



Instituto Tecnológico
de Buenos Aires

TESIS DE GRADO
DEPARTAMENTO DE BIOINGENIERÍA

**Desarrollo de un Dispositivo Contador de Células
Mediante la Aplicación de Redes Neuronales YOLO
Entrenadas con Imágenes Sintéticas**

Autores:

Proietti Anastasi Alejandro Marcelo Legajo: 56256 Correo: aproietti@itba.edu.ar

Volman Uriel Brian Legajo: 55095 Correo: uvolman@itba.edu.ar

Tutores:

Ing. Selmo Carlos

Lic. Esposito Giuliana Antonella (Co-Tutora)

Índice

References	1
Glosario	3
Abreviaciones	6
1. Introducción	8
1.1. Cultivo Celular	8
1.1.1. Aplicaciones y ventajas	8
1.1.2. Principios del Cultivo Celular	10
1.1.3. Cuantificación y Recuento Celular	13
1.2. Redes Neuronales Artificiales	17
1.2.1. Red Neuronal: <i>You Only Look Once</i> (YOLO)	20
1.2.2. Ventajas de YOLO	28
1.2.3. YOLO vs otros sistemas de detección	28
1.3. Estado del Arte	31
1.3.1. Desarrollos <i>open-source</i>	38
1.4. Objetivos	40
1.4.1. Objetivos de mínima	40
1.4.2. Objetivos de máxima	40
1.5. Motivación	41
2. Materiales y Métodos	41
2.1. Hardware	42
2.1.1. Procesamiento de datos	43
2.1.2. Adquisición de Imágenes	44
2.1.3. Diseño mecánico y construcción del dispositivo	44
2.2. Dataset	49
2.2.1. Dataset sintético	50
2.2.2. Dataset Real	55

2.3. Modelo	55
2.3.1. Darknet	56
2.3.2. Entrenamiento	56
2.4. Métricas de evaluación	59
2.5. Software	63
3. Resultados	64
3.1. Características técnicas del prototipo diseñado	64
3.2. Dataset	70
3.3. Modelos Entrenados	70
3.4. Software	73
4. Discusión	82
4.1. Características del Dispositivo y Comparación con contadores del mercado . . .	82
4.2. Construcción del <i>Dataset</i>	86
4.3. Análisis, Comparación y Selección del Modelo de Detección	87
5. Conclusión	105
6. Futuras Mejoras y Aplicaciones	106
6.1. Modelo de detección	106
6.2. Dispositivo	110
7. Agradecimientos	112

Glosario

Anchor Box caja de altura y ancho predefinido que permite capturar el tamaño y aspecto de los objetos que se quieren detectar, y a partir de las cuales los modelos realizan predicciones.

Apertura Numérica número adimensional que caracteriza el rango de ángulos para los cuales un sistema óptico acepta luz.

Azul de tripán colorante utilizado en tinciones histológicas para ensayos de viabilidad mediante el método de tinción por exclusión.

Bounding Box caja contenedora asociada a las coordenadas x e y de un objeto de la imagen. Un *bounding box* puede ser anotado manualmente por un experto, en cuyo caso funciona como verdad de tierra o puede ser predicha por la red neuronal.

Campo de Visión área de detección de la cámara sobre la muestra analizada.

Coefficiente de Variación forma de medir la variabilidad entre mediciones definida como la relación entre el desvío estandar y la media de la población ($CV = \frac{\sigma}{\mu}$).

Contador Celular Automático instrumento de laboratorio utilizado para detectar, contar y analizar células en suspensión.

Conteo Celular procedimiento mediante el cual se puede determinar la concentración de células en suspensión de una muestra.

Cultivo Celular remoción de células de un organismo vivo y posterior crecimiento en ambientes artificiales controlados.

Cultivo Primario conjunto de células aisladas de un tejido que proliferan en las condiciones apropiadas.

Darknet entorno desarrollado por [Redmon et al.](#) para el entrenamiento y ejecución de redes neuronales YOLO.

Data Augmentation conjunto de técnicas utilizadas para aumentar la cantidad de datos utilizados en un entrenamiento a partir de la introducción de copias modificadas de los datos originales.

Exhaustividad (*Recall*) métrica que evalúa la proporción de predicciones correctas (verdaderos positivos) sobre el total de objetos presentes en una imagen. Se calcula como $R = \frac{TP}{TP+FN}$.

Falso Negativo resultado en el que el modelo no realiza una predicción para un objeto que esta presente en la imagen.

Falso Positivo resultado en el que el modelo predice incorrectamente la presencia de un objeto.

Hemocitómetro portaobjetos transparente con una hendidura rectangular marcada con diferentes grillas o cuadrículas de dimensiones conocidas utilizado para el conteo de células en suspensión..

Intersección sobre Unión (*Intersection over Union*) métrica que evalúa el área compartida entre una caja contenedora predicha y una verdadera. Se calcula como $IoU = \frac{\text{area de interseccion}}{\text{area de union}}$.

Linea Celular población definida de células que se pueden mantener en cultivo durante un período prolongado de tiempo, conservando la estabilidad de ciertos fenotipos y funciones.

Precisión métrica que evalúa la proporción de predicciones correctas (verdaderos positivos) sobre el total de predicciones realizadas. Se calcula como $P = \frac{TP}{TP+FP}$.

Precisión Media (*Average Precision*) promedio de las precisiones del modelo en diferentes niveles de exhaustividad. Puede calcularse como el área bajo la curva de precisión-exhaustividad o como $AP = \frac{1}{11} \sum_{r \in \{0,0,\dots,1\}} P_r$.

Precisión Media Promedio (*Mean Average Precision*) promedio de las precisiones medias de cada clase.

Red Neuronal Artificial modelo computacional basado en la estructura neuronal del cerebro y con capacidad de aprender a resolver problemas complejos.

Resolución Óptica capacidad de un instrumento para separar o distinguir dos objetos de una misma imagen.

Subcultivo proceso mediante el cual se traspasan las células de un recipiente a otro proporcionándoles una nueva superficie de proliferación e incorporación de nutrientes.

Valor F1 (*F1 Score*) media armónica entre la precisión y exhaustividad del modelo. Se calcula como $F_1 = 2 \times \frac{P \times R}{P + R}$.

Verdadero Positivo resultado en el que el modelo predice correctamente la presencia de un objeto. Suele catalogarse como verdaderos positivos a todas las predicciones en las cuales la clasificación del objeto es correcta y el IoU de la caja predicha respecto de la verdadera es mayor al 50 %.

Viabilidad proporción de células vivas sobre el total de células presentes al momento de realizar un conteo celular.

YOLO (*You Only Look Once*) red neuronal para detección de objetos consistiendo en una sola red convolucional que simultáneamente predice múltiples cajas contenedoras y niveles de confianza para dichas cajas..

Abreviaciones

ANN Red Neuronal Artificial.

AP Precisión Media (*Average Precision*).

CV Coeficiente de Variación.

F1 Valor F1 (*F1 Score*).

FN Falso Negativo.

FOV Campo de Visión.

FP Falso Positivo.

IoU Intersección sobre Unión (*Intersection over Union*).

mAP Precisión Media Promedio (*Mean Average Precision*).

NA Apertura Numérica.

OR Resolución Óptica.

P Precisión.

R Exhaustividad (*Recall*).

TB Azul de tripán.

TP Verdadero Positivo.

Resumen

Al realizar un conteo celular existen 2 inconvenientes (I) el proceso ocupa mucho tiempo al realizar diferentes conteos y (II) es necesaria cierta experiencia para realizar el conteo correctamente. En el siguiente proyecto se propuso desarrollar un dispositivo de conteo celular automático que sea capaz de, mediante el procesamiento de imágenes y la implementación de redes neuronales, reconocer células, contarlas y realizar cálculos para dar un diagnóstico del cultivo celular. El proyecto se centró en el desarrollo de un sistema que permite obtener imágenes a nivel microscópico de una muestra de un cultivo celular (con una tinción de Azul de tripán, en una cámara de Neubauer) y en el desarrollo de una red neuronal capaz de reconocer células y determinar el número de células, vivas y muertas, presentes. A partir de esos datos, el dispositivo puede calcular el stock celular total y su viabilidad para realizar el sembrado de la siguiente generación de la línea en la que se trabaja. Se desarrollaron 12 *datasets* a partir de los cuales, se entrenaron la misma cantidad de redes neuronales y se compararon sus desempeños con el motivo de escoger un modelo de detección idóneo para su implementación en el dispositivo diseñado. Como resultado, no solo se desarrolló una red neuronal que consiguió alcanzar un mAP superior al 95 %, sino que también, se desarrolló una interfaz gráfica que, integrado al sistema de adquisición de imágenes, permite su uso de manera sencilla. El *dataset* de imágenes utilizado para el entrenamiento, y el modelo de conteo de células desarrollado se encuentran publicados en [2].

1. Introducción

1.1. Cultivo Celular

El cultivo celular refiere a la remoción de células de un organismo vivo y su posterior crecimiento en ambientes artificiales controlados [3] y fue tomado como método de estudio de células a comienzos del siglo XX, a partir de muestras tomadas directamente de tejidos de animales. Hoy en día, la evolución del conocimiento y el desarrollo de técnicas de cultivo celular permite, mantener con vida una gran diversidad de líneas celulares en laboratorios y su eventual estudio.

1.1.1. Aplicaciones y ventajas

El uso de cultivos celulares tiene diversas aplicaciones en la biología, medicina y bioingeniería (Figura 1.1). Históricamente, su aplicación más difundida fue en el desarrollo y producción de vacunas antivirales, pero a esto se le han ido agregando otras como:

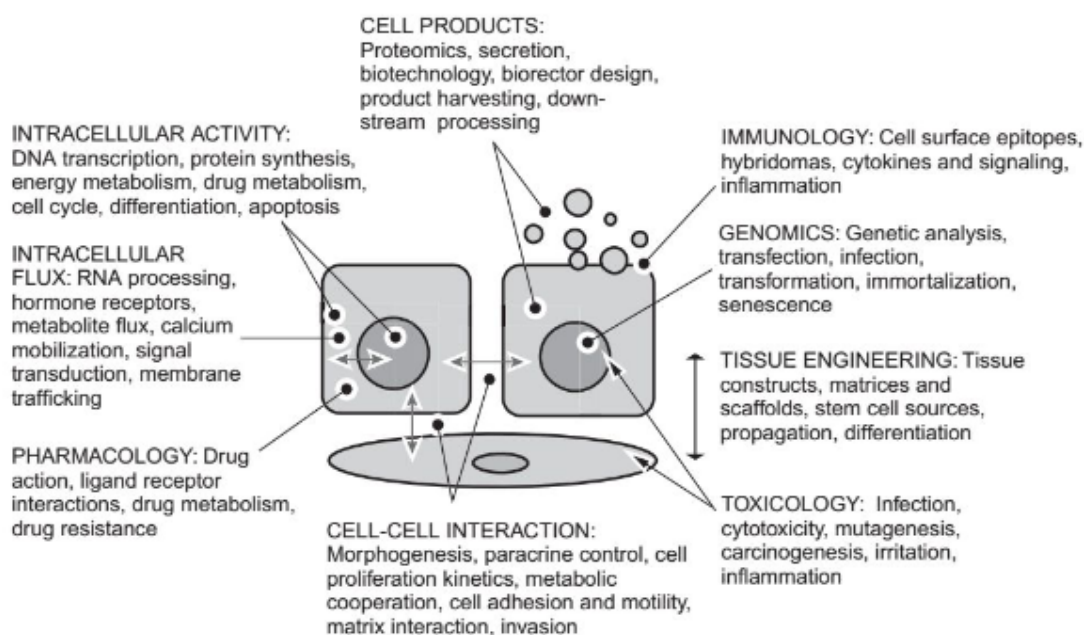


Fig. 1.1: Aplicaciones para cultivos celulares. Figura obtenida de: Freshney, R. I. (2015). *Culture of animal cells: a manual of basic technique and specialized applications*. John Wiley & Sons

- Creación de modelos para el estudio de la fisiología normal de la célula, las interacciones

de células con agentes patógenos (virus, bacteria), los efectos de diferentes drogas y compuestos tóxicos, su actividad intracelular e interacción con otras células y la carcinogénesis y mutagénesis.

- Cultivos de células cancerígenas como método para el estudio de carcinogénesis y del efecto de diferentes drogas y terapias.
- Estudio de la toxicidad de diferentes drogas, así como las dosis máximas permitidas.
- Producción de proteínas (como hormonas y anticuerpos) y otros compuestos biológicos en grandes escalas.
- Aplicaciones en ingeniería genética y terapia génica a partir de la introducción de nuevo material genético, la eliminación o la modificación de material genético viejo en células de un cultivo para estudiar los efectos que pueden traer a las mismas.
- Aplicaciones en ingeniería de tejidos donde el cultivo de células (en general células madre) *in vitro* sirve para el desarrollo de sustitutos biológicos que reparan tejidos, mejoran la función o reemplazan a todo un órgano.

En cuanto a las ventajas que presentan estas técnicas, el cultivo celular permite el control del ambiente fisicoquímico de las células, pudiendo regular de manera precisa ciertas variables del entorno celular como el pH, la temperatura, la presión osmótica, la presión de O₂ y CO₂ así como la composición del medio fisiológico, lo que permite establecer las condiciones de estudio de las células, y entender los factores que llevan a su proliferación, entre otras cosas.

Además, a diferencia de las muestras tomadas de tejidos, los cultivos celulares son altamente homogéneos en cuanto al tipo celular. Se ha visto que, en un cultivo, tiende a proliferar un solo tipo celular y que, luego de unos pocos pasajes, el cultivo adopta una constitución uniforme en la mayoría de los casos. Esto significa que los estudios realizados en cultivos de células carecen de errores debidos a variaciones entre muestras, y garantiza la reproducibilidad de cualquier experimento realizado en cultivos con características idénticas a los anteriores.

Finalmente, el cultivo celular permite la realización de diferentes experimentos con volúmenes menores y costos bajos. Con este tipo de técnicas, los cultivos pueden ser expuestos a diferentes

reactivos a bajas concentraciones, pero que tienen un acceso directo a la célula, lo que evita el uso de grandes cantidades de reactivos que, en estudios *in vivo* se pierden por la distribución y excreción de los tejidos [4, 5].

1.1.2. Principios del Cultivo Celular

A la hora de comenzar un cultivo celular, hay tres opciones: (I) cultivo de un órgano, donde se conservan las características propias del tejido original, y el crecimiento se da en una interfaz líquido-gas usando un soporte o sustrato capaz de conservar la estructura tridimensional propia del tejido, (II) cultivo de un tejido o explante, donde un fragmento de un tejido se posiciona en una superficie donde se espera que las células se adhieran y migren y finalmente, (III) cultivo celular, donde las células obtenidas de la muestra original son dispersadas en una suspensión y serán luego cultivadas ya sea adheridas a una superficie o suspendidas en su propio medio de cultivo. La principal diferencia entre estas formas de cultivo se encuentra en el hecho de que, mientras el cultivo de órganos y tejidos (cultivos de explantes) mantienen las características histológicas del tejido *in vivo*, el cultivo celular no. En los dos primeros casos, las células dentro del cultivo retienen su condición de células diferenciadas y tienen una baja capacidad de proliferación. En cultivos celulares, en cambio, existe un solo tipo celular. Esto hace que la proliferación sea mucho mayor y que el cultivo sea homogéneo [4].

Todo cultivo celular se inicia a partir de la obtención de una muestra del tejido (mediante biopsias, escisiones, aspiraciones, punciones, trepanaciones o raspados) y la disociación de las células individuales (con métodos mecánicos o enzimáticos) [4, 6]. En todo caso, las células que se cultivan luego de este paso corresponden a un *cultivo primario*, que puede ser un cultivo de células en suspensión en el cual las células crecen independientemente de un anclaje y dispersas en el medio de cultivo (células hemapoyéticas o células derivadas de tumores), o un cultivo en *monocapa* en el cual células que dependen de un anclaje crecen adheridas a un sustrato artificial (como frascos, platos, botellas, matrices, geles, etc.) [3].

Brevemente, la evolución de cualquier cultivo comienza con el sembrado del cultivo primario en un sustrato físico junto a un medio fisiológico necesario para su desarrollo. El medio de crecimiento es el encargado de proveer a las células los nutrientes (carbohidratos, aminoácidos y proteínas, lípidos y ácidos grasos, vitaminas, etc), factores de crecimiento y hormonas necesarias

así como regular el pH, la presión osmótica y la concentración de CO₂ y O₂ del cultivo [3, 7]. Las células sembradas, entonces, proliferan y comienzan a ocupar un mayor espacio del sustrato o medio llegando al estado de *confluencia* cuando todo el espacio del sustrato se encuentra ocupado y las células ven su capacidad de crecimiento o proliferación inhibida por el contacto con otras células [4]. En este momento, es necesario realizar un procedimiento de *subcultivo o pasaje*, cambiando de sustrato y medio de cultivo para conservar una baja densidad celular y conservar el fenotipo y la capacidad de proliferación de las células. Este primer pasaje, da a inicio a lo que se denomina una *línea celular* (implicando que los cultivos de una línea determinada consisten en numerosas generaciones de un tipo celular particular presente originalmente en el cultivo primario o tejido) que deberá ser sub-cultivada más de una vez (cada vez que cada subcultivo llega a la confluencia). Los sucesivos pasajes llevan a que las células con una tasa de crecimiento mayor predominen, y a que cada generación de la línea consista en una población genotípica y fenotípica más uniforme que la anterior¹. Una línea celular normal es capaz de sobrevivir un número determinado de pasajes (que puede variar entre 20 y 100) después de que las células están genéticamente programadas para morir en un proceso llamado *senescencia*².

Mantenimiento del Cultivo

Cada generación o subcultivo sigue su propia curva de crecimiento necesitando de un operador para su mantenimiento con el objetivo básico de procurar que las células de cada generación crezcan sanas y en las condiciones deseadas y consta, generalmente, de cambios en el medio o sustrato en el cual crecen las células. En la vida de una línea celular este proceso se repite con cada generación (Figura 1.2).

Un cambio de medio de cultivo tiene como propósito restablecer los nutrientes y factores

¹En otras palabras, cuando se obtiene el cultivo primario de la muestra original las células suelen retener las características propias del tejido *in vivo*, y en el total de la población suelen existir diferentes tipos celulares. Sin embargo, con el tiempo aquellas células capaces de una mayor proliferación en ese ambiente tienden a ocupar una mayor proporción de la población, por lo que el cultivo se vuelve más homogéneo con cada subcultivo.

²Una línea celular conformada por este tipo de células se denomina *línea celular finita*. Sin embargo, existe una cierta porción de líneas que posee la capacidad de proliferar indefinidamente. Las *líneas celulares infinitas o continuas* son líneas celulares inmortales que no pasan por el periodo de senescencia. En cambio, las células pasan por un proceso de cambio genotípico llamado *transformación* de manera espontánea (exposición a radiaciones ionizantes o a carcinógenos químicos) o inducida (infección vírica o transfección de ADN). En cualquier caso, las células transformadas se caracterizan por ser inmortales (crecen indefinidamente), presentar un crecimiento aberrante (se pierde la inhibición por contacto, la limitación de la densidad celular durante la proliferación y no dependen del anclaje), ser malignas (invaden tejidos y dan lugar al crecimiento de tumores) y genéticamente inestables.

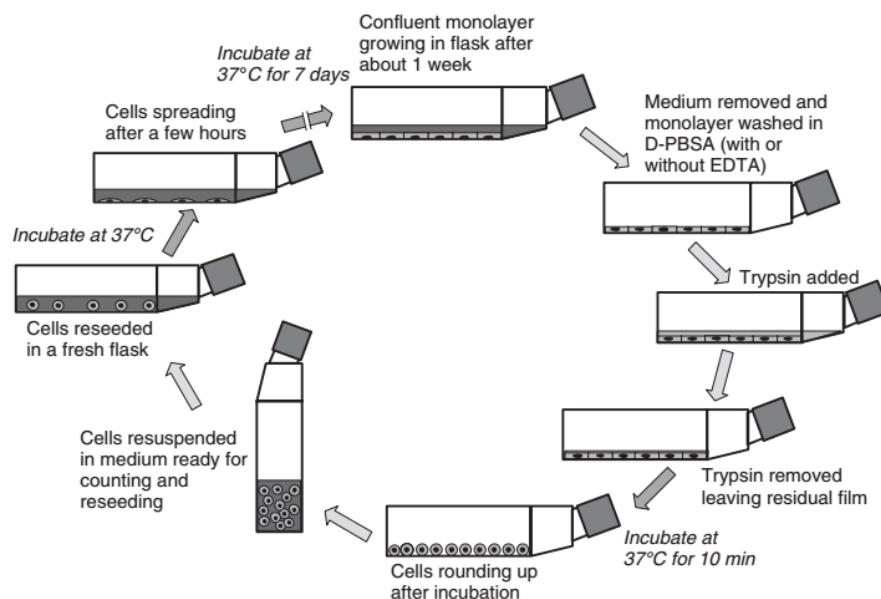


Fig. 1.2: Proceso típico de crecimiento de un cultivo en monocapa. Las células crecen adheridas a una superficie (generalmente plástica) y deben ser removidas con un tratamiento enzimático cada vez que llegan a la confluencia. Luego, serán transferidas a otro sustrato con medio de cultivo fresco, e incubadas nuevamente para repetir el proceso. Figura obtenida de: Freshney, R. I. (2015). *Culture of animal cells: a manual of basic technique and specialized applications*. John Wiley & Sons

fundamentales para el crecimiento celular y se realiza, típicamente, cuando se da una caída en el pH producto de la actividad celular (que pone en riesgo la viabilidad celular), un aumento en la concentración celular (respecto del volumen de medio) o por cambios en la morfología de la célula que indiquen un posible deterioro.

Por otra parte, dado que la mayoría de las células normales son incapaces de crecer a grandes densidades celulares, los cultivos requieren de cambios periódicos de sustrato para mantener la capacidad de proliferación celular. Estos se realizan no bien se llega a la *confluencia*, cuando se agotan los nutrientes del medio, cuando existe un rápido descenso del pH, en un momento arbitrario elegido para garantizar la reproducibilidad de los experimentos o bien cuando es necesario utilizar el cultivo para otros procedimientos [4]. Este proceso se realiza usando proteasas o métodos mecánicos para desprender las células del sustrato (cuando las células crecen en monocapa), separando las células del medio viejo (con un centrifugado) y sembrando las células en un sustrato nuevo con medio fresco. Al momento de sembrar las células, es posible separar la población entera en más de un frasco o utilizando frascos o botellas más grandes con

un mayor volumen de medio, para mantener el cultivo con una densidad celular que favorezca la proliferación.

1.1.3. Cuantificación y Recuento Celular

La citometría, consiste en la disciplina encargada de analizar las características de las células, midiendo diferentes variables como pueden ser tamaño, morfología, contenido de ADN y proteínas, o número de células. El conteo celular es una parte de la citometría realizada periódicamente dentro del proceso de cultivo celular como forma de caracterizar las propiedades de crecimiento de las diferentes líneas celulares, por motivos de análisis experimentales, o como método de diagnóstico y tratamiento. Entre sus usos se pueden destacar:

- En medicina, la concentración de diferentes tipos celulares (glóbulos rojos o blancos) pueden dar información acerca del estado de salud del paciente.
- En terapias celulares, para controlar las dosis de células administradas al paciente.
- En patología para conocer la presencia y concentración de bacterias, virus y otros patógenos en muestras de tejidos.
- En biología molecular, para conocer la concentración de reactivos o químicos necesarios para los diferentes experimentos.
- En estudios *in vitro* que precisen conocer las tasas de proliferación de las células.
- En estudios *in vitro* para conocer la viabilidad de las células (proporción de células vivas y muertas del total de la población). [8].

Conteo Manual

Históricamente, la forma más común para realizar conteos es manualmente usando una *cámara de conteo* o *hemocitómetro*. El mismo, consta de un portaobjetos transparente con una hendidura rectangular de una profundidad específica marcada con diferentes grillas o cuadrículas de dimensiones conocidas que delimitan la superficie de conteo. El objetivo de un hemocitómetro es poder colocar un volumen de una suspensión celular o alícuota del cultivo dentro de la hendidura y visualizar, bajo un microscopio, las células contenidas dentro de la cuadrícula o retícula,

pudiendo contar las células en un volumen conocido y calcular, a partir de eso, la concentración de células por volumen y la densidad por superficie.

Si bien existen diferentes tipos de hemocitómetros, la más usada es la “cámara de Neubauer” o “cámara de Neubauer mejorada”. Estas tienen, en su área de conteo, 9 cuadrados de 1 mm^2 cada uno, de los cuales 4 están subdivididos en 16 cuadrados de 0.0625 mm^2 y uno central subdividido en 400 cuadrados de un área de 0.0025 mm^2 cada uno (Figura 1.3). Dado que se conoce el área y el volumen (ya que se sabe la profundidad de la hendidura) de conteo, es posible, una vez que se cuentan las células presentes en cada cuadrado, obtener un estimado de la cantidad de células por superficie de sustrato o por volumen de medio (densidad celular) [9].

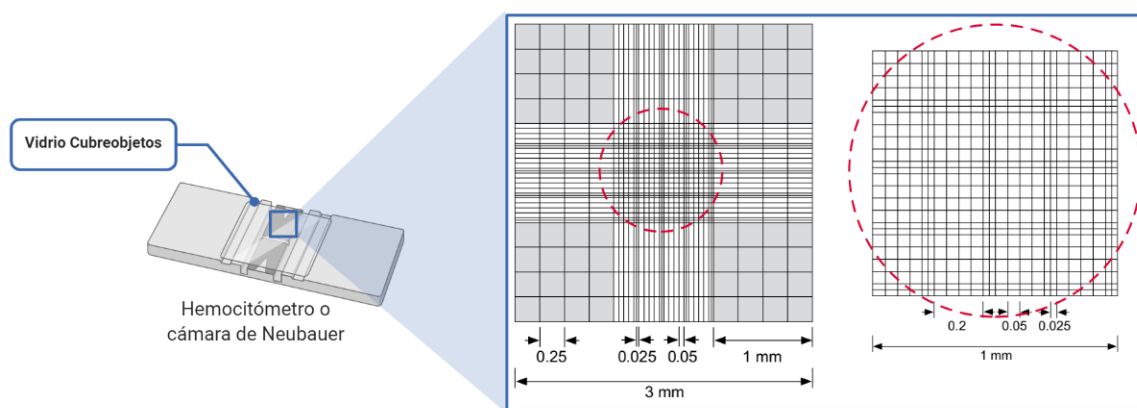


Fig. 1.3: Hemocitómetro (Cámara de Neubauer)

Si bien un conteo del total de las células es útil para un investigador, en general, cuando se realizan cultivos celulares, se requiere conocer la viabilidad de la población de células de un subcultivo. Para esto, es común utilizar diferentes tipos de colorantes que permitan distinguir y contar células vivas y muertas y que posibiliten el cálculo de la proporción sobre el total de células [10].

Tradicionalmente, el colorante mas usado es el *azul de tripán*. El *método de tinción por exclusión* con azul de tripán consiste en realizar un conteo con un hemocitómetro de una alícuota del cultivo celular al cual se le agrega previamente dicho colorante. En células muertas, el azul de tripán puede ingresar (ya que poseen una membrana celular dañada incapaz de controlar el tránsito de macromoléculas hacia la célula) y unirse a algunas proteínas intracelulares. Por otro lado, las células vivas o viables con una membrana intacta permanecerán sin ser teñidas. Esto hace que, bajo un microscopio invertido, las células muertas se vean de color azul opaco

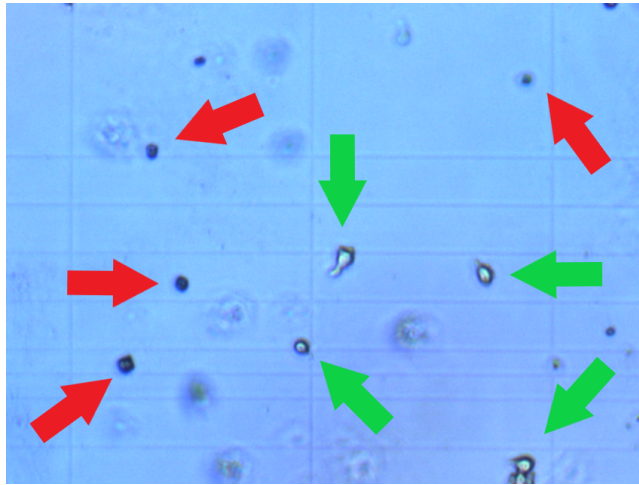


Fig. 1.4: Diferencias entre células vivas y muertas. Bajo un microscopio óptico de campo claro, las células vivas aparecen claras y brillantes (flechas verdes), mientras que las muertas se tiñen con un tono azul oscuro (flechas rojas).

y las vivas transparentes o brillantes (Figura 1.4) [11]. Al realizar, entonces, un conteo usando un hemocitómetro, y diferenciando, en dicho conteo, las células vivas de las muertas es posible obtener una medida de la viabilidad de la población celular [12].

Teóricamente, al realizar un conteo celular, la densidad celular se obtiene realizando:

$$celulas/ml = \frac{Celulas\ contadas \times Factor\ de\ dilucion}{Volumen\ de\ Conteo} \quad (1)$$

donde el factor de dilución depende de la proporción de la mezcla de azul de tripán y suspensión celular, y el volumen de conteo depende de la cantidad de cuadrantes observados. En general, la profundidad de un hemocitómetro es de 0.1 mm, por lo que el volumen en un cuadrante es de 0.1 mm^3 o $1 \times 10^{-4}\text{ ml}$. Si la dilución es 1:1 ($Factor\ de\ Dilucion = 2$):

$$celulas/ml = \frac{Celulas\ contadas \times 2 \times 10000}{n} \quad (2)$$

donde n es el numero de cuadrantes utilizados en el conteo.

Conteo Automático

Si bien el conteo manual con un hemocitómetro y un microscopio óptico es el método mas usado para conocer la viabilidad celular en un cultivo gracias a su adaptabilidad y bajo costo, el mismo presenta algunas desventajas, por lo que podría no ser la mejor opción, o al menos la

mas confiable, a la hora de realizar conteos celulares [13, 14].

Entre los problemas que puede presentar el conteo manual, se identifican:

- El tiempo y esfuerzo que implican al investigador encargado de realizar el conteo lo que impide el procesamiento de varias muestras (de diversos cultivos) en un período corto de tiempo;
- La alta dependencia de la precisión y confiabilidad del resultado del grado de la experiencia o habilidad del operador;
- La subjetividad del resultado respecto del operador;
- Largos periodos de incubación de las células con azul de tripán pueden llevar a la muerte de algunas células inicialmente viables debido a la toxicidad del colorante. Esto limita el tiempo que el operador tiene para llevar a cabo un conteo representativo de la realidad.
- Largos periodos de incubación de las células con azul de tripán también pueden llevar a la tinción de algunas células viables, produciendo, una vez más, resultados erróneos [13-16].

Por algunas de estas razones, hacia finales de la década de 1950 se introdujo el primer contador de células automático. El mismo, desarrollado por *Wallace Coulter*, permitía realizar un conteo de células que se movían individualmente a través de un canal y a las cuales se les aplicaba un campo eléctrico. Hasta el día de hoy, y en rasgos generales, se pueden identificar tres tipos diferentes de contadores de células.

El primero, corresponde a un contador basado en resistencia. Este, tal como lo pensó Coulter, mide los cambios de impedancia que se dan en una solución conductora (o los cambios en el flujo de corriente) gracias al paso de células a través de una apertura en la cual se aplica un campo eléctrico. En este contador, cada célula, que se mueve individualmente a través de un canal, provoca un aumento en la resistencia que es registrado. Así, la cantidad de eventos registrados (cambios en la resistencia) corresponde a la cantidad de células contadas. Además, como la variación de resistencia es proporcional al tamaño (volumen) de la célula que pasa por la apertura, además del numero de células es posible conocer el tamaño medio. En este tipo de contadores también se puede distinguir entre células viables y no viables (viabilidad del cultivo) dado que las células no viables presentan una membrana plasmática desmembrada que ofrece

una menor resistencia al paso de corriente eléctrica. La gran ventaja de este tipo de contadores es su velocidad y su facilidad para procesar grandes volúmenes de células en suspensión, lo que evita que existan errores en el conteo debido al uso de pequeños volúmenes como muestra del cultivo entero.

El segundo tipo de contadores corresponde a los citómetros de flujo. Estos funcionan, al igual que los basados en resistencia, con células que se mueven individualmente a través de un canal. En los citómetros de flujo se incide la célula con un haz láser que es dispersado de acuerdo a las características de la misma. En citometría de flujo es usual marcar con fluorescencia las células mediante anticuerpos de modo que se pueda diferenciar que tipo de célula pasa por el canal dependiendo de la luz absorbida y emitida. Si bien los citómetros de flujo suelen tener varias aplicaciones (conteo celular, detección de marcadores, microorganismos y proteínas, separación de células, etc) representa una opción costosa para un laboratorio si solo se necesita para realizar conteos.

Finalmente, el tercer grupo de contadores corresponde a aquellos basados en análisis de imágenes. En ese tipo de contadores rige el mismo concepto que en el conteo manual. Suelen obtener imágenes microscópicas digitales y usar diferentes métodos de procesamiento de imágenes y algoritmos de clasificación que permitan detectar las células y contarlas. Además, suelen presentar la posibilidad de conocer la viabilidad del cultivo si a la muestra usada para el conteo se le agrega azul de tripán u otro tipo de tinción (por ejemplo fluorescente). Si bien su rapidez y costo (menor que un citómetro de flujo) lo hacen más atractivo para su uso en un laboratorio, su precisión suele estar limitada gracias al error estadístico producto del conteo en solamente un pequeño volumen del total del cultivo. Además, suelen estar diseñados para cierto rango de tamaños celulares y para una cantidad limitada de líneas celulares.

1.2. Redes Neuronales Artificiales

Las redes neuronales artificiales (ANN) son modelos inspirados en la estructura neuronal del cerebro. Una red neuronal artificial, al igual que el cerebro, es capaz de aprender a resolver una variedad de problemas a partir de la experiencia, de manera inductiva. En este sentido, a diferencia de la programación tradicional, una ANN implica la creación de redes masivamente paralelas y la capacitación de esas redes para resolver problemas específicos. Las ANN son un

modelo matemático o computacional que esta dado por neuronas artificiales interconectadas programadas para imitar las propiedades de las neuronas biológicas, donde cada neurona que forma parte de la red es capaz de recibir señales o estímulos, externos o de otras neuronas, procesarlos y transmitirlos. En una ANN, las neuronas envían y reciben señales en forma numérica [17, 18].

En una ANN, cada unidad básica (neurona) pondera la información que recibe utilizando un “peso” (distinto para cada dato de la entrada). Dado que cada red se compone de múltiples capas de neuronas, la combinación de los pesos de todas las neuronas determina cual sera la salida producida por la red al recibir una entrada específica [19, 20]. El “entrenamiento” de una red consiste entonces en encontrar cual es la combinación de dichos pesos que minimice el alguna medida del error entre la salida esperada o real y la salida que genera la red. Esto se logra presentando a la red ejemplos de diferentes entradas junto a la salidas deseadas. Al conocer la salida deseada correspondiente a una entrada determinada, la red es capaz de comparar su predicción respecto de la salida deseada y ajustar los pesos usados para minimizar el error entre ambas. Esto se realiza de forma iterativa cambiando los pesos y calculando el error en cada iteración del entrenamiento. Idealmente, al final del entrenamiento la red encuentra, en base a los ejemplos que le fueron presentados y al aprendizaje producto de la realimentación del proceso, una forma de resolver el problema específico, y de generalizar dicha solución para un gran número de casos (no solo los que le fueron presentados). Existen otros casos, sin embargo, en los cuales hacia el final del entrenamiento se da un “sobreajuste” o *overfitting* del modelo que afecta su desempeño. Dicho fenómeno suele ocurrir, cuando el entrenamiento se prolonga demasiado, cuando el conjunto de datos de entrenamiento es pequeño o cuando el modelo es demasiado complicado llevando a que el modelo solo “memorice” la información del entrenamiento y que sea incapaz de generalizar lo aprendido a más casos.

El uso de ANN ha tenido un auge gracias a las mejoras en las capacidades de procesamiento de las computadoras actuales, y debido al aumento en la cantidad de información (en forma de datos, señales, imágenes, etc) disponible públicamente. Su aplicación es variada en cualquier campo donde se requiera procesar información[17]. Además, el uso de ANN en el procesamiento de la información, presenta muchas ventajas:

1. Aprendizaje adaptativo: Capacidad de aprender a realizar tareas basado en los datos

proporcionados para el entrenamiento.

2. Auto-organización: Capacidad de crear su propio sistema de organización o representación de la información.
3. Operación en tiempo real: Los cálculos pueden ser llevados a cabo en paralelo.
4. Reconocimiento de patrones: Las redes neuronales aprenden a reconocer patrones existentes en el conjunto de datos.
5. Flexibles: Capacidad de adaptarse a la información en constante cambio.
6. Construcción de modelos: Capacidad de modelar datos complejos, reflejando su estructura, en poco tiempo.

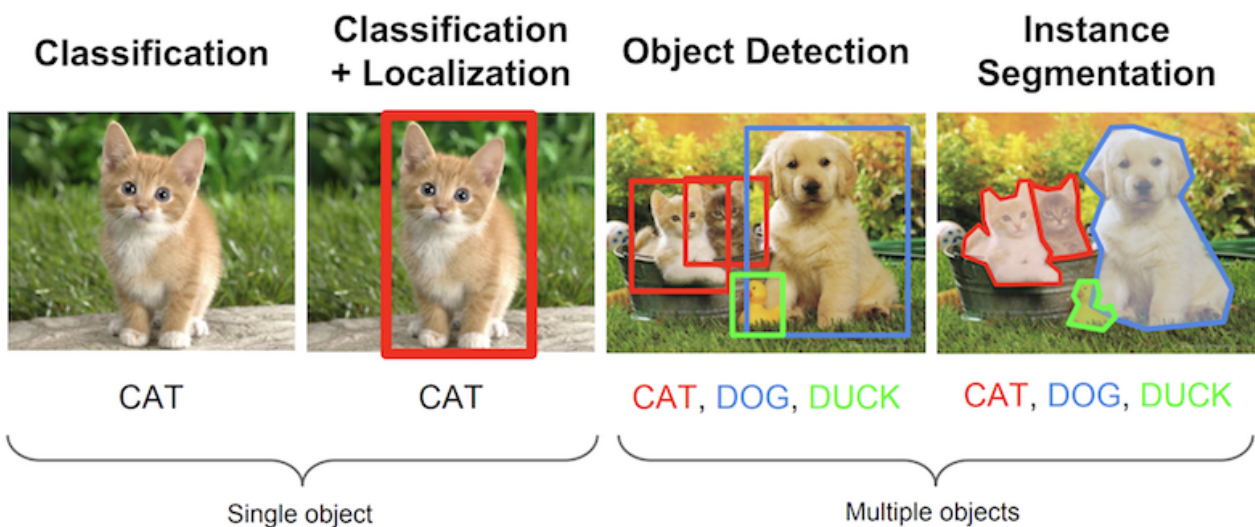


Fig. 1.5: Diferentes técnicas de análisis de imágenes. En la clasificación se le asigna una categoría a la imagen (“Gato”) basado en la presencia o no de un objeto; en la detección de objetos, más de un objeto (gatos, perros y patos) debe ser localizado y categorizado; en la segmentación de imágenes se debe encontrar, de forma precisa, la región de la imagen que contiene cada objeto individual. Figura obtenida de: Khan, A. I. & Al-Habsi, S. (2020). Machine learning in computer vision. *Procedia Computer Science*, 167, 1444-1451

En particular, las redes neuronales pueden ser utilizadas para procesar diferentes tipos de imágenes ya sea en la clasificación de imágenes, en la detección de objetos, o en la segmentación de objetos. Brevemente, la diferencia es que mientras que la clasificación de imágenes trata de

asignar una sola categoría a una imagen (en base al objeto/s que contiene), la detección de objetos intenta localizar y clasificar más de un objeto presente en la imagen y la segmentación de imágenes se propone encontrar, de manera más precisa, la porción o segmento de la imagen que contiene al objeto (Figura 1.5) [22].

El conteo celular es un problema de detección de objetos; más de un objeto (células) debe ser localizado y clasificado dentro de una imagen. En este caso, el entrenamiento de la red requiere de una gran cantidad de imágenes representativas de la realidad que contengan células similares a las que se quieren detectar y clasificar. Para el aprendizaje, es además necesario que el modelo reciba imágenes con células marcadas mediante cajas contenedoras o *bounding boxes* y clasificadas (como la tercera imagen a la derecha de la figura 1.5).

1.2.1. Red Neuronal: *You Only Look Once* (YOLO)

La red neuronal de tipo “You Only Look Once” (YOLO) consiste de un sistema de detección basado en una sola red convolucional que unifica los componentes separados de la detección de objetos en una sola red neuronal y predice simultáneamente múltiples cajas contenedoras (*bounding boxes*) con sus probabilidades de clase.

A diferencia de otros métodos de detección que necesitan analizar la imagen más de una vez o que deben atravesar más de una etapa para realizar la detección, YOLO tiene la capacidad de detectar todos los objetos en la imagen mirándola una única vez, razón por la cual lleva su nombre (Figura 1.6). Esto significa que la red infiere globalmente sobre la imagen completa y todos los objetos en la imagen.

YOLO (YOLOv1) La primera versión de YOLO fue introducida en 2015 por [Redmon et al.](#) El modelo propuesto divide la imagen de entrada, ya redimensionada, en una cuadrícula de $S \times S$ ($S = 7$ en el modelo original) y si el centro de un objeto cae en una celda de cuadrícula, esa celda será la responsable de detectar ese objeto. Cada celda realiza la predicción de B ($B = 2$ en el modelo original) cajas contenedoras en conjunto con un *puntaje de confianza* de la caja predicha, que refleja que tan seguro está el modelo de que la caja contiene un objeto (siendo cero si la caja no contiene ningún objeto o 1 si lo contiene), y que tan preciso cree que es la caja de la predicción (equivalente al IoU entre la caja predicha y la verdadera). El puntaje

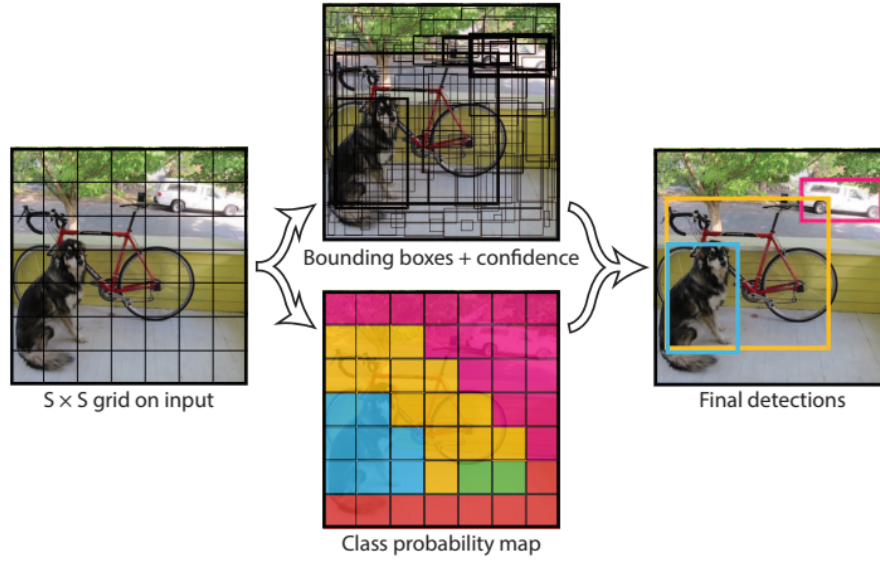


Fig. 1.6: Predicciones realizadas simultáneamente por la red neuronal convolucional YOLO. El modelo divide la imagen en $S \times S$ celdas y por cada celda predice simultáneamente B cajas, un puntaje de confianza para cada caja y C probabilidades condicionales de clase. Figura obtenida de: Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016). You only look once: Unified, real-time object detection, *Proceedings of the IEEE conference on computer vision and pattern recognition*

de confianza de la caja se obtiene multiplicando la probabilidad de que la caja contenga un objeto (P_r) y el coeficiente de intersección sobre unión entre la caja verdadera y la predicha (IOU_{pred}^{truth}):

$$C_{box} = P_r(\text{Objeto}) * IOU_{pred}^{truth} \quad (3)$$

Adicionalmente, la predicción del modelo consiste en otros 4 números; las coordenadas x e y del centro de la caja (relativas a la celda encargada de la predicción de ese objeto) y el ancho (w) y alto (h) de la caja (relativos a la imagen entera). Finalmente, el modelo también calcula en cada celda C (una por clase) probabilidades condicionales $P_r(Clase_i | \text{Objeto})$ (probabilidad de que el objeto pertenezca a la clase i dado que en la caja existe un objeto). Por esto, el *output* del modelo es un tensor de tamaño $S \times S \times (B \times 5 + C)$. Durante una prueba, por otra parte, el puntaje de confianza de cada predicción se obtiene multiplicando la probabilidad condicional de clase y el puntaje de confianza de la caja:

$$P_r(Clase_i|Objeto) * C_{box} = P_r(Clase_i) * IOU \quad (4)$$

donde $P_r(Clase_i)$ es la probabilidad de que el objeto encontrado pertenezca a la clase i . Dicho puntaje de confianza es específico para una clase en cada caja y da una medida de la confianza en la clasificación y localización del objeto predicho [1, 23-26].

En cuanto a la arquitectura del modelo, éste utiliza una sola red convolucional para predecir simultáneamente las cajas y sus probabilidades de clases. YOLO utiliza 24 capas convolucionales³ seguidas de 2 capas *fully connected* (Figura 1.7).

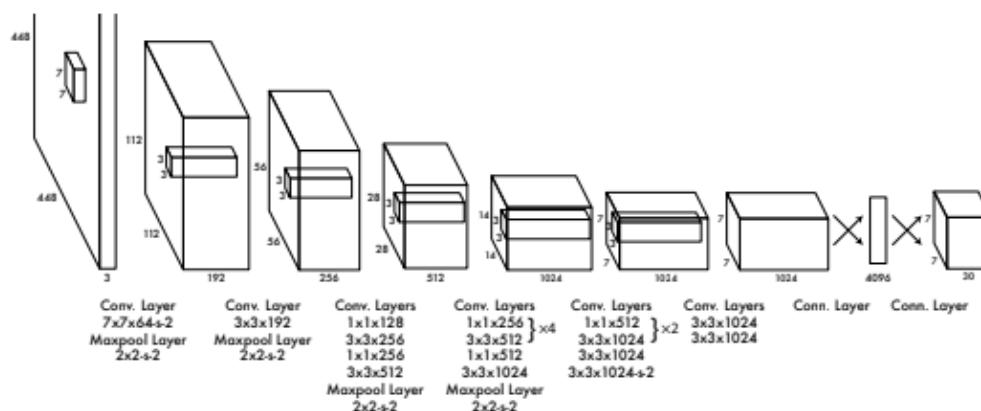


Fig. 1.7: Arquitectura YOLO. Estructura basada en capas convolucionales seguidas de 2 capas completamente conectadas. Imagen obtenida de: Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016). You only look once: Unified, real-time object detection, *Proceedings of the IEEE conference on computer vision and pattern recognition*

Durante el entrenamiento, la función de optimización usada es una función de varios términos conteniendo las sumas de los errores cuadrados (SSE) asociados a la clasificación, localización y confianza de la predicción (Figura 1.8). El error de localización toma la SSE entre las coordenadas x e y de la caja predicha y la verdadera (primer término) y la SSE entre las raíces cuadradas del ancho (w) y alto (h) de la caja predicha y la verdadera (segundo término). El error de confianza (tercer y cuarto término), por otra parte, toma la SSE entre la confianza de la predicción hecha (C) y la confianza real ($\hat{C}$) que es igual a 1 (si hay un objeto en esa grilla)

³Las neuronas en una capa convolucional aplican “filtros” (máscaras o *kernels* con parámetros predeterminados) a la imagen de entrada utilizando la operación de convolución. Esta operación permite transformar la entrada de manera de que ciertas características presentes se vuelvan más predominantes y otras menos, funcionando así, como un “extractor de características” [27-29].

$$\begin{aligned}
& \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \\
& + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \\
& + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\
& + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\
& + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned}$$

Fig. 1.8: Función de minimización. La función de minimización consiste de la suma de los errores cuadráticos relacionados a la localización del objeto (primer y segundo término) más el error de confianza atribuido a las celdas que no contienen ningún objeto (tercer y cuarto término) más el error en la clasificación del objeto encontrado (último término). Ecuación obtenida de: Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016). You only look once: Unified, real-time object detection, *Proceedings of the IEEE conference on computer vision and pattern recognition*

o 0 (si no). El error de clasificación toma, entonces, la SSE para las probabilidades predichas de todas las clases. Vale la pena aclarar que la función de optimización diseñada calcula el error de clasificación sí y solo sí se detecta un objeto en esa celda ($1_i^{\text{obj}} = 1$) y que los errores de localización y confianza se calculan solo para la caja responsable de la detección del objeto presente en la celda ($1_{ij}^{\text{obj}} = 1$; la caja predicha con mayor IOU con la caja verdadera). Por otra parte, si no hubiese ningún objeto presente en la celda analizada, o si hubiese un objeto presente pero la celda no es la responsable de realizar la detección ($1_{ij}^{\text{obj}} = 0$ y $1_{ij}^{\text{noobj}} = 1$) el único termino que se tiene en cuenta para el cálculo de error, es el de confianza. Además, para minimizar el efecto que tiene en el error aquellas celdas que no contienen un objeto, la función de minimización posee dos parámetros que incrementan el peso de los términos que calculan el error de localización ($\lambda_{\text{coord}} = 5$), y disminuyen el peso de los términos que calculan el error en la confianza cuando la celda no contiene ningún objeto ($\lambda_{\text{noobj}} = 0,5$) [25, 26].

Para evitar la repetición en la detección del mismo objeto, YOLO utiliza el algoritmo de *Non Maximal Supression* (NMS) como un paso posterior al procesamiento. Dicho algoritmo consiste en una serie de filtros mediante los cuales se eliminan detecciones y se seleccionan las

predicciones de mayor relevancia. A continuación se enumeran los pasos del mismo para cada una de las clases:

1. Descartar todas las cajas con confianza $C < C_{umbral}$.
2. Ordenar las cajas según puntaje de confianza.
3. Utilizar la caja con la confianza más alta como una predicción.
4. Descartar cualquier caja con $IOU > IOU_{umbral}$ con la caja utilizada anteriormente.
5. Utilizar las cajas restantes al igual que en el paso 3.

El descarte inicial de predicciones con puntaje de confianza menor a cierto umbral mejora el rendimiento de los pasos posteriores ya que les permite trabajar con un numero reducido de elementos. Al ordenar las mismas de manera decreciente, según su puntaje de confianza, se puede determinar la predicción que, para el modelo, tiene mayores probabilidades de contener un objeto. La misma se selecciona como una de las salidas del algoritmo de detección mientras que se descartan, de las predicciones restantes, las que posean un IOU, mayor a cierto umbral, respecto de la de mayor confianza. Finalmente, estas predicciones restantes, ya ordenadas, son utilizadas nuevamente en el paso 3 del algoritmo hasta que todas hayan sido verificadas. Si bien no es fundamental para el rendimiento del sistema de detección, NMS agrega un 2-3 % en mAP (Precisión Media Promedio) [1, 25, 26].

YOLOv2 En YOLOv1 el modelo sólo podía predecir dos cajas y una sola clase por celda, lo que le impide tener un buen desempeño en la detección de objetos pequeños y/o aglomerados. En la segunda versión de Yolo, se introdujo el uso de *anchor boxes* para permitir la detección de múltiples objetos por celda. Un *anchor box* es una caja, de un ancho y alto determinado, relativa a la cual el modelo puede predecir las cajas contenedoras o *bounding boxes* [30]. De esta manera, durante el entrenamiento, el modelo debe solo aprender y predecir *offsets* o variaciones relativas a los *anchor boxes* en lugar de realizar predicciones arbitrarias para las cajas (Figura 1.9). Los *offsets* que pueden ser predichos están, a su vez, restringidos a pequeños cambios de los *anchor boxes*. Esto estabiliza el entrenamiento (evitando grandes cambios en la función de minimización provocados por los cambios arbitrarios en las coordenadas de las cajas) y mejora

la predicción de objetos de diferentes aspectos [23]. Redmon y Farhadi propusieron 5 *anchor boxes* diferentes en base a un estudio con las imágenes de los *datasets COCO* y *VOC*.

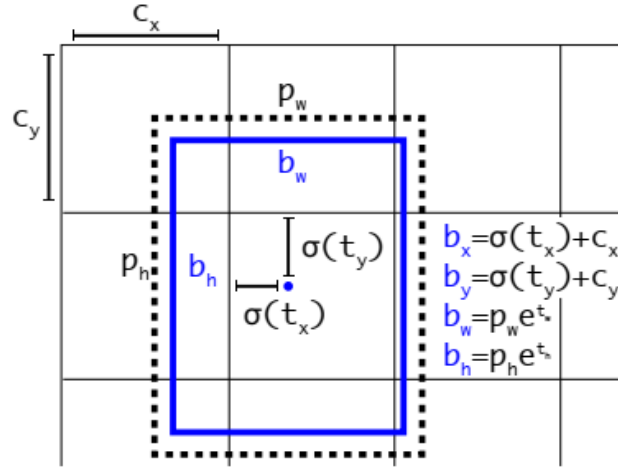


Fig. 1.9: Caja predicha en conjunto con el anchor box usado. La caja azul representa la caja predicha mientras que la caja punteada es el *anchor box* usado. Las predicciones se realizan como variaciones o *offsets* del *anchor box*, que están restringidos en su rango mediante una función sigma. Imagen obtenida de: Redmon, J. & Farhadi, A. (2017). YOLO9000: better, faster, stronger, *Proceedings of the IEEE conference on computer vision and pattern recognition*

Además de esto, YOLOv2 es entrenada con imágenes de diferentes dimensiones; cada 10 iteraciones, la red redimensiona aleatoriamente la imagen de entrada a una imagen de dimensión múltiplo de 32 de entre 320 y 608 (la imagen más pequeña siendo de 320×320 y la más grande de 608×608). Esto significa que el modelo es entrenado para realizar predicciones en imágenes de diferentes dimensiones y resoluciones [30].

En cuanto a la arquitectura, el modelo llamado Darknet-19 posee 19 capas convolucionales y 5 capas *maxpooling*. YOLOv2, al igual que todas las versiones de YOLO fueron optimizadas para implementarse mediante la estructura desarrollada por Redmon et al. llamada Darknet, un marco para redes neuronales escrito en C, C++ y CUDA, fácil de usar e instalar [31].

YOLOv3 Al igual que YOLOv2, YOLOv3 realiza las predicciones de la caja contenedora relativas a un *anchor box*. Sin embargo, en la tercera versión el calculo del error para la función de minimización cambia. En lugar de utilizar b_x , b_y , b_w y b_h (coordenadas de la caja contenedora), los errores se calculan realizando $t_* - t_*$ (diferencia de los *offsets* predichos y reales, ver figura 1.9). Además, para cada caja contenedora YOLOv3 predice un puntaje de “objetividad” utili-

zando una regresión logística, de modo que si cierto *anchor box* se superpone con el *bounding box* real más que cualquier otro *anchor box*, dicho puntaje será 1. Así, el modelo asigna un solo *anchor box* por cada objeto, y el resto de las cajas precursoras no son utilizadas en el cálculo del error de localización y clasificación. Por otra parte, YOLOv3 permite la clasificación de objetos con múltiples clases, asumiendo que no todos los objetos pertenecen a una única clase y que no todas las clases son mutuamente excluyentes (por ejemplo una mujer puede ser al mismo tiempo clasificada como *persona* y *mujer*). Esto, lo logra utilizando clasificadores logísticos individuales para cada clase en lugar de una función Softmax para calcular la probabilidad de que el objeto encontrado pertenezca a una clase determinada⁴. A diferencia de la versión anterior, YOLOv3 utiliza 9 precursores para la predicción de cajas contenedoras. YOLOv3 utiliza una arquitectura con 53 capas convolucionales sucesivas de 3×3 y 1×1 (Darknet-53). Respecto de las versiones anteriores, YOLOv3 presenta claras mejoras en la predicción de objetos pequeños [32].

YOLOv4 En 2020 [Bochkovskiy et al.](#) introdujeron la cuarta versión de YOLO con algunas mejoras en su eficiencia y desempeño en la detección de objetos. Para YOLOv4, los autores proponen y prueban un número de métodos que permiten obtener detecciones mas precisas sin aumentar (o con un leve aumento) el costo de entrenamiento ni el tiempo de detección. Si bien los métodos probados e incorporados son varios, algunos se describen a continuación.

Las versiones anteriores de YOLO utilizaban, durante el entrenamiento, diferentes herramientas de *data augmentation* para aumentar la variabilidad de las imágenes y evitar una situación de *overfitting*. Dichas herramientas constaban de transformaciones píxel a píxel (cambios en el contraste, brillo, saturación, espejados, escalados rotaciones, recortes, etc.). Por el contrario, YOLOv4 incorpora dos técnicas diferentes de aumento de datos:

- *CutMix*: consiste en “tapar” una región rectangular de una imagen mediante la superposición de otra, forzando que el modelo no memorice ciertos rasgos de los objetos detectados.
- *Mosaic*: consiste en combinar 4 imágenes diferentes en una sola, permitiendo la detección de objetos fuera de su contexto común.

⁴Un clasificador logístico individual permite calcular la probabilidad de que un objeto pertenezca a una clase en particular (conociendo sus rasgos) utilizando la función sigmoidea. La función Softmax, por otra parte, calcula todas las probabilidades que un mismo objeto tiene de pertenecer a cada clase [27]

Análogamente, YOLOv4 también utiliza el método de *DropBlock regularization* que busca “esconder” bloques de la imagen para que las capas del modelo no puedan extraer características presentes de esas regiones y el modelo no dependa exclusivamente de dichas características del objeto para llevar a cabo la detección. Otra técnica de *data augmentation* utilizada es la de *Self-Adversarial Training* (SAT) que consiste en entrenar el modelo con una imagen determinada, optimizando los pesos para dicha imagen, luego modificar la imagen degradando el desempeño de la red y vuelve a entrenar el modelo a partir de los pesos obtenidos pero con la imagen modificada [33, 34].

En esta versión de YOLO, los autores también decidieron cambiar la función de optimización utilizada durante el entrenamiento. Mientras que en las anteriores versiones se buscaba minimizar la suma de los errores cuadrados en base a las coordenadas de las cajas contenedoras, YOLOv4 utiliza la función de costo CIoU. Si se utiliza el IoU entre la caja predicha y la verdadera para estimar el error y dirigir el entrenamiento, se evita tratar las coordenadas de las cajas contenedoras como variables independientes. Esto significa no considerar al objeto de forma integral. Además, para poder comparar casos en los cuales las cajas no se superponen ($IoU = 0$), la función de costo CIoU también contempla en su cálculo la distancia entre los centros de las cajas. Cambiar la función de costo permite una mejor performance del modelo y mayor velocidad de convergencia del entrenamiento [33].

En YOLOv4 también cambia el algoritmo de *Non Maximal Supression*. Mientras que en las versiones anteriores las detecciones repetidas se evaluaban teniendo en cuenta el IoU, en YOLOv4 se determinan teniendo en cuenta el DIoU que incorpora, además del área de superposición entre cajas predichas y reales, la distancia entre los centros de las cajas. El algoritmo de DIoU-NMS permite la detección de aquellos objetos que se encuentran ocluidos por otro [34].

Los cambios mencionados no son los únicos realizados en la ultima versión de YOLO. [Bochkovskiy et al.](#) también implementaron cambios en la arquitectura, funciones de activación, normalización de datos y selección de hiperparámetros entre otras cosas [34-36].

1.2.2. Ventajas de YOLO

Una ventaja de este sistema de detección se da en el entrenamiento del mismo, dado que para ello, se utilizan imágenes completas lo que optimiza directamente el rendimiento de la detección. Otros beneficios que se obtienen utilizando este sistema se enumeran a continuación:

1. Rápida detección: YOLO toma la detección como un problema de regresión. La red se ejecuta sobre la nueva imagen y predice las detecciones.
2. Razonamiento global: YOLO ve la imagen completa e implícitamente codifica información contextual sobre las clases y su apariencia, tanto en el entrenamiento como durante las pruebas.
3. Aprendizaje de representaciones generalizables: YOLO es altamente generalizable y es menos probable que se descomponga cuando se aplica nuevos dominios o entradas inesperadas.

1.2.3. YOLO vs otros sistemas de detección

Dentro de los sistemas de detección de objetos en imagen, se analizaron algunos de los cuales, al igual que YOLO, están constituidos por redes neuronales. A continuación se hace una comparación de dichos sistemas con YOLO.

R-CNN, Fast R-CNN, Faster R-CNN y R-FCN

R-CNN⁵ utiliza propuestas de región para encontrar objetos en las imágenes. Las cajas obtenidas a partir de la búsqueda selectiva se introducen en una red neuronal que extrae características para su clasificación mediante SVM. A partir de la puntuación obtenida de la clasificación, un modelo lineal ajusta las cajas contenedoras y un método de eliminación de superposiciones no máximas suprime las detecciones duplicadas [1, 37]. Este sistema trae consigo algunos problemas:

- Los tiempos de entrenamiento tienden a ser muy largos ya que para cada imagen se deben analizar 2000 regiones de interés.

⁵Ver [37].

- La detección en tiempo real no puede ser implementada ya que le toma tiempo analizar una única imagen.
- Cada etapa de este hilo debe ajustarse con precisión de forma independiente y el sistema resultante es muy lento.
- A pesar de las similitudes que comparte con YOLO, este impone restricciones espaciales en las celdas para mitigar las detecciones múltiples de un mismo objeto disminuyendo las cajas contenedoras que puede generar.

En el algoritmo Fast R-CNN⁶ se resuelven algunos de los inconvenientes de de R-CNN para construir un método de detección de objetos más rápido. Este tiene un enfoque similar al algoritmo anterior pero utiliza un mapa de características convolucionales para encontrar las regiones de interés de una imagen y luego las reforma para introducirlas en una capa completamente conectada integrada a una capa softmax que predice las clases de las regiones y los valores de las cajas contenedoras. Fast R-CNN acorta los tiempos de clasificación obtenidos en R-CNN pero aún se basa en la búsqueda selectiva que puede llevar alrededor 2 segundos por imagen para generar propuestas de cajas contenedoras [1, 38].

Como una mejora al algoritmo anterior, se desarrolló un tercer algoritmo llamado Faster R-CNN⁷, similar a Fast R-CNN, en el que se reemplazó el algoritmo de búsqueda selectiva por una red separada para predecir las propuestas de región. Este último algoritmo es más rápido que sus predecesores aunque continua siendo más lento y menos preciso que YOLO. Además, el combinar todas las etapas en un solo modelo, como sucede en YOLO, permite una mejor optimización [1, 39].

Por último, se desarrolló un algoritmo llamado R-FCN⁸. Este consta de arquitecturas compartidas y totalmente convolucionales, como es el caso de FCN (*“Fully convolutional Network”*). A diferencia de otros detectores basados en regiones que aplican una costosa subred por región, como Fast R-CNN o Faster R-CNN, este detector es totalmente convolucional y casi todos los cálculos se comparten en la imagen completa. En este algoritmo, todas las capas son convolucionales y están diseñadas para clasificar las regiones de interés en categorías de objetos y

⁶Ver [38].

⁷Ver [39].

⁸Ver [40].

fondos. Además, el cálculo requerido por imagen se redujo drásticamente, en lugar de recortar las características de la misma capa donde se predicen los recortes, estos se toman de la última capa de características antes de las predicciones. A pesar de las mejoras respecto a su algoritmo predecesor, las etapas del proceso de detección siguen separadas [40].

SSD (*Single Shot Detector*)

SSD⁹ se basa en una red convolucional que produce una colección de cajas contenedoras de tamaño fijo junto con puntuaciones para la presencia de instancias de clase de objeto en esas cajas, seguido de un paso de supresión no máxima para producir las detecciones finales. Las primeras capas de red se basan en una arquitectura estándar utilizada para la clasificación de imágenes, llamada red base. A partir de esta estructura se anexa una estructura auxiliar a la red para producir mapas de características de múltiples escalas para la detección de objetos de diferentes tamaños, predictores convolucionales para la detección y cajas predeterminadas para cada celda del mapa de características junto con relaciones de aspecto. Este algoritmo supera en velocidad a los algoritmos previamente mencionados y compite en precisión. Además elimina por completo la generación de propuestas y encapsula todos los cálculos en una sola red. Esto hace que su entrenamiento sea más sencillo [41].

A partir de la tercera versión de YOLO, se incorporaron algunas de las características que se le atribuyen a este algoritmo mejorando su velocidad de detección y su precisión.

SPP (*Spatial Pyramid Pooling*)

SPP¹⁰ es una red convolucional similar a la estructura base mencionada en el algoritmo anterior a la que se le incorporó una capa, denominada piramidal, que agrupa las características provenientes de las capas convolucionales y generan una salida fija para las capas completamente conectadas. Esta capa intermedia agrupa las características de la imagen y genera una salida de dimensión conocida. Esta modificación a la red elimina la necesidad de deformar o recortar la imagen de entrada, lo que podría causar una modificación indeseable a los objetos de la imagen que se desean detectar [42]. A pesar de las mejoras que SPP puede aportar a una estructura de red convolucional, no es necesaria para el desarrollo en el que este trabajo se enfoca, ya

⁹Ver [41].

¹⁰Ver [42].

que las dimensiones de las imágenes de entrada no varían, y por lo tanto no se tuvo en cuenta. Además, las versiones mas recientes de YOLO, consideran cambios en los parámetros de redimensionamiento de la imagen durante el entrenamiento.

1.3. Estado del Arte

Como ya se mencionó anteriormente en la sección 1.1.3, existen en la actualidad diversas opciones basadas en el conteo automático de células como forma de resolver la potencial problemática que presenta el conteo manual (ver tabla 1).

En general, al comparar el conteo realizado manualmente con un hemocitómetro y el realizado con un contador automático, los aspectos a considerar son varios.

Precisión.- En primer lugar, la precisión del conteo es esencial sin importar el método que se elija. Un conteo con resultados erróneos significa potenciales problemas a la hora de continuar con el cultivo celular o de evaluar resultados de tratamientos, drogas, etc. La precisión de un método se evalúa puntualmente en base a dos aspectos; uno es la capacidad de distinguir una célula de una partícula o residuo y por ende obtener un resultado correcto en cuanto a la cantidad total de células en la muestra, el otro se relaciona con la capacidad de distinguir una célula viva de una muerta definiendo adecuadamente el porcentaje de viabilidad de la muestra. Un conteo preciso es, entonces, aquel que logra contar la totalidad de las células correctamente diferenciando, a su vez, las células vivas de las muertas de partículas y residuos. En conteos manuales, dado que el mismo es llevado a cabo por una persona, tanto la percepción de lo que es una célula y la percepción de lo que es una célula muerta son subjetivas, y dependen del juicio y experiencia del operador. Esto es, potencialmente, una fuente de errores para el conteo y podría afectar la precisión del conteo (manual) respecto de los resultados del conteo total de células y de porcentaje de viabilidad [16]. Los contadores automáticos, por otra parte, carecen de esta “subjetividad” dado que depende en algoritmos de clasificación basados en información obtenida de imágenes u otras fuentes. En la práctica, sin embargo, se ha visto que, en general, los operadores logran resultados similares con hemocítoímetros a los que logran los contadores automáticos del mercado. En un artículo científico presentado por [Cadena-Herrera et al.](#), se obtuvieron precisiones similares al comparar resultados de dos contadores automáticos

Tabla 1: Contadores de células automáticos en el mercado.

Modelo	Marca	Principio de acción	Parámetros	Tiempo (seg)	Concentraciones (células/ml)	Tamaños (μm)	Volumen medido (μl)	Sistema óptico
NucleoCounter [®] NC-100 ^{TM1}	ChemoMetec	Análisis de imágenes de microscopia de fluorescencia con ioduro de propidio	Conteo total, células vivas y muertas, Viabilidad	30	$1 \times 10^5 - 2 \times 10^6$		2	
NucleoCounter [®] NC-200 ^{TM2}	ChemoMetec	Análisis de imágenes de microscopia de fluorescencia con naranja de acridina y DAPI	Conteo total, células vivas y muertas, Viabilidad	50 - 120	$5 \times 10^4 - 5 \times 10^6$		1.4	Lente con magnificación $\times 1,3$, Sensor CMOS 1/2"
NucleoCounter [®] NC-202 ^{TM3}	ChemoMetec	Análisis de imágenes de microscopia de fluorescencia con naranja de acridina y DAPI	Conteo total, células vivas y muertas, Viabilidad, Tamaño celular, Residuos Celulares	30	$5 \times 10^4 - 1 \times 10^7$		1.35	Lente con $\times 2,1$ de magnificación, Sensor CMOS 2/3"
CellDrop TM BF ⁵	DeNovix Inc.	Análisis de imágenes de microscopia de campo claro con o sin azul de tripán	Conteo total, células vivas y muertas, Viabilidad, Tamaño celular	3	$7 \times 10^2 - 2,5 \times 10^7$	4 - 400	-	Cámara Basler daA2500-14um: Sensor CMOS ON Semiconductor MT9P031 de 5 MP (2592 x 1944 pixels). Iluminación: LED 530 nm
CellDrop TM FL ⁵	DeNovix Inc.	Análisis de imágenes de microscopia de fluorescencia con naranja de acridina y DAPI o microscopia de campo claro con o sin azul de tripán	Conteo total, células vivas y muertas, Viabilidad, Tamaño celular	8.5	$7 \times 10^2 - 2,5 \times 10^7$	4 - 400	-	Cámara Basler acA2040-55um: Sensor CMOS Sony IMX265 de 3 MP (2048 x 1536 pixels). Iluminación: LED 470 nm
Corning [®] Cell Counter ⁶	Corning & CytoSmart	Análisis de imágenes de microscopia de campo claro con o sin azul de tripán	Conteo total, células vivas y muertas, Viabilidad, Tamaño celular,	<3	$5 \times 10^4 - 1 \times 10^7$	10 - 70	0.3	Magnificación total de $\times 200$, Sensor CMOS 5MP. Iluminación LED
LUNA ^{TM7}	Logos Biosystems	Análisis de imágenes de microscopia de campo claro con o sin azul de tripán	Conteo total, células vivas y muertas, Viabilidad, Tamaño celular, Factor de dilución	<7	$5 \times 10^4 - 1 \times 10^7$	5 - 60	0.5	Sensor de 5MP
LUNA-FL ^{TM8}	Logos Biosystems	Análisis de imágenes de microscopia de fluorescencia o de campo claro con o sin azul de tripán	Viabilidad, Conteo total, células vivas y muertas, Viabilidad, Tamaño celular, Factor de Dilución	10 - 30	$5 \times 10^4 - 1 \times 10^7$	5 - 60	0.5	Sensor CMOS de 5MP. Iluminación LED
LUNA-II ^{TM9}	Logos Biosystems	Análisis de imágenes de microscopia de campo claro con o sin azul de tripán	Conteo total, células vivas y muertas, Viabilidad, Tamaño celular, Factor de dilución	10 - 15	$5 \times 10^4 - 1 \times 10^7$	8 - 30	0.5	Sensor de 5MP
Guava Muse [®] 11	Luminex Corporation	Citometría de flujo basado en fluorescencia	Conteo total, células vivas y muertas, Viabilidad, Tamaño celular, Proliferación celular, Apoptosis, Ciclo Celular, Daño de ADN, Autofagia, Inmunología, etc.	60 - 120	$1 \times 10^4 - 5 \times 10^6$	2 - 60		Sensor: fotodiodos Iluminación: láser verde 532-nm.

Continúa en la página siguiente

Tabla 1 – Continuada de la página anterior

Modelo	Marca	Principio de acción	Parámetros	Tiempo (seg)	Concentraciones (células/ml)	Tamaños (μm)	Volumen medido (μl)	Sistema óptico
TC20™ ¹²	Bio-Rad	Análisis de imágenes de microscopia de campo claro con o sin azul de tripán	Conteo total, células vivas y muertas, Viabilidad, Tamaño celular	30	$1 \times 10^5 - 5 \times 10^6$	6 - 50		
Cellometer Auto T4 ¹³	Nexcelom Bioscience	Análisis de imágenes de microscopia de campo claro con azul de tripán	Conteo total, células vivas y muertas, Viabilidad, Tamaño celular	<30	$2,5 \times 10^5 - 1 \times 10^7$	5 - 60		Magnificación $\times 4$
ADAM MC ¹⁴	Bulldog Inc.	Bio Análisis de imágenes de microscopia de fluorescencia con yoduro de propidio	Conteo total, células vivas y muertas, Viabilidad, Tamaño celular	45 - 120	$5 \times 10^4 - 4 \times 10^6$		3 - 8.5	Magnificación $\times 4$, Sensor CCD, Iluminación LED 532nm
ADAM MC2 ¹⁵	Nanoentek	Análisis de imágenes de microscopia de fluorescencia con yoduro de propidio	Conteo total, células vivas y muertas, Viabilidad, Tamaño celular	25 - 50	$5 \times 10^4 - 4 \times 10^6$		3.4 - 8.6	Magnificación de $\times 4$, Sensor CMOS, Iluminación LED Verde
R1 ¹⁶	Olympus	Análisis de imágenes de microscopia de campo claro con o sin azul de tripán	Conteo total, células vivas y muertas, Viabilidad, Tamaño celular	10 - 15	$5 \times 10^4 - 1 \times 10^7$	8 - 30	0.5	Sensor de 5 MP
Countess™II ¹⁷	Thermo Fisher Scientific	Análisis de imágenes de microscopia de campo claro con azul de tripán	Conteo total, células vivas y muertas, Viabilidad, Tamaño celular	15	$1 \times 10^5 - 4 \times 10^6$	7 - 60	0.38	Magnificación $\times 2,5$, Sensor de 5MP
Countess™II FL ¹⁷	Thermo Fisher Scientific	Análisis de imágenes de microscopia de fluorescencia o de campo claro con azul de tripán	Conteo total, células vivas y muertas, Viabilidad, Tamaño celular, Apoptosis, Transfección	15	$1 \times 10^5 - 4 \times 10^6$	7 - 60	0.38	Magnificación $\times 2,5$, Sensor de 5MP
Scepter™2.0 ¹⁸	Millipore Sigma	Basado en resistencia	Conteo total, Tamaño celular,	30	$1 \times 10^4 - 5 \times 10^5$ o $5 \times 10^4 - 1,5 \times 10^6$	3 - 17 o 6 - 36	50	Sensor de $60\mu\text{m}$ o $40\mu\text{m}$

¹ Chemometec. (2016). NucleoCounter® NC-100™. https://chemometec.com/Downloads/Brochures/NC100/990-0002_NC-100%20Brochure_US-web.pdf² Chemometec. (2018). The NucleoCounter® NC-200™. <https://chemometec.com/cell-counters/cell-counter-nc-200-nucleocounter/#1511446059049-3250ad53-0078>³ Chemometec. (2020). NucleoCounter® NC-202™. https://chemometec.com/wp-content/uploads/2019/04/NC-202-Brochure_280220_ml.DIGITAL.pdf⁵ DeNovix. (2020). CellDrop® Count Cells without slides. https://www.denovix.com/DeNovix_CellDrop_eBrochure.pdf⁶ Corning. (2019). Corning® Cell Counter. <https://www.corning.com/catalog/cls/documents/sell-sheets/CLS-EQ-129.pdf>⁷ Biosystems, L. (2014a). Luna™ Automated Cell Counter. <https://logosbio.com/download/luna?wpdmdl=1215&refresh=5e93c5e0eadcc1586742752>⁸ Biosystems, L. (2014b). Luna fl™ The Ultimate Automated Cell Counter. <https://logosbio.com/download/luna-fl?wpdmdl=1273&refresh=5e93a37ec89141586733950>⁹ Biosystems, L. (2015). Luna II™ Automated Cell Counter. <https://logosbio.com/download/luna-ii?wpdmdl=1236&refresh=5e93a37b2a29c1586733947>¹¹ Luminex. (2019). Guava® Muse® Cell Analyzer. <https://www.luminexcorp.com/wp-content/uploads/2019/05/BR168289.FlowCyt.Muse.WR..pdf>¹² Inc., B.-R. L. (s.f.). TC20™ Automatec Cell Counter. https://www.bio-rad.com/webroot/web/pdf/lsr/literature/Bulletin_6282.pdf¹³ Nexcelom. (2010). Cellometer® Auto T4 Automated Bright Field Cell Counter. <https://www.nexcelom.com/Literature/Cellometer-Auto-T4-automated-cell-counter.pdf>¹⁴ Bio, B. (2008). ADAM A New Standard of Automatic Cell Counter. https://www.bulldog-bio.com/wp-content/uploads/2019/05/adam_MC_brochure.pdf¹⁵ NanoEnTek. (2018). ADAM MC2 A New Standard of Automatic Cell Counter. http://www.nanoentek.com/theme/nanont2-en/shop/02/product01-view.php?it_id=1547538805#detail-spec¹⁶ Olympus. (2015). Cell Culture Essentials. https://www.olympus-lifescience.com/en/.downloads/download/?file=855639643&fl=en_US&inline¹⁷ Scientific, T. F. (2015). Countess II Automated Cell Counters. <http://assets.thermofisher.com/TFS-Assets/LSG/brochures/countess-II-automated-cell-counters-brochure.pdf>¹⁸ Millipore. (2017). Scepter™2.0 Cell Counter. <https://www.sigmaaldrich.com/technical-documents/articles/labware/scepter-handheld-automated-cell-counter.html>

diferentes y del conteo manual de muestras con tinción de azul de tripán. En otra investigación de [Shah et al.](#) se obtuvieron resultados similares, en cuanto a la precisión, al realizar conteos manuales o usando contadores automáticos basados en fluorescencia dual.

Tiempo.- El tiempo consumido por cada muestra es, si se quiere, el principal problema de los conteos manuales. En un conteo manual es necesario realizar la preparación de la muestra, colocarla en el hemocitómetro, posicionarla bajo el microscopio, realizar foco sobre las células y luego analizar cada célula individualmente. Esto, además de ser un trabajo tedioso, constituye una gran pérdida de tiempo para el operador. [Hsiung et al.](#) llevaron a cabo una comparación en los tiempos de conteo de ambos métodos, y reportaron tiempos de conteo con el método manual de hasta 3 minutos. Si bien, esto puede variar dependiendo la concentración de células en la muestra, en contadores automáticos, los tiempos de conteo no suelen superar los 60 segundos, y contadores basados en análisis de imágenes suelen tardar alrededor de 30 segundos hasta mostrar resultados (ver tabla 1). Esta diferencia puede afectar también el desempeño de los conteos manuales ya que, como se mencionó anteriormente, períodos prolongados de incubación con azul de tripán pueden afectar la viabilidad de las células introduciendo errores en los resultados.

Reporte.- En un conteo manual, el operador es capaz de reportar el conteo total de células, el porcentaje de viabilidad y las cantidades por mililitro de células vivas y muertas. A partir de esto el investigador puede estimar el total de células en su cultivo. En conteos automáticos, sin embargo, los reportes generados (por el dispositivo) suelen incluir, además de lo previo, información sobre el tamaño de las células (histogramas o diámetro medio de las células) e imágenes de la muestra analizada con marcadores gráficos en las células contadas. En contadores más complejos (como citómetros de flujo) puede surgir información acerca del ciclo celular de las células, la transfección de las mismas o del ADN, entre otras características. Además, los contadores automáticos suelen ofrecer la capacidad de realizar reportes digitales o impresos automáticamente, exportando toda la información obtenida. Esto, si bien no hace a la precisión del conteo, otorga información útil a la hora de analizar resultados.

Precio.- La diferencia entre ambos métodos respecto al precio es considerable. En el conteo manual, la inversión inicial es muy baja, y consta solamente de un microscopio óptico (presente en cualquier laboratorio de cultivo celular) y de uno o más hemocitómetros usados para el conteo. Para ensayos de viabilidad también es necesario tener azul de tripán, o cualquiera fuere el colorante usado. La independencia de cualquier dispositivo comercial sumado a que los hemocitómetros son generalmente reutilizables, hace que este método presente costos despreciables para un laboratorio. Los contadores automáticos, por otra parte suelen tener precios de lista altos; los basados en análisis de imágenes de microscopia de campo claro suelen tener valores entre US\$3,000 y US\$5,000, mientras que aquellos que utilizan ensayos de fluorescencia cuestan mas de US\$6,000 y pueden llegar a superar los US\$10,000. A su vez, es importante considerar que muchas veces estos equipos realizan otros tipos de ensayos diferentes al de exclusión con azul de tripán, por lo que los costos incluyen todos aquellos reactivos necesarios para su funcionamiento. Existen contadores que utilizan la fluorescencia como modo de tinción, mediante colorantes fluorescentes permeables a la membrana células para identificar células vivas (como por ejemplo naranja de acridina) e impermeables para identificar células muertas (como ioduro de propidio o DAPI). Este tipo de “fluorescencia” dual supone la compra y el uso de dos consumibles diferentes, lo que encarece aún más el proceso. En otros casos se utilizan otros reactivos para diluir las soluciones fluorescentes o para romper las células (lisis) y poder teñir los núcleos. Finalmente, todos los contadores automáticos hacen uso de cámaras de recuento propias y específicas para un modelo de contador o para una familia de contadores. Esto significa que no es posible utilizar hemocitómetros comerciales regulares, ni cámaras de recuento de otros contadores. En muchos casos, además las cámaras de recuento del contador no pueden reutilizarse, lo que supone un gasto aun mayor que se acrecenta con cada conteo realizado. En fin, los gastos en materia de consumibles, ya sea reactivos o cámaras, representan un gran impedimento para los contadores manuales.

Repetibilidad y reproducibilidad.- La repetibilidad de un proceso refiere a la similaridad de dos mediciones subsiguientes del mismo objeto llevadas a cabo bajo las mismas condiciones. La reproducibilidad, se define como la similitud de mediciones del mismo objeto en diferentes condiciones. En conteos, la variable más importante que puede ser considerada como una

“condición” es el operador. Entonces, siendo el objeto de medición las muestras de células, un conteo es repetible si garantiza el mismo resultado con el mismo operador, y reproducible si diferentes operadores pueden obtener el mismo resultado [16]. De esta manera, cuando se debe analizar y comparar métodos de conteo, uno de los aspectos más relevantes es la varianza de resultados inter e intra-operador. En el conteo manual, esta variabilidad está inherentemente asociada a la “subjetividad” propia del método mencionado anteriormente. El hecho de que cada operador se rija por sus propios criterios, conocimiento, percepción y experiencia a la hora de clasificar las células hace que la variabilidad inter-operador suela ser alta. Por otra parte, existen también situaciones comunes en el conteo como la presencia de residuos en la muestra, aglomeración de células o presencia de partículas, que pueden inducir errores en el conteo provocando un aumento en la variabilidad intra-operador. Como contrapartida, la automatización de los contadores supone una mejora en este aspecto dada la ausencia de un operador humano que evite la subjetividad en la clasificación de células. En general, para medir la variabilidad (ya sea intra o inter operador) se suele utilizar el coeficiente de variación (CV) como parámetro. El mismo se define como la relación entre el desvío estándar de la población (de conteos) y su media o $CV = \frac{\sigma}{\mu}$. Sabiendo esto, en un estudio, [Cadena-Herrera et al.](#) compararon dos contadores automáticos con el conteo manual corriente, obteniendo CVs de entre 3 % y 8 % para el conteo manual, y de 2 % y 5 % para los contadores automáticos. Por otra parte, [Shah et al.](#) compararon la eficiencia del conteo automático con un dispositivo contra la del conteo con hemocitómetro y obtuvieron un CV intra personal promedio de 8 % para el conteo manual y uno del 4 % para el conteo automático respecto de la densidad celular. Además, en este tipo de mediciones, las variaciones interpersonales fueron de una media de 12 % y 4 % para el conteo manual y automático respectivamente. Por otra parte, al medir la viabilidad de los cultivos, el conteo manual mostró un coeficiente de 8 % y 9 % en variaciones intra e interpersonales comparado con un 2 % y 8 % para el conteo automático en los mismos aspectos. [Hsiung et al.](#) llevaron a cabo una comparación de los métodos manuales y automático usando diferentes áreas de conteo y a diferentes concentraciones celulares. Para diferentes operadores contando la misma muestra, se obtuvieron CVs de entre 7.1 % a 15.6 % (respecto de la cantidad total de células) mientras que con el contador automático este coeficiente no superó el 2.4 % en ninguna ocasión. Finalmente, [Piccinini et al.](#) analizaron la variabilidad del conteo manual en determinaciones

de viabilidad y densidad celular, obteniendo CV interpersonales de aproximadamente 5 % para viabilidad celular y del 20 % para la densidad. Los errores interpersonales, en cambio, llegaron hasta el 21 % en mediciones de viabilidad del cultivo. Todo esto indica, en pocas palabras, que el método manual de conteo tiende a presentar variaciones mas amplias en sus resultados y, por tanto, son menos fiables en cuanto a la reproducibilidad y repetibilidad de los ensayos.

Versatilidad.- La versatilidad del método es, tal vez, otra de las grandes ventajas del conteo manual. La utilización de contadores automáticos, implica la dependencia del resultado en algoritmos de clasificación y detección que sólo fueron diseñados para aplicarse en un numero limitado de líneas celulares o que solo pueden aplicarse con cierto tipo de colorante. En el método manual, la presencia de un operador humano hace que el conteo puede ser aplicado a un número mas amplio de situaciones.

Volumen de análisis.- En el conteo celular, el volumen del cultivo analizado es siempre un limitante para la validez de los resultados. Realizar un conteo sobre una muestra pequeña del cultivo implica la posibilidad de errores estadísticos dado que lleva a un aumento en la variabilidad. En efecto, pequeñas diferencias en un conteo de una muestra pequeña, pueden llevar a grandes diferencias en la concentración total del cultivo. En el método manual, cada cuadrícula (ver Figura 1.3) tiene un área de 1 mm^2 . En un hemocitómetro común la profundidad del área de conteo suele ser de 0.1 mm, lo que significa que el volumen de cada cuadrícula seria de 0.1 μl . En general, se utilizan entre 1 y 4 cuadrados (dependiendo de la concentración de células), lo que significa un volumen de muestra de entre 0.1 μl y 0.4 μl . Si bien en el conteo manual el tamaño de la muestra puede ser bajo, hasta la actualidad no existen tantos dispositivos en el mercado que utilicen volúmenes mayores que 0.5 μl (ver tabla 1). Esta es, entonces, una gran barrera para mejorar la precisión y confiabilidad de los contadores automáticos.

A modo de resumen, se puede decir que, si bien las opciones en el mercado de contadores automáticos son varias y diversas, su costo, poca versatilidad y falta de mejoras significativas en cuanto a la precisión, hace que la gran mayoría de los laboratorios e investigadores (71 % según Barghshoon) elijan seguir usando el método manual a la hora de realizar conteos.

1.3.1. Desarrollos *open-source*

El termino *open source* se utiliza indistintamente para hacer referencia a todos aquellos desarrollos o innovaciones en las cuales el creador permite la “colaboración abierta” de otras personas mediante la distribución libre y gratuita de códigos, documentación, planos, modelos y otros archivos pertinentes al producto.

En lo que concierne a este proyecto, en los últimos años han habido algunos desarrollos destacables. [Purwar et al.](#) lograron el desarrollo de un microscopio invertido compacto (10 cm×5 cm×5 cm), portable (~5 kg) y de bajo costo (~US\$ 300). En ese caso, la construcción de dicho microscopio fue lograda usando dos lentes objetivo de microscopios comerciales diferentes y cuatro sensores de imágenes diversos mas un diodo emisor de luz como fuente de iluminación. El microscopio desarrollado demostró ser capaz de captar imágenes de células vivas y de objetos de dimensiones submicrométricas con resoluciones ópticas y campos de visión comparables a las de un microscopio óptico comercial. El proyecto *Open-flexure* revisado en [63] y [64] consta de un microscopio *open-source* personalizable e imprimible en 3D, útil para propósitos educacionales, de investigación y de diagnóstico. El mismo consta de una parte óptica realizable con una *webcam* de bajo costo, que puede ser modificada usando lentes objetivos de microscopios y ofrece la posibilidad de cambiar el tipo de iluminación, permitiendo obtener imágenes por fluorescencia o por contraste además de las de campo claro. [Aidukas et al.](#) reportaron una nueva técnica de microscopia de bajo costo implementada con una placa computadora (SBC) comercial y una cámara combinada con una técnica de reconstrucción de imágenes que permite la obtención de imágenes de alta resolución con un amplio campo de visión. El microscopio es de bajo costo (~US\$150) y portable (6 cm×9 cm×11 cm y ~200 g), y aprovecha elementos disponibles en el mercado y una estructura imprimible en 3D para obtener imágenes de alta resolución (hasta 780nm) y un campo de visión amplio (4 mm²) aplicables para cultivos celulares u otros usos.

[S. I. Kim et al.](#) crearon un “mini-microscopio” portable (8 cm×10 cm×9 cm) usando componentes disponibles en una cámara web comercial y un diodo emisor de luz. Según reportaron, el microscopio pudo ser usado dentro de una incubadora para llevar a cabo el monitoreo de células vivas en un ensayo de migración. [Walzik et al.](#) desarrollaron una plataforma de bajo costo utilizable para la adquisición de imágenes de células vivas en largos períodos. La misma consta

de una incubadora capaz de mantener constantes las condiciones de cultivo (temperatura, CO₂, humedad) mas un microscopio elaborado a partir de la lente y el sensor de cámara web comercial fijada a una plataforma motorizada conectada a un microcontrolador. El sistema completo resulto ser pequeño (22 cm×22 cm×22 cm), liviano (3.6kg) y de bajo costo (<€1250), y con resoluciones compatibles con la adquisición de imágenes para monitoreo de cultivos celulares.

A su vez, en los últimos años han surgido nuevos desarrollos de microscopia basados en teléfonos *smartphones* como los de Cybulski et al., un microscopio de muy bajo costo que puede ser armado en menos de 10 minutos a partir de una hoja de papel y una lente esférica, Smith et al., un espectrómetro y un microscopio basados en un sistema de lentes montables sobre un teléfono celular y Breslauer et al., un accesorio para un teléfono celular que permite la obtención de imágenes de microscopia de campo claro y fluorescencia.

En cuanto al conteo celular, existen diversos métodos utilizados basados en el procesamiento de imágenes de cultivos u hemocitómetros. Uno de los desarrollos de software más utilizados para el conteo de células, es *ImageJ*¹¹, un programa *open source* de procesamiento de imágenes digitales creado por el *National Institutes of Health* (NIH). El mismo, entre otras cosas, permite el conteo automático o manual de células a partir de imágenes importadas [70, 71]. *OpenCFU* es una herramienta abierta¹² que permite la detección y conteo de células en cultivos y de otros objetos circulares a partir de vídeos e imágenes digitales de una manera rápida y precisa. También existen numerosas investigaciones y proyectos de detección, clasificación y conteo de diferentes tipos de células utilizando herramientas de inteligencia artificial y redes neuronales [72-82].

En resumen, es evidente que si bien los desarrollos abiertos de sistemas de microscopia y de *softwares* de conteo son varios, hasta el momento no existen contadores celulares automáticos para su uso en laboratorios de cultivo celular que implementen tecnologías de bajo costo y que constituyan un producto *open source* en cuanto a su construcción física (*hardware*) y *software*. Asimismo, el elevado costo de los contadores automáticos comerciales sumado a su escasa mejora en la precisión respecto del conteo manual hace que muchos laboratorios opten por el método tradicional a la hora de realizar conteos celulares.

¹¹Disponible en <https://imagej.nih.gov/>

¹²Disponible en <http://opencfu.sourceforge.net/>

1.4. Objetivos

La realización de este proyecto estuvo guiada por objetivos, de mínima y máxima, que fueron establecidos inicialmente con el propósito de aplicar un conjunto de conocimientos variados que un bioingeniero puede utilizar. A continuación se listan los objetivos establecidos.

1.4.1. Objetivos de mínima

- Que el dispositivo sea capaz de realizar foco en la muestra manualmente y que permita obtener imágenes microscópicas de alta definición.
- Que el dispositivo sea capaz de identificar células vivas y muertas en suspensiones teñidas con Azul de Tripán.
- Lograr que el conteo sea rápido (menor a 120 segundos).
- Lograr el conteo en cultivos de distintos tipos celulares.
- Uso de celdas de conteo reutilizables y disponibles en el laboratorio.

1.4.2. Objetivos de máxima

- Poder realizar el conteo sin tinción.
- Detección automática de muestra con o sin tinción.
- Que el algoritmo de conteo sea capaz de identificar células aglomeradas, y contarlas.
- Conteo en Co-cultivos (células de diferentes diámetros).
- Poder realizar el enfoque automático.
- Implementar el dispositivo en conjunto con una interfaz gráfica que permita al usuario establecer el factor de dilución de la muestra.
- Que la interfaz permita la visualización de la imagen tomada.
- Poder guardar en memoria los resultados del conteo.

- Poder guardar en memoria las imágenes.
- Comunicar el dispositivo con la PC o vía USB.
- Lograr que el conteo sea rápido (menor a 30 segundos).
- Lograr un dispositivo portable.

1.5. Motivación

Nuestro interés radica en la aplicación de distintas disciplinas dentro de un solo proyecto. Uno de los trabajos del bioingeniero consiste en generar puentes para poder realizar proyectos multidisciplinarios orientados al desarrollo de tecnologías que trabajen sobre el cuerpo humano o muestras biológicas. Esto implica que un bioingeniero debe tener conocimientos de cada una de ellas y debe saber aplicarlas. Este proyecto nos permitió trabajar analizando muestras biológicas para desarrollar un dispositivo electrónico que le permita a los profesionales, que desarrollan actividades dentro del laboratorio, el conteo celular de manera rápida y eficaz. Por otro lado, este dispositivo tiene por objetivo ser utilizado dentro del laboratorio de bioingeniería del Instituto Tecnológico de Buenos Aires para prácticas y futuras investigaciones relacionadas con cultivos celulares. Es decir que el dispositivo desarrollado debe estar a la altura de un dispositivo de laboratorio.

2. Materiales y Métodos

En esta sección se mencionan los materiales usados y se detallan los protocolos y pasos seguidos en el diseño y desarrollo del dispositivo. La misma fue dividida en tres partes; en la primera se explica todo lo relacionado a los diferentes componentes elegidos para el diseño del hardware; en la segunda se menciona la estrategia y protocolos para la construcción de un conjunto de imágenes (*dataset*) usadas en el proceso de aprendizaje de la red neuronal detallada en la tercera sección.

Durante la etapa de desarrollo se tomaron decisiones con el objetivo de evitar determinados factores que aumenten el error dentro del sistema de conteo. Se determinó como primer paso del proyecto la construcción de un dispositivo que permitiera tomar fotos con la misma cámara que

el dispositivo resultante del proyecto. Esto permite la construcción de un dataset compuesto por imágenes con características similares a las que va a recibir la red al momento de tener que realizar una prueba de conteo. El efecto final de esto es un aumento en la eficiencia del aprendizaje de la red neuronal, que garantiza la precisión en la detección.

El orden natural de las tareas realizadas fue, entonces, acorde a como se presentan los siguientes títulos.

2.1. Hardware

En el método manual de conteo de células, como ya se ha explicado, se suele utilizar un microscopio óptico con el motivo de visualizar las células teñidas y clasificarlas según considere el operador. Los microscopios usados en la actualidad suelen utilizar sistemas complejos de lentes que permiten alcanzar grandes magnificaciones. Un microscopio compuesto común usa dos sistemas diferentes, una lente objetivo u *objetivo* posicionada cerca del objeto, que forma una imagen en el *ocular* donde se sitúa el ojo del observador [83]. Dado que, generalmente los objetivos son intercambiables, un microscopio de estas características puede ofrecer magnificaciones que varían típicamente entre $\times 10$ y $\times 1000$ [84, 85]. A su vez, para obtener imágenes digitales a partir de un microscopio existen una gran diversidad de opciones comerciales que se basan en la adición de una cámara digital al sistema de lentes que, generalmente, puede ser conectada a una computadora. Sin embargo, esta combinación de cámara y lentes es de alto costo, y ofrecen ventajas (altas magnificaciones y resoluciones) que en el caso de un contador no serían aprovechadas.

Con el objetivo de minimizar costos, y lograr un diseño replicable en otros ámbitos, para la construcción del dispositivo se utilizaron componentes comerciales fácilmente obtenibles, en conjunto con un modelo tridimensional imprimible con tecnologías de impresión 3D. Las siguientes secciones se dedican a describir los elementos que forman el dispositivo según la función que cumplen:

- procesamiento de datos
- adquisición de imágenes
- soporte mecánico y estructura

2.1.1. Procesamiento de datos

Con el objetivo de lograr un conteo en un tiempo aproximado de 30 segundos se evaluaron distintos dispositivos sobre los cuales ejecutar la red neuronal desarrollada. Inicialmente, se propuso la utilización de una placa Raspberry Pi (Raspberry Pi Foundation, Reino Unido) que es una computadora de placa única (SBC), bajo costo y pequeño tamaño. Luego de un análisis más profundo y con el objetivo de mejorar el procesamiento, se reemplazó por una Jetson Nano™ Developer Kit (NVIDIA, Estados Unidos).

Esta placa (Figura 2.1) permite la ejecución de múltiples redes neuronales en paralelo para aplicaciones como clasificación de imágenes, detección de objetos, segmentación y procesamiento de voz dentro en una plataforma que funciona con 5V de alimentación.

Jetson Nano™ Developer Kit - Especificaciones técnicas

- GPU: 128-core Maxwell
- CPU: Quad-core ARM A57 @ 1.43 GHz
- RAM: 4 GB 64-bit LPDDR4 25.6 GB/s
- Alimentación: 5V-2A/4A



Fig. 2.1: Jetson Nano™ Developer Kit

2.1.2. Adquisición de Imágenes

Para la parte óptica del contador se utilizó el módulo cámara *Raspberry Pi Camera Module V2.1*. La decisión se basó en la versatilidad que presenta al ser compatible con placas computadoras de marca *Raspberry Pi* y otras placas, y en el bajo costo de adquisición ($\sim$ US\$25). La cámara utiliza un sensor de tecnología CMOS Sony IMX219PQH5-C con capacidad de capturar imágenes de 8MP (3280×2464) de resolución, y una pequeña lente de plástico [86].

Tabla 2: Características del Sensor.

Tipo de Sensor	Tamaño	Cantidad de Píxeles	Tamaño del píxel
CMOS Sony <i>IMX219PQH5-C</i> ¹	3.68 mm (H) $\times$ 2.76 mm (V) (4.60 mm diagonal, Formato 1/4")	3280×2464 ($\sim$ 8MP)	$1.12 \mu\text{m} \times 1.12 \mu\text{m}$

¹ Sony. (2019). IMX219PQH5-C

2.1.3. Diseño mecánico y construcción del dispositivo

Si bien la cámara utilizada tal como se presenta comercialmente no es capaz de obtener imágenes de magnificación compatibles con la microscopía, ésta fue modificada de manera que las imágenes obtenidas si lo fuesen. En general, las cámaras digitales con lentes de distancia focal fijo están diseñadas para trabajar con distancias, al objeto, grandes. Esto quiere decir, los haces de luz que incidan en la lente convergerán mas o menos a una distancia focal (f) detrás de la lente, donde se encuentra el sensor¹³. Tal es así, que, aproximando para una lente de espesor despreciable (fina), se debe cumplir que

$$\frac{1}{f} = \frac{1}{v} + \frac{1}{u} \quad (5)$$

donde v es la distancia de la lente al punto donde se forma la imagen y u la distancia de la lente al objeto o “distancia de trabajo”. En todo caso, la magnificación de cualquier sistema óptico (m) viene dada finalmente por

¹³En un caso ideal, un haz colimado proveniente del infinito convergería exactamente a una distancia f detrás de la lente ($f \approx u$).

¹⁴Dependiendo la convención de signos usada se puede considerar como $\frac{1}{f} = \frac{1}{v} - \frac{1}{u}$

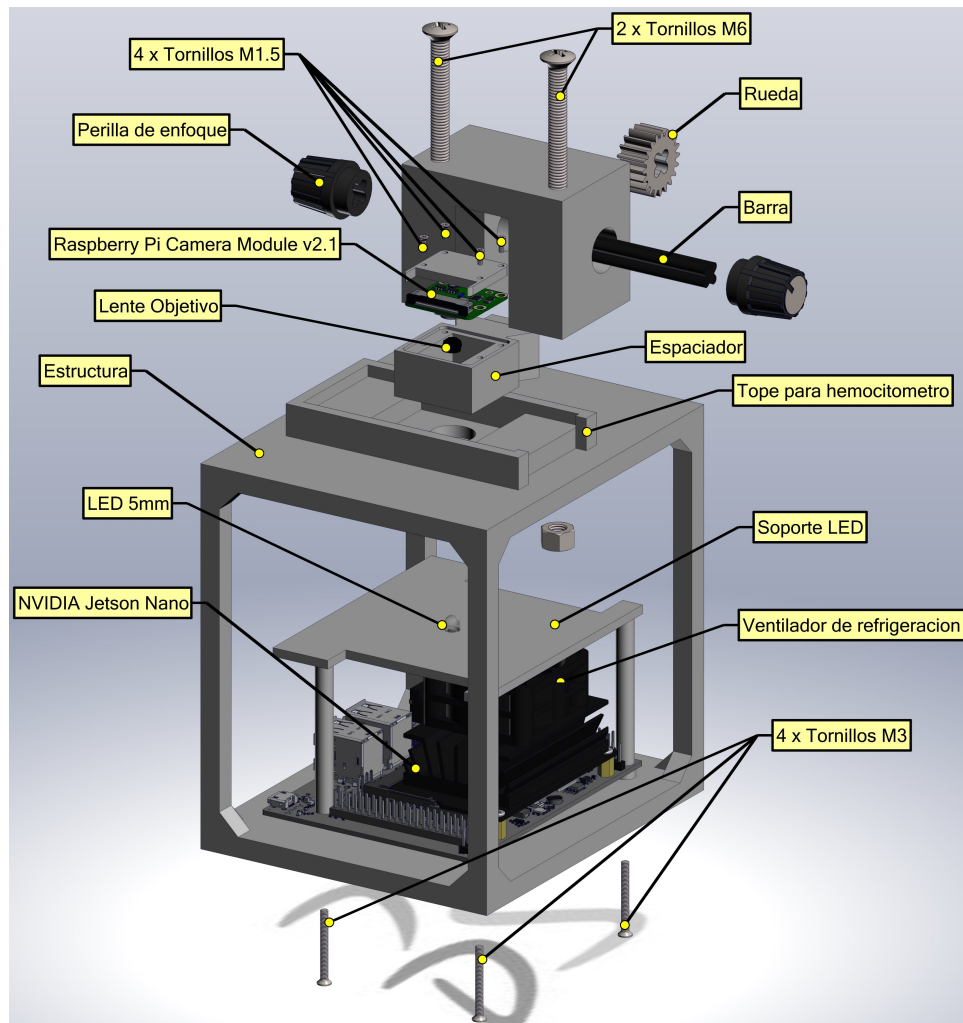


Fig. 2.2: Representación del diseño del dispositivo y sus componentes.

$$m = \frac{v}{u} \quad (6)$$

[87, 88]

Teniendo en cuenta esto, es común en la fotomacrografía¹⁵ la utilización de “fuelles” o tubos extensores que aumenten la distancia lente-sensor (v) y, por ende, la magnificación. Además, si bien no afecta la magnificación lograda, se suele revertir la lente además de alejarla del sensor

¹⁵La fotomacrografía o macrofotografía y la fotografía de aproximación consisten en técnicas que permiten tomar imágenes a distancias del objeto más cercanas que las habituales. La fotomacrografía, por su parte, incluye aquellas técnicas en las cuales se obtienen imágenes con magnificaciones entre $\times 1$ - $\times 50$. Por arriba de estos valores, se considera que se incurre en la fotomicrografía, aunque esta técnica también incluye la utilización de más de un lente, típicamente uno objetivo y otro ocular (como sucede en los microscopios).

de modo de obtener mejores rendimientos¹⁶ [89]. Dado que la cámara *Raspberry Pi Camera Module V2.1* viene con una lente de plástico que puede desenroscarse, es especialmente útil para este caso. Para el armado del sistema óptico del contador, se quitó la lente de la cámara, dejando al descubierto el sensor, y luego se la montó al inverso espaciándola del sensor por una distancia determinada. Este método es similar al utilizado en [64-66, 90-93]. La magnificación y la distancia lente-sensor fueron calculadas al momento del diseño para obtener imágenes con un campo de visión (FOV) de $2\text{ mm} \times 1.5\text{ mm}$, lo que significa un área de conteo de 3 mm^2 , equivalente a tres cuadrantes de una cámara de Neubauer (Figura 2.3). De este modo, el volumen de conteo es de $0.3\text{ }\mu\text{l}$ (en un hemocitómetro de 0.1 mm de profundidad), que está en línea con los volúmenes de análisis de contadores comerciales (tabla 1) y el volumen analizado en un conteo manual (entre $0.1\text{ }\mu\text{l}$ y $0.4\text{ }\mu\text{l}$). Considerando las dimensiones del sensor (tabla 2) la magnificación (m) necesaria para obtener el FOV deseado es de $\times 1.85$.

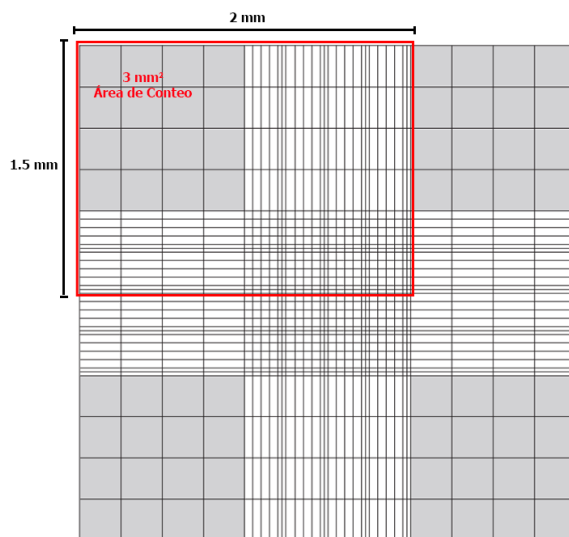


Fig. 2.3: Campo de visión y área de conteo deseada.

El diseño estructural del dispositivo (Figura 2.2) está basado en la idea de poseer un “espaciador” que permite tener el sensor y la lente separados mediante una distancia conocida. Dicho espaciador va, a su vez, montado sobre una guía que permite su desplazamiento vertical para poder hacer foco sobre la muestra. El movimiento del espaciador (junto a la cámara) en

¹⁶Dado que las lentes son generalmente fabricadas para trabajar con distancias lente-objeto grandes y lente-imagen cortas, la imagen puede sufrir aberraciones si el objeto se sitúa demasiado cerca de la lente. Por esto, revertir la lente puede ayudar al desempeño de la cámara en este tipo de casos [89].

la guía se realiza girando una o dos de las perillas que accionan, mediante un eje que gira en solidario a las perillas, un piñón que interactúa con una cremallera para mover el espaciador hacia arriba o abajo dependiendo del sentido del giro. La pareja piñón-cremallera fue diseñada con un módulo igual a uno, una longitud de la cremallera de 14mm y una rueda dentada (piñón) de 20 dientes (20mm de diámetro). El espaciador junto a la guía son inmovilizados mediante un par de tornillos a la base plana donde se posiciona la muestra a observar. Esta base corresponde a la parte superior de una estructura que mantiene al sistema elevado, y que aloja en su parte inferior al modulo procesador. La parte superior de dicha estructura posee, a su vez, un tope que demarca los limites para el movimiento del hemocitómetro o celda de recuento. El mismo fue precisamente diseñado para que, al posicionar una cámara de Neubauer, las partes visibles en una imagen tomada por el sensor coincidiesen con las grillas de conteo.

Cabe aclarar, que cada parte del diseño mecánico fue concebida para poder interactuar con el resto de los componentes. Así, las medidas del espaciador fueron pensadas para que el módulo cámara para Raspberry Pi 2.1 pudiera ser inmovilizado con cuatro tornillos, y para que la lente de plástico pudiese ser insertada a presión dentro de un agujero alineado al sensor de la cámara pero desplazado por la distancia previamente elegida. El espaciador, la guía las perillas y el par piñón-cremallera también fueron pensados de manera que el espaciador pudiese desplazarse de manera continua y estuviese situado a una distancia determinada del piñón encargado de accionar el movimiento. La base del dispositivo, por otra parte, fue ideada para alojar el módulo de procesamiento en su parte inferior, y para que tuviese una apertura circular en su parte superior concéntrica a la lente objetivo, de manera que los rayos de luz pudiesen llegar al sensor pasando a través del agujero, la muestra y la lente.

Las medidas de la cámara y la placa utilizadas para el diseño fueron obtenidas a partir de los dibujos mecánicos proporcionados por el fabricante, y disponibles en <https://www.raspberrypi.org/documentation/hardware/raspberrypi/mechanical/> y <https://developer.nvidia.com>. El diseño de todas las piezas se realizó usando el software SOLIDWORKS (Dassault Systèmes SolidWorks Corporation), mientras que las impresiones en 3D fueron realizadas sobre un modelo similar al de la Prusa i3.

Como fuente de iluminación de la muestra se utilizó un diodo emisor de luz (LED) blanco de 5 mm posicionado directamente debajo de la muestra y en el eje de la lente y el sensor. Para

Tabla 3: Componentes comerciales necesarios para la construcción del dispositivo.

Componente	Cantidad	Marca	Uso	Precio
Jetson Nano™ Developer Kit 3	1	NVIDIA	Procesamiento de datos	(~US\$120)
Raspberry Pi Camera Module V2.1	1	Raspberry Pi	Adquisición de Imágenes	(~US\$25)
Lente de plástico	1	Raspberry Pi	Lente objetivo	-
Fuente de alimentación 5V-4A	1	-	Alimentación del dispositivo	(~US\$10)
Lente condensadora	1	-	Iluminación	(~US\$2)
LED 5 mm	1	-	Fuente de iluminación	(<US\$1)
Cooling Fan	1	-	Refrigeración del procesador	(~US\$5)
Tornillo M6 - 50mm	2	-	Sujeción de la guía a la estructura	-
Tuerca M6	2	-	Sujeción de la guía a la estructura	-
Tornillo M3 25mm	4	-	Sujeción de la placa a la estructura	-
Tornillo M1.5 5mm	4	-	Sujeción de la cámara al espaciador	-

esto, el LED fue colocado en un soporte que está por encima de la placa y que posee un agujero de 5 mm de diámetro. El centro de dicho agujero es coincidente con el eje lente-sensor. Por otra parte, se utilizó una lente de vidrio de ~ 12 mm de diámetro y ~ 15 mm de distancia focal como lente condensadora. La misma permite recoger la luz de la fuente de iluminación (LED), y producir un cono de luz que ilumina la muestra con una intensidad uniforme a lo largo y ancho de todo el campo de visión. La lente fue insertada a presión en la parte superior de la estructura en un hueco diseñado específicamente para ello y dejar pasar la luz hacia la muestra.

Dado que todos los componentes electrónicos que integran el dispositivo se conectan a la placa, la alimentación necesaria para todo el dispositivo es la requerida para el correcto funcionamiento de la misma. En Particular, la Jetson Nano posee 2 entradas de alimentación (5V-2A y 5V-4A) y un *jumper* que permite habilitar o deshabilitar cada una de ellas. De estas entradas, se seleccionó la de 5V-4A para optimizar el funcionamiento de la placa al correr la red neuronal.

Todos los componentes comerciales usados en la construcción del dispositivo se encuentran detallados en la tabla 3. El resto de los componentes fueron obtenidos mediante impresiones 3D de modelos propios.

2.2. Dataset

En el área de la *visión artificial*, la detección de objetos, consiste en localizar y clasificar objetos dentro de una imagen o vídeo. A diferencia de la clasificación de imágenes, la detección de objetos supone, además de categorizar las imágenes, encontrar la ubicación de uno o más objetos sobre la misma, dibujar una caja contenedora o *bounding box* alrededor del objeto y etiquetar esa caja con la clase del objeto que contiene. Por esto, la clave en el uso de métodos de inteligencia artificial en la detección de objetos, yace en poseer imágenes con varios objetos (que deben ser clasificados). Este conjunto o *dataset* de imágenes sirven para el entrenamiento de la red neuronal. En la actualidad, existen diversos *datasets* públicamente disponibles en Internet, que se componen de hasta varios millones de imágenes diferentes. Asimismo, en cada *dataset*, una parte de las imágenes ya se encuentra manualmente clasificada o anotada lo que permite su uso para el entrenamiento de una red neuronal. Hasta el momento, sin embargo, no existe ningún conjunto de imágenes microscópicas de células teñidas con azul de tripán en hemocitómetros.

La principal dificultad que se presenta para un grupo de investigación que se propone construir su propio conjunto de imágenes para uso en inteligencia artificial, es la necesidad de obtener una gran cantidad que debe ser procesada manualmente para añadir las etiquetas de los objetos presentes. El etiquetado manual demanda una gran cantidad de horas de trabajo por parte del operador, y constituye una labor tediosa resultando en un gran consumo de recursos humanos. En el caso de la clasificación de células, el operador debe ser capaz de distinguir el estado de las mismas (viva o muerta) para evitar la aparición de errores en el etiquetado. Asimismo, la generación de un conjunto de este tipo de imágenes implica el cultivo de varias líneas celulares diferentes, lo que significa nuevos costos para la investigación debido a la adquisición de las líneas celulares y el consumo de reactivos y de materiales de trabajo descartables. Además, para obtener una imagen de células teñidas con azul de tripán se debe llevar a cabo un conteo celular que se debe realizar siempre al mismo tiempo que el cambio de sustrato de cultivo (subcultivo). Éste se realiza, generalmente, cuando el subcultivo llega a la confluencia, que suele ser algunos días luego del sembrado de la generación anterior (dependiendo de la tasa de duplicación de la línea usada). Esto implica que el tiempo entre dos conteos sucesivos (y por ende los momentos en los cuales se pueden obtener imágenes) puede llegar a ser de varios días.

Teniendo esta problemática en cuenta, se adoptaron las estrategias explicadas en la siguiente subsección para la construcción de un conjunto de imágenes que permitiese un entrenamiento apropiado del modelo.

2.2.1. Dataset sintético

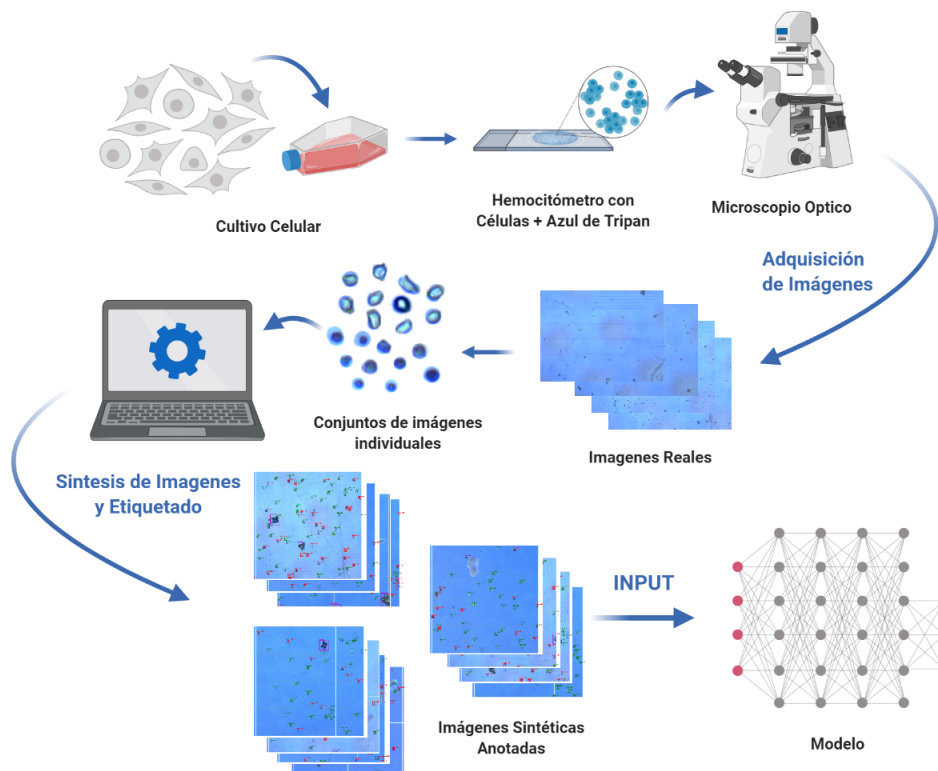


Fig. 2.4: Estrategia de generación del dataset sintético. Al momento de realizar un subcultivo, una alícuota de células en suspensión fue tomada y teñida con azul de tripán. La suspensión teñida fue colocada en un hemocitómetro y bajo un microscopio óptico se obtuvieron imágenes digitales de un campo de visión de $2\text{ mm} \times 1.5\text{ mm}$. De cada imagen se obtuvieron y clasificaron mascarlas de células individuales. Dichas células fueron usadas en la generación de un gran número de imágenes sintéticas, necesarias para el entrenamiento del modelo.

Una forma de minimizar el trabajo necesario para la construcción de un *dataset* de imágenes sintéticas es generarlas a partir de un pequeño grupo de imágenes u objetos reales. Dicha estrategia ha sido utilizada exitosamente en diferentes casos. [Ward et al.](#) lograron la segmentación de hojas de plantas del género *Arabidopsis* con una red neuronal convolucional entrenada a partir de imágenes sintéticas de plantas. En este caso las imágenes 2D de las plantas fueron generadas a partir de modelos tridimensionales de las hojas. Por otro lado, [Toda et al.](#) desarrollaron

un modelo para la clasificación y segmentación de semillas de cebada según su morfología. El entrenamiento del modelo fue llevado a cabo con 1200 imágenes generadas a partir de un grupo de 400 imágenes de semillas individuales posicionadas aleatoriamente por sobre un fondo real.

Las ventajas que presenta el uso de imágenes sintéticas incluyen la posibilidad de generar un número infinito de imágenes de entrenamiento; el etiquetado automático a partir de las imágenes reales sin la necesidad de anotar manualmente cada imagen del conjunto de entrenamiento; la posibilidad de introducir variaciones aleatorias dentro de las imágenes generadas ya sea alterando las características de los objetos o de los fondos de las imágenes.

En el presente trabajo, el entrenamiento de los modelos propuestos se llevó a cabo con diferentes *datasets* de imágenes sintéticas, cuyas características se detallan a continuación.

Preparación del dataset 21 imágenes a color fueron obtenidas utilizando una cámara de Neubauer reutilizable (Marienfeld, Alemania) y mezcla 1:1 de células NIH 3T3 en suspensión y solución de azul de tripán 0,4 %. Las imágenes fueron obtenidas con un microscopio Eclipse E200 (Nikon, Japón) utilizando una lente objetivo 4×. Los cultivos utilizados fueron realizados en el laboratorio de la universidad siguiendo los protocolos correspondientes. El campo de visión de cada imagen es de 2 mm × 1.5 mm, y su resolución de 2592 × 1936 píxeles.

Asimismo, otras 27 imágenes fueron recolectadas a partir de documentos puestos a disposición públicamente por parte de dos fabricantes diferentes de contadores celulares automáticos R1 (Olympus, Japon) y Luna II (Logos Biosystems, Corea), disponibles en [https://www.olympus-lifescience.com/en/cell_culture_solution/r1/#!cms\[focus\]=cmsContent2396](https://www.olympus-lifescience.com/en/cell_culture_solution/r1/#!cms[focus]=cmsContent2396) y <https://logosbio.com/automated-cell-counters/brightfield/luna-ii>. Dichas imágenes constan de muestras de células teñidas con azul de tripán de 21 líneas celulares diferentes: FF hiPS, MEF, A549, AsPC-1, CHO, C6, Capan-2, DU 145, 293-GFP, H1299, H4, HCT116, HeLa, Hep3B, HL-60, Hs 578T, MCF7, NIH3T3, PANC-1, PC-3, STO, U-2 y OS U937. Las células individuales de estas imágenes se hallaban previamente marcadas según su estado (vivo o muerto).

Finalmente, otras 18 imágenes de células teñidas con azul de tripán fueron obtenidas a partir de las figuras publicadas en la investigación de Chan et al. Dicha publicación estudia la ruptura de células Jurkat y de Células mononucleares de sangre periférica (PBMC) al entrar en contacto con azul de tripán. Las imágenes usadas constan de muestras de células de ambas

líneas teñidas con azul de tripán en una mezcla 1:1 a diferentes concentraciones (0.4 %, 0.2 %, and 0.1 %) utilizando los contadores automáticos Cellometer AutoT4 y AutoM10 (Nexcelom Bioscience, Estados Unidos) en campo claro con aumentos ópticos de $4\times$ y $10\times$.

De las 66 imágenes, 15 ($\approx 20\%$ del total) diferentes fueron seleccionadas y separadas para ser usadas en una instancia final de prueba del modelo. Con las 51 imágenes restantes se construyó el conjunto de imágenes *sintéticas* utilizado en la etapa de entrenamiento del modelo. Las diferentes imágenes de células individuales fueron recortadas y separadas de su fondo correspondiente y guardadas como archivos independientes. Este proceso fue realizado con Adobe Photoshop CC 2019 (Adobe Inc., Estados Unidos) que permitió generar máscaras de células individuales con un fondo transparente en formato .png. Las imágenes de células vivas y células muertas fueron almacenadas en directorios diferentes de modo de poseer dos grandes conjuntos de imágenes utilizables para la generación de imágenes sintéticas. Por otra parte, los fondos de cada imagen fueron separados mediante la edición de las imágenes originales (*borrando* los objetos presentes en cada imagen), y fueron almacenados en otro directorio. Finalmente, a partir de algunas imágenes se obtuvieron una pequeña cantidad de imágenes de residuos visualizados y de grandes grupos de células aglomerados en grandes cúmulos utilizando la misma estrategia que para las células individuales.

En total se obtuvieron imágenes de 1427 células vivas individuales y 527 células muertas. A partir de las imágenes obtenidas en laboratorio, se extrajeron 317 y 344 células vivas y muertas respectivamente. El resto de las imágenes de células individuales (1110 vivas y 183 muertas) fueron obtenidas de imágenes disponibles de contadores automáticos comerciales. De dichas imágenes, se separaron también 66 fondos diferentes, 32 residuos y 32 cúmulos.

La división de los objetos usados para el entrenamiento y la validación del modelo fue previa a la generación de las imágenes sintéticas. Esto significa que la estrategia de división consistió en separar las imágenes de células individuales, aglomeraciones, residuos y fondos evitando que el mismo objeto se hiciese presente en dos conjuntos diferentes (entrenamiento y validación). Por otra parte, los objetos y fondos de las imágenes usadas para el testeo final no formaron parte de dicha división, lo cual dejó un total de 876 máscaras de células vivas y 373 muertas usadas en la etapa de entrenamiento y validación. La división de los objetos en cada conjunto se realizó respetando una proporción de 80/20 entre los conjuntos de entrenamiento y validación.

Las cantidades y porcentajes reales logrados en cada conjunto se observan en la tabla 4.

Tabla 4: División de los objetos usados en imágenes de entrenamiento y validación.

Conjunto	Células Vivas	Células Muertas	Fondos	Células Vivas (%)	Células Muertas (%)	Fondos (%)
Entrenamiento	701	321	40	80,48 %	83,81 %	78,43 %
Validación	170	62	11	19,52 %	16,19 %	21,57 %

Generación de las imágenes Con el motivo de encontrar el método de entrenamiento de la red más apropiado, se generaron y utilizaron diferentes *datasets* para entrenar diferentes modelos.

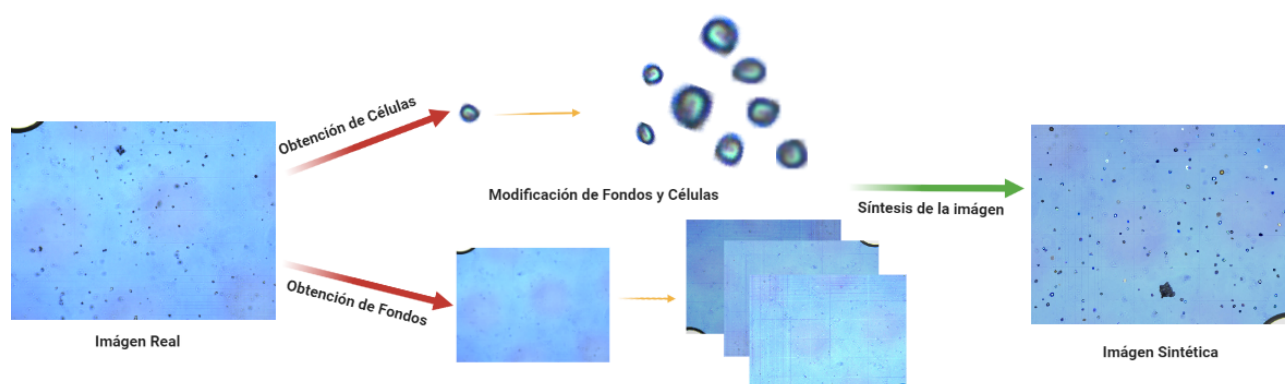


Fig. 2.5: Procedimiento para la generación de una imagen sintética.

El procedimiento de generación de imágenes sintéticas común a todos los conjuntos usados fue tal como se describe a continuación (Figura 2.5). Primero, una imagen de un fondo fue seleccionada al azar y redimensionada para ocupar un espacio de 3280×2464 píxeles (equivalente a la resolución del sensor de la cámara usada). Dicho fondo seleccionado fue, en casos aleatorios (en el resto de los casos se lo utilizó sin modificaciones), volteado y/o espejado, y sobre él se colocaron cantidades aleatorias de imágenes de residuos, aglomeraciones de células y células vivas y muertas, determinando su posición en la imagen según las coordenadas x e y del centro del objeto aleatorias. Previo a su posicionamiento cada objeto fue modificado mediante una rotación, un espejado (en algunos casos) y un redimensionamiento en ambas direcciones. Dichas operaciones permiten aumentar la variabilidad de las imágenes. El código de generación de imágenes fue diseñado teniendo en cuenta las siguientes restricciones y condiciones. En

primer lugar, se procuró que ningún objeto pegado en la imagen se superpusiera con otro colocado previamente en una proporción mayor al 50 % de su área. Además, se tuvo en cuenta que todos los objetos debían tener al menos un 50 % de su superficie dentro del área de la imagen, impidiendo así, que el centro de un objeto determinado excediese el tamaño del fondo. El redimensionamiento de las imágenes de células individuales fue realizado para que el tamaño final de cada célula estuviera en un rango de aproximadamente 11 a 100 píxeles en cada dimensión, lo que equivale a la generación de células con un diámetro en el rango de 7 a 60 μm (teniendo en cuenta que para un campo de visión de $2\text{ mm} \times 1.5\text{ mm}$ y una resolución de 3280×2464 píxeles, un píxel equivale a $\approx 0,609\text{ }\mu\text{m}$ por píxel). En adición, para evitar que las células generadas fuesen excesivamente planas o alargadas, se limitó el escalado de una de las dos dimensiones a un rango de 10 píxeles mayor o menor que la otra. Por otra parte, como los tamaños de las imágenes de aglomeraciones celulares y residuos son más heterogéneos que aquellos de las células, se decidió que el redimensionamiento de estos objetos se basase en multiplicar ambas dimensiones en la misma proporción por un factor aleatorio entre 1 y 2, evitando así que su tamaño fuese excesivamente grande o pequeño, y que su forma fuese irreal. Las anotaciones que sirven como entrada para la red neuronal se realizaron automáticamente en conjunto con la generación de las imágenes, utilizando las dimensiones y coordenadas del centro del objeto pegado para generar el ancho, alto y centro de la caja contenedora del objeto.

Si bien el proceso anterior es común a todos los *datasets* utilizados, para poder entender el efecto de las diferentes imágenes usadas para entrenar en el desempeño de la red e identificar la mejor rutina de entrenamiento, a la hora de generar cada conjunto de imágenes se introdujeron variaciones en cuanto a:

- Modificaciones en el brillo, contraste y nitidez de los fondos y los objetos de la imagen
- Cantidad de células vivas y muertas por imagen
- Cantidad de imágenes generadas
- Cantidad de imágenes negativas (sin objetos para detectar) generadas
- Cantidad de imágenes reales incluidas

Las características de cada conjunto en particular se pueden ver en la tabla 5. A grandes rasgos, los conjuntos generados constan de 1000 o 4000 imágenes positivas y 200, 1000 o 4000 negativas (proporciones de 5:1 o 1:1). La cantidad mínima de células por imagen siempre fue 1, y la máxima vario entre 60, 75, 100, 150 y 250 por clase. Teniendo en cuenta la ecuación 1, y sabiendo que; el área cubierta en cada imagen es de 3 mm^2 , la profundidad de una Cámara de Neubauer común es de 0.1 mm (volumen de conteo de $0.3\text{ }\mu\text{l}$ o $3 \times 10^{-4}\text{ ml}$), el factor de dilución igual a 2 (dilución 1:1) el rango establecido de cantidad de células totales equivale a concentraciones celulares en el rango de $1 \times 10^4 - 3 \times 10^6$ dependiendo el *dataset*. El conjunto de validación generado fue siempre de 250 imágenes y el de prueba de 15 imágenes reales.

El *dataset* de imágenes sintéticas generado en este trabajo, el conjunto de imágenes reales obtenidas y anotadas y las máscaras de células individuales individuales recortadas y clasificadas fueron publicadas y puestas a disposición de otros investigadores en [2].

2.2.2. Dataset Real

Como se mencionó anteriormente, 15 imágenes fueron separadas para el testeo final. 4 de ellas fueron obtenidas en el laboratorio mientras que las 11 restantes fueron descargadas de otras fuentes. De esta manera, se procuró que la prueba final del modelo fuese realizada con imágenes reales que nunca fueron observadas durante el entrenamiento. Las 15 imágenes usadas en la prueba contienen 551 células vivas y 144 muertas. Las anotaciones de cada imagen se realizaron manualmente utilizando el software *LabelImg*, que permite visualizar las imágenes, etiquetar los objetos y exportar las anotaciones en archivos *.txt* en el formato deseado. El software se encuentra disponible al público en <https://github.com/tzutalin/labelImg>.

2.3. Modelo

Se entrenaron y compararon diferentes modelos de la familia de redes neuronales YOLO. El proceso de selección del modelo más apto para el dispositivo incluyó comparaciones en los desempeños de diferentes redes neuronales (YOLOv3 y YOLOv4), y comparaciones en los desempeños de la misma red entrenada en condiciones diferentes. En particular, los modelos utilizados son los presentados en [32] y [33] con ligeras modificaciones. Las condiciones de entrenamiento y las modificaciones analizadas se encuentran descriptas a continuación.

2.3.1. Darknet

Todas las versiones de YOLO son implementadas mediante la estructura desarrollada por [Redmon et al.](#) llamada Darknet.

Darknet es un marco para redes neuronales escrito en C, C++ y CUDA mediante el uso de la CPU y GPU del computador. El marco Darknet es fácil de usar e instalar y está optimizado para la implementación y entrenamiento de todas las versiones de YOLO. Las instrucciones para su instalación, junto a los modelos y pesos de las diferentes versiones de YOLO se encuentran disponibles en el siguiente repositorio: <https://github.com/AlexeyAB/darknet>.

2.3.2. Entrenamiento

Se utilizó el marco Darknet para el entrenamiento y pruebas de los modelos. La etapa de entrenamiento fue llevada a cabo utilizando los servicios de *Google Colaboratory* que brinda acceso gratuito a recursos computacionales, incluidas GPUs.

Si bien se entrenaron diferentes modelos, la estrategia fue siempre la misma. Todos los modelos fueron entrenados por 6000 iteraciones, lo que demora alrededor de 60 horas, a partir de pesos pre-entrenados con el *dataset* COCO, los cuales pueden predecir objetos pertenecientes a 80 clases. El marco utilizado para la implementación (Darknet) permite guardar los pesos obtenidos cada 100 iteraciones, y ofrece, además, la posibilidad de obtener el mAP_{50} en base al conjunto de imágenes de validación cada 100 iteraciones a partir de la iteración número 1000. Dado que los pesos de diferentes momentos del entrenamiento son conservados, al momento de realizar las pruebas es posible comparar la performance de la red con los diferentes conjunto de pesos y elegir el mejor sin tener que realizar una detención temprana o *early stopping* del entrenamiento para evitar el sobreajuste del modelo a los datos presentados.

Todos los modelos fueron entrenados con una resolución de 608×608 píxeles, es decir, que durante el entrenamiento las imágenes de entrada fueron redimensionadas en imágenes de 608×608 . Sin embargo, una de las características de YOLO es que es capaz de realizar un entrenamiento multiescala. Durante el entrenamiento, el modelo escoge un factor entre 1 y 1,5 de forma aleatoria para escalar el redimensionamiento de la imagen. Esto que significa que las imágenes utilizadas fueron efectivamente redimensionadas usando anchos y altos entre 608 y 912 píxeles.

El tamaño del lote o *batch* usado en cada iteración fue de 64 imágenes (una iteración corresponde a 64 imágenes procesadas). Por otra parte el tamaño del *mini-batch* usado fue de 64 o 32 dependiendo de las limitaciones de memoria. Esto significa que el GPU procesa un numero de imágenes dependiendo del tamaño del *mini-batch*, pero que los pesos del modelo son actualizados luego de analizar el *batch* entero de imágenes.

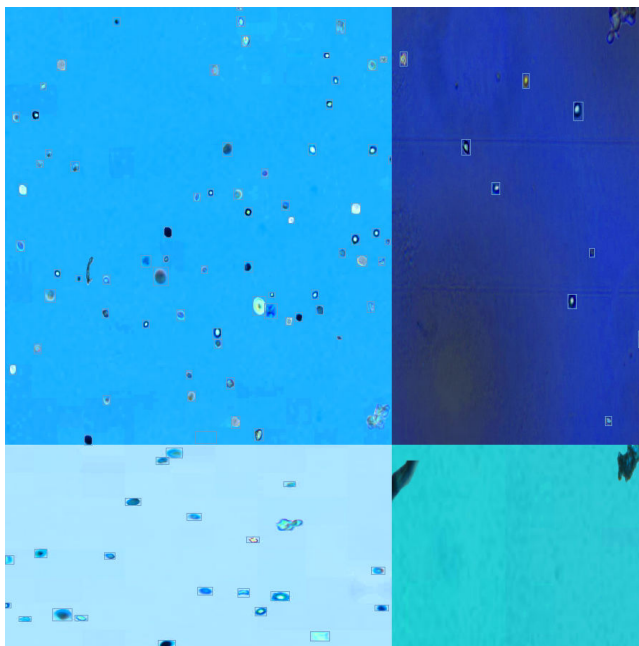


Fig. 2.6: Imagen modificada por YOLO y utilizada en el entrenamiento. Como se puede ver en la imagen al entrenarse YOLO realiza modificaciones en el tono, exposición y saturación de las imágenes y realiza combinaciones de varias imágenes en una mediante el método *mosaic*.

Como se menciona anteriormente, YOLO también utiliza diferentes métodos de aumento de datos o *data augmentation* durante el entrenamiento (Figura 2.6). Los métodos adoptados en este trabajo consisten en variar:

- Saturación: se varia aleatoriamente (hasta un factor de 1,5) la saturación de la imagen.
- Exposición: se varia aleatoriamente (hasta un factor de 1,5) la exposición (brillo) de la imagen.
- Tono (*hue*): se varia aleatoriamente el tono de la imagen.
- Mosaic: consiste en combinar cuatro imágenes en una.

El *learning rate* inicial utilizado en todos los entrenamientos fue de 0,001. Sin embargo, YOLO actualiza este parámetro al alcanzar el 80 % y 90 % del máximo de iteraciones (4800 y 5400 en este caso) multiplicando el *learning rate* actual por 0,1. Por otra parte, se utilizó un momento (*momentum*) de 0,9 y un factor de decaimiento (*weight decay*) de 0,0005, tal como lo presentan en [33]. Mientras que el momento controla cuanto afecta el historial del entrenamiento a la variación de los pesos, el *decay* limita la actualización de pesos según su valor absoluto, reduciendo en mayor medida el valor de los pesos mas grandes (relacionados a características mas frecuentes). Esto mejora el aprendizaje del modelo de características menos usuales y ayuda a eliminar el desequilibrio entre clases (ej.: el conjunto de imágenes posee un número mayor de células vivas que muertas).

Para la mayoría de los modelos, los *anchor boxes* usados durante el entrenamiento son los propuestos en [33]. Los mismos corresponden a cajas de los siguientes tamaños (en píxeles):

$$12 \times 16, 19 \times 36, 40 \times 28, 36 \times 75, 76 \times 55, 72 \times 146, 142 \times 110, 192 \times 243 \text{ y } 459 \times 401$$

Para el resto de los modelos se utilizaron *anchor boxes* optimizados para el conjunto de imágenes de este trabajo. En estos casos, 9 *anchor boxes* fueron elegidos aplicando un algoritmo de agrupamiento de *K-medias* (*K-means clustering*) con $K = 9$, usando las cajas de las anotaciones de las imágenes reales de entrenamiento [97]. Las cajas obtenidas fueron:

$$5 \times 6, 7 \times 9, 8 \times 12, 10 \times 14, 13 \times 14, 12 \times 17, 17 \times 23, 23 \times 31 \text{ y } 30 \times 39$$

En total se entrenaron y compararon 12 modelos diferentes. Si bien la estructura de la red, los parámetros e hiperparámetros se conservaron para todos los modelos, se establecieron ciertas diferencias en cada entrenamiento. Se llevo a cabo, entonces, un estudio de “ablación” eliminando o variando aspectos del modelo y/o el *dataset* usado en el entrenamiento, con el motivo de entender como afectaban al aprendizaje del modelo, y con el motivo de identificar la mejor rutina de entrenamiento del modelo de detección y clasificación de células. Las variaciones consistieron en:

- DA: Aumento de datos o *Data Augmentation* incluido en la generación de las imágenes - a través de modificaciones en el brillo, contraste y nitidez de los fondos y los objetos al generar las imágenes.

- FT: Ajuste fino de las anotaciones de los objetos - realizando un recorte más fino de las mascararas de células individuales para evitar que la anotación o *bounding box* utilizada en el entrenamiento fuese más grande que el objeto en sí.
- OA: Optimización de los *anchor boxes* - para el tamaño de las células de las imágenes reales utilizadas para el entrenamiento.
- Mix: Entrenamiento con un *dataset mixto* incluyendo, además de las imágenes sintéticas, las imágenes reales precursoras de dichas imágenes sintéticas. En caso de realizar un entrenamiento de esta manera, 24 imágenes reales del conjunto de entrenamiento fueron espejadas y/o volteadas, por lo que de cada imagen se obtuvieron 4 (una espejada, una volteada y otra espejada y volteada). De cada imagen real también se obtuvo una imagen negativa (sin las células) y de esa imagen negativa 3 más espejando y volteándolo. En total, para cada dataset mixto 192 imágenes reales (96 positivas y 96 negativas) fueron incluidas. Cabe aclarar que las imágenes reales usadas para el entrenamiento fueron obtenidas del conjunto de entrenamiento (diferente al de validación y prueba) a modo de evitar entrecruzamiento de datos y de condicionar los resultados. Las imágenes reales de entrenamiento son las mismas imágenes a partir de las cuales se obtuvieron los objetos y fondos precursores de imágenes sintéticas de entrenamiento.
- CL: Cantidad de clases utilizadas en la clasificación - De usarse dos, las clases fueron *Viva* y *Muerta*; de usarse tres, las clases fueron *Viva*, *Muerta* y *Cúmulo*.
- L/D: Cantidad máxima de células vivas (L) y muertas (D) por imagen.
- P/N: Cantidad de imágenes positivas (P) y negativas (N) generadas.

La lista y características de los modelos entrenados se encuentra en la tabla 5.

2.4. Métricas de evaluación

Como se ha mencionado anteriormente, la detección de objetos en una imagen consta en realizar predicciones sobre la misma acerca de la ocurrencia y posición de ciertos objetos. Los objetos detectados en cada imagen, vienen determinados por una caja o rectángulo que encierra

Tabla 5: Características de los modelos entrenados.

	DA	FT	OA	Mix	Cl	L/D	P/N
YOLOv3-608	✓				2	75/75	1000/200
YOLOv4-608-3C	✓				3	75/75	1000
YOLOv4-608-3	✓				2	75/75	1000/200
YOLOv4-608-4	✓				2	150/150	1000/200
YOLOv4-608-5	✓				2	100/60	1000/200
YOLOv4-608-6		✓	✓	✓	2	100/100	1000/1000
YOLOv4-608-7		✓	✓	✓	2	100/100	4000/4000
YOLOv4-608-8		✓	✓	✓	2	100/100	1000/200
YOLOv4-608-9		✓			2	100/100	1000/200
YOLOv4-608-10	✓	✓			2	100/100	1000/200
YOLOv4-608-11	✓	✓			2	100/100	1000/1000
YOLOv4-608-12	✓	✓		✓	2	100/100	1000/1000

DA: Aumento de datos en la generación de imágenes sintéticas

FT: Ajuste de las anotaciones de imágenes de entrenamiento

OA: *Anchor boxes* optimizados

Mix: Dataset mixto

Cl: Cantidad de clases

L/D: Numero máximo de células por imagen por clase (L=vivas, D=muertas)

P/N: Número de imágenes con objetos (P) e imágenes negativas (N)

ese objeto, y un porcentaje de confianza acerca de la clase de objeto que hay dentro de esa caja. Sin embargo, evaluar los resultados de una detección mediante una observación de la imagen procesada suele no ser suficiente. Para la evaluación y comparación de los modelos analizados, se utilizaron algunas de las métricas mas habituales en detección de objetos: intersección sobre unión (IoU), precisión (P), exhaustividad (R), precisión media (AP) y precisión media promedio (mAP).

IoU Para juzgar si una detección es correcta o no, se suele comparar la medida en la cual una caja predicha por el modelo y la caja verdadera se superponen. Así, esta medida de superposición entre dos cajas viene dada por:

$$IoU = \frac{\text{area de interseccion}}{\text{area de union}} \quad (7)$$

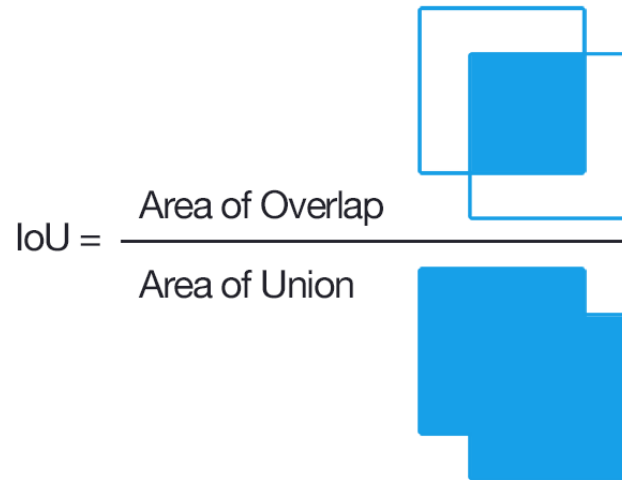


Fig. 2.7: Diagrama representativo del calculo del IoU entre dos imágenes. Imagen obtenida de: Rosebrock, A. (2017). *Deep Learning for Computer Vision with Python: Practitioner Bundle*. PyImageSearch

Para determinar si una detección corresponde a un verdadero positivo, un falso positivo o un falso negativo, se debe establecer un umbral para el IoU (una porcentaje del área que debe ser compartida por ambas cajas para considerar la predicción correcta). A su vez, es posible medir el promedio del IoU de las detecciones realizadas, obteniendo un parámetro de la exactitud de las cajas generadas por el modelo y de su habilidad en la localización de objetos [98].

Precisión La precisión de un modelo mide la exactitud de las predicciones, es decir, el porcentaje de predicciones realizadas correctamente. Se calcula como:

$$P = \frac{TP}{TP + FP} \quad (8)$$

donde TP son los verdaderos positivos, FP los falsos positivos y P la proporción de verdaderos positivos sobre el total de positivos marcados por el modelo [99-102].

Exhaustividad La exhaustividad o *recall* mide que tan bien los objetos son detectados, es decir, el porcentaje de objetos encontrados sobre la cantidad total de objetos presentes en la imagen. Se calcula como:

$$R = \frac{TP}{TP + FN} \quad (9)$$

donde TP son los verdaderos positivos, FN los falsos negativos y R la proporción de verdaderos positivos sobre el total de objetos contenidos en la imagen [99-102]..

AP y mAP El inconveniente de solo utilizar la precisión y exhaustividad de las detecciones para caracterizar el desempeño del modelo es que ambos dependen del umbral utilizado en la detección (el nivel o porcentaje de confianza que debe tener una detección para ser considerada como positivo). Un umbral alto provoca que hayan pocos falsos positivos (alta precisión), pero muchos falsos negativos (baja exhaustividad) y un umbral bajo provoca lo contrario. La precisión media o *average precision* (AP) permite medir la exactitud de las detecciones en términos de ambos, precisión y exhaustividad. La AP se calcula como la precisión promedio en 11 puntos diferentes de la curva de precisión-exhaustividad (que muestra la precisión obtenida en cada nivel de exhaustividad):

$$AP = \frac{1}{11} \sum_{r \in \{0,0,\dots,1\}} P_r \quad (10)$$

Si el AP es calculado usando un umbral de IoU del 50 %, una detección positiva corresponde a una en la cual la caja predicha coincide en al menos un 50 % de su área con la caja real del objeto. Dado que el AP_{50} se calcula para cada clase de objeto (células vivas, muertas), el mAP_{50} (precisión media promedio o *mean average precision*) toma el promedio de todas las precisiones medias ofreciendo una métrica global útil para caracterizar el modelo entrenado [100, 103].

Valor F_1 El valor F_1 es otra medida de la exactitud global del modelo, que combina ambos, precisión y exhaustividad para un determinado umbral de confianza. Se calcula como la media armónica de P y R :

$$F_1 = 2 \times \frac{P \times R}{P + R} \quad (11)$$

Evaluación de los modelos El desempeño global de cada modelo se determino teniendo en cuenta la Precisión Media Promedio (mAP), Precisión (P), Exhaustividad (R), el Valor F_1

e IoU promedio. Por otra parte, para medir el desempeño desglosado por clase se midieron la Precisión Media (AP) y Precisión por clase (L: viva; D: muerta; C: cumulo). Con el propósito de analizar el desempeño, ya sea en cuanto a la localización de los objetos (predicción de las coordenadas de la caja contenedora) como a la clasificación de los objetos, las métricas propuestas se obtuvieron usando umbrales del 50 % y 30 % de IoU. Disminuir el umbral del IoU permite tener en cuenta para el análisis casos en los cuales, si bien la la clasificación es correcta, la caja predicha se encuentra desplazada respecto de la verdadera, priorizando así la exactitud en la clasificación de los objetos por sobre su localización. En otras palabras, utilizar un umbral más bajo da una medida de que tan bien el modelo “encuentra y clasifica” los objetos, mientras que un umbral mayor tiene en cuenta, en mayor medida, que tan bien el modelo “dibuja la caja” alrededor del objeto. Cabe aclarar, que en ambos casos las predicciones realizadas por el modelo estudiado son las mismas; lo que varía en cada caso es el criterio que se utiliza para considerar si una detección es un verdadero positivo o un falso positivo.

Como ya se mencionó al momento de realizar las predicciones YOLO utiliza el algoritmo de *Non Maximal Supression* con el propósito de filtrar aquellas predicciones con una baja confianza. Para poder ser equitativo en la comparación de los modelos se utilizó un umbral de confianza de 25 % en todos los casos, tal como sucede en [33].

2.5. Software

Con el motivo de poder utilizar el dispositivo para realizar conteos, se construyó una plataforma desarrollada en el lenguaje de programación *Python 3.7* que permitiese integrar el funcionamiento de la cámara digital y de la red neuronal junto a una interfaz gráfica intuitiva y fácil de usar. La aplicación diseñada permite:

- Crear conteos nuevos junto a campos de información básica (nombre, fecha, proyecto, línea celular y número de pasaje)
- Acceder a la cámara, visualizar la muestra en tiempo real, tomar una foto y realizar un conteo de las células presentes en esa foto
- Visualizar los resultados del conteo realizado (imagen anotada con las detecciones, cantidad de células contadas totales y por clase, concentraciones celulares total y por clase,

viabilidad del subcultivo, media y distribución de los tamaños celulares totales y por clase)

- Guardar los resultados en la memoria del dispositivo
- Exportar los resultados en formatos *.pdf* y/o *.jpg*
- Acceder a la memoria, visualizar resultados y editar información de conteos previos

La interfaz gráfica se desarrollo utilizando *Tkinter*, una interfaz de programación de aplicaciones (API) que permite la creación de una interfaz gráfica de usuario (GUI) en *Python* utilizando la biblioteca de elementos *Tk*.

3. Resultados

3.1. Características técnicas del prototipo diseñado

En este trabajo se diseñó y construyó un contador de células automático de bajo costo y portátil, apto para soluciones celulares teñidas con azul de tripán (Figura 3.1). El equipo desarrollado es compatible con hemocitómetros reutilizables comerciales presentes en la mayoría de los laboratorios de cultivo celular gracias a su uso en la realización de conteos manuales. El algoritmo de detección del contador fue entrenado para permitir la detección de células vivas y muertas con tamaños en el rango de $7\text{ }\mu\text{m}$ - $60\text{ }\mu\text{m}$ y concentraciones celulares entre 1×10^4 - 3×10^6 células/ml. Como se mencionó en la sección 2.2, el modelo fue entrenado con imágenes de células pertenecientes a 25 líneas celulares animales diferentes.

El tamaño total del dispositivo logrado es de aproximadamente $10.5\text{ cm} \times 12.5\text{ cm} \times 16\text{ cm}$ y su peso $<0.4\text{ kg}$ lo que permite que sea posicionado sobre una mesada y que pueda ser trasladado sin complicaciones. Para su funcionamiento requiere de un monitor conectado mediante un cable HDMI, un teclado y un *mouse* conectados mediante USB. Por otra parte, el dispositivo puede ser utilizado sobre una laptop si se realizan las configuraciones necesarias y se utiliza un cable Ethernet.

Acorde a lo detallado en la sección 2.1.3, todos los componentes electrónicos que integran al dispositivo fueron pensados para ser compatibles con la tensión que ofrece la placa en sus pines.

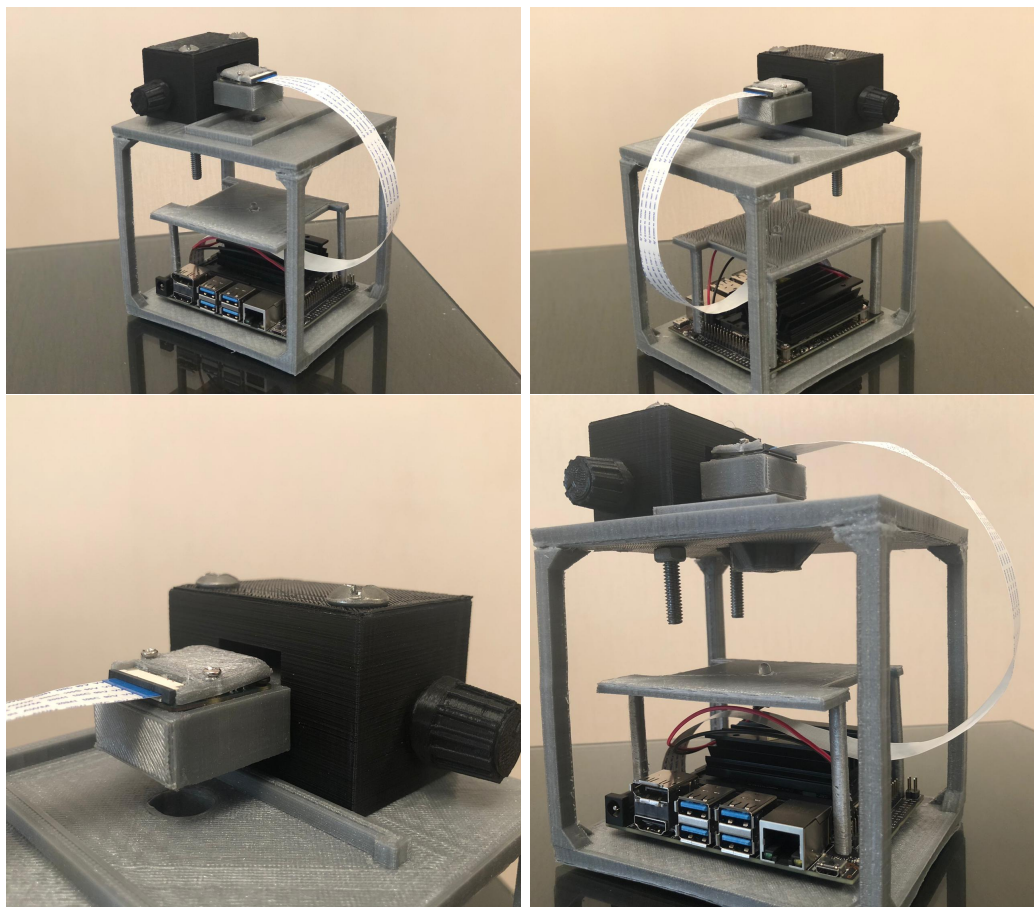


Fig. 3.1: Dispositivo Diseñado. En estas imágenes se muestra el dispositivo desde distintos ángulos.

Esto significa que solo se tomaron en cuenta los requerimientos de la placa para su correcto funcionamiento. Dado esto, se decidió utilizar una fuente *switch* que permite alimentar a la placa con una tensión de 5V y 4A.

Por último, si a los componentes mencionados le agregamos el costo de servicio de impresión 3D, el costo total del dispositivo no supera los US\$200. Un servicio de impresión para un dispositivo de las medidas mencionadas tiene un costo aproximado de US\$20 pero se encuentra atado a los requerimientos de detalle y material del usuario.

Magnificación En cualquier sistema óptico la magnificación de la imagen se calcula y expresa como la relación entre las dimensiones del objeto y la imagen o $m = \frac{h_I}{h_O}$ donde h_I y h_O son los tamaños en una dimensión (horizontal, vertical o diagonal) de la imagen formada y del objeto respectivamente [87]. Por otra parte, si se toma las ecuaciones 5 y 6, se llega a que:

$$m = \frac{f}{u - f} = \frac{v - f}{f} \quad (12)$$

[87-89], y que

$$v = f \times (m + 1) \quad (13)$$

[87, 88]¹⁷

Usando estas dos ecuaciones, se varió el largo del “espaciador” para obtener la magnificación (m) deseada. Con una distancia lente-sensor de ~ 8.65 mm, la magnificación óptica obtenida fue de aproximadamente $\times 1,85$. La magnificación resultante fue estimada comparando las dimensiones del sensor declaradas por el fabricante (tabla 2) y el campo de visión del sistema final.

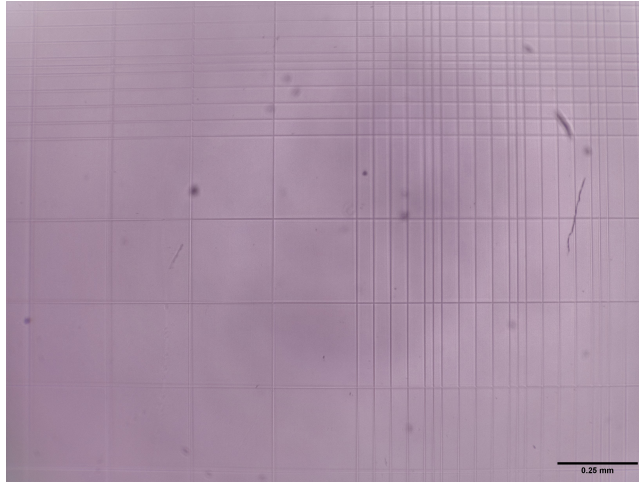


Fig. 3.2: Campo de visión logrado. En la imagen se pueden observar, aproximadamente, 8 cuadrados de 0.25 mm a lo ancho y 6 a lo largo, lo que equivale a un FOV de ~ 2 mm $\times$ 1.5 mm.

Campo de Visión En fotografía, el campo de visión (FOV) es el área observable del objeto bajo inspección. De manera simple, sí se conocen las dimensiones del sensor y la magnificación

¹⁷Este parámetro es también conocido como magnificación lineal o lateral, consiste en la magnificación de la imagen en una dimensión y que es diferente a la magnificación del área del objeto, que es simplemente el cuadrado de la magnificación lineal.

del sistema, se puede calcular como:

$$FOV = \frac{\text{Tamaño del sensor}}{m} \quad (14)$$

Dado que m corresponde a la magnificación lineal, se debe usar como tamaño del sensor el largo, ancho o diagonal del sensor, y se obtendrá entonces, el FOV horizontal, vertical o diagonal. Para obtener la dimensión restante se puede también multiplicar el FOV obtenido por la relación entre las dimensiones del sensor (relación de aspecto) [83]. Cuando se trata de contadores de células, el FOV de la imagen es equivalente al área de conteo que, consecuentemente, define el volumen de análisis del contador. El FOV en área se puede calcular multiplicando los campos de visión verticales y horizontales o bien haciendo $FOV = \frac{\text{Area del sensor}}{m^2}$. Teniendo en cuenta esto, es evidente que mayores distancias entre la lente y el sensor (v) significan mayores magnificaciones (m) que resultan en un menor FOV.

Tal como se lo diseñó y construyó, el dispositivo permite obtener imágenes de un FOV de 3 mm^2 . Dada la magnificación del sistema ($m = 1,85$) y el tamaño del sensor ($3.68 \text{ mm} \times 2.76 \text{ mm}$) el FOV resultante es de $2 \text{ mm} \times 1.5 \text{ mm}$. Ya que una cámara de Neubauer común posee líneas espaciadas por 1 mm , 0.25 mm , 0.05 mm y 0.025 mm , contar la cantidad de líneas y cuadrantes visualizados en una imagen permite estimar el FOV del sistema.

Iluminación Como se mencionó anteriormente, la iluminación de la muestra se llevó a cabo mediante un LED blanco y un lente condensador posicionado por debajo de la muestra. Este tipo de iluminación, llamada *Iluminación Crítica* o *Abbe*, consiste en enfocar el haz de luz sobre la muestra a observar, generando una imagen de la fuente de luz en el objeto. Para esto, es aconsejable que la fuente sea lo más uniforme posible, de modo de evitar variaciones de intensidad en la imagen observada [88, 89, 105, 106].

Una correcta iluminación de la muestra es indispensable para obtener mejores resoluciones ópticas del sistema, y para esto, son necesarias una fuente de iluminación y un sistema de lentes apropiado. En este sentido, la lente condensadora utilizada en este dispositivo, permite iluminar el campo de visión entero de una manera homogénea y permite una mayor resolución óptica.

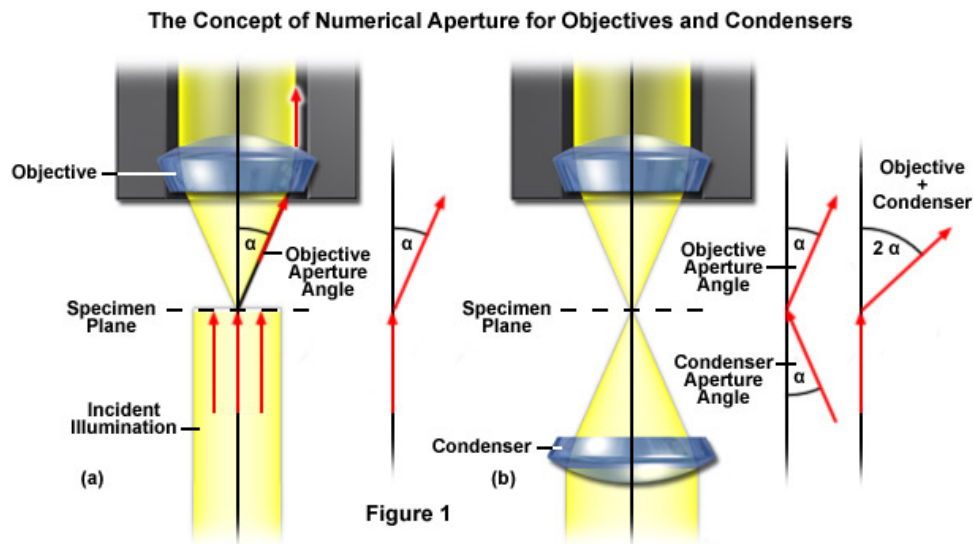


Fig. 3.3: Efecto del uso de una lente condensadora en la Apertura Numérica del sistema. Mayores valores de Apertura Numérica (NA) permiten que una cantidad más grande de rayos de luz entren en la lente objetivo, lo que produce imágenes con mejores resoluciones. La figura (a) muestra el caso en el cual un sistema básico compuesto por una lente objetivo es iluminado una fuente de luz colimada. En ese caso los límites de luz que ingresan por la lente están dados por el ángulo α . La figura (b) es el caso en el cual una lente condensadora es añadida al sistema. La misma permite recoger rayos de luz con ángulos mas grandes de incidencia del lado de la muestra. La suma de ambas aperturas numéricas corresponden a la Apertura Numérica Efectiva o de trabajo. Imagen obtenida de Rottenfusser, R., Wilson, E. E. & Davidson, M. W. (2014). Education in Microscopy and Digital Imaging. *ZEISS Microscopy*

Resolución La resolución de un sistema óptico se define como la menor distancia entre dos puntos diferentes que pueden todavía ser distinguidos como dos objetos separados. En un sistema digital, sin embargo, la resolución del complejo total comprende dos partes diferentes, la Resolución Óptica (OR) y la Resolución del Sensor (SR), de modo que la mayor de ellas será la que limite la resolución del sistema [104, 107].

La Resolución Óptica viene determinada por la lente del equipo (o sistema de lentes), y por su capacidad de la misma de “recolectar” luz con una muestra situada a una determinada distancia. La Apertura Numérica (NA) describe el rango de ángulos de haces incidentes que la lente acepta. La misma viene definida por:

$$NA = n \times \sin \alpha \quad (15)$$

donde n es el índice de refracción del medio entre la lente y la muestra (en aire $n = 1$), y α

es la mitad del ángulo máximo que puede entrar en la lente (con el objeto a una distancia determinada). Mediante cálculos de trigonometría se puede llegar a que $\alpha = \arctan \frac{D}{2 \times u}$, donde D es el diámetro de la lente (en la entrada) y u es la distancia de la lente al objeto [108]. Finalmente, el límite de resolución óptica (Criterio de *Rayleigh*) es:

$$OR = 1,22 \times \frac{\lambda}{NA_{obj} + NA_{cond}} \quad (16)$$

donde λ es la longitud de onda de la luz que ilumina (para luz blanca se suele usar $\lambda=550$ nm que corresponde al centro del espectro visible). Como es fácil de notar, un mayor valor de NA permite obtener mejores resoluciones. Es importante notar que para el cálculo de la resolución óptica del sistema se tienen en consideración las aperturas numéricas tanto de la lente objetivo (NA_{obj}) como del condensador (NA_{cond}) significando que la capacidad del sistema de resolver dos objetos a una distancia pequeña depende también del modo de iluminación de la muestra (Figura 3.3). En la práctica, la máxima resolución se obtiene cuando la apertura numérica del condensador es igual o mayor que la del objetivo, no variando si $NA_{cond} > NA_{obj}$ (en ese caso $OR = 1,22 \times \frac{\lambda}{2 \times NA_{obj}}$) [83-85, 104, 106, 108, 109].

Por otra parte, la resolución del sistema viene limitada por la capacidad del sensor de distinguir objetos cercanos. La resolución del sensor, es equivalente al detalle más pequeño que puede detectar un píxel del sensor. Cuando la imagen se magnifica, cada píxel puede detectar detalles aún más pequeños. Dado que en general se requiere que el tamaño del píxel sea como mínimo el doble del tamaño del objeto detectado¹⁸, el límite de resolución de un sensor viene dado por:

$$SR = 2 \times \frac{\text{Tamaño del pixel}}{m} \quad (17)$$

En un sistema como el presentado, entonces, el límite de resolución viene dado por la peor (mayor) de las resoluciones. En el dispositivo de este trabajo, el diámetro de la lente objetivo usada es de 1.52 mm lo que produce un NA de ~ 0.16 , mientras que la lente condensadora tiene un diámetro de ~ 12 mm y una distancia focal de ~ 15 mm lo que se traduce en un NA de ~ 0.37 . Esto significa que el límite óptico de la resolución es de $2.1 \mu\text{m}$ (con $\lambda=550$ nm).

¹⁸Según el Teorema de Nyquist

Siendo que la magnificación es de ~ 1.85 , y el tamaño de un píxel de $1.12\mu\text{m}$ (ver tabla 2), el límite del sensor se encuentra por debajo del óptico ($\approx 607\text{ nm}$). Se estimó entonces que la resolución final sería de aproximadamente $2.1\mu\text{m}$, aunque una mejor caracterización podría ser realizada utilizando un tablero de prueba de resoluciones.

3.2. Dataset

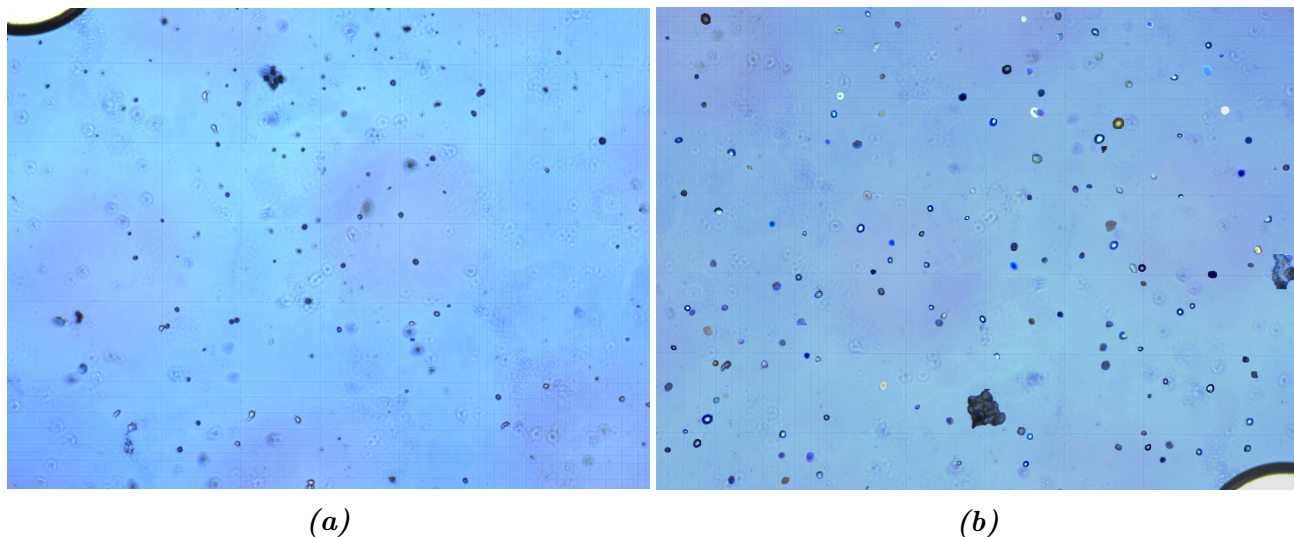


Fig. 3.4: Comparación de una imagen real y una sintética. En (a) se muestra una imagen real obtenida en el laboratorio cuyas células y fondo fueron utilizados para generar múltiples imágenes sintéticas como la que se muestra en (b).

Mediante el procedimiento detallado en la sección 2.2.1 fue posible generar 12 *datasets* de imágenes diferentes usados para el entrenamiento de diversos modelos neuronales. La figura 3.4 muestra una imagen real obtenida en el laboratorio junto a una imagen sintética utilizada en el entrenamiento de la red. Mediante el método propuesto, es posible generar miles de imágenes junto a sus anotaciones en pocos minutos randomizando la cantidad, distribución y aspecto de los objetos, así como el aspecto de los fondos de las imágenes.

3.3. Modelos Entrenados

La tabla 6 muestra los desempeños de cada modelo (mAP e IoU) junto con las características de cada uno. Las tablas 7, 8 y 9 muestran todas las métricas de cada modelo para los conjuntos de validación y test y con umbrales IoU de 30 % y 50 %.

Tabla 6: Características y resultados de los modelos entrenados.

YOLO	DA	FT	OA	Mix	Cl	L/D	P/N	mAP ₅₀	IoU
V3-608	✓				2	75/75	1000/200	54,20 %	56,74 %
V4-608-3C	✓				3	75/75	1000/0	68,31 %	62,56 %
V4-608-3	✓				2	75/75	1000/200	67,45 %	64,71 %
V4-608-4	✓				2	150/150	1000/200	64,41 %	61,38 %
V4-608-5	✓				2	100/60	1000/200	66,12 %	64,22 %
V4-608-6		✓	✓	✓	2	100/100	1000/1000	83,50 %	56,15 %
V4-608-7		✓	✓	✓	2	100/100	4000/4000	80,15 %	58,03 %
V4-608-8		✓	✓	✓	2	100/100	1000/200	59,04 %	36,28 %
V4-608-9		✓			2	100/100	1000/200	65,02 %	64,43 %
V4-608-10	✓	✓			2	100/100	1000/200	69,70 %	66,52 %
V4-608-11	✓	✓			2	100/100	1000/1000	73,91 %	68,27 %
V4-608-12	✓	✓		✓	2	100/100	1000/1000	87,30 %	67,63 %

DA: Aumento de datos en la generación de imágenes sintéticas

FT: Ajuste de las anotaciones de imágenes de entrenamiento

OA: *Anchor boxes* optimizados

Mix: Dataset mixto

Cl: Cantidad de clases

L/D: Numero máximo de células por imagen por clase (L=vivas, D=muertas)

P/N: Numero de imágenes con objetos (P) e imágenes negativas (N)

Al realizar predicciones en imágenes del conjunto de validación, todos los modelos obtuvieron un mAP superior al 95 %, lo que demuestra la habilidad de YOLO para localizar y clasificar células en imágenes sintéticas. Asimismo, todos los modelos obtuvieron altos niveles de Precisión y Exhaustividad lo que supone bajas tasas de falsos positivos y negativos en detecciones sobre imágenes sintéticas. La excepción es el caso de YOLOv4-608-8 que obtuvo peores resultados de mAP, R y P en imágenes reales y sintéticas. En cuanto al desempeño por clase, se notan leves diferencias entre las clases *viva* y *muerta*, con peores AP y P en todos los modelos para la clase *muerta*. El modelo de tres clases, por otra parte, mostró un desempeño notablemente peor para la clase *Cumulo*. Finalmente, en todos los casos se obtuvieron IoU promedios mayores al 70 %, lo que demuestra la capacidad de YOLO de aprender a localizar objetos pequeños a partir de imágenes sintéticas.

Al realizar predicciones en imágenes reales, se observa un marcado empeoramiento en todas las métricas. Sin embargo, al disminuir el umbral del IoU usado de 50 % (tabla 8) a 30 % (tabla

Tabla 7: Evaluación de modelos con el conjunto de validación.

	mAP ₅₀	P	R	F ₁	IoU	AP _L	AP _D	AP _C	P _L	P _D	P _C
YOLOv3-608	96,72 %	93,16 %	96,25 %	94,68 %	71,48 %	97,58 %	95,86 %	-	92,79 %	93,51 %	-
YOLOv4-608-3C	95,49 %	95,19 %	97,82 %	96,49 %	76,44 %	99,02 %	98,15 %	89,31 %	95,82 %	95,43 %	80,79 %
YOLOv4-608-3	98,07 %	93,64 %	97,58 %	95,57 %	71,95 %	98,63 %	97,51 %	-	92,79 %	94,44 %	-
YOLOv4-608-4	98,33 %	93,74 %	97,90 %	95,78 %	74,17 %	99,14 %	97,51 %	-	93,02 %	94,46 %	-
YOLOv4-608-5	99,27 %	96,44 %	99,05 %	97,73 %	76,26 %	99,54 %	99,00 %	-	98,45 %	94,69 %	-
YOLOv4-608-6	99,15 %	91,80 %	99,17 %	95,34 %	78,45 %	99,35 %	98,95 %	-	93,64 %	90,21 %	-
YOLOv4-608-7	99,45 %	92,90 %	99,03 %	95,86 %	79,41 %	99,48 %	99,42 %	-	94,96 %	91,11 %	-
YOLOv4-608-8	91,97 %	29,86 %	97,37 %	45,70 %	23,59 %	87,55 %	96,40 %	-	27,66 %	32,12 %	-
YOLOv4-608-9	99,80 %	97,09 %	99,12 %	98,09 %	82,77 %	99,87 %	99,73 %	-	99,20 %	95,25 %	-
YOLOv4-608-10	99,76 %	96,35 %	99,22 %	97,76 %	82,03 %	99,86 %	99,66 %	-	98,34 %	94,61 %	-
YOLOv4-608-11	99,79 %	96,22 %	99,30 %	97,74 %	81,19 %	99,89 %	99,68 %	-	98,64 %	94,13 %	-
YOLOv4-608-12	99,77 %	95,74 %	98,87 %	97,28 %	81,88 %	99,82 %	99,71 %	-	99,44 %	92,66 %	-

Todos los valores de la Precisión Media Promedio (mAP), Precisión (P) y Exhaustividad (R) fueron obtenidos usando un umbral de IoU del 50 %. Las Precisiones y Precisiones Media por clase se diferencian según su subíndice; Viva (L), Muerta (D), Cumulo (C).

9) las métricas tienden a mejorar. Esto permite suponer que, en imágenes reales, los modelos tienen mayores dificultades para conseguir una buena alineación de las cajas predichas con las reales. Esto también se ve reflejado en los bajos IoU promedios (alrededor del 65 %).

Visto que a los modelos les cuesta localizar perfectamente las células, analizar los resultados de la tabla 9, significa priorizar la clasificación de los objetos que el modelo puede encontrar. Teniendo en cuenta esto, se observa que tanto el modelo basado en YOLOv3 como aquel entrenado para detectar 3 clases poseen peores desempeños. El resto de los modelos, presentan métricas comparables en cuanto al mAP y valor F₁. Esto permite suponer que, globalmente, su desempeño es similar. Por otra parte, se repite la tendencia observada en la tabla 7 en cuanto a la peor capacidad de detección de los modelos de células muertas respecto de las vivas. Los bajos niveles de precisión en células muertas (P_D) indican una elevada proporción de falsos positivos en detecciones de células muertas. Asimismo, los bajos niveles de precisión media en células muertas (AP_D) indican que todos los modelos son menos robustos a la hora de detectar y clasificar células muertas. Los efectos de las diferentes variaciones en los *datasets* y modelos se analizan con mayor profundidad en la sección 4.3.

Visualización de predicciones realizadas Con el motivo de identificar los errores puntuales del modelo, determinar las causas de dichos errores e hipotetizar acerca de posibles mejoras, también se observaron las predicciones realizadas por los diferentes modelos en imágenes indi-

Tabla 8: Evaluación de modelos con el conjunto de pruebas.

	mAP₅₀	P	R	F₁	IoU	AP_L	AP_D	AP_C	P_L	P_D	P_C
YOLOv3-608	54,20 %	77,63 %	72,44 %	74,95 %	56,74 %	86,87 %	21,54 %	-	87,57 %	40,69 %	-
YOLOv4-608-3C	68,31 %	83,21 %	77,13 %	80,06 %	62,56 %	86,21 %	35,79 %	82,94 %	88,35 %	59,18 %	33,33 %
YOLOv4-608-3	67,45 %	88,36 %	78,72 %	83,26 %	64,71 %	90,81 %	44,10 %	-	91,71 %	69,39 %	-
YOLOv4-608-4	64,41 %	85,02 %	72,03 %	77,99 %	61,38 %	89,78 %	39,05 %	-	91,60 %	54,13 %	-
YOLOv4-608-5	66,12 %	88,24 %	78,85 %	83,29 %	64,22 %	91,51 %	40,74 %	-	93,65 %	59,62 %	-
YOLOv4-608-6	83,50 %	74,67 %	91,27 %	82,14 %	56,15 %	93,66 %	73,34 %	-	86,38 %	47,79 %	-
YOLOv4-608-7	80,15 %	76,02 %	88,68 %	81,86 %	58,03 %	93,42 %	69,32 %	-	88,08 %	47,01 %	-
YOLOv4-608-8	59,04 %	49,81 %	73,40 %	59,35 %	36,28 %	87,30 %	30,78 %	-	84,90 %	18,42 %	-
YOLOv4-608-9	65,02 %	83,94 %	78,44 %	81,10 %	64,43 %	93,33 %	36,71 %	-	93,67 %	48,65 %	-
YOLOv4-608-10	69,70 %	86,65 %	81,45 %	83,97 %	66,52 %	96,07 %	43,34 %	-	94,61 %	53,03 %	-
YOLOv4-608-11	73,91 %	88,46 %	83,63 %	85,97 %	68,27 %	95,03 %	52,79 %	-	94,82 %	61,65 %	-
YOLOv4-608-12	87,30 %	88,47 %	92,09 %	90,24 %	67,63 %	97,09 %	77,52 %	-	95,65 %	66,49 %	-

Todos los valores de la Precisión Media Promedio (mAP), Precisión (P) y Exhaustividad (R) fueron obtenidos usando un umbral de IoU del 50 %. Las Precisiones y Precisiones Media por clase se diferencian según su subíndice; Viva (L), Muerta (D), Cumulo (C).

viduales de prueba.

3.4. Software

En esta sección se muestran los resultados de la interfaz gráfica en forma de capturas de pantalla junto a explicaciones acerca de las funcionalidades de la aplicación.

Pantalla inicial de la interfaz

Al encender el dispositivo, la pantalla de inicio de la aplicación se ve como en la figura 3.5, e incluye las funcionalidades descritas a continuación:

- “*New Count*” permite iniciar un nuevo conteo.
- “*Previous Counts*” permite visualizar los conteos previos realizados, acceder a sus reportes, editarlos y eliminarlos de la base de datos.
- “*Manage Cell Lines*” permite editar la lista de líneas celulares comúnmente usadas agregando o quitando elementos de la lista.
- “*Home*” permite regresar a la pantalla de inicio.
- “*Quit*” permite cerrar la aplicación.

Tabla 9: Evaluación de modelos con el conjunto de pruebas.

	mAP₃₀	P	R	F₁	IoU	AP_L	AP_D	AP_C	P_L	P_D	P_C
YOLOv3-608	74,50 %	84,55 %	79,13 %	81,75 %	58,81 %	92,78 %	56,22 %	-	90,17 %	63,95 %	-
YOLOv4-608-3C	77,98 %	89,34 %	82,81 %	85,96 %	65,22 %	93,12 %	57,87 %	82,94 %	92,87 %	75,51 %	33,33 %
YOLOv4-608-3	80,53 %	91,88 %	81,86 %	86,58 %	65,65 %	95,36 %	65,69 %	-	93,15 %	84,69 %	-
YOLOv4-608-4	81,33 %	91,63 %	77,63 %	84,05 %	64,08 %	95,86 %	66,80 %	-	94,73 %	77,06 %	-
YOLOv4-608-5	78,21 %	92,37 %	82,54 %	87,18 %	65,63 %	96,26 %	60,16 %	-	95,83 %	74,04 %	-
YOLOv4-608-6	87,10 %	75,56 %	92,36 %	83,12 %	55,47 %	96,22 %	77,98 %	-	87,02 %	49,26 %	-
YOLOv4-608-7	82,91 %	76,02 %	91,27 %	82,95 %	56,41 %	94,68 %	71,13 %	-	87,40 %	47,38 %	-
YOLOv4-608-8	63,66 %	51,30 %	75,58 %	61,11 %	36,21 %	92,57 %	34,75 %	-	86,47 %	19,82 %	-
YOLOv4-608-9	68,38 %	84,96 %	64,76 %	82,09 %	79,40 %	82,09 %	64,76 %	-	94,99 %	41,77 %	-
YOLOv4-608-10	72,31 %	86,65 %	81,45 %	83,97 %	66,11 %	97,52 %	47,10 %	-	94,08 %	55,30 %	-
YOLOv4-608-11	77,05 %	88,17 %	83,36 %	85,69 %	67,69 %	96,87 %	57,22 %	-	94,11 %	63,16 %	-
YOLOv4-608-12	89,51 %	89,38 %	93,04 %	91,18 %	68,00 %	98,52 %	80,51 %	-	96,52 %	67,55 %	-

Todos los valores de la Precisión Media Promedio (mAP), Precisión (P) y Exhaustividad (R) fueron obtenidos usando un umbral de IoU del 30%. Las Precisiones y Precisiones Media por clase se diferencian según su subíndice; Viva (L), Muerta (D), Cumulo (C).

Inicio de un nuevo conteo

Al apretar el botón “*New Count*”, una nueva ventana aparece donde el usuario puede introducir el nombre del conteo, del proyecto y del operador, la línea celular y el número de pasaje, siendo el único campo obligatorio (y no modificable) la fecha y hora del conteo (Figura 3.6). Para agilizar el llenado de los datos, los campos del nombre del operador y línea celular poseen una lista desplegable con nombres de operadores anteriores (es decir, operadores que ya hayan realizado conteos) y con nombres de líneas celulares comunes a partir de los cuales el usuario puede elegir. De no ser completados, los campos permanecerán vacíos a excepción del nombre del conteo que sera rellenado, por defecto, con el nombre de la línea celular y la fecha separados por un guión. Cualquier información introducida en esta primera instancia puede ser modificada al terminar el conteo, por lo que no es necesario que el operador cargue toda la información antes de comenzar la tarea.

Una vez realizado esto, el operador puede presionar el botón “*Start Count*” para guardar los datos y comenzar. En esta instancia, la cámara se enciende y el usuario podrá ver, en tiempo real, la imagen capturada por la cámara (Figura 3.7 izquierda). Al posicionar el hemocitómetro sobre la superficie plana del dispositivo y dentro de los límites, el usuario podrá ver el área donde se llevará a cabo el conteo. Durante este tiempo, tiene la posibilidad de desplazar el hemocitómetro para elegir el área de conteo que prefiera y puede accionar manualmente las

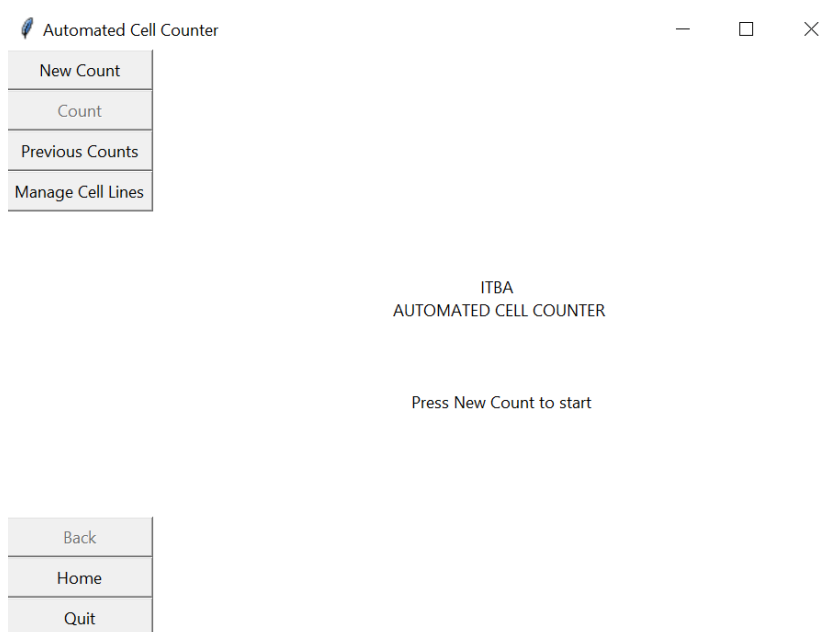


Fig. 3.5: Pantalla de inicio de la interfaz.

Fig. 3.6: Ventana para el ingreso de información de un nuevo conteo.

perillas de enfoque para enfocar la imagen sobre la muestra de modo de poder observar correctamente las células presentes. Durante este paso, el usuario debe procurar que la muestra de células en solución haya sido preparada y colocada en el hemocitómetro correctamente, evitando la presencia de cúmulos celulares, residuos, burbujas de aire y otros artefactos que puedan afectar la calidad del conteo y la precisión del modelo.

Una vez conforme con el área de conteo y calidad de la imagen el operador puede presionar el botón “*Count*” para tomar la foto y ejecutar la red neuronal. En este momento, y hasta que la red neuronal termine su ejecución, todos los botones de la interfaz se encontraran deshabilitados (Figura 3.7 derecha).

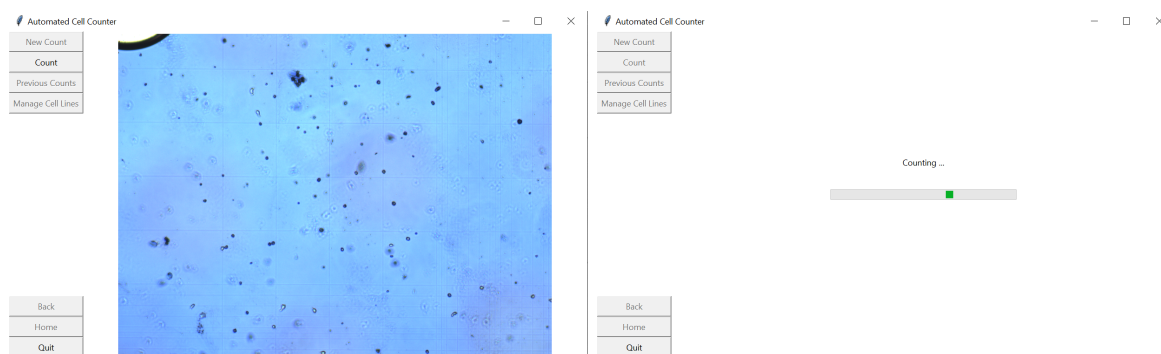


Fig. 3.7: Proceso de conteo. Vista preliminar de la imagen a analizar (izquierda) y pantalla de espera durante la ejecución de la red neuronal (derecha).

Visualización de resultados

Finalizado el conteo, los resultados obtenidos serán automáticamente presentados en la pantalla

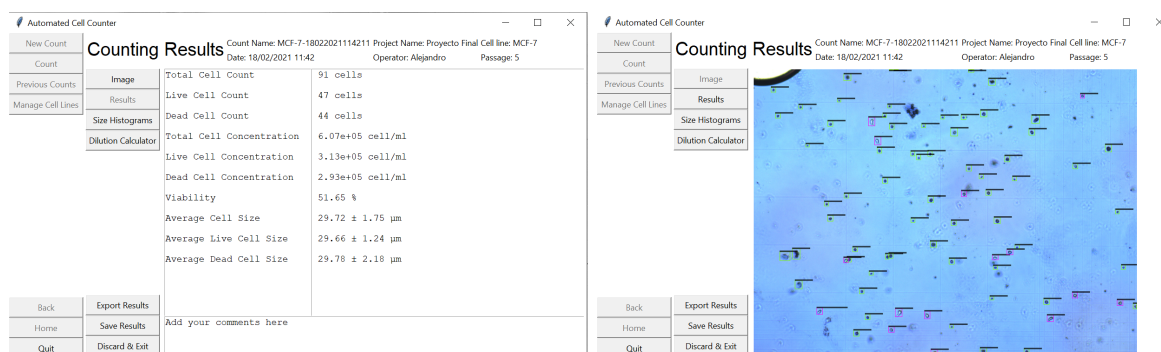


Fig. 3.8: Visualización de resultados del conteo. En forma de lista (izquierda), en forma de imagen anotada (derecha).

de la interfaz. El tiempo de espera desde que el usuario oprime el botón “Count” hasta que es capaz de observar los resultados en pantalla es de, aproximadamente, 30 segundos. Los resultados pueden ser visualizados ya sea a través de la imagen anotada como en forma de lista (figura 3.8). Mientras que la imagen procesada permite corroborar al operador el correcto desempeño del modelo al realizar el conteo, los resultados en formato numérico permiten conocer el estado real del cultivo celular. En cuanto a los datos presentados, los resultados mostrados en la interfaz gráfica, incluyen:

- Cantidad total de células contadas
- Cantidad de células vivas contadas

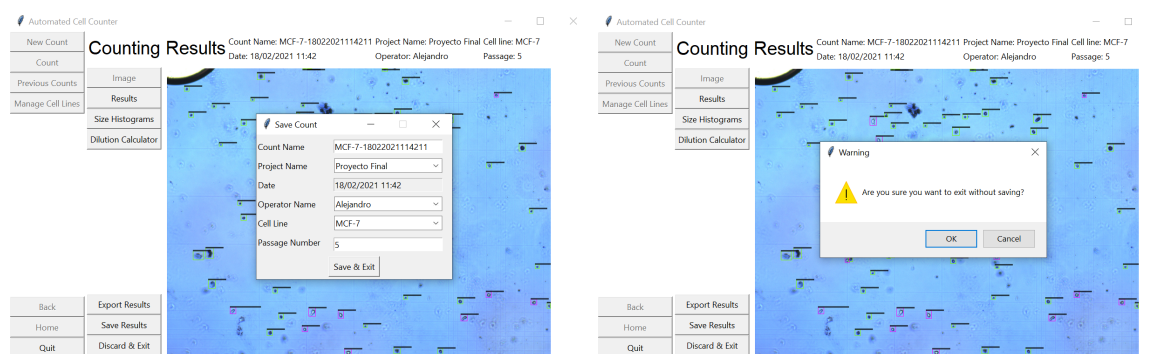


Fig. 3.9: Guardado y Descarte de un conteo.

- Cantidad de células muertas contadas
- Concentración total de células (células/ml)
- Concentración de células vivas (células/ml)
- Concentración de células muertas (células/ml)
- Viabilidad del cultivo (%)
- Tamaño celular medio (μm)
- Tamaño medio de células vivas (μm)
- Tamaño medio de células muertas (μm)

Es importante aclarar que, para realizar los cálculos de concentración celular, el programa asume una dilución 1:1 con azul de tripán 0.4% y una profundidad del hemocitómetro de 0.1 mm, siendo las concentraciones expresadas en los resultados las originales previas a la dilución, es decir, las concentraciones en el frasco de cultivo.

Dentro de la pantalla de resultados, el operador tiene, también, la posibilidad de añadir comentarios en formato de texto libre, que serán guardados junto con el resto de la información y resultados en la base de datos de conteos. Una vez terminado el conteo, el usuario puede exportar los resultados, guardar los resultados o eliminarlos y regresar a la pantalla inicial. Los resultados guardados serán almacenados en una base de datos en la memoria del dispositivo y podrán ser accedidos a través de la ventana de conteos previos. Antes de guardar los resultados, el usuario tendrá la posibilidad de modificar la información básica del conteo, es decir, nombre

del conteo, nombre del operador, nombre del proyecto, línea celular, número de pasaje (Figura 3.9).

Exportación de resultados

Al presionar “*Export Results*”, el usuario podrá elegir si exportar los resultados en forma de

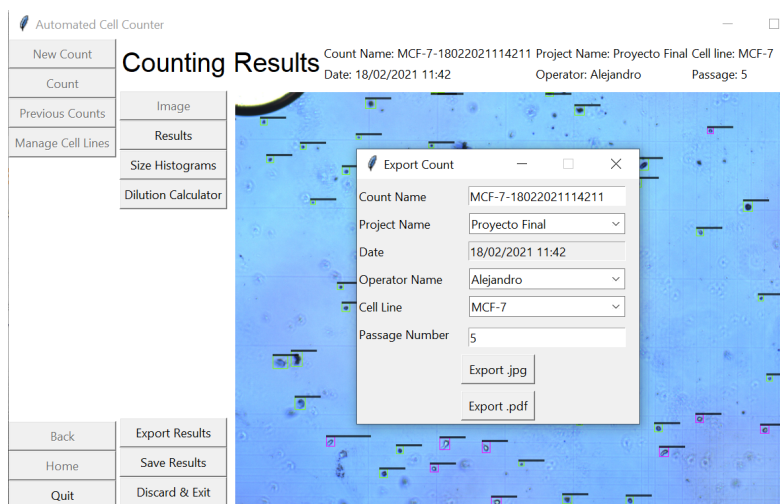


Fig. 3.10: Dialogo para la exportación de un resultado.

una imagen en formato *.jpg* que contenga las células localizadas y clasificadas o en forma de un reporte en formato *.pdf* (Figura 3.11) conteniendo la información del conteo, los resultados, los comentarios, la imagen anotada y los histogramas de tamaño. Antes de exportar los resultados, el usuario tendrá la posibilidad de modificar la información básica del conteo (Figura 3.10).

Histogramas de Tamaño

Al presionar “*Size Histograms*”, el usuario podrá visualizar en dos ventanas independientes los histogramas de tamaño de los resultados. Una de las ventanas contiene un histograma con la información del total de las células, mientras que la otra presenta, en un mismo gráfico, los histogramas de los tamaños de las células vivas (barras verdes) y muertas (barras rojas) de forma superpuesta (Figura 3.12). Por defecto, los límites del histograma en el eje x son 0 μm y 60 μm , mientras que el eje y se ajusta a la cantidad máxima de células para una barra (altura de la barra mas alta). Sin embargo, dado que los gráficos son presentados en ventanas externas, la misma aplicación permite desplazar el gráfico, realizar acercamientos y alejamientos, moverse con el cursor y seleccionar datos y guardar el gráfico, entre otras cosas.

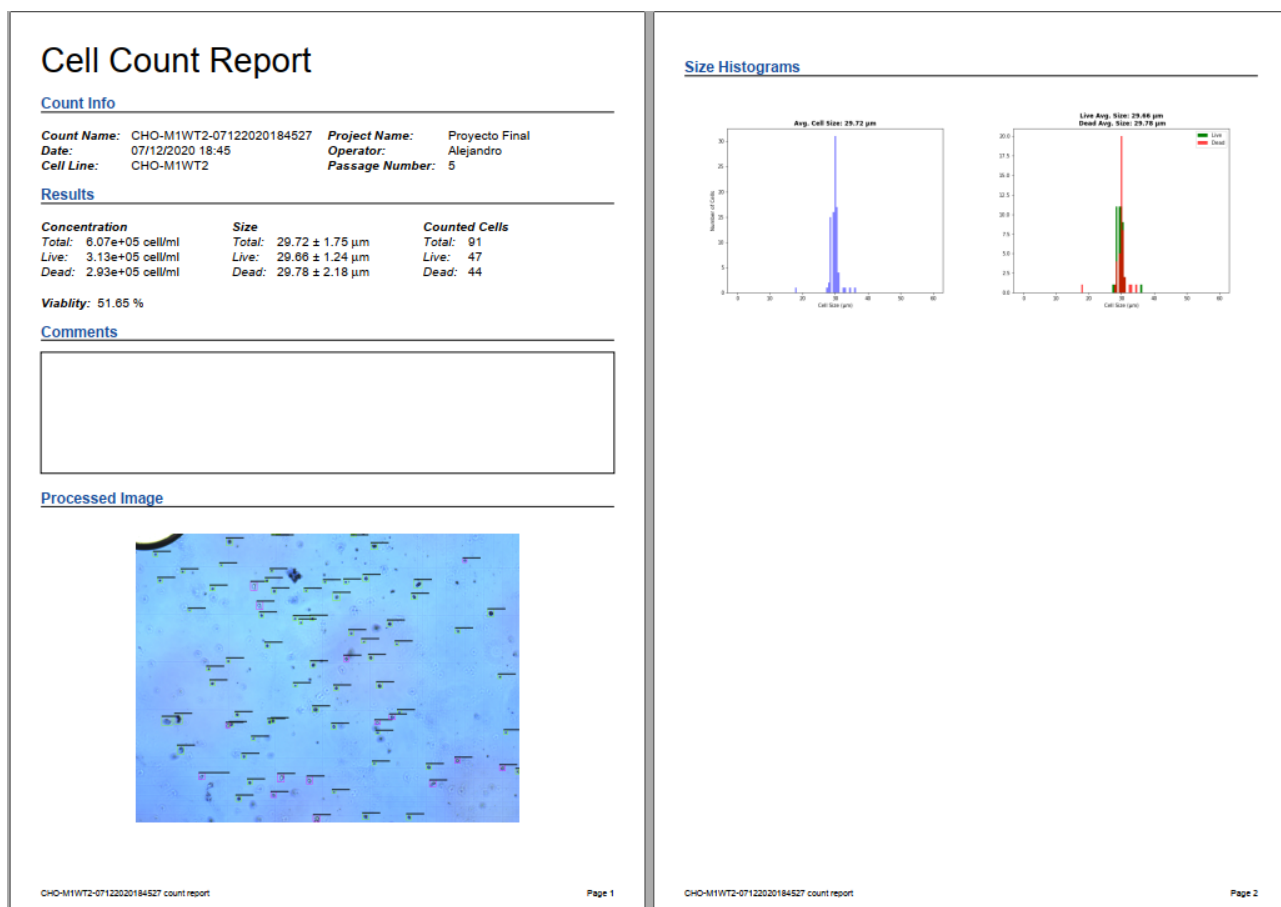


Fig. 3.11: Archivo .pdf generado con el reporte del conteo.

Cálculo de dilución

Al presionar “*Dilution Calculator*”, el usuario podrá acceder a una calculadora incorporada en la interfaz que le permite determinar la cantidad de suspensión celular y la cantidad de medio de cultivo necesarios para llegar a la concentración celular de siembra deseada (Figura 3.13).

Al iniciar la calculadora, se deberán completar tres campos: concentración actual, concentración deseada y volumen deseado. La concentración de células actual se completa en base a los resultados obtenidos del conteo, por lo que no es un campo editable para el operador. Sin embargo, el usuario si puede seleccionar si desea realizar los cálculos considerando la concentración de células vivas o totales. Al cambiar de opción en el menú desplegable, el valor mostrado en la casilla también cambiará. Por defecto la calculadora utiliza la concentración de células vivas. La concentración deseada, es la cantidad de células por mililitro que se espera haya en el nuevo frasco de cultivo. Para completar el campo, hay que seleccionar el multiplicador (orden

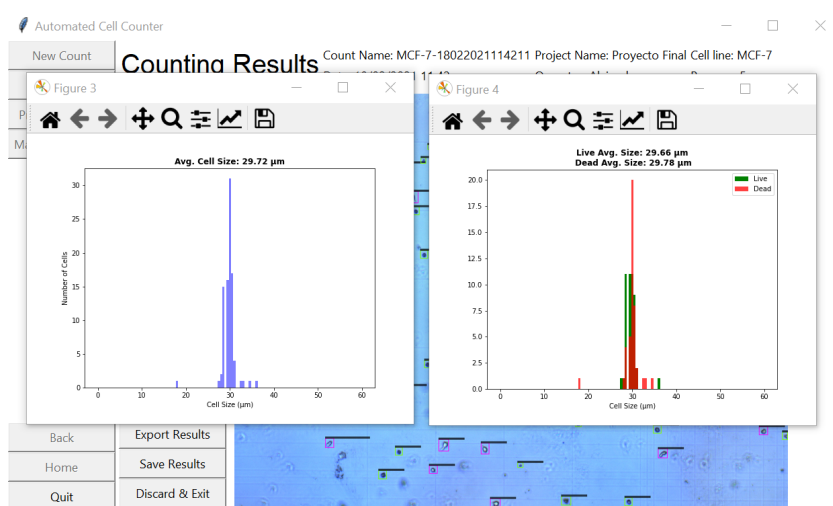


Fig. 3.12: Histogramas de tamaño de las células detectadas.

The screenshot displays the 'Automated Cell Counter' interface, specifically the 'Dilution Calculator' section. The main window is titled 'Counting Results' and shows the 'Dilution Calculator' section. It includes input fields for 'Current Concentration' (3.13e+05 cell/ml), 'Desired Concentration' (5), and 'Desired Volume' (12). A 'Calculate' button is present. The result shows the required volume of cell solution and buffer: 'Mix 0.19 ml of your cell solution with 11.81 ml of buffer.' The interface includes a sidebar with buttons for 'New Count', 'Count', 'Previous Counts', and 'Manage Cell Lines'. The main window also has buttons for 'Back', 'Export Results', 'Home', 'Save Results', 'Quit', and 'Discard & Exit'.

Fig. 3.13: Calculadora de diluciones.

de magnitud) utilizando la lista desplegable a la derecha. Por defecto, el multiplicador es 10^3 . Finalmente, es necesario ingresar el valor del volumen total final deseado (suspensión de células y medio de cultivo) como un número con parte entera y decimal separadas por un punto. Al presionar la tecla “Enter” o el botón “Calculate”, la interfaz mostrará un texto expresando el volumen de solución celular y de medio en ml necesarios. Si el usuario insertase texto no válido en las casillas (caracteres no numéricos o diferentes a un punto), o ingresase una concentración deseada mayor a la actual un mensaje de error se desplegará en una ventana externa.

Acceso a resultados de conteos previos

La aplicación desarrollada también permite un acceso a conteos previos guardados. Dichos

ID	Count Name	Project Name	Date	Operator	Cell Line	Passage Number	Total Cell Concentration	Viability	Avg. Cell Size
9	Capan-2-29112020193418	Proyecto Final	29/11/2020 19:34	Ale	Capan-2	1	6.07e+05	51.65	29.72
8	-29112020182724	Proyecto Final	29/11/2020 18:27	Ale	AsPC-1	1	6.07e+05	51.65	29.72
7	AsPC-1-27112020192202	Proyecto Final	27/11/2020 19:22	Ale	AsPC-1	1	6.07e+05	51.65	29.72
6	bhk 21 (c13)-2711202011271	Proyecto Final	27/11/2020 11:27	Alejandro	bhk 21 (c13)	3	6.07e+05	51.65	29.72
5	A549-27112020112412	Proyecto Final	27/11/2020 11:24	Alejandro	A549	4	6.07e+05	51.65	29.72
4	CCD-1137Sk-271120201122:	Proyecto Final	27/11/2020 11:22	Alejandro	CCD-1137Sk	1	6.07e+05	51.65	29.72
3	-26112020195927	Proyecto Final	26/11/2020 19:59	Ale	RKO	6	6.07e+05	51.65	29.72
2	-25112020173055	Proyecto Final	25/11/2020 17:30	Alejandro	RKO	6	6.07e+05	51.65	29.72
1	A431-25112020155755	Proyecto Final	25/11/2020 15:57	Ale	A431	2	6.07e+05	51.65	29.72

Fig. 3.14: Visualización de conteos previos.

ID	Count Name	Project Name	Date	Operator	Cell Line	Passage Number
9	Capan-2-29112020193418	Proyecto Final	29/11/2020 19:34	Ale	Capan-2	1

Count Name:	Capan-2-29112020193418
Project Name:	Proyecto Final
Date:	29/11/2020 19:34
Operator:	Ale
Cell Line:	Capan-2
Passage Number:	1
Total Cell Concentration:	6.07e+05 cell/ml
Live Cell Concentration:	3.13e+05 cell/ml
Dead Cell Concentration:	2.93e+05 cell/ml
Viability:	51.65 %
Average Cell Size:	29.72 µm
Comments:	Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras feugiat commodo ante, eu maximus sem finibus id. Praesent imperdiet, elit non gravida tincidunt, risus erat hendrerit est, nec viverra justo orci in ligula. Duis consectetur volutpat nunc sed tristique. Fusce feugiat interdum ipsum eu accumsan. Vivamus efficitur viverra maximus. Aliquam lacus diam, bibendum ut tortor vel, cursus lacinia magna. Ut facilisis cursus purus. Morbi augue enim, laoreet quis elit. A, dictum tincidunt nisi. Proin faucibus leo nec nibh pharetra hendrerit. Quisque fermentum et lacus id gravida. Suspendisse eget ultrices elit.

Fig. 3.15: Visualización de reporte de un conteo previo.

conteos son salvados en una base de datos junto a los campos básicos y los resultados obtenidos. Su visualización se puede realizar en forma de tabla (Figura 3.14) o bien individualmente en un cuadro separado (Figura 3.15 izquierda).

Al acceder la tabla de conteos previos, el usuario podrá observar rápidamente la información del conteo (nombre, proyecto, fecha, operador, línea celular y número de pasaje) junto a la concentración total de células, viabilidad y tamaño medio celular de esa muestra. El usuario también puede ordenar los conteos de la tabla en base a cada una de las columnas, facilitando la búsqueda de los resultados de un conteo en particular. Por otra parte, la información básica de cada conteo puede ser siempre modificada, y cualquier conteo puede ser eliminado de la base de datos, siempre y cuando el usuario lo desee.

En caso de necesitarlo, el operador también puede seleccionar un conteo en particular y observar sus resultados completos y comentarios realizados en un cuadro aparte.

Edición de Líneas Celulares

Al presionar “*Manage Cell Lines*” en la pantalla de inicio, se podrá acceder a una ventana y observar la lista de líneas celulares incluidas en la base de datos y utilizadas como opciones al momento de realizar un conteo nuevo. Se cuenta también con la opción de agregar y/o eliminar líneas celulares de la lista.

4. Discusión

4.1. Características del Dispositivo y Comparación con contadores del mercado

Costo

Con un costo total aproximado de US\$180 el dispositivo logrado es entre 20 y 35 veces más barato que los contadores automáticos disponibles en el mercado (tomando como referencia los dispositivos de la tabla 1). El desarrollo presentado en este trabajo supone entonces una alternativa de bajo costo para aquellos investigadores y laboratorios de cultivo celular que no están dispuestos a adquirir contadores automáticos comerciales debido a su elevado precio y que, por esta razón, continúan realizando conteos manuales desperdiciando tiempo e incurriendo en errores. Por otra parte, en años recientes el surgimiento de componentes electrónicos y computacionales de bajo costo junto a la difusión de las técnicas de impresión 3D ha dado lugar a la aparición de desarrollos económicamente accesibles en el campo de la microscopía y otros aparatos de laboratorio permitiendo que estas poderosas herramientas científicas estén a disposición de una gran cantidad de investigadores y estudiantes alrededor del mundo. Siguiendo dicha tendencia, el contador presentado puede ser replicado, construido y/o modificado con facilidad por investigadores interesados utilizando los componentes comerciales mencionado y las piezas impresas en 3D necesarias para la estructura. Cabe aclarar, sin embargo, que el costo del dispositivo se calculó sin tener en cuenta los accesorios externos necesarios para su funcionamiento (monitor, teclado y *mouse* o computadora portátil) dado que se supone que

son elementos que cualquier estudiante o investigador posee. A pesar de esto, dichos elementos podrían ser remplazados por un *display* o pantalla táctil compatible con la placa utilizada, que tiene un costo estimado de entre 15 US\$ y 70 US\$.

Asimismo, el costo del dispositivo esta asociado a los reactivos y otros consumibles necesarios para su funcionamiento. Con esto en mente, es importante destacar que el contador presentado está concebido para poder ser utilizado con celdas de conteo reutilizables incluyendo cámaras de Neubauer y cualquier otro hemocitómetro de 0.1 mm de profundidad. Entonces, a diferencia de la mayoría de los contadores automáticos comerciales, que utilizan celdas de conteo o portaobjetos descartables y específicos para el contador de esa marca, la compatibilidad del dispositivo presentado con cualquier hemocitómetro permite que el investigador acceda a los beneficios del conteo automático sin incurrir en gastos extraordinarios relacionados a la adquisición de celdas de conteo descartables de manera periódica.

Portabilidad

El equipo diseñado tiene un peso total menor a 0.4 kg y ocupa un espacio aproximado de $10.5\text{ cm} \times 12.5\text{ cm} \times 16\text{ cm}$. Comparado con los contadores disponibles en el mercado el dispositivo logrado es hasta 5 veces más pequeño y 10 veces más liviano, teniendo en cuenta que la media de dispositivos de la tabla 1 mide 23 cm en cada dimensión y tiene un peso de 4 kg. Siendo pequeño y liviano, el contador puede ser utilizado en espacios reducidos haciéndolo ideal para laboratorios con salas de cultivo celular pequeñas.

Desempeño y Aplicabilidad

El propósito de este trabajo fue desarrollar un contador de células automático aplicable en estudios de cultivo celular. Además de su bajo costo y portabilidad, el contador desarrollado debe tener la capacidad de detectar células de distintos diámetros. La resolución óptica lograda es de aproximadamente $2.1\text{ }\mu\text{m}$, esto significa que cualquier célula presente en el hemocitómetro será distinguible en la imagen capturada, y podrá ser detectada por el algoritmo de detección. Por otra parte, al utilizar un sensor de 8MP las imágenes de 3280×2464 píxeles obtenidas son de muy alta resolución, algo valioso a la hora de analizar visualmente las muestras. Probablemente, mejores resoluciones (a misma magnificación) puedan ser obtenidas utilizando una lente objetivo de mejor calidad (como por ejemplo un objetivo de microscopio), utilizando mejores fuentes de

iluminación y/o incorporando algoritmos de reconstrucción y mejoramiento de imágenes como los presentado por Dong et al. en [91], por Aidukas et al. en [65] y por Zheng en [110-112]. Es importante tener en cuenta, sin embargo, que utilizar una lente objetivo con un mayor NA (como la puede ser una lente de microscopio) disminuye la profundidad de campo del sistema dificultando el enfoque sobre la muestra. En todo caso, una resolución del orden de los $2.1\ \mu\text{m}$ es suficiente para poder observar y contar células animales en suspensión que generalmente tienen diámetros que varían entre $10\ \mu\text{m}$ y $100\ \mu\text{m}$, y que en promedio tienen diámetros de $10\text{-}20\ \mu\text{m}$ [113]. Asimismo, la magnificación y el FOV obtenido ($2\text{mm}\times 1.5\text{mm}$) es lo suficientemente grande como para que el conteo sea realizado en una muestra de volumen representativo. Los $0.3\ \mu\text{l}$ analizados por el contador presentado son comparables con los volúmenes de análisis de los contadores presentes en el mercado (ver tabla 1), y son comparables con los volúmenes analizados por un investigador al realizar conteos manuales. Esto significa que los errores en el conteo debidos al tamaño de la muestra analizada son del mismo orden que los errores en los que incurren los contadores automáticos comerciales y hasta un investigador experimentado al realizar un conteo manual.

El modelo de red neuronal entrenado e incorporado al dispositivo es, por otro lado, capaz de realizar conteos de células a partir de imágenes capturadas incurriendo en pocos errores. El mismo fue entrenado y testeado con imágenes conteniendo células vivas y muertas de 25 líneas celulares animales diferentes con tamaños que varían entre $7\ \mu\text{m}$ y $60\ \mu\text{m}$ y con concentraciones que van desde 1×10^4 hasta 3×10^6 células/ml lo que hace que el dispositivo sea compatible con el conteo de células pertenecientes a un amplio rango de líneas celulares, de diferentes tamaños, aspectos y a diferentes concentraciones. Las métricas finales del modelo (tabla 10) muestran que al evaluarlo usando 15 imágenes reales la precisión es del 89 % y la exhaustividad es del 92 % lo que significa que el método de conteo presentado es capaz de detectar la gran mayoría de las células presentes cometiendo, al mismo tiempo, pocos errores de clasificación. En adición, los altos valores del mAP del modelo (87,30 % global y 97,09 % y 77,52 % para las células vivas y muertas respectivamente) demuestran que el mismo es robusto y que sus detecciones tienen altos niveles de confianza.

Es difícil establecer un punto de comparación con los contadores automáticos del mercado dado que no es común que la marca propietaria de dicho contador presente un parámetro de

la precisión del mismo o, al menos, ninguna métrica comparable con las presentadas en este trabajo. Particularmente, esto se da porque los contadores vendidos suelen utilizar algoritmos de procesamiento de imágenes para realizar el conteo de células, que vienen evaluados con diferentes métricas a las utilizadas comúnmente en la detección de objetos con inteligencia artificial. Mientras que un contador comercial utiliza parámetros que determinan la variabilidad (medido como un desvío estándar, coeficiente de variación o R^2) de un conteo puntual respecto de una concentración o viabilidad (teórica o esperada), el contador presentado es valorado con métricas que evalúan el desempeño del algoritmo de detección y conteo de una forma más global. Entonces, si bien dicho análisis fue realizado utilizando 15 imágenes reales, se entiende que estas son un parámetro de las imágenes que pueden ser capturadas por el dispositivo en funcionamiento y que, por ende, las métricas de la tabla 10 son representativas del desempeño del contador desarrollado. En consecuencia, un buen desempeño del algoritmo de conteo es crucial para la confiabilidad del contador ya que asegura al usuario que las células son contadas y clasificadas correctamente y que, por ende, los resultados finales de concentraciones celulares, viabilidad y tamaños celulares son precisos. Esto, sumado a la variabilidad de las imágenes del entrenamiento (líneas, tamaños y concentraciones celulares) garantiza que el contador además de ser confiable y preciso es versátil en su aplicación y puede ser empleado en muchas situaciones diferentes para la detección y conteo de células.

La utilización del kit de desarrollo *NVIDIA Jetson Nano* permitió el uso tanto de Cuda 10.2 como de CUDNN 8, *Toolkits* que permiten el acceso de la red a la Unidad de Procesamiento Gráfico (GPU) de la placa. La activación de estas herramientas junto con otras librerías de Python (3.7) permitieron acelerar el proceso de análisis de la imagen y reducir los tiempos de ejecución de la red en gran medida. Mientras que el tiempo de ejecución de YOLO en la placa computadora *Raspberry Pi 3B* es de varios minutos, la implementación realizada con la placa *Jetson Nano* logra tiempos de ejecución menores a los 30 segundos. El tiempo de conteo logrado es, entonces, comparable a los de los mejores contadores comerciales (ver tabla 1) y es notablemente menor al tiempo que le lleva realizar el conteo manualmente a un operador experimentado que varía según la concentración celular, pudiendo alcanzar los 3 minutos de duración [60]. Teniendo en cuenta esto, es evidente que la rapidez del dispositivo desarrollado supone un beneficio para el usuario dado que alivia y reduce el tiempo gastado en una

tarea tediosa y realizada frecuentemente por un investigador. Es importante destacar, sin embargo, que el tiempo de ejecución puede ser optimizado aún más ya que la mayor parte del mismo corresponde a la inicialización de la red. Una red ya inicializada (probablemente apenas se enciende el dispositivo) tendría la capacidad de analizar la imagen casi instantáneamente permitiendo que los tiempos del proceso se redujeran a algunos milisegundos.

Finalmente, es importante recalcar que el dispositivo que se presenta es amigable al usuario y conveniente en la visualización y análisis de los resultados. La interfaz gráfica desarrollada presenta un flujo de uso simple en el cual tanto los botones como la cámara se habilitan y deshabilitan en base a lo que el usuario puede hacer durante el proceso de conteo. Además, la misma genera validaciones en los campos con los que el usuario interactúa para guiarlo si intenta ingresar entradas inesperadas. Por otro lado, la interfaz proporciona múltiples opciones para el análisis de la información facilitando la comprensión de los datos obtenidos para cada análisis. Cuenta también con la opción de escoger el formato de la información para exportar o bien puede guardarla en la interfaz. La exportación de datos proporciona un informe, que incluye todos los datos (visuales y numéricos) en un solo archivo facilitando la lectura y análisis de los mismos. Los resultados presentados en la aplicación se realizan también en forma de una imagen anotada con los *bounding boxes* encontrados y sus respectivas clases, y gráficos de barras con los tamaños de las células encontradas para poder hacer un análisis exhaustivo de la población del cultivo. La interfaz del contador permite también calcular, a partir de los resultados, los volúmenes de solución celular y medio de cultivo que debe combinar para lograr la concentración de células deseada en el nuevo subcultivo. Las características mencionadas permiten, entonces, realizar un mejor conteo y análisis de los cultivos de manera sencilla, intuitiva y exhaustiva.

4.2. Construcción del *Dataset*

Durante este trabajo se llevo a cabo la construcción de un conjunto de imágenes que, hasta el momento, no había sido realizado o puesto a disposición por ningún investigador. Fue posible obtener y clasificar una notable cantidad de imágenes de células individuales que, en un futuro, pueden ser utilizadas para entrenar y validar modelos de clasificación de imágenes o que pueden ser utilizadas para generar imágenes sintéticas usadas en el entrenamiento de modelos de detección o segmentación de objetos. La importancia de esta parte del desarrollo

yace en que habilita a nuevos investigadores a encontrar mejores soluciones y métodos para la detección y clasificación de células en hemocitómetros, utilizables para el conteo de células en imágenes reales, entre otras cosas. Por esta razón, el *dataset* de imágenes sintéticas generado en este trabajo, el conjunto de imágenes reales obtenidas y anotadas y las máscaras de células individuales individuales recortadas y clasificadas fueron publicadas y puestas a disposición de otros investigadores en [2].

Por otra parte, durante este trabajo también se demostró que es posible llevar a cabo el entrenamiento de diferentes modelos de inteligencia artificial para detección de objetos en situaciones reales utilizando imágenes sintéticas. Se ha visto que la generación de imágenes utilizando objetos y fondos reales modificados mediante un procesos de randomización permite la construcción de un *dataset* compatible con el entrenamiento de modelos de detección de objetos. Dicho método permite ahorrar tiempo y recursos usados en la obtención y etiquetado de imágenes reales, y podría ser aplicado no solo en imágenes de células vivas y muertas sino también en cualquier otro tipo de imágenes.

Asimismo, este trabajo ha permitido determinar los parámetros ideales al momento de la generación de imágenes, evaluando como varía el desempeño de cada modelo al entrenarlo con diferentes cantidades de imágenes, diferentes cantidades de objetos por imagen y al variar los aspectos de los objetos incluidos. Este tipo de estudio también es útil para futuras investigaciones que hagan uso de imágenes sintéticas para el entrenamiento de redes neuronales.

4.3. Análisis, Comparación y Selección del Modelo de Detección

Desempeño por clase

Como se menciona anteriormente, todos los modelos presentan un mayor grado de dificultad cuando se trata de localizar y clasificar células muertas. APs y precisiones menores muestran que los modelos son menos robustos para la clase *muerta* cometiendo más errores en la clasificación y detectando una menor cantidad de casos.

Es probable que esto se relacione con el hecho de haber contado con una menor cantidad de objetos de la clase *muerta* para generar las imágenes del entrenamiento. De hecho, para el entrenamiento se utilizaron 321 células muertas pero 701 células vivas; duplicando así la cantidad de células muertas. Así, un mayor numero de objetos diferentes usados en la generación de las

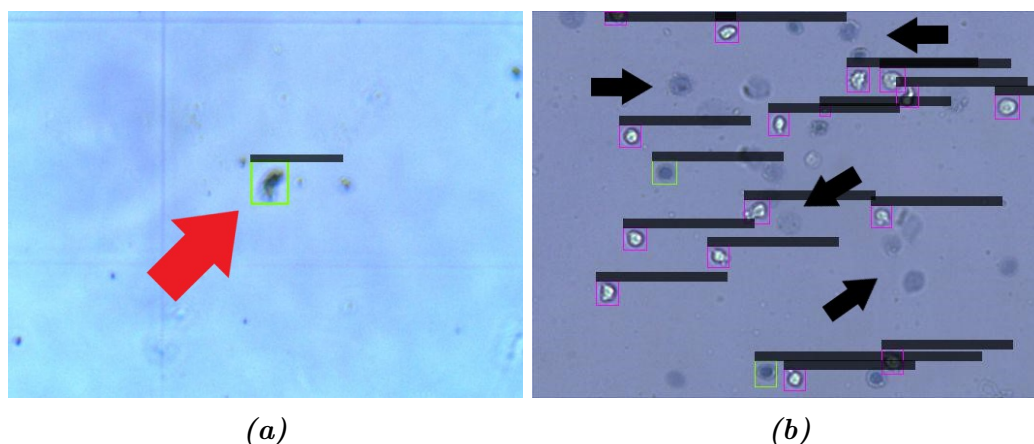


Fig. 4.1: Problemas en la detección de Células Muertas. Las imágenes (a) y (b) corresponden a predicciones realizadas por YOLOv4-608-4 en imágenes reales de prueba. En (a) se muestra un caso en el cual el modelo confunde una pequeña partícula o residuo azules por una célula muerta (flecha roja). En (b) se muestran casos en los cuales el modelo no detecta células muertas difusas.

imágenes, supone una mayor variabilidad de los objetos y una mayor cantidad de características a partir de las cuales el modelo puede aprender. En la práctica es habitual que las muestras contadas posean una mayor proporción de células vivas que muertas, lo que se ve reflejado en las cantidades de objetos obtenidos a partir de las 66 imágenes reales: 1421 vivas contra 520 muertas. En este sentido, si bien es probable que una mayor cantidad de objetos de la clase *muerta* mejoren los resultados, no está claro cuantas imágenes más deberían obtenerse para incrementar considerablemente el número de objetos de la clase *muerta*. Esto puede representar un gran consumo de tiempo, esfuerzo y otros recursos.

También, es posible que las propias características de una célula muerta la conviertan en un objeto difícil de clasificar y detectar. En particular, las células vivas son más claras y definidas y con un mayor contraste respecto del fondo de la imagen. Esto hace que su detección para un humano sea relativamente fácil. Por el contrario, las células muertas suelen ser de un azul oscuro, similar al tono del fondo de la imagen. Además, en muchos casos las células pierden su morfología al morir por lo que dejan de tener una forma redondeada y pasan a tener una forma difusa. Una morfología de este tipo es poco distinguible en imágenes microscópicas, lo que lleva habitualmente a errores de clasificación por operadores humanos [96]. Asimismo, existen casos en los cuales las células muertas son de menor tamaño que las vivas. En estas oportunidades identificar dichas células es también difícil dado que se pueden confundir con

pequeñas partículas o residuos presentes en el hemocitómetro (Figura 4.1).

Es importante tener en cuenta, entonces, que si se quiere obtener un modelo robusto para ambas clases, hace falta evaluar el mAP más que la Precisión y Exhaustividad. Mientras que los niveles de Precisión y Exhaustividad solo tienen en cuenta el número de verdaderos positivos, falsos positivos y falsos negativos (sin importar la clase), el mAP realiza el promedio de los AP de ambas clases eliminando el desbalance que pueda existir entre la cantidad de objetos de cada clase. Una alta Precisión y Exhaustividad no garantizan que el modelo sea robusto en la detección de objetos de todas las clases, ya que se pueden deber exclusivamente a una gran presencia de objetos de la misma clase en las imágenes de prueba y buena detección de objetos de esa clase.

Cantidad de clases

El objetivo de utilizar un modelo con 3 clases es incluir la detección de, además de células vivas y muertas, aglomeraciones o cúmulos celulares. Dichos objetos aparecen debido a errores en el proceso de conteo por parte del operador (pobre pipeteo o resuspensión). Su presencia no es deseada dado que determinar el número de células que lo componen no es posible y, si lo fuese, el colorante no es capaz de penetrar el cumulo celular y llegar a las células en el centro del aglomerado, por lo que conocer el estado de cada célula tampoco es posible. Por todo esto, un modelo que incluya esta tercera clase permitiría no tener en cuenta las células aglomeradas para realizar los cálculos de concentración celular y viabilidad, dando así un resultado más certero. Por otra parte, la detección de su presencia puede servir para alertar al operador de un error en el proceso de resuspensión o preparado de la muestra quien podría decidir repetir el conteo o tener en cuenta la presencia de esta fuente de error. El modelo de 3 clases evaluado presenta, al igual que los demás modelos, un muy buen desempeño en el conjunto de validación. Sin embargo, teniendo en cuenta que los AP en el conjunto de validación para las clases *viva* y *muerta* de todos los modelos en el rango de 96% - 99%, un AP para la clase *cúmulo* de 89% significa que el modelo es menos robusto en la detección de este tipo de objetos. Esto se ve también respaldado si se tiene en cuenta la precisión del modelo para esta clase que es de 80,79% en imágenes sintéticas y llega al 33,33% en imágenes reales.

Estos números demuestran que con este tipo de entrenamiento YOLOv4 no fue capaz de

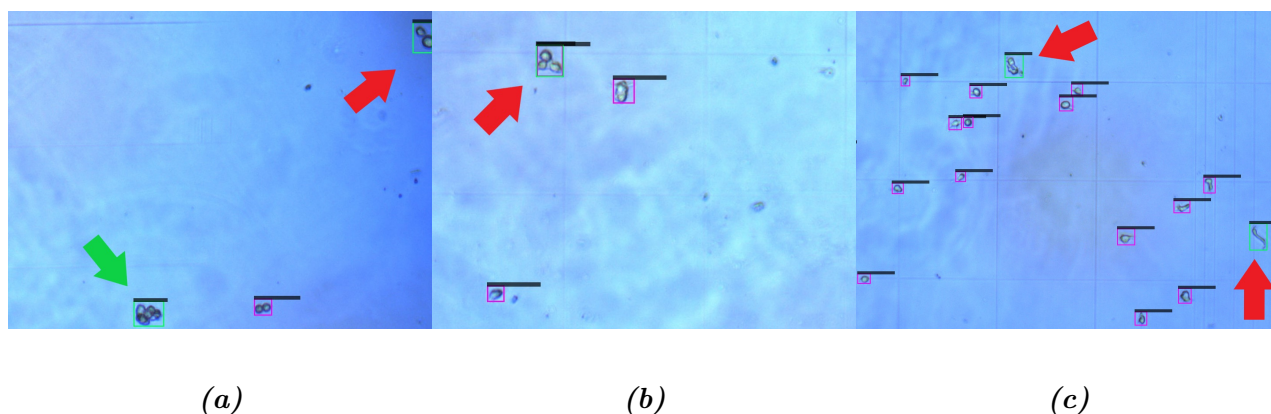


Fig. 4.2: Problemas en la detección de Cúmulos. Las imágenes (a), (b) y (c) corresponden a predicciones realizadas por **YOLOv4-608-3C** en imágenes reales de prueba. En (a) y (b) se muestran dos casos en los cuales el modelo confunde dos células próximas con un cúmulo (flechas rojas). En (c) el modelo confunde dos células con una morfología poco común con un cúmulo (flechas rojas). En (a) el modelo clasifica bien un cúmulo (flecha verde).

aprender a detectar cúmulos celulares. Una alta tasa de falsos positivos, además, indica que **YOLOv4-608-3C** confunde otro tipo de objetos con cúmulos celulares. En efecto, analizando con atención las predicciones en las imágenes de prueba (Figura 4.2) es posible notar dos casos particulares en los cuales el modelo confunde diferentes objetos con cúmulos celulares. El primero de los casos se da cuando dos o más células próximas pero distinguibles entre si son clasificadas como un cúmulo. El segundo caso se da cuando el modelo clasifica células con una morfología no común (alargadas o más grandes de lo común) como si fuesen cúmulos. Ambos casos representan un problema para el contador ya que constituyen una fuente de error en el conteo que con un operador humano no suele estar presente.

Es posible que estos errores se deban a la baja cantidad de cúmulos obtenidos a partir de las imágenes reales y usados en la generación de imágenes sintéticas. Una baja variabilidad en los objetos usados en el entrenamiento significa que el modelo aprenderá solamente acerca de unas pocas características presentes en dichos objetos. Dicho de otra manera: es altamente probable que el modelo haya visto pocos cúmulos celulares o haya visto muchas veces los mismos cúmulos durante el entrenamiento impidiendo que pueda aprender a generalizarlos. Esto tiene sentido si se piensa que, para la generación de las imágenes sintéticas usadas en el entrenamiento se utilizaron alrededor de 32 cúmulos diferentes. Comparado con las 701 células vivas y 321 muertas (tabla 4), es evidente que la cantidad es baja. Esto se debe, particularmente, a que si

el proceso previo al conteo se encuentra bien realizado (por un operador medianamente hábil) no deberían quedar células aglomeradas en la suspensión, por lo que la potencial ocurrencia de cúmulos celulares en las imágenes reales será baja.

En todo caso, queda visto que el uso de un modelo de 3 clases que incluya la detección de cúmulos no agrega el valor deseado al detector. En todo caso, la alta tasa de falsos positivos lleva a pensar que agregar una clase empeora también la clasificación de los objetos del resto de las clases.

YOLOv3 vs YOLOv4

Los resultados presentados en la sección anterior muestran que el desempeño del modelo de YOLOv3 es peor que el de YOLOv4. Cuando se comparan las métricas de **YOLOv3-608** y **YOLOv4-608-3**, modelos entrenados con el mismo conjunto de imágenes, se observa que YOLOv4 supera en todos los aspectos a YOLOv3. Ya sea en el conjunto de validación sintético o de test real YOLOv4 obtiene un mejor mAP, mejor Precisión, Exhaustividad e IoU promedio. A su vez, supera a YOLOv3 en cuanto al AP y Precisión en ambas clases. Esto demuestra que aún con el mismo método de entrenamiento (*dataset* y parámetros), YOLOv4 es un modelo más robusto en cuanto a la detección y clasificación de objetos pequeños y, en particular, células vivas y muertas. En otras palabras, YOLOv4 es capaz de detectar más (mayor R) y mejor (mayor P). Esto, en definitiva, es lo que se espera de un detector de objetos. Por otra parte, teniendo en cuenta que la brecha es aún mayor para el conjunto de pruebas con un umbral de 0.5 (Tabla 8), es posible pensar que YOLOv4 realiza detecciones con mayor exactitud, siendo capaz de alinear de forma más precisa las cajas predichas al objeto real, y que YOLOv4 posee una mayor capacidad de generalizar a partir de un entrenamiento con imágenes sintéticas.

Cantidad de Células por imagen

Una de las ventajas de realizar un entrenamiento con imágenes sintéticas es que permite diseñar el aspecto de las imágenes. En particular, cuando se trata de de detección de objetos es posible variar voluntariamente la cantidad de objetos presentes por imagen. Esto determina, por otra parte, cuantos objetos utiliza el modelo para aprender. Si a esto se le suma la posibilidad de variar el aspecto de cada objeto cuando se genera la imagen (brillo, color, orientación, tamaño, etc.) se convierte en una poderosa herramienta para mejorar los resultado del entrenamiento.

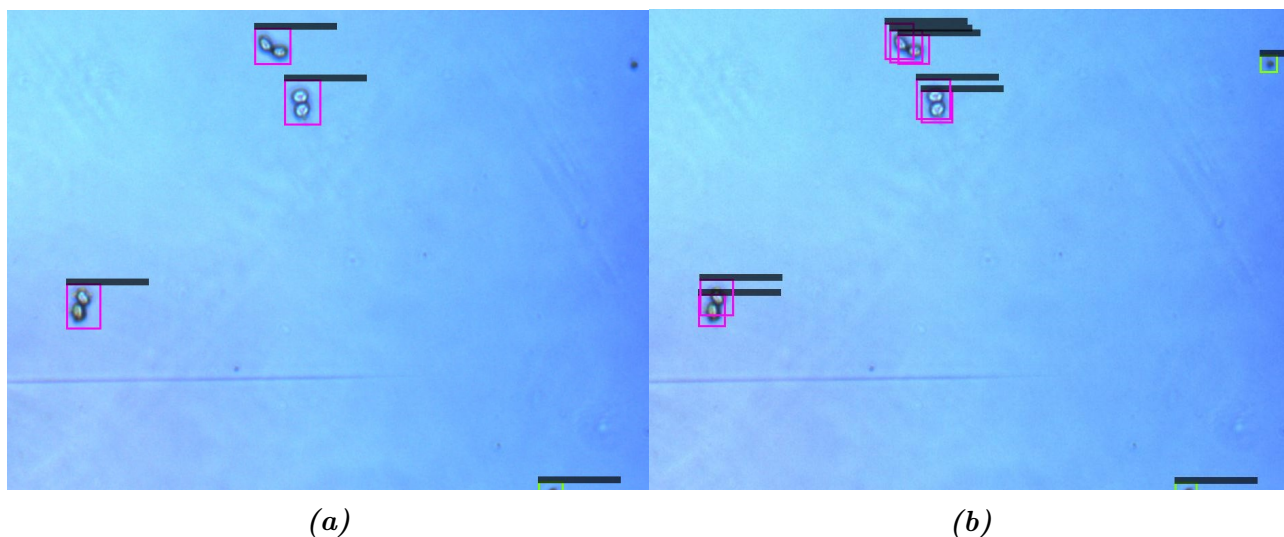


Fig. 4.3: Detecciones próximas. La imagen (a) corresponde a una predicción del modelo **YOLOv4-608-3** y la (b) a una predicción de **YOLOv4-608-4**. Se puede notar como el entrenamiento con imágenes de mayor concentración de células permite una mejor detección de objetos próximos, evitando que dos células que están en contacto sean consideradas como una sola.

En este sentido, se compararon los resultados en detecciones del mismo modelo entrenado con tres *datasets* diferentes, donde la única variable fue la cantidad de células por imagen (**YOLOv4-608-3**, **YOLOv4-608-4** y **YOLOv4-608-5**). De los resultados obtenidos se puede obtener que los modelos más robustos son aquellos entrenados con una mayor cantidad de células. En particular, el modelo entrenado con 0-75 células vivas por imagen es aquel con el AP para la clase *viva* más bajo; y el modelo entrenado con 0-60 células muertas por imagen es el que posee un AP para células muertas más bajo. Por otra parte, el modelo entrenado con más células muertas por imagen (**YOLOv4-608-4**) presenta el mejor AP_D de los tres modelos (ver tabla 7).

Otra ventaja de aumentar la cantidad de células por imagen, es una mejora en la detección de células próximas (Figura 4.3). Probablemente, se deba al producto de un aumento de células próximas entre si en las imágenes sintéticas generadas. Esto, en definitiva ayuda al aprendizaje de la red a detectar este tipo de casos.

Curiosamente, el modelo entrenado con mas células por imagen (**YOLOv4-608-4**) presenta peores exhaustividades en imágenes reales (ver tabla 8 y 9). Es probable que esto se deba a que el modelo se haya entrenado con imágenes donde el mismo objeto se repite muchas muchas veces.

Esto hace que la red incorpore mucha información acerca de una cantidad reducida de objetos pero que sea incapaz de generalizar para detectar objetos que no conoce. La exhaustividad entonces disminuye en imágenes desconocidas para el modelo, donde las células son diferentes a las del entrenamiento. El balance, de cualquier manera, debe encontrarse evitando que una elevada cantidad de células por imagen hagan que el modelo observe demasiadas veces el mismo objeto, llevándolo a un posible *overfitting*. Se determinó entonces, que generar imágenes con una cantidad de células vivas y muertas por imagen de entre 0 y 100 permite imitar las situaciones de conteos reales, permite la ocurrencia de células cercanas y evita un desbalance entre las cantidades de objetos por clase, lo que podría condicionar el aprendizaje de la red. En todo caso, para determinar una cantidad de objetos por imagen ideal servirían mas ensayos.

Data Augmentation

Otra de las estrategias ensayadas para mejorar el desempeño del modelo de detecciones en

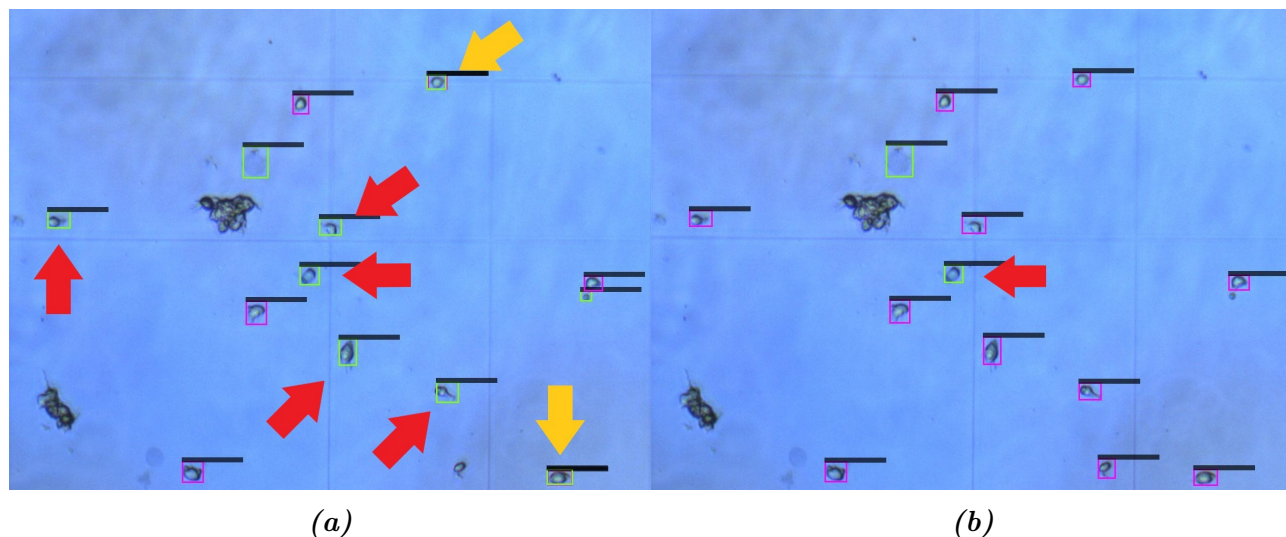


Fig. 4.4: Detección de células con y sin DA. La imagen (a) corresponde a una predicción del modelo **YOLOv4-608-9** y la (b) a una predicción de **YOLOv4-608-10**. Se puede notar como el entrenamiento con imágenes con DA permite una mejor clasificación de las células detectadas, evitando que células vivas (recuadros rosa) sean consideradas como muertas (recuadros amarillo). En (a) se dan 5 casos de células vivas clasificadas como muertas (flechas rojas), incluyendo 2 casos clasificados como viva y muerta simultáneamente (flechas naranja). En (b), el modelo solo incurre en uno de esos errores.

imágenes reales fue la incorporación de cambios en el brillo, contraste y nitidez en los fondos y objetos al momento de generar cada imagen sintética. Al comparar el modelo **YOLOv4-608-9**

(sin DA) contra **YOLOv4-608-10** (con DA), se observa que el segundo supera al primero en mAP, Precisión y Exhaustividad (ver tablas 8 y 9). Para ambas clases los APs obtenidos y las precisiones logradas se incrementaron en porcentaje, notablemente, al incluir los cambios mencionados en el aspecto de las células. Posiblemente esta mejora se deba a que, durante el entrenamiento, el modelo observe una mayor cantidad de objetos visualmente diferentes pero que pertenecen a la misma clase. Esto evitaría que el modelo memorice cierto aspecto del objeto para determinada clase y que, en cambio, pueda generalizar la detección para una mayor cantidad de casos diversos. Particularmente, es posible que existan casos en los cuales debido a las condiciones en las que la imagen fue capturada, la misma posea un brillo, contraste o nitidez distorsionado. En estos casos, una célula viva podría aparecer más oscura y confundirse con una muerta o viceversa. Incluir el aumento de datos (DA) en la generación de imágenes significa que el modelo se entrena y aprende contemplando casos fuera de la norma en los cuales las imágenes (y consecuentemente las células) poseen un aspecto distinto respecto al brillo, color, nitidez o contraste. **YOLOv4-608-10**, el modelo entrenado de esta forma, demuestra ser más robusto en las predicciones en ambas clases, obteniendo una mayor proporción de verdaderos positivos y una menor proporción falsos positivos. En efecto, al compararlo con **YOLOv4-608-9**, **YOLOv4-608-10** se equivoca menos en la clasificación de las células en imágenes reales (Figura 4.4).

Cantidad de imágenes positivas y negativas

Al aumentar la proporción de imágenes negativas respecto de las positivas de 1:5 (**YOLOv4-608-10**) a 1:1 (**YOLOv4-608-11**) se notan mejoras en el mAP, Precisión y Exhaustividad (tablas 8 y 9). Esto indica una mayor robustez general del modelo, y una mejora en el aprendizaje para ambas clases. Probablemente, una mayor cantidad de imágenes negativas ayude al modelo a evitar incurrir en falsos positivos al realizar detecciones con diferentes umbrales de confianza. En definitiva esto permite mejores precisiones a diferentes niveles de exhaustividad llevando a un mAP más alto y a detecciones con mayores niveles de confianza.

Diferencia de performance según el umbral de IoU

En el trabajo original de YOLOv3 [Redmon y Farhadi](#) escriben: “el desempeño decae significativamente a medida que el umbral del IoU aumenta indicando que YOLOv3 tiene problemas

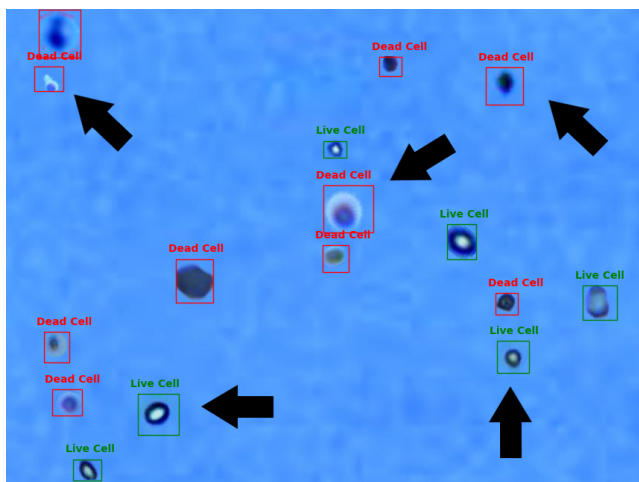


Fig. 4.5: Errores en la generación de anotaciones automáticas. El problema surge con una mala segmentación de las imágenes a la hora de obtener las células usadas en la generación de imágenes sintéticas. Las células marcadas en esta imagen fueron anotadas con una caja contenedora mas grande que lo que corresponde. La imagen de la figura es del conjunto de entrenamiento de un Dataset sin FT.

para obtener cajas contenedoras perfectamente alineadas con los objetos”. Esto se ve claramente reflejado en todos los modelos analizados (no solo YOLOv3), donde las métricas del IoU 50 % son de hasta un 15 % peores que las métricas de IoU 30 %.

Sin embargo, es posible pensar que el problema no sea causado enteramente por la naturaleza del modelo usado. Los resultados en imágenes sintéticas (Tabla 7) muestran que YOLO (v3 y v4) es capaz de localizar relativamente bien, objetos pequeños como lo pueden ser células. En efecto, un IoU promedio por sobre el 70 % demuestra una buena capacidad de localizar y dimensionar objetos. Es probable, entonces, que el problema se encuentre en las diferencias entre una imagen sintética y una real y que surja a partir de un entrenamiento basado en imágenes sintéticas.

Para entender esto, hace falta tener en cuenta el proceso utilizado para crear una imagen sintética. En este sentido, los objetos utilizados para la generación de imágenes sintéticas fueron manualmente “recortados” de imágenes reales. Sin embargo al momento de recortar las células, es posible que la segmentación no haya sido perfecta, y que el límite de la máscara resultante no fuese perfectamente coincidente al borde de la célula. Al suceder esto, y dado que la generación de las anotaciones utilizadas para el entrenamiento se realizó automáticamente y de forma simultánea con la generación de las imágenes, es probable entonces que el error en la segmenta-

ción del objeto se trasladase a la anotación de las imágenes utilizadas en el entrenamiento. En otras palabras, la generación involuntaria de máscaras más grandes que la célula en sí, deriva en anotaciones donde la caja contenedora es más grande que la célula presente en la imagen. Esto es posible que haya forzado a YOLO a aprender a predecir cajas de mayor tamaño que la real. En efecto, la imagen de la Figura 4.5 muestra el problema mencionado.

Al mismo tiempo, esto demuestra las diferencias entre los resultados de las predicciones en imágenes sintéticas (Tabla 7) contra las predicciones en imágenes reales (Tabla 8). A pesar de que las imágenes de ambos conjuntos nunca fueron usadas en el entrenamiento, las métricas en imágenes sintéticas son invariablemente mejores. Podemos explicar esto teniendo en cuenta que, mientras las imágenes de validación fueron generadas con la misma metodología que las de entrenamiento, las imágenes de prueba son reales, y sus anotaciones fueron realizadas manualmente. Anotar manualmente una imagen supone que el operador “dibuja” acorde a su criterio la caja alrededor de la célula. Si el operador es medianamente hábil, es evidente que las anotaciones realizadas de esta manera carecerán del error anteriormente mencionado. Asimismo, dado que el IoU equivale al porcentaje del área compartida entre las dos cajas, es evidente que en objetos más pequeños una pequeña diferencia entre la caja predicha y la verdadera equivale a un peor IoU. Efectivamente, una desalineación de pocos píxeles entre las dos cajas provoca que el IoU se vea altamente afectado. Esta, podría ser otra de las razones por la cual las pequeñas desalineaciones al predecir la localización de las células de imágenes reales causarían una gran caída en las métricas del modelo. Además, por lo mencionado anteriormente, cajas contenedoras más grandes (en el conjunto de validación) permiten obtener un mayor IoU con las cajas de las predicciones, resultando en mejores AP, P, R e IoU medio.

Entendiendo este problema, se decidió disminuir el umbral del IoU para poder analizar y comparar la clasificación de los objetos según el modelo. En este sentido, si bien en la práctica se considera una predicción con un IoU mayor al 50 % como buena, consideramos utilizar un umbral del 30 % permite tener en consideración aquellos casos en los cuales el modelo es capaz de encontrar y clasificar correctamente la célula, por mas que la caja predicha no este perfectamente alineada con la real. Calculando y comparando métricas con diferentes umbrales de IoU es posible, además, conocer la cantidad de detecciones que si bien no llegan al 50 % de IoU, se encuentran por encima del 30 %, y entender si el modelo sufre de problemas de

clasificación de los objetos o si presenta solamente dificultades en la alineación de las cajas predichas.

Por otra parte, analizar métricas en diferentes umbrales de IoU sirve para poner en discusión la utilidad y valor diferencial que suman detecciones con 20 puntos más de IoU dado que el propósito del dispositivo es “encontrar y clasificar” de forma correcta las células. En efecto, [Russakovsky et al.](#) reportan que “entrenar humanos a inspeccionar visualmente una caja contenedora con un IoU de 0.3 y distinguirla de una con un IoU de 0.5 es sorprendentemente difícil”. Esto es aún más válido cuando se trata de objetos pequeños donde la diferencia entre un IoU de 0.3 o 0.5 puede ser de algunos pocos píxeles.

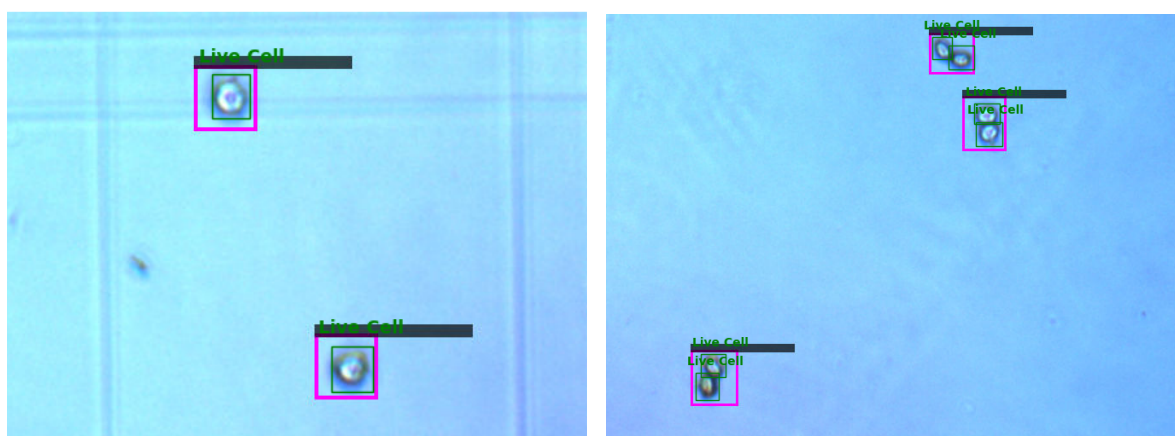


Fig. 4.6: Cambio de umbral de IoU. El efecto resultante del cambio de umbral de IoU durante la clasificación de objetos puede generar una mejora en las métricas de la red.

La figura 4.6 ilustra dos casos que muestran las consecuencias de cambiar el umbral de IoU. A la derecha se observa un caso en el cual dos células cuyas predicciones (caja rosa), a pesar de tener una correcta clasificación (vivas), tienen un IoU menor a 0.5 respecto de la anotación real (caja verde). En este caso, disminuir el umbral de 0.5 a 0.3 hace que estas dos predicciones pasen de ser considerados como Falsos Positivos a ser considerados Verdaderos Positivos, por lo que en lugar de tener dos Falsos Positivos y dos Falsos Negativos (con un umbral de 0.5), con un umbral de 0.3 se tienen dos Verdaderos Positivos. Por otra parte, a la derecha se observa el caso en el cual una caja predicha abarca dos objetos diferentes. Aquí, la clasificación es nuevamente correcta y el IoU de la predicción con cualquiera de los dos objetos es menor a 0.5. En este caso, un umbral de 0.3 hace que cada predicción sea considerada como un solo verdadero positivo, en lugar de dos falsos positivos. Las predicciones en las 6 células pasarían de ser 3

Falsos Positivos y 6 Falsos Negativos a ser 3 Verdaderos Positivos y 3 Falsos Negativos. De esta manera, se considera que se ha detectado y clasificado bien una de las dos células vecinas, pero se penaliza que no se hallan detectado ambas. En todo caso, queda puesto en evidencia que cambiar el umbral para el cálculo de las métricas del modelo es una alternativa viable ya que permite obtener una medida razonable de que tan bien el modelo localiza y clasifica los objetos, sin dejar de penalizar clasificaciones erróneas ni objetos no detectados. Aún más, dado que se trata de detecciones de objetos pequeños, donde un cambio del IoU entre 0.3 y 0.5 es de unos pocos píxeles. Cabe aclarar, sin embargo, que detecciones con menores niveles de IoU afectarían el análisis del tamaño de los objetos detectados. Es decir, que los potenciales errores en la información obtenida a cerca del tamaño la célula detectada serian mayores y que, por ende, un análisis acerca del tamaño de las células del cultivo sería menos confiable.

Ajuste fino de *Bounding boxes*

Teniendo en cuenta los problemas mencionados, el ajuste fino de los *bounding boxes* consiste

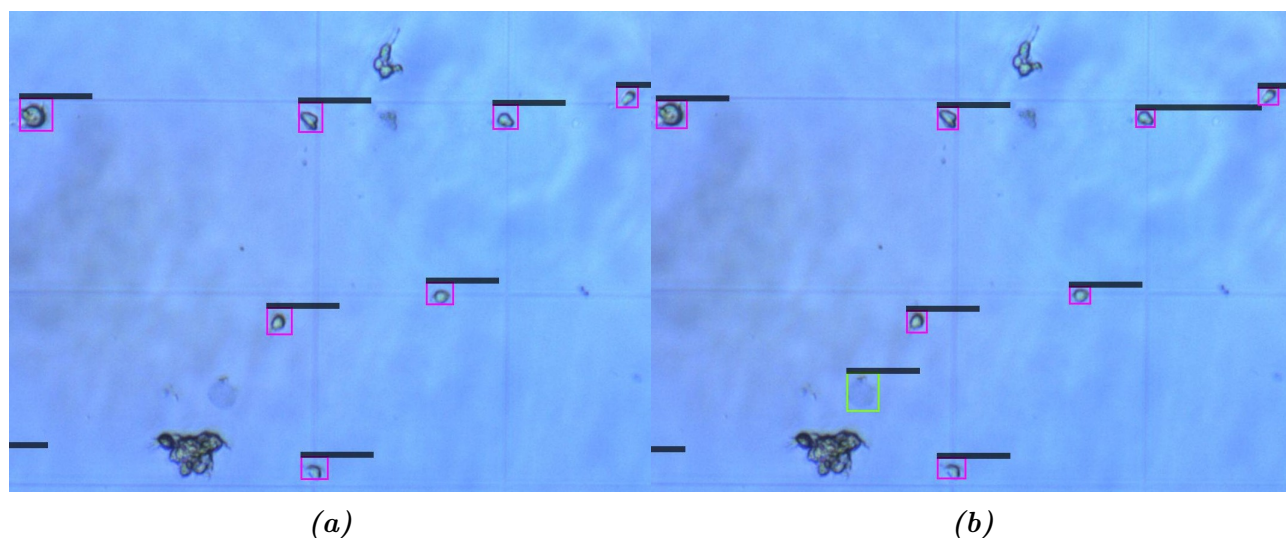


Fig. 4.7: Deteccion de celulas con y sin FT. La imagen (a) corresponde a una predicción del modelo YOLOv4-608-5 y la (b) a una predicción de YOLOv4-608-10. En (a) las cajas predichas por el modelo sin FT son más amplias y tienen un área de coincidencia con el objeto menor. En (b) las cajas predichas por el modelo con FT las cajas se “ajustan” en mayor medida a las células detectadas obteniendo así mejores niveles de IoU.

en corregir y recortar de manera más exacta las células individuales a la hora de realizar las máscaras usadas en la generación de imágenes, permitiendo que las anotaciones usadas en el

entrenamiento sean más exactas y que cada *bounding box* se ajuste de mejor manera a la célula en sí, tal como sería en una anotación manual. En teoría se espera que esto permita que el modelo aprenda a alinear de mejor manera sus predicciones a los objetos en sí, mejorando su desempeño en mayores umbrales de IoU.

En la práctica se vió que **YOLOv4-603-10** posee un mayor mAP e IoU promedio que **YOLOv4-603-3**, **YOLOv4-603-4** y **YOLOv4-603-5**, entrenados bajo las mismas condiciones pero sin el ajuste fino, cuando se considera el umbral de 0.5. Dicho resultado permite ver como, cuando las anotaciones de cada objeto en las imágenes de entrenamiento son más exactas el modelo es capaz de lograr mejores localizaciones de objetos reales dado que durante el entrenamiento el modelo aprende a predecir mejores cajas (Figura 4.7).

Por otra parte, el entrenamiento con un ajuste fino previo también resulta en menores diferencias entre las métricas con IoU 0.5 y 0.3, mientras que aquellos sin el ajuste fino (primeros 5 modelos) exhiben diferencias de entre 12 y 20 puntos de mAP (con diferentes umbrales). Los modelos entrenados con dicho ajuste sólo llegan a tener diferencias de 3 puntos de mAP. Esto significa que cuando se realiza un ajuste fino en las anotaciones usadas en el entrenamiento, las métricas obtenidas con un IoU de 0.5 describen relativamente bien al modelo analizado en cuanto a su capacidad de clasificación, por lo que no sería indispensable evaluar las métricas de un IoU 0.3.

Curiosamente, los modelos entrenados sin FT (**YOLOv4-608-3**, **YOLOv4-608-4** y **YOLOv4-608-5**) presentan mejores mAPs (78 % - 81 %) que **YOLOv4-608-10** (72,31 %) cuando el umbral utilizado es del 30 % de IoU. Dado que entre estos modelos la principal diferencia es el ajuste de las anotaciones usadas en el entrenamiento, es posible hipotetizar que el hecho de ser entrenado con cajas contenedoras más estrechas al objeto provoque que durante el entrenamiento, el modelo no pueda imitar exactamente esas cajas y que presente más dificultades para minimizar la función de optimización. A la hora de realizar detecciones en imágenes reales el modelo entonces tendría una peor capacidad de “encontrar” y clasificar correctamente las células presentes (verdaderos positivos) resultando en peores precisiones (ver tabla 9). Por otro lado, las predicciones que si realiza correctamente tienen un mayor IoU que los modelos sin FT, mostrando mejores métricas con un umbral de 0.5 (ver tabla 8).

Optimización de *anchor boxes*

Otra explicación para los problemas de YOLO en la alineación de las cajas predichas, puede

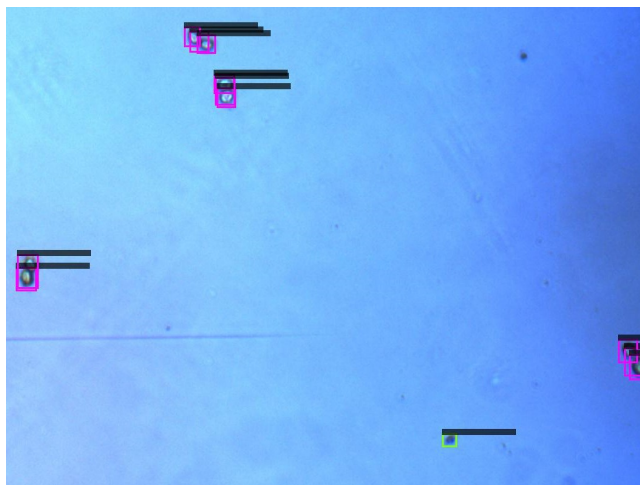


Fig. 4.8: Problemas en detecciones con OA. La imagen corresponde a una predicción del modelo **YOLOv4-608-6**. En la imagen se pueden observar múltiples cajas (detecciones) asignadas para cada célula presente. Esto deriva en un aumento en la cantidad de falsos positivos y un empeoramiento en la precisión del modelo.

darse como producto de la selección de los *anchor boxes* usados en el entrenamiento. Dado que las células son, en su mayoría, objetos pequeños, sería ideal optimizar las cajas utilizadas con el motivo de agilizar el entrenamiento y mejorar las predicciones finales.

Al entrenar redes con *anchor boxes* optimizados (OA) (modelos 6, 7 y 8), sin embargo, se vió un importante empeoramiento en todas las métricas. Notablemente, todos los modelos entrenados con cajas optimizadas obtuvieron precisiones globales mas bajas ($<76\%$ en test y $<93\%$ en validación), y peores precisiones por clase en imágenes reales ($<88\%$ en vivas y $<48\%$ en muertas). Particularmente, YOLOv4-608-9 que fue entrenado con una menor cantidad de imágenes negativas, presentó un desempeño peor de lo esperado. Este modelo logró los peores desempeños en cuanto al mAP, Precisión y Exhaustividad en imágenes reales. Es curioso que, además, los tres modelos entrenados de esta manera hayan obtenido los peores niveles de IoU, mostrando que aún con una optimización de las cajas usadas en las predicciones, la alineación no mejora. Debemos mencionar, entonces, que los tres modelos presentan una gran cantidad de falsos positivos al realizar predicciones lo que provoca que las precisiones sean bajas.

Todo esto puede estar asociado a que al momento de entrenar, el modelo no es capaz de minimizar la función de *loss* utilizando *anchor boxes* optimizados. Probablemente esto se deba a

que los *anchor boxes* más pequeños son más difíciles de alinear a las cajas reales, lo cual dificulta la minimización de la función CIoU que dirige el entrenamiento. Valores más grandes de la función *loss* y mayores dificultades para alinear las predicciones con las cajas verdaderas pueden significar que el entrenamiento no llegue a converger en una solución ideal. Este razonamiento también explica las diferencias que se dan cuando a un modelo con OA se le aumenta la proporción de imágenes negativas. Los modelos con proporciones 1:1 (6 y 7) tendrían mayor probabilidad de converger a una mejor solución dado que durante el entrenamiento utilizan una menor cantidad de imágenes positivas, lo que le evita al modelo tener que alinear perfectamente una gran cantidad de cajas predichas para minimizar la *loss*. Los resultados, sin embargo, aún presentan peores precisiones.

Al observar las predicciones realizadas en imágenes reales, se nota además que con frecuencia estos modelos realizan múltiples detecciones para el mismo objeto (Figura 4.8). Esto puede estar asociado al hecho de que, durante el entrenamiento, el modelo tenga dificultades de elegir y usar una sola caja precursora para realizar las predicciones dado que las dimensiones de las cajas precursoras son similares entre sí. Todo esto deriva en múltiples detecciones con diferentes *anchor boxes* para el mismo objeto, lo que resulta en un elevado número de falsos positivos y bajos niveles de precisión.

Efectos de un Dataset sintético

El primer problema con el uso de imágenes sintéticas, es que la extracción de los objetos individuales implica la pérdida de su contexto real. Esto significa que el modelo pierde la capacidad de aprender características acerca del contexto natural de los objetos. Una de las principales características de YOLO es que, al ser un detector de una sola etapa, razona en base a la imagen en su totalidad y codifica, implícitamente, información acerca del contexto de los objetos que son utilizados durante el entrenamiento y la detección. El uso de imágenes sintéticas, donde cada objeto ha sido modificado y colocado junto a un fondo aleatorio, no mantiene la información del contexto real de cada célula para llevar a cabo su detección y clasificación.

En general, información acerca del fondo de la imagen, como por ejemplo el color y brillo se relaciona con la visualización del objeto. En particular, es probable que si al obtener la imagen

real la iluminación es alta, el brillo del fondo sea alto, su color un azul claro, que las células vivas se vean con un alto brillo y un color claro, y las muertas con un alto contraste. En cambio, una imagen poco iluminada tendrá un fondo azul oscuro, células vivas más oscuras y muertas poco contrastadas con el fondo. Siguiendo con el ejemplo, un modelo sintético combina objetos y fondos obtenidos en diferentes condiciones por lo que es probable que células vivas oscuras (aquellas obtenidas con poca iluminación) sean colocadas en fondos claros.

Por todo esto, si bien la generación de imágenes sintéticas se presenta como una opción válida para un entrenamiento, la importancia yace en poder imitar lo mejor posible imágenes reales. Esto en general dependerá de los parámetros utilizados para la generación del *dataset*, y de la habilidad de lograr una randomización de las células sintéticas conservando el realismo. Herramientas como la rotación, espejado y redimensionamiento son altamente potentes a la hora de crear una mayor variabilidad en las imágenes. Por otra parte, cambios en el brillo o tono de las células puede causar que se incluyan aspectos no reales de los objetos en imágenes de entrenamiento. Sería necesario, entonces, realizar un estudio más profundo acerca de la variación en el desempeño de los modelos para encontrar los parámetros óptimos de generación de imágenes. A todo esto, se le suma la cantidad de células por imagen que conviene utilizar y la cantidad de imágenes óptimas para el entrenamiento.

Efectos de un Dataset mixto

Entendiendo las limitaciones de un dataset puramente sintético, un entrenamiento mixto (en el cual se utilizan imágenes sintéticas y reales) permite obtener mejores resultados. Los modelos **YOLOv4-608-6**, **YOLOv4-608-7** y **YOLOv4-608-12** (entrenados con un dataset mixto) lograron altos valores de mAP y AP por clase en imágenes reales, mostrando, así, una mayor robustez en la detección de células y una capacidad de detectar y clasificar células con mayor confianza (ver tablas 8 y 9).

A su vez, al comparar **YOLOv4-608-12** (entrenamiento mixto) con **YOLOv4-608-11** (entrenamiento sintético) se observa que el modelo mixto consigue un incremento notable en su mAP y en su AP de la clase muerta. Esto indica un mejor aprendizaje para detectar y clasificar células vivas y, sobre todo, muertas cuando el dataset incluye imágenes reales. Uno de los efectos, entonces, es la capacidad de los modelos mixtos de detectar células muertas difusas que son,

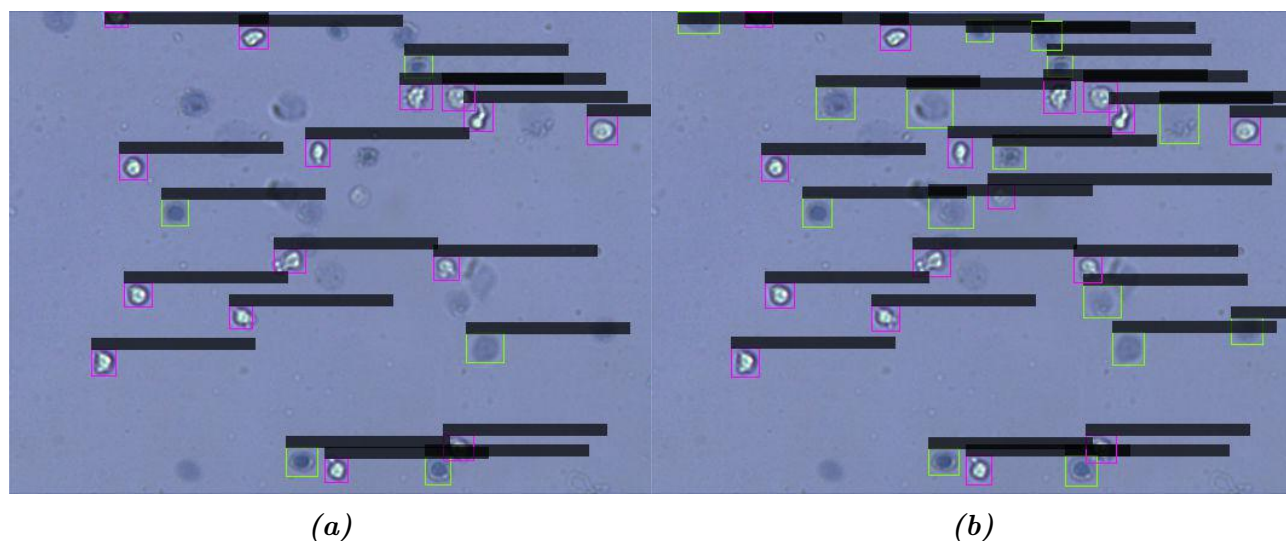


Fig. 4.9: Detección de células con y sin Mix. La imagen (a) corresponde a una predicción del modelo **YOLOv4-608-11** y la (b) a una predicción de **YOLOv4-608-12**. En (a) gran parte de las células muertas con un aspecto difuso y poco determinado no son detectadas mientras que en (b) la mayoría de dichas células son detectadas y clasificadas correctamente.

en gran parte, observables y detectables gracias al contraste del objeto con el fondo (Figura 4.9). **YOLOv4-608-12** también logra una mayor cantidad de verdaderos positivos entre sus detecciones, lo que se traduce en un R mayor, mientras consigue niveles de precisión ligeramente mayores. En definitiva, se puede ver que el entrenamiento mixto genera un modelo de detección más robusto, exhaustivo y preciso que un entrenamiento puramente sintético.

Como se mencionó en la anterior sección, es probable que esto se deba a la pérdida del contexto de los objetos durante la generación de las imágenes sintéticas. Un entrenamiento mixto permite que YOLO observe y aprenda a detectar células en su contexto real utilizando información acerca del entorno de las células (a parte de la información de los objetos en sí). Es de esperar que una mayor cantidad de imágenes reales usadas en el entrenamiento devenga en mejores resultados dado que la proporción de imágenes reales a sintéticas sería mayor. Esto se puede ver cuando se comparan **YOLOv4-608-6** y **YOLOv4-608-7** en donde se utilizan 192 imágenes reales en un conjunto de 2000 y 8000 imágenes respectivamente (ver tabla 8). Posiblemente, un dataset puramente real permitiría obtener aún mejores resultados en cuanto a niveles de mAP, P y R.

Selección del modelo final

La selección del modelo final se realizó observando el desempeño de los diferentes modelos en imágenes reales (tabla 8 y 9) y teniendo en cuenta:

- Robustez global del modelo (mAP)
- Robustez del modelo para cada clase (AP_L y AP_D)
- Precisión del modelo (P)
- Exhaustividad del modelo (R)
- Capacidad de localización de objetos (IoU promedio y métricas con diferentes umbrales)

A partir del análisis realizado se llegó a que:

- El desempeño de YOLOv3 respecto de YOLOv4 es más pobre, por lo que se recomienda utilizar modelos de la cuarta versión.
- El modelo entrenado para detectar 3 clases diferentes posee un peor desempeño que aquel entrenado para 2 principalmente gracias al aspecto intrínseco de esa tercera clase y gracias a la baja ocurrencia y disponibilidad para entrenar de objetos de esa tercera clase.
- Los modelos 3, 4 y 5 poseen un buen desempeño en general pero presentan grandes problemas para conseguir detecciones que superen el umbral de 50 % del IoU, debido a errores en la segmentación manual de los objetos usados para el entrenamiento.
- Los modelos 6, 7 y 8, entrenados con *anchor boxes* modificados incurrir en mayores errores de detección derivando en un mayor número de falsos positivos y peores precisiones.
- El ajuste fino de las anotaciones usadas en el entrenamiento permite llevar a una mayor exactitud en las cajas de las predicciones generadas aumentando el IoU de las detecciones y mejorando el desempeño de los modelos cuando el umbral del IoU es del 50 %.
- La inclusión de modificaciones en el aspecto de los objetos (brillo, contraste y nitidez) ayuda a que los modelos logren una mejor performance.

- Una proporción 1:1 de imágenes positivas contra negativas produce modelos mas robustos que una proporción 5:1.
- La inclusión de imágenes reales en el *dataset* permite que el modelo aprenda a utilizar información acerca del entorno de los objetos para llevar a cabo las detecciones. Esto mejora su performance.

Tomando en consideración todo esto, el modelo elegido para la implementación del dispositivo fue **YOLOv4-608-12** que, en imágenes reales logra las métricas de la tabla 10.

Tabla 10: Métricas del modelo final.

	mAP ₅₀	P	R	F ₁	IoU	AP _L	AP _D	AP _C	P _L	P _D	P _C
YOLOv4-608-12	87,30 %	88,47 %	92,09 %	90,24 %	67,63 %	97,09 %	77,52 %	-	95,65 %	66,49 %	-

Todos los valores de la Precisión Media Promedio (mAP), Precisión (P) y Exhaustividad (R) fueron obtenidos usando un umbral de IoU del 50 %. Las Precisiones y Precisiones Media por clase se diferencian según su subíndice; Viva (L), Muerta (D), Cumulo (C).

5. Conclusión

Este Proyecto Final de Carrera consistió en un desarrollo multidisciplinario que involucró conocimientos de biología, diseño e impresión 3D, electrónica, informática, ingeniería de tejidos, óptica y redes neuronales, teniendo como resultado un dispositivo contador de células automático funcional. En la vida de un Bioingeniero el trabajo multidisciplinario así como el trabajo con personas especializadas en diferentes disciplinas es de vital importancia y es por eso que se decidió llevar a cabo el mismo.

El dispositivo presentado en este trabajo es portátil, de muy bajo costo y replicable utilizando componentes comercialmente disponible y piezas impresas con tecnologías 3D, lo que lo hacen accesible para un gran rango de profesionales. Tanto su estructura y funcionamiento, como la magnificación y resolución logradas lo hacen adecuado para el conteo de células y, potencialmente, para otras aplicaciones en el campo de la microscopía y/o cultivo celular. Asimismo, el proyecto incluyó la construcción de un *dataset* compuesto de imágenes sintéticas con células teñidas con azul de tripán y el entrenamiento de un modelo de red neuronal optimizado para la detección y conteo de células animales que constituyen desarrollos altamente innovadores y que, hasta ahora, no habían sido encontrados en ninguna investigación científica

realizada previamente. Al analizar, entonces, el contador de células construido en este trabajo se observa que además de ser económico y portátil, los desempeños en cuanto a su precisión y al tiempo de conteo son comparables con los dispositivos disponibles en el mercado, haciendo que este desarrollo tenga valor por su carácter innovador y por su aplicabilidad.

Finalmente, es oportuno destacar que, para este proyecto no solo se cumplieron los objetivos mínimos planteados, sino que también gran parte de los objetivos de máxima incluidos. Dada la situación mundial y la imposibilidad de acceder a los laboratorios de la institución, nos vimos forzados a volver sobre las bases del proyecto y plantearlo desde otra perspectiva. Es así como surgió la implementación de un *dataset* sintético que, por sí solo, consideramos un desarrollo de gran importancia junto con los resultados y pesos obtenidos de entrenar, una red neuronal YOLOV4 con estas imágenes ya que se lograron resultados más que aceptables sin la necesidad de generar un *dataset* totalmente real, lo que hubiera significado un gasto en tiempo y costos. Ambos desarrollos, el *dataset* de imágenes sintéticas y el modelo entrenado, quedan publicados en [2].

6. Futuras Mejoras y Aplicaciones

6.1. Modelo de detección

El mayor desafío dentro de este desarrollo fue la creación de un *dataset* de características similares a un conjunto de imágenes de subcultivos tomadas en el laboratorio. Como se ha explicado, las imágenes utilizadas para el entrenamiento del modelo fueron obtenidas a partir de un *dataset sintético* creado con el propósito de sobreponerse a la imposibilidad de obtener una gran cantidad de imágenes reales. Si bien las ventajas de este método son varias (tiempo, dinero, cantidad de imágenes generadas, anotaciones automáticas, etc.), su uso no es ideal por diferentes razones. Así, de la imposibilidad de generar un *dataset* con imágenes reales de subcultivos, surge el primer y principal aspecto a mejorar del proyecto.

En primer lugar, la ampliación del conjunto de imágenes reales de cultivo proporcionaría un mejor parámetro de cómo se ven en una imagen las células en el proceso de conteo y nos permitiría generar un modelo más robusto y generalizable. Entonces, la utilización de un mayor número de imágenes reales (ya sea en un entrenamiento mixto o puramente real) sería benéfico

para el modelo. Además, un aumento en la cantidad de células en todos los conjuntos de imágenes nos permitiría ampliar la cantidad de imágenes en el caso de generar imágenes sintéticas. Por último, un trabajo más amplio dedicado a la construcción de un *dataset* de imágenes reales podría permitir la inclusión de diferentes casos y situaciones mediante la utilización de diversas líneas celulares, concentraciones de azul de tripán, tipos de hemocitómetros o celdas de conteo y diferentes etapas de crecimiento celular entre otras. En definitiva, una mayor cantidad y variedad de datos posibilita el aprendizaje a partir de una mayor cantidad y variedad de casos, lo que mejoraría el desempeño del modelo y permitiría su aplicación en diferentes situaciones. Otras ventajas del aumento de imágenes reales incluyen pero no se limitan a:

- **Mejoras en la clasificación:** Una mayor cantidad de imágenes anotadas mejoraría la precisión de la red en los casos confusos de clasificar. Dado que la utilización de imágenes reales en lugar de imágenes sintéticas permite el aprendizaje por parte del modelo de información acerca del contexto real de los objetos, un entrenamiento con imágenes reales podría ayudar al modelo a diferenciar células vivas de muertas y células de otros objetos (partículas, desechos, burbujas de aire, etc.).
- **Mejoras en la localización:** Dado que el contexto de una célula en una imagen presenta información útil para llevar a cabo su localización, utilizar imágenes reales mejoraría el desempeño del modelo en la predicción de los *bounding boxes* permitiéndole alcanzar mayores niveles de IoU.
- **Mayor robustez del modelo:** El simple hecho de utilizar más imágenes supone una mayor cantidad y variedad de situaciones. Entonces, si el conjunto posee un mayor número de imágenes, también presentará más células de diferentes tamaños, formas y líneas celulares e imágenes con diferentes tonos, brillos, tamaños y aumentos. Indefectiblemente, el modelo entrenado con mayor diversidad en la información será más robusto.
- **Eliminación de detecciones múltiples:** Un conjunto de imágenes más amplio permitiría una mejor discriminación entre las clases y evitaría que una célula sea clasificada simultáneamente como viva y muerta.
- **Ampliación del conjunto de prueba:** Un aumento en la cantidad de imágenes reales

también podría significar la construcción de un conjunto de prueba más amplio que permitirá determinar con más detalle la performance del modelo.

Por otra parte, un mayor número de imágenes reales podría ayudar a potenciar el modelo incluyendo otras funcionalidades como por ejemplo:

- **Detección de artefactos:** Junto con el aumento de imágenes reales, aumentan los ejemplos de artefactos en las mismas (partículas, residuos, burbujas de aire, etc.). El aprendizaje de estos objetos podría mejorar la detección en los casos confusos y nos proporcionaría información sobre la calidad del proceso de subcultivo y de la salud del cultivo.
- **Detección de aglomerados celulares:** Dada la mejora anterior, el aprendizaje de los artefactos de tipo “*Clump*” proporcionaría una mejora para los casos de detección de células cuya distancia sea mínima.
- **Detección a distintas concentraciones de Azul de Tripán:** La incorporación de imágenes con distintas concentraciones de azul de tripán, o sin tinción, le permitirían a la red aprender los distintos contextos en el que la célula puede encontrarse y permitiría una detección precisa en los distintos casos en el que un conteo puede llevarse a cabo. Generalmente, el uso de diferentes concentraciones de azul de tripán provoca variaciones en el tono de azul del fondo de la imagen y que las células presenten un contraste diferente con el fondo.
- **Detección a diferentes aumentos ópticos:** La incorporación de distintos aumentos ópticos y la posibilidad de especificar el aumento al momento del cálculo de dilución permitiría personalizar el proceso de conteo aumentando o disminuyendo el área de conteo permitiendo generar conteos más generales sobre la muestra o evitar que algún aspecto de la muestra interfiera con el conteo. Por otro lado, esta incorporación podría facilitar el análisis de células de distintos diámetros.

A partir de la información obtenida para el desarrollo del proyecto se procedió a construir el conjunto de imágenes que luego serían usados como datos de entradas para el aprendizaje de la red. Este procedimiento toma, por separado, las células, fondos y artefactos de las células originales y genera imágenes aleatorias a partir de modificaciones que se aplican a dichos objetos.

Un mayor entendimiento de las células en el contexto de subcultivo proporcionaría información valiosa para incrementar la performance del código generador de imágenes. Asimismo, un mejor entendimiento de como varía el desempeño de la red de acuerdo a las características de las imágenes sintéticas usadas durante el entrenamiento permitirían encontrar los parámetros óptimos usados en la síntesis de imágenes. En este proyecto, dicho trabajo se llevo a acabo mediante estudio de ablación de ciertos parámetros de la generación de imágenes. Entonces, si bien el estudio realizado intentó ser lo más exhaustivo posible, sería conveniente completar dicho estudio ampliando las combinaciones en las variaciones de:

- Brillo, tono y nitidez de las células
- Brillo, tono y nitidez de los fondos
- Tamaño de las células
- Cantidad de células por imagen
- Cantidad de imágenes por conjunto
- Cantidad de imágenes negativas

Finalmente, con el objetivo de encontrar mejores soluciones al momento de detectar y clasificar células en cámara de neubauer con tinción por azul de tripán, sería necesario llevar a cabo estudios más profundos que apunten a encontrar el mejor modelo o red neuronal. Algunas opciones a analizar podrían ser:

- Estudio de las variaciones del desempeño de YOLOv4 al introducir variaciones en su estructura
- Estudio de las variaciones del desempeño de YOLOv4 al introducir variaciones en sus parámetros e hiperparámetros
- Análisis y comparación de otras redes neuronales y su desempeño

6.2. Dispositivo

Si bien el objetivo de este proyecto fue lograr el diseño y construcción de un contador automático de células, lo logrado es susceptible de mejoras y puede también servir de punto de partida para desarrollos de otro tipo de dispositivos. Entre las oportunidades de mejoras relacionadas al dispositivo contador de células se identificaron:

- **Autofoco:** La inclusión de un sistema de autofocus permitiría la automatización completa del dispositivo al eliminar la necesidad de que el operador tenga que enfocar manualmente la muestra observada. El autofocus puede ser logrado mediante la adición de uno o dos motores que accionen la perilla de enfoque y muevan el conjunto cámara-espaciador-lente objetivo. El sistema de autofocus requiere, sin embargo, el desarrollo de algún algoritmo que controle el movimiento de los motores hasta lograr que la imagen este perfectamente enfocada en la muestra.
- **Mayor volumen de medición:** Si bien el contador diseñado utiliza un volumen de $0.3\ \mu\text{l}$ que se encuentra en el rango de los volúmenes utilizados por la mayoría de los contadores automáticos del mercado (Tabla 1) y los utilizados en el conteo manual, incrementar el volumen de medición permite la obtención de resultados más confiables y estadísticamente representativos del cultivo celular. Para incrementar el volumen de medición podría incorporarse una plataforma de desplazamiento automática en el plano xy de la muestra o bien disminuir el aumento de la cámara (agrandando el campo de visión). En el primer caso, sería necesario coordinar, mediante un código, el desplazamiento de la muestra, la captura de imágenes de diferentes áreas del hemocitómetro y el conteo en cada una de esas imágenes, teniendo en cuenta el solapamiento de las imágenes. Además, habría que llevar a cabo una adaptación del diseño de la estructura del dispositivo para que la misma pueda albergar en su superficie la plataforma xy, construyéndose una de muy bajo costo mediante componentes obtenidos de cualquier lector de CD o DVD como se muestra en [93]. En el segundo caso debería tenerse en cuenta la resolución mínima necesaria para que el modelo de detección pueda efectivamente localizar y clasificar las células de la imagen.
- **Otros métodos de tinción:** A pesar de ser el más usado debido a su simplicidad y accesibilidad, el ensayo de exclusión con azul de tripán no es el único método usado para

realizar conteos y ensayos de viabilidad de los cultivos, el dispositivo podría ser mejorado sustituyendo o incluyendo otros métodos de tinción para realizar el procedimiento. Particularmente, muchos contadores realizan conteos con fluorescencia donde diferentes tintes fluorescentes (fluorocromos) se unen a los ácidos nucleicos de las células y emiten luz en una frecuencia específica luego de absorber luz en otra longitud de onda. En este caso, sería necesario adaptar el dispositivo incorporando una fuente de iluminación de una frecuencia determinada para poder excitar los fluorocromos. Además, como para poder realizar un análisis de viabilidad (contando células vivas y muertas) se requieren dos tintes diferentes con frecuencias de excitación diferentes, sería necesario incluir dos fuentes de iluminación de diferentes colores. En todo caso, esto podría ser realizado con LEDs de diferentes colores o un solo LED RGB controlado por el procesador. Por otra parte, por cada método de conteo incluido sería necesario llevar a cabo la construcción del *dataset* y el entrenamiento de un modelo.

- **Conteo en tiempo real:** Aprovechando las características de YOLO en cuanto a su rapidez, sería posible realizar conteos en tiempo real, permitiendo al operador visualizar resultados instantáneos y de analizar varias muestras en poco tiempo.
- **Abaratamiento de costos de construcción:** La posibilidad de abaratar costos es una mejora factible que requiere un mayor análisis de las opciones disponibles para cada componente del dispositivo. Cambiando, por ejemplo, la placa por un procesador más económico, más un acelerador de hardware que permita alcanzar rendimientos compatibles con la inteligencia artificial.
- **Construcción de un dispositivo *stand-alone*:** Si se incluyese al dispositivo un display o pantalla compatible con el procesador sería posible lograr un dispositivo que, además de ser económico y portátil, sea independiente de cualquier tipo de *hardware* extra (monitores, teclados, laptops, etc.).

Finalmente, se identificó que parte del proceso llevado a cabo en la construcción de este dispositivo podría ser imitado para el diseño de otros aparatos con otros propósitos como:

- **Microscopio de bajo costo y portátil:** Utilizando las mismas herramientas de la par-

te óptica y estructurales, un microscopio sencillo y de bajo costo podría ser construido. Sus usos podrían variar entre la observación en tiempo real de células vivas, la obtención de videos de largas duraciones para analizar la evolución temporal de una muestra determinada y la realización de ensayos de fluorescencia.

- **Dispositivo de microscopia compatibles con incubadoras:** Teniendo en cuenta las condiciones de una incubadora (tamaño, flujo de aire, temperatura y humedad) diferentes dispositivos de microscopia podrían ser diseñados para la observación de células vivas dentro de la incubadora. Sus usos pueden ser el seguimiento de cultivos y monitoreo de su salud, seguimiento de ensayos de herida, monitoreo de colonias, analizadores de confluencia, entre otros.

7. Agradecimientos

Durante el desarrollo del proyecto recibimos la ayuda y el apoyo de algunas personas. En primer lugar queremos agradecer a nuestros tutores Giuliana Antonella Espósito y Carlos Selmo quienes nos guiaron en este camino y nos ayudaron en sus respectivos campos de experiencia. En segundo lugar, agradecer a Norberto Marcelo Lerendegui por proponer este proyecto y al ITBA por financiarlo ya que el equipo será instalado en el laboratorio de Bioingeniería con el fin de que los alumnos puedan utilizarlo y continuar desarrollando funciones sobre el sistema. En tercer lugar, queremos agradecer a Federico Paschetta, Gabriela Palloti y Fabricio Ballarini por brindarnos su apoyo y ayudarnos en la adquisición de los componentes necesarios para la realización del proyecto.

Por ultimo queremos agradecer a Joaquín Smucler por la obtención de imágenes de células en cámaras de conteo, a Maximiliano Figueroa por ayudarnos con algunos aspectos de la instalación de la placa utilizada, a Alejandro Copati por ayudarnos con las impresiones 3D de las partes del dispositivo y a Alexey Bochkovskiy quien nos aclaró dudas sobre el funcionamiento del repositorio *Darknet* y conceptos particulares de YOLOv4.

Referencias

- [1] Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016). You only look once: Unified, real-time object detection, En *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- [2] Proietti, A. & Volman, U. (2021). Trypan Blue stained Cells Image Dataset. Figshare. <https://doi.org/10.6084/m9.figshare.14818080.v1>
- [3] Invitrogen, G. (2014). Cell culture basics. *Life technologies*.
- [4] Freshney, R. I. (2015). *Culture of animal cells: a manual of basic technique and specialized applications*. John Wiley & Sons.
- [5] Philippeos, C., Hughes, R. D., Dhawan, A. & Mitry, R. R. (2012). Introduction to cell culture, En *Human Cell Culture Protocols*. Springer.
- [6] ALDRICH, S. (s.f.). Fundamental techniques in cell culture: A Laboratory Handbook, 2003.
- [7] Arora, M. (2013). Cell culture media: a review. *Mater Methods*, 3(175), 24.
- [8] Piccinini, F., Tesei, A., Paganelli, G., Zoli, W. & Bevilacqua, A. (2014). Improving reliability of live/dead cell counting through automated image mosaicing. *Computer methods and programs in biomedicine*, 117(3), 448-463.
- [9] Louis, K. S. & Siegel, A. C. (2011). Cell viability analysis using trypan blue: manual and automated methods, En *Mammalian cell viability*. Springer.
- [10] Strober, W. (2015). Trypan blue exclusion test of cell viability. *Current protocols in immunology*, 111(1), A3-B.
- [11] Chen, A., Leith, M., Tu, R., Tahim, G., Sudra, A. & Bhargava, S. (2017). Effects of diluents on cell culture viability measured by automated cell counter. *PloS one*, 12(3).
- [12] Stoddart, M. J. (2011). Cell viability assays: introduction, En *Mammalian cell viability*. Springer.
- [13] Kim, S. I., Kim, H. J., Lee, H.-J., Lee, K., Hong, D., Lim, H., Cho, K., Jung, N. & Yi, Y. W. (2016). Application of a non-hazardous vital dye for cell counting with automated cell counters. *Analytical biochemistry*, 492, 8-12.

- [14] Cadena-Herrera, D., Esparza-De Lara, J. E., Ramírez-Ibañez, N. D., López-Morales, C. A., Pérez, N. O., Flores-Ortiz, L. F. & Medina-Rivero, E. (2015). Validation of three viable-cell counting methods: Manual, semi-automated, and automated. *Biotechnology Reports*, 7, 9-16.
- [15] Riss, T., Niles, A., Moravec, R., Karassina, N. & Vidugiriene, J. (2019). Cytotoxicity Assays: In Vitro Methods to Measure Dead Cells, En *Assay Guidance Manual [Internet]*. Eli Lilly & Company; the National Center for Advancing Translational Sciences.
- [16] Piccinini, F., Tesei, A., Arienti, C. & Bevilacqua, A. (2017). Cell counting and viability assessment of 2d and 3d cell cultures: expected reliability of the trypan blue assay. *Biological procedures online*, 19(1), 8.
- [17] Maind, S. B., Wankar, P. Et al. (2014). Research paper on basic of artificial neural network. *International Journal on Recent and Innovation Trends in Computing and Communication*, 2(1), 96-100.
- [18] Krogh, A. (2008). What are artificial neural networks? *Nature biotechnology*, 26(2), 195-197.
- [19] Hagan, M. T., Demuth, H. B. & Jesús, O. D. (2002). An introduction to the use of neural networks in control systems. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, 12(11), 959-985.
- [20] Wang, S.-C. (2003). Artificial neural network, En *Interdisciplinary computing in java programming*. Springer.
- [21] Khan, A. I. & Al-Habsi, S. (2020). Machine learning in computer vision. *Procedia Computer Science*, 167, 1444-1451.
- [22] Sharma, P. (2019). Image Classification vs. Object Detection vs. Image Segmentation. <https://medium.com/analytics-vidhya/image-classification-vs-object-detection-vs-image-segmentation-f36db85fe81>
- [23] Brownlee, J. (2019). A gentle introduction to object recognition with deep learning. *Machine Learning Mastery*, 5. <https://machinelearningmastery.com/object-recognition-with-deep-learning/>

- [24] Chablani, M. (2017). YOLO - You only look once, real time object detection explained. *Towards Data Science*. <https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006>
- [25] Hui, J. (2018a). Real-time object detection with YOLO, YOLOv2 and now YOLOv3. *Medium*. <https://jonathan-hui.medium.com/real-time-object-detection-with-yolo-yolov2-28b1b93e2088>
- [26] Kamal, A. (2019). YOLO, YOLOv2 and YOLOv3: All You want to know. *Medium*. https://medium.com/@amrokamal_47691/yolo-yolov2-and-yolov3-all-you-want-to-know-7e3e92dc4899
- [27] Ian Goodfellow, A. C., Yoshua Bengio. (2016). *Deep Learning* (Final Edition). The MIT Press.
- [28] Rawat, W. & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, 29(9), 2352-2449.
- [29] LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- [30] Redmon, J. & Farhadi, A. (2017). YOLO9000: better, faster, stronger, En *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- [31] Redmon, J. (2013–2016). Darknet: Open Source Neural Networks in C.
- [32] Redmon, J. & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- [33] Bochkovskiy, A., Wang, C.-Y. & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv preprint arXiv:2004.10934*.
- [34] Hui, J. (2020). YOLOv4. *Medium*. <https://jonathan-hui.medium.com/yolov4-c9901eaa8e61>
- [35] Kanjee, R. (2020). YOLOv4 - Superior, Faster & More Accurate Object Detection. *Medium*. <https://medium.com/@riteshkanjee/yolov4-superior-faster-more-accurate-object-detection-7e8194bf1872>
- [36] Solawetz, J. (2020). Breaking Down YOLOv4. *Roboflow*. <https://blog.roboflow.com/a-thorough-breakdown-of-yolov4/>
- [37] Girshick, R., Donahue, J., Darrell, T. & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation, En *Proceedings of the IEEE conference on computer vision and pattern recognition*.

- [38] Girshick, R. (2015). Fast r-cnn, En *Proceedings of the IEEE international conference on computer vision*.
- [39] Ren, S., He, K., Girshick, R. & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks, En *Advances in neural information processing systems*.
- [40] Dai, J., Li, Y., He, K. & Sun, J. (2016). R-fcn: Object detection via region-based fully convolutional networks, En *Advances in neural information processing systems*.
- [41] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. & Berg, A. C. (2016). Ssd: Single shot multibox detector, En *European conference on computer vision*. Springer.
- [42] He, K., Zhang, X., Ren, S. & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9), 1904-1916.
- [43] Shah, D., Naciri, M., Clee, P. & Al-Rubeai, M. (2006). NucleoCounter—an efficient technique for the determination of cell number and viability in animal cell culture processes. *Cytotechnology*, 51(1), 39-44.
- [44] Chemometec. (2016). NucleoCounter[®] NC-100[™]. https://chemometec.com/Downloads/Brochures/NC100/990-0002_NC-100%20Brochure_US_web.pdf
- [45] Chemometec. (2018). The NucleoCounter[®] NC-200[™]. <https://chemometec.com/cell-counters/cell-counter-nc-200-nucleocounter/#1511446059049-3250ad53-0078>
- [46] Chemometec. (2020). NucleoCounter[®] NC-202[™]. https://chemometec.com/wp-content/uploads/2019/04/NC-202-Brochure_280220_ml_DIGITAL.pdf
- [47] DeNovix. (2020). CellDrop[®] Count Cells without slides. https://www.denovix.com/DeNovix_CellDrop_eBrochure.pdf
- [48] Corning. (2019). Corning[®] Cell Counter. <https://www.corning.com/catalog/cls/documents/sell-sheets/CLS-EQ-129.pdf>
- [49] Biosystems, L. (2014a). Luna[™] Automated Cell Counter. <https://logosbio.com/download/luna?wpdmdl=1215&refresh=5e93c5e0eadcc1586742752>
- [50] Biosystems, L. (2014b). Luna fl[™] The Ultimate Automated Cell Counter. <https://logosbio.com/download/luna-fl?wpdmdl=1273&refresh=5e93a37ec89141586733950>

- [51] Biosystems, L. (2015). Luna II™ Automated Cell Counter. <https://logosbio.com/download/luna-ii?wpdmdl=1236&refresh=5e93a37b2a29c1586733947>
- [52] Luminex. (2019). Guava® Muse® Cell Analyzer. <https://www.luminexcorp.com/wp-content/uploads/2019/05/BR168289.FlowCyt.Muse.WR..pdf>
- [53] Inc., B.-R. L. (s.f.). TC20™ Automatec Cell Counter. https://www.bio-rad.com/webroot/web/pdf/lsr/literature/Bulletin_6282.pdf
- [54] Nexcelom. (2010). Cellometer® Auto T4 Automated Bright Field Cell Counter. <https://www.nexcelom.com/Literature/Cellometer-Auto-T4-automated-cell-counter.pdf>
- [55] Bio, B. (2008). ADAM A New Standard of Automatic Cell Counter. https://www.bulldog-bio.com/wp-content/uploads/2019/05/adam_MC_brochure.pdf
- [56] NanoEnTek. (2018). ADAM MC2 A New Standard of Automatic Cell Counter. http://www.nanoentek.com/theme/nanont2_en/shop/02/product01_view.php?it_id=1547538805#detail_spec
- [57] Olympus. (2015). Cell Culture Essentials. https://www.olympus-lifescience.com/en/downloads/download/?file=855639643&fl=en_US&inline
- [58] Scientific, T. F. (2015). Countess II Automated Cell Counters. <http://assets.thermofisher.com/TFS-Assets/LSG/brochures/countess-II-automated-cell-counters-brochure.pdf>
- [59] Millipore. (2017). Scepter™ 2.0 Cell Counter. <https://www.sigmaaldrich.com/technical-documents/articles/labware/scepter-handheld-automated-cell-counter.html>
- [60] Hsiung, F., McCollum, T., Hefner, E. & Rubio, T. (2013). Comparison of Count Reproducibility, Accuracy, and Time to Results between a Hemocytometer and the TC20 Automated Cell Counter. *Bulletin 6003 Rev B*, 1-4.
- [61] Barghshoon, S. (2009). Cell Counting Survey. Millipore.
- [62] Purwar, P., Han, S., Lee, Y., Saha, B., Sandhan, T. & Lee, J. (2019). High-resolution cost-effective compact portable inverted light microscope. *Journal of microscopy*, 273(3), 199-209.
- [63] Sharkey, J. P., Foo, D. C., Kabla, A., Baumberg, J. J. & Bowman, R. W. (2016). A one-piece 3D printed flexure translation stage for open-source microscopy. *Review of Scientific Instruments*, 87(2), 025104.

- [64] Collins, J. T., Knapper, J., Stirling, J., Mduda, J., Mkindi, C., Mayagaya, V., Mwakajinga, G. A., Nyakyi, P. T., Sanga, V. L., Carbery, D. Et al. (2020). Robotic microscopy for everyone: the OpenFlexure Microscope. *Biomedical Optics Express*, 11(5), 2447-2460.
- [65] Aidukas, T., Eckert, R., Harvey, A. R., Waller, L. & Konda, P. C. (2019). Low-cost, sub-micron resolution, wide-field computational microscopy using opensource hardware. *Scientific reports*, 9(1), 1-12.
- [66] Walzik, M. P., Vollmar, V., Lachnit, T., Dietz, H., Haug, S., Bachmann, H., Fath, M., Aschenbrenner, D., Mofrad, S. A., Friedrich, O. Et al. (2015). A portable low-cost long-term live-cell imaging platform for biomedical research and education. *Biosensors and Bioelectronics*, 64, 639-649.
- [67] Cybulski, J. S., Clements, J. & Prakash, M. (2014). Foldscope: origami-based paper microscope. *PloS one*, 9(6), e98781.
- [68] Smith, Z. J., Chu, K., Espenson, A. R., Rahimzadeh, M., Gryshuk, A., Molinaro, M., Dwyre, D. M., Lane, S., Matthews, D. & Wachsmann-Hogiu, S. (2011). Cell-phone-based platform for biomedical device development and education applications. *PloS one*, 6(3), e17150.
- [69] Breslauer, D. N., Maamari, R. N., Switz, N. A., Lam, W. A. & Fletcher, D. A. (2009). Mobile phone based clinical microscopy for global health applications. *PloS one*, 4(7).
- [70] Grishagin, I. V. (2015). Automatic cell counting with ImageJ. *Analytical biochemistry*, 473, 63-65.
- [71] Collins, T. J. (2007). ImageJ for microscopy. *Biotechniques*, 43(S1), S25-S30.
- [72] Xue, Y. & Ray, N. (2017). Cell Detection in microscopy images with deep convolutional neural network and compressed sensing. *arXiv preprint arXiv:1708.03307*.
- [73] Hernández, C. X., Sultan, M. M. & Pande, V. S. (2018). Using deep learning for segmentation and counting within microscopy data. *arXiv preprint arXiv:1802.10548*.
- [74] Alzubaidi, L., Fadhel, M. A., Al-Shamma, O., Zhang, J. & Duan, Y. (2020). Deep Learning Models for Classification of Red Blood Cells in Microscopy Images to Aid in Sickle Cell Anemia Diagnosis. *Electronics*, 9(3), 427.
- [75] Tan, R.-x., Zhang, J., Chen, P., Wang, B. & Xia, Y. (2018). Cells counting with convolutional neural network, En *International Conference on Intelligent Computing*. Springer.

- [76] Xie, Y., Xing, F., Shi, X., Kong, X., Su, H. & Yang, L. (2018). Efficient and robust cell detection: A structured regression approach. *Medical image analysis*, 44, 245-254.
- [77] Xie, W., Noble, J. A. & Zisserman, A. (2018). Microscopy cell counting and detection with fully convolutional regression networks. *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization*, 6(3), 283-292.
- [78] Dharmawan, A. B., Scholz, G., Mariana, S., Hörmann, P., Ardiyanto, I., Wibirama, S., Hartmann, J., Prades, J. D., Hiller, K., Waag, A. Et al. (2018). Artificial neural networks for automated cell quantification in lensless LED imaging systems, En *Multidisciplinary Digital Publishing Institute Proceedings*.
- [79] Riba, J., Schoendube, J., Zimmermann, S., Koltay, P. & Zengerle, R. (2020). Single-cell dispensing and ‘real-time’ cell classification using convolutional neural networks for higher efficiency in single-cell cloning. *Scientific reports*, 10(1), 1-9.
- [80] Tscherepanow, M., Zöllner, F., Hillebrand, M. & Kummert, F. (2008). Automatic segmentation of unstained living cells in bright-field microscope images, En *International Conference on Mass Data Analysis of Images and Signals in Medicine, Biotechnology, and Chemistry*. Springer.
- [81] Ozkan, A., Isgor, S. B., Sengul, G. & Isgor, Y. G. (2018). Computer vision based automated cell counting pipeline: a case study for HL60 cancer cell on hemocytometer.
- [82] Manda-Handzlik, A., Fiok, K., Cieloch, A., Heropolitanska-Pliszka, E. & Demkow, U. (2020). Convolutional Neural Networks–Based Image Analysis for the Detection and Quantification of Neutrophil Extracellular Traps. *Cells*, 9(2), 508.
- [83] Hollows, G. & James, N. (2017). Imaging Resource Guide. *Edmund Optics Worldwide, Edmund Optics*.
- [84] Abramowitz, M. & Davidson, M. (2012). Basic Concepts in Optical Microscopy. *Microscopy Resource Center. Olympus*.
- [85] Spring, K. & Davidson, M. (2010). Basic Concepts and Formulas in Microscopy. *Microscopy U, The Source for Microscopy Education. Nikon*.
- [86] Sony. (2019). IMX219PQH5-C.
- [87] Ray, S. (1999). *Scientific photography and applied imaging*. CRC Press.

- [88] Greivenkamp, J. E. (2004). *Field guide to geometrical optics* (Vol. 1). SPIE Press Bellingham.
- [89] Peres, M. R. & Delly, J. G. (2007). Photomacrography and Close-Up Photograph, En *The Focal Encyclopedia of Photography*. Elsevier.
- [90] Switz, N. A., D'Ambrosio, M. V. & Fletcher, D. A. (2014). Low-cost mobile phone microscopy with a reversed mobile phone camera lens. *PloS one*, 9(5), e95330.
- [91] Dong, S., Guo, K., Nanda, P., Shiradkar, R. & Zheng, G. (2014). FPscope: a field-portable high-resolution microscope using a cellphone lens. *Biomedical optics express*, 5(10), 3305-3310.
- [92] Kim, S. B., Koo, K.-i., Bae, H., Dokmeci, M. R., Hamilton, G. A., Bahinski, A., Kim, S. M., Ingber, D. E. & Khademhosseini, A. (2012). A mini-microscope for in situ monitoring of cells. *Lab on a Chip*, 12(20), 3976-3982.
- [93] Schaefer, S., Boehm, S. A. & Chau, K. J. (2012). Automated, portable, low-cost bright-field and fluorescence microscope with autofocus and autoscanning capabilities. *Applied optics*, 51(14), 2581-2588.
- [94] Ward, D., Moghadam, P. & Hudson, N. (2018). Deep leaf segmentation using synthetic data. *arXiv preprint arXiv:1807.10931*.
- [95] Toda, Y., Okura, F., Ito, J., Okada, S., Kinoshita, T., Tsuji, H. & Saisho, D. (2020). Training instance segmentation neural network with synthetic datasets for crop seed phenotyping. *Communications biology*, 3(1), 1-12.
- [96] Chan, L. L.-Y., Rice, W. L. & Qiu, J. (2020). Observation and quantification of the morphological effect of trypan blue rupturing dead or dying cells. *Plos one*, 15(1), e0227950.
- [97] Bishop, C. M. (2006). Pattern recognition. *Machine learning*, 128(9).
- [98] Rosebrock, A. (2017). *Deep Learning for Computer Vision with Python: Practitioner Bundle*. PyImageSearch.
- [99] Huilgol, P. (2019). Accuracy vs. F1-Score. *Medium*. <https://medium.com/analytics-vidhya/accuracy-vs-f1-score-6258237beca2>
- [100] Hui, J. (2018b). mAP (mean Average Precision) for Object Detection. https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173

- [101] Aoullay, A. (2018). What's WRONG with Metrics? *Towards Data Science*. <https://towardsdatascience.com/choosing-the-right-metric-is-a-huge-issue-99ccbe73de61>
- [102] Tan, R. J. (2019). Breaking down mean average precision (map). *Towards Data Science*. <https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52>
- [103] Everingham, M., Van Gool, L., Williams, C. K., Winn, J. & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2), 303-338.
- [104] Rottenfusser, R., Wilson, E. E. & Davidson, M. W. (2014). Education in Microscopy and Digital Imaging. *ZEISS Microscopy*.
- [105] Arecchi, A. V., Koshel, R. J. & Messadi, T. (2007). Field guide to illumination. SPIE Press, Bellingham.
- [106] Tkaczyk, T. S. (2010). Field guide to microscopy. SPIE Press, Bellingham.
- [107] Sellaro, T. L., Filkins, R., Hoffman, C., Fine, J. L., Ho, J., Parwani, A. V., Pantanowitz, L. & Montalto, M. (2013). Relationship between magnification and resolution in digital pathology systems. *Journal of pathology informatics*, 4.
- [108] Wilson, M. (2017). Collecting Light: The Importance of Numerical Aperture in Microscopy. *Leica Microsystems*.
- [109] Holik, A. (2001). Optical microscopy.
- [110] Zheng, G., Kolner, C. & Yang, C. (2011). Microscopy refocusing and dark-field imaging by using a simple LED array. *Optics letters*, 36(20), 3987-3989.
- [111] Zheng, G., Horstmeyer, R. & Yang, C. (2013). Wide-field, high-resolution Fourier ptychographic microscopy. *Nature photonics*, 7(9), 739-745.
- [112] Zheng, G., Ou, X. & Yang, C. (2014). 0.5 gigapixel microscopy using a flatbed scanner. *Biomedical optics express*, 5(1), 1-8.
- [113] Guertin, D. A. & Sabatini, D. M. (2001). Cell size control. *e LS*.
- [114] Russakovsky, O., Li, L.-J. & Fei-Fei, L. (2015). Best of both worlds: human-machine collaboration for object annotation, En *Proceedings of the IEEE conference on computer vision and pattern recognition*.