En la figura 3.3 puede verse todo el proceso del algoritmo.

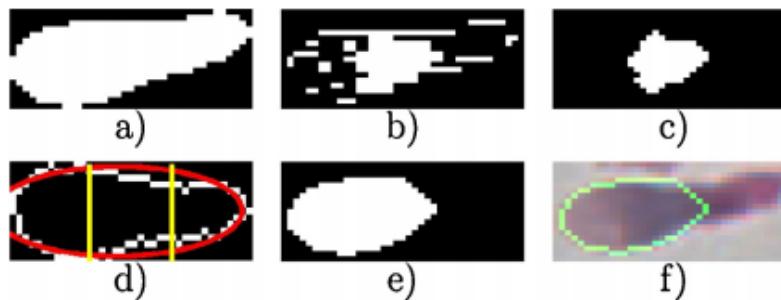


Figura 3.3: Procedimiento de algoritmo de segmentación de cabezas de espermatozoides. Proceso del algoritmo de segmentación de cabeza utilizando *k-means* y *clustering*. A partir de la cabeza obtenida en el primer algoritmo (paso a)), pasando por proceso de *k-means* y *clustering* (b, c) y d)) para finalmente retornar la cabeza de la mejor manera (imagen e)) sobretodo comparando con la imagen original (sección f) de la figura 2.14)

**Fuente:** (Chang et al., 2014)

El método innova con otros procedimientos, que retornan la dirección y ángulos de la cabeza, además de una segmentación adicional de la cabeza con la finalidad de encontrar el acrosoma y el núcleo, lo cual es de aporta a la hora de detectar anomalías o malformaciones, y así, comparar en la investigación sobre este tema. Una muestra de este último proceso se puede ver en la figura 3.4.

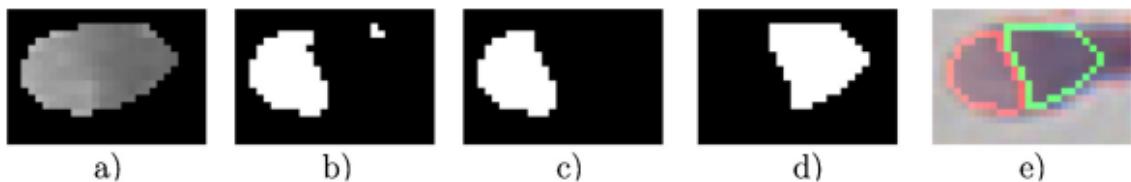


Figura 3.4: Segmentación particular de cabezas de espermatozoides: acrosoma y núcleo. a) Resultado del algoritmo de segmentación de cabezas con *clustering* y *k-medias*. b) Limpieza e binarización de la imagen y detección de ángulo y dirección de la cabeza del espermatozoide. c) Separación de acrosoma. d) Separación de núcleo)

**Fuente:** (Chang et al., 2014)

El método presentado por (Chang et al., 2014), en conjunto con el *gold standard*, que determina métricas determinadas como el coeficiente de Dice, entrega mejores resultados que otros métodos planteados, como el método de Carrillo, publicado 7 años antes, logrando un importante avance en el área.

La creación del *gold standard* presentado por (Chang et al., 2014) permitió a los investigadores del área tener una base para poder comparar sus métodos de segmentación, y así, realizar mejores métodos incluyendo diversas tecnologías, como es el caso de la inteligencia

artificial. Así mismo, también surgieron diversas plataformas computacionales que buscan facilitar el trabajo de segmentación a personas interesadas, pero que no poseen ni el conocimiento, ni los recursos para aplicar estos conceptos novedosos, tal como es el caso, por ejemplo, de plataformas como U-net (Ronneberger et al., 2015) o Deep Cell 2.0. (Bannon et al., 2018)

### **3.2 PROCESAMIENTO DE IMÁGENES Y SEGMENTACIÓN AUTOMÁTICA DE PARTES DE ESPERMATOZOIDES**

En el capítulo anterior, fue posible visualizar un ejemplo de uso de técnicas de análisis de datos para poder realizar una segmentación automática de espermatozoides, en donde se menciona el método Chang (Chang et al., 2014). Sin embargo, han existido diversas otras técnicas y métodos para realizar esta labor, y para expandir el espectro de soluciones posibles, se incluyen nuevas herramientas computacionales, como lo son la inteligencia artificial, y particularmente, el *deep learning*.

La autora Fariba Shaker, tomando en consideración la investigación y el gold standard presentados en el artículo (Chang et al., 2014), entrega una alternativa para la detección y segmentación automática de espermatozoides humanos, basado en la detección de bordes, uso de vectores de gradiente y algoritmos de subdivisión de la muestra (Shaker et al., 2016). En un inicio, luego de transformar la imagen de la muestra a una escala de grises, y analizar su histograma (distribución de tonos en la imagen), es posible detectar fácilmente los espermatozoides humanos de la muestra, sin embargo, no es posible segmentar las partes del espermio. Para esto, extrae los bordes de la imagen utilizando un filtro gaussiano y el método de Canny, con el fin de encontrar los contornos activos de la muestra, utilizando el método de GVF Snake, o flujo vectorial de gradiente (Cartas-Ayala, 2011). A partir de esta transformación es posible encontrar tanto la cabeza, como la parte media del espermio, separando la cola, tal como se puede ver en la figura 3.5. A partir de esto, es necesario encontrar las dos partes que componen la cabeza, el acrosoma y el núcleo, aquí entran algoritmos desarrollados por los autores, que permiten localizar y remover la parte media, y poder generar la segmentación de la cabeza de manera correcta, mediante la manipulación de los píxeles de la imagen.

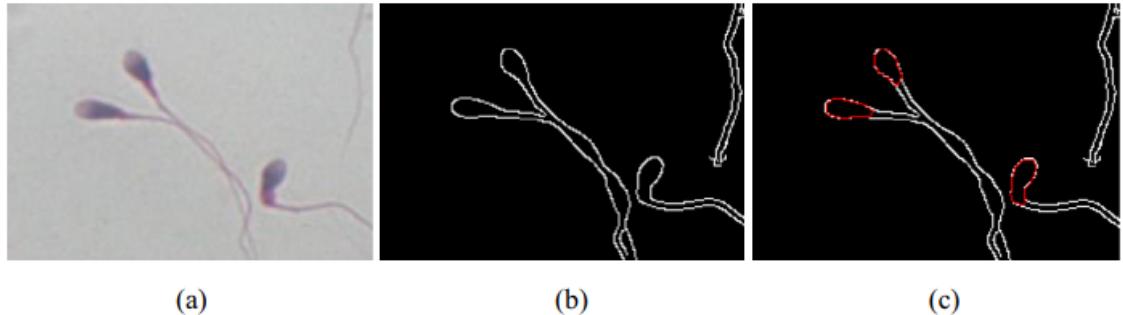


Figura 3.5: Segmentación de cabezas usando el método Shaker. (a) Imagen original. (b) Detección de bordes usando Canny. (c) Segmentación de cabezas y pieza media con GVF.

**Fuente:** (Shaker et al., 2016)

La explosión de la inteligencia artificial también se demuestra en la segmentación automática de espermatozoides, generando modelos de redes neuronales convolucionales profundas para extracción de características y, posteriormente, una segmentación automatizada a partir del entrenamiento de la red. Nissen y otros autores (Nissen et al., 2017) hablan sobre la segmentación de espermios humanos a través de distintas redes convolucionales, demostrando que es un beneficio en comparación con otros métodos convencionales (como son el trabajo y preprocesamiento de imágenes y técnicas de *clustering*). Se utilizaron diversos tipos de CNN, que se diferenciaban en el número de operaciones de convolución utilizadas, el método de *pooling*, u otras características (el modelo descrito por Nissen se puede ver en la figura 3.6). Además, detalla la importancia de expandir el dataset de imágenes para realizar un entrenamiento correcto de las CNN, realizando modificaciones a las imágenes. Este artículo concluye la importancia que puede tener el uso de la inteligencia artificial y el *deep learning* en la segmentación automática de espermatozoides humanos.

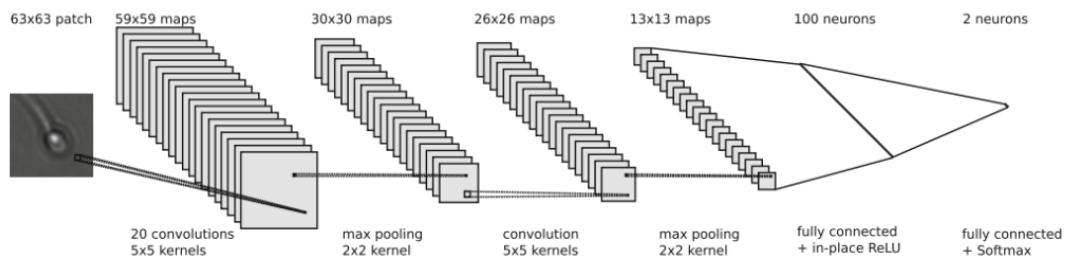


Figura 3.6: CNN utilizada por Nissen para evaluar la segmentación de espermatozoides humanos.

**Fuente:** (Nissen et al., 2017)

Siguiendo la aproximación del artículo anterior, el uso de técnicas de inteligencia

artificial y las CNN ha determinado avances en comparación con otros tipos de técnicas, como lo son los clasificadores del análisis de datos, y técnicas de imágenes. A partir de esto, el uso de técnicas de *deep learning* es una alternativa para comparar resultados con estas técnicas, aportando en aspectos como la eficiencia, el costo computacional, y la certeza en la segmentación. Así mismo, en el artículo (Movahed et al., 2019), se presenta un método basado en la aplicación de técnicas de *deep learning* para la segmentación automática de espermatozoides humanos, el cual se basa en cuatro etapas principales. El primero consta del preprocesamiento de las imágenes, basado en subelementos principales, filtros homomórficos que logran una corrección de iluminación de la imagen, atenuadores de fondo para mejorar la calidad para las redes neuronales y un estiramiento del histograma, logrando una mejoramiento en el contraste de la imagen. A partir de este punto, se inicia la segmentación subdividida en cada una de las partes. La primera, definida como la segmentación de partes externas, utiliza dos CNN, una centrada en la cabeza unida a la parte media, y la siguiente en el filamento axial u cola del esperma, estas dos redes poseen distintas arquitecturas, como también funciones de activación y resultados. Con esta separación lista, es posible diferenciar las cuatro partes principales, la cual se desarrolla en las últimas dos etapas, para segmentar la cabeza, identificando el acrosoma y el núcleo, se utiliza un *clustering* de k-medias, similar al método Chang (Chang et al., 2014). En cambio, para la segmentación de la cola y la parte media existe un procedimiento más complejo, a partir de las imágenes preprocesadas, se generan nuevas imágenes para iniciar una red neuronal de clasificación. Para su funcionamiento, se extraen una serie de características de estas imágenes, a saber, el contraste, la entropía, la varianza, entre otras, que sirven para entrenar la red y entregar resultados de los nuevos datos, conocidos como las características de Haralick. Un resumen de este método puede verse en la figura 3.7. Esta publicación corresponde a la última en la segmentación de espermatozoides humanos.

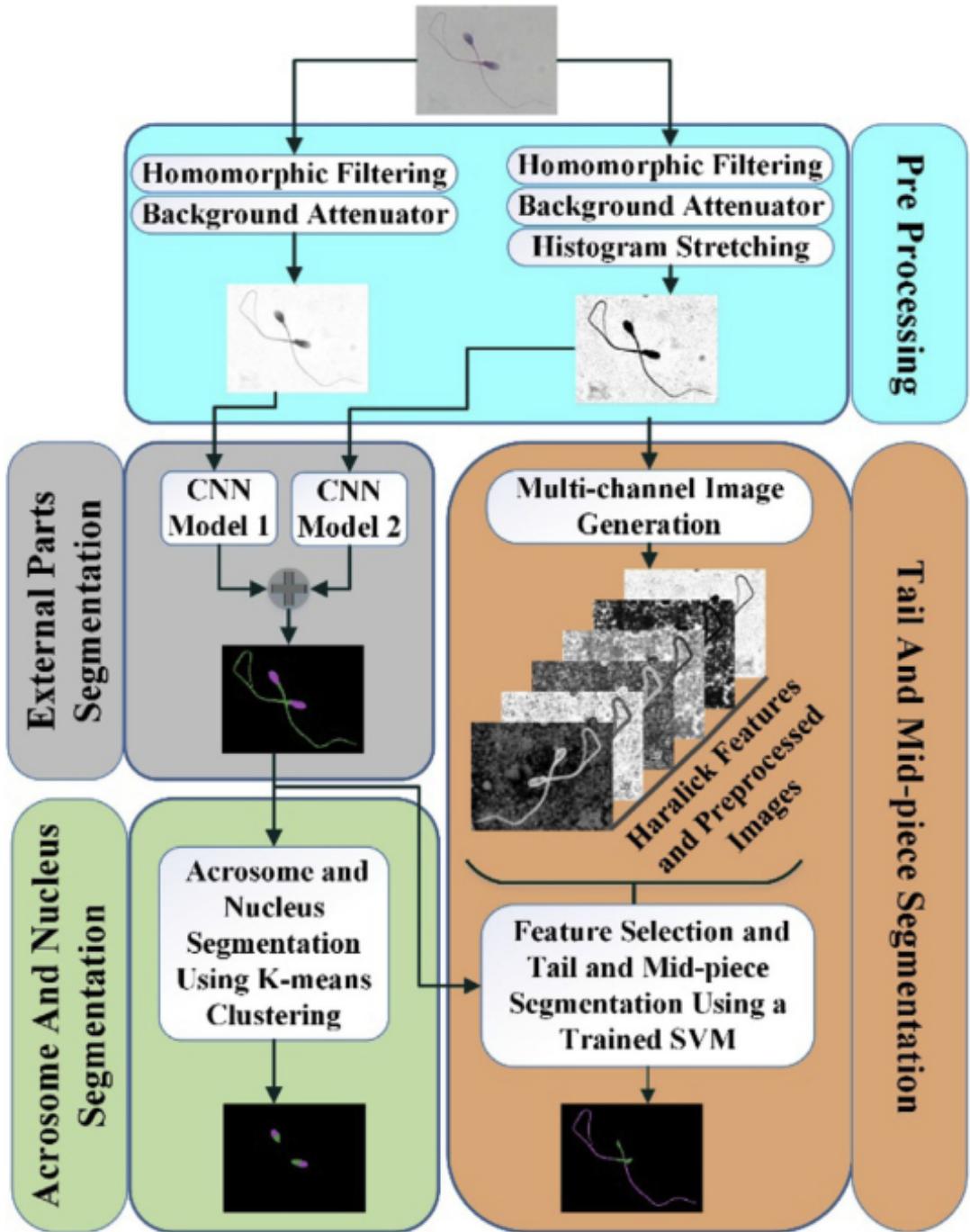


Figura 3.7: Representación gráfica del método Movahed. Se distinguen 4 secciones definidas en el método Movahed, el preprocesamiento de las imágenes tanto para las redes convolucionales como la red neuronal de clasificación (sección celeste). Con dos redes convolucionales, se realiza la segmentación de la cabeza, en relación a la parte axial (pieza media y cola) (sección gris). Con esta segmentación inicial, es posible realizar la segmentación de núcleo y acrosoma con técnicas de *clustering* y *k-means* (sección verde) y la segmentación de la cola y la pieza media con una red neuronal de clasificación pre entrenada.

Fuente: (Movahed et al., 2019)

Para poder medir los resultados de estas publicaciones y avances en el área, es importante tener un patrón, regla o protocolo para comparar los distintos métodos, a partir de esto, se utiliza un *gold standard* (ver Glosario). Este *gold standard* para la segmentación automática de espermatozoides, propuesto en el artículo (Chang et al., 2014), posee información sobre la obtención y preparación de las muestras a utilizar, como también el número de espermatozoides en las imágenes, y define métricas esenciales para la comparación de métodos. Además, posee la segmentación manual o de referencia realizada por especialistas, que sirve para comparar cualquier método de segmentación automática, con una segmentación tradicional. La principal labor del *gold standard* es lograr que las distintas técnicas publicadas sobre la segmentación automática de espermatozoides sean evaluados por el mismo conjunto de datos, y así, generar una comparación certera y eficaz entre los distintos métodos, sin importar cuál es su funcionamiento o naturaleza.

## CAPÍTULO 4. DISEÑO E IMPLEMENTACIÓN

Para poder cumplir el objetivo principal propuesto para este trabajo de titulación, es necesario realizar una serie de acciones relacionadas a la implementación de software necesario, integración de herramientas como Deep Cell, y un amplio proceso de análisis. Este capítulo busca expandir sobre el diseño y el proceso de implementación de estas acciones.

### 4.1 DESCRIPCIÓN DE LOS DATOS

El dataset de imágenes a utilizar se obtiene de SCIAN-SpermSegGS (Chang et al., 2014). Las muestras fueron obtenidas por el Laboratorio de Espermiograma del Instituto de Ciencias Biomédicas (ICBM) de la Universidad de Chile, como también del Instituto de Investigaciones Materno - Infantil (IDIMI) del Hospital Clínico San Borja Arriarán, ambos ubicados en Santiago de Chile. El dataset posee 19 imágenes con 264 espermatozoides, de los cuales existen 210 que son válidos. Las imágenes poseen una resolución de 780 x 580 píxeles, y cada una de estas imágenes posee una máscara de referencia realizada por un especialista por cada parte, es decir, un imagen RGB con un número específico de espermatozoides posee 5 imágenes “máscaras”, donde se visualiza la segmentación de cada una de sus partes: la segmentación de referencia de las cabezas, núcleos, acrosomas, partes medias y colas de los espermatozoides en esta imagen. Todas estas imágenes pasarán por un proceso de aumento de datos, generando nuevas muestras con sus máscaras correspondientes. Cabe destacar que cualquier imagen resultante del proceso de aumento de imágenes, debe poseer la misma resolución que el dataset propuesto, como también no tener modificaciones importantes de la original, que puedan entorpecer el proceso de entrenamiento del modelo.

Existe una gran cantidad de datasets públicos con imágenes médicas y sus correspondientes máscaras destinadas para segmentación en el ambiente médico, sin embargo, es necesario realizar un análisis de cada uno de estos para determinar cual puede aportar a la segmentación utilizando *transfer learning* al dataset SCIAN-SpermSegGS.

Hela\_S3 es el dataset definido para realizar *transfer learning* en modelos Panoptic e RetinaNet. Obtenido a partir de una investigación de la universidad de Colorado, y se describe como “muy útil en el análisis clonal de las poblaciones de células de mamíferos” (Muyano, 2011), y consiste en células humanas principalmente, con una forma circular en su mayoría. Es además “el dataset de células humanas permanente más antigua y más ampliamente distribuida en el área médica” (Rahbari et al., 2009). La versión pública de este dataset se encuentra integrada en Deep Cell 2.0, y posee cerca de 7000 imágenes, con sus respectivas máscaras para la

segmentación. Los organismos celulares de este dataset poseen una forma similar a las cabezas de los espermatozoides humanos, siendo esta la principal característica para escoger este set de datos. Al comparar con otros datos accesibles con la herramienta Deep Cell, como los tejidos de ratones albinos del dataset NIH\_3T3 que poseen una forma similar a una neurona humana, o el set de datos HEK293, correspondiente a células embrionarias de riñón, con forma de hoja de cardo corredor. Estos datasets poseen un menor número de células que Hela\_S3, y su forma no se asemeja a las celulas presentadas por el dataset SCIAN-SpermSegGS, lo cual puede afectar negativamente al proceso de *transfer learning*. Cabe destacar que este dataset contiene imágenes tomadas en fluorescencia, y, a pesar de los resultados expuestos por una investigación anterior (Kromp et al., 2019), no existe ninguna indicación de los creadores de DeepCell que desacredita el uso de la herramienta con este tipo de imágenes (Bannon et al., 2018).

## 4.2 AMBIENTE EXPERIMENTAL

El ambiente experimental permite conocer la forma en que se realiza el experimento para aceptar o refutar la hipótesis planteada, considerando variables como el procedimiento, software y hardware y los datos. Tal como indica la metodología basada en objetivos (véase capítulo 1), se definen 3 pasos principales, la obtención y aumento de imágenes para entrenar el modelo, la adopción del software Deep Cell 2.0 para generar el modelo, y su posterior prueba y análisis.

Antes de iniciar la explicación detallada de las tareas definidas en la tabla 1.1, es necesario entender el ambiente de trabajo en relación al uso de directorios, archivos, y carpetas. En este ámbito, se puede diferenciar tres grupos principales, el almacenado en Github, un grupo de carpetas guardados localmente, y otros conectados con la herramienta Deep Cell. En relación al primer grupo, está compuesto por la carpeta src, donde se almacenan *scripts* creados para iniciar el proceso de aumentación (incremento de imágenes), separador de imágenes en cantidades definidas, el comprimidor de datasets, y el encargado de realizar de medición para evaluar resultados. Se almacenan localmente las imágenes, que por su tamaño y confidencialidad, no pueden alojarse en un repositorio de datos. Podemos ver dos carpetas principales img y img\_aug, en las cuales se encuentran las imágenes del *gold standard* y las imágenes generadas luego del proceso de incremento y separación de imágenes, respectivamente. Finalmente, el último grupo de archivos también se almacena en Github, pero su principal objetivo es ser el intermediario entre la máquina local, y la máquina donde se encuentra instalado Deep Cell, en el directorio data se encuentran los datos de entrada (imágenes aumentadas) y el modelo generado por el entrenamiento (red neuronal) y en la carpeta

scripts, se almacena todo el código correspondiente al funcionamiento de la librería *deepcell-tf*, incluido el proceso de integración del dataset (almacenado en la carpeta data) y el proceso de entrenamiento. El detalle gráfico se puede ver en la figura 4.1.

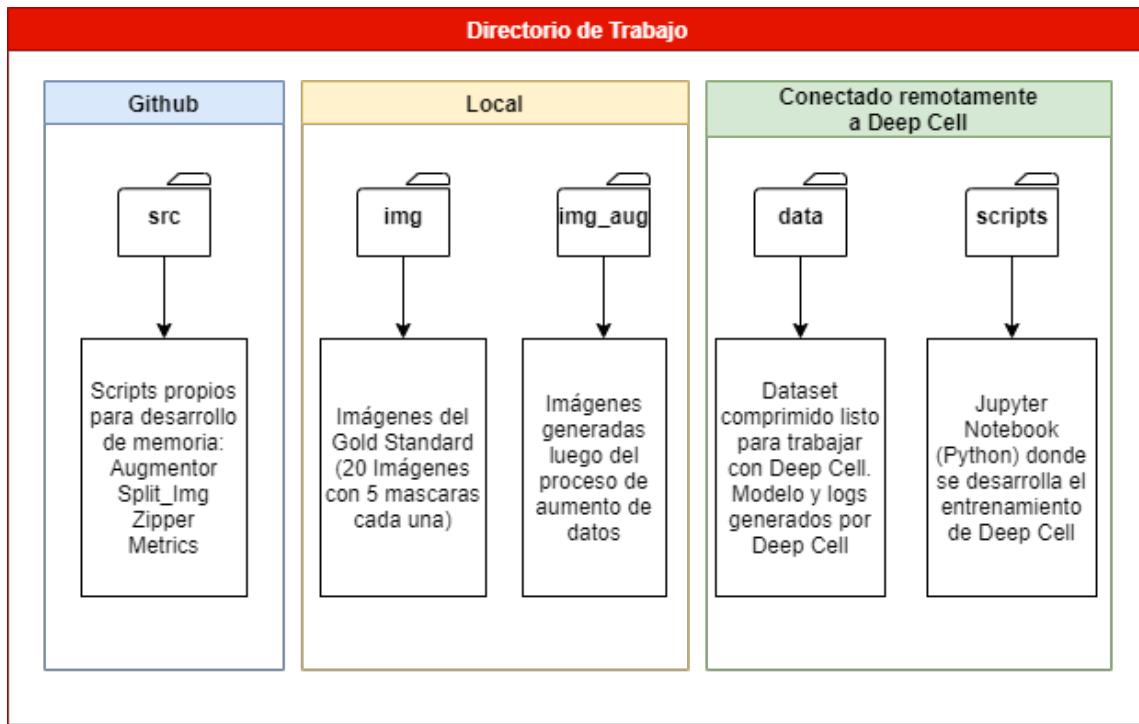


Figura 4.1: Directorio de Trabajo. Existe una separación de 3 principales directorios: Scripts propios para el apoyo del trabajo de investigación (sección azul). Archivos locales, específicamente las imágenes utilizadas para entrenar el modelo y Directorios de Deep Cell montado a Docker, que permiten conectar el ordenador utilizado con el contenedor que posee la herramienta.

**Fuente:** Elaboración propia, 2020

La estructura de trabajo, en general, se evalúa en la figura 4.2, donde se puede diferenciar las herramientas necesarias para realizar el trabajo y los distintos ambientes de trabajo generado. Existe una clara división entre el ordenador personal, llamado “de baja carga computacional”, y un servidor remoto, llamado “de alta carga computacional”, los cuales comparten un carpeta donde se almacena la red neuronal y el código correspondiente a su entrenamiento y el dataset de imágenes. El ordenador personal usa códigos del lenguaje Python, principalmente apoyados por librerías como **Albumentations**, **Splits\_Folders**, entre otros. También, a modo de prueba, se realiza la modificación del código de las carpetas compartidas desde el ordenador de baja carga, utilizando una imagen de Docker, apoyada por la GPU del ordenador de baja carga computacional, para no utilizar de sobremanera el servidor remoto. Debido a las especificaciones técnicas, el entrenamiento de la red neuronal se realiza en el servidor remoto de alta carga computacional, apoyado por una GPU, CUDA y Docker-Nvidia. Al

iniciar el contenedor de Docker, se abre de manera local un servicio de *Jupyter Notebook* donde se puede modificar el código e iniciar el entrenamiento de la red neuronal, almacenándose en la carpeta compartida.

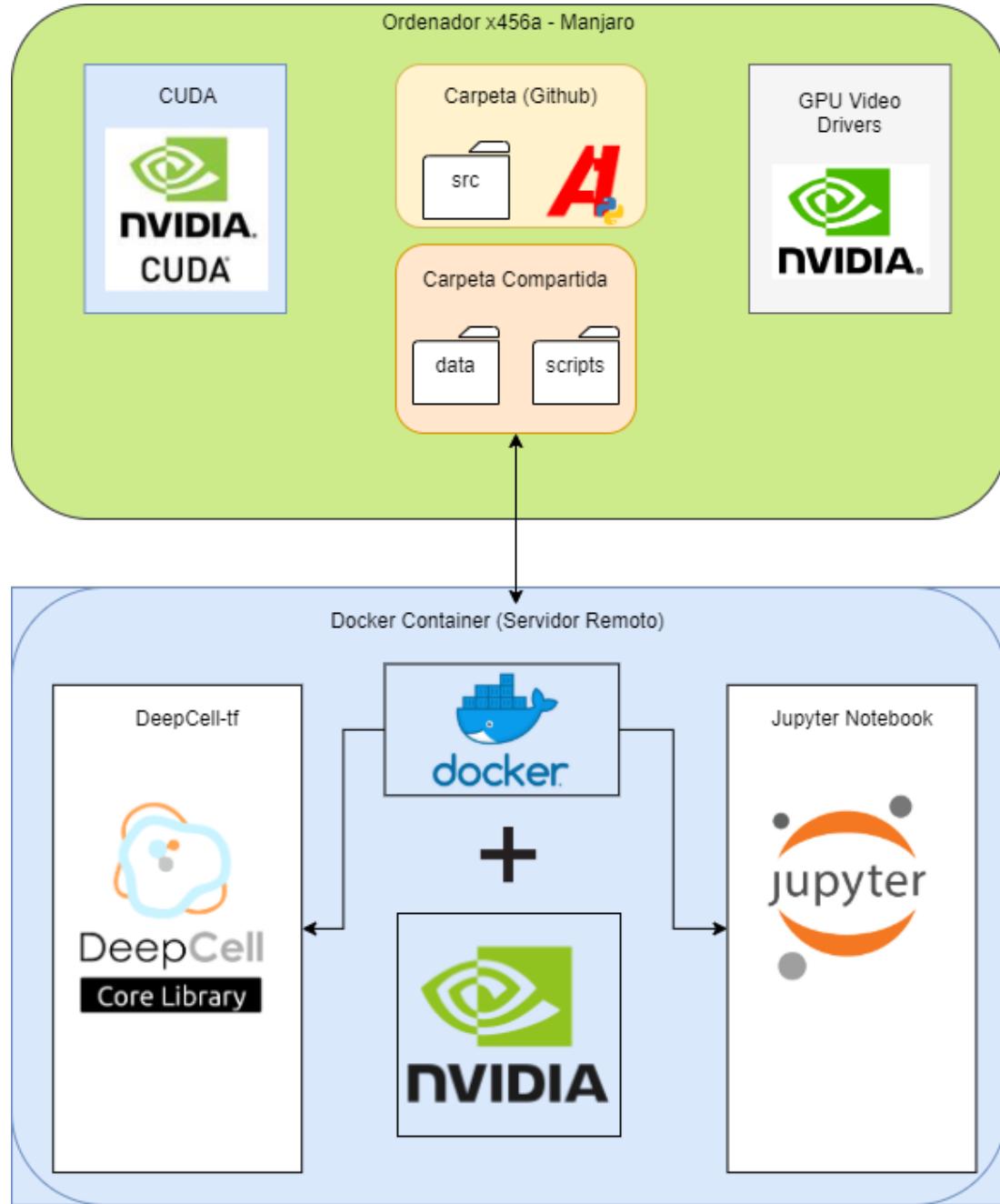


Figura 4.2: Estructura/Ambiente de Trabajo. Mientras que el ordenador personal (Tabla 1.2) se encarga de la creación de scripts propios de código, en el servidor remoto (Tabla 1.3) se almacena Docker y los drivers necesarios para la utilización de la GPU como principal motor para el entrenamiento. La biblioteca deepcell-tf se encuentra en la imagen de Docker **vanhalen:deepcell**, y posee una instancia de *Jupyter Notebook* para facilitar el trabajo

**Fuente:** Elaboración propia, 2020

## 4.3 PROCEDIMIENTO EXPERIMENTAL: METODOLOGÍA BASADO EN OBJETIVOS

A continuación, se presenta en detalle el trabajo realizado a través de las distintas tareas realizadas durante el desarrollo de la investigación, según lo informado en la tabla 1.1.

### 4.3.1 Aumento de Datos

El aumento de datos es una fase fundamental del trabajo de cualquier red neuronal, y esto adquiere una dificultad mayor con el trabajo de las redes convolucionales, debido al tipo de datos utilizados, imágenes. En el contexto de la segmentación automatizada de espermatozoides, es necesario realizar un aumento de imágenes consecuente con las muestras médicas correspondientes, teniendo mucha atención con las imágenes generadas.

A modo preliminar, uno de los principales elementos es realizar la separación y organización de imágenes, para esto, se realiza un proceso de separación de archivos automática, en donde las 19 imágenes originales (en conjunto con sus máscaras) del *Gold Standard* se dividieron en dos categorías, entrenamiento (*train*) y evaluación (*test*), en porcentajes definidos de 80% y 20%, respectivamente. Este proceso se realiza con el objetivo de analizar utilizando datos reales, y no generados artificialmente para evaluar el funcionamiento de la red neuronal entrenada. Este trabajo se realiza, a nivel de código, por un *script* propio con el uso de librería **split\_folders**, con licencia MIT<sup>1</sup>, que genera la separación incluyendo todas las máscaras de la imagen, manteniendo el nombre del archivo, además de definir una semilla para siempre realizar la misma división al cambiar de hardware o software.

El proceso de incremento de imágenes se ha documentado a lo largo de los años, como principal solución al concepto de *Overfitting* (Véase capítulo 2). Además, los diversos avances en el área de la visión han permitido realizar, fácilmente, modificaciones del tipo geométrico, como fotométrico. Tal como se mencionó en la sección “descripción de los datos”, el número de imágenes del dataset SCIAN-SpermSegGS es de 19 imágenes, contabilizando cerca de 200 espermatozoides válidos. En un inicio, esta modificación se planeaba realizar mediante *scripts* propios; empleando técnicas simples de transformación geométrica, particularmente, rotación, traslación, reflejo, entre otras. Sin embargo, se generaron problemas asociados a estas transformaciones, por ejemplo, qué hacer con los espacios generados al realizar rotaciones, la necesidad de mantener el mismo tamaño de imágenes, lograr una transición sin pérdidas

---

<sup>1</sup><https://pypi.org/project/split-folders/> - <https://github.com/jfilter/split-folders>

de células, entre otros. Frente a estos problemas, la posibilidad de utilizar una biblioteca de programación especializada en el aumento de imágenes, obtenible a partir de paquetes de biblioteca *pypi*, empezó a tomar fuerza. La primera opción corresponde a *Keras\_preprocessing*, una de las más utilizadas para tareas de clasificación en redes neuronales con *TensorFlow*. Sin embargo, esta librería no posee un método para aumentar imágenes con múltiples máscaras, y, al poseer una semilla dinámica con cada uso de la función, es imposible generar una consistencia entre las imágenes generadas y sus máscaras. Por otra parte, *ImgAug* es una de las librerías más utilizadas para el aumento de imágenes, utilizada por ser la principal base de otras librerías, sumada a su simpleza, sin embargo, la poca documentación, en relación al aumento de imágenes con múltiples máscaras, fue razón suficiente para escoger otra opción.

La página **Kaggle**, reconocida por ser proveedora de sets de datos públicos para ámbitos relacionados a la estadística y la inteligencia artificial, realiza diversos concursos para evaluar nuevas librerías de programación en estas áreas. Dentro del ámbito de aumento de imágenes, dos librerías destacan sobre el resto (Véase Tabla A.3). **Augmentor** y **Albumentations**, la primera, recomendada para generar imágenes con múltiples capas, es una de las más utilizadas por los usuarios de la página web, debido a su facilidad. El funcionamiento de *Augmentor* en el dataset SCIAN-SpermSegGS es correcto, sin embargo, el número de transformaciones es limitada, ya que solo incluye transformaciones geométricas (hasta abril del 2020), y debido a un bug en su funcionamiento, la aplicación de transformaciones a múltiples máscaras solo funciona hasta 4 máscaras, una menos en relación al dataset ya mencionado, como también a la teoría de las células a segmentar.

A partir de esto, se decidió escoger la biblioteca **Albumentations** (Buslaev et al., 2020) para ser la base de un script propio para el proceso de aumento de datos. El uso de esta plataforma se debe a los múltiples reconocimientos recibidos durante distintas competiciones<sup>2</sup>, su rapidez en funcionamiento y la opción “*multiple\_targets*”, que permite generar una nueva imagen y aplicar las mismas transformaciones a más de una máscara sin complicaciones. El procedimiento específico es el siguiente: inicia con una lectura y posterior almacenamiento en variables de las imágenes del dataset SCIAN-SpermSegGS en formato *numpy array*, luego, se definen las transformaciones a aplicar y se inicia el proceso de incremento de datos. Debido a la necesidad de generar imágenes fehacientes con una muestra real como las expuestas en el *gold standard*, y basado en las últimas publicaciones en relación al aumento de datos para redes convolucionales profundas (Shorten & Khoshgoftaar, 2019), se escoge una serie de técnicas básicas no invasivas, basadas principalmente en transformaciones geométricas simples (rotación, traslación, espejo), cambio de luminosidad (contraste, gamma, brillo), de tonalidad (Modificación de componentes RGB, transformación a gris o sepia), y una transformación propia llamada rotación sin expansión,

---

<sup>2</sup>[https://github.com/alembert/Albumentations/blob/master/docs/hall\\_of\\_fame.rst](https://github.com/alembert/Albumentations/blob/master/docs/hall_of_fame.rst)

que consiste en la rotación en ángulos de 5°, 15° y 25°, usando como pivote la esquina inferior izquierda, sin modificar el tamaño de la imagen. Luego, los espacios generados adquieren el tono de gris más común en la imagen original. Todas las transformaciones se aplican en base a una posibilidad y/o una condición, para garantizar un número importante de imágenes generadas distintas entre sí. Las transformaciones utilizadas, su descripción, condiciones y probabilidades utilizadas se visualizan en la tabla 4.2. Luego del proceso de incremento de imágenes, se posee un número de 4500 imágenes por cada muestra original, con sus máscaras correspondientes. Las imágenes poseen el nombre *Placa\_Aug\_Imagen\_X\_Y.jpg*, donde X corresponde a la imagen original, e Y a la iteración de imagen generada por el método de aumento de imágenes. El número total de imágenes y sus respectivos conjuntos se puede ver en la tabla 4.1.

Tabla 4.1: Porcentajes y número de imágenes para entrenamiento, testeo y validación de la red neuronal.

**Fuente:** Elaboración propia, 2020

Directorio	Porcentaje (%)	Nº Imágenes
Entrenamiento (train)	80	15
Evaluación (test)	20	4
Aumentación para entrenar (aug_train)	90 % del set de Entrenamiento	4089
Aumentación para validación en entrenamiento (aug_val)	10 % del set de Entrenamiento	450

Tabla 4.2: Transformaciones escogidas para el proceso de aumento de imágenes.

**Fuente:** Elaboración propia, 2020

Condición	Transformación	%	Descripción
Siempre aplicable	Rotación con Fondo	100	Rotación en 5°, 15° y 25° sin cambio de imagen, añadiendo el tono de gris más común.
Siempre aplicable	Reflejar (Flip)	75	Reflejar la imagen y la máscara de forma horizontal, vertical u ambas.
Siempre aplicable	Ruido Gaussiano (GaussianNoise)	20	Añade ruido gausiano de bajo impacto a la imagen rgb, no aplica la máscara.
Una de Estas (80% de posibilidad de realizar una transformación)	Contraste (RandomContrast)	33	Modifica el contraste de la imagen en general, no afecta a la máscara
	Brillo (Random-Brightness)	33	Modifica el brillo de la imagen, no afecta a la máscara.
	Gamma (RandomGamma)	33	Modifica la gamma de la imagen, no afecta a la máscara.
Una de Estas (60% de posibilidad de realizar una transformación)	Modificación de parámetros RGB (RGBShift)	10	Incrementa o disminuye en 5 valores una o varias componentes de RGB en la imagen. Como la máscara no es RGB no se ve afectada
	Cambiar a Sepia (ToSepia)	40	Modifica la tonalidad de la imagen a Sepia, disminuyendo los valores RGB. Como no modifica el fondo de la imagen, la máscara no se ve afectada
	Cambiar a Blanco y Negro (ToGray)	50	Modifica la tonalidad de la imagen a blanco y negro. Como la tonalidad del fondo sigue siendo distinta al fondo, no afecta a la máscara.

Las imágenes generadas de manera automática pueden generar cierta inconsistencias con los conceptos teóricos, siendo invalidados como muestras reales, y, en el caso de que se usen estas imágenes para entrenar la red neuronal, puede generar problemas en su posterior funcionamiento. Debido a esto, luego de la generación de muestras, se inicia un proceso de filtrado manual, donde se revisan los resultados de las imágenes RGB generadas, y aquellas que no cumplen una serie de criterios son desecharadas en conjunto con sus máscaras. Los criterios

para eliminar una imagen en este proceso de filtrado son:

- Interconexión entre células (cabezas).
- Célula incompleta en la muestra (por ejemplo, cola incompleta).
- Muestra con ninguna célula.
- Transformación agresiva que no es consecuente con conceptos establecidos. (Véase capítulo 2)
- Interpolación agresiva.
- Baja calidad de la imagen en relación a su imagen original.
- Diferencia drástica entre máscara original y aumentada.

Luego del filtrado manual, se eliminan 38 imágenes del set aumentado de entrenamiento, generando el nuevo dataset de imágenes, utilizado para el entrenamiento de la red neuronal. Con este dataset completo, se inician dos procesos en paralelo, que son necesarios para generar la red utilizando la herramienta Deep Cell. La separación del dataset aumentado (para Panoptic) y la creación del dataset en formato .npz (para Retina Net).

Debido a que Deep Cell requiere de datos de entrenamiento y testeo como variables durante el entrenamiento de un modelo Panoptic, es necesario realizar una separación del dataset aumentado en razón 9:1, este procedimiento se realiza de la misma manera que la primera separación del dataset SCIAN-SpermSegGS. El siguiente paso es compresión del dataset en formato .npz. La herramienta Deep Cell inicia el proceso de entrenamiento con el método `train_model_retinanet` y, requiere como parámetro todo el dataset disponible, comprimido como un formato .npz. Para el correcto desarrollo de este método, se requiere del dataset comprimido utilizando la técnica de encriptamiento `zip`, con el nombre .npz, contenido las imágenes RGB (sin la separación de imágenes) con el nombre X.npy y sus respectivas máscaras de una de sus partes, con el nombre y.npy. Los archivos npy corresponde a arreglos de la librería NumPy almacenados como números binarios, y se caracterizan por su rapidez e integración rápida a cualquier sistema, mientras que el comprimido npz solo los almacena con encriptación `zip` (NumPy Steering Council, 2020). Estos archivos se generan con el método `numpy.savez` con un `numpy array` como entrada, formato utilizado para leer las imágenes.

### 4.3.2 Experimentos

Al igual que la mayoría de memorias, publicaciones y/o artículos relacionados a redes neuronales profundas en conjunto a imágenes del área médica, es importante definir diversos experimentos, con el fin encontrar el mejor tipo de red, con sus parámetros definidos, para cada problema particular. En este caso, y al ser el primer análisis de segmentación de espermatozoides con redes neuronales profundas, es importante definir el o los modelos a utilizar, los datos, y las variables importantes.

Para realizar la selección de estos experimentos, se inició un proceso de investigación, buscando artículos científicos que respalden el uso de determinados modelos, datos, procedimientos y variables esenciales en el uso del *deep learning*. Luego de esta investigación, se llegó a la conclusión de realizar 2 experimentos, utilizando dos tipos de modelos base de red neuronal convolucional profunda, **Retina Net** y **Panoptic**, utilizando como datos el *gold standard* original de segmentación de espermatozoides propuesto por (Chang et al., 2017) (con técnicas de aumento de datos), y otro utilizando **transfer learning** con el dataset público **Hela\_S3**, para segmentar cabezas de espermatozoides humanos. El uso de *transfer learning* ha demostrado avances en investigaciones de segmentación de imágenes médicas, principalmente con el uso de datos ajenos a su propósito original (Sultana et al., 2020), y se realiza mediante el entrenamiento de una red neuronal usando el dataset Hela\_S3, para posteriormente cargar los pesos de aquella red en una que usa el dataset SCIAN-SpermSegGS. La elección de parámetros de entrenamiento se basan en experimentos propios, investigaciones pasadas, y artículos que muestran distintos procesos de optimización utilizando RetinaNet con el dataset Hela\_S3. Los experimentos de esta memoria se encuentran descritos en la figura 4.3.

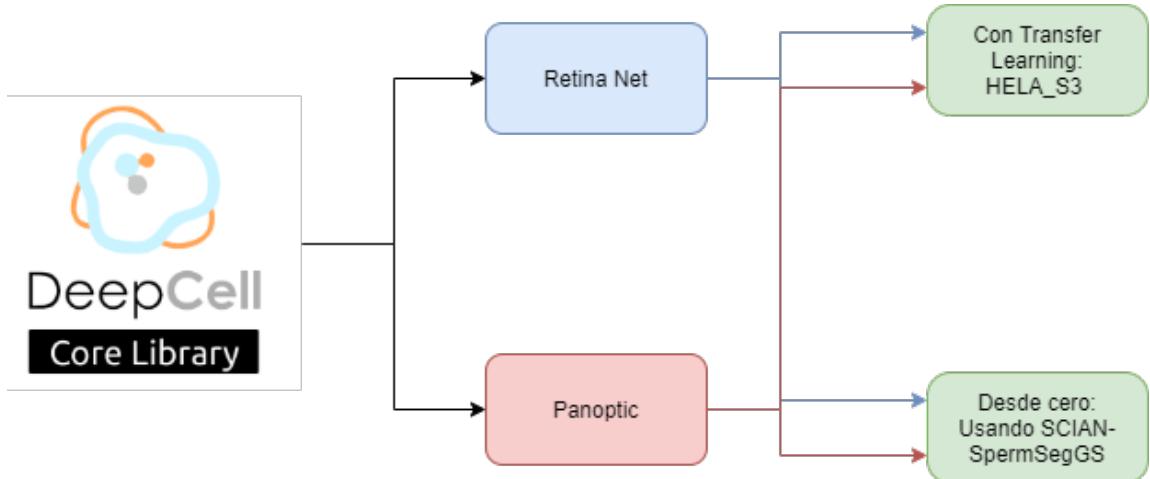


Figura 4.3: Experimentos realizados para determinar el rendimiento de la segmentación automática de partes de espermatozoides.

**Fuente:** Elaboración propia, 2020

#### 4.3.3 Entrenamiento de la red

El proceso de entrenamiento es el proceso más largo de todo este trabajo de investigación, debido a las distintas pruebas, código, y problemas involucrados en una red neuronal de segmentación.

La herramienta Deep Cell consiste en una biblioteca que aporta, principalmente, una gran cantidad de parámetros, funciones, clases y métodos que configuran una versión de TensorFlow, para entrenar modelos especializados en la segmentación de organismos unicelulares. Las principales funcionalidades de Deep Cell son: la obtención de datasets públicos de organismos unicelulares, la creación de una red de tipo piramidal (como RetinaNet y Panoptic), creación y entrenamiento de modelos de segmentación, el uso de una interfaz mediante *Docker* y *Jupyter Notebook*, y un módulo de métricas para analizar el rendimiento de la red neuronal. Sin embargo, para realizar esto requiere una gran cantidad de recursos computacionales, principalmente memoria RAM para almacenar una gran cantidad de imágenes, y GPU para realizar el entrenamiento de manera correcta, según los creadores de la aplicación (Deep Cell, 2019).

La forma recomendada para acceder a la biblioteca *deepcell-tf*, es mediante la imagen de Docker “*vvanvalenlab/deepcell-tf:0.5.0-gpu*”, con las carpetas compartidas mencionadas en la tabla 4.1. Luego de almacenar la imagen y los requerimientos de manera local, se abre un puerto local (*localhost*) de *Jupyter Notebook* accesible desde un navegador. En caso de estar

utilizando DeepCell desde un servidor, es necesario realizar una redirección de puertos, específicamente un *port forwarding* en un túnel SSH. Esto permite transmitir información desde un puerto SSH en el servidor a una máquina local. Luego, el funcionamiento se trabaja a través del Jupyter Notebook y la creación de notebook o archivos .ipynb. Esto además, permite un rápido acceso a la GPU, ya que Docker utiliza su módulo -gpus para utilizar este recurso computacional de manera inmediata. Para iniciar este entorno, es necesario instalar dos bibliotecas esenciales en el sistema; CUDA y Nvidia-Docker.

Sin importar el tipo de red a entrenar, siempre es necesario integrar las imágenes a la plataforma. Para esto, es necesario utilizar una biblioteca con el uso de imágenes, las cuales están integradas con la imagen de Docker. La biblioteca utilizada, por experiencia, facilidad y documentación, es *Pillow (PIL)*, y su módulo *Image*. Otro aspecto esencial es integrar la biblioteca *deepcell-tf* a un kernel de *Jupyter Notebook*, mediante el comando *import deepcell*. Otras librerías utilizadas durante el entrenamiento de una red neuronal, corresponde a os obtención de rutas de archivos y directorios), *glob* (lectura de archivos por directorio), *Matplotlib* (creación de gráficos) *Scipy* y *NumPy* (estructuras de datos asociadas a procedimientos científicos).

Otros parámetros que se deben definir, sin depender del tipo de red a entrenar, son los asociados a la inclusión del dataset en formato .npz (véase capítulo 4.3.1) y la definición de directorios, donde se almacenarán principalmente los modelos obtenidos y los *logs* del entrenamiento. Para el primer tipo de definiciones, se utiliza *Python3* básico, en conjunto con *os.path* para acceder al archivo que debe estar almacenado, por recomendación de los creadores de Deep Cell, en la carpeta data. Para el segundo tipo de parámetros, se utilizan los mismos módulos para definir las carpetas que almacenan los modelos generados, en conjunto con los registros del proceso de entrenamiento, conocidas como *MODEL\_DIR* y *LOGS\_DIR*, respectivamente. Debido a que estas constantes corresponden a argumentos de las funciones *train\_model\_retinanet* (para iniciar el entrenamiento usando un modelo de RetinaNet) y *SemanticDataGenerator.fit\_generator* (para iniciar el entrenamiento usando un modelo de Panoptic), es importante asegurarse de que estos directorios existen y están almacenados dentro del entorno que puede acceder *Jupyter Notebook* en el contenedor de Docker, en caso contrario, deben ser creados. A partir de este punto, el entrenamiento deriva dependiendo del modelo a utilizar.

#### 4.3.3.1 Entrenamiento con RetinaNet

RetinaNet es uno de los modelos más populares en tareas de inteligencia artificial, principalmente en tareas de clasificación y detección de objetos en imágenes. Debido a esto, también han surgido diversas modificaciones basadas en el modelo, aportando a otras áreas

de la visión por computador, por ejemplo, **RetinaMask** (Fu et al., 2019). DeepCell utiliza una combinación propia entre la creación de modelos **RetinaMask**, y el entrenamiento común de un modelo de clasificación RetinaNet, logrando una segmentación por célula basado en la clasificación de muestras y máscaras.

Los entrenamientos de RetinaNet, utilizando la herramienta DeepCell, requieren de una serie de procedimientos previos para iniciar el entrenamiento. Inicialmente, es necesario poseer el set de datos a utilizar como un archivo .npz, la proporción de separación entre los datos de entrenamiento y prueba, y una semilla fija. Además, para la construcción del modelo, es necesario obtener el número de capas necesarias para el modelo, las cuales se obtienen a partir de las máscaras utilizadas para el entrenamiento con el método `get_anchor_params` de la biblioteca DeepCell, indicando el número de capas de base y segmentación del modelo FPN.

Con estos pasos completados, es necesario definir los parámetros de entrenamiento del modelo, incluyendo la tasa de aprendizaje, su optimizador y planificación durante el entrenamiento, el tamaño del lote, y el número de épocas, para ambos set de datos. Existe un artículo (Cui, 2020) que presenta variables con un resultado positivo en la segmentación de células con Hela\_S3, tanto para modelos propios, como su utilización en otros modelos utilizando *transfer learning*. Frente a lo planteado en esta investigación, se utilizaron estos valores, visibles en la Tabla 4.3.

En cambio, no existe ninguna investigación que realice una segmentación de células con el dataset SCIAN-SpermSegGS, por lo que es necesario realizar una serie de pruebas para calcular este proceso. La elección de la tasa de aprendizaje, su planificador y su optimizador, se basa principalmente en investigaciones anteriores que utilizan RetinaNet para segmentar células visualmente similares a Hela\_S3, como es el caso de cuerpos extraños en mamografías (Jung et al., 2018) y segmentación de organismos unicelulares tomadas directamente de un microscopio (Al-Kofahi et al., 2018). En cambio, el tamaño del lote y el número de épocas se obtienen a partir de las pruebas controladas. La obtención de este último se realiza mediante un entrenamiento controlado con parámetros mínimos definidos, y un número de épocas considerablemente alto (1000 épocas). Durante el entrenamiento, es posible visualizar si existe mejora entre cada época, y, los modelos de RetinaNet suelen no tener una mejoría visible luego de cierto número de épocas, lo cual define el valor utilizado en el entrenamiento final, este valor es 5. En el caso del tamaño del lote, se realiza otra sesión de pruebas con parámetros controlados, variando el tamaño en potencias de 2 (1,2,4,8,16,32,64,128). La elección del tamaño de lote se decide a partir del error más bajo encontrado durante el entrenamiento, el cual es 16. La comparación entre los errores ejercidos se puede ver en el gráfico 4.4 y 4.5.

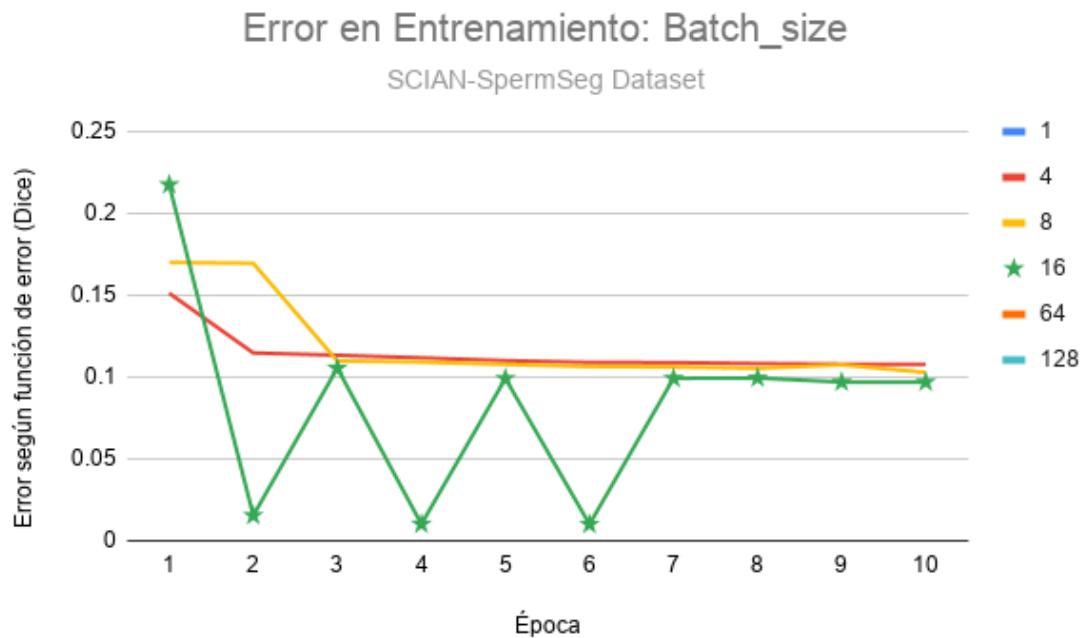


Figura 4.4: Error durante entrenamiento en modelo Retina Net según tamaño de lote con el dataset Hela\_S3. Estrella indica el valor con menor error. Valores que no se muestran poseen un error muy alto para comparar.

**Fuente:** Elaboración propia, 2020

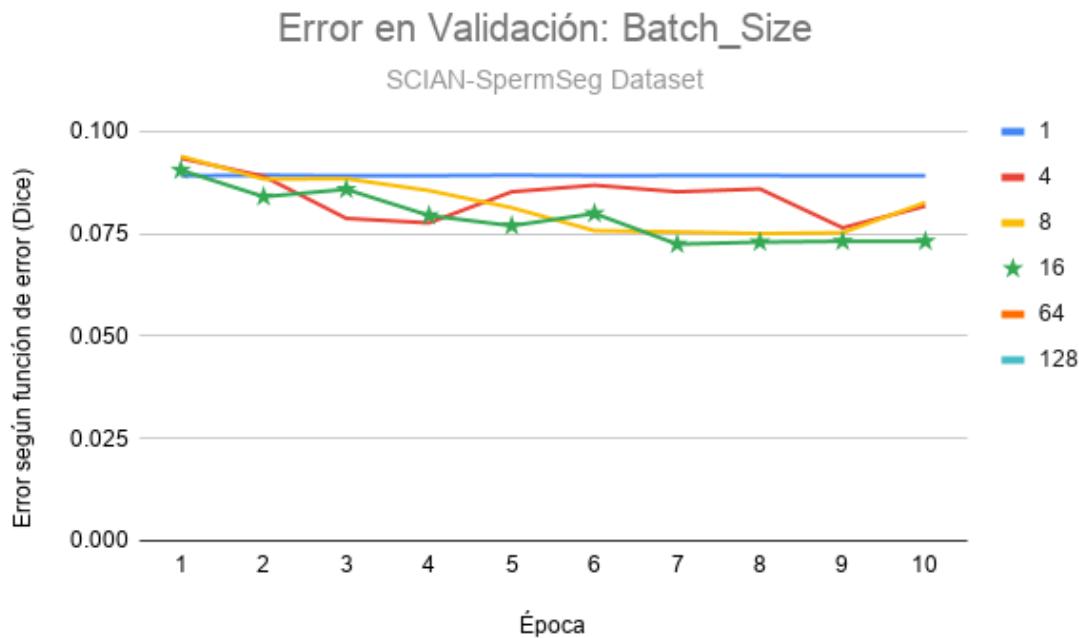


Figura 4.5: Error durante validación en modelo Retina Net según tamaño de lote con el dataset Hela\_S3. Estrella indica el valor con menor error. Valores que no se muestran poseen un error muy alto para comparar.

**Fuente:** Elaboración propia, 2020

A partir de estas pruebas, es posible definir los parámetros para el entrenamiento final con los datos definidos anteriormente para cabezas. Los parámetros se ven en la tabla 4.3.

Tabla 4.3: Parámetros de entrenamiento para modelo RetinaNet con ambos dataset

**Fuente:** Elaboración propia, 2020

Parámetro	Nombre de la Variable	Valor(GoldStandard)	Valor (Hela_S3 Dataset)
Número de épocas	n_epoch	10	10
Tamaño del lote	batch_size	16	4
Tasa de aprendizaje	lr	1e-5	1e-5
Optimizador de aprendizaje	optimizer	ADAM	ADAM
Scheduling de tasa de aprendizaje	lr_sched	Deepcell rate_scheduler(decay=0.99)	Deepcell rate_scheduler(decay=0.99)
Número de clases a encontrar	num_classes	1 (La cabeza, núcleo o acrosoma)	1 (Solo un tipo de célula en el dataset)

Con los parámetros definidos para los experimentos esenciales, se inicia el entrenamiento. En el caso del modelo con el set de datos SCIAN-SpermSegGS, éste se realiza a partir del método *train\_model\_retinanet* de DeepCell, en conjunto con diversos parámetros, a saber, los mencionados en la tabla 4.3, los parámetros que indican el número de capas, la ruta

al archivo `.npz`, entre otros. El método inicia el entrenamiento, y durante su desarrollo, aporta con métricas como el error del modelo y la precisión obtenida al comparar las máscaras. Durante cada época se evalúa si el error disminuye, y en caso de ser así, se guardan los pesos que generaron un error mínimo durante el proceso de entrenamiento. Además, y con el fin de generar un modelo más robusto, y revisar la calidad de la segmentación en cada célula del dataset SCIAN-SpermSegGS, se realiza el proceso de validación cruzada con 6. Al finalizar, se almacena cada modelo con el método `save_model` de Keras, que genera un archivo `.h5`.

En cambio, y debido a que las muestras del dataset Hela\_S3 posee un tamaño mucho menor (216x256) en comparación al dataset SCIAN-SpermSegGS (780x580) para lograr el proceso de *transfer learning* es necesario realizar algunas modificaciones. En un inicio, se amplía el tamaño de las imágenes del dataset Hela\_S3 a 580x780, reduciendo el tamaño de las células de Hela\_S3 en comparación a la imagen completa, haciéndolas más similares a las células del dataset SCIAN-SpermSegGS. Así, las muestras poseen las mismas dimensiones, lo que no generará problemas a la hora de realizar el proceso de *transfer learning*. Luego de este proceso, se crean los nuevos archivos `.npz` (Vease Sección 4.3.1), y se genera un modelo a partir el entrenamiento inicial con el dataset Hela\_S3, y los pesos de estos se cargan mediante la función `load_weights` a un modelo con el dataset SCIAN-SpermSegGS. Al igual que con el experimento anterior, se realiza un proceso de validación cruzada de 6 iteraciones.

#### 4.3.3.2 Entrenamiento con Panoptic

Debido a la estructura piramidal (FPN), y al uso de determinadas operaciones en el modelo de Panoptic, es necesario transformar los datos de entrenamientos, para poseer un razón 1:1, es decir, que la dimensión de largo y ancho sean idénticas, generando muestras cuadradas de potencias de 2. Para esto, se utiliza la función propia `reshape_to_512`, que recorta las muestras del dataset SCIAN-SpermSegGS en el centro, generando una imagen de dimensión 512x512, eliminando cualquier célula que se encuentre en los bordes, para no generar problemas en el entrenamiento posterior. Esto permite al set de datos poseer una proporción cuadrada según un valor determinado, y sigue siendo un espermatozoide válido. Con este método, las imágenes del entrenamiento utilizando el dataset SCIAN-SpermSegGS cambiaron su dimensión de 580x780 a 512x512. Para las muestras del modelo Hela\_S3, se realiza el mismo proceso de ampliación de la muestra de RetinaNet, pero a una dimensionalidad de 512x512, cumpliendo los requerimientos para cargar los pesos de un modelo a otro.

En relación a los parámetros para la creación del modelo, el aspecto más importante corresponde a la definición de las **cabezas semánticas**, y **clases semánticas**. Los modelos Panoptic realizan una segmentación semántica, es decir, separan el fondo de las imágenes que se

buscan detectar y segmentar. Para realizar este proceso, es necesario definir como variables, el número de objetos que se buscan diferenciar (conocidos también como **distancias semánticas**) (Ver Glosario), como el número de clases que pueden encontrarse en cada muestra. Para los 2 modelos, es necesario encontrar tanto la distancia interna y externa de una célula, cada uno con un clase semántica, ya que se derivan sólo de la máscara, y la diferencia entre objeto y forma, corresponde a la parte identificada por la máscara, y las clases semánticas corresponde al objeto, que adquiere un tono de píxel con valor 1 en la máscara, y el fondo, que posee un tono de píxel 0, es decir, se observan 2 clases.

Los modelos FPN de Panoptic, debido a su doble segmentación, poseen la capacidad de generar máscaras de nuevos datos, por lo que es necesario crear un generador de datos para entrenar, a pesar de que no sea usado. En este caso, la biblioteca Deep Cell posee un módulo llamado *image\_generations*, que son, valga la redundancia, generadores de datos basados en la librería *keras*. Estos generadores de datos no realizan transformaciones, ya que este trabajo se realizó en el desarrollo del objetivo anterior.

A continuación, se define la función de error, que busca determinar cuándo y cómo se ajustan los pesos de las capas luego de comparar el error real con el error predecido. La función de error escogida por los creadores de la plataforma, para segmentar elementos con 1 cabeza semántica es la función *Dice\_Coef\_Loss*, o el complemento del coeficiente de dice. El funcionamiento del coeficiente de Dice como función de error consistió en el cálculo de coeficiente de Dice al final de cada época, comparando la máscara generada por el modelo, con la real, mientras este valor sea más bajo, el coeficiente de Dice es más alto. Además, y para un posterior análisis, se incluyeron dos métricas de medición adicionales, la función de error de los mínimos cuadrados MSE y la función *weighted categorical cross entropy loss* (Ver Glosario), las cuales son las funciones más utilizadas para medir el error en redes neuronales de segmentación semántica (Jordan, 2018). Esta función se almacena en un diccionario y es ingresada al modelo mediante la función *model.compile*. Con este último paso, es posible empezar el entrenamiento de un modelo Panoptic.

En un inicio, debido a la falta de un aparato computacional potente, y para corroborar el funcionamiento general de la red neuronal, se realizaron una serie de pruebas utilizando un número menor de muestras (específicamente, las 19 muestras originales del dataset SCIAN-SpermSegGS), parámetros reducidos (Ver tabla 4.4), y solo la segmentación de cabezas, para lograr un entrenamiento en el ordenador personal (Ver tabla 1.2). El entrenamiento de esta red neuronal demoró aproximadamente de 1 hora y 30 minutos, debido principalmente a los bajos recursos del ordenador. El error final de este experimento fue de 0.0025 en la época 5, lo cual es considerablemente bajo, y se explica por el poco tiempo entrenando la red neuronal, sumado a la mínima cantidad de imágenes utilizadas. Al evaluar el modelo utilizando el dataset de prueba

(Test), en conjunto con el módulo *Metrics* de *deepcell-tf*, encontramos resultados muy bajos, con un índice de Jaccard de 0.0214, y sin detectar un 98% de las células reales. De manera visual, podemos ver que el modelo logra distinguir correctamente células del fondo, pero con muy poca precisión, lo cual se explica debido al mínimo entrenamiento y baja cantidad de imágenes. Un ejemplo de lo anterior se puede ver en la imagen 4.6.

Tabla 4.4: Parámetros utilizados en red neuronal Panoptic con computador personal.

**Fuente:** Elaboración propia, 2020

Parámetro	Nombre de la Variable	Valor
Número de épocas	n_epoch	5
Tamaño del lote	batch_size	1
Tasa de aprendizaje	lr	1e-5
Optimizador de aprendizaje	optimizer	Adam (clipnorm=0.001)
Scheduling de tasa de aprendizaje	lr_sched	decay=0.99
Método de normalización	norm_method	Imagen completa (whole_image)
Número de objetos por muestra.	min_objects	3

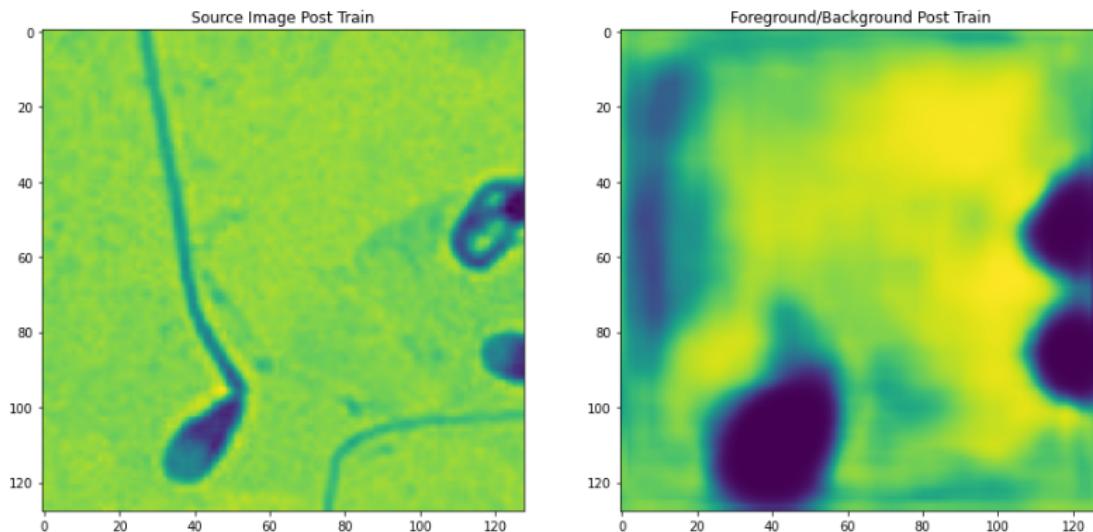


Figura 4.6: Ejemplo de detección de células con modelo Panoptic, usando solo SCIAN-SpermSegGS, con experimento inicial en ordenador personal.

**Fuente:** Elaboración propia, 2020

Debido a que no existe ninguna investigación, o artículo que realice una segmentación usando un modelo Panoptic en organismos unicelulares, utilizando el dataset SCIAN-SpermSegGS o Hela\_S3, es necesario realizar experimentos para encontrar buenos parámetros para iniciar el proceso de entrenamiento. Los principales parámetros que fueron necesarios encontrar son el número de épocas, el tamaño del lote, y la tasa de aprendizaje, en conjunto con su optimizador.

En la búsqueda de los parámetros para el set de datos de Hela\_S3, el tamaño del lote se escogió luego de realizar entrenamientos en un ambiente controlados de 10 épocas, en conjunto con los parámetros utilizados en el ejemplo de DeepCell, utilizando potencias de dos como posibles valores (sin considerar el valor 2, por la similitud del error al usar el valor 1), hasta llegar al valor 128. Tomando como métrica principal el error obtenido derivado de la función de pérdida del coeficiente de Dice, el valor escogido es 16, basado en los resultados visibles en los gráficos 4.7 y 4.8 y el tiempo necesario para entrenar. A pesar de que en entrenamiento el error con un *batch\_size* es mayor que otros valores, en validación el error es menor, y este es el principal factor a la hora de almacenar los pesos en la red neuronal. Otros parámetros como el *Scheduling* de la tasa de aprendizaje, el método de normalización, y el número de objetos mínimos se basó en la documentación y ejemplos de la plataforma Deep Cell.

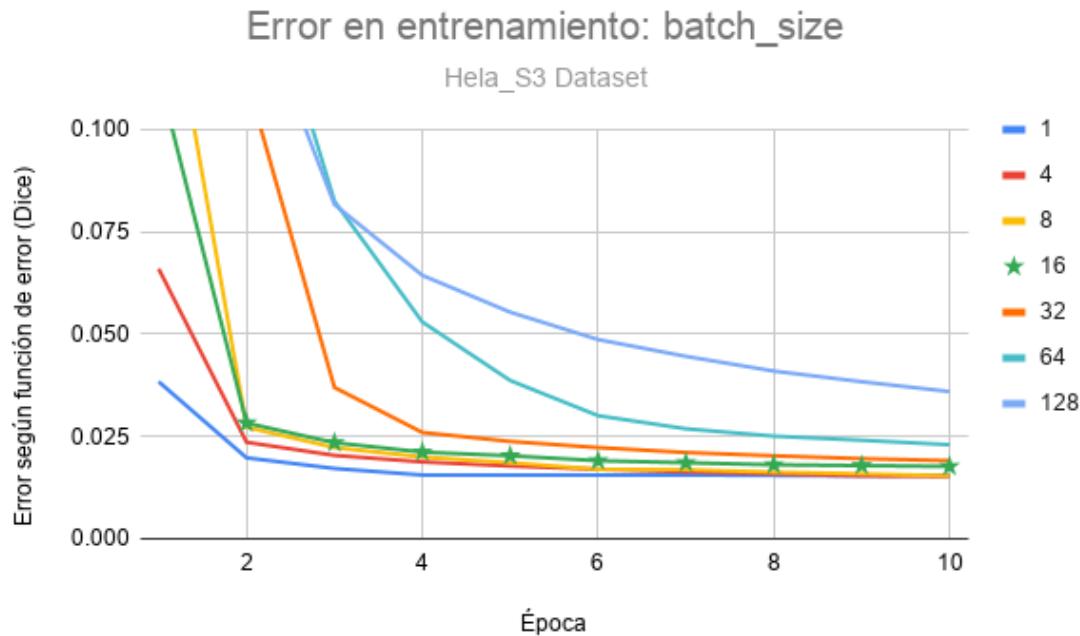


Figura 4.7: Error durante entrenamiento según tamaño de lote con el dataset Hela\_S3. Estrella indica el valor escogido.

**Fuente:** Elaboración propia, 2020

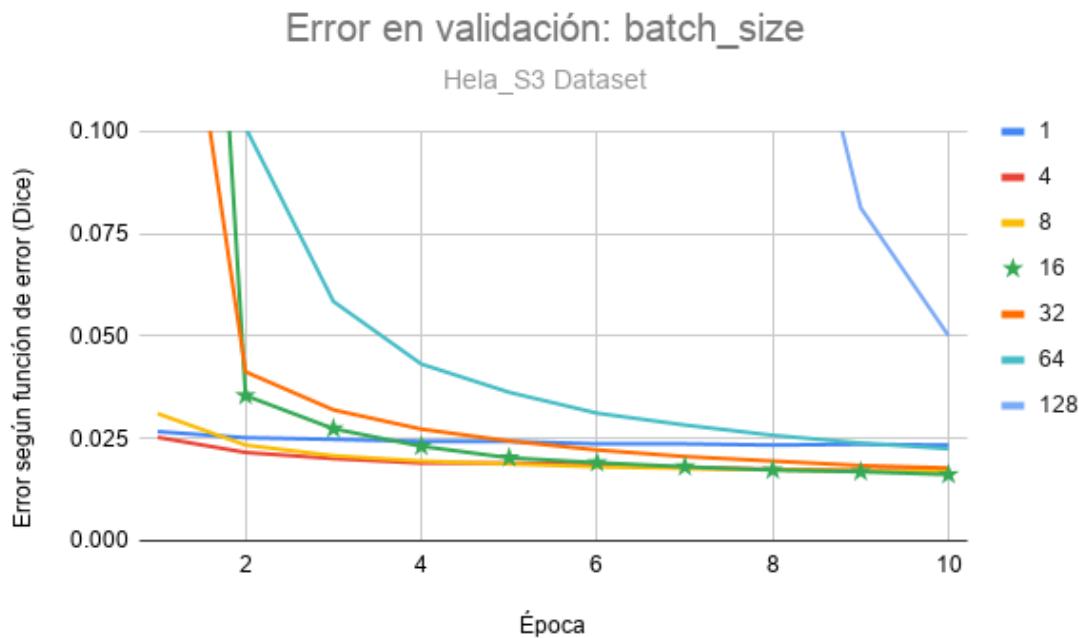


Figura 4.8: Error durante validación según tamaño de lote con el dataset Hela\_S3. Estrella indica el valor escogido.

**Fuente:** Elaboración propia, 2020

En el caso del dataset derivado del *Gold Standard*, el número de épocas se deriva del promedio de épocas ocurridas hasta que el error del modelo no mejoró, utilizando valores por defecto en otros parámetros. Este experimento inicia entrenando la red con 1000 épocas, y a través de la revisión constante de cada paso de época (con la ayuda de la librería *Tensor Board*), se encontró que el error del modelo no mejoró luego de 25 épocas, por lo que se decidió este valor como número de etapas para el modelo. El número de lote, en cambio, se definió luego de entrenar el modelo con distintos valores; 1, 4, 8, 16, 32, 64 y 128, escogiendo el parámetro basado en los mismos criterios para el dataset con Hela\_S3, el cual es 8. En el caso específico del tamaño de lote con valor 64 y 128, los recursos computacionales no eran suficientes, por lo que se consideró un error máximo. La comparación entre el valores se puede ver en las imágenes 4.9 y 4.10. Luego de obtener los dos valores, la elección del optimizador se basó en 2 principales opciones, debido a que son los únicos optimizadores utilizados en segmentación con modelos Panoptic: SGD (gradiente descendiente), y su versión optimizada, ADAM. La elección fue esta última, en base principalmente a los comentarios, sugerencias y ejemplos que se expresan en la página de la herramienta Deep Cell. Además, durante el desarrollo de las distintas pruebas, se notó un considerable uso de recursos computacionales, lo que impulsó la decisión de escoger una versión del optimizador más liviana. Al mismo tiempo, la tasa de aprendizaje escogida fue de

$1e-5$ , por ser la tasa más utilizada en modelos de segmentación con modelos Panoptic (Liu et al., 2020).

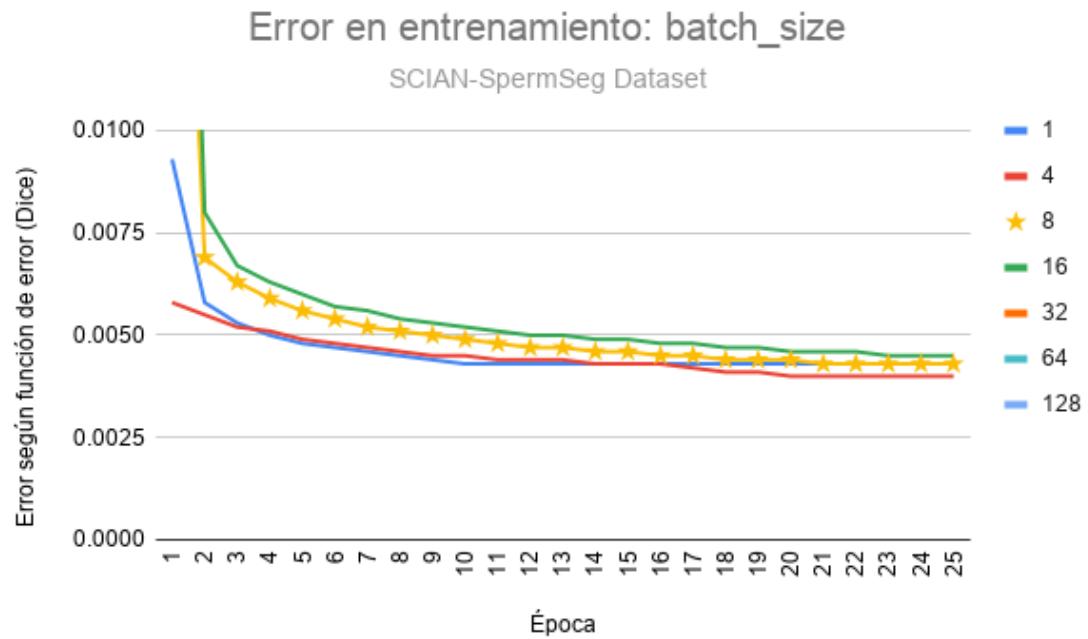


Figura 4.9: Error durante entrenamiento según tamaño de lote con el dataset SCIAN-SpermSegGS. Estrella indica el valor escogido.

**Fuente:** Elaboración propia, 2020

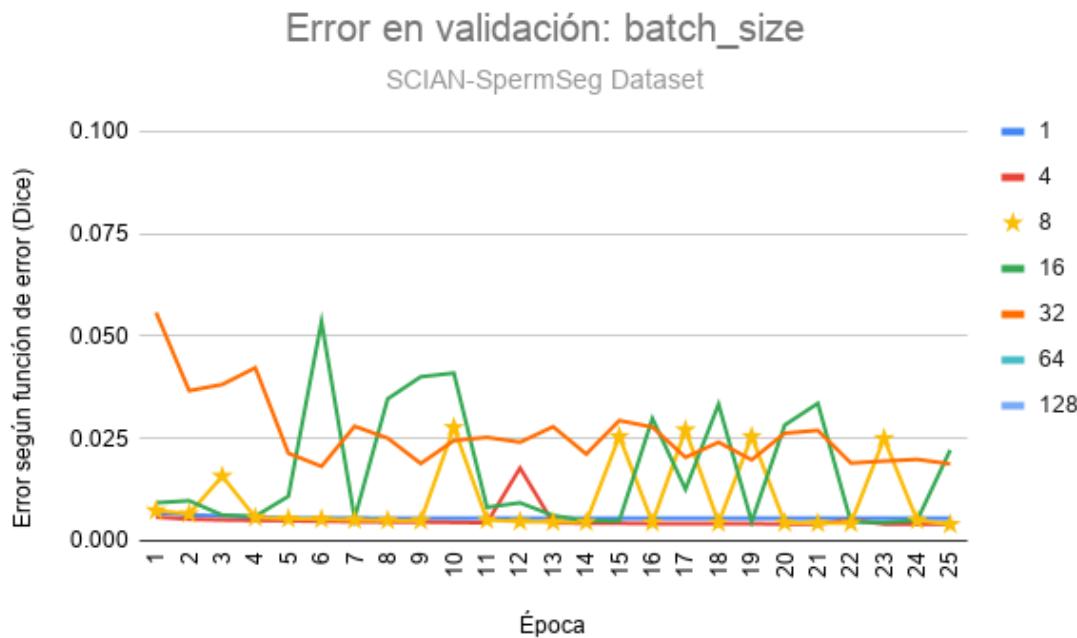


Figura 4.10: Error durante validación según tamaño de lote con el dataset SCIAN-SpermSegGS.  
Estrella indica el valor escogido.

**Fuente:** Elaboración propia, 2020

La creación del modelo, en la variable *model*, depende de la clase Panoptic de Deep Cell. El modelo indica cerca de 300 capas, que incluye capas de modificación, clasificación y regresión, todo esto en base a los parámetros de entrenamiento. Cabe destacar que estos parámetros se utilizan por igual considerando las cabezas de los espermatozoides. Estos parámetros se visualizan en la Tabla 4.5

Tabla 4.5: Parámetros de entrenamiento para modelo Panoptic.

**Fuente:** Elaboración propia, 2020

Parámetro	Nombre de la Variable	Valor(GoldStandard)	Valor (Hela_S3 Dataset)
Número de épocas	n_epoch	15	10
Tamaño del lote	batch_size	8	16
Tasa de aprendizaje	lr	1e-5	1e-5
Optimizador de aprendizaje	optimizer	ADAM	ADAM
Scheduling de tasa de aprendizaje	lr_sched	Deepcell rate_scheduler(decay=0.99)	Deepcell rate_scheduler(decay=0.99)
Método de normalización	norm_method	'whole-image' (Toda la imagen pasa al rango [0,1])	'whole-image' (Toda la imagen pasa al rango [0,1])

Con todos los parámetros definidos, se inicia el entrenamiento de los modelos. El

primer modelo a entrenar es el utilizable con Hela\_S3. DeepCell, durante el desarrollo, almacena tanto los registros como los mejores pesos del modelo en un archivo .h5. Al finalizar, y luego de generar la arquitectura utilizando el dataset SCIAN-SpermSegGS, se inicia el proceso de *transfer learning* con la integración de los pesos al modelo como pesos iniciales, mediante el comando `load_weights` que recibe como argumento el archivo con los pesos e iniciar el entrenamiento. En el caso del modelo con SCIAN-SpermSegGS, este proceso se ignora debido a que es un modelo creado desde cero. Durante el entrenamiento de los modelos se miden 3 métricas que permiten visualizar el rendimiento del entrenamiento, el coeficiente de Dice (que actúa como función de error), el MSE, y la entropía. Con los modelos finalizados, se almacena el modelo en formato .h5 utilizando la función `save_model` de keras. Luego, para obtener las máscaras predichas por los modelos, y generar una imagen final de las mismas dimensiones que las imágenes originales del set de datos SCIAN-SpermSegGS, se aplican los métodos `desarm_to_512` y `rearm_from_512`, el primero, que divide cada muestras del conjunto de validación de tamaño 780x580 en 4 imágenes de 512x512, aplicables para la predicción del modelo, y se unen mediante el segundo método, generaron la imagen original a partir de las 4 imágenes divididas.

#### 4.3.4 Análisis y comparación de resultados

Con un modelo ya entrenado en DeepCell, es posible predecir nuevos datos, obteniendo máscaras de partes de espermatozoides humanos segmentados de manera automática, recibiendo como entrada una imagen. Sin embargo, para medir su precisión y funcionamiento en general y poder compararlo con el material original (*Ground Truth*) y el estado del arte, es necesario especificar la forma en que se entregan los resultados de cada modelo, las métricas utilizadas para realizar la comparación entre modelos y con la investigación vigente, y los resultados específicos para cada muestra, entre otros pasos previos antes del análisis concreto.

Luego de poseer un modelo ya entrenado, es posible predecir sobre nuevos datos de entrada, obteniendo máscaras binarias con la parte segmentada de ese valor de entrada. Esto se realiza mediante el método `predict` del modelo ya entrenado, además de una serie de técnicas de limpieza y mejoramiento de máscaras, proporcionadas por la biblioteca Deep Cell, que varían según el tipo de modelo generado.

En los modelos generados a partir de un arquitectura RetinaNet, la predicción de datos se realiza por imagen de entrada, es decir, a partir de una imagen en escala de grises, se genera un *numpy array*, correspondiente a un arreglo de datos que contiene las posiciones, dimensiones máximas, precisiones y máscaras binarias de cada una de las células encontradas en la imagen. Como todo se encuentra en una estructura de datos, es necesario desmenuzar

este arreglo, y generar una imagen que posea las máscaras binarias. Esto se realiza mediante la función *tile* y *expand\_dims* de la librería *Numpy* y la función *draw\_masks* de la librería *plot\_utils* de DeepCell. Finalmente, y debido a que la matriz se genera en base a la imagen en escala de grises, se realiza una binarización de la imagen, obteniendo la máscara predecida final.

Por otra parte, los modelos basados en Panoptic, realizan la predicción a partir de un set de datos, utilizando el mismo método *predict*. Al ejecutar, se generan las distancias semánticas definidas (distancia interna, distancia externa y separación fondo/forma) de cada célula por cada una de las imágenes del set de datos de entrada. A partir de estos elementos, es posible crear máscaras con el método *deep\_watershed* de la librería de apoyo de DeepCell: *deepcell\_toolbox*, que se basa en la segmentación profunda con forma de cuenca en organismos unicelulares (Bai & Urtasun, 2017).

Para medir el resultado de la segmentación por cada célula, es necesario realizar una comparación entre la máscara original (*Ground Truth*), y la generada por cada modelo, y para cada célula existente en el dataset SCIAN-SpermSegGS. En primera instancia, se debe identificar cada célula del dataset SCIAN-SpermSegGS de manera manual en una hoja de cálculo Excel. Luego, con el apoyo de una submáscara de 64x64 que recorre las dos imágenes al mismo tiempo, se realiza la aplicación de métricas de evaluación por célula. En el caso de que esta submáscara genere conflictos (dos células en una misma revisión, célula recortada, entre otros), se reduce el tamaño general de la submáscara a 32x32. Para cada célula se aplica las métricas de evaluación mencionadas en la sección 2.2.5, es decir: Coeficiente de Dice, Jaccard y Distancia de Hausdorff. Las implementaciones de estas métricas se realizan con funciones de desarrollo propio, apoyadas por bibliotecas computadoras como *Numpy* y *Scipy*. Con todos los datos disponibles, es posible calcular métricas de tendencia central, las cuales serán las principales métricas para analizar y comparar el rendimiento y calidad de la segmentación, entre modelos, y en comparación con el estado del arte.

Para comparar los métodos generados durante este trabajo de investigación, con el avance de la investigación en la segmentación automatizada de espermatozoides humanos, se realiza una comparación con los resultados obtenidos en el artículo “Automatic segmentation of Sperm’s parts in microscopic images of human semen smears using concatenated learning approaches” (Movahed et al., 2019). Este artículo corresponde a la investigación más reciente en esta área, además de utilizar el mismo dataset SCIAN-SpermSegGS para medir sus resultados. Las máscaras generadas por este método fueron obtenidas mediante una solicitud hacia los autores del artículo que facilitaron los datos.

# CAPÍTULO 5. RESULTADOS EXPERIMENTALES

En este capítulo, se busca mostrar de manera clara y concisa, los resultados de los experimentos ya mencionados en capítulos anteriores. Estos se muestran separados por el tipo de experimento según el dataset y subdivido por el tipo de modelo.

## 5.1 EXPERIMENTO 1: GOLD STANDARD DESDE CERO

### 5.1.1 Protocolo de Experimentos

El proceso de entrenamiento se realiza según lo diseñado e implementado en el capítulo 4. Teniendo como entrada las imágenes aumentadas para cada iteración de validación cruzada, tanto en formato **.npz**, como **.jpg** en el ambiente local. El algoritmo, en pseudocódigo, es el siguiente:

- Para un modelo de Retina Net o Panoptic:
  - Para la primera iteración de validación cruzada.
    1. Generación de un Kernel de *Jupyter Notebook* con *deepcell-tf* aumentado.
    2. Integración de los datos de entrenamiento, validación y evaluación de cada primera iteración a *Jupyter Notebook* y se realiza un preprocesamiento en caso de generar un modelo Panoptic.
    3. Generación del modelo utilizando *model\_zoo* de *deepcell-tf*
    4. Definición de parámetros de entrenamientos.
    5. Inicio del entrenamiento del modelo.
    6. Guardado local del modelo generado y sus pesos correspondiente.
    7. Predicción de máscaras a partir del conjunto de evaluación.
    8. Aplicación de las métricas de evaluación a los resultados de la segmentación y guardado en una hoja de cálculo.
  - Para cada una de las iteraciones restantes:
    1. Generación un Kernel de *Jupyter Notebook* con *deepcell-tf* aumentado.
    2. Se integran los datos de entrenamiento, validación y evaluación de iteración correspondiente a *Jupyter-Notebook*. y se aplica preprocesamiento en caso de necesitarlo.

3. Carga del modelo de la iteración anterior y sus pesos.
  4. Inicio del entrenamiento del modelo.
  5. Almacenamiento local del modelo generado y sus pesos.
  6. Predicción de máscaras a partir del conjunto de evaluación.
  7. Aplicación de las métricas de evaluación a los resultados de la segmentación y registro en una hoja de cálculo.
- Se obtienen medidas de tendencia central, gráficos y funciones de densidad a partir de los resultados

### **5.1.2 Segmentación de Cabezas**

Para la segmentación de cabezas, los resultados se dividieron en el tipo de modelo a utilizar: Retina Net y Panoptic. A continuación, se presenta una tabla de datos contenido en medidas de tendencia central en base a las métricas definidas en las tablas 5.1 y 5.2, respectivamente. Cada tabla se encuentra acompañada de un gráfico de cajas para representarlas, visibles en las figuras 5.1 y 5.3. Además, se adjunta la superposición entre muestra (imagen sin editar), máscara de referencia (color blanco) y máscara segmentada (azul) para ambos modelos en las imágenes 5.2 para Retina Net y 5.4 para Panoptic.

### **5.1.3 Retina Net**

Tabla 5.1: Promedios y mediana del modelo Retina Net desde cero, en distintas instancias.

**Fuente:** Elaboración propia, 2020

	<b>Dice</b>	<b>Jaccard</b>	<b>Hausdorff</b>
Promedio de cabezas de espermatozoides	0,917458484	0,852925447	0,742010342
Promedio: Mejores 50 Resultados	0,95838968	0,920007244	0,963588021
Promedio: Peores 50 Resultados	0,625534572	0,5453805	0,459943225
Mediana	0,92967	0,86770	0,70725

## Evaluación: Retina Net desde cero.

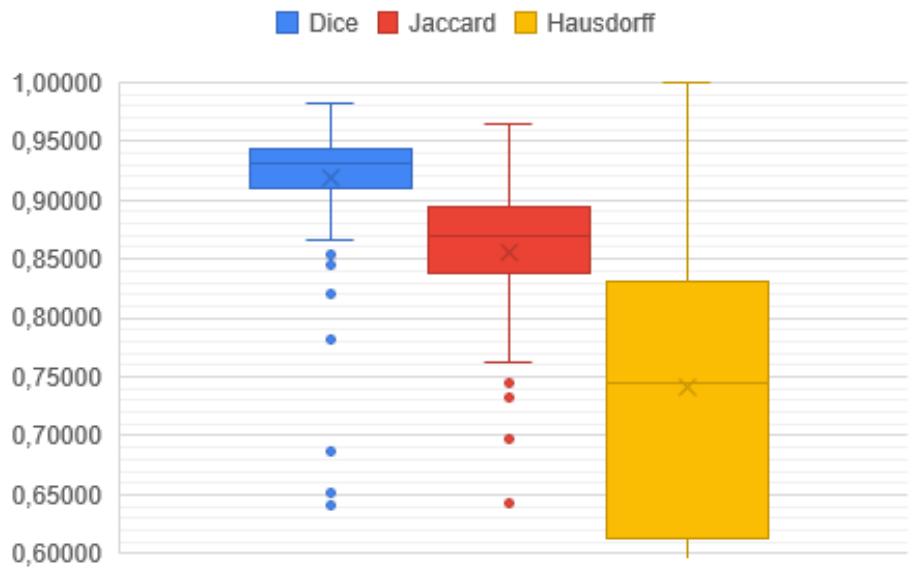


Figura 5.1: Resultados del modelo con Retina Net desde cero, sin considerar falsos positivos. Línea central dentro de las cajas indica Promedio (Dice: 0,917458484, Jaccard: 0,852925447, Haussdorf: 0,742010342)

**Fuente:** Elaboración propia, 2020

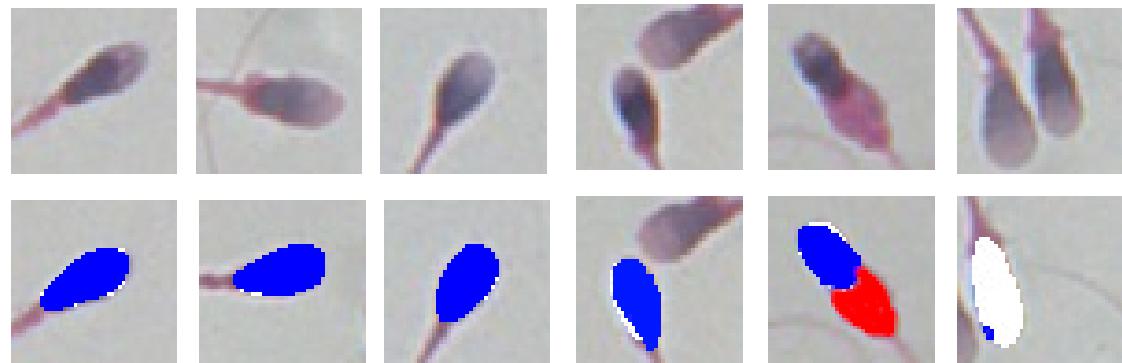


Figura 5.2: Superposición entre muestras (Imagen RGB), máscaras de referencia (Color Blanco) y máscaras predecidas por modelo Retina Net desde cero (Color Azul) con las células con mayor y menor coeficiente de Dice. El color rojo representa partes que el modelo predijo, pero que no existen en la máscara de referencia. En la primera fila se pueden ver las imágenes originales, y en la segunda fila el proceso de superposición. Los valores de Dice para cada imagen son, de izquierda a derecha: **0.9752, 0.981884, 0.98217, 0.6418, 0.60493, 0.1904**

**Fuente:** Elaboración propia, 2020

### 5.1.4 Panoptic

Tabla 5.2: Promedios y mediana del modelo Panoptic desde cero, en distintas instancias.

**Fuente:** Elaboración propia, 2020

	Dice	Jaccard	Hausdorff
Promedio de cabezas de espermatozoides	0,793131426	0,672802819	0,74237645
Promedio: Mejores 50 Resultados	0,8776612	0,8182076	0,966162715
Promedio: Peores 50 Resultados	0,60851988	0,4560856	0,517870846
Mediana	0,81257	0,68450	0,70723

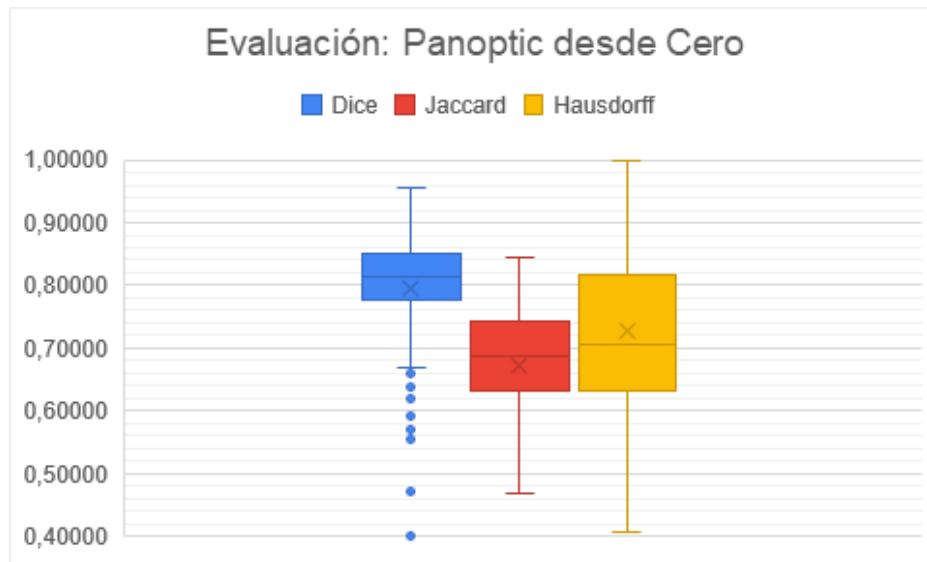


Figura 5.3: Resultados del modelo con Panoptic desde cero, sin considerar células no detectadas. Línea central dentro de las cajas indica Promedio (Dice: 0,80036165, Jaccard: 0,675362142, Haussdorff: 0,742010342)

**Fuente:** Elaboración propia, 2020

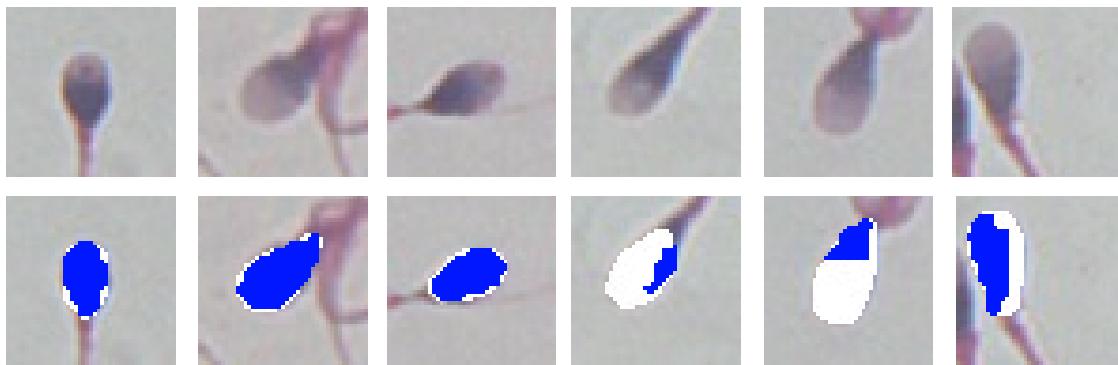


Figura 5.4: Superposición entre muestras (Imagen RGB), máscaras de referencia (Color Blanco) y máscaras predecidas por modelo Panoptic desde cero (Color Azul) con las células con mayor y menor coeficiente de Dice. En la primera fila se pueden ver las imágenes originales, y en la segunda fila el proceso de superposición. Los valores de Dice para cada imagen son, de izquierda a derecha: **0.90339, 0.9115, 0.95551, 0.3197, 0.4009, 0.4733**

**Fuente:** Elaboración propia, 2020

## 5.2 EXPERIMENTO 2: TRANSFER LEARNING CON HELA\_S3

### 5.2.1 Protocolo de Experimentos

El proceso de entrenamiento se realiza según lo diseñado e implementado en el Capítulo 3. Teniendo como entrada las imágenes aumentadas para cada iteración de validación cruzada, tanto en formato **.npz**, como **.jpg** en el ambiente local. El algoritmo, en pseudocódigo, es el siguiente:

- Para un modelo de Retina Net o Panoptic:
  - Para la primera iteración.
    1. Generación de un Kernel de *Jupyter Notebook* con *deepcell-tf* aumentado.
    2. Integración de los datos de entrenamiento y validación de Hela\_S3 a *Jupyter Notebook*, modificación de tamaño de muestras según arquitectura del modelo. (512x512 para Panoptic, 580x780 para Retina Net)
    3. Creación de **.npz** en caso de modelo Retina Net.
    4. Generación del modelo utilizando *model\_zoo* de *deepcell-tf*
    5. Definición de parámetros de entrenamientos.
    6. Inicio del entrenamiento del modelo.
    7. Almacenamiento local del modelo generado y sus pesos correspondiente.

- Para cada una de las iteraciones de validación cruzada:
  1. Generación un Kernel de *Jupyter Notebook* con *deepcell-tf* aumentado.
  2. Se integran los datos de entrenamiento, validación y evaluación de iteración correspondiente a *Jupyter-Notebook*. y se aplica preprocesamiento en caso de necesitarlo.
  3. Carga del modelo de la iteración anterior y sus pesos.
  4. Inicio del entrenamiento del modelo.
  5. Almacenamiento local del modelo generado y sus pesos correspondiente.
  6. Predicción de máscaras a partir del conjunto de evaluación.
  7. Aplicación de las métricas de evaluación a los resultados de la segmentación y registro en una hoja de cálculo.
- Se obtienen medidas de tendencia central, gráficos y funciones de densidad a partir de los resultados

### 5.2.2 Segmentación de Cabezas

Para la segmentación de cabezas, los resultados se dividieron en el tipo de modelo a utilizar, al igual que en el primer experimento: Retina Net y Panoptic. Así mismo, se adjuntan las tablas 5.3 y 5.4, que contienen datos con medidas de tendencia central en base a las métricas definidas y sus respectivos un gráfico de cajas (figuras 5.5 y 5.7). Además, se adjunta la superposición entre muestra (Imagen sin editar), máscara de referencia (color blanco) y máscara segmentada (azul) para el modelo Retina Net (en la figura 5.6) y Panoptic (en la figura 5.8).

### 5.2.3 Retina Net

Tabla 5.3: Promedio y mediana del modelo Retina net con transfer learning

**Fuente:** Elaboración propia, 2020

	Dice	Jaccard	Hausdorff
Promedio de cabezas de espermatozoides	0,894784029	0,820961902	0,741437743
Promedio: Mejores 50 Resultados	0,9452678	0,8953704	0,934583455
Promedio: Peores 50 Resultados	0,7295142	0,6448754	0,500655217
Mediana	0,90665	0,83130	0,70922

## Evaluación: Retina Net con Transfer Learning

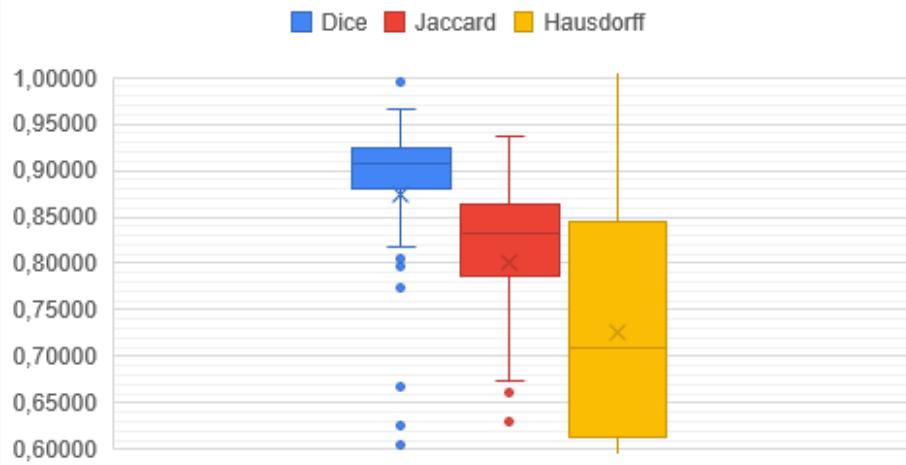


Figura 5.5: Resultados del modelo Retina Net con transfer learning. Línea central dentro de las cajas indica Promedio (Dice: 0,8947, Jaccard: 0,82096, Haussdorf: 0,7414)

**Fuente:** Elaboración propia, 2020

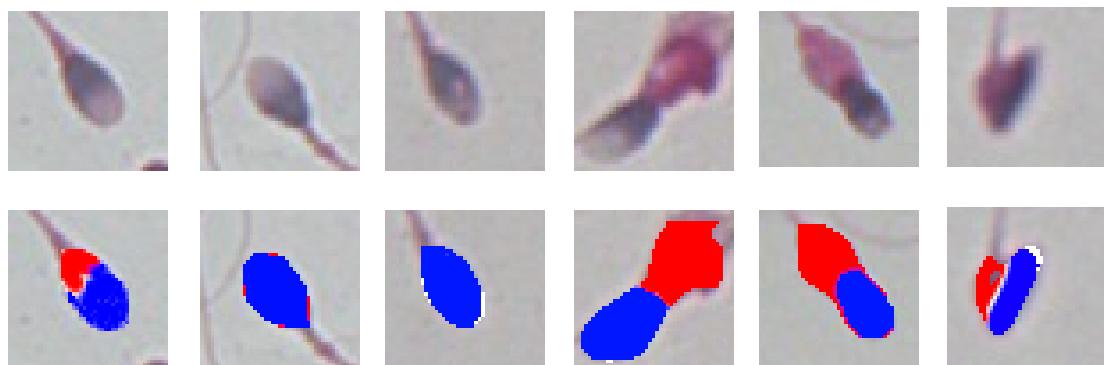


Figura 5.6: Superposición entre muestras (Imagen RGB), máscaras de referencia (Color Blanco) y máscaras predecidas por modelo Retina Net con *transfer learning* (Color Azul) con las células con mayor y menor coeficiente de Dice. El color rojo representa partes que el modelo predijo, pero que no existen en la máscara de referencia. En la primera fila se pueden ver las imágenes originales, y en la segunda fila el proceso de superposición. Los valores de Dice para cada imagen son, de izquierda a derecha: **0,9669, 0,9608, 0,9581, 0,4829, 0,60579, 0,773**.

**Fuente:** Elaboración propia, 2020

#### 5.2.4 Panoptic

Tabla 5.4: Promedios y mediana del modelo Panoptic con transfer learning.

**Fuente:** Elaboración propia, 2020

	Dice	Jaccard	Hausdorff
Promedio de cabezas de espermatozoides	0,724927358	0,570430247	0,786017676
Promedio: Mejores 50 Resultados	0,7897344	0,654516863	0,979426329
Promedio: Peores 50 Resultados	0,64272636	0,4687428	0,581366069
Mediana	0,73000	0,57480	0,79066

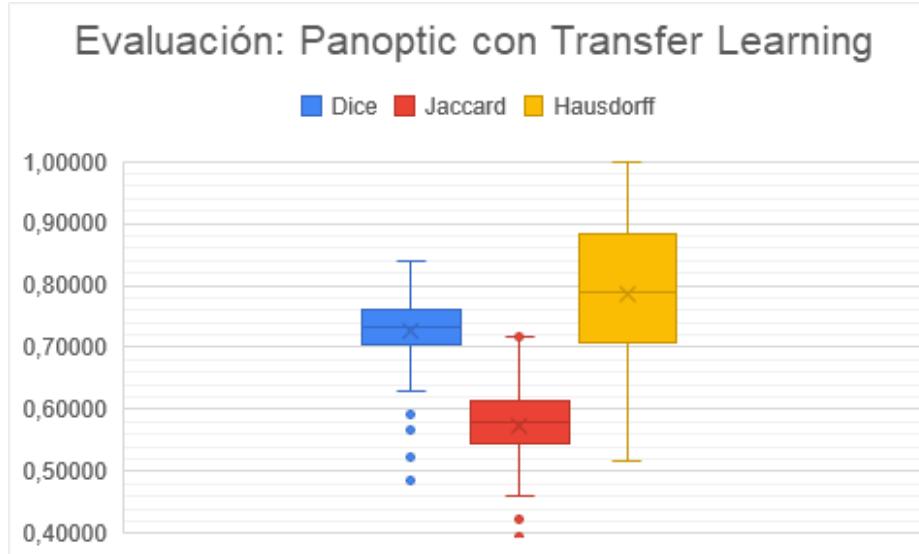


Figura 5.7: Resultados del modelo Panoptic con transfer learning. Línea central dentro de las cajas indica Promedio (Dice: 0,72809, Jaccard:0,57412, Haussdorf: 0,78644)

**Fuente:** Elaboración propia, 2020

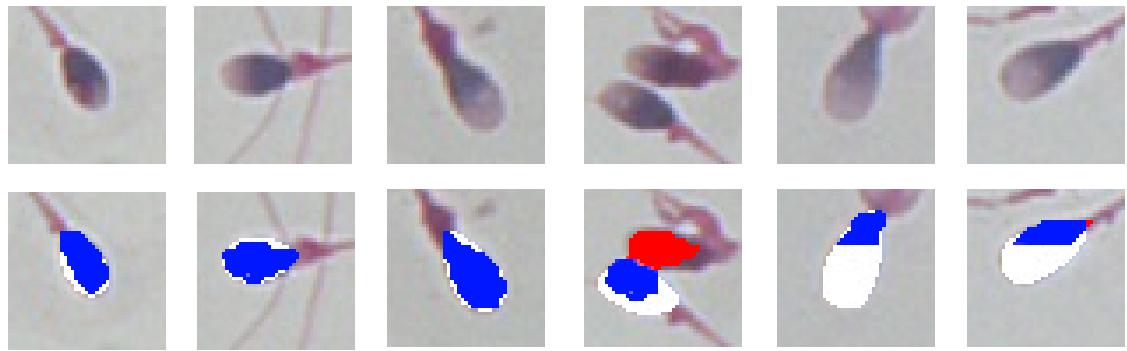


Figura 5.8: Superposición entre muestras (Imagen RGB), máscaras de referencia (Color Blanco) y máscaras predecidas por modelo Panoptic con *transfer learning* (Color Azul) con las células con mayor y menor coeficiente de Dice. El color rojo representa partes que el modelo predijo, pero que no existen en la máscara de referencia. En la primera fila se pueden ver las imágenes originales, y en la segunda fila el proceso de superposición. Los valores de Dice para cada imagen son, de izquierda a derecha: **0.8216, 0.8383, 0.8347, 0.3814, 0.382, 0.486**.

**Fuente:** Elaboración propia, 2020

## CAPÍTULO 6. ANÁLISIS DE RESULTADOS

Luego de que los experimentos han sido realizados, y sus resultados expuestos, es posible iniciar un proceso de análisis de estos, debatiendo sobre el por qué de los resultados logrados, y su comparación con la última investigación hasta la fecha.

Inicialmente, es importante destacar que durante el proceso de segmentación de cabezas, en todos los experimentos realizados, se genera un número importante de falsos positivos, los cuales, representan predicciones de espermatozoides humanos que no se encuentran en el *ground truth*. Esto no es un problema del uso de la herramienta Deep Cell, si no que el dataset SCIAN-SpermSegGS no considera algunos espermatozoides como válidos debido a que, desde un aspecto clínico, estas muestras no son espermatozoides válidos, en base a las características presentadas en el Capítulo 2, y en la imagen 2.1. Al tratarse de un trabajo de investigación que busca medir la segmentación de espermatozoides humanos, y no la clasificación de estos, este proceso no afecta en los resultados expuestos en el capítulo 4. Esto también se ve reflejado al comparar los resultados de la última investigación, que también posee 57 células que son falsos positivos al compararlos con el dataset SCIAN-SpermSegGS, lo cual es un número importante a considerar. Estos datos fueron ignorados durante la exposición de resultados.

Considerando todos los experimentos, la métrica de distancia de Hausdorff no evidencia diferencias importantes entre modelos, lo que indica que los modelos generados por Deep Cell, sin importar la arquitectura ni el dataset, logran predecir máscaras en posiciones correctas considerando el tipo de muestra. Sin embargo, al realizar una comparación de los modelos con el método Movahed (Véase Anexo A.1), la distancia de Hausdorff en la segmentación de cabezas posee mejores valores en comparación a los modelos de la herramienta Deep Cell. La explicación para esto se debe a dos factores principales: el pre-procesamiento de imágenes de entrada y la segmentación de partes externas del espermatozoide. La segmentación de partes externas localiza a los espermatozoides de manera global en una muestra, identificando su cabeza, como el resto de las partes, logrando cuadrar perfectamente la cabeza en la posición de la muestra, al contrario de los modelos generados por Deep Cell, que se basan en la extracción de características de la muestra, principalmente, de la tonalidad de gris en una muestra. De igual forma, el método Movahed realiza un preprocesamiento de imágenes: una modificación del contraste y el brillo de la muestra, para apoyar el trabajo de las redes convolucionales encargadas de la segmentación ya mencionada, algo que según el alcance definido, no se realizó.

A partir de este punto, el análisis de resultados se realiza diferenciando los tipos de experimentos principales: desde cero, solo utilizando el dataset SCIAN-SpermSegGS, y aplicando *transfer learning*, con el dataset Hela\_S3 como pieza inicial, para dar paso al dataset SCIAN-

## 6.1 EXPERIMENTO 1

Un aspecto inicial a analizar es el costo de recurso computacional entre Retina Net y Panoptic, siendo este último mayor al primero, principalmente en el uso de memoria volátil. La necesidad de replicar el set de datos, para componer cada una de las clases semánticas definidas, requiere que el equipo computacional posea una gran cantidad de memoria RAM para almacenar los resultados durante el proceso de entrenamiento. En el caso contrario, la necesidad de generar un archivo comprimido de formato .npz, optimizando la carga de imágenes y posterior integración a la red neuronal profunda.

Los resultados generados por los modelos poseen falsos negativos, principalmente durante las primeras iteraciones en todos los experimentos en el proceso de validación cruzada. Las imágenes que más generan falsos negativos son las que poseen células separadas por 1 a 5 píxeles, logrando solamente una segmentación de las dos células. De igual forma, otras células que los modelos no logran predecir son los espermatozoides que no poseen una forma elíptica. En general, la mayoría de los modelos genera entre 4-8 células como falsos negativos. Algunos ejemplos de muestras que poseen estas características se pueden ver en la figura 6.1.

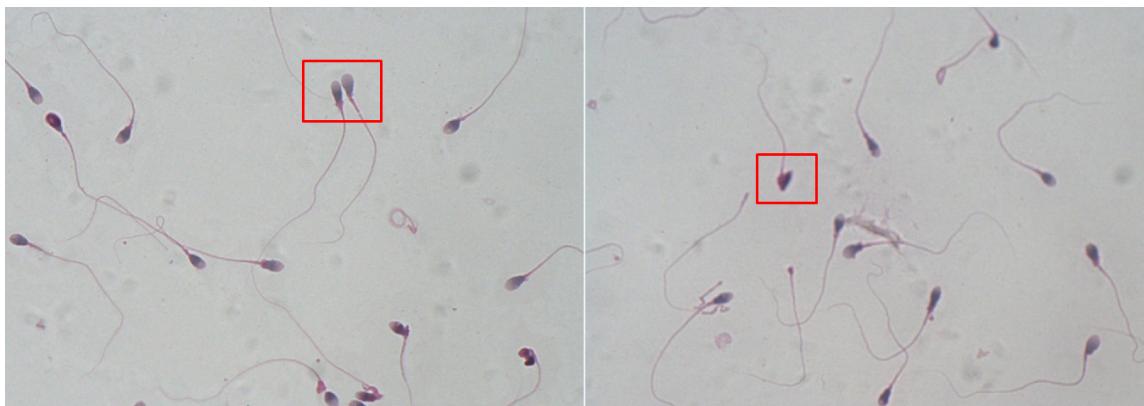


Figura 6.1: Ejemplos de células no detectadas (falsos negativos) durante el experimento 1, con ambas arquitecturas. Estas imágenes en el dataset SCIAN-SpermSegGS corresponden a las imágenes 14 y 2, de izquierda a derecha)

**Fuente:** Elaboración propia, 2020

En base a los resultados expuestos en las tabla 5.1 y 5.2, podemos observar una clara diferencia entre los modelos entrenados en base a la arquitectura Retina Net, y Panoptic, siendo este último peor al primero en aproximadamente, un 10% en las métricas de

Dice y Jaccard. Esto se debe principalmente debido los requisitos necesarios para realizar un entrenamiento con la arquitectura Panoptic, que requiere, en primera instancia, un recorte de las imágenes del dataset SCIAN-SpermSegGS para poseer una razón 1:1, sumado a la eliminación de muestras válidas debido al generar células válidas para entrenar el modelo. En la mayoría de los casos, el recorte a 512x512 generaba que algunas células se vieran cortadas entre el recuadro generado, y estas eran eliminadas mediante un *script* propio. En cambio, al trabajar con una arquitectura RetinaNet, las imágenes no requerían de un nuevo procesamiento de datos luego del aumento de datos para el entrenamiento. Esto no tiene que ver particularmente con el uso de la herramienta, ya que, durante las pruebas para la obtención de parámetros de entrenamiento, la función de pérdida de ambos modelos (Dice), lograba un resultado similar, llegando a 0.043, utilizando un dataset de tamaño reducido.

Al comparar estos dos modelos con el método Movahed, cuyos resultados del mismo dataset SCIAN-SpermSegGS se encuentran en la tabla A.1, podemos ver claras diferencias. Primero, las medidas de tendencia central del método Movahed son mejores, tanto a los resultados del modelo en base a Retina Net, por cerca de 4 puntos porcentuales, y a Panoptic, por 15. Ahora, si vemos particulares por célula, y se plantea con una función de densidad de probabilidades, podemos observar la dispersión entre los modelos del experimento 1 y el estado del arte. Mientras que el modelo de Retina Net entrega resultados aproximados en el intervalo [0.6, 0.97], los resultados con Panoptic se ven reflejados en el intervalo [0.4, 0.90], lo cual es muy bajo al compararlos con el modelo Movahed, con el intervalo [0.8, 0.99]. Para ver este resultado de manera visual, se visualiza la distribución normal de datos en una función de densidad de probabilidad, en la figura 6.2.

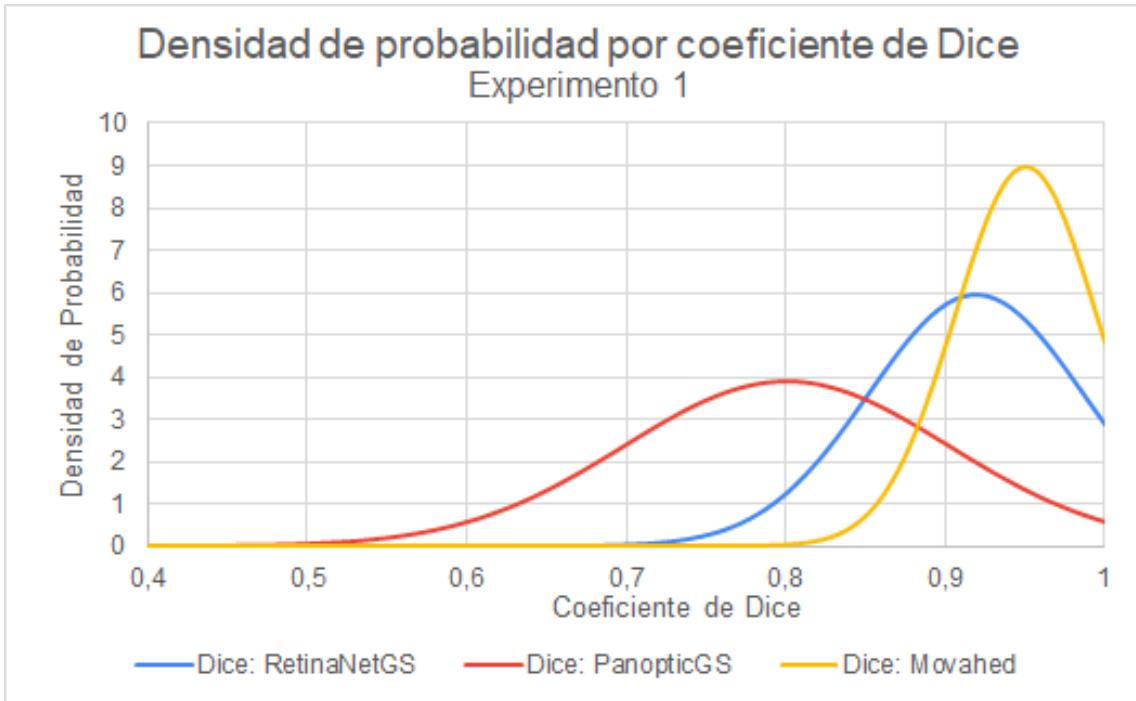


Figura 6.2: Función de densidad de probabilidad para coeficiente de Dice con un modelo Retina Net (en azul), Panoptic (línea roja) y el último método del estado del arte, el método Movahed (línea amarilla) en el experimento 1.

**Fuente:** Elaboración propia, 2020

## 6.2 EXPERIMENTO 2

Al igual que el experimento 1, la utilización de recursos computacionales es un aspecto importante a la hora de realizar de analizar el rendimiento y los resultados de una red neuronal profunda. En este caso, el uso de recursos es aún mayor que el primer experimento, debido a la ampliación de las imágenes del dataset Hela\_S3 a 512x512 en el modelo Panoptic y a 580x780, en el modelo Retina Net. Esto conlleva a que 7000 imágenes en total aumentaron su tamaño a más del doble para el modelo, lo cual eleva la carga de memoria RAM al doble durante el desarrollo de la primera y segunda iteración. En el caso de Panoptic, este aumento lleva los recursos al límite, utilizando cerca de 122 GB's de RAM, de los 128 disponibles. En Retina Net, el uso de recursos es menor, llegando a los 105 GB's. Esto significa que, en caso de realizar una modificación de parámetros de entrenamiento o al número de imágenes, es importante considerar estos recursos.

Al contrario del primer experimento, los resultados generados a partir de un modelo de Panoptic con *transfer learning* no generan falsos negativos, es decir, el modelo detecta todos

las células de referencia del dataset SCIAN-SpermSegGS, incluyendo los casos particulares expresados durante el experimento 1. La razón principal de esto es la disposición de células en el dataset Hela\_S3, como al funcionamiento de la red Panoptic; en el dataset Hela\_S3, las células se encuentran distancias por pocos píxeles, algo que pasa en los casos vistos en la imagen 6.1, además, respecto al funcionamiento, la segmentación de instancia obtiene la forma esférica de las primeras células, y bajo esa base, se adapta a las cabezas de espermatozoides a medida que se realiza el proceso de validación cruzada, diferenciando cada una de las células. En caso contrario, el modelo Retina Net con *transfer learning* no presenta mejoría en comparación a su contraparte, generando exactamente los mismos falsos negativos.

Al igual que los resultados expuestos en el experimento 1, los modelos de Retina Net poseen mejores resultados que los modelos de Panoptic, con una mejora aproximada de 10 puntos en el coeficiente de Dice, y un 25 en el coeficiente de Jaccard. Las razones de esta diferencia son exactamente las mismas que en el primer experimento, la necesidad de reconstruir los datos para poseer una determinada proporción en un modelo con arquitectura Panoptic a un tamaño determinado, buscando el equilibrio entre un gasto computacional alcanzable por el servidor (4500 imágenes de 1024x1024 requiere mucha capacidad en memoria volátil) y no modificar el tipo de segmentación a una segmentación por partes, lo cual produciría cambios importante al alcance de este trabajo de investigación. El caso del coeficiente de Jaccard, sin embargo, se debe a que el modelo Panoptic con *transfer learning*, predice las máscaras en un posición diferente que su contraparte con Retina Net; eso se produce por la diferencia en los datos durante el entrenamiento del modelo inicial (con Hela\_S3), siendo que el entrenamiento en un modelo Panoptic con *transfer learning* se realiza con imágenes más pequeñas de la máscara de referencia y un número menor de células válidas por muestra en comparación con Retina Net, generando incongruencias en la posición de células, sobretodo en posteriores iteraciones de validación cruzada, afectando la intersección entre la máscara generada a partir de la predicción y la máscara de referencia. En otras palabras, el uso de *transfer learning* impacta negativamente en el uso de Panoptic.

Si se compara el estado del arte con los resultados obtenidos por este experimento, se puede verificar que los resultados son inferiores al método Movahed (Véase Tabla A.2), al igual que lo sucedido en el primer experimento, con un coeficiente de Dice 6 puntos menor con Retina Net, y 25 con Panoptic. Si se quiere analizar la distribución de los resultados por célula, podemos ver que los valores de Panoptic y Retina Net se encuentran dispersos de manera similar, pero este último posee mejores resultados con las células con un coeficiente de Dice en el intervalo [0.8,0.95], en comparación con los resultados de Panoptic (intervalo [0.55,0.85]). Sin embargo, la dispersión de datos, y los resultados por células son mejores con el método Movahed, tal como se visualiza en la imagen 6.3.

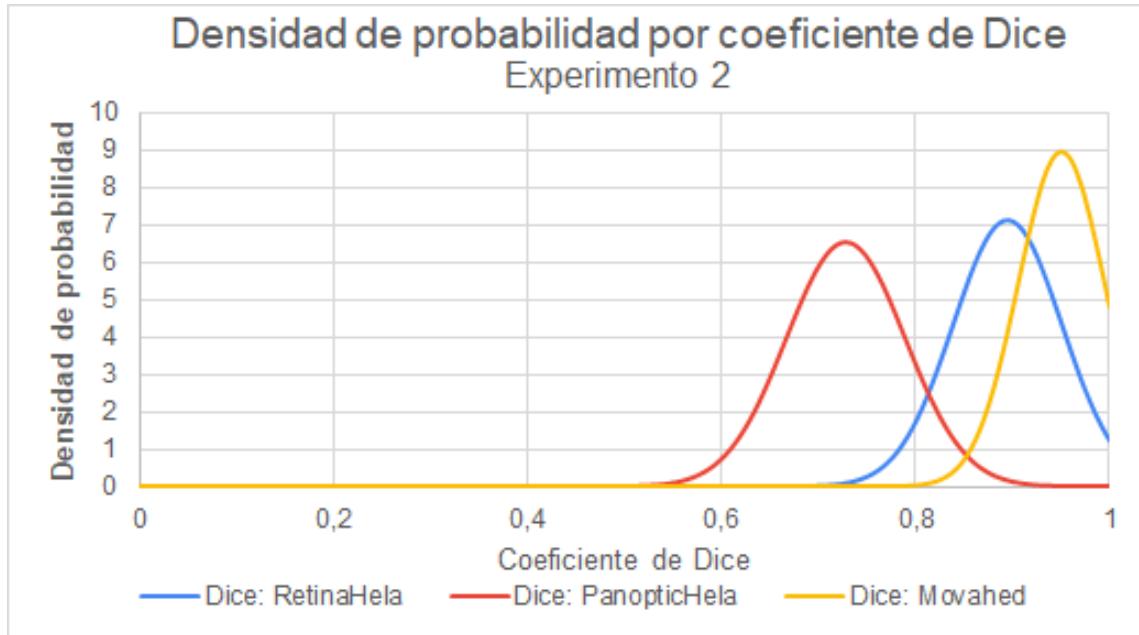


Figura 6.3: Función de densidad de probabilidad para coeficiente de Dice con un modelo Retina Net (en azul), Panoptic (línea roja) y el último método del estado del arte, el método Movahed (línea amarilla) en el experimento 2.

**Fuente:** Elaboración propia, 2020

### 6.3 ANÁLISIS ENTRE EXPERIMENTOS

Al comparar los resultados de los modelos Panoptic en ambos experimentos, se visualiza que poseen las métricas de evaluación más bajas de todos los modelos, en especial el modelo con *transfer learning* en la métrica de Jaccard. Esto se debe a la baja calidad de las imágenes generadas, particularmente en los bordes. La segmentación semántica se encarga de diferenciar el fondo de la imágenes que se requiere segmentar, y se une con la segmentación de instancia para entregar un resultado final. Como las imágenes del dataset SCIAN-SpermSegGS no poseen un fondo similar entre imágenes, con diferencias en la tonalidad y brillo del fondo, la segmentación semántica no puede realizar un predicción correcta de los fondos, y al unirse con la segmentación de instancia, hace que los acrosomas (que son más claros que el espermatozoide en general) sea considerada como fondo, generando un resultado insatisfactorio, el cual empeora si se aplica *transfer learning*, debido principalmente al fondo negro que poseen las imágenes de Hela\_S3, todo lo contrario a las imágenes del dataset SCIAN-SpermSegGS. Estos resultados se ven en la imagen 6.4. La solución a este problema sería la regulación del fondo del dataset, a través de un preprocesamiento de las imágenes, algo fuera del alcance definido de este trabajo

de investigación.

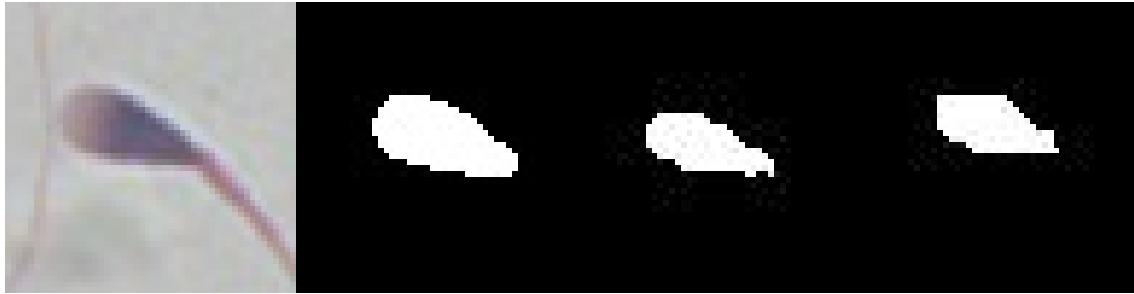


Figura 6.4: Comparación de resultados entre métodos Panoptic De izquierda a derecha: Imagen RGB original. Máscara de referencia, Máscara generada a partir del modelo Panoptic desde cero. Máscara generada a partir del modelo Panoptic con Transfer Learning. Imagen 19 del dataset SCIAN-SpermSegGS.

**Fuente:** Elaboración propia, 2020

En relación con los modelos de Retina Net, los valores de las métricas de evaluación son similares entre experimentos, siendo más bajas los modelos con *transfer learning*. Al analizar los datos por célula, los resultados de las primeras iteraciones de los experimentos con *transfer learning* son menores a su contraparte desde cero, aproximadamente en 3 a 6 puntos conceptuales. A medida que se desarrollan más iteraciones, y el modelo se va haciendo más robusto, los valores de las métricas de evaluación son similares entre experimentos, principalmente a lo diferentes que son las muestras de Hela\_S3 y SCIAN-SpermSegGS. Con esta información, podemos afirmar que el proceso de *transfer learning* con el dataset Hela\_S3 afecta el rendimiento del modelo con arquitectura Retina Net. Esto se puede ver en la figura 6.5 con un ejemplo particular.

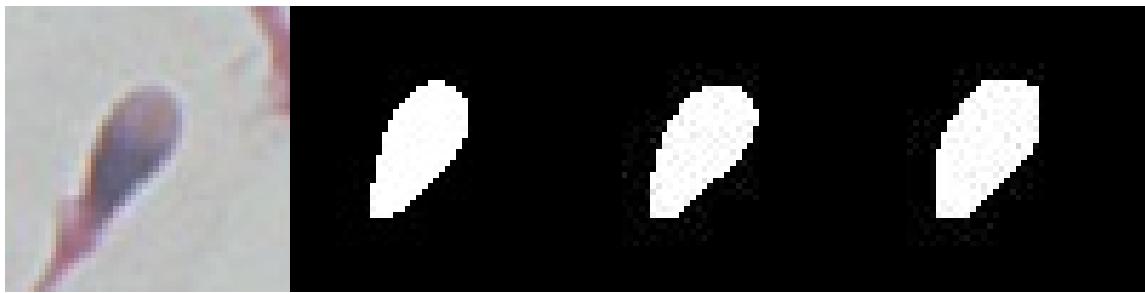


Figura 6.5: Comparación de resultados entre métodos Retina Net. De izquierda a derecha: Imagen RGB original. Máscara de referencia, Máscara generada a partir del modelo Retina Net desde cero. Máscara generada a partir del modelo Retina Net con *transfer learning* en primera iteración de validación cruzada .Imagen 15 del dataset SCIAN-SpermSegGS.

**Fuente:** Elaboración propia, 2020

Finalmente, a la hora de comparar todos los modelos generados por ambos

experimentos y el estado del arte, usando como parámetros las métricas de evaluación explicadas en este documento en base al cálculo de todas las células de dataset SCIAN-SpermSegGS., se obtiene el gráfico expuesto en la figura 6.6, demostrando que ningún modelo generado con la herramienta Deep Cell, bajo este ambiente experimental, supera a la investigación actual, en cuanto a segmentación automática de cabezas de espermatozoides humanos.

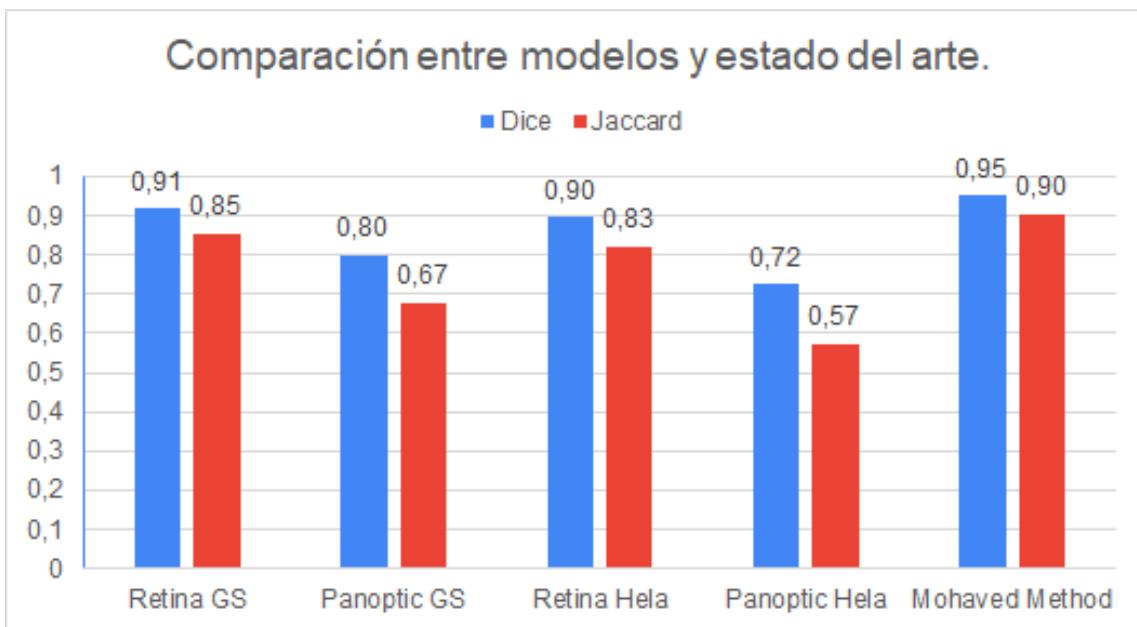


Figura 6.6: Comparación entre resultados de modelos generados por Deep Cell y el método Movahed (Movahed et al., 2019) que segmentan automáticamente cabezas de espermatozoides humanos.

**Fuente:** Elaboración propia, 2020

## CAPÍTULO 7. CONCLUSIONES

Luego de la exposición de resultados, y su posterior discusión, es posible volver al origen de esta memoria y trabajo de investigación a responder la hipótesis planteada: **La herramienta Deep Cell mejora la precisión en la segmentación automatizada de partes de espermatozoides humanos.** Frente a la evidencia planteada en este trabajo de investigación, se **rechaza la hipótesis** para la segmentación automática de cabezas de espermatozoides, debido a que no supera la precisión en comparación con los últimas investigaciones en el área. Sin embargo, es importante mencionar algunas cosas relacionadas a este estamento.

### 7.1 OBJETIVOS ESPECÍFICOS

En cuanto al cumplimiento de los tres objetivos específicos mencionados en este documento, se realizaron en su totalidad, cada uno entregando resultados que aportaron la definición de la hipótesis planteada.

#### 7.1.1 Incremento de datos

El incremento de datos es considerado un objetivo cumplido en su totalidad, logrando un número total de 4500 imágenes, que, a pesar de ser generados de manera artificial, siguen siendo válidas como muestras médicas, con el uso de métodos simples con bases en transformaciones geométricas y fotométricas. Este aumento de datos representa casi un aumento de un 23500% en comparación con el dataset SCIAN-SpermSegGS original. Al mismo tiempo, como el número de imágenes fue suficiente, no se produjo overfitting en los modelos durante su entrenamiento, sin importar la técnica a utilizar.

#### 7.1.2 Entrenamiento de modelos con Deep Cell 2.0

El proceso de pruebas realizado antes del entrenamiento del modelo permitió que este fuera un entrenamiento controlado y completado en su totalidad, generando errores bajos (alrededor de 0.04-0.06) para trabajar con un dataset nunca antes usado en una red neuronal profunda, generando mejoras consistentes en cada época de entrenamiento. Al mismo tiempo,

el uso de validación cruzada disminuye más el error del modelo, generando modelos robustos con un número de iteraciones poco frecuentes, escogido principalmente por temas de tiempo. En cuanto al entrenamiento con *transfer learning*, el preprocesamiento realizado a las imágenes antes del entrenamiento también es considerado un éxito, debido a la necesidad de los modelos Panoptic de poseer una proporción 1:1, con un tamaño potencia de 2, se mantuvo un equilibrio entre los recursos computacionales utilizables, y el alcance de la investigación, permitiendo una segmentación de imagen completa, y no por partes.

### 7.1.3 Análisis de técnicas y parámetros

La definición temprana de métricas adecuadas para la segmentación permite entregar resultados claros, y realizar comparaciones entre los 4 modelos generados a partir de la herramienta Deep Cell. Y, como se planteó en el inicio de este documento, esto permite entregar importantes conclusiones sobre el funcionamiento de la herramienta bajo este ambiente experimental, demostrando, por ejemplo, que los modelos con arquitectura Retina Net entregan mejores valores de coeficiente de Dice en comparación con los modelos de Panoptic, y que la aplicación de *transfer learning* con el dataset Hela\_S3 empeora la calidad de las máscaras en ambas arquitecturas, sobretodo en Panoptic. Esto permite a otros investigadores saber qué hacer en caso de utilizar un ambiente experimental similar en la creación de modelos de aprendizaje profundo con el dataset SCIAN-SpermSegGS u otro que comparta semejanzas.

## 7.2 EXPERIENCIA

El principal problema que afectó a este trabajo de investigación fue la falta de un sistema computacional estable lo suficientemente potente para realizar el entrenamiento de redes neuronales profundas utilizando la herramienta Deep Cell, la cual depende principalmente del uso excesivo de GPU, requiriendo unidades con una cantidad de VRAM superior a 8GB en adelante, disminuyendo el espectro de equipos utilizables. La falla en la refrigeración de la sala de servidores en la universidad, que inició en el mes de abril, hasta los primeros días del mes de octubre, sumado a cortes de luz y al contexto de estar viviendo una emergencia sanitaria debido a la pandemia COVID-19, y todos los efectos que esto conlleva, hizo imposible poseer un sistema computacional capaz de realizar el entrenamiento de la herramienta con Deep Cell durante el tiempo de desarrollo de este trabajo de investigación, a pesar de respetar la planificación realizada

durante el curso de Seminario de Informática.

Acorde a lo anterior, la herramienta Deep Cell es una herramienta con menos de dos años de antigüedad y que se encuentra en la versión 0.6.0, liberada el día 12 de Septiembre del presente año. Esto conlleva que la herramienta no posee una documentación establecida más que algunos ejemplos oficiales y comentarios en el código fuente, generados de manera automática como documentación por el servicio que aloja la herramienta, *GitHub* y *ReadTheDocs*. Esto deja poco espacio para resolver problemas durante el uso de la plataforma, expandir su uso, generar pruebas trascendentales o informar al usuario sobre qué tipo de set de datos tienen un mayor potencial para la plataforma, a diferencia de tecnologías que son base de Deep Cell como Keras y Tensorflow. Además, al tratarse de una herramienta tan nueva, es complicado enfrentarse a errores, fallas, requisitos, y su funcionamiento en general durante el desarrollo de este trabajo de investigación, siendo la única opción realizar un contacto con los creadores de Deep Cell, lo cual, toma tiempo considerable. Por ejemplo, uno de los mayores problemas ocurridos con esta premisa fue el formato de entrada de las imágenes de entrada, ya que ni la documentación, ni los ejemplos, mostraban cuál era el formato particular las muestras, y sus correspondientes máscaras. Esto, sumado a que el dataset SCIAN-SpermSegGS no tenía un formato correcto desde el inicio (máscaras no normalizadas, sin etiquetas para diferenciar células) género muchos retrasos en el proceso, lo cual se soluciona mediante conversaciones con los creadores de la herramienta. Otro problema similar fue la incongruencia de versiones entre documentación y ejemplos, ya que la versión que recomendaban los autores era distinta a la que utilizaban los ejemplos; esto se solucionó mediante la publicación y resolución de un *issue* en el repositorio de la herramienta<sup>2</sup>. A pesar de mencionado anteriormente, la herramienta Deep Cell ha demostrado ser una buena alternativa en un área poco frecuente como la segmentación y seguimiento en vídeo de organismos unicelulares, como se ve reflejado en esta y otras publicaciones (Caicedo et al., 2019), por lo que su uso, sumado a ser una herramienta novedosa y prometedora, es considerado en la investigación de la segmentación automatizada de espermatozoides humanos como un avance.

De igual manera, la forma de instalación de la herramienta sugerida por los creadores consiste en el uso de una imagen de Docker para poseer todos los elementos necesarios, además de la instalación de una serie de controladores para utilizar el hardware. La instalación de Docker, principalmente en un servidor remoto, es un proceso lento y complejo, principalmente por aspectos de seguridad, consumo de recursos, y el uso de permisos de *root* para funcionar, algo muy difícil de obtener en un acceso remoto de, por ejemplo, una empresa. Estos 3 factores llevaron a la decisión de descartar la realización de segmentación automática utilizando Deep Cell de las partes restantes del espermatozoide humano, como se planteo en el curso de Seminario

---

<sup>2</sup><https://github.com/vanvalenlab/deepcell-tf/issues/400>

de Informática: núcleo, acrosoma, pieza media y cola.

### 7.3 TRABAJO FUTURO

A pesar de que los modelos generados por la herramienta Deep Cell no sufrieron de sobreajuste u *overfitting*, el número de imágenes, según lo planteado en el área de investigación de la segmentación de espermatozoides humanos (Nissen et al., 2017), no es suficiente. El gran limitante para esto, durante el desarrollo de esta memoria, fue la falta de un servidor o equipo computacional lo suficientemente potente para almacenar un número mayor de imágenes, y ser utilizado con la herramienta Deep Cell. Este aumento en el número de datos puede generar que los resultados mejoren. Así mismo, tal como se plantea en la sección 2.2.3, el uso de técnicas de inteligencia artificial y redes profundas para realizar el incremento de datos es una posibilidad que puede mejorar el rendimiento de una red. Por eso, es considerado trabajo futuro la realización del entrenamiento utilizando la herramienta Deep Cell con un número mayor de datos, como también generados a partir de otros métodos, como por ejemplo, *GAN* (Ver Glosario)

Las últimas investigaciones en el área (Movahed et al., 2019) hablan sobre la importancia de realizar un proceso de preprocesamiento al dataset SCIAN-SpermSegGS para funcionar en una red convolucional no profunda. Durante el desarrollo de este trabajo de investigación, tal como se planteó en el alcance, no se realizó ningún proceso de pre-procesamiento de imágenes al dataset, sin considerar el aumento de datos para el entrenamiento. Tal como mencionan los autores, esto debe mejorar los resultados de la segmentación en una red neuronal convolucional, o de carácter profundo. La realización del pre-procesamiento de imágenes del dataset SCIAN-SpermSegGS es considerado un trabajo futuro para el desarrollo de la investigación.

Finalmente, es importante cuestionarse cuáles serían los resultados de estos mismos modelos bajo el mismo ambiente experimental, sin limitarse a la herramienta Deep Cell, para verificar cual es el impacto del uso de esta herramienta en la segmentación de cabezas de espermatozoides humanos utilizando este mismo dataset. El trabajo a realizar consiste en realizar la misma hipótesis, pero utilizando otras herramientas y/o una implementación propia de una red neuronal profunda.

## 7.4 OTRAS REFLEXIONES

Uno de los aspectos principales de la herramienta Deep Cell, es que todos sus ejemplos se basan en el uso del dataset Hela\_S3, mostrando resultados favorables en comparación con otras herramientas. Pero, hablando específicamente del tipo de imágenes, y sus características, el dataset Hela\_S3 y el dataset SCIAN-SpermSegGS son diferentes. Sin considerar la dimensionalidad, las imágenes del set de datos Hela\_S3 poseen un fondo de tonalidad negra, en cambio, en el dataset SCIAN-SpermSegGS esto es a la inversa. Además, diversos usuarios de la herramienta reportan problemas al utilizar otros datos, aportados por la plataforma como también ajenos. Esto lleva a la conclusión de que la herramienta está moldeada a un determinado formato, y la adaptación de modelos a este tipo puede mejorar los resultados, sobretodo al realizar el proceso de *transfer learning*.

En retrospectiva, esta memoria significó un gran aporte al estudiante en todo lo relacionado a la creación de redes neuronales profundas, debido a la necesidad de entender el ambiente y preparación que es necesario para generar un modelo de inteligencia artificial. Al mismo tiempo, lo nuevo de la herramienta Deep Cell, y su falta de documentación, significó dedicar un lapso de tiempo importante para entender el código fuente, la creación de modelos, y el funcionamiento de redes neuronales profundas especializadas en segmentación utilizando *tensorflow* y *keras*, módulos base de Deep Cell. Este proceso pudo generar conocimiento en el funcionamiento de las capas de un modelo de aprendizaje profundo, el proceso de aprendizaje mediante una función de error, el proceso de experimentación previo a la hora de escoger, visión por computador, incremento aleatorio de datos, entrenamiento de una red neuronal profunda, entre otros.

## GLOSARIO

- **Aprendizaje Profundo (Deep Learning):** Una rama de la inteligencia artificial que, mediante el entrenamiento de redes neuronales con grandes cantidades de datos, permiten, entre otras cosas, resolver problemas en base a la experiencia, encontrar características, entre otros. Una de las principales aplicaciones en esta área es el reconocimiento de características en imágenes médicas, reconociendo anomalías, patrones, falencias, etcétera. (Bakator & Radosav, 2018).
- **Gold Standard (GS):** Este concepto viene de la medicina, y, por definición, “una técnica diagnóstica que define la presencia de la condición con la máxima certeza conocida”. Es utilizado como benchmark, o regla a seguir (de ahí su nombre) frente a cualquier test diagnóstico. Comúnmente se realiza frente a que no existe una prueba con datos utilizables en algún área de investigación relacionada a las ciencias médicas. (Salech et al., 2008). Particularmente en este informe, cuando se hable del uso del Gold Standard, se referirá a SCIAN-SpermSegGS.
- **Área de Interés (ROI):** Concepto utilizado principalmente en el manejo de imágenes, pero también muy aplicado a la investigación celular. Corresponde a una porción de una imagen, en la cual se realizará una acción.
- **Función de error de los mínimos cuadrados (MSE):** Función utilizada en inteligencia artificial para determinar el error. Se define como la diferencia entre el promedio de los píxeles originales y los predecidos. Un MSE alto indica un gran error entre las imágenes predichas y el original, por lo que es necesario modificar pesos donde el error sea más grande. (Grover, 2020).
- **Función de error “*weighted categorical cross entropy loss*”:** Función utilizada para calcular el error en redes neuronales, principalmente aquellas que utilizan segmentaciones semánticas. Es una variante de la función “*cross entropy loss*”, que corresponde a la diferencia entre dos distribuciones probabilísticas para un set de datos. Esta variación utiliza distribuciones categóricas (también conocida como distribución de Bernoulli) y posee un coeficiente que agrega más importancia o peso a una distribución, por lo que este método suele ser utilizada cuando los datos se encuentran sesgados (Jadon, 2020).
- **Distancias Semánticas Interior y Exterior:** Conceptos dentro del concepto de Distancia Euclíadiana en  $R^2$ , que hablan sobre la distancia más pequeña y más grande que existe desde el centro de un forma, hasta su perímetro. Este concepto se utiliza bastante como método de clasificación, detección y segmentación de objetos en el uso de redes neuronales convolucionales (Ling & Jacobs, 2007), sobre todo en segmentadores de instancia como es el modelo Panoptic.
- **GAN:** Sistema de redes neuronales de aprendizaje profundo utilizada en la creación de datos, principalmente imágenes, sonidos o textos, que se basa en la constante lucha y aprendizaje entre dos redes contrarias. Una red genera datos específicos, siendo revisados por una red discriminadora, que determina la calidad de la imagen. Este intercambio de información, hace que las redes mejoren en sus respectivas tareas, generando una red que genere datos de calidad, y otra que permite diferenciar pequeños detalles en otras muestras (Merino, 2019).

## REFERENCIAS BIBLIOGRÁFICAS

- Adé-Damilano, M. (2008). Gametogenesis. In *embryology.ch*. Sweden.  
URL <http://www.embryology.ch/anglais/cgametogen/spermato01.html>
- Agarwal, A., Mulgund, A., Hamada, A., & Chyatte, M. R. (2015). A unique view on male infertility around the globe. *Reproductive Biology and Endocrinology*, 13(1), 37.  
URL <http://www.rbej.com/content/13/1/37>
- Al-Kofahi, Y., Zaltsman, A., Graves, R., Marshall, W., & Rusu, M. (2018). A deep learning-based algorithm for 2-D cell segmentation in microscopy images. *BMC Bioinformatics*, 19(1), 365.  
URL <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-018-2375-z>
- Amann, R. P. (2008). The Cycle of the Seminiferous Epithelium in Humans: A Need to Revisit. *Journal of Andrology*, 29(5), 469–487. Eprint:  
<https://onlinelibrary.wiley.com/doi/pdf/10.2164/jandrol.107.004655>  
URL <https://onlinelibrary.wiley.com/doi/abs/10.2164/jandrol.107.004655>
- Anwar, S. M., Majid, M., Qayyum, A., Awais, M., Alnowami, M., & Khan, M. K. (2018). Medical Image Analysis using Convolutional Neural Networks: A Review. *Journal of Medical Systems*, 42(11), 226.  
URL <https://doi.org/10.1007/s10916-018-1088-1>
- Applied Maths BioNumerics (2019). *Difference between Jaccard and Dice coefficient*. Publication Title: Applied Maths NV Type: Q&A.  
URL <https://www.applied-maths.com/knowledge-base/difference-between-jaccard-and-dice-coefficient>
- ASALE, R., & RAE (2020). morfología | Diccionario de la lengua española. Library Catalog: dle.rae.es.  
URL <https://dle.rae.es/morfología>
- Auger, J., Jouannet, P., & Eustache, F. (2016). Another look at human sperm morphology. *Human Reproduction*, 31(1), 10–23.  
URL <https://academic.oup.com/humrep/article-lookup/doi/10.1093/humrep/dev251>
- AWS (2016). Validación cruzada - Amazon Machine Learning.  
URL [https://docs.aws.amazon.com/es\\_es/machine-learning/latest/dg/cross-validation.html](https://docs.aws.amazon.com/es_es/machine-learning/latest/dg/cross-validation.html)
- Bai, M., & Urtasun, R. (2017). Deep Watershed Transform for Instance Segmentation. *arXiv:1611.08303 [cs]*. ArXiv: 1611.08303.  
URL <http://arxiv.org/abs/1611.08303>
- Bajcsy, P., Cardone, A., Chalfoun, J., Halter, M., Juba, D., Kocielek, M., Majurski, M., Peskin, A., Simon, C., Simon, M., Vandecrème, A., & Brady, M. (2015). Survey statistics of automated segmentations applied to optical imaging of mammalian cells. *BMC Bioinformatics*, 16(1), 330.  
URL <https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-015-0762-2>
- Bakator, M., & Radosav, D. (2018). Deep learning and medical diagnosis: a review of literature. *Multimodal Technologies and Interaction*, 2(3), 47.  
URL <http://www.mdpi.com/2414-4088/2/3/47>
- Balaji, K., & Lavanya, K. (2019). Chapter 5 - Medical Image Analysis With Deep Neural Networks. In A. K. Sangaiah (Ed.) *Deep Learning and Parallel Computing Environment for Bioengineering Systems*, (pp. 75–97). Academic Press.  
URL <http://www.sciencedirect.com/science/article/pii/B9780128167182000129>

- Bannon, D., Moen, E., Borba, E., Ho, A., Camplisson, I., Chang, B., Osterman, E., Graf, W., & Van Valen, D. (2018a). DeepCell 2.0: Automated cloud deployment of deep learning models for large-scale cellular image analysis. *Tech. rep.*
- Bannon, D., Moen, E., Schwartz, M., Borba, E., Cui, S., Huang, K., Camplisson, I., Koe, N., Kyme, D., Kudo, T., Chang, B., Pao, E., Osterman, E., Graf, W., & Van Valen, D. (2018b). Dynamic allocation of computational resources for deep learning-enabled cellular image analysis with Kubernetes. *preprint, Systems Biology*.  
URL <http://biorxiv.org/lookup/doi/10.1101/505032>
- Bannon, D., Moen, E., Schwartz, M., Borba, E., Cui, S., Huang, K., Camplisson, I., Koe, N., Kyme, D., Kudo, T., Chang, B., Pao, E., Osterman, E., Graf, W., & Van Valen, D. (2018c). Dynamic allocation of computational resources for deep learning-enabled cellular image analysis with Kubernetes. *preprint, Systems Biology*.  
URL <http://biorxiv.org/lookup/doi/10.1101/505032>
- Bannon, D., Moen, E., Schwartz, M., Borba, E., Cui, S., Huang, K., Camplisson, I., Koe, N., Kyme, D., Kudo, T., Chang, B., Pao, E., Osterman, E., Graf, W., & Van Valen, D. (2018d). Dynamic allocation of computational resources for deep learning-enabled cellular image analysis with Kubernetes. *preprint, Systems Biology*.  
URL <http://biorxiv.org/lookup/doi/10.1101/505032>
- Brazil, C. (2010). Practical semen analysis: from A to Z. *Asian Journal of Andrology*, 12(1), 14–20.  
URL <http://www.asiaandro.com/Abstract.asp?doi=10.1038/aja.2008.51>
- Brosch, T., Tang, L. Y. W., Yoo, Y., Li, D. K. B., Traboulsee, A., & Tam, R. (2016). Deep 3d convolutional encoder networks with shortcuts for multiscale feature integration applied to multiple sclerosis lesion segmentation. *IEEE Transactions on Medical Imaging*, 35(5), 1229–1239.  
URL <http://ieeexplore.ieee.org/document/7404285/>
- Brosch, T., Yoo, Y., Tang, L. Y. W., Li, D. K. B., Traboulsee, A., & Tam, R. (2015). Deep convolutional encoder networks for multiple sclerosis lesion segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, vol. 9351, (pp. 3–11). Cham: Springer International Publishing.  
URL [http://link.springer.com/10.1007/978-3-319-24574-4\\_1](http://link.springer.com/10.1007/978-3-319-24574-4_1)
- Brownlee, J. (2017). A Gentle Introduction to Transfer Learning for Deep Learning.  
URL <https://machinelearningmastery.com/transfer-learning-for-deep-learning/>
- Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2020). Albumentations: Fast and Flexible Image Augmentations. *Information*, 11(2), 125. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.  
URL <https://www.mdpi.com/2078-2489/11/2/125>
- Caicedo, J. C., Roth, J., Goodman, A., Becker, T., Karhohs, K. W., Broisin, M., Molnar, C., McQuin, C., Singh, S., Theis, F. J., & Carpenter, A. E. (2019). Evaluation of deep learning strategies for nucleus segmentation in fluorescence images. *Cytometry Part A*, 95(9), 952–965.  
URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/cyto.a.23863>
- Carrillo, H., Villarreal, J., Sotaquira, M., Goelkel, A., & Gutierrez, R. (2007). A Computer Aided Tool for the Assessment of Human Sperm Morphology. *2007 IEEE 7th International Symposium on BioInformatics and BioEngineering*, (pp. 1152 – 1157).
- Cartas-Ayala, A. (2011). *Gradient Vector Flow Snakes*. Publication Title: Gradient Vector Flow Snakes Type: Document.  
URL [http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/AV1011/cartas.pdf](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV1011/cartas.pdf)

- Chang, V., Garcia, A., Hitschfeld, N., & Härtel, S. (2017). Gold-standard for computer-assisted morphological sperm analysis. *Computers in Biology and Medicine*, 83, 143–150.  
URL <https://linkinghub.elsevier.com/retrieve/pii/S0010482517300525>
- Chang, V., Saavedra, J. M., Castañeda, V., Sarabia, L., Hitschfeld, N., & Härtel, S. (2014). Gold-standard and improved framework for sperm head segmentation. *Computer Methods and Programs in Biomedicine*, 117(2), 225–237.  
URL <https://linkinghub.elsevier.com/retrieve/pii/S0169260714002521>
- Cheng, B., Collins, M. D., Zhu, Y., Liu, T., Huang, T. S., Adam, H., & Chen, L.-C. (????). Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation. (p. 11).
- Cheng, B., Collins, M. D., Zhu, Y., Liu, T., Huang, T. S., Adam, H., & Chen, L.-C. (2020). Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation. (pp. 12475–12485).  
URL [https://openaccess.thecvf.com/content\\_CVPR\\_2020/html/Cheng\\_Panoptic-DeepLab\\_A\\_Simple\\_Strong\\_and\\_Fast\\_Baseline\\_for\\_Bottom-Up\\_Panoptic\\_CVPR\\_2020\\_paper.html](https://openaccess.thecvf.com/content_CVPR_2020/html/Cheng_Panoptic-DeepLab_A_Simple_Strong_and_Fast_Baseline_for_Bottom-Up_Panoptic_CVPR_2020_paper.html)
- Clare, O., & Jill, A. (2010). *Cell biology for seminars*. Principles of Science. NPG Education 2010.  
URL <http://www.nature.com/scitable/ebooks/cell-biology-for-seminars-14760004>
- Cui, C. S. (2020). *Optimizing Deep Neural Networks for Single Cell Segmentation*. senior\_major, California Institute of Technology.  
URL <https://resolver.caltech.edu/CaltechTHESIS:06052020-210957529>
- Cui, W. (2010). Mother of nothing: the agony of infertility. *Bulletin of the World Health Organization*, 88(12), 881–882.  
URL <http://www.who.int/bulletin/volumes/88/12/10.011210.pdf>
- Deep Cell (2019). *Deep Cell 2.0: Advanced Documentation*. Publication Title: Advanced documentation Type: Documentation.  
URL [https://github.com/vanvalenlab/kiosk/blob/master/docs/ADVANCED\\_DOCUMENTATION.md](https://github.com/vanvalenlab/kiosk/blob/master/docs/ADVANCED_DOCUMENTATION.md)
- Dhanachandra, N., Manglem, K., & Chanu, Y. J. (2015). Image segmentation using k -means clustering algorithm and subtractive clustering algorithm. *Procedia Computer Science*, 54, 764–771.  
URL <https://linkinghub.elsevier.com/retrieve/pii/S1877050915014143>
- Domar, A., Broome, A., Zuttermeister, P., Seibel, M., & Friedman, R. (1993). The prevalence and predictability of depression in infertile women. *International Journal of Gynecology & Obstetrics*, 42(2), 220–221.  
URL <http://doi.wiley.com/10.1016/0020-7292%2893%2990662-G>
- Draelos, R. (2019). Best Use of Train/Val/Test Splits, with Tips for Medical Data. Library Catalog: glassboxmedicine.com.  
URL <https://glassboxmedicine.com/2019/09/15/best-use-of-train-val-test-splits-with-tips-for-medical-data/>
- EcuRed (2015). Células sexuales - EcuRed.  
URL [https://www.ecured.cu/C%C3%A9lulas\\_sexuales](https://www.ecured.cu/C%C3%A9lulas_sexuales)
- Farthing, M. J. G., Edwards, C. R. W., Rees, L. H., & Dawson, A. M. (1982). Male gonadal function in coeliac disease: 1. Sexual dysfunction, infertility, and semen quality. *Gut*, 23(7), 608–614.  
URL <http://gut.bmjjournals.org/cgi/doi/10.1136/gut.23.7.608>
- Forsyth, D., & Ponce, J. (2003). *Computer vision: a modern approach*. Upper Saddle River, N.J. ; London: Prentice Hall.

- Fu, C.-Y., Shvets, M., & Berg, A. C. (2019a). RetinaMask: Learning to predict masks improves state-of-the-art single-shot detection for free. *arXiv:1901.03353 [cs]*. ArXiv: 1901.03353. URL <http://arxiv.org/abs/1901.03353>
- Fu, C.-Y., Shvets, M., & Berg, A. C. (2019b). RetinaMask: Learning to predict masks improves state-of-the-art single-shot detection for free. *arXiv:1901.03353 [cs]*. ArXiv: 1901.03353. URL <http://arxiv.org/abs/1901.03353>
- Gilbert, S. F. (2000). *Developmental biology*. Sunderland, Mass: Sinauer Associates, 6th ed ed.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*.
- Grover, P. (2020). 5 Regression Loss Functions All Machine Learners Should Know. URL <https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0>
- Grégoire, N., & Bouillot, M. (1998). *Hausdorff distances between complex polygons*. Publication Title: Hausdorff distance Type: Page. URL <http://cgm.cs.mcgill.ca/~godfried/teaching/cg-projects/98/normand/main.html>
- Hesamian, M. H., Jia, W., He, X., & Kennedy, P. (2019). Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges. *Journal of Digital Imaging*, 32(4), 582–596. URL <http://link.springer.com/10.1007/s10278-019-00227-x>
- Hussain, Z., Gimenez, F., Yi, D., & Rubin, D. (2018). Differential Data Augmentation Techniques for Medical Imaging Classification Tasks. *AMIA Annual Symposium Proceedings*, 2017, 979–984. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5977656/>
- Jadon, S. (2020). A survey of loss functions for semantic segmentation. *arXiv:2006.14822 [cs, eess]*. ArXiv: 2006.14822. URL <http://arxiv.org/abs/2006.14822>
- Jemila, S. J., & Therese, A. B. (2019). Selection of suitable segmentation technique based on image quality metrics. *The Imaging Science Journal*, 67(8), 475–480. Publisher: Taylor & Francis \_eprint: <https://doi.org/10.1080/13682199.2020.1718298>. URL <https://doi.org/10.1080/13682199.2020.1718298>
- Jensen, T. K., Jacobsen, R., Christensen, K., Nielsen, N. C., & Bostofte, E. (2009). Good semen quality and life expectancy: a cohort study of 43,277 men. *American Journal of Epidemiology*, 170(5), 559–565. URL <https://academic.oup.com/aje/article-lookup/doi/10.1093/aje/kwp168>
- Jordan, J. (2018). An overview of semantic image segmentation. URL <https://www.jeremyjordan.me/semantic-segmentation/>
- Jung, H., Kim, B., Lee, I., Yoo, M., Lee, J., Ham, S., Woo, O., & Kang, J. (2018). Detection of masses in mammograms using a one-stage object detector based on a deep convolutional neural network. *PLOS ONE*, 13(9), e0203355. URL <https://dx.plos.org/10.1371/journal.pone.0203355>
- Kirillov, A., He, K., Girshick, R., Rother, C., & Dollár, P. (2018). Panoptic Segmentation. URL <https://arxiv.org/abs/1801.00868v3>
- Kromp, F., Fischer, L., Bozsaky, E., Ambros, I., Doerr, W., Taschner-Mandl, S., Ambros, P., & Hanbury, A. (2019). Deep Learning architectures for generalized immunofluorescence based nuclear image segmentation. *arXiv:1907.12975 [cs, q-bio]*. URL <http://arxiv.org/abs/1907.12975>

- Kumar, H. (2019). Introduction to Panoptic Segmentation: A Tutorial.  
 URL <https://kharshit.github.io/blog/2019/10/18/introduction-to-panoptic-segmentation-tutorial>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.  
 URL <http://www.nature.com/articles/nature14539>
- Lee, J.-G., Jun, S., Cho, Y.-W., Lee, H., Kim, G. B., Seo, J. B., & Kim, N. (2017). Deep learning in medical imaging: general overview. *Korean Journal of Radiology*, 18(4), 570.  
 URL <https://synapse.koreamed.org/DOIx.php?id=10.3348/kjr.2017.18.4.570>
- Li, X., Meng, Y., Sun, X., Han, Q., Yuan, A., & Li, J. (2019). Is Word Segmentation Necessary for Deep Learning of Chinese Representations? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, (pp. 3242–3252). Florence, Italy: Association for Computational Linguistics.  
 URL <https://www.aclweb.org/anthology/P19-1314>
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017a). Focal Loss for Dense Object Detection.  
 URL <https://arxiv.org/abs/1708.02002v2>
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017b). Focal Loss for Dense Object Detection.  
 URL <https://arxiv.org/abs/1708.02002v2>
- Ling, H., & Jacobs, D. (2007). Shape Classification Using the Inner-Distance. *IEEE transactions on pattern analysis and machine intelligence*, 29, 286–99.
- Liu, D., Zhang, D., Song, Y., Huang, H., & Cai, W. (2020a). Cell R-CNN V3: A Novel Panoptic Paradigm for Instance Segmentation in Biomedical Images. *arXiv:2002.06345 [cs]*. ArXiv: 2002.06345.  
 URL <http://arxiv.org/abs/2002.06345>
- Liu, D., Zhang, D., Song, Y., Zhang, C., Zhang, F., O'Donnell, L., & Cai, W. (2019). Nuclei Segmentation via a Deep Panoptic Model with Semantic Feature Fusion. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, (pp. 861–868). Macao, China: International Joint Conferences on Artificial Intelligence Organization.  
 URL <https://www.ijcai.org/proceedings/2019/121>
- Liu, D., Zhang, D., Song, Y., Zhang, F., O'Donnell, L., Huang, H., Chen, M., & Cai, W. (2020b). Unsupervised Instance Segmentation in Microscopy Images via Panoptic Domain Adaptation and Task Re-Weighting. (pp. 4243–4252).  
 URL [https://openaccess.thecvf.com/content\\_CVPR\\_2020/html/Liu\\_Unsupervised\\_Instance\\_Segmentation\\_in\\_Microscopy\\_Images\\_via\\_Panoptic\\_Domain\\_Adaptation\\_CVPR\\_2020\\_paper.html](https://openaccess.thecvf.com/content_CVPR_2020/html/Liu_Unsupervised_Instance_Segmentation_in_Microscopy_Images_via_Panoptic_Domain_Adaptation_CVPR_2020_paper.html)
- MacLeod, J., & Gold, R. Z. (1951). The male factor in fertility and infertility. *Fertility and Sterility*, 2(5), 394–414.  
 URL <https://linkinghub.elsevier.com/retrieve/pii/S0015028216306616>
- McCaffrey, E., Donato, M., Keren, L., Chen, Z., Fitzpatrick, M., Jojic, V., Delmastro, A., Greenwald, N., Baranski, A., Graf, W., Bossé, M., Ramdial, P., Forgo, E., Valen, D., van de Rijn, M., Bendall, S., Banaei, N., Steyn, A., Khatri, P., & Angelo, M. (2020). *Multiplexed imaging of human tuberculosis granulomas uncovers immunoregulatory features conserved across tissue and blood*.
- MedicalNewsToday (2020). Sperm: Everything you need to know. Library Catalog: [www.medicalnewstoday.com](http://www.medicalnewstoday.com).  
 URL <https://www.medicalnewstoday.com/articles/sperm>

- Merino, M. (2019). Conceptos de inteligencia artificial: qué son las GANs o redes generativas antagónicas.  
 URL <https://www.xataka.com/inteligencia-artificial/conceptos-inteligencia-artificial-que-gans-redes-generativas-antagonicas>
- Moen, E., Bannon, D., Kudo, T., Graf, W., Covert, M., & Van Valen, D. (2019). Deep learning for cellular image analysis. *Nature Methods*, 16(12), 1233–1246.  
 URL <http://www.nature.com/articles/s41592-019-0403-1>
- Monroy, A. (2020). fertilization | Steps, Process, & Facts. Library Catalog: www.britannica.com.  
 URL <https://www.britannica.com/science/fertilization-reproduction>
- Movahed, R. A., Mohammadi, E., & Orooji, M. (2019). Automatic segmentation of Sperm's parts in microscopic images of human semen smears using concatenated learning approaches. *Computers in Biology and Medicine*, 109, 242–253.  
 URL <https://linkinghub.elsevier.com/retrieve/pii/S0010482519301404>
- Muyano, A. (2011). The immortal life of cells: HeLa vs HeLa S3. Library Catalog: Blogger.  
 URL <http://immortalcells.blogspot.com/2011/12/hela-vs-hela-s3.html>
- Netherton, J., Ogle, R. A., Hetherington, L., Silva Balbin Villaverde, A. I., Hondermarck, H., & Baker, M. A. (2020). Proteomic Analysis Reveals that Topoisomerase 2A is Associated with Defective Sperm Head Morphology. *Molecular & cellular proteomics: MCP*, 19(3), 444–455.
- Nissen, M. S., Krause, O., Almstrup, K., Kjærulff, S., Nielsen, T. T., & Nielsen, M. (2017). Convolutional neural networks for segmentation and object detection of human semen. In P. Sharma, & F. M. Bianchi (Eds.) *Image Analysis*, vol. 10269, (pp. 397–406). Cham: Springer International Publishing.  
 URL [http://link.springer.com/10.1007/978-3-319-59126-1\\_33](http://link.springer.com/10.1007/978-3-319-59126-1_33)
- Niwattanakul, S., Singthongchai, J., Naenudorn, E., & Wanapu, S. (2013). *Using of Jaccard Coefficient for Keywords Similarity*, vol. I. Hong Kong: IMCES 2013, proceedings of the international multiconference of engineers and computer scientists 2013 ed.
- Nooteboom, L. (2018). Design methodologies for Deep Learning - Humanising Autonomy - Medium.  
 URL <https://medium.com/@HumanisingAutonomy/design-methodologies-for-deep-learning-fdbc160deff7>
- NumPy Steering Council (2020). NumPy Reference — NumPy v1.19 Manual.  
 URL <https://numpy.org/doc/stable/reference/index.html>
- O'Reilly (2020). 1. Activity Definition (PMBOK® Guide 6.2) - The Project Management Tool Kit, 3rd Edition [Book]. Library Catalog: www.oreilly.com.  
 URL <https://www.oreilly.com/library/view/the-project-management/9780814433454/chap1.xhtml>
- PYMNTS (2020). Cyberexperts Warn Of The Dangers Of Deepfakes. Library Catalog: www.pymnts.com.  
 URL <https://www.pymnts.com/safety-and-security/2020/experts-raise-concerns-over-deepfakes-at-house-hearing/>
- Rahbari, R., Sheahan, T., Modes, V., Collier, P., Macfarlane, C., & Badge, R. M. (2009). A novel L1 retrotransposon marker for HeLa cell line identification. *BioTechniques*, 46(4), 277–284.  
 URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2696096/>
- Revollo Sarmiento, N. V., Thomsen, F. S. L., Delrieux, C., & González-José, R. (2020). Supervised learning for semantic segmentation of human spermatozoa. In J. Brieva, E. Romero, & N. Lopore (Eds.) *15th International Symposium on Medical Information Processing and*

- Analysis*, (p. 21). Medelin, Colombia: SPIE.  
 URL <https://www.spiedigitallibrary.org/conference-proceedings-of-spie/11330/2542464/Supervised-learning-for-semantic-segmentation-of-human-spermatozoa/10.1117/12.2542464.full>
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *International Conference on Medical image computing and computer-assisted intervention*, (pp. 234–241).
- Rybkin, O. (2018). The reasonable ineffectiveness of pixel metrics for future prediction (and what to do about it).  
 URL [https://medium.com/@olegrybkin\\_20684/the-reasonable-ineffectiveness-of-mse-pixel-loss-for-future-prediction-and-what-to-do-about-it-4dca8152355d](https://medium.com/@olegrybkin_20684/the-reasonable-ineffectiveness-of-mse-pixel-loss-for-future-prediction-and-what-to-do-about-it-4dca8152355d)
- Salech, F., Mery, V., Larrondo, F., & Rada, G. (2008). Estudios que evalúan un test diagnóstico: interpretando sus resultados. *Revista médica de Chile*, 136(9).  
 URL [http://www.scielo.cl/scielo.php?script=sci\\_arttext&pid=S0034-98872008000900018&lng=en&nrm=iso&tlang=en](http://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0034-98872008000900018&lng=en&nrm=iso&tlang=en)
- Sarkar, D. D. (2018). A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning. Library Catalog: towardsdatascience.com.  
 URL <https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a>
- Selman, J. M. (2004). Aplicaciones clínicas del procesamiento digital de imágenes médicas. *Revista Clinica Las Condes*, Vol.15(N° 2), 16.  
 URL [http://www.clc.cl/clcprod/media/contenidos/pdf/med\\_15\\_3/aplicacionesclinicas.pdf](http://www.clc.cl/clcprod/media/contenidos/pdf/med_15_3/aplicacionesclinicas.pdf)
- Shaker, F., Monadjemi, S. A., & Naghsh-Nilchi, A. R. (2016). Automatic detection and segmentation of sperm head, acrosome and nucleus in microscopic images of human semen smears. *Computer Methods and Programs in Biomedicine*, 132, 11–20.  
 URL <https://linkinghub.elsevier.com/retrieve/pii/S0169260716304023>
- Shorten, C., & Khoshgoftaar, T. M. (2019a). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 60.  
 URL <https://doi.org/10.1186/s40537-019-0197-0>
- Shorten, C., & Khoshgoftaar, T. M. (2019b). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 60.  
 URL <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0>
- Shorten, C., & Khoshgoftaar, T. M. (2019c). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 60.  
 URL <https://doi.org/10.1186/s40537-019-0197-0>
- Simard, P., Steinkraus, D., & Platt, J. (2003). Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis. (pp. 958–962).
- SSH Academy (2020). SSH port forwarding - Example, command, server config.  
 URL <https://www.ssh.com/ssh/tunneling/example>
- Steckelberg, J. (2019). Granuloma: ¿qué significa? Library Catalog: www.mayoclinic.org.  
 URL <https://www.mayoclinic.org/es-es/granuloma/expert-answers/faq-20057838>
- Sultana, F., Sufian, A., & Dutta, P. (2020). Evolution of Image Segmentation using Deep Convolutional Neural Network: A Survey. *Knowledge-Based Systems*, 201-202, 106062.  
 URL <http://www.sciencedirect.com/science/article/pii/S0950705120303464>

- Sun, J., Tárnok, A., & Su, X. (2020). Deep Learning-Based Single-Cell Optical Image Studies. *Cytometry Part A*, 97(3), 226–240. Eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/cyto.a.23973>  
URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/cyto.a.23973>
- Taulli, T. (2019). *Artificial Intelligence Basics: A Non-Technical Introduction*. Berkeley, CA: Apress.  
URL <http://link.springer.com/10.1007/978-1-4842-5028-0>
- WHO (2020). OMS | Tratamiento de la tuberculosis: situación actual y perspectivas. Library Catalog: www.who.int Publisher: World Health Organization.  
URL <https://www.who.int/bulletin/volumes/83/11/onyebujoh1105abstract/es/>
- Zacà, C., Coticchio, G., Tarozzi, N., Nadalini, M., Lagalla, C., Garolla, A., & Borini, A. (2020). Sperm count affects cumulative birth rate of assisted reproduction cycles in relation to ovarian response. *Journal of Assisted Reproduction and Genetics*.  
URL <https://doi.org/10.1007/s10815-020-01807-5>
- Zhang, D., Song, Y., Liu, D., Jia, H., Liu, S., Xia, Y., Huang, H., & Cai, W. (2018). Panoptic Segmentation with an End-to-End Cell R-CNN for Pathology Image Analysis. In A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, & G. Fichtinger (Eds.) *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, Lecture Notes in Computer Science, (pp. 237–244). Cham: Springer International Publishing.

## APÉNDICE A. TABLAS

Tabla A.1: Promedios y mediana del método Movahed desde cero, en distintas instancias.

**Fuente:** Elaboración propia, 2020

	Dice	Jaccard	Hausdorff
Promedio: Solo detecciones	0,950336066	0,90226327	0,736492077
Promedio: Mejores 50	0,9776416	0,9557608	0,955589283
Promedio: Peores 50	0,9010476	0,8030468	0,52854144
Mediana:	0,96150	0,92480	0,74533

Tabla A.2: Tabla comparativa de resultados de pruebas entre bibliotecas que aportan al aumento de datos. La prueba muestra cuantas imágenes por segundo pueden ser procesados en un solo core, con un dataset de 2000 imágenes provenientes de ImageNet con una CPU Intel Xeon Gold 6140. Puntaje mas alto es mejor.

**Fuente:** (Buslaev et al., 2020)

	<b>albumentations 0.5.0</b>	<b>imgaug 0.4.0</b>	<b>torchvision (Pillow-SIMD backend) 0.7.0</b>	<b>keras 2.4.3</b>	<b>augmentor 0.2.8</b>	<b>solt 0.1.9</b>
HorizontalFlip	<b>9909</b>	2821	2267	873	2301	6223
VerticalFlip	<b>4374</b>	2218	1952	4339	1968	3562
Rotate	<b>371</b>	296	163	27	60	345
ShiftScaleRotate	<b>635</b>	437	147	28	-	-
Brightness	<b>2751</b>	1178	419	229	418	2300
Contrast	<b>2756</b>	1213	352	-	348	2305
BrightnessContrast	<b>2738</b>	699	195	-	193	1179
ShiftRGB	<b>2757</b>	1176	-	348	-	-
ShiftHSV	<b>597</b>	284	58	-	-	137
Gamma	<b>2844</b>	-	382	-	-	946
Grayscale	<b>5159</b>	428	709	-	1064	1273
RandomCrop64	<b>175886</b>	3018	52103	-	41774	20732
PadToSize512	<b>3418</b>	-	574	-	-	2874
Resize512	1003	634	<b>1036</b>	-	1016	977
RandomSizedCrop_64_512	<b>3191</b>	939	1594	-	1529	2563
Posterize	<b>2778</b>	-	-	-	-	-
Solarize	<b>2762</b>	-	-	-	-	-
Equalize	644	413	-	-	<b>735</b>	-
Multiply	<b>2727</b>	1248	-	-	-	-
MultiplyElementwise	118	<b>209</b>	-	-	-	-
ColorJitter	<b>368</b>	78	57	-	-	-