

**INSTITUTO TECNOLÓGICO DE BUENOS AIRES – ITBA**

**ESCUELA DE INGENIERÍA Y TECNOLOGÍA**

# **DETECCIÓN DE GLÓBULOS BLANCOS EN IMÁGENES DE MICROSCOPIA MEDIANTE EL USO DE REDES NEURONALES**

**AUTORA: Glaser, Karina Jennifer (Leg. N° 55364)**

**PROFESOR RESPONSABLE O TUTOR: Selmo, Carlos**

**TRABAJO FINAL PRESENTADO PARA LA OBTENCIÓN DEL TÍTULO DE BIOINGENIERA**

**BUENOS AIRES**

**SEGUNDO CUATRIMESTRE, 2020**

## Resumen

El recuento diferencial de glóbulos blancos es un estudio cuantitativo que informa la cantidad y/o el porcentaje de cada tipo de glóbulo blanco. Se realiza a partir de una extracción de sangre al igual que otros estudios diagnósticos que están basados en sus componentes celulares. La importancia de estos exámenes es que son indicadores fácilmente accesibles de alteraciones en los órganos de origen o degradación de estos componentes que son mucho menos accesibles.

Para la realización del recuento existen métodos manuales y automáticos. El método manual consta de la extensión de una gota de sangre en un portaobjetos que es coloreada mediante alguna tinción. Este paso es necesario para la visualización de diferentes estructuras mediante el uso del microscopio y así poder diferenciar entre los diversos elementos de la sangre y entre los mismos glóbulos blancos. Luego, es un técnico o un bioquímico el encargado de hacer el recuento y clasificación de forma manual del espécimen. Este método continúa hasta el día de hoy siendo el “gold standard” para este tipo de análisis. Sin embargo, es una de las tareas más intensas en mano de obra en el laboratorio y que requiere de personal altamente entrenado. Otro de los inconvenientes de esta técnica es la limitación en términos de espacio físico para el almacenamiento de los portaobjetos y la dificultad para realizar una interconsulta sobre los resultados hallados.

Debido a estos problemas es que surgen los métodos automáticos. Entre los mismos se encuentra la clasificación asistida por computadora de los glóbulos blancos en imágenes de microscopía de sangre periférica. Para eso el sistema debe realizar dos tareas sobre la imagen: la localización del glóbulo blanco y la posterior clasificación del mismo. Estas tareas se pueden englobar dentro de lo que se denomina detección de objetos, siendo los glóbulos blancos los objetos en este caso.

Existen diferentes algoritmos de detección. En particular, se destaca YOLO (You Only Look Once) que se basa en las redes neuronales convolucionales para realizar esta tarea. La presente tesis se basa en tomar diferentes características de YOLO como, por ejemplo, elementos de su arquitectura e hiperparámetros para poder realizar la detección. Se proponen dos modelos sobre los cuales se compara su rendimiento. Para la construcción y entrenamiento de estos se utilizó Python y Keras (con Tensorflow como backend).

Como resultado, ambos modelos entrenados logran realizar la tarea propuesta consiguiendo un  $F_2$  score mayor a 0.93 y 0.87 para la localización y detección, respectivamente, tanto para el conjunto de validación como el de evaluación.

## Agradecimientos

En primer lugar, quería agradecer a Carlos Selmo, mi tutor en esta tesis, por su acompañamiento y continuo soporte y enseñanza. También agradecer a Alberto Tablón por su consejo y apoyo al principio de esta tesis.

Por otro lado, quiero mencionar a Yanina Glaser y Manuela Alanis por ayudarme en el final de esta tesis a revisarla y su aliento.

Al mismo tiempo, reconocer a mis amigos que fueron el sostén durante toda la carrera y que me ayudaron a crecer.

Por último, agradecer a mi familia que siempre me apoyó incondicionalmente, me escuchó cuando más lo necesitaba y me brindó su paciencia.

## Índice

Resumen .....	1
Agradecimientos .....	2
Ilustraciones .....	5
Tablas .....	8
Introducción .....	9
El Sistema circulatorio y su funcionamiento .....	9
Componentes de la sangre .....	10
Plasma .....	10
Elementos corpusculares .....	10
Glóbulos rojos .....	11
Plaquetas .....	12
Glóbulos blancos .....	12
Estructura y funcionamiento .....	13
Clasificación de leucocitos .....	13
Recuento diferencial de glóbulos blancos .....	16
Métodos manuales .....	16
Método de conteo .....	18
Métodos automáticos .....	19
Citometría de flujo .....	20
Sistema de fluidos .....	20
Sistema óptico .....	21
Sistema computacional/electrónico .....	22
Sistemas de análisis morfológicos automatizados .....	22
Detección de objetos .....	24
Filtros y convolución .....	24
Historia sobre la detección de objetos .....	25
Elementos de algoritmos de aprendizaje profundo .....	27
Modelo .....	27
Función de costo .....	29
Optimización .....	30
Algoritmos de optimización .....	31
Descenso por gradiente en lote (Batch Gradient Descent) .....	31
Descenso por gradiente estocástico (Stochastic Gradient Descent (SGD)) .....	31
Descenso por gradiente en mini lote (Mini Batch Gradient Descent) .....	31
Desafíos de los algoritmos de descenso por gradiente .....	31

Variaciones de los algoritmos de descenso por gradiente .....	32
Algoritmos de detección de objetos .....	33
Evaluación de los algoritmos .....	34
Métricas para la evaluación .....	36
YOLO (You Only Look Once) .....	38
Capas de la red neuronal .....	39
Convolucionales .....	39
Activación.....	39
Batch Normalization .....	40
Max Pooling .....	40
Función de costo .....	41
Tasa de aprendizaje .....	42
Non Maximum Suppression.....	43
Datos y Pre-Procesamiento.....	44
Pre-Procesamiento.....	45
Herramientas .....	49
Modelos .....	49
Resultados.....	51
Discusión .....	72
Futuros trabajos y conclusiones .....	75
Bibliografía .....	78
Anexo A .....	81

## Ilustraciones

Ilustración 1 Microfotografía de frotis sanguíneo donde se observan los diferentes elementos corpusculares de la sangre.....	11
Ilustración 2 Imagen ilustrativa de neutrófilos rodeados de glóbulos rojos .....	14
Ilustración 3 Imagen ilustrativa de eosinófilos rodeados de glóbulos rojos.....	14
Ilustración 4 Imagen ilustrativa de basófilos rodeados de glóbulos rojos .....	14
Ilustración 5 Imagen ilustrativa de linfocitos rodeados de glóbulos rojos .....	15
Ilustración 6 Imagen ilustrativa de monocito rodeados de glóbulos rojos .....	15
Ilustración 7: Técnica en cuña para realizar el frotis de sangre periférica. A. Ángulo correcto al sostener el frotis extensor. B. La sangre se extiende en toda la superficie de contacto entre los portaobjetos. C. Frotis en cuña completado .....	16
Ilustración 8: Dibujo esquemático de un frotis sanguíneo con la indicación de la distribución de los glóbulos blancos .....	17
Ilustración 9: Ejemplo de frotis de sangre periférica bien realizado .....	17
Ilustración 10: Ejemplos de frotis mal realizados .....	17
Ilustración 11: Ejemplo de frotis mal extendido. A. Se muestran muchos leucocitos en la cola y B. muy pocos leucocitos en el cuerpo del frotis .....	18
Ilustración 12: Área correcta del frotis de sangre para realizar el procedimiento. ....	18
Ilustración 13: Ejemplo de distribución de las franjas longitudinales para realizar el recuento. ....	18
Ilustración 14: Ejemplo de microfotografía de frotis sanguíneo de paciente con leucemia. ....	19
Ilustración 15: Esquema del sistema de fluidos con enfoque hidrodinámico. La forma de la celda de flujo (flow cell) ayuda a minimizar la turbulencia mientras que asegura que la muestra se encuentre enfocada en el centro de la corriente para la presentación al laser .....	21
Ilustración 16: Ejemplo del sistema óptico del citómetro con los diferentes láser y filtros .....	21
Ilustración 17: Diagrama esquemático de la diferenciación de los leucocitos en sangre periférica en base a la dispersión frontal de la luz y la dispersión en ángulo recto .....	22
Ilustración 18: Imágenes de analizadores de CellaVision™ y la preclasificación del software de las imágenes. Se muestran el modelo DM96 (arriba) y DC-1 (abajo) que es una versión más moderna..	23
Ilustración 19 Ejemplo de la operación de convolución sobre una imagen para la detección de bordes horizontales y verticales .....	24
Ilustración 20 Flujo de trabajo tradicional de visión por computadora (a) vs. aprendizaje profundo (b) .....	26
Ilustración 21 Esquema de una neurona y sus conexiones .....	28
Ilustración 22 Esquema de funcionamiento de una neurona artificial con sus componentes.....	29
Ilustración 23: Gráfico ilustrando el cambio de los pesos hacia el mínimo global durante el proceso de optimización.....	30
Ilustración 24: Ilustración de puntos silla en una función. Se puede observar que en la dimensión entre A y B el punto es un mínimo, mientras que entre C y D es un máximo. ....	32
Ilustración 25: Esquema representativo de la estructura de Faster RCNN como ejemplo de detectores de dos etapas .....	33
Ilustración 26: Esquema representativo de la estructura de YOLO como ejemplo de detector unificado .....	33
Ilustración 27: Representación de subajuste y sobreajuste en modelos de clasificación. ....	35
Ilustración 28: Gráfico del error en función de la capacidad durante el entrenamiento y diferencia entre error de entrenamiento y error de generalización .....	35
Ilustración 29: Cálculo de Intersection over Union (IoU) entre dos formas rectangulares .....	36

Ilustración 30: Imagen ilustrativa del funcionamiento de YOLO respecto de una imagen de entrada para poder obtener las predicciones de los objetos. ....	38
Ilustración 31: Imagen ilustrativa del funcionamiento de una capa de max pooling de 2x2 y paso de 1 .....	41
Ilustración 32: Imagen ilustrativa de la variación de la tasa de aprendizaje en función de las iteraciones durante el Cyclical Learning Rate. El tiempo de crecimiento y decrecimiento está dado por el paso (step size) y la tasa fluctúa entre el mínimo y máximo. ....	43
Ilustración 33: Imagen ilustrativa de la predicción de una red neuronal que predice dos cajas delimitadoras superpuestas para el mismo objeto. Solo una caja válida es elegida y es la de mayor puntaje (score) que es 0.99 .....	44
Ilustración 34: Imágenes del conjunto de datos de cada clase. De izq. a der.: basófilo (imagen 0), eosinófilo (imagen 54), linfocito (imagen 110), neutrófilo (imagen 159) y monocito (imagen 207) ..	45
Ilustración 35: Ejemplo del método de espejado que se decidió emplear. (imagen 0 y su transformada) .....	46
Ilustración 36: Ejemplo de globulo blanco espurio producto de la transformación realizada sobre la imagen. (imagen 0 y su transformada) .....	47
Ilustración 37: Esquema general de los modelos de la presente tesis y de los parámetros de las diferentes capas.....	50
Ilustración 38: $F_2$ score de localización en función del umbral para cada combinación de Modelo y Variante para el conjunto de entrenamiento .....	52
Ilustración 39: $F_2$ score de detección en función del umbral para cada Modelo y Variante para el conjunto de entrenamiento.....	53
Ilustración 40: $F_2$ score de localización y de detección en función del umbral para el Modelo 1 Variante 1 para el conjunto de entrenamiento.....	54
Ilustración 41: $F_2$ score de localización en función del umbral para cada combinación de Modelo y Variante para el conjunto de validación .....	55
Ilustración 42: $F_2$ score de detección en función del umbral para cada combinación de Modelo y Variante para el conjunto de validación .....	56
Ilustración 43: Comparación de las curvas del $F_2$ score de localización en función del umbral entre el conjunto de entrenamiento y el de validación .....	57
Ilustración 44: Comparación de las curvas del $F_2$ score de detección en función del umbral entre el conjunto de entrenamiento y el de validación.....	58
Ilustración 45: $F_2$ score de localización y de detección en función del umbral para el Modelo 1 Variante 1 para el conjunto de validación.....	59
Ilustración 46: Resultados de la detección de ambos modelos para el total de las imágenes del conjunto de validación .....	60
Ilustración 47: Resultados de la detección de ambos modelos por clase para las imágenes del conjunto de validación .....	61
Ilustración 48: Resultados de la detección de ambos modelos agrupados por imagen original y clase del conjunto de validación.....	62
Ilustración 49: Imagen 46 del dataset original.....	63
Ilustración 50: De izq. a der. Imágenes 5 a 8 del conjunto original.....	63
Ilustración 51: Imagen 70 del conjunto de datos original .....	64
Ilustración 52: De izq a der. Imágenes 54, 56, 57, 59 del conjunto de imágenes originales .....	64
Ilustración 53: Ejemplo de imágenes aumentadas con glóbulo blanco espurio producto de la reflexión de la imagen original.....	65
Ilustración 54: Gráfico con la ubicación de los centros de las predicciones (o de las etiquetas según corresponda) para el conjunto de datos de validación .....	66

Ilustración 55: Gráfico con la ubicación específica de los falsos negativos por clase de glóbulo blanco para el conjunto de datos de validación.....	66
Ilustración 56: Gráfico con la distribución del nivel de confianza general por imagen original.....	67
Ilustración 57: Gráfico con la distribución del nivel de confianza de clase por imagen original.....	68
Ilustración 58: Gráfico con la distribución del nivel de confianza de objeto por imagen original.....	68
Ilustración 59: Gráfico con la distribución del nivel de confianza general, de clase y objeto por imagen original.....	69
Ilustración 60: Resultados de la detección de ambos modelos para el total de las imágenes del conjunto de evaluación.....	70
Ilustración 61: Resultados de la detección de ambos modelos por clase para las imágenes del conjunto de evaluación.....	70
Ilustración 62: Resultados de la detección de ambos modelos agrupados por imagen original y clase del conjunto de evaluación.....	71
Ilustración 63: Gráfico con la ubicación específica de los falsos negativos por clase de glóbulo blanco para el conjunto de datos de evaluación.....	72
Ilustración 64: Gráfico comparativo de los $F_2$ score para los tres conjuntos de datos para el Modelo 1.....	75
Ilustración 65: $F_2$ score de localización y de detección en función del umbral para el Modelo 1 Variante 2 para el conjunto de entrenamiento.....	81
Ilustración 66: $F_2$ score de localización y de detección en función del umbral para el Modelo 2 Variante 1 para el conjunto de entrenamiento.....	82
Ilustración 67: $F_2$ score de localización y de detección en función del umbral para el Modelo 2 Variante 2 para el conjunto de entrenamiento.....	83
Ilustración 68: Comparación de las curvas del $F_2$ score de localización en función del umbral entre el conjunto de entrenamiento y el de validación para el Modelo 1 Variante 2.....	84
Ilustración 69: Comparación de las curvas del $F_2$ score de localización en función del umbral entre el conjunto de entrenamiento y el de validación para el Modelo 2 Variante 1.....	85
Ilustración 70: Comparación de las curvas del $F_2$ score de localización en función del umbral entre el conjunto de entrenamiento y el de validación para el Modelo 2 Variante 2.....	86
Ilustración 71: Comparación de las curvas del $F_2$ score de detección en función del umbral entre el conjunto de entrenamiento y el de validación para el Modelo 1 Variante 2.....	87
Ilustración 72: Comparación de las curvas del $F_2$ score de detección en función del umbral entre el conjunto de entrenamiento y el de validación para el Modelo 2 Variante 1.....	88
Ilustración 73: Comparación de las curvas del $F_2$ score de detección en función del umbral entre el conjunto de entrenamiento y el de validación para el Modelo 2 Variante 2.....	89
Ilustración 74: $F_2$ score de localización y de detección en función del umbral para el Modelo 1 Variante 2 para el conjunto de validación.....	90
Ilustración 75: $F_2$ score de localización y de detección en función del umbral para el Modelo 2 Variante 1 para el conjunto de validación.....	91
Ilustración 76: $F_2$ score de localización y de detección en función del umbral para el Modelo 2 Variante 2 para el conjunto de validación.....	92

## Tablas

Tabla 1: Descripción de hiperparámetros correspondientes a cada modelo y variable .....	51
Tabla 2: Umbral óptimo para cada Modelo y Variante con sus valores de $F_2$ score de localización y de detección.....	59
Tabla 3: Matriz de confusión del Modelo 1 Variante 1 utilizando el conjunto de validación para el umbral óptimo de 0.0692 .....	60
Tabla 4: Matriz de confusión del Modelo 2 Variante 1 utilizando el conjunto de validación para el umbral óptimo de 0.0832 .....	60
Tabla 5: Comparativa entre los $F_2$ score de cada modelo para los diferentes conjuntos de datos.....	69
Tabla 6: Matriz de confusión del Modelo 1 Variante 1 utilizando el conjunto de evaluación para el umbral óptimo de 0.0692 .....	70
Tabla 7: Matriz de confusión del Modelo 2 Variante 1 utilizando el conjunto de evaluación para el umbral óptimo de 0.0832 .....	70
Tabla 8: Métricas de mAP y AP por clase para los modelos de la presente tesis y los expuestos en Wang et. al. ....	73

## Introducción

### El Sistema circulatorio y su funcionamiento

Los organismos unicelulares y algunos animales multicelulares más simples logran satisfacer sus necesidades de difusión de sustancias con el ambiente a través de las membranas de sus células. Sin embargo, a medida que los mismos evolucionaron en cuerpos más grandes, algunas células se encuentran demasiado lejos de la superficie como para lograr satisfacer dichas necesidades tan fácilmente. Para solucionar este problema, fueron desarrollando un sistema circulatorio para distribuir esas necesidades de diferentes sustancias a todas las células del cuerpo. El mismo consta del corazón, los vasos sanguíneos y la sangre en el caso de los humanos. En este complejo sistema, la sangre es el medio líquido que permite entre otras cosas, el transporte de dichas sustancias.

La sangre es el único tejido fluido en el cuerpo. Aunque parece ser un líquido espeso y homogéneo, bajo el microscopio se observa que tiene componentes tanto sólidos como líquidos. En concreto, es un tejido conectivo líquido compuesto de plasma y elementos corpusculares.

La mayoría de los adultos tienen entre 4 y 6 litros de sangre. Usualmente el volumen se define aproximadamente de la siguiente forma: 55% plasma y 45% elementos corpusculares. El primer elemento mencionado es un fluido amarillo claro que forma la matriz extracelular del tejido. Los elementos corpusculares son células y fragmentos celulares: glóbulos rojos, glóbulos blancos y plaquetas. Esta denominación se debe a que son entidades envueltas por membranas con una estructura específica; no pueden ser llamadas células, porque, por ejemplo, las plaquetas son simplemente fragmentos de ciertas células de médula ósea [1].

Algunas de las funciones de la sangre son las siguientes:

- Transporte:
  - Moviliza el oxígeno desde los pulmones al resto de los tejidos y también lleva el dióxido de carbono (producto de desecho de la respiración celular) en el sentido contrario para que sea eliminado del cuerpo.
  - Lleva los nutrientes desde el tracto digestivo a los diferentes tejidos.
  - Conduce los desechos metabólicos a los riñones para su desecho
  - Acarrea las hormonas desde las células endocrinas a las células blancas
  - Traslada las células madre a sus destinos
- Protección
  - Glóbulos blancos destruyen microorganismos nocivos y células cancerosas
  - Anticuerpos y otras proteínas sanguíneas neutralizan toxinas y ayudan con la destrucción de patógenos.
  - Juega un rol clave en los procesos inflamatorios, un mecanismo que permite limitar la propagación de infecciones y promover la reparación de tejido herido
  - Plaquetas y otros agentes inician la coagulación de la sangre y otros procesos para minimizar la pérdida de sangre.
- Regulación
  - Ayuda a mantener un equilibrio de los fluidos y la distribución en el cuerpo absorbiéndolos o liberándolos bajo diferentes circunstancias.
  - Proteínas sanguíneas ayudan a estabilizar el pH de los fluidos extracelulares siendo un buffer entre ácidos y bases.

- Cambios en el flujo sanguíneo ayudan a regular la temperatura corporal. Por ejemplo, cuando es necesaria la pérdida de calor la sangre va hacia la piel y cuando es necesario retenerlo ésta va a niveles más profundos en el cuerpo

En resumen, este tejido fluido juega numerosos roles para mantener homeostasis. Considerando sus funciones vitales, es fácil entender por qué una pérdida excesiva es fatal, y por qué los mecanismos del sistema circulatorio han evolucionado para minimizar dicha pérdida.

A continuación, se dará una mayor definición de los componentes de la sangre y sus funciones específicas.

## Componentes de la sangre

### Plasma

El plasma, como ya se estableció, es la parte más líquida y es aproximadamente 90% agua. En este fluido amarillento translúcido se encuentran más de 100 sustancias diluidas. Algunos ejemplos de estas incluyen nutrientes, electrolitos, gases respiratorios, hormonas, proteínas plasmáticas, y varios desechos y productos de metabolismo celular.

Las proteínas plasmáticas son los solutos más abundantes y el hígado genera la mayoría de estas proteínas, excepto por los anticuerpos y las hormonas peptídicas. Éstas tienen una gran variedad de funciones. Por ejemplo, la albúmina funciona como un transportador de ciertas moléculas a través del sistema circulatorio, es un buffer sanguíneo importante, y contribuye a la presión osmótica, lo que permite retener el agua en el flujo sanguíneo. Las proteínas de coagulación ayudan a detener el sangrado cuando un vaso sanguíneo se daña, y los anticuerpos contribuyen a proteger el cuerpo de los patógenos. Las proteínas plasmáticas no son utilizadas por las células como nutrientes metabólicos, a diferencia de otros solutos como la glucosa, ácidos grasos y oxígeno.

La composición del plasma varía continuamente a medida que las células intercambian sustancias con la sangre. Asumiendo una dieta sana, sin embargo, la composición del plasma se mantiene relativamente constante por varios mecanismos homeostáticos del cuerpo. Por ejemplo, cuando las proteínas sanguíneas disminuyen a niveles indeseados, el hígado es estimulado para generar más, y cuando la sangre comienza a ser muy ácida (acidosis) o muy básica (alcalosis), tanto el sistema respiratorio como urinario se activan para restaurar su estado normal que es un pH levemente alcalino entre 7.35 y 7.45. Varios órganos hacen una docena de ajustes por día para mantener muchos solutos plasmáticos en niveles aptos para la vida. Además de transportar varias sustancias a través del cuerpo, el plasma ayuda a distribuir el calor corporal, un subproducto del metabolismo celular, de forma pareja en todo el cuerpo [2].

### Elementos corpusculares

Dentro de los elementos corpusculares se encuentran los glóbulos rojos (eritrocitos), glóbulos blancos (leucocitos) y plaquetas.

Si se examina un frotis de sangre humana teñido bajo un microscopio óptico, se observa unos glóbulos rojos con forma de disco, una variedad de glóbulos blancos de forma esférica con tinciones llamativas, y algunas plaquetas dispersas [1]. También se podrá ver que los eritrocitos superan ampliamente en número a otros tipos de elementos corpusculares, como se puede ver en la siguiente imagen:

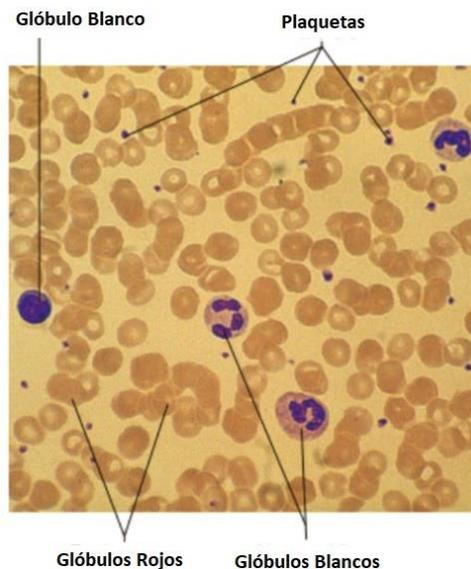


Ilustración 1: Microfotografía de frotis sanguíneo donde se observan los diferentes elementos corpusculares de la sangre [2]

## Glóbulos rojos

Los eritrocitos, o glóbulos rojos, tienen dos funciones principales:

- captar el oxígeno de los pulmones y transportarlo a otros tejidos corporales.
- captar el dióxido de carbono de los tejidos y descargarlo en los pulmones.

Estas células son un gran ejemplo de la relación entre estructura y función, con su forma discoidal con un borde grueso y un centro fino y hundido, representando la forma bicóncava distintiva. El principal beneficio de ésta es que permite que la suspensión espesa de glóbulos rojos fluya suavemente a través de los vasos sanguíneos con una turbulencia mínima y posibilita a los mismos curvarse y pasar a través de los vasos más pequeños.

Esta estructura es posible ya que durante su desarrollo pierden el núcleo, por lo tanto, los eritrocitos maduros son células anucleadas. Sin embargo, esta falta de núcleo y ADN los hace incapaces de mitosis y síntesis proteica, resultando en una incapacidad de auto regeneración [1]. También tienen pocas organelas y carecen de mitocondrias. Esto hace que la generación de ATP sea por medios anaeróbicos, o sea que no utilizan nada del oxígeno que transportan, lo que los convierte en transportadores muy eficientes.

Como si esto fuera poco, poseen una gran cantidad de hemoglobina, una proteína roja que contiene hierro, que transporta la mayor parte del oxígeno en la sangre. También se une con pequeñas cantidades de dióxido de carbono. La forma pequeña y peculiar del eritrocito, provee una gran superficie en relación con su volumen, haciéndolos ideales para el intercambio gaseoso.

Resultan los elementos corpusculares más abundantes. De hecho, cuando una muestra de sangre se ve al microscopio, los glóbulos rojos llenan el campo debido a que representan entre el 37% y el 52% del volumen sanguíneo en condiciones normales. Además, superan en número a los glóbulos blancos (que son más grandes) en aproximadamente 1000 a 1 y son el principal factor que contribuye a la viscosidad de este tejido. Aunque los números varían, normalmente hay cerca de 5 millones de células por milímetro cúbico de sangre [2]. Cuando el número por milímetro cúbico aumenta, la viscosidad también. De manera similar, cuando disminuye, la sangre se diluye y fluye más rápido.

Aunque el número de eritrocitos es importante, la cantidad de hemoglobina en el torrente sanguíneo en cualquier momento es lo que realmente determina que tan bien están desempeñando su rol de transportadores de oxígeno. Cuantas más moléculas de hemoglobina los glóbulos rojos contienen, más oxígeno van a ser capaces de movilizar. Un único eritrocito contiene aproximadamente 250

millones moléculas de hemoglobina, cada una capaz de unirse a 4 moléculas de oxígeno, así que cada una de estas pequeñas células puede transportar cerca de 1000 millones de moléculas de oxígeno. Sin embargo, técnicamente es mucho más importante el hecho de que la sangre contiene normalmente entre 12 y 18 gramos de hemoglobina por cada 100 mililitros.

### Plaquetas

Los trombocitos, o plaquetas, son pequeños fragmentos circulantes de células de médula ósea llamadas megacariocitos y cumple diversos roles en la coagulación, entre otros. Se encuentran dentro de los elementos corpusculares, pero no son células debido a que no poseen un núcleo propio en ningún momento de su desarrollo. Son el segundo elemento corpuscular más abundante (después de los eritrocitos) y su conteo normal es de 300.000 fragmentos por  $\text{mm}^3$  [2]. Sin embargo, contribuyen aún menos que los glóbulos blancos al volumen sanguíneo total debido a su pequeño tamaño.

Éstas tienen una gran variedad de funciones [1]:

1. Secretan vasoconstrictores, químicos que causan contracciones espasmódicas de vasos sanguíneos rotos y así ayudan a reducir la pérdida de sangre.
2. Se unen para formar tapones plaquetarios temporales y relativamente frágiles para sellar pequeñas roturas en los vasos sanguíneos lesionados
3. Secretan factores que promueven la coagulación.
4. Inician la formación de enzimas que disuelven coágulos sanguíneos que deshace aquellos que duraron más de lo necesario.
5. Secretan factores de crecimiento que estimulan la mitosis en fibroblastos y músculo liso y así ayudan a mantener y reparar los vasos sanguíneos.

### Glóbulos blancos

Aunque los leucocitos, o glóbulos blancos, son mucho menos numerosos que los glóbulos rojos, son cruciales en la defensa. En promedio, hay entre 4800 a 10800 por  $\text{mm}^3$  de sangre, y representan menos del 1% del volumen total [2]. Los glóbulos blancos contienen núcleos y las organelas usuales, lo que los hace las únicas células completas de la sangre.

Los leucocitos forman un ejército protector y móvil que ayuda a defender el cuerpo contra daño por bacterias, virus, parásitos y células tumorales. Como tal, tienen algunas características muy especiales. A diferencia de los glóbulos rojos que se encuentran confinados al torrente sanguíneo, los glóbulos blancos son capaces de entrar y salir de estos en un proceso denominado diapédesis. El sistema circulatorio es simplemente su medio de transporte a áreas del cuerpo donde sus servicios son requeridos para respuestas inmunes o inflamatorias.

Además, pueden localizar áreas de daño tisular e infección en el cuerpo respondiendo a ciertos químicos que se difunden desde las células dañadas. Esta capacidad se denomina quimiotaxis positiva. Una vez que “captan la esencia”, se mueven a través de los espacios de los tejidos mediante un movimiento ameboide (forman extensiones citoplasmáticas fluidas que los ayudan a moverse). Siguiendo el gradiente de difusión, localizan las áreas de daño tisular y afluyen en grandes cantidades para destruir los microorganismos y disponer de las células muertas.

Cuando se movilizan para la acción, el cuerpo aumenta la velocidad de producción de estos, y en pocas horas pueden aparecer hasta el doble de la cantidad normal. Un conteo total de glóbulos blancos mayor a 11000 células por  $\text{mm}^3$  se denomina leucocitosis [2]. Ésta es una respuesta protectora al combate de microorganismos. Por ejemplo, durante la etapa activa de la infección el recuento de leucocitos puede duplicarse, triplicarse o cuadruplicarse, de acuerdo con la gravedad de esta. Entre las enfermedades que pueden provocar este tipo de elevación del recuento de leucocitos figuran la meningitis, la mononucleosis infecciosa, la apendicitis, la neumonía neumocócica y la gonorrea [3].

La condición opuesta, leucopenia, es un número anormalmente bajo del conteo de glóbulos blancos. Es comúnmente causado por ciertas drogas, como corticosteroides y agentes anticancerígenos. Otras enfermedades, como la salmonelosis y la brucelosis, y algunas infecciones virales y por rickettsias también pueden causarla. La leucopenia puede relacionarse con un deterioro de la producción de leucocitos o con el efecto del aumento de la sensibilidad de las membranas de estas células al daño causado por el complemento, las proteínas plasmáticas antimicrobianas.

La leucocitosis es una respuesta normal y deseable frente a infecciones que amenazan al cuerpo. Por contraste, la producción excesiva de glóbulos blancos anormales que ocurre en la mononucleosis y la leucemia es distintivamente patológica. En la leucemia, la médula ósea se vuelve cancerígena, y un gran número de glóbulos blancos son producidos rápidamente. Aunque esto puede no parecer un problema, los nuevos glóbulos blancos son inmaduros e incapaces de llevar a cabo sus funciones protectoras normales. En consecuencia, el cuerpo se vuelve una presa fácil de aquellas bacterias y virus que causan enfermedades. Adicionalmente, como otras líneas celulares sanguíneas son desplazadas, se producen anemia severa y problemas de sangrado.

El aumento o la disminución de los leucocitos pueden detectarse mediante un recuento diferencial de glóbulos blancos, que es un cálculo del porcentaje de cada clase de célula en una muestra de 100 leucocitos.

#### *Estructura y funcionamiento*

Los leucocitos no son uniformes y varían en forma y función. En contraste a los eritrocitos, estos retienen sus organelas a través de toda su vida; aunque cuando son vistos a través de un microscopio electrónico de transmisión, estos muestran una compleja estructura interna. Entre sus organelas se encuentran los lisosomas y los usuales instrumentos para la síntesis de proteínas - el núcleo, retículo endoplasmático rugoso, ribosomas y aparato de Golgi - ya que los leucocitos deben sintetizar proteínas como enzimas lisosómicas para cumplir una variedad de funciones [1].

#### *Clasificación de leucocitos*

Los cinco tipos de leucocitos son distinguidos entre sí por su tamaño relativo y abundancia, el tamaño y forma de su núcleo, la presencia o ausencia de ciertos gránulos citoplasmáticos, rugosidad y tinción de dichos gránulos, y más importante sus funciones.

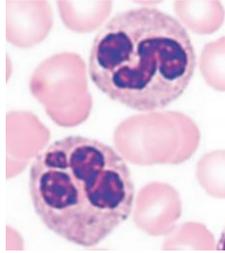
Todos los glóbulos blancos tienen organelas que parecen ser gránulos bajo la luz del microscopio. Algunos de estos lisosomas son llamados gránulos no específicos porque ocurren en las 5 clases de glóbulos blancos. Sobre la base de su aspecto en la microscopia óptica, los leucocitos se dividen en dos categorías principales: granulocitos y agranulocitos. Los granulocitos deben su nombre a la presencia de grandes gránulos en su citoplasma que pueden apreciarse con el microscopio óptico después de su tinción. En cambio, en los agranulocitos no se pueden apreciar y su núcleo es más parecido a la norma, es decir, son esféricos, ovales o con forma de riñón.

Hay que tener en cuenta que el color del núcleo de los glóbulos blancos (usualmente violeta) y los colores descriptos para los gránulos no son colores naturales. Estos son resultados de los colorantes utilizados para teñir los frotis, y pueden variar dependiendo de la tinción que se utilice en los mismos para estudiarlos. En particular, se referenciará a los colores que se ven cuando se utiliza la tinción de Giemsa–Wright.

Granulocitos:

- Neutrófilos: son los más abundantes de los glóbulos blancos, constituyendo entre un 60% y 70% de los leucocitos circulantes [1]. No solo son fáciles de encontrar en un frotis, sino que son claramente reconocibles por su tinción violeta y núcleo multilobulado (usualmente hay

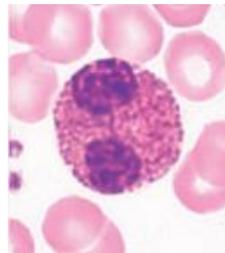
tres o cuatro lóbulos conectados por finas hebras). Por eso también se los denominan leucocitos polimorfonucleares o polimorfos. El citoplasma es ligeramente punteado con pequeños gránulos pálidos.



*Ilustración 2: Imagen ilustrativa de neutrófilos rodeados de glóbulos rojos [1]*

Los neutrófilos son células antibacteriales agresivas. Son rápidamente atraídos a áreas de infección o inflamación, donde fagocitan y digieren las bacterias que encuentran. También secretan una nube de químicos tóxicos (peróxido de hidrógeno e hipoclorito) que mata a un número mayor de bacterias. Los químicos son letales para los propios neutrófilos que usualmente tienen un corto período de vida. Aquellos que mueren son uno de los principales componentes de la pus y responsables de su color. Un número elevado de neutrófilos en el conteo en sangre es clínicamente importante porque puede indicar un proceso infeccioso.

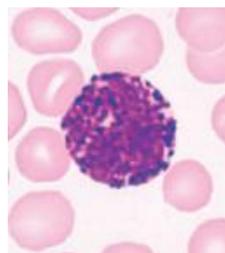
- Eosinófilos: representan solo entre 2% y 4% del total de glóbulos blancos, pero también son fáciles de reconocer [1]. Usualmente tienen núcleos bilobulados grandes unidos por una hebra corta. El citoplasma contiene abundantes gránulos específicos de color rosa o rosa anaranjado.



*Ilustración 3: Imagen ilustrativa de eosinófilos rodeados de glóbulos rojos [1]*

Los eosinófilos secretan químicos que debilitan o destruyen las paredes de grandes parásitos como anquilostomas o tenias que son muy grandes para ser destruidos por cualquier glóbulo blanco. También ayudan a eliminar cualquier complejo antígeno - anticuerpo formado durante la respuesta inmune. Sus números aumentan durante infecciones parasitarias, alergias y enfermedades del bazo o del sistema nervioso central, y fluctúan durante el día y la noche y en el transcurso del ciclo menstrual.

- Basófilos: son los más raros de los glóbulos blancos, comprendiendo menos del 0,5% y hasta un 1% del total [1]. Son reconocidos por sus gránulos muy abundantes y gruesos, generalmente teñidos de un violeta oscuros (a veces rosa). Los gránulos ocultan a los núcleos de la vista, pero éste es largo, pálido, y típicamente con forma de U o S.



*Ilustración 4: Imagen ilustrativa de basófilos rodeados de glóbulos rojos [1]*

Los basófilos secretan histamina, un vasodilatador que ensancha los vasos sanguíneos, aumentando el flujo al tejido lesionado, y hace que los vasos sean más permeables para que dichos agentes como los neutrófilos y proteínas de coagulación puedan llegar a los tejidos conectivos más rápidamente. También secretan heparina, un anticoagulante que inhibe la coagulación sanguínea y así promueve la movilización de otros glóbulos blancos a la zona. Son importantes en la alergia, inflamaciones y atracción de neutrófilos y otros glóbulos blancos a los sitios de infección. Un conteo elevado de basófilos se suele ver comúnmente en diabetes, varicela y varias otras enfermedades.

#### Agranulocitos:

- Linfocitos: son el segundo grupo más numeroso (después de los neutrófilos) representando entre un 25% y 33% del conteo de glóbulos blancos [1], y se detectan rápidamente en un frotis sanguíneo. Sus núcleos de color violeta oscuro llenan casi toda la célula. El citoplasma se tiñe de un color celeste claro y forma un borde estrecho y, a veces, apenas detectable alrededor del núcleo.



*Ilustración 5 Imagen ilustrativa de linfocitos rodeados de glóbulos rojos [1]*

Los subtipos de linfocitos (linfocitos T, linfocitos B y células Natural Killer) son indetectables bajo microscopio óptico, pero tienen diferentes funciones inmunes. El conteo de linfocitos aumenta en diversas enfermedades infecciosas y respuestas inmunes. Éstos suelen residir en los tejidos linfáticos, como las amígdalas, donde juegan un papel importante en la respuesta inmune.

- Monocitos: son los glóbulos blancos más grandes y comprenden entre un 3% y 8% del conteo de glóbulos blancos [1]. Su núcleo es grande y típicamente ovoide, con forma de riñón o herradura. El citoplasma contiene gránulos finos y escasos.



*Ilustración 6 Imagen ilustrativa de monocito rodeados de glóbulos rojos [1]*

Luego de migrar de la sangre a los tejidos conectivos, los monocitos maduran en células aún más grandes denominadas macrófagos. De hecho, la maduración y la proliferación de éstos (junto con los linfocitos) es un factor que determina la tumefacción de los ganglios linfáticos durante una infección. Los macrófagos consumen y destruyen microorganismos, desechos foráneos, y células muerta o moribundas, ingiriendo tanto como el 25% de su propio volumen por hora [1]. También ayudan a alertar al sistema inmune de patógenos o materia foránea. El conteo de monocitos tiende a aumentar con el aumento en inflamación e infecciones virales y son importantes para combatir infecciones crónicas, como tuberculosis.

Muchos de los valores mencionados anteriormente pueden ser obtenidos a través técnicas diagnósticas realizadas a partir de una extracción de sangre. Estos exámenes permiten obtener diferentes métricas como, por ejemplo, el recuento de los elementos corpusculares o hematocrito (la fracción de la sangre compuesta por glóbulos rojos). La razón por la que son tan importantes los diagnósticos cuantitativos y cualitativos basados en componentes celulares de la sangre es que son indicadores fácilmente accesibles de alteraciones en sus órganos de origen o degradación que son mucho menos accesibles [4]. Así, alteraciones de series de eritrocitos, leucocitos y trombocitos permiten sacar conclusiones importantes sobre la función de la médula ósea, al igual que las alteraciones de las células linfáticas indican reacciones o estados de enfermedad en los órganos linfopoyéticos especializados (básicamente, los ganglios linfáticos, el bazo y el órgano intestinal linfático difuso)

## Recuento diferencial de glóbulos blancos

El recuento diferencial de glóbulos blancos informa la cantidad y/o el porcentaje de cada tipo de glóbulo blanco (neutrófilos, eosinófilos, basófilos, monocitos y linfocitos) en una muestra. Este recuento puede ser obtenido mediante métodos manuales o automáticos.

### Métodos manuales

Para el método manual, se necesita extender en un portaobjetos una muestra de sangre (frotis) que más adelante se va a observar utilizando un microscopio.

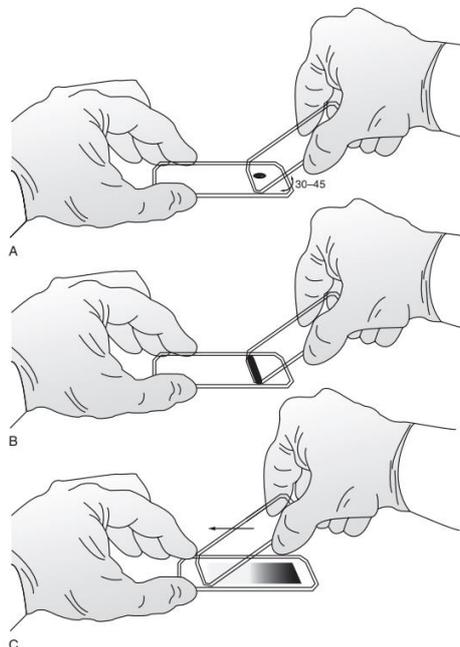


Ilustración 7: Técnica en cuña para realizar el frotis de sangre periférica. A. Ángulo correcto al sostener el frotis extensor. B. La sangre se extiende en toda la superficie de contacto entre los portaobjetos. C. Frotis en cuña completado [5]

Desafortunadamente, inclusive en frotis correctamente extendidos, la distribución de los distintos tipos celulares no es totalmente al azar.

Para un conteo diferencial confiable, el frotis no debe estar muy fino y la cola de este debe ser suave. Para lograr esto, el frotis debe ser hecho con un rápido movimiento utilizando un cubreobjeto. Esto

debería resultar en un frotis en el que haya alguna superposición de los glóbulos rojos, disminuyendo a la separación cerca de la cola, y en la que los glóbulos blancos en el cuerpo del mismo no están demasiado encogidos. Si es muy delgado, muchos de los glóbulos blancos, quizás hasta un 50% de los mismos, se acumulan en los bordes y en la cola [6]. Además, lo usual es que haya una irregularidad cualitativa en la distribución: neutrófilos polimorfonucleares y monocitos predominan en los márgenes y en la cola; linfocitos predominan en el medio del frotis. Esta separación probablemente depende de diferencias en la rigidez, tamaño y la gravedad específica de los diferentes tipos de células.

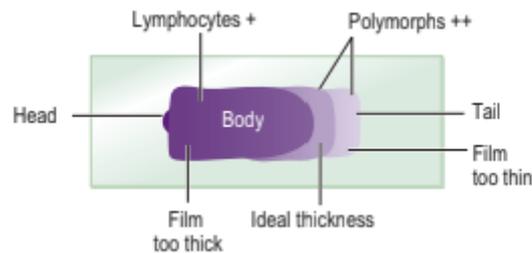


Ilustración 8: Dibujo esquemático de un frotis sanguíneo con la indicación de la distribución de los glóbulos blancos [6]

Diferencias en la distribución de los diversos tipos de células probablemente siempre estén presentes en pequeña medida, incluso en frotis bien hechos. Se han propuesto varios sistemas para realizar el recuento diferencial, pero ninguno puede compensar las grandes irregularidades en la distribución de un frotis mal hecho.



Ilustración 9: Ejemplo de frotis de sangre periférica bien realizado [5]

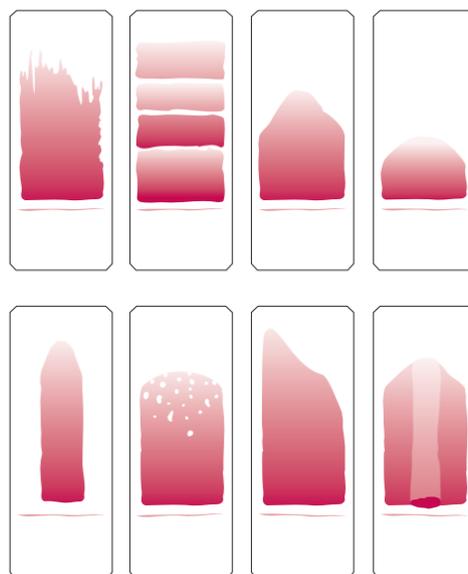


Ilustración 10: Ejemplos de frotis mal realizados [5]

## Método de conteo

Para poder inspeccionar el frotis, es necesario realizar una tinción, como ya se explicó anteriormente. El objetivo de esta es identificar las células y reconocer con facilidad la morfología a través del microscopio. La tinción de Wright o de Giemsa-Wright es la utilizada con mayor frecuencia para los frotis de sangre periférica [5]. Los colores que adquieren los glóbulos blancos bajo esta tinción se abordaron anteriormente. Es importante mencionar que tanto la tinción como una mala preparación del frotis son los principales contribuyentes a dificultades en el reconocimiento celular y los errores de distribución celular [7].

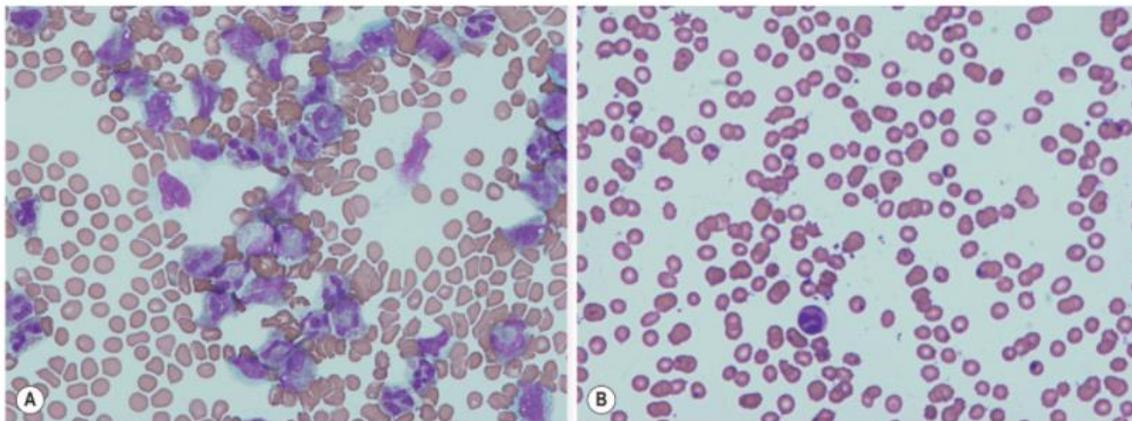


Ilustración 11: Ejemplo de frotis mal extendido. A. Se muestran muchos leucocitos en la cola y B. muy pocos leucocitos en el cuerpo del frotis [6]

Luego de la tinción, se coloca el frotis en el microscopio y se procede a contar las células utilizando un objetivo de x40 recorriendo las franjas del frotis.

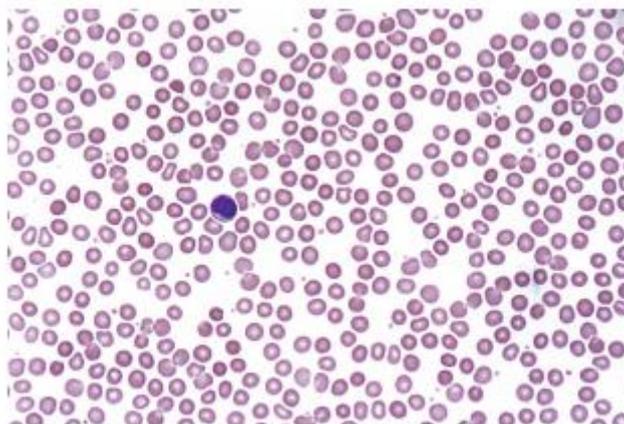


Ilustración 12: Área correcta del frotis de sangre para realizar el procedimiento. [5]

Se busca evitar los límites laterales del mismo. Hay que inspeccionar el frotis de la cabeza a la cola y si menos de 100 glóbulos blancos son encontrados en una sola franja, se deben examinar una o varias franjas más hasta contar al menos 100 células.

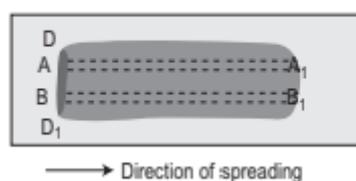
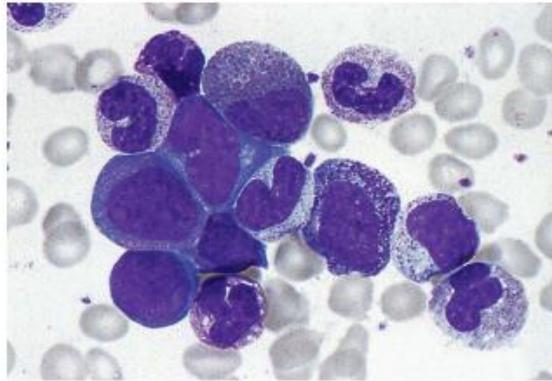


Ilustración 13: Ejemplo de distribución de las franjas longitudinales para realizar el recuento. [6]

Cada franja longitudinal representa la sangre extraída de una pequeña parte de la gota de sangre original cuando es esparcida entre el portaobjetos y el cubreobjetos extensor. Si todas las células son contadas en una franja, los totales diferenciales se aproximarán bastante al conteo diferencial real. Esta técnica es susceptible de error si las células en la parte densa del frotis no se pueden identificar; además, no permite ningún exceso de neutrófilos y monocitos en los bordes, pero esta preponderancia es leve en un frotis bien hecho y en la práctica hace poca diferencia en el resultado.

Esta técnica es fácil de llevar a cabo; con altos conteos ( $10\text{-}30 \times 10^9/\text{l}$ ) un frotis de 2-3 cm es deseable. En pacientes con muy altos conteos (como en la leucemia), el método debe ser abandonado y las células deben ser contadas en cualquier área extendida donde los tipos celulares sean fáciles de identificar.



*Ilustración 14: Ejemplo de microfotografía de frotis sanguíneo de paciente con leucemia. [6]*

La varianza del conteo diferencial depende no sólo de diferencias artefactuales en la distribución debido al proceso de extensión del frotis, pero también de la distribución 'aleatoria'; juntos son las causas más importantes de un recuento diferencial no confiable. La distribución aleatoria significa que, si se cuenta un total de 100 células, con una proporción real de neutrófilos de 50%, el rango ( $\pm 2\text{SD}$ ) dentro del cual caerá el 95% de los conteos es del orden de  $\pm 14\%$  (es decir, 36% - 64%) neutrófilos. En un conteo de 200 células se puede proveer un estimado más preciso; en el ejemplo anterior, el rango  $\pm 2\text{SD}$  va a ser entre 40% y 60%. En un conteo de 500 células, el rango sería reducido a 44% - 56% neutrófilos. En la práctica, un conteo de 100 células es recomendado como un procedimiento de rutina. Sin embargo, si se encuentran presentes pequeñas cantidades de células anormales, es más probable que se detecten cuando se realizan recuentos de 200 a 500 células que con un recuento de 100 células [6].

Otros de los problemas que presenta el recuento manual es que es una de las tareas más intensas en mano de obra en el laboratorio. Para lograr una buena ejecución se requiere un personal bien preparado, motivado y concentrado. Además, la confiabilidad del procedimiento debe supervisarse regularmente por los encargados especializados e intercambiando frotis con otros laboratorios [8].

A pesar de estas limitaciones, el recuento diferencial manual constituye el estándar para comparar resultados obtenidos con analizadores hematológicos automatizados. El papel del examen visual de frotis sanguíneos sigue siendo importante en el laboratorio clínico, ya que los analizadores hematológicos automáticos pueden rechazar un alto porcentaje de muestras consideradas como anormales que deben evaluarse visualmente.

## Métodos automáticos

Tradicionalmente, el análisis morfológico del frotis sanguíneo se ha realizado mediante microscopía óptica manual. Aunque este método es el "gold standard", tiene varias desventajas como las ya mencionadas, además de estar sujeto a una variabilidad interobservador relativamente grande.

También, existen limitaciones de espacio para el almacenamiento de portaobjetos de vidrio y es difícil acceder a una interconsulta utilizando el método clásico de microscopía manual [9].

Es por eso por lo que hace varios años que existe una demanda de equipos automatizados.

Los desarrollos iniciales en estos equipos se remontan a la década de 1960. El CELLSCAN, el Hematrak y el Cydak Scanning Microscope System se encuentran entre los primeros sistemas de análisis morfológicos automáticos introducidos. Sin embargo, estos primeros sistemas no proporcionaron mejoras significativas en el flujo de trabajo, ya que eran relativamente lentos y mostraban grados limitados de automatización.

En la década de 1980, el campo del diagnóstico de muestras de sangre en hematología avanzó significativamente con la introducción de los citómetros de flujo. Estos equipos son capaces de clasificar y detectar anomalías de leucocitos, entre otros análisis. Como se verá más adelante, aunque hubo grandes avances en la mejora de estos instrumentos y se han convertido en una parte integral de los diagnósticos de hematología de laboratorio en la actualidad, sigue existiendo la necesidad de revisión de frotis de sangre [9].

Más recientemente, se han desarrollado e introducido sistemas de microscopía digital automatizados más avanzados que permitieron superar algunos de los problemas que se encontraban en los primeros equipos.

A continuación, se van a detallar el funcionamiento, ventajas y desventajas de estos dos métodos de automáticos: citometría de flujo y sistemas de análisis de morfológicos automatizados.

## Citometría de flujo

La mayoría de los equipos que realizan el conteo diferencial automático que se encuentran disponibles usan citometría de flujo incorporado a un contador de sangre completo en lugar de ser solamente contadores diferenciales.

El principio básico de la citometría de flujo es inherente a la capacidad de analizar múltiples características de una sola célula dentro de una población heterogénea (por ejemplo, entre medio de otros tipos de células), en un corto período de tiempo.

Los citómetros de flujo modernos tienen la capacidad de analizar varios miles de células por segundo. Las células en suspensión pasan a través de un haz de luz (generalmente un rayo láser) en una sola fila; las señales generadas están relacionadas con el tamaño de la célula y la complejidad interna o granularidad de esta, lo que permite que el citómetro identifique diferentes poblaciones celulares en función de estas características [10].

Hay tres componentes principales en un citómetro de flujo:

1. El sistema de fluidos: Presentación de la muestra al láser
2. El sistema óptico: Recopilación de información de la luz dispersa del análisis
3. Sistema computacional/electrónico: Conversión de señales ópticas a digitales para visualización

### *Sistema de fluidos*

El sistema de fluidos incluye las bombas usadas para aspirar la suspensión celular al citómetro y el tubo a través del cual las células son impulsadas a través del equipo. Una vez aspirada, la muestra está rodeada por una corriente de solución salina tamponada introducida en el instrumento a una presión más alta que la muestra, de modo que las células en la muestra asumen una posición de una sola fila, aproximadamente, debido al fenómeno de enfoque hidrodinámico. La corriente enfocada luego llega a la celda de flujo, una cubeta de cuarzo donde las celdas están iluminadas por la luz de uno o más láseres. Finalmente, la corriente de fluido se dirige a un receptáculo de desechos [11].

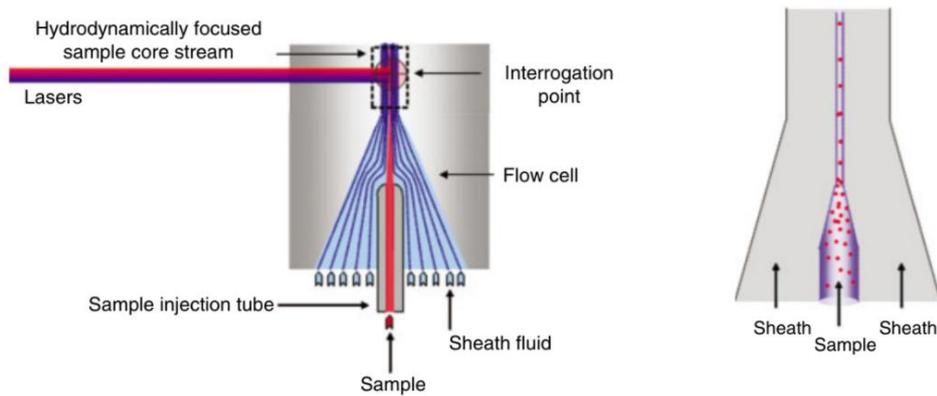


Ilustración 15: Esquema del sistema de fluidos con enfoque hidrodinámico. La forma de la celda de flujo (flow cell) ayuda a minimizar la turbulencia mientras que asegura que la muestra se encuentre enfocada en el centro de la corriente para la presentación al laser [10]

### Sistema óptico

A través del sistema óptico se busca poder medir características ópticas y de fluorescencia de células individuales. Algunas propiedades físicas que permiten resolver ciertas poblaciones celulares son el tamaño y la complejidad interna que se pueden obtener a partir de la dispersión frontal de la luz y la dispersión en ángulo recto, respectivamente [12]. Sin embargo, existen ciertos grupos que pueden ser complicados de distinguir basándose solamente en las características ópticas. Debido a esto es que se suelen medir también características de fluorescencia realizando ensayos previos en los cuales se unen diferentes anticuerpos con fluorocromos a los glóbulos blancos.

Finalmente, el sistema óptico incluye (1) los láseres utilizados para excitar fluorocromos, (2) el sistema para transportar la luz láser a la celda de flujo, (3) el sistema para transportar la luz fluorescente emitida desde las celdas hasta detectores específicos, y (4) los detectores mismos [12].

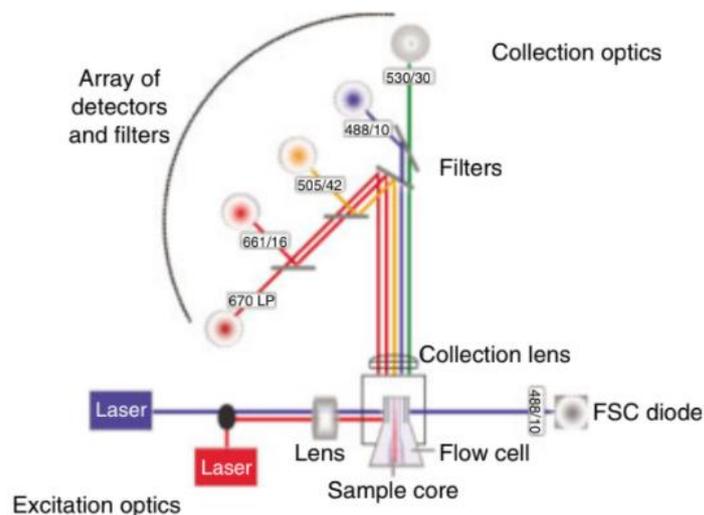


Ilustración 16: Ejemplo del sistema óptico del citómetro con los diferentes láser y filtros [10]

Los detectores suelen ser tubos fotomultiplicadores (PMT) o conjuntos de fotodiodos (PDA) que convierten fotones en impulsos eléctricos. La luz emitida por las células marcadas con fluorescencia se transporta a los detectores a través de una combinación de espejos dicróicos que permiten que pase la luz de longitudes de onda definidas mientras refleja la luz de otras longitudes de onda y filtros ópticos que reducen aún más las longitudes de onda que llegan a un detector. En los citómetros más nuevos, los cables de fibra óptica pueden ayudar a transmitir la luz emitida a los grupos de detectores apropiados, mientras que la luz viaja en el aire en citómetros más antiguos [12].

### Sistema computacional/electrónico

El sistema electrónico mide los impulsos eléctricos generados por los detectores y convierte estas mediciones en información digital que el software de análisis recopila e interpreta. El sistema informático asociado interactúa directamente con el citómetro y controla sus funciones.

En la mayoría de los sistemas más nuevos, el análisis de datos puede hacerse en la computadora conectada al equipo o en otras que acceden a los datos a través de un servidor central (es decir, análisis fuera de línea) [10].

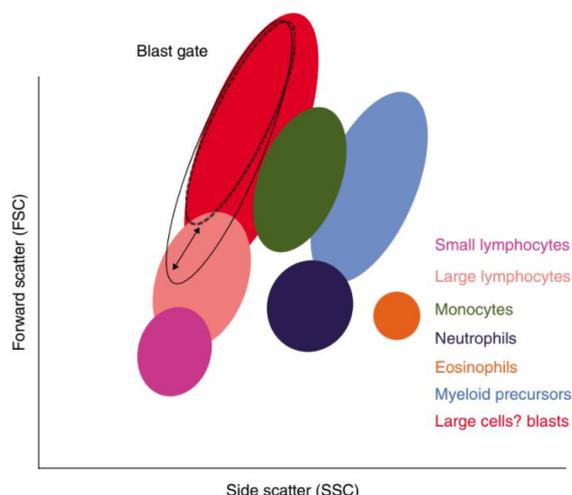


Ilustración 17: Diagrama esquemático de la diferenciación de los leucocitos en sangre periférica en base a la dispersión frontal de la luz y la dispersión en ángulo recto [10]

Los contadores diferenciales automatizados que utilizan este método cuentan un número de células mucho mayor que el que es posible con un conteo diferencial manual. En consecuencia, los recuentos automáticos son mucho más precisos que los recuentos manuales [6] además de reducir el tiempo que tarda en realizarse el estudio. Sin embargo, el análisis automatizado es incapaz de identificar y clasificar con precisión todos los tipos de células y es particularmente insensible a las células anormales o inmaduras. Por lo tanto, la mayoría de los analizadores marcarán posibles poblaciones anormales de glóbulos blancos, lo que indica la necesidad de un examen por parte de un morfológico experto para su identificación. Esto ocurre en un 10 a 25% de las muestras, lo que implican costos y tiempo en el procesamiento de la muestra [7].

También es importante la calibración del equipo para evitar errores potenciales que puedan invalidar las mediciones y que finalmente pueden afectar la precisión de estos. Este paso, seguido de una evaluación frecuente de la reproducibilidad mediante el análisis de muestras con concentraciones celulares conocidas, es una medida esencial de control de calidad.

### Sistemas de análisis morfológicos automatizados

La mayoría de estos sistemas hacen uso de una cámara digital acoplada a un sistema informático. Las imágenes digitales de células individuales se utilizan como material de entrada para una clasificación asistida por computadora. Esta está basada en parámetros de análisis de imágenes de células sanguíneas como características geométricas, de color y de textura. En ciertos equipos, también es automático el proceso de preparación y tinción del frotis, pero no siempre es el caso.

Un ejemplo de estos sistemas, son los DM Systems de CellaVision™ que son analizadores de imágenes digitales automatizados que localizan células en un frotis de sangre, capturan imágenes de las células, clasifican previamente las mismas usando un software de análisis de imágenes y luego las muestran en la pantalla de una computadora [9].

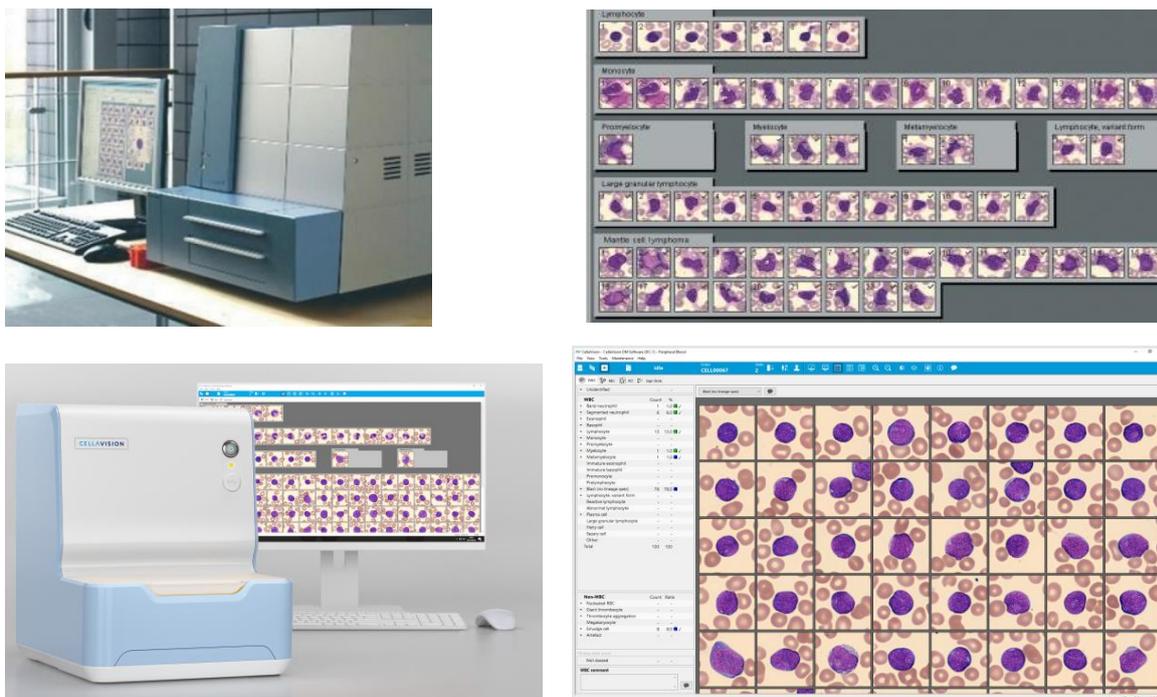


Ilustración 18: Imágenes de analizadores de CellaVision™ y la preclasificación del software de las imágenes. Se muestran el modelo DM96 (arriba) [13] y DC-1 (abajo) [14] que es una versión más moderna

El analizador escanea una parte del frotis e identifica automáticamente un área de análisis apropiada (monocapa) en la que localiza y captura las imágenes con un objetivo de x100 (con inmersión en aceite). El software clasifica previamente los glóbulos blancos para dar un recuento diferencial automático de frotis de sangre periférica. El reconocimiento del tipo de célula se basa en una red neuronal artificial, que analiza las imágenes digitales y preclasifica las células. El analizador requiere un morfológico experto para revisar todas las celdas y reclasificar todas las celdas no identificadas antes de que sea posible publicar los resultados. La velocidad estimada del CellaVision™ DM96 (uno de los equipos de la gama DM Systems) es de aproximadamente 30 portaobjetos por hora que es 10% a 25% más rápido que un recuento diferencial manual, incluso después de la reclasificación manual [9].

Existen estudios que buscan comparar el uso de estos sistemas con los conteos diferenciales manuales. Sin embargo, la imprecisión del conteo diferencial manual aumenta con aquellos leucocitos de baja incidencia, como los basófilos. Además, como ya fue mencionado, existe una gran variación en la clasificación dependiendo de las habilidades de los científicos junto con una falta de definiciones precisas respecto de anomalías morfológicas, lo que afecta la reclasificación y por ende los resultados.

Algunas de las ventajas del uso de estos sistemas comparado con el recuento manual incluyen la posibilidad de realizar interconsultas remotas, dada la digitalización de las imágenes, y la reducción en la fatiga visual del operador. Otra ventaja, es que las células son fácilmente rastreables y pueden recuperarse para su revisión. Además, la educación y evaluación del personal técnico puede estandarizarse y realizar los controles de calidad sobre los conteos.

Sin embargo, existen ciertas limitaciones de estos equipos en la identificación de células anormales. Es por esto, que las muestras de neonatos y pacientes sospechados de leucemia y otras enfermedades que causan la presencia de glóbulos blancos patológicos pueden requerir la revisión mediante microscopía manual. Por lo tanto, dependiendo de la población de pacientes, 10 a 20% de todos los frotis pueden requerir dicha revisión [9]. Sin embargo, debe tenerse en cuenta que la morfología de estas células patológicas todavía se puede evaluar utilizando imágenes generadas por el sistema, reduciendo así la tasa de evaluación manual adicional.

Por último, es importante mencionar, que las imágenes digitales de los frotis resultan más fáciles de guardar y ocupan menos espacio que los portaobjetos con los frotis, además de evitar problemas por envejecimiento de la muestra.

## DetECCIÓN DE OBJETOS

El objetivo de la siguiente tesis será la detección y clasificación de glóbulos blancos de muestras morfológicamente normales en imágenes de microscopía. Esta tarea se denomina detección de objetos (u object detection) cuyo objetivo en general es determinar si hay instancias de objetos dadas ciertas categorías (en este caso un tipo de glóbulo blanco) en una imagen y, si está presente, informar la ubicación espacial y la extensión de cada instancia del objeto [15].

La detección de objetos se puede enmarcar dentro de un área más grande denominada visión por computadora (o computer vision) que busca que las computadoras puedan tener una capacidad similar, o superar, la habilidad humana de ver y entender el mundo a su alrededor [16]. Esto puede parecer una tarea simple, pero en realidad el sentido de visión humana (y de los mamíferos) es bastante avanzado. En fracciones de segundos, los humanos pueden identificar los objetos dentro del campo de visión, sin pensar ni dudar. No solo pueden nombrar (clasificar) los objetos que se están viendo, sino distinguir los contornos, separar los objetos de su fondo e identificar su ubicación en el espacio [17]. Estas tareas que pueden ser triviales para los seres humanos, no lo son para las computadoras.

En un esfuerzo por entender mejor cómo se forman e interpretan las imágenes es que Hubel y Wiesel realizaron experimentos en la corteza visual de los gatos en las décadas de 1950 y 1960. La corteza visual es un área del cerebro que tiene pequeñas regiones de células que son sensibles a regiones específicas en el campo visual. En otras palabras, si se excitan áreas específicas del campo visual, entonces esas células en la corteza visual también se activarán. Además, las células excitadas también dependen de la forma y orientación de los objetos en el campo visual. Por ejemplo, los bordes verticales provocan la excitación de algunas células neuronales, mientras que los bordes horizontales provocan la excitación de otras células. Las mismas están conectadas usando una arquitectura en capas, y este descubrimiento condujo a la conjetura de que los mamíferos usan estas diferentes capas para construir porciones de imágenes en diferentes niveles de abstracción [18].

### Filtros y convolución

El primer concepto que surge a partir del experimento antes citado es el de un filtro que es esencialmente un detector de características. Un ejemplo de cómo funcionan estos filtros se puede ver en la siguiente imagen:

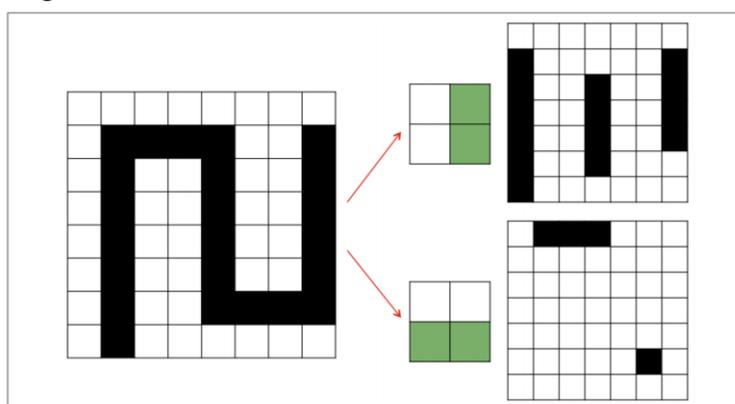


Ilustración 19 Ejemplo de la operación de convolución sobre una imagen para la detección de bordes horizontales y verticales [17]

En la figura se puede ver a la izquierda lo que sería la imagen de entrada sobre la que se quiere detectar líneas verticales y horizontales. Para detectar líneas verticales, se puede usar el detector de funciones en la parte superior, se desliza por toda la imagen y en cada paso, se verifica si hay una coincidencia. Se lleva un registro de las respuestas en la matriz de la parte superior derecha. Si hay una coincidencia entre la porción de la imagen analizada y el detector, se sombrea el cuadro correspondiente de negro.

Si no la hay, se deja en blanco. Este resultado, es el mapa de funciones e indica dónde se ha encontrado la función que se está buscando en la imagen original. Se puede hacer lo mismo para el detector de línea horizontal (abajo), lo que da como resultado el mapa de características en la esquina inferior derecha [17].

Esta operación se denomina convolución. El concepto básico de esta es que una máscara o kernel, una matriz de  $k \times k$  elementos, se gira  $180^\circ$  y se mueve en un patrón de trama a través de una imagen de entrada de  $M \times M$  píxeles;  $k$  suele ser un número entero impar, mucho más pequeño que el tamaño de la imagen. Cada píxel de la imagen de salida es la suma ponderada de los píxeles de entrada dentro de una región definida por el kernel, con los elementos de este definiendo los pesos [19]. Si la imagen de entrada es  $F$  (de tamaño  $M \times M$ ) y el kernel es  $H$  (de tamaño  $k \times k$ ), la imagen de salida  $G$  está dada por

$$G = F * H$$

Donde cada píxel de la imagen de salida está dado por:

$$g(i, j) = \sum_{m=-a}^a \sum_{n=-a}^a f(m, n)h(i - m, j - n)$$

donde  $a = \frac{k-1}{2}$

## Historia sobre la detección de objetos

Las primeras investigaciones sobre reconocimiento de objetos se basaron en técnicas que emparejaban patrones, enfocándose en objetos específicos cuya distribución espacial era rígida, como las caras. Antes de 1990 el paradigma principal del reconocimiento de objetos estaba basada en representaciones geométricas, luego corriéndose el foco de geometrías al uso de clasificadores estadísticos (como redes neuronales, Support Vector Machine (SVM) y Adaboost) basados en características relacionadas con la apariencia como, por ejemplo, la textura. Esta exitosa familia de detectores de objetos preparó el escenario para la mayoría de las investigaciones posteriores en este campo.

Luego, estas características relacionadas con la apariencia pasaron de representaciones globales a locales que estaban diseñadas para ser invariantes a cambios en traslación, escala, rotación, iluminación, punto de vista y oclusión. El desarrollo de características locales invariantes diseñadas manualmente ganaron una gran popularidad, comenzando por la función de Transformación de características invariables de escala (Scale Invariant Feature Transform SIFT), y el avance en varias tareas de reconocimiento visual que se basaron en el uso de descriptores locales, histograma de gradientes, SIFT, entre otros. Estas características locales suelen agregarse en general concatenandolas o usando otros métodos como Bolsas de palabras visuales (Bag of Visual Words) [15]. A este conjunto de técnicas se las suelen llamar el enfoque tradicional en visión por computadora.

Por años, estos flujos de trabajo de múltiples etapas conformados por descriptores locales diseñados manualmente y clasificadores discriminativos dominaron una variedad de dominios en el campo de la visión por computadora, incluyendo la detección de objetos. En 2012, hubo un punto de inflexión cuando las redes neuronales convolucionales profundas lograron resultados récord en la clasificación de imágenes. Ese año, Alex Krizhevsky, del laboratorio de George Hinton en la Universidad de Toronto, desarrolló Alexnet (una red neuronal convolucional) que ganó la competencia ImageNet. Este desafío consistía en generar un modelo que permitiera clasificar imágenes en una de las 200 posibles clases con un set de entrenamiento de 450.000 imágenes aproximadamente. A los algoritmos se les da cinco oportunidades para adivinar la respuesta correcta antes de pasar a la siguiente imagen del set de prueba. El objetivo del desafío es lograr con visión por computadora algo similar a la precisión de la visión humana que es aproximadamente del 95%. En 2011, el año anterior a Alexnet, el ganador

del desafío de ImageNet tuvo una tasa de error del 25.7%. En 2012, el segundo puesto tuvo una tasa de 26.1%, mientras que AlexNet tuvo una tasa de 16% [17].

El uso de redes neuronales convolucionales para detección y localización se puede rastrear hasta la década de 1990, cuando se usaba un modesto número de capas ocultas para detección de objetos, de forma exitosa para dominios restringidos como detección de caras. Sin embargo, más recientemente, las redes neuronales convolucionales profundas han llevado a mejoras récord en la detección de categorías más generales. Este cambio se produjo cuando la aplicación exitosa de estas redes en la clasificación de imágenes se transfirió a la detección de objetos [15].

El éxito de estos detectores se basa fuertemente en una vasta cantidad de datos para entrenamiento y grandes redes con millones o miles de millones de parámetros. La disponibilidad de GPUs con una gran capacidad computacional y conjuntos de datos para detección de gran escala (como ImageNet y MS COCO) juegan un papel clave en su éxito. Los grandes conjunto de datos permitieron a los investigadores enfocarse en problemas más realistas y complejos a partir de imágenes con grandes variaciones intraclasses y similitudes entre clases. Sin embargo, los conjuntos de datos con anotaciones precisas requieren mucho trabajo para obtenerlos, por lo que los detectores deben considerar métodos que puedan aliviar las dificultades de las conseguir las anotaciones o puedan aprender con conjuntos de entrenamiento más pequeños [15].

Una de las dificultades del enfoque tradicional es que es necesario elegir qué rasgos son importantes en cada imagen. A medida que el número de clases a clasificar aumenta, la extracción de características se vuelve más difícil y pesado. Se vuelve una tarea del ingeniero decidir qué rasgos o características (features) describen mejor las diferentes clases de objetos basado en prueba y error y en su juicio. Además, la definición de cada característica requiere lidiar con una gran cantidad de parámetros y todos deben ser correctamente calibrados por este ingeniero [20].

El aprendizaje profundo (o deep learning) y el uso de redes convolucionales introdujo el concepto de aprendizaje de punta a punta (end to end learning) donde a la computadora se le da solamente un conjunto de imágenes que tienen anotadas qué clases de objetos están presentes en cada imagen. En el caso de detección de objetos, se tienen anotaciones de los objetos y su localización. Con esto, el modelo de aprendizaje profundo es entrenado con los datos dados, donde redes neuronales descubren los patrones subyacentes en las diferentes imágenes y automáticamente resuelve los rasgos más descriptivos y salientes respecto de cada clase específica de objeto y para cada objeto. Se ha establecido que, en ciertas tareas, entre las que se incluye la detección de objetos, las redes neuronales convolucionales profundas se desempeñan mejor que los algoritmos tradicionales, aunque existe una relación de compromiso por los requisitos computacionales y el tiempo de entrenamiento. El empleo de estas técnicas produjo un cambio dramático en el trabajo de los ingenieros antes mencionados ya que el conocimiento y expertise requerido para extraer características diseñadas manualmente fue reemplazado por el conocimiento y expertise en iterar a través de las arquitecturas de aprendizaje profundo como se muestra a continuación.

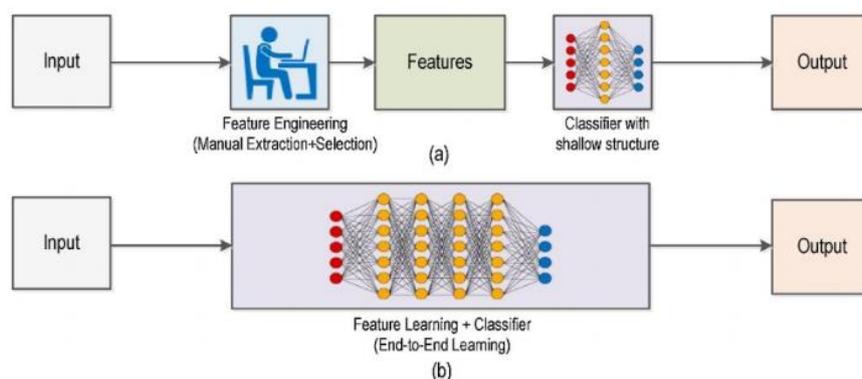


Ilustración 20 Flujo de trabajo tradicional de visión por computadora (a) vs. aprendizaje profundo (b) [20]

Más allá de las ventajas, existen algunos desafíos introducidos por el aprendizaje profundo. Estos algoritmos pueden lograr una precisión sustancialmente mejor. Sin embargo, esto viene con un costo de miles de millones de operaciones matemáticas adicionales y un requerimiento más grande de poder de procesamiento. El aprendizaje profundo requiere de estos recursos informáticos para el entrenamiento y en menor medida cuando se hacen inferencias. Es esencial tener hardware dedicado (por ejemplo, GPUs de alta potencia) para los desarrolladores de inteligencia artificial.

El entrenamiento de estos modelos lleva bastante tiempo. Dependiendo de la capacidad de cómputo del hardware, este puede llevar entre horas a días. También, el entrenamiento para cualquier aplicación en general requiere varias iteraciones ya que conlleva un proceso de prueba y error con diferentes parámetros.

Por otro lado, estos modelos necesitan grandes cantidades de datos. Muchas veces requieren de millones de registros. Por ejemplo, un conjunto de datos conocido denominado PASCAL VOC [21] consiste en 500 mil imágenes de 20 categorías. El tamaño necesario del conjunto de datos depende del problema a resolver y de las clases. También se desarrollan más adelante técnicas que se pueden utilizar cuando no se tiene un set tan grande. Para estas técnicas se usan algunas operaciones tradicionales de visión por computadora. Es importante mencionar esto para mostrar que no necesariamente son mutuamente excluyentes.

## Elementos de algoritmos de aprendizaje profundo

Los algoritmos de aprendizaje profundo se podrían describir como consistentes de tres partes:

- Un modelo
- Una función de costo
- Un optimizador

### Modelo

En la detección de objetos hay dos tareas: definir dónde hay un objeto y qué clase de objeto es. La primera es una tarea de regresión. El objetivo de estos tipos de modelos es encontrar una función  $f: \mathbb{R}^{M \times N \times 3} \rightarrow \mathbb{R}^{J \times 4}$  que a partir de una imagen (input) se obtenga la ubicación de los objetos. El tensor de entrada sería de tamaño  $M \times N \times 3$ , siendo  $M$  y  $N$  el alto y ancho de la imagen, respectivamente. En este caso hay una tercera dimensión que es 3 ya que se trata de imágenes RGB y por lo tanto tiene tres canales para la representación de los colores. Respecto a la salida, es un vector de tamaño  $J \times 4$ , siendo  $J$  el número de objetos localizados y el 4 tiene que ver con las coordenadas de la caja delimitadora del objeto. Esta caja contiene dentro el objeto deseado y buscar marcar la localización de este. Las coordenadas de esta pueden ser el punto medio junto con la altura y el ancho o las coordenadas de dos vértices opuestos, entre otras opciones. La función busca modelar la relación entre la imagen de entrada y la localización de los objetos de interés en dicha imagen.

La segunda es una tarea de clasificación en la cual se busca una función  $f: \mathbb{R}^{M \times N \times 3} \rightarrow \mathbb{R}^{J \times K}$  donde la salida es la probabilidad de que el objeto encontrado pertenezca a cada una de las clases. En este caso, la entrada es nuevamente el mismo tensor de tamaño  $M \times N \times 3$ , pero la salida es de  $J \times K$  donde  $J$  sigue siendo el número de objetos localizados, pero  $K$  es el número de clases al que puede pertenecer el objeto. Obviamente se suele asignar al objeto la clase de mayor probabilidad.

En particular, en los modelos generados con redes neuronales hay dos propiedades principales que siguen la idea general de cómo funciona el cerebro. Estas propiedades son parte de lo que define el funcionamiento de estas. Primero, la unidad básica de cualquier red neuronal es la neurona artificial. Éstas son modeladas basadas en las neuronas biológicas del cerebro, y al igual que éstas, son

estimuladas por las señales de las entradas. Estas neuronas artificiales pasan parte, pero no toda, la información que reciben de otras neuronas, a menudo con transformaciones [22].

En segundo lugar, muchas de las neuronas en el cerebro pueden ser entrenadas para solo transmitir señales que son útiles para lograr las metas finales del cerebro. Esto se refleja en las redes neuronales en que se puede entrenar a las neuronas para que solo transmitan las señales útiles.

Las neuronas biológicas en general están constituidas por cuatro partes [22]:

- Soma: cuerpo celular de la neurona que tiene dendritas y axón
- Dendritas: permiten que la célula reciba señales de neuronas vecinas conectadas. Una característica de estas es que cada una puede tener un peso distinto en la señal que genera dentro la neurona en respuesta a la señal recibida. Esto significa que, si dos dendritas reciben la misma señal, no necesariamente producen lo mismo en la neurona.
- Axón: Es una fibra que se extiende desde el soma y es lo que permite conectarse con las dendritas de otras neuronas.
- Sinapsis: es la unión de conexión entre el axón y las dendritas. La mayoría de las sinapsis envían señales desde el axón de una neurona a la dendrita de otra neurona.

La ubicación espacial se puede apreciar en la siguiente imagen:

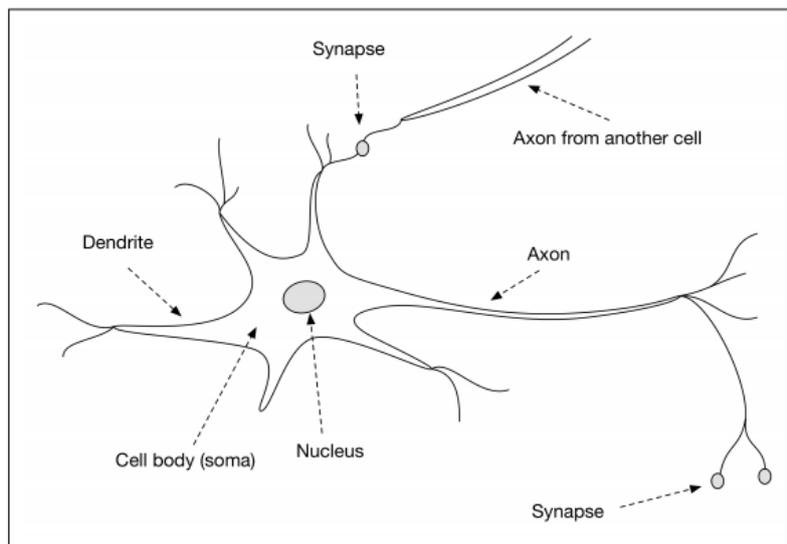


Ilustración 21 Esquema de una neurona y sus conexiones [22]

Las neuronas se comunican por el pasaje de impulsos electroquímicos a través de las sinapsis, de una célula a la siguiente. La liberación de químicos a través del espacio sináptico se produce siempre y cuando el impulso sea lo suficientemente fuerte para activar dicha liberación. La intensidad del impulso debe superar un umbral mínimo o no se liberan químicos.

Teniendo en cuenta este comportamiento de las neuronas biológicas es que se modela las neuronas artificiales, como se puede ver en el siguiente diagrama.

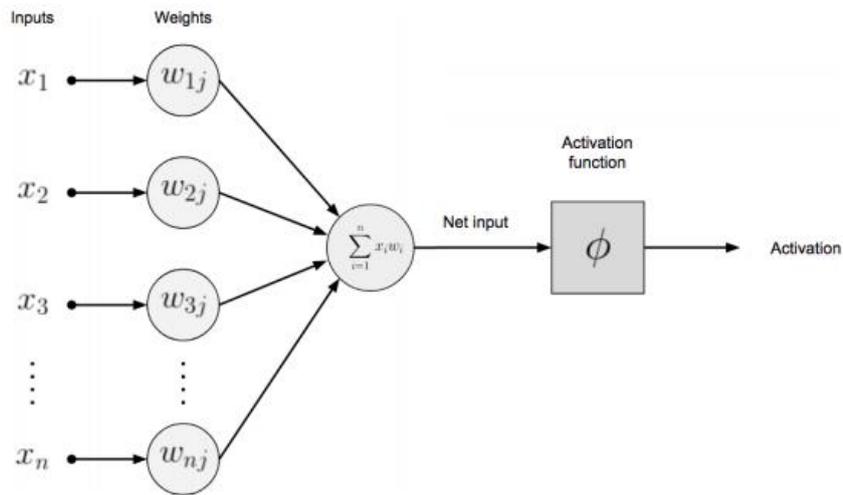


Ilustración 22 Esquema de funcionamiento de una neurona artificial con sus componentes [22]

En este caso, las entradas serían como las señales recibidas de otras neuronas. Los pesos representarían el peso de cada dendrita. La función de activación sería el umbral que debe superar la intensidad del impulso que recibe la neurona biológica para liberar los químicos que continúan con la señal hacia otra neurona.

Obviamente, el esquema representa el funcionamiento de una neurona. Para formar la red neuronal se conectan varias neuronas artificiales como se puede inferir del nombre. A partir de varias neuronas se forman lo que se denominan capas. Además de capas formadas por estas neuronas, existen otros tipos de capas que tienen diferentes funcionalidades en el entrenamiento, que se desarrollarán más adelante.

En el caso de las redes neuronales convolucionales, una de las capas es la convolucional. En estas, los pesos son los de los kernel de la convolución, que se definió anteriormente. Como ya se mencionó, en lugar de definir los pesos de los kernel de antemano, a través del entrenamiento de la red se va definiendo cuáles son las características relevantes con el ajuste de los pesos con la optimización y la función de costo. Los pesos son la función que mapea de la imagen de entrada a una salida con la ubicación del objeto y la probabilidad de que pertenezca a cada una de las clases, resolviendo el problema de regresión y clasificación, respectivamente.

### Función de costo

La función de costo permite cuantificar cuán cercanos son las salidas del modelo a las etiquetas (salidas deseadas) y son una medida del error que se busca minimizar. A medida que se varían los parámetros del modelo (los pesos) se evalúan las salidas utilizando esta función para poder comparar entre los distintos sets de pesos cuáles realizan mejor la tarea en cuestión (dependiendo de si el valor aumenta o disminuye). La función de costo se define específicamente para cada problema dependiendo de si es un problema de regresión, clasificación binaria, clasificación m-aria, segmentación, o salidas combinadas (multitasking) como es el caso de la detección de objetos [22].

Lo que se hace es calcular un único valor representativo de esa configuración de parámetros a partir del promedio de los errores entre la salida predicha y la deseada sobre todo los datos.

La idea es buscar los parámetros que minimizan ese “costo” incurrido por los errores. De esta forma, las funciones de costo permiten replantear el entrenamiento de las redes neuronales como un problema de optimización. En la mayoría de los casos, estos parámetros no pueden resolverse

analíticamente, pero pueden aproximarse bien con algoritmos de optimización iterativos como los que se mencionarán a continuación.

## Optimización

Como se mencionó en la sección anterior, para resolver el problema de optimización que presenta el entrenamiento de la red lo que se buscaría es encontrar los parámetros que logren llevar la función de costo, que se va a denominar  $J(\theta)$ , a un mínimo. Es importante mencionar que la optimización de modelos no lineales, como suele ser el caso de las redes convolucionales, es una tarea compleja exacerbado por el hecho de que estos modelos están compuestos por un gran número de parámetros ajustables ( $\theta$ ). Dado la complejidad del problema es que se opta por algoritmos iterativos que buscan la solución local óptima en cada punto [16].

Una de las opciones es utilizando el gradiente de  $J(\theta)$  ( $\nabla_{\theta}J(\theta)$ ) ya que este señala en cada punto la dirección en la que la velocidad de crecimiento de  $J(\theta)$  es mayor; por lo tanto, el opuesto del gradiente ( $-\nabla_{\theta}J(\theta)$ ) señala en cada punto la dirección en la que la velocidad de decrecimiento de la función es mayor. Para utilizar este método, es necesario que la función de costo  $J(\theta)$  sea continua y derivable para poder hallar el gradiente [23].

Antes de continuar, es necesario mencionar que además de los parámetros que se definen en la red neuronal, los algoritmos requieren de otros parámetros adicionales para llevar a cabo el entrenamiento. Estos se denominan hiperparámetros. Uno de los hiperparámetros es la tasa de aprendizaje (o learning rate)( $\eta$ ). Este valor se setea antes de iniciar el entrenamiento y modula cuanto se modificarán los parámetros (pesos) en cada iteración respecto del gradiente.

Cada iteración del entrenamiento en el tiempo  $t$  se puede describir con la siguiente ecuación para la actualización de los parámetros [16]:

$$\theta_t = \theta_{t-1} - \eta \delta_t \text{ con } \delta_t = \nabla_{\theta}J(\theta_t)$$

También se puede ver en la siguiente imagen cómo sería el proceso deseado con la optimización de  $J(\theta)$ .

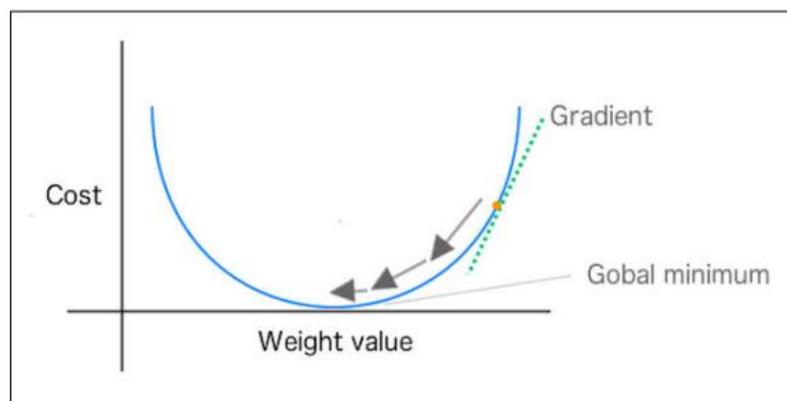


Ilustración 23: Gráfico ilustrando el cambio de los pesos hacia el mínimo global durante el proceso de optimización [22]

La elección de la tasa de aprendizaje resulta de vital importancia. Si se elige una tasa de aprendizaje demasiado chica, se corre el riesgo de tardar demasiado en el proceso de entrenamiento ya que los pasos (indicados en la imagen como flechas) serían demasiado chicos. En cambio, si se elige una tasa muy grande, se corre el riesgo de empezar a divergir del mínimo. Es por eso por lo que hay varias técnicas de optimización que utilizan tasas de aprendizaje adaptables para automatizar el proceso de seleccionarlas.

Aunque resulte obvio, este algoritmo se denomina de descenso por gradiente. Existen tres variantes de este que difieren en cuántos datos se usan para calcular el gradiente de la función de costo. Dependiendo de la cantidad de datos, hay una relación costo beneficio entre la precisión de la actualización de los parámetros y el tiempo que lleva realizarla.

## Algoritmos de optimización

### *Descenso por gradiente en lote (Batch Gradient Descent)*

Calcula el gradiente de la función de costo para todo el conjunto de entrenamiento. Como necesita calcular el gradiente sobre todo el conjunto de datos, solo realiza una actualización de los pesos. El problema de este algoritmo es que puede ser muy lento y sobre todo que es inviable para conjuntos de datos que no entran en memoria, como suele pasar en el caso de detección de objetos donde los datos son imágenes. Dentro de las ventajas, se encuentra que para problemas convexos (donde  $J(\theta)$  es una función convexa) el algoritmo converge al mínimo global. En el caso de problemas no convexos, converge a un mínimo local [16].

### *Descenso por gradiente estocástico (Stochastic Gradient Descent (SGD))*

En contraste al anterior, este algoritmo realiza la actualización de los parámetros para cada par de entrada y salida presente en el set de entrenamiento. Como consecuencia, converge más rápidamente con respecto al descenso por gradiente en lote. Además, los parámetros pueden ser ajustados por la presencia de nuevos ejemplos. El único problema es que su convergencia es usualmente inestable, especialmente para tasas de crecimiento relativamente grandes y cuando los conjuntos de datos de entrenamientos son diversos. Cuando la tasa de aprendizaje se establece correctamente, el descenso por gradiente estocástico consigue generalmente una convergencia similar, comparado con el descenso por gradiente en lote, tanto para problemas convexos como no convexos [16].

### *Descenso por gradiente en mini lote (Mini Batch Gradient Descent)*

Finalmente, el método de descenso por gradiente en mini lote es una forma mejorada del enfoque de SGD, que provee una buena relación de compromiso entre eficiencia y estabilidad de la convergencia. El método divide el conjunto de datos de entrenamiento en un número de mini lotes, cada uno consistente de un número relativamente chico de entradas. La actualización de los parámetros se realiza después de calcular el gradiente de cada mini lote. Es importante mencionar que en general los datos de entrenamiento se mezclan de forma aleatoria para mejorar la homogeneidad de este conjunto. Esto asegura una mejor tasa de convergencia comparada al descenso por gradiente en lote y una mejor estabilidad respecto de SGD [16].

## Desafíos de los algoritmos de descenso por gradiente

Algunos de los desafíos que tienen estos algoritmos son [24]:

- Muchas veces, se establece que en diferentes momentos del entrenamiento se vaya ajustando la tasa de aprendizaje ya que, por ejemplo, se puede querer empezar con un valor más alto para que vaya más rápido el algoritmo, pero a medida que va pasando el entrenamiento y se acerca a un mínimo, se reduzca esta tasa para evitar que diverja. En estos casos, se define de antemano en qué momento se va a cambiar y qué valores se van a utilizar. El problema de este enfoque es que no se adapta a las características del conjunto de datos.

- Por otro lado, la misma tasa de aprendizaje se aplica a la actualización de todos los parámetros. Si el conjunto de datos es disperso y los rasgos tienen diferentes frecuencias, puede que no se desee actualizar todos los parámetros en la misma medida, pero realizar una actualización más grande en características que aparecen raramente.
- Otro desafío importante en minimizar funciones de costo altamente no convexas, que suelen ser comunes en redes neuronales, es evitar quedar atrapado en los numerosos mínimos locales subóptimos. Un problema aún mayor son los puntos silla o minmax, donde en una dimensión la pendiente es positiva y en otra negativa. Estos puntos suelen estar rodeados de una meseta del mismo error, lo que hace especialmente difícil para el algoritmo de descenso por gradiente escapar. Esto sucede porque el gradiente es cercano a cero en todas las dimensiones. A continuación, una imagen ilustrando el fenómeno.

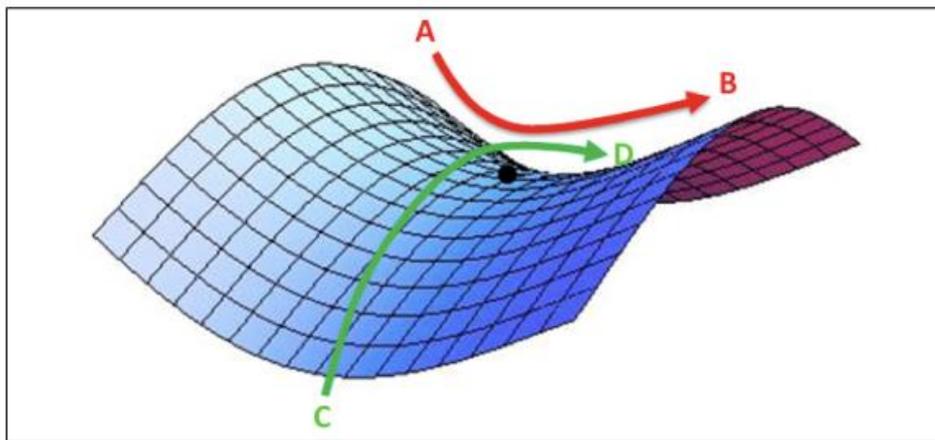


Ilustración 24: Ilustración de puntos silla en una función. Se puede observar que en la dimensión entre A y B el punto es un mínimo, mientras que entre C y D es un máximo. [17]

#### Variaciones de los algoritmos de descenso por gradiente

Debido a los inconvenientes descritos es que existen diferentes algoritmos con algunas variaciones para hacer frente a estos desafíos. Dentro de los mismos se encuentran las optimizaciones basadas en momento lo que provee una versión mejorada de SGD con mejores propiedades de convergencia. Por ejemplo, el SGD puede oscilar cerca de un mínimo local, resultando en un retraso en la convergencia [17]. El momento se suma al gradiente calculado en paso de tiempo anterior ponderado por un hiperparámetro ( $\gamma$ ) a la ecuación de actualización de pesos:

$$\begin{aligned}\theta_t &= \theta_{t-1} - a_t \\ a_t &= \eta \nabla_{\theta} J(\theta_t) + \gamma a_{t-1}\end{aligned}$$

El término momento tiene sentido físico. Las dimensiones cuyos gradientes apuntan en la misma dirección se amplían rápidamente, mientras que las dimensiones cuyos gradientes siguen cambiando de dirección se suprimen. Esencialmente, la velocidad de convergencia aumenta porque se evitan oscilaciones innecesarias.

Como se mencionó, existen también otros problemas relacionados con la selección de la tasa de crecimiento y con la adaptación a las características del conjunto de datos. Para estos problemas existen distintas estrategias que utilizan los diferentes optimizadores, cuyo detalle escapa al foco de la presente tesis. Sin embargo, se desea mencionar el algoritmo ADAM (ADAPtive Moment Estimation) que es un algoritmo adaptativo que estima una tasa de crecimiento distinta para cada parámetro.

Además, para actualizar los parámetros usa el primer y segundo momento del gradiente. Este algoritmo suele escalar bastante bien a problemas de gran escala y exhibe buenas propiedades de convergencia. Es por esto por lo que ADAM suele ser a menudo la elección por defecto de muchas aplicaciones de visión por computadora basadas en aprendizaje profundo o deep learning.

## Algoritmos de detección de objetos

Actualmente, existen principalmente dos tipos de algoritmos. Por un lado, detectores de dos etapas, como Faster R-CNN [25] o Mask R-CNN [26], que usan en la primera etapa una Region Proposal Network. Esta genera las propuestas de regiones de interés que posteriormente serán enviadas a la segunda etapa para la clasificación de objetos y de regresión para obtener la localización de estos. Estos modelos suelen tener altos niveles de precisión, pero suelen ser más lentos para realizar las predicciones, además de ser más complejo su entrenamiento [15].

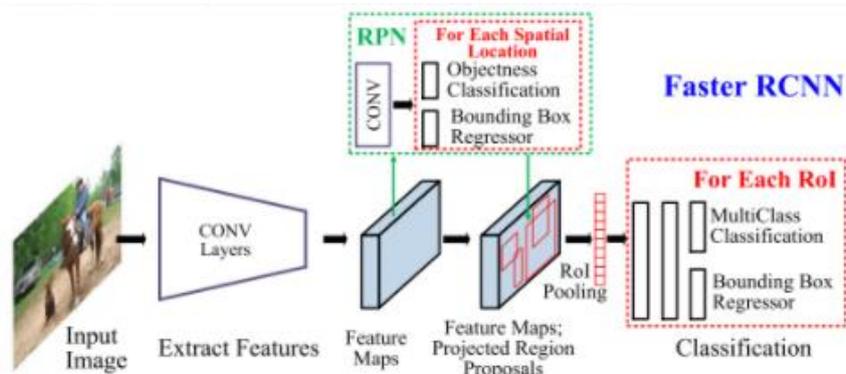


Ilustración 25: Esquema representativo de la estructura de Faster RCNN como ejemplo de detectores de dos etapas [15]

Por otro lado, existen los detectores unificados o de una etapa que refieren a arquitecturas que directamente predicen la probabilidad de cada clase y la localización de los objetos a partir de imágenes completas. Constan de una red neuronal convolucional que no involucra la generación de propuestas de regiones o post clasificación y así encapsulan todos los cálculos en una sola red. Dado que todo el proceso consta de esta única red, se puede optimizar de punta a punta directamente con el rendimiento de la detección.

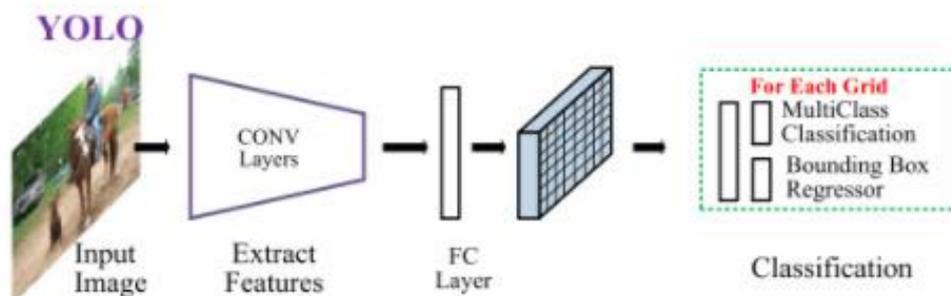


Ilustración 26: Esquema representativo de la estructura de YOLO como ejemplo de detector unificado [15]

Un ejemplo de este tipo de modelos es YOLO [27] (You Only Look Once). Este es un detector unificado que detecta objetos como un problema de regresión de píxeles de imágenes a cajas delimitadoras de objetos separadas espacialmente y probabilidades de clase asociadas. Dado que la etapa de generación de la propuesta de región se abandona por completo, YOLO predice directamente la detección utilizando un pequeño conjunto de regiones candidatas. A diferencia de los enfoques basados en regiones, como Faster R-CNN, que predicen detecciones basadas en características de una

región local, YOLO usa características de una imagen a nivel global. En particular, YOLO divide la imagen en una grilla de  $S \times S$ , cada una de las cuales predice probabilidades ( $p$ ) de  $c$  clases, ubicaciones de  $B$  cajas delimitadoras (bounding boxes) y niveles de confianza. Al no tener la etapa de propuestas de regiones, YOLO es un algoritmo rápido por diseño, corriendo en tiempo real. Como este algoritmo ve la imagen completa al realizar predicciones, implícitamente codifica información contextual sobre clases de objetos, y es menos probable predecir falsos positivos en el fondo. YOLO realiza mayores errores de localización que Faster R-CNN, como resultado de la división aproximada de la ubicación de las cajas delimitadoras, la escala y la proporción. Como se establece en el artículo original, esta red puede no localizar algunos objetos, en especial los pequeños, posiblemente debido a la división de la cuadrícula y porque cada celda puede contener un solo objeto. Esto último se modifica en las siguientes versiones de la arquitectura (YOLO9000 [28] y YOLOv3 [29]). Estas versiones implementan mejoras sobre YOLO, aunque manteniendo parte de la estructura original, como la función de costo o la división en grilla.

Para la presente tesis, se decidió que los modelos se basen en la estructura de YOLO dado que el entrenamiento de Faster R-CNN es complejo. Por otro lado, las sucesivas versiones de este detector unificado además de ser más rápido que su contraparte de dos etapas tanto para entrenamiento como para predicción, logran una alta precisión y una buena relación de compromiso entre las dos (precisión y rapidez). Asimismo, la desventaja de YOLO es en la detección de objetos pequeños, pero en las imágenes de microscopía los glóbulos blancos son los objetos más grandes.

## Evaluación de los algoritmos

El desafío central en el aprendizaje profundo es que el algoritmo debe funcionar bien con datos nuevos, nunca vistos, no solo aquellos con los que el modelo fue entrenado. La capacidad de desempeñarse bien con entradas que no fueron observadas previamente se llama generalización.

Típicamente, cuando se entrena un modelo, se tiene acceso a un set de entrenamiento; se puede calcular una medida del error sobre esta colección de datos, llamado error de entrenamiento; y se reduce dicho error. Hasta este punto, se está describiendo un simple problema de optimización. Lo que distingue a los algoritmos de aprendizaje profundo es que se busca además que el error de generalización, también llamado error de evaluación sea bajo. El error de generalización se define como el valor esperado del error para una nueva entrada [30]. Usualmente, se estima este error midiendo el rendimiento del modelo en un grupo de datos de evaluación con muestras que se separan del conjunto de entrenamiento (previo al mismo).

Los factores que determinan que tan bueno es el rendimiento de estos modelos es su habilidad para:

1. Reducir el error de entrenamiento
2. Hacer que la diferencia entre el error de entrenamiento y evaluación sea pequeña

Estos dos factores corresponden a los dos desafíos centrales en aprendizaje profundo.

Uno de los mayores problemas de los modelos de redes neuronales es que resultan bastante complejos. Esto puede llevar a que el modelo se ajuste correctamente a los datos de entrenamiento, subajuste (underfitting) y sobreajuste (overfitting). El subajuste ocurre cuando un modelo no es capaz de obtener un error suficientemente bajo sobre el set de entrenamiento. El sobreajuste ocurre cuando la diferencia entre el error de entrenamiento y de prueba es demasiado grande. En la siguiente imagen se muestran los tres casos:

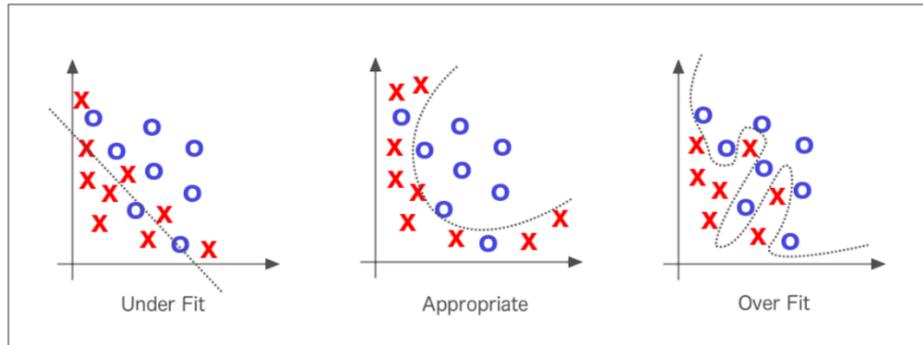


Ilustración 27: Representación de subajuste y sobreajuste en modelos de clasificación. [22]

La forma de controlar si un modelo es más probable al sobreajuste o subajuste es alterando su capacidad. Informalmente, la capacidad de un modelo es su habilidad para ajustarse a una gran variedad de funciones. A los modelos con baja capacidad se les puede dificultar adecuarse al set de entrenamiento. Mientras que los modelos con alta capacidad pueden sobreajustarse al memorizar propiedades de los datos de entrenamiento que no son útiles con el set de evaluación [30]. Básicamente, lo que sucede con estos últimos modelos es que son tan complejos, que es fácil que se ajusten perfectamente al conjunto de datos de entrenamiento porque tienen los suficientes grados de libertad, evitando que logre generalizar y por eso el alto error sobre el set de evaluación.

Esto hace que al diseñar un modelo de aprendizaje profundo siempre haya una relación de compromiso directa entre el sobreajuste y la complejidad del modelo. Si el modelo no es lo suficientemente complejo, puede que no sea lo suficientemente poderoso para capturar toda la información relevante y necesaria para resolver el problema. Sin embargo, si el modelo es demasiado complejo (especialmente si se tiene una cantidad limitada de datos), se corre el riesgo de sobreajustar [17]. En esta área, se toma el enfoque de resolver problemas muy complejos con modelos complejos y contramedidas para evitar el sobreajuste, que se desarrollarán más adelante.

Por último, cuando se realiza el entrenamiento, existe un punto en el cual en lugar de aprender características útiles y se empieza a sobreajustar sobre el set de entrenamiento. Para evitar esto, se desea detener el entrenamiento apenas se detecta que se comienza con el sobreajuste, para impedir una generalización pobre. Este punto se puede ver en la siguiente imagen.

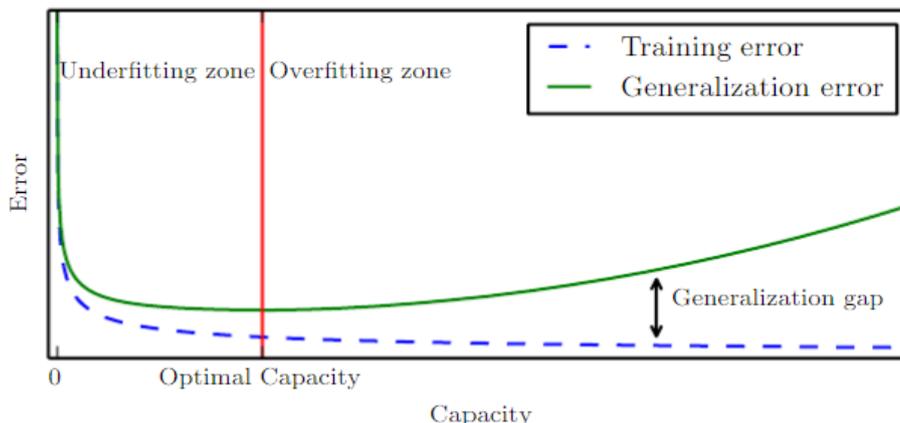


Ilustración 28: Gráfico del error en función de la capacidad durante el entrenamiento y diferencia entre error de entrenamiento y error de generalización [30]

Para el entrenamiento, se divide el proceso en epochs. Una epoch es una única iteración sobre todo el set de entrenamiento. En otras palabras, si se tiene un set de tamaño  $d$  y se realiza descenso por gradiente de mini lotes con un lote de tamaño  $b$ , entonces una epoch equivaldría a  $d/b$  actualizaciones del modelo. Al final de cada epoch, se desea medir que tan bien está generalizando el modelo. Para realizar esto, se puede utilizar un set adicional de validación (separado del de entrenamiento y

evaluación) [17]. El set de validación va a mostrar cómo se comporta el modelo con imágenes que todavía no vio. Si el error del modelo sobre el conjunto de entrenamiento continúa disminuyendo, pero el error sobre el grupo de datos de validación empieza a aumentar, es una buena señal para detener el entrenamiento porque se está sobreajustando.

El set de validación también es útil como una medida aproximada del rendimiento del modelo durante el proceso de optimización de hiperparámetros. Hasta ahora, se mencionaron algunos, como la tasa de crecimiento y el tamaño del lote, pero no se mencionó como realizar la búsqueda de estos para el modelo. Una forma puede ser aplicando una búsqueda en grilla en la que se elige un valor para cada hiperparámetro de un número finito de opciones y se entrena el modelo con cada permutación posible de valores. Luego, se selecciona la combinación de hiperparámetros con el mejor rendimiento respecto del set de validación, y se reporta los valores de las métricas elegidas sobre el set de evaluación con las predicciones de ese modelo.

### Métricas para la evaluación

En la sección anterior se definió la metodología para evaluar los modelos y las consideraciones necesarias para establecer si el modelo logra generalizar o no. Ahora es necesario definir las métricas que determinarán el rendimiento de estos.

La detección de objetos consiste en dos tareas: la localización de los objetos y la clasificación de estos. Para cada tarea se evaluarán diferentes métricas.

En la localización de objetos lo que se busca es ver si dentro de la caja delimitadora en la que el modelo predice que hay un objeto, hay realmente un objeto y que tan acertado es el tamaño y ubicación de dicha caja. La métrica que se utiliza para esta tarea es intersección sobre unión (IOU, Intersection over Union) entre la caja del conjunto de datos de salida (donde se sabe que hay un objeto) y la caja predicha por el modelo. Esta métrica, también conocida como índice de Jaccard, es la más comúnmente utilizada para comparar la similitud entre dos formas arbitrarias (en este caso rectángulos que serían estas cajas) [31]. La definición sería la siguiente, para dos formas

$$A, B \subseteq S \in \mathbb{R}^n \Rightarrow IoU = \frac{|A \cap B|}{|A \cup B|}$$

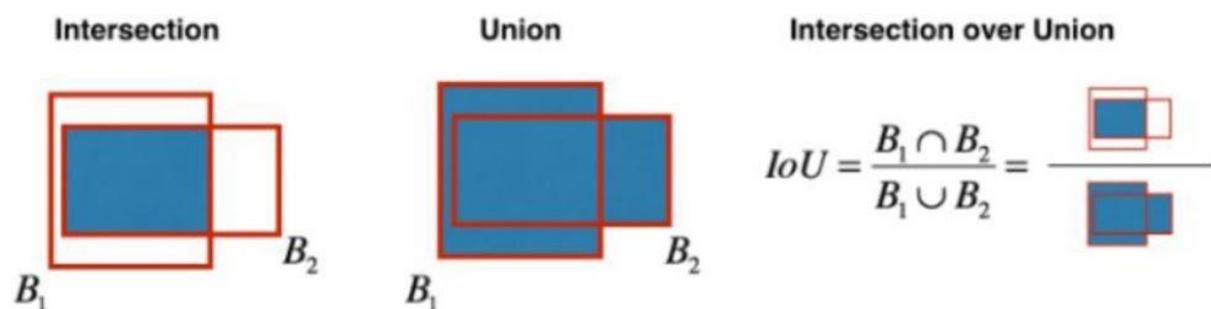


Ilustración 29: Cálculo de Intersection over Union (IoU) entre dos formas rectangulares [32]

Esta métrica codifica las propiedades de forma de los objetos en comparación, como pueden ser: los anchos, alturas y ubicaciones de dos cajas y lo traduce en la propiedad de la región. Luego, calcula una medida normalizada que se centra en sus áreas (o volúmenes). Obviamente, si las cajas no se superponen, la intersección es 0 [31]. Esta métrica da un valor entre 0 y 1, siendo 1 que ambas cajas están perfectamente superpuestas.

Una de las condiciones para definir que una predicción es un verdadero positivo, es que el IOU entre la predicción y la etiqueta del conjunto de datos tiene que superar un cierto umbral. En la presente

tesis, se decidió utilizar un valor típico de 0.5 que se observa en el desafío de PASCAL VOC, dado que no se busca detectar objetos de tamaños muy diferentes (pequeños, medianos y grandes).

La segunda condición para definir un verdadero positivo es que la clase predicha y la etiqueta sean iguales. Si alguna de las dos condiciones no se cumple, se considera como un falso positivo [15].

Antes de continuar sobre las métricas respecto de la clasificación, es importante mencionar que, como se desarrolló anteriormente, en los métodos actuales de detección de glóbulos blancos, no es lo mismo un falso negativo que un falso positivo para estas aplicaciones. Como en general, éstos terminan siendo métodos semi supervisados, en el sentido que los termina evaluando un técnico o bioquímico especializado, un falso negativo podría hacer que la persona que termina de verificar los resultados no evalúe ese resultado, mientras que un falso positivo lo terminará corrigiendo. También en este sentido, resulta muy importante la detección de los objetos a pesar de errores en la clasificación debido a que eso suele ser corregido por el operador.

En la tarea de clasificación dentro de la detección de objetos se suele usar la precisión y la sensibilidad. La precisión se define como la fracción de verdaderos positivos sobre el total de detecciones (verdaderos positivos más falsos positivos) realizados por el detector. Mientras que la sensibilidad se define como la fracción de verdaderos positivos sobre el total de objetos (verdaderos positivos más falsos negativos) [33].

$$\text{Precisión} = \frac{\text{Verdaderos Positivos (VP)}}{\text{Verdaderos Positivos (VP)} + \text{Falsos Positivos (FP)}}$$

$$\text{Sensibilidad} = \frac{\text{Verdaderos Positivos (VP)}}{\text{Verdaderos Positivos (VP)} + \text{Falsos Negativos (FN)}}$$

Con los valores de precisión y sensibilidad se suelen combinar en el  $F_1$  score que representa la media armónica entre precisión y sensibilidad. Los valores van en un rango entre 0 y 1, y los valores más altos del  $F_1$  score indican un alto rendimiento en la clasificación. Existe una variante denominada  $F_\beta$  score, que representa la media armónica ponderada entre la precisión y la sensibilidad [33]. La fórmula es la siguiente:

$$F_\beta = (1 + \beta^2) * \frac{\text{Precisión} * \text{Sensibilidad}}{(\beta^2 * \text{Precisión}) + \text{Sensibilidad}} = \frac{(1 + \beta^2) * VP}{(1 + \beta^2) * VP + \beta^2 * FN + FP}$$

Uno de los valores comúnmente utilizado de  $\beta$  es 2, lo que pondera la sensibilidad más alto que la precisión (al poner más énfasis en los falsos negativos). Este es el valor que se va a utilizar en la presente tesis dado que, como se mencionó en esta sección, se busca una baja tasa de falsos negativos ya que no los calificaría el operador.

Las salidas para la detección de objetos suelen ser las coordenadas de la caja delimitadora, la clase del objeto y un valor de confianza de dicha predicción. Para definir que hay un objeto, la confianza tiene que ser mayor a un cierto umbral. Dicho umbral sería un hiperparámetro del modelo que se determina con el set de validación. Para esto la metodología utilizada fue calcular el  $F_2$  score utilizando la precisión y sensibilidad para cada valor posible de umbral dentro un rango finito de valores y aquella configuración que tenga el mayor  $F_2$  score es con la que se evalúa el modelo. En esta tesis se decidió probar umbrales con un vector con 50 valores entre 0.01 y 1 con una distribución logarítmica. Esto se hizo tomando como criterio que no es igual una diferencia de valor  $x$  entre los elementos más pequeños del vector que entre los más grandes. Por ejemplo, no es lo mismo una diferencia de 0.1 entre 0.01 y 0.11 que entre 0.8 y 0.9

## YOLO (You Only Look Once)

El objetivo de la presente sección es explicar brevemente algunos de los lineamientos de YOLO y cuáles fueron los principales componentes en los cuáles se basan los modelos de la presente tesis.

YOLO enmarca el problema de detección de objetos como un problema de regresión y es una sola red neuronal la que predice las cajas y probabilidades de clase directamente sobre imágenes enteras en una única evaluación de esta [27].

El sistema divide la imagen de entrada en una grilla de  $S \times S$ . Si el centro de un objeto cae dentro de una celda de la grilla, esta celda es la responsable de detectar dicho objeto.

Cada celda predice B cajas delimitadoras y los valores de confianza de dichas cajas. Estos niveles de confianza reflejan que tan seguro está el modelo que esa caja contiene un objeto y también qué tan preciso cree que es la caja que predice.

Cada caja consiste en 5 predicciones: x, y, w, h y nivel de confianza de objeto, sobre el que se desarrollará más adelante. El par (x, y) representa el centro de esta relativo a los límites de la celda de la grilla. El ancho y el alto predicho es relativo a toda la imagen. Los 5 valores se encuentran entre 0 y 1.

Cada celda de la grilla también predice c probabilidades condicionales de clase,  $Pr(Class_i|Object)$ . Estas probabilidades están condicionadas a que realmente haya un objeto en dicha celda.

Entonces, por un lado, está el nivel de confianza de la caja delimitadora que se denominará nivel de confianza (o score) de objeto ya que está relacionado con el nivel de confianza de la red en que realmente hay un objeto en esa celda. Por otro lado, está la probabilidad de que el objeto pertenezca a una determinada clase ( $p_c$ ). Con cada predicción del modelo, hay un nivel de confianza en la clase del objeto si se determina que realmente que hay un objeto en dicha celda. Este valor se denominará nivel de confianza (score) de clase y está constituido por el máximo de las probabilidades condicionales de clase de esa celda ( $\max(p_c)$ ). Obviamente, la clase a la que pertenezca esa probabilidad será la asignada al objeto.

Para establecer si existe o no un objeto en dicha celda se define lo que se denominará un nivel de confianza general, o score general, que está relacionada con la confianza en que en dicha celda haya un objeto perteneciente a una determinada clase. Este valor se define como el nivel de confianza de objeto multiplicado por el nivel de confianza de clase. Aquellas celdas cuyos valores de nivel de confianza general sean mayores al umbral determinado se considerará que contiene un objeto de una determinada clase. En la siguiente imagen se puede ver el proceso de forma más esquemática.

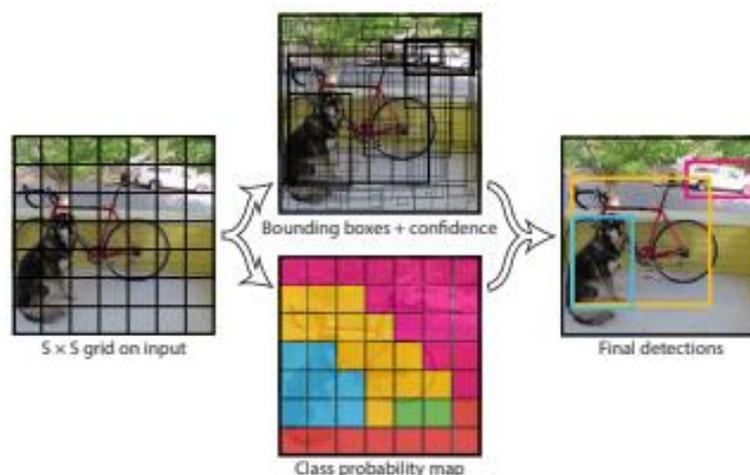


Ilustración 30: Imagen ilustrativa del funcionamiento de YOLO respecto de una imagen de entrada para poder obtener las predicciones de los objetos. [27]

Para los modelos de la presente tesis, las diferentes capas utilizadas se basaron en las arquitecturas de YOLO. Sin embargo, la cantidad y los parámetros de cada una fueron determinados heurísticamente para el problema particular de detección de glóbulos blancos y el conjunto de datos, cuyas características se desarrollarán más adelante. A continuación, la descripción de las diferentes capas:

## Capas de la red neuronal

### Convolucionales

Sobre estas capas ya se desarrolló anteriormente cómo funcionan y su utilidad. Algunos de los parámetros que se suelen modificar es el tamaño de los filtros y la cantidad de filtros de cada capa.

### Activación

Las funciones de activación fueron mencionadas cuando se describieron los modelos de neurona artificial y su analogía con el comportamiento de la neurona natural. Muchas de las funciones de activación son no lineales y es lo que permite introducir la no linealidad en las capacidades de modelado de la red.

Una de estas funciones es la Unidad Lineal Rectificada o ReLU (Rectifier Linear Unit). Es una función de activación simple y de especial importancia práctica debido a su rápido cálculo. Una función ReLU asigna a la entrada un 0 si es negativo y mantiene su valor sin cambios si es positivo [16]. Esto se representa con la siguiente función:

$$f_{ReLU}(x) = \max(0, x)$$

Esta activación se encuentra motivada por la forma de procesamiento en la corteza visual humana. La popularidad y eficiencia de la ReLU ha llevado a un número de variantes como, por ejemplo, la Leaky ReLU que se utiliza en YOLO.

La Leaky ReLU no reduce el valor completamente a cero cuando la entrada es negativa, más bien genera una versión reducida de ese valor [16]. La función es la siguiente:

$$f_{leakyReLU} = \begin{cases} x, & \text{si } x > 0 \\ cx, & \text{si } x \leq 0 \end{cases}$$

donde  $c$  es el factor de fuga que es una constante y generalmente es un valor pequeño. En el caso de YOLO este valor era de 0.1 [27]

Otras dos funciones de activación que se utilizaran solo para la etapa final de la red son la softmax y la sigmoidea.

La función sigmoidea toma un número real como entrada y su salida es un número entre 0 y 1. Se define de la siguiente manera [16]:

$$f_{sigm}(x) = \frac{1}{1 + e^{-x}}$$

Esta función de activación se va a utilizar para obtener el score de objeto del modelo.

Por otro lado, la función de activación softmax se utiliza cuando se desea que el vector de salida sea una distribución de probabilidades sobre un conjunto mutuamente excluyen de etiquetas, como sucede en la clasificación de los glóbulos blancos. A diferencia de otras capas, la salida de una neurona con una función de activación softmax depende de las salidas de todas las otras neuronas en esa capa.

Esto es porque requiere que la suma de todas las salidas sea 1. Siendo  $x_i$  el logit de la neurona  $i^{\text{th}}$ , se puede conseguir la normalización haciendo que la salida sea

$$f_{\text{softmax}}(x_i) = \frac{e^{-x_i}}{\sum_j e^{-x_j}}$$

Una predicción sólida tendría una sola salida en el vector cercana a 1, mientras que las salidas restantes están cerca de 0 [16]. Una predicción débil tendría múltiples clases posibles que son más o menos igualmente probables

### Batch Normalization

Es una de las capas que se agrega en las versiones posteriores de YOLO que permite una mejora en la detección y ayuda a la regularización del modelo. Su lugar es entre la capa convolucional y la capa de activación.

Esta capa normaliza la media y la varianza de las salidas de una capa convolucional para seguir una distribución gaussiana unitaria. Resulta muy útil para el entrenamiento eficiente de una red profunda (deep network) porque reduce el “cambio de la covarianza interna” de las capas de activación. Los cambios de la covarianza interna refieren a los cambios en la distribución de las activaciones de cada capa a medida que los parámetros son actualizados durante el entrenamiento. Si la distribución, que es lo que intenta de modelar la red neuronal convolucional, cambia constantemente (o sea, los cambios de la covarianza interna son altos), el proceso de entrenamiento va a ser más lento y le va a llevar más tiempo a la red la convergencia. La normalización de esta distribución lleva a una distribución consistente de la activación durante el proceso de entrenamiento, lo que mejora la convergencia [16].

Como ya se mencionó, esta capa se suele ubicar entre la salida de las capas convolucionales y antes de aplicar la función de activación no lineal. Alguno de los beneficios es:

- En la práctica, el proceso de entrenamiento de la red se vuelve menos sensible a la elección de hiperparámetros cuando se utiliza batch normalization
- Estabiliza el entrenamiento de redes convolucionales profundas y provee robustez contra una mala inicialización de los pesos de la capa convolucional.
- Batch normalization genera una gran mejora en la tasa de convergencia de la red neuronal
- Integra la normalización en la red al permitir la propagación hacia atrás de los errores a través de la capa de batch normalization y, por lo tanto, permite el entrenamiento de punta a punta de las redes profundas.
- Hace al modelo menos dependiente de ciertas estrategias de regularización para evitar el sobreajuste de la red

### Max Pooling

Una capa de pooling opera en bloques del mapa de características de entrada y combina las activaciones de esos rasgos. Esta operación de combinación se define mediante una función de agrupación como la función media o máxima. Similar a la capa de convolución, se debe especificar el tamaño de la región agrupada y el paso. La siguiente imagen muestra la operación de agrupación máxima (max pooling), donde se elige la activación máxima del bloque de valores seleccionado. Esta ventana se desliza por los mapas de entidades de entrada con un tamaño de corrimiento definido por el paso (1 en el caso de la imagen) [16].

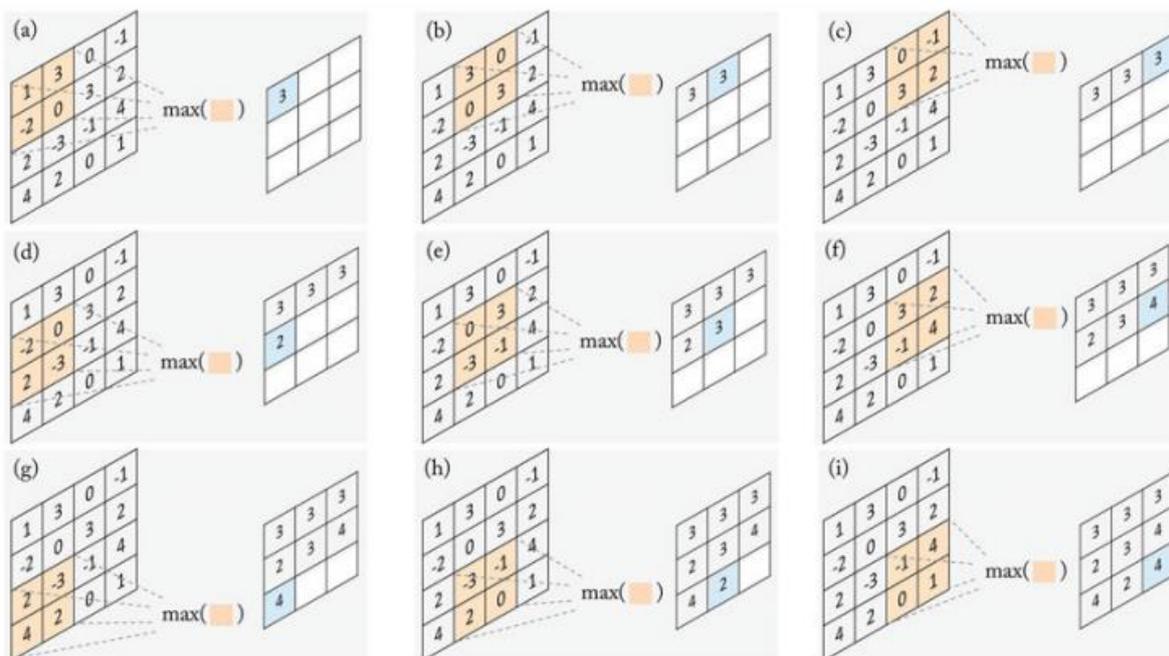


Ilustración 31: Imagen ilustrativa del funcionamiento de una capa de max pooling de 2x2 y paso de 1 [16]

La operación de la capa de max pooling de la imagen muestra la operación paso a paso cuando el tamaño de la región es de 2x2 y el paso es 1. Las imágenes (a) a (i) muestran los cálculos realizados a medida que la región agrupada en el mapa de características de entrada (que se muestra en naranja) se desliza en cada paso para calcular el valor correspondiente con el mapa de características de la salida (que se muestra en azul)

Esta operación reduce de manera efectiva el mapa de características de entrada. Este proceso de muestreo es útil para obtener una representación de este mapa de forma compacta y que es invariante a cambios moderados en la escala, posición y traslación del objeto en una imagen.

Por los parámetros utilizados en los modelos de la presente tesis, la cantidad de capas de max pooling es lo que define el tamaño de la grilla de salida ya que es la única capa que modifica el tamaño del mapa de salida respecto del de entrada. Sobre la grilla de salida es que el modelo, al igual que YOLO, realiza las predicciones.

El diseño de las diferentes redes utilizadas en la presente tesis se basa en los componentes esenciales mencionados anteriormente. Los detalles de dichos parámetros se encuentran en secciones posteriores

### Función de costo

En YOLO se optimiza una función de error cuadrático, por eso es por lo que se mencionó que este algoritmo toma el problema de detección de objetos como un problema de regresión. Esta es una función fácil de optimizar pero que puede no estar perfectamente alineada con el objetivo de minimizar el  $F_2$  score. El problema es que les da igual peso a los errores de clasificación que a los de localización. Además, en cada imagen hay muchas celdas de la grilla que no contienen ningún objeto. Esto hace que los valores de confianza de esas celdas sean cercanos a cero, a menudo dominando sobre el gradiente de las celdas que sí contienen objetos. Esto puede llevar a cierta inestabilidad del modelo, causando que el entrenamiento diverja en las primeras etapas.

Para remediar esto, dentro de YOLO se decidió aumentar el término de la función de costo de las coordenadas de la caja y disminuir los términos derivados de las predicciones de confianza de aquellas cajas que no contienen objetos. Se utilizan dos parámetros,  $\lambda_{coord}$  y  $\lambda_{noobj}$  lograr lo ya descrito. En el artículo [27] se utilizan  $\lambda_{coord} = 5$  y  $\lambda_{noobj} = 0.5$ , y que se tomarán en la presente tesis.

El error cuadrático también pondera igualmente los errores en cajas grandes que en chicas. La métrica de error de YOLO busca reflejar que las pequeñas desviaciones en cajas grandes importan menos que en las pequeñas. Para abordar este problema se utiliza la raíz cuadrada de la altura y el ancho en la función de error.

Durante el entrenamiento se optimiza la siguiente función de error:

*Función de costo*

$$\begin{aligned}
&= \lambda_{coord} \sum_{i=0}^{s^2} \mathbb{1}_i^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\
&+ \lambda_{coord} \sum_{i=0}^{s^2} \mathbb{1}_i^{obj} \left[ (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\
&+ \sum_{i=0}^{s^2} \mathbb{1}_i^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{s^2} \mathbb{1}_i^{noobj} (C_i - \hat{C}_i)^2 \\
&+ \sum_{i=0}^{s^2} \mathbb{1}_i^{obj} \sum_{c \in \text{clases}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned}$$

Donde:

- $s^2$ : número total de celdas en la grilla.
- $(x_i, y_i)$ : coordenada del punto medio de la caja delimitadora predicha por el modelo.
- $h_i, w_i$ : alto y ancho, respectivamente, de la caja delimitadora predicha por el modelo.
- $(\hat{x}_i, \hat{y}_i)$ : coordenada del punto medio de la etiqueta de la caja delimitadora asignada a la celda.
- $\hat{h}_i, \hat{w}_i$ : alto y ancho, respectivamente, de la etiqueta de la caja delimitadora asignada a la celda.
- $C_i$ : nivel de confianza sobre la presencia de un objeto en la celda predicha por el modelo.
- $\hat{C}_i$ : nivel de confianza sobre la presencia de un objeto asignado a la celda. Tiene un valor de 0 o 1.
- $p_i(c)$ : probabilidad de pertenencia de un objeto a la clase  $c$  predicha por el modelo.
- $\hat{p}_i(c)$ : probabilidad de pertenencia de un objeto a la clase  $c$  asignada a la celda.
- $\mathbb{1}_i^{obj}$ : denota si el objeto aparece en la celda  $i^{\text{th}}$

Es importante notar que la función de costo solo penaliza el error de clasificación si hay un objeto presente en dicha celda.

## Tasa de aprendizaje

En el artículo de YOLO se plantea un calendario de ajuste de la tasa de aprendizaje dependiendo de las epochs. En la siguiente tesis, se decidió utilizar Cyclical Learning Rate, que prácticamente elimina la necesidad de encontrar experimentalmente los mejores valores y calendario para estas tasas. En lugar de disminuir monotónicamente la tasa de aprendizaje, este método deja que la misma varíe cíclicamente entre límites razonables. El entrenamiento con tasas de aprendizaje cíclicas en lugar de

valores fijos logra una precisión de clasificación mejorada sin necesidad de ir modificando el valor  $\gamma$ , a menudo, en menos iteraciones.

La esencia de esta política respecto de la tasa de aprendizaje viene de la observación que un aumento en la misma puede tener un efecto negativo a corto plazo, pero obtener un efecto beneficioso a largo plazo. Esta observación lleva a la idea de dejar variar este número en un rango de valores en lugar de adoptar un valor fijo que se modifica después de una cantidad fija de epochs [34]. Esto es, se establece un límite inferior y superior. Entre estos dos valores es que varía cíclicamente este hiperparámetro.

Una idea intuitiva de por qué estos métodos funcionan vienen de considerar la topología de las funciones de costo. Como ya se mencionó, la dificultad en minimizar estas funciones viene dado por los puntos silla más que por mínimos locales. Como estos puntos silla pueden ralentizar el proceso de aprendizaje, el aumento de la tasa permite superar más rápidamente los plateau dados por estos puntos. Otra razón más práctica de por qué funciona este enfoque está dado porque es probable que el valor óptimo se encuentre entre esos límites [34]. Lo que termina sucediendo es que se utilizan valores cercanos al óptimo durante el entrenamiento.

La función básica de Cyclic Learning Rate es la siguiente:

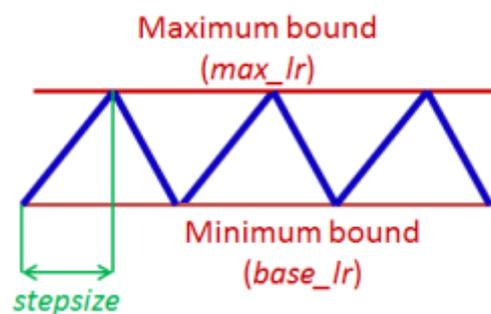


Ilustración 32: Imagen ilustrativa de la variación de la tasa de aprendizaje en función de las iteraciones durante el Cyclic Learning Rate. El tiempo de crecimiento y decrecimiento está dado por el paso (step size) y la tasa fluctúa entre el mínimo y máximo. [34]

Los parámetros de la función son los límites y el paso (step size). Este último es el número de iteraciones en medio ciclo, siendo el número de iteraciones el número de lotes necesarios para completar una epoch.

Existen diferentes implementaciones de este método, en la presente tesis se utilizará una versión en la cual la función triangular se encuentra modulada por un factor exponencial de  $\gamma^{\text{iteración}}$  (donde gamma es otro parámetro de la función).

### Non Maximum Suppression

Una vez terminado el entrenamiento del modelo, la red predice las cajas delimitadoras con sus correspondientes clases. En algunos casos, dos o más cajas refieren al mismo objeto, creando predicciones redundantes, como se puede ver en la siguiente imagen



*Ilustración 33: Imagen ilustrativa de la predicción de una red neuronal que predice dos cajas delimitadoras superpuestas para el mismo objeto. Solo una caja válida es elegida y es la de mayor puntaje (score) que es 0.99 [35]*

Para remover esas predicciones redundantes se utiliza el algoritmo de Non Maximum Suppression.

En este algoritmo, se seleccionan las cajas basadas en sus valores de confianza. Se utiliza como referencia la que tiene el máximo valor de confianza. Luego se comparan el resto de las cajas con menor nivel de confianza. Para hacer esto se calcula el IOU entre cada caja y la de referencia. Aquella cuyo IOU supera un determinado umbral, en este caso será de 0.5, será descartada como predicción ya que se supone que está prediciendo el mismo objeto, pero con un nivel menor de confianza (como en el ejemplo de la figura) [35].

Aquellas cajas que sobrevivan serán las predicciones reales del modelo que llegan al usuario y serán objeto de evaluación.

## Datos y Pre-Procesamiento

Para el siguiente trabajo, se utilizará el conjunto de datos Leukocyte Image for Segmentation and Classification (Imágenes de leucocitos para segmentación y clasificación) o LISC [36]. Esta base contiene imágenes hematológicas tomadas de sangre periférica de sujetos saludables. Las muestras fueron tomadas de 8 sujetos normales y 400 muestras fueron obtenidas de 100 frotis de microscopía. Las muestras fueron teñidas con la técnica de Giemsa-Wright y obtenidas con un microscopio de luz (Axioskope 40) usando un lente acromático y un aumento de 100. Luego, fueron retratadas por una cámara digital (Sony Model No. SSCDC50AP) y guardadas en formato BMP. Las imágenes son RGB de 576x720 pixeles y las máscaras de 568x712. Esta discrepancia en el tamaño será abordada durante el pre-procesamiento.

Todas las imágenes fueron recolectadas por el Centro de Investigación Hematológico-Oncológico y BMT del hospital Imam Khomeini en Tehran, Iran. De las 400 imágenes, hay 250 que tienen la segmentación manual de los glóbulos blancos realizada por un experto. Las clases que hay son: basófilos, eosinófilos, linfocitos, monocitos, neutrófilos y mixtos. Ésta última se refiere a aquellas imágenes que contienen más de un tipo de glóbulo blanco en la imagen. Cada clase contiene 53, 39, 53, 48, 50 y 8 imágenes, respectivamente. A partir del procesamiento, se obtiene que hay 229 imágenes con un solo glóbulo blanco, 13 imágenes con más de un glóbulo blanco (pero todos pertenecen a la misma clase) y 8 imágenes con más de un glóbulo blanco de diferentes clases. A continuación, un ejemplo de las imágenes de cada clase, excepto mixtos:

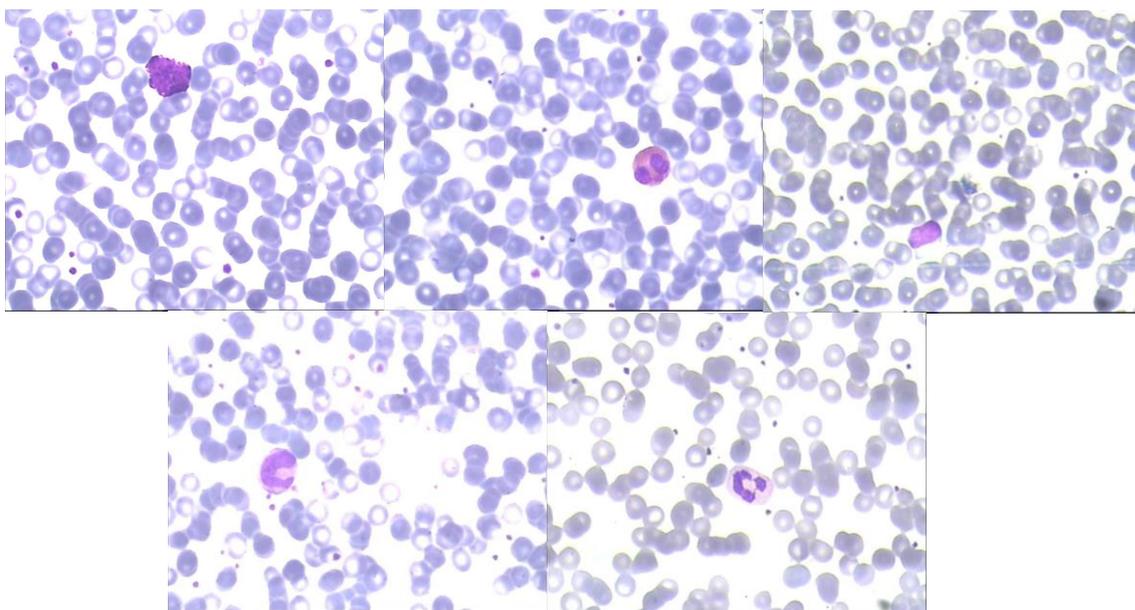


Ilustración 34: Imágenes del conjunto de datos de cada clase. De izq. a der.: basófilo (imagen 0), eosinófilo (imagen 54), linfocito (imagen 110), neutrófilo (imagen 159) y monocito (imagen 207) [36]

## Pre-Procesamiento

El primer paso realizado fue ajustar el tamaño de las máscaras para que coincida con el de las imágenes. Se verificó que, si se agregan cuatro filas y columnas a cada lado, la máscara quedaba centrada respecto de la imagen.

Las máscaras tienen diferentes valores dentro de la escala de grises ya que contienen la segmentación del citoplasma y el núcleo. Dado que el objetivo del presente trabajo es la detección de los glóbulos blancos y no la segmentación de estos se procedió a generar máscaras binarias a partir de las disponibles. Luego se realizó un proceso de etiquetado para poder obtener la ubicación de las diferentes células de interés. Esto es porque como se aclaró en la descripción de los datos, hay ciertas imágenes que tienen más de una célula y es necesario detectar todas las posibles máscaras para generar las cajas delimitadoras.

El siguiente paso dentro del preprocesamiento fue un proceso denominado data augmentation. Este proceso busca aumentar la cantidad limitada de imágenes para poder entrenar el modelo de aprendizaje automático con más datos y evitar el sobreajuste que se mencionó anteriormente. Se basa en crear datos falsos a partir del conjunto de datos original y agregarla al grupo de entrenamientos.

Esta técnica se suele utilizar en clasificadores dado que la tarea de estos es tomar una entrada  $x$  compleja y de alta dimensionalidad y obtener como salida una única categoría  $y$ . Esto significa que la principal tarea de un clasificador es ser invariante a una amplia variedad de transformaciones. En particular, las imágenes son entradas de alta dimensionalidad que pueden tener una gran variabilidad que muchas veces es fácilmente simulable. Lo importante para tener en cuenta al realizar la técnica es que las falsas imágenes sean posibles en un escenario real [30]. Por ejemplo, en el caso de clasificación de imágenes con números, si se hace una transformación de una imagen con un 6 y se le da vuelta  $180^\circ$  pasaría a ser un 9 por lo que no sería la misma categoría y por lo tanto no se podría realizar esa transformación para ese conjunto de datos. En este caso, en las imágenes de microscopía no existe realmente una orientación (arriba o abajo) por lo que se podían utilizar transformaciones de traslación, rotación y espejado para este proceso.

Como ya se estableció, dada la naturaleza del set se pueden aplicar las transformaciones antes mencionadas sin que haya modificaciones en la clasificación del glóbulo blanco. Sin embargo, surge un problema con la ubicación de estos para la tarea de detección. El mismo surge a partir del hecho de que cuando se realizan las transformaciones se pierde la ubicación de las cajas. Es por eso que se decidió realizar las mismas transformaciones tanto a las imágenes como a las máscaras binarias.

A partir de cada imagen del conjunto original, se obtuvieron 30 imágenes transformadas. Por lo tanto, para el set aumentado se obtuvieron: 1590 (basófilos), 1170 (eosinófilos), 1590 (linfocitos), 1440 (monocitos), 1500 (neutrófilos) y 240 (mixtos). Dando un total de 7470 imágenes. Es importante mencionar que las imágenes mixtas no fueron consideradas ya que no se podía establecer qué máscara pertenecía a que clase de glóbulo blanco.

Existen tres transformaciones tenidas en cuenta: traslación, rotación y espejado. Para la traslación, se establecieron dos parámetros: uno de máximo y otro de mínima. En el caso de la traslación horizontal, un valor positivo significa una traslación hacia la derecha y el valor negativo una traslación hacia la izquierda. En cambio, en una traslación vertical, un valor positivo indica una traslación hacia arriba, mientras que el negativo indica el movimiento hacia abajo. En particular, el límite que se puso fue de -15 y 15 píxeles de mínima y máxima, respectivamente.

Respecto de la rotación, los límites colocados fueron entre 0 y 180° con paso de 1°. Además, se estableció que se podía realizar un espejado por el eje vertical y/o horizontal.

Para determinar qué transformaciones se realizan para una imagen dada, se utilizaron funciones que extraen muestras de una función uniforme. En el caso de las rotaciones y traslaciones se utilizaron los límites mínimos y máximos como límites de la función uniforme. Para el espejado, los límites eran entre 0 y 1, pero se colocaba una condición que si el valor era mayor a 0,5 se realizaba el espejado. Había un parámetro para el espejado sobre el eje vertical y otro para definir el espejado sobre el eje horizontal.

Cada vez que se realiza una transformación, como puede ser una rotación, puede ser que sobre ciertos píxeles no haya información. A veces, se coloca una constante en esos píxeles, por ejemplo 0. Sin embargo, en la presente tesis se decidió utilizar un modo espejado que refleja la información original por un eje como forma de llenar esos píxeles

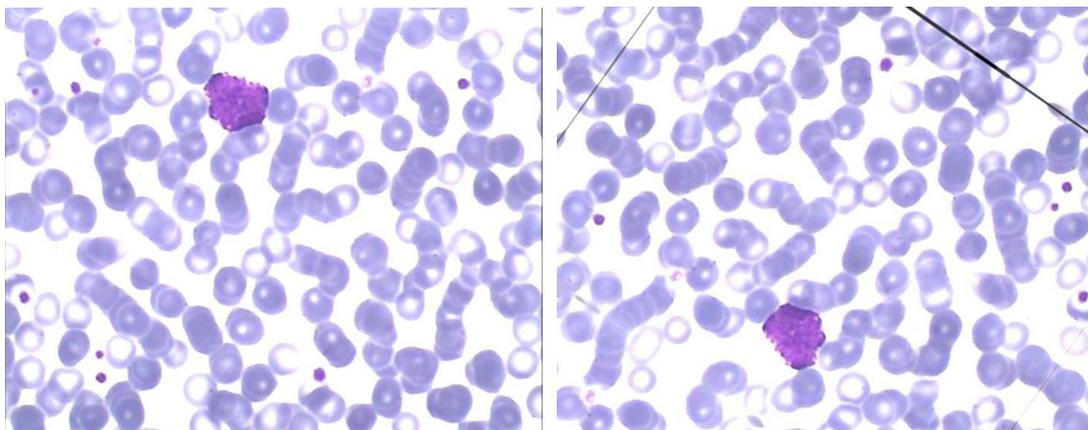


Ilustración 35: Ejemplo del método de espejado que se decidió emplear. (imagen 0 y su transformada)

El modo espejado genera glóbulos blancos que antes no estaban, lo que permite que en ciertas imágenes que había solo un glóbulo blanco, haya más de uno. Esto puede resultar beneficioso para la generalización en la tarea de detección.

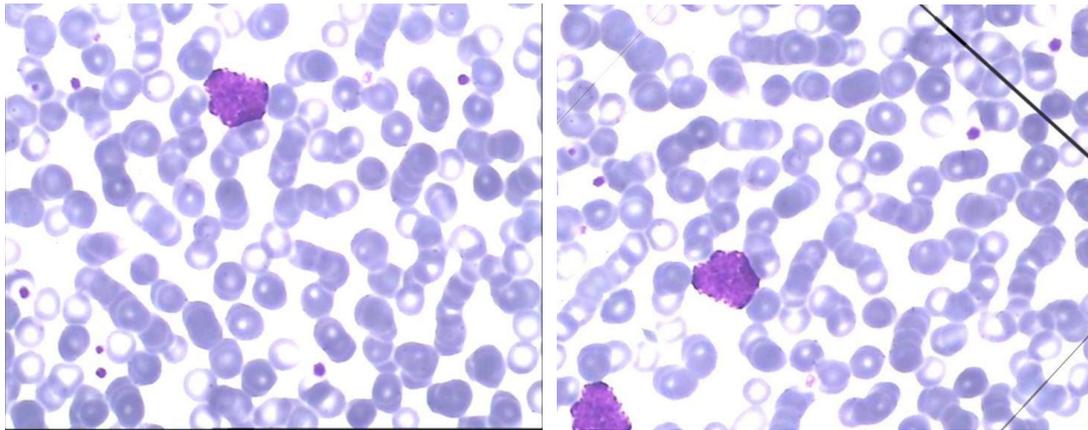


Ilustración 36: Ejemplo de glóbulo blanco espurio producto de la transformación realizada sobre la imagen. (imagen 0 y su transformada)

Posteriormente a la transformación de la imagen y la máscara, es necesario extraer de cada máscara transformada la cantidad de glóbulos blancos y su ubicación. Para eso, se hizo un procesamiento de estas que consistió primero en umbralizado, seguido de operaciones de morfología matemática y finalizando con un proceso de etiquetado.

- **Umbralizado:** se utiliza un umbral para generar una imagen binaria a partir de una imagen en escala de grises [19]. En particular, se definió un umbral de 10 para una imagen de 255 niveles luego de varias pruebas. El uso de ese umbral se debió a que se encontró que un valor menor generaba que hubiera mucho ruido en la imagen. Esto dificulta el post-procesamiento. Un umbral más alto generaba que haya mayores “huecos” en los glóbulos blancos y traía problemas para el etiquetado.
- **Morfología matemática:** son operaciones booleanas que se realizan entre la máscara binarizada y un elemento estructurante [19], que en este caso fue una matriz de unos de 2x2. El procedimiento es similar a la de los filtros en imágenes utilizando kernels. Las operaciones utilizadas fueron las siguientes:
  - Dilatación: es una de las operaciones básicas de este grupo. Se podría definir como una operación de OR entre la máscara y el elemento estructurante.
  - Apertura: Se podría definir como una operación de AND y posteriormente una de OR utilizando el mismo elemento estructurante.
  - Cierre: Se podría definir como una operación de OR, seguida de una de AND utilizando el mismo elemento estructurante.
- **Etiquetado:** es un proceso que involucra asignar una etiqueta a los pixeles del frente (o 1 por ser una imagen binaria) y recursivamente dar la misma etiqueta a los pixeles conectados o a sus vecinos. Este proceso se repite hasta que a todos los pixeles del frente se les haya asignado una etiqueta [19].

Es importante aclarar que las operaciones de morfología matemática se realizaron para poder unificar las estructuras de tal forma que cuando se realice el etiquetado, se encuentre una etiqueta por glóbulo blanco. La cantidad y orden de las operaciones de morfología matemática se realizaron pensando en un equilibrio entre el objetivo de unificación y el cuidado con los cambios en la máscara. Como resultado de este proceso, se encontró que las máscaras terminaban siendo más grandes que antes del mismo y por lo tanto las cajas delimitadoras serían mayores. Sin embargo, la diferencia se consideraba despreciable respecto del hecho de conseguir la unificación de las etiquetas.

Para cada etiqueta que identifica a un glóbulo blanco se extrae la posición mínima y máxima en ambas direcciones (vertical y horizontal) para extraer los límites de las cajas de las imágenes transformadas.

Luego de obtener las etiquetas, se procedió a reducir el tamaño de las imágenes por tres para que pasen a tener un tamaño de 192x240. Esto se hizo para reducir la dimensionalidad de estas y que sean

menos complejos los modelos necesarios para la detección de objetos (menos cantidad de parámetros). Uno de los riesgos de realizar esta técnica es que los objetos queden demasiado pequeños o que aquellas características que la red utiliza para la clasificación de los objetos detectados se pierdan. Esto fue tenido en cuenta a la hora de realizar esta reducción, pero se evaluó que no debería afectar los resultados dado que los glóbulos blancos son los objetos más grandes de las imágenes y que su tamaño era suficiente como para que el proceso fuera exitoso. Obviamente, como parte del proceso, se redujeron los valores de las coordenadas de las etiquetas y se verificó que siguieran perteneciendo a las mismas cajas.

El siguiente paso consistió en calcular el punto medio de las cajas (coordenadas  $x$  e  $y$  del centro), el alto y el ancho para que el formato de las coordenadas se pueda adecuar al modelo de YOLO.

Para que estos datos tengan el formato correcto era necesario ciertas transformaciones adicionales:

- El ancho y el alto se dividió por el tamaño en esa dimensión ( $x$  e  $y$ ) de la imagen para que estos valores queden expresados en términos de la imagen global y sus valores sean entre 0 y 1 siguiendo la línea de YOLO.
- Las coordenadas de los puntos medios deben estar expresadas de forma relativa a la celda en el que se encuentra dicho punto. Además, es necesario establecer para cada punto medio en qué celda se encuentra. Para esto se siguieron los siguientes pasos:
  - Se definió que la grilla para la detección sea de  $24 \times 30$ , con cada celda siendo de  $8 \times 8$ . Esto se hizo pensando en que el tamaño de la celda fuera lo suficientemente pequeño para poder detectar dos glóbulos blancos cercanos.
  - Cada coordenada de cada punto medio se dividió por 8. El cociente indicaba la celda en la cual se ubicaba la coordenada y el resto la coordenada relativa respecto del inicio de la celda.
  - La coordenada relativa del centro (el resto de la división) se divide por 8, para que la misma esté expresada como un porcentaje respecto del total de la celda y para que el valor sea entre 0 y 1. Este es un aspecto crucial para la función de costo de YOLO.

Como ya se discutió anteriormente, es importante separar el set en tres partes: entrenamiento, prueba y validación.

Esta separación se hace a partir de las imágenes originales. Esto quiere decir que, si una imagen original es designada para el set de entrenamiento, entonces todas las imágenes producto del proceso de data augmentation de esa imagen también pertenecerán al set de entrenamiento. Este factor es crucial a la hora de validar los modelos y su funcionamiento porque es necesario que nunca haya entrenado con la información proveniente de la imagen para que la conclusión tenga validez. La razón detrás de esta condición es que, aunque la imagen producto del proceso de data augmentation puede parecer diferente, en realidad contiene la misma información que la imagen original, pero presentada de diferente forma.

Para el set de prueba se decidió tomar 5 imágenes originales de cada clase elegidas al azar. Se eligió 5 porque representaba aproximadamente el 10% de cada clase, aunque en el caso de los eosinófilos representa un porcentaje mayor por la poca cantidad de estos. Se decidió que sea igual la cantidad de imágenes para que sea balanceado el conjunto e intentar de minimizar el sesgo de este al evaluar los resultados. El set de validación se armó de la misma forma y el de entrenamiento fue el resultado de lo que no fue incluido en ninguno de los dos conjuntos.

## Herramientas

La presente tesis se realizó en el lenguaje de programación Python. Este es un lenguaje interpretado, interactivo y orientado a objetos. Para el preprocesamiento de las imágenes se usaron una serie de librerías, como Numpy y Scipy, entre otras.

El desarrollo de los modelos, así como su entrenamiento y las predicciones se hicieron utilizando las librerías de Keras [37] y Tensorflow [38].

Tensorflow fue originalmente desarrollado por el equipo de Google Brain. Fue pensado como un motor para cálculos numéricos utilizando grafos de flujo de datos. Esta librería es utilizada por un gran rango de usuarios con diferentes necesidades. Aquellos investigadores que necesitan un gran nivel de detalle y control sobre sus modelos puede utilizar Tensorflow Core que permite ese manejo a bajo nivel del modelo. También existen otras API de alto nivel que resultan más fáciles de usar y aprender [16].

Una de esas API es Keras que provee un set de abstracciones más intuitivas que hacen que sea más fácil configurar las redes neuronales. La motivación primaria de Keras es permitir una experimentación rápida con redes neuronales y que se pueda ir de la idea a los resultados de la forma más rápida posible.

Dado que estos modelos requieren una gran cantidad de recursos computacionales, se decidió desarrollarlos en Google Colab que es un producto de Google Research. Colab permite que cualquier usuario pueda escribir y ejecutar código de Python en un explorador web sin necesidad de configurar el ambiente. Además, el código se ejecuta en servidores de Google lo que permite aprovechar el poder del hardware de Google, incluidas GPUs y TPUs, sin importar el poder de la computadora que se está utilizando.

Por último, para los gráficos y la visualización de los resultados se utilizó Tableau que es un software que permite generar diferentes visualizaciones de datos de forma rápida.

## Modelos

Los modelos que se probaron se explican a continuación. Como ya se mencionó, la estructura básica de cada capa convolucional va a referir a un bloque de estos tres elementos:

- La capa convolucional propiamente dicha que realiza la operación de convolución. Los parámetros más importantes de esta capa:
  - Tamaño del kernel: 3x3 o 1x1
  - Número de filtros: 32, 64 o 96
- La capa de batch normalization
- La capa de activación: Relu o LeakyRelu

Luego la otra capa que aparece en la esquematización del modelo es Max Pooling. Los parámetros de esta son:

- Tamaño kernel: 2x2
- Paso: 2

Esto significa que se queda con el máximo de zonas de 2x2 y se va moviendo a un paso de 2, lo que implica que no hay superposición de las zonas analizadas. Esta configuración de la capa de Max Pooling es lo que hace que las dimensiones de la salida de la capa sea la mitad que la entrada. Para que la

grilla final sea de 24x30, es necesario que haya 3 capas de Max Pooling con esta configuración en todo el modelo.

Por último, al final del modelo se divide en tres salidas. Esto se hace para obtener los diferentes scores que requieren de diferentes funciones de activación. Por un lado, está el valor de confianza en que haya un objeto en esa celda, para lo cual se utiliza como función de activación una función sigmoidea. Después hay otra salida de 5 valores por celda que es el valor de confianza de clase para cada clase. Para esta tarea, la función de activación es una softmax. Por último, hay una salida de 4 valores por celda que identifica a las coordenadas de la caja en caso de que haya un objeto. Esta salida se obtiene después de utilizar una ReLU porque se considera que no tienen sentido las coordenadas negativas.

Luego de obtener cada una de esas salidas, se concatenan al final del modelo para obtener una salida unificada de 10 valores por cada celda.

El diagrama que esquematiza los modelos probados es el siguiente:

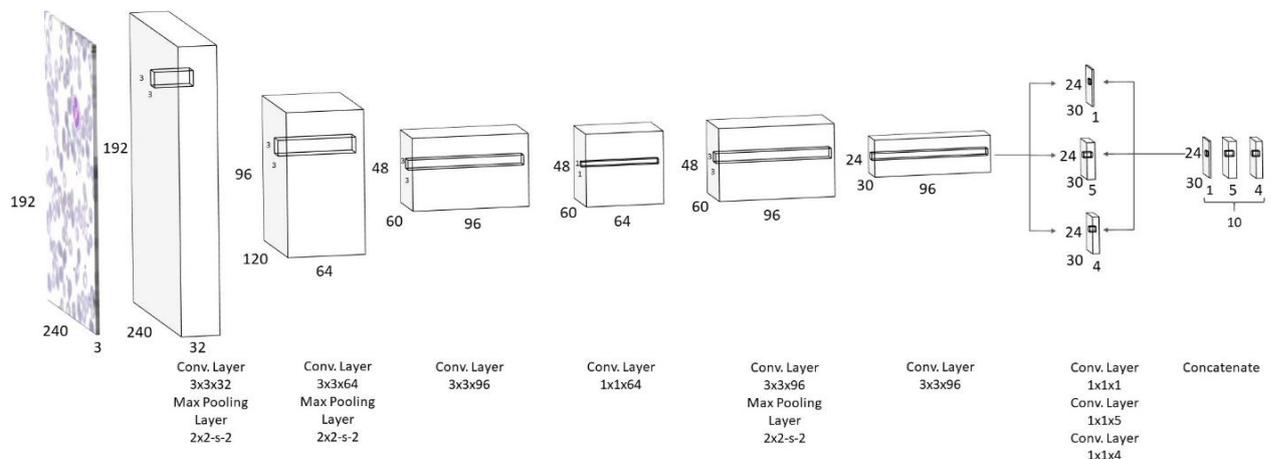


Ilustración 37: Esquema general de los modelos de la presente tesis y de los parámetros de las diferentes capas

Para los diferentes modelos se probaron la combinación de tres factores:

- Step size del cyclical learning rate:
  - 0.08 epoch
  - 4 epoch
- Tamaño de mini lote:
  - 50 imágenes
  - 150 imágenes
- Función de activación:
  - ReLU
  - LeakyReLU

Se hizo una separación definiendo dos modelos, uno aquel cuya función de activación es la ReLU y otro el que es la LeakyReLU. La combinación de los otros hiperparámetros (step size y batch size) da origen a las variantes del modelo. A continuación, una tabla que define qué parámetros e hiperparámetros corresponde a cada modelo y variante:

Modelo	Variante	Step size	Batch size	Función de activación
Modelo 1	Variante 1	0.08	50	ReLU
	Variante 2	0.08	150	ReLU
	Variante 3	4	50	ReLU
	Variante 4	4	150	ReLU
Modelo 2	Variante 1	0.08	50	LeakyReLU
	Variante 2	0.08	150	LeakyReLU
	Variante 3	4	50	LeakyReLU
	Variante 4	4	150	LeakyReLU

Tabla 1: Descripción de hiperparámetros correspondientes a cada modelo y variable

Cada variante se entrenó con los parámetros indicados. Además, se utilizó una función de Keras denominada Early Stopping que detiene el entrenamiento de los modelos basado en una cierta condición. Esto permite evitar el sobreajuste al intentar detectar el momento en el cual el entrenamiento empieza a ser perjudicial. Para esto, se pone una condición sobre alguna métrica del set de validación. En particular, para la presente tesis se estableció que la condición de corte era si no mejoraba la función de costo sobre el set de validación en 10 epochs.

Otra función utilizada fue la de Model Checkpoint que permite guardar los pesos (los parámetros) de la red para un determinado momento del entrenamiento. En este caso, se decidió utilizar para aquella configuración que durante el entrenamiento lograba el menor valor de la función de costo para el set de validación.

En la tabla anterior, se plantearon los diferentes modelos y variantes que se buscaban probar en el marco de la presente tesis. La Variante 3 y 4 del Modelo 2 se decidió excluir y esos modelos no fueron entrenados luego de observar el pobre desempeño de dichas variantes del Modelo 1, como se verá a continuación en la sección de resultados.

## Resultados

Cuando se mide el rendimiento de un modelo de aprendizaje profundo es importante hacerlo con los tres sets como ya se estableció anteriormente. Con la medición de las métricas sobre el conjunto de entrenamiento es que se define que el modelo no esté subajustando. Luego, utilizando las métricas sobre los datos del grupo de validación se ajustan y definen los hiperparámetros que utilizará el modelo para las predicciones. Finalmente, con el grupo de prueba es que se evalúa que el modelo no esté sobreajustando.

Otro elemento ya mencionado, es el umbral del valor de confianza general. Este número es lo que define que el modelo prediga que hay o no un objeto en una determinada celda de la grilla. Si el valor de confianza es mayor al umbral, entonces esa predicción es evaluada junto con todas las otras predicciones con el algoritmo de Non Maximum Suppression. Si el valor es menor al umbral, se descarta dicha predicción. Es importante mencionar esto porque se puede ver que hay dos instancias en las cuáles se puede descartar una predicción: ya sea porque no supera el umbral y se considera que no hay un objeto en esa celda o porque se considera que esa predicción es redundante y fue realizada por otra celda con un nivel de confianza mayor.

Como ya se estableció anteriormente, en la presente tesis se va a utilizar el  $F_2$  score como métrica para medir el rendimiento de los modelos. Se van a utilizar dos variantes de este score para evaluar los mismos. La primera es para medir qué tan buenos son los modelos reconociendo la presencia de un glóbulo blanco, sin importar si está correctamente clasificado. Se utilizó este enfoque dado que, como se mencionó en la introducción, estos sistemas serían semi supervisados y por lo tanto siempre

se necesitaría de un técnico especialista que defina que está correctamente clasificado dado que se trata de un estudio médico. Es por esta situación que resulta altamente relevante que el sistema tenga un buen rendimiento estableciendo la presencia de los glóbulos blancos. A este score se le denominará  $F_2$  score de localización.

Por otro lado, se va a utilizar el promedio de los  $F_2$  score para cada clase como métrica del rendimiento en la detección de objetos (localización y clasificación de estos) que es el procedimiento estándar para problemas con múltiples etiquetas. A este score se lo denominará  $F_2$  score de detección.

A continuación, se muestran los resultados de los Modelos 1 y 2, con cada subtipo, y el  $F_2$  score de localización en función de los umbrales del valor de confianza para el set de entrenamiento.

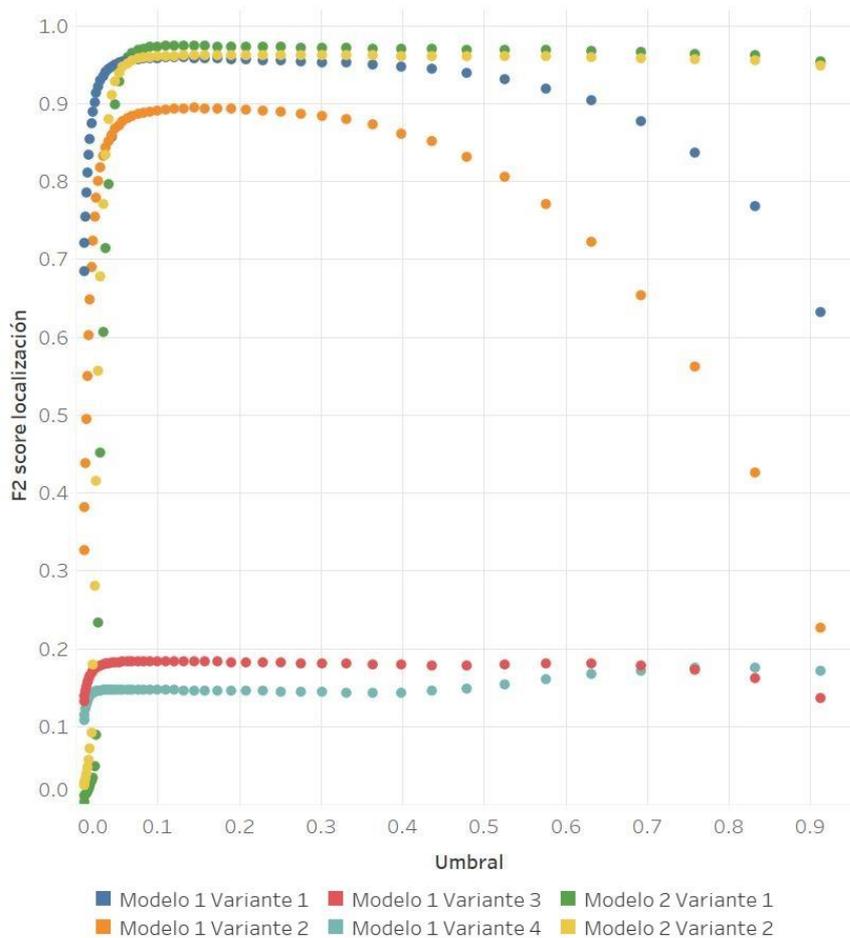


Ilustración 38:  $F_2$  score de localización en función del umbral para cada combinación de Modelo y Variante para el conjunto de entrenamiento

Se puede ver que los subtipos 3 y 4 del Modelo 1 tienen un problema de subajuste. Esto se nota cuando se evalúa la precisión y la sensibilidad porque los valores de precisión son bajos para todos los umbrales. Este fenómeno se debe a que el modelo realiza un montón de predicciones, muchas de las cuales son falsas, de ahí los falsos positivos. Esto se puede deber a que el modelo no haya logrado aprender cuáles son las características para identificar los glóbulos blancos. Parecería estar relacionado con el step size y una de las posibilidades es que al ser más alto este valor, no haya variado la tasa de aprendizaje lo suficientemente rápido y el optimizador se haya quedado en algún punto silla de la función de costo. Dado estos resultados con el set de entrenamiento es que se van a excluir estos dos subtipos de los análisis posteriores.

Otro elemento que se puede ver de los gráficos es que las curvas del Modelo 1 parecen descender después de alcanzar los máximos, mientras que las del Modelo 2 se mantienen. Este hecho se abordará más adelante cuando se evalúe la distribución de los niveles de confianza.

A continuación, se muestran los gráficos del F<sub>2</sub> score de detección en función de los umbrales para el set de entrenamiento de cada uno de los subtipos.

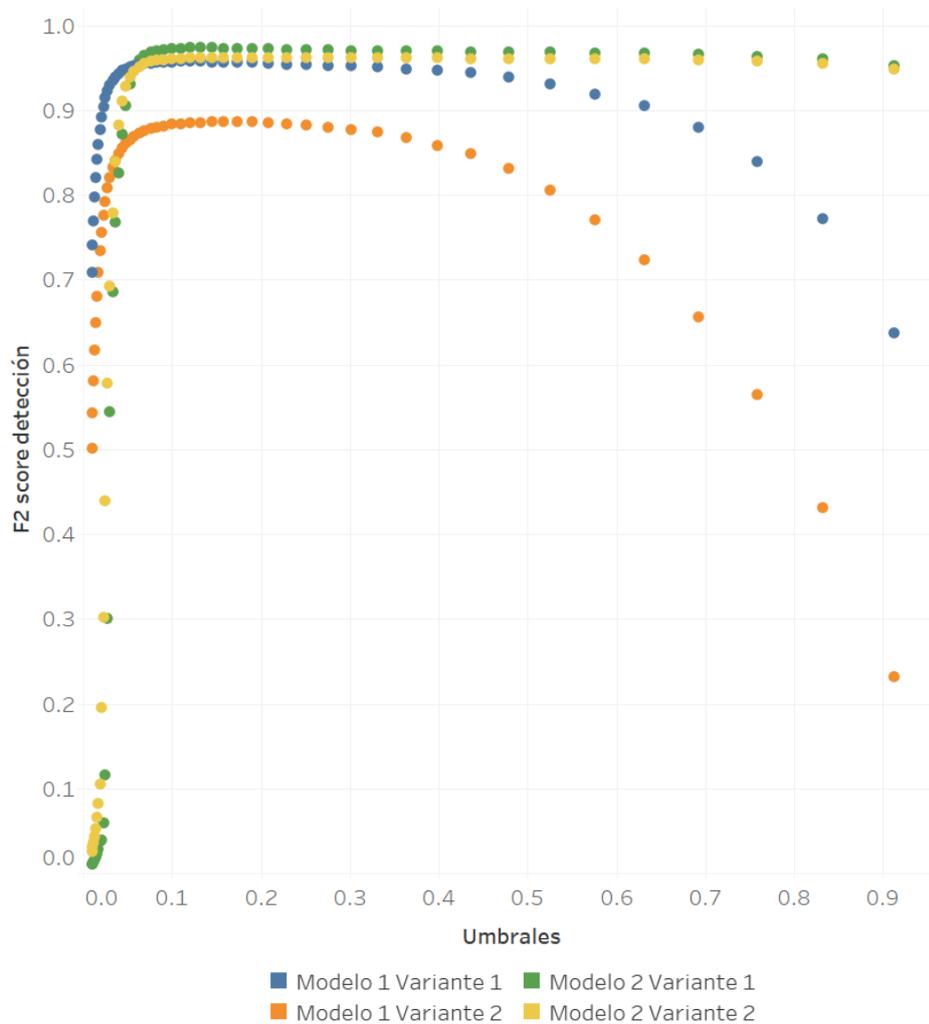


Ilustración 39: F<sub>2</sub> score de detección en función del umbral para cada Modelo y Variante para el conjunto de entrenamiento

Se puede observar que la morfología de las curvas de detección y localización es similar y como se refleja en la siguiente imagen también los valores. A continuación, se muestra el ejemplo para el Modelo 1 Variante 1, pero en el resto de los modelos se observan características similares y los gráficos se encuentran en el Anexo A.

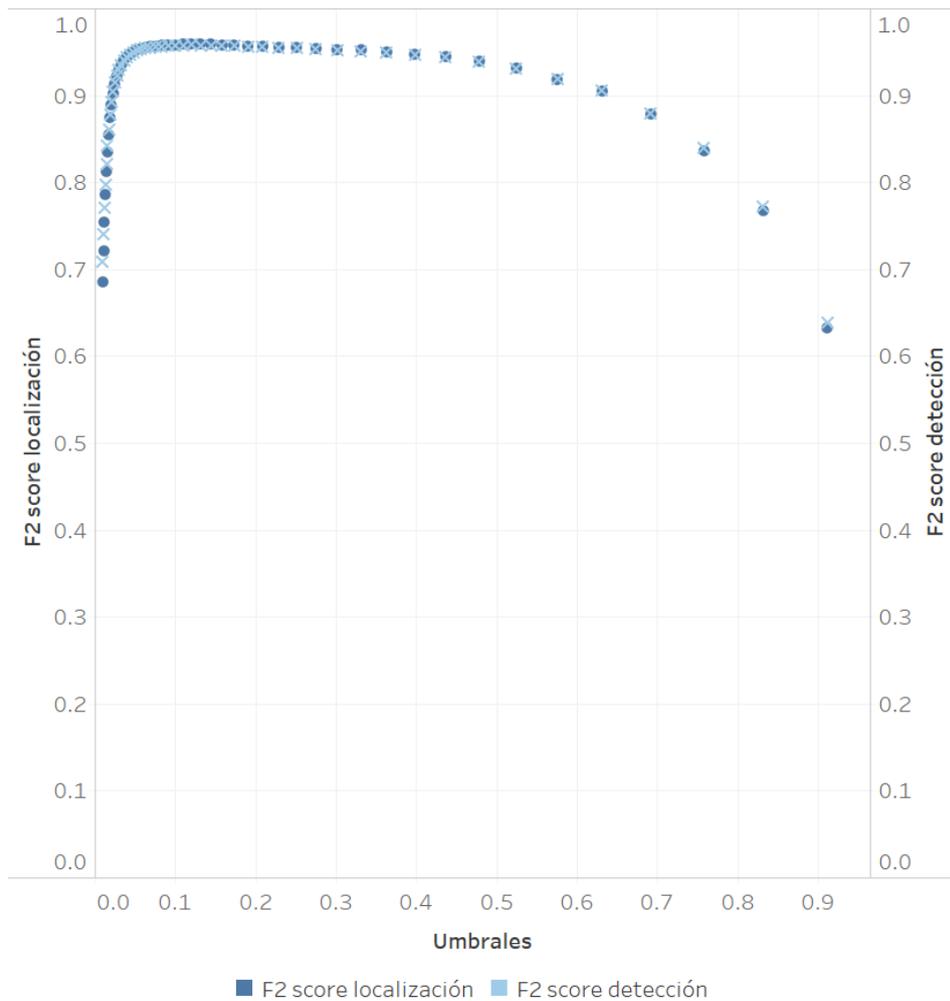


Ilustración 40:  $F_2$  score de localización y de detección en función del umbral para el Modelo 1 Variante 1 para el conjunto de entrenamiento

A partir de las curvas sobre los datos de entrenamiento, se puede observar que los cuatro modelos seleccionados lograron aprender a detectar los glóbulos blancos para este conjunto de datos. La pregunta que surge es si los modelos lograron aprender a detectar un glóbulo blanco de forma generalizada, o sea verificar que no haya sobreajuste. Para eso se va a hacer el análisis sobre las curvas del set de validación y posteriormente sobre el de evaluación.

Otra tarea importante para realizar con los resultados de las curvas sobre los datos de validación es encontrar el umbral óptimo para cada modelo. A continuación, se muestran las curvas para el  $F_2$  score de localización y de detección:

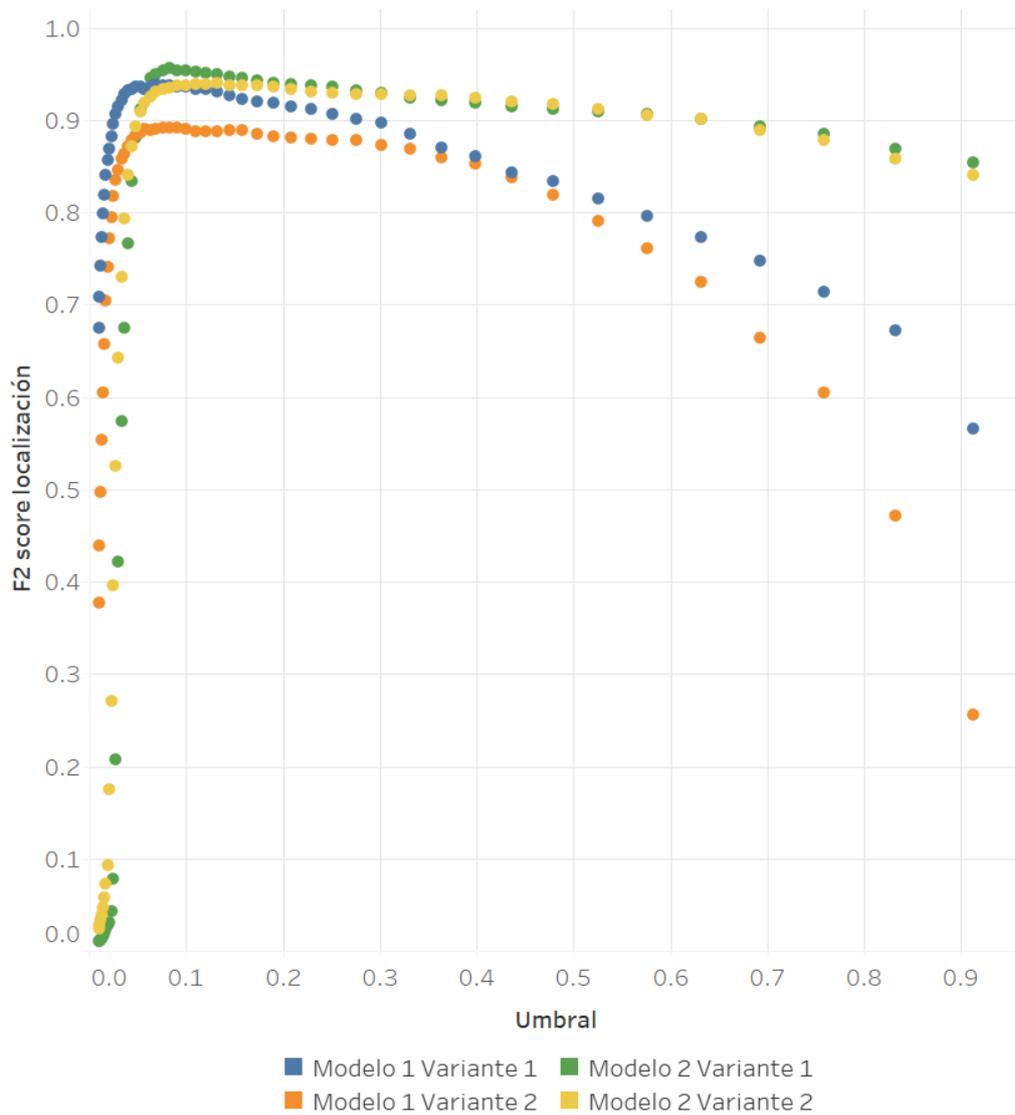


Ilustración 41:  $F_2$  score de localización en función del umbral para cada combinación de Modelo y Variante para el conjunto de validación

Como se puede observar en el gráfico, el rendimiento de los cuatro modelos es similar, aunque las variantes 1 de ambos modelos parecen tener un desempeño mejor que las variantes 2. Por otro lado, las variantes del modelo 2 parecerían superar a sus respectivas contrapartes del modelo 1. Estos indicios se ven acentuados cuando se observan los valores del  $F_2$  score detección.

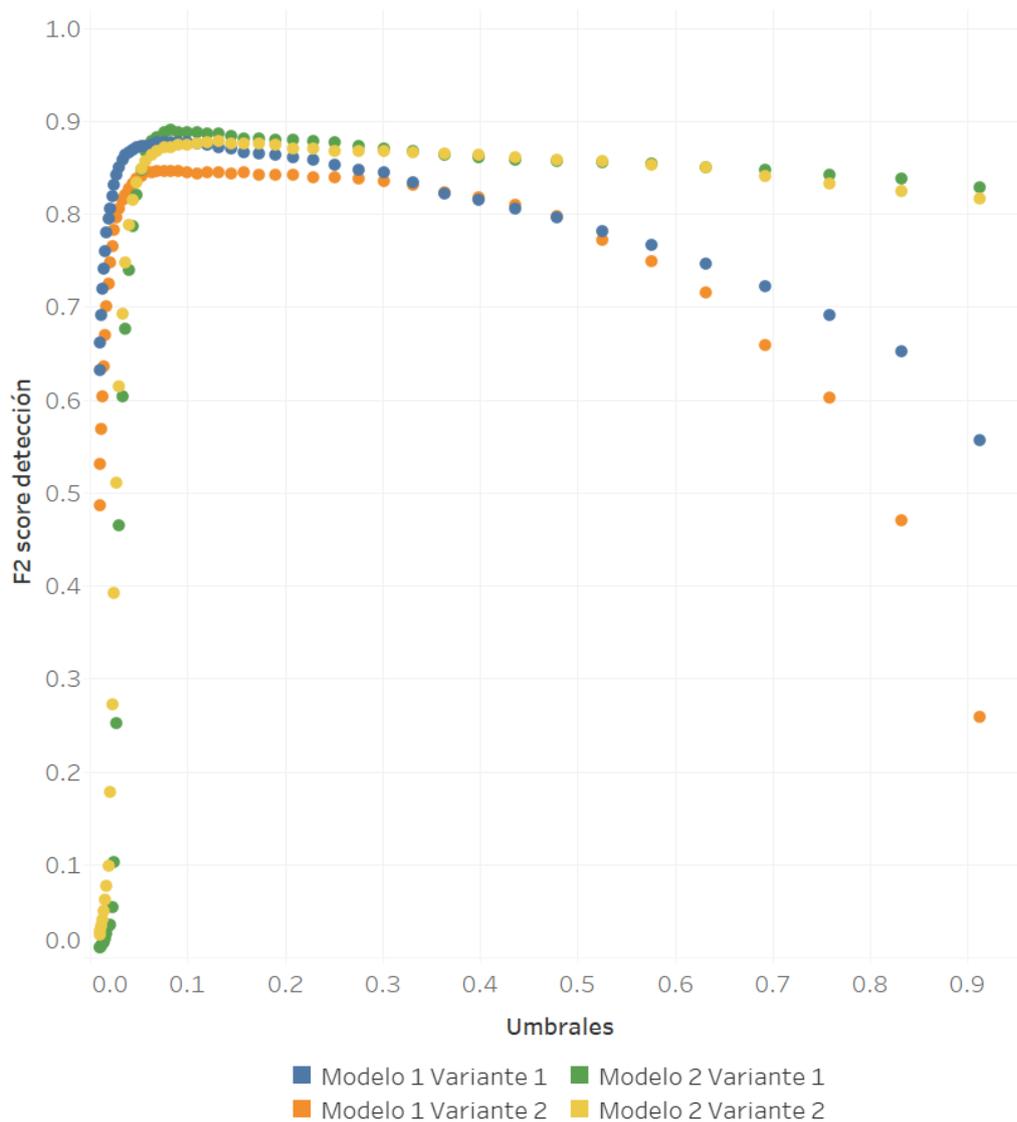


Ilustración 42:  $F_2$  score de detección en función del umbral para cada combinación de Modelo y Variante para el conjunto de validación

En este gráfico, se puede observar también que para las variantes del modelo 2 parecen tener un desempeño más parecido entre sí, a diferencia de las variantes del modelo 1. Estas últimas parecen tener un comportamiento similar entre variantes (dado que la morfología es similar), pero con un desempeño en la clasificación bastante diferente.

Otra característica importante que analizar para evaluar si existe sobreajuste del modelo es observar la curva para el set de entrenamiento y para el de validación. A continuación, se muestra la curva para el modelo 1 y variante 1, cuyos resultados son similares a los de los otros modelos y variantes como se puede observar en el Anexo A:

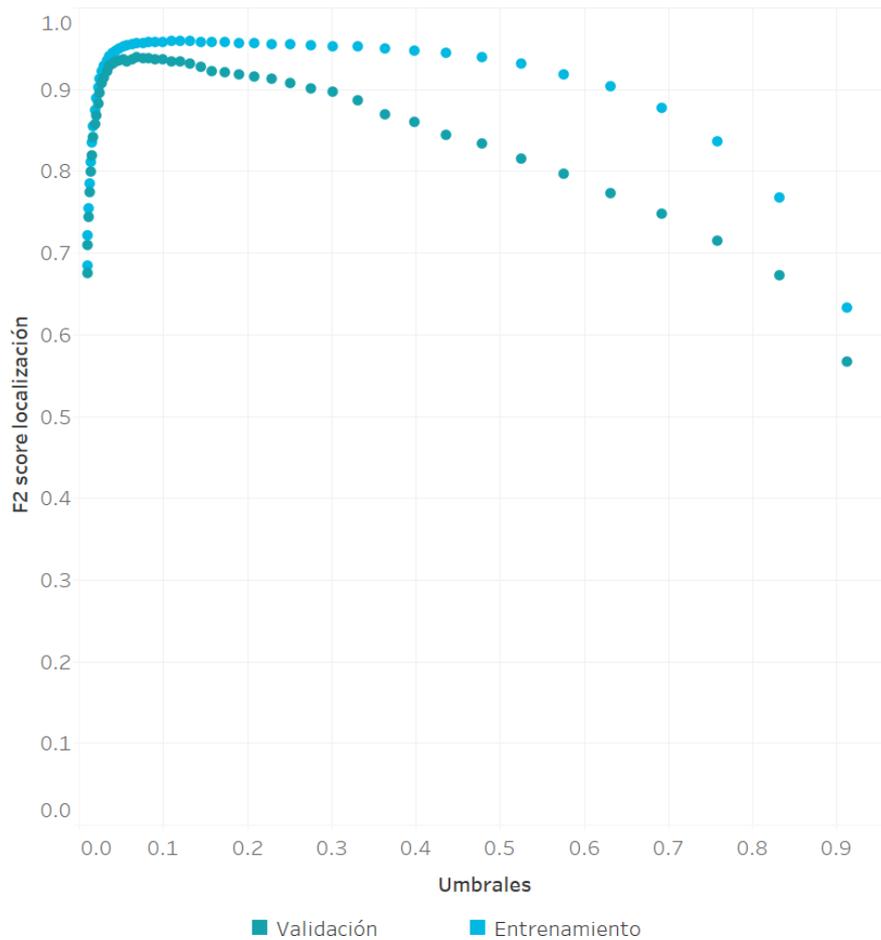


Ilustración 43: Comparación de las curvas del F<sub>2</sub> score de localización en función del umbral entre el conjunto de entrenamiento y el de validación

Del gráfico se puede extraer que el rendimiento es bastante similar en ambos conjuntos para la tarea de localización, o sea detectar dónde hay o no un objeto. Aunque para el set de validación es ligeramente menor el F<sub>2</sub> score lo que es esperable dado que el modelo nunca vio dichas imágenes. En el caso del F<sub>2</sub> score de detección la distancia entre ambas curvas es mayor. Esto es esperable porque además de la tarea de localización se suma la tarea de clasificación para evaluar el desempeño del modelo. A continuación, se muestran los resultados para el mismo modelo y variante que el gráfico anterior:

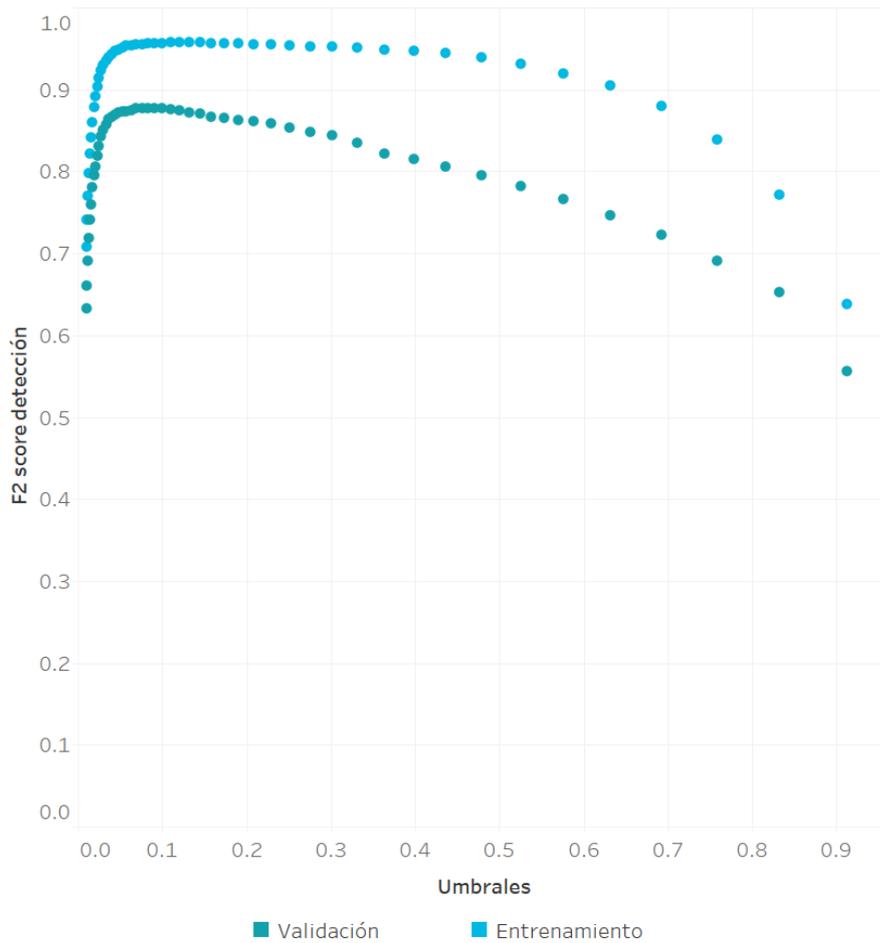


Ilustración 44: Comparación de las curvas del  $F_2$  score de detección en función del umbral entre el conjunto de entrenamiento y el de validación.

Como ya fue comentado, se pueden ver mayores diferencias entre las curvas para el set de entrenamiento y el de validación. Es importante recordar que el  $F_2$  score de detección consiste en un promedio entre los  $F_2$  score de detección de cada clase. Por ende, puede ser que el rendimiento de cada clase sea diferente y como se verá más adelante, las causas de dichas diferencias en rendimiento pueden ser distintas para cada clase.

Al igual que con el set de entrenamiento, se muestran a continuación las curvas de  $F_2$  score de detección y localización en un mismo gráfico para el modelo 1 variante 1.

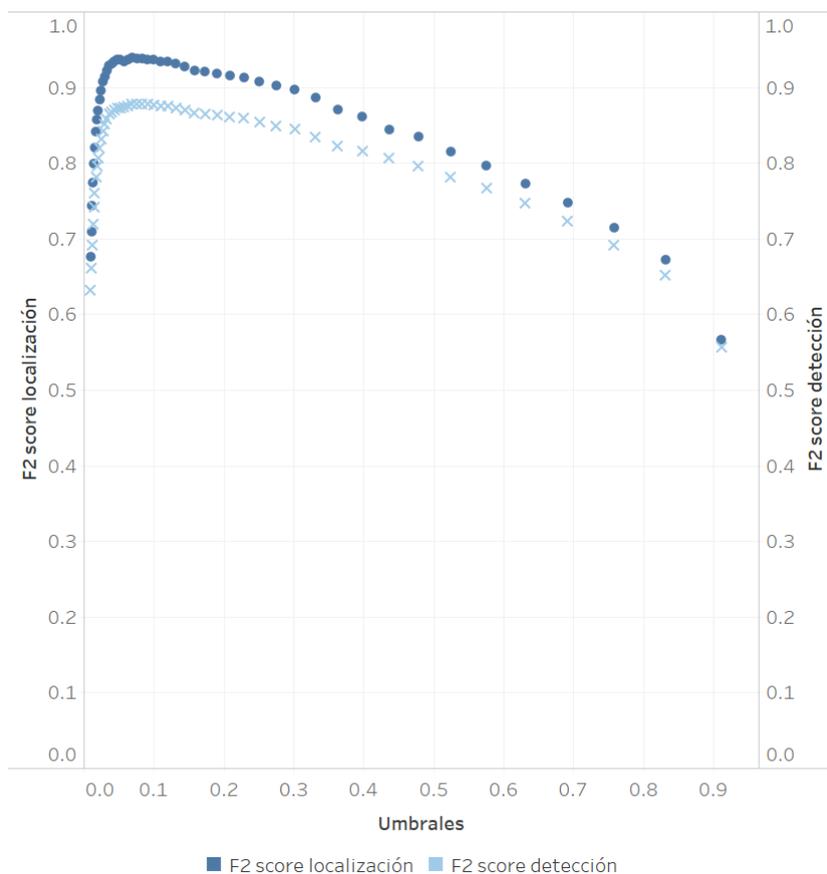


Ilustración 45:  $F_2$  score de localización y de detección en función del umbral para el Modelo 1 Variante 1 para el conjunto de validación

A diferencia del gráfico para el set de entrenamiento, se pueden ver mayores diferencias entre la curva de  $F_2$  score de localización y de detección, aunque esto se podía observar al ver la diferencia entre las curvas de  $F_2$  score de detección entre el set de entrenamiento y validación.

Estos gráficos permiten evaluar el rendimiento en general de los modelos para los diferentes umbrales. Para elegir el umbral óptimo de cada modelo-variante, se eligió el umbral que hacía máximo el  $F_2$  score de localización para el conjunto de imágenes de validación. Se utiliza el  $F_2$  score de localización porque, como ya se mencionó en repetidas ocasiones a lo largo del presente documento, la tarea de detección de glóbulos blancos se podría considerar una tarea semi supervisada por lo que, aunque la clasificación resulta importante; es de vital importancia la localización de los objetos ya que la clasificación se puede corregir a posteriori en caso de ser necesario. A continuación, los umbrales óptimos y  $F_2$  score de localización y de detección para los cuatro modelos en cuestión:

Modelo	Variante	Umbral	$F_2$ score localización	$F_2$ score detección
Modelo 1	Variante 1	0.0692	0.9386	0.8776
	Variante 2	0.0912	0.8919	0.8465
Modelo 2	Variante 1	0.0832	0.9570	0.8902
	Variante 2	0.1318	0.9408	0.8785

Tabla 2: Umbral óptimo para cada Modelo y Variante con sus valores de  $F_2$  score de localización y de detección

De la misma, se puede observar que los umbrales óptimos para las variantes del Modelo 2 son mayores a los del Modelo 1. Lo mismo ocurre entre los umbrales de la variante 2 respecto de la variante 1 de cada modelo. Por otro lado, evaluando el rendimiento, las variantes 1 de los modelos parecen tener un mejor rendimiento que las variantes 2 y lo mismo ocurre entre las variantes del Modelo 2 respecto de la misma variante del Modelo 1. Por último, del análisis de las curvas, así como

de la tabla se puede ver que parecería ser que las variantes del Modelo 2 tienen un rendimiento más parejo entre sí que las variantes del Modelo 1.

A continuación, se analizará en mayor detalle el rendimiento de las variantes 1 de ambos modelos ya que son las de mejor desempeño. Primero que nada, se mostrará las matrices de confusión de ambos modelos para el umbral óptimo.

Predicción \ Etiqueta	Basófilo	Eosinófilo	Linfocito	Monocito	Neutrófilo	No hay objeto
Basófilo	133	26	1	1	0	13
Eosinófilo	9	146	3	3	1	27
Linfocito	0	1	185	0	0	11
Monocito	2	0	1	148	2	5
Neutrófilo	1	0	0	3	178	4
No hay objeto	9	3	10	12	2	

Tabla 3: Matriz de confusión del Modelo 1 Variante 1 utilizando el conjunto de validación para el umbral óptimo de 0.0692

Predicción \ Etiqueta	Basófilo	Eosinófilo	Linfocito	Monocito	Neutrófilo	No hay objeto
Basófilo	142	21	0	3	0	8
Eosinófilo	16	150	1	8	0	14
Linfocito	0	0	185	3	0	9
Monocito	0	0	2	150	2	4
Neutrófilo	0	0	0	4	180	2
No hay objeto	14	14	3	16	0	

Tabla 4: Matriz de confusión del Modelo 2 Variante 1 utilizando el conjunto de validación para el umbral óptimo de 0.0832

Para analizar las matrices de confusión y los siguientes gráficos se pasarán a definir ciertos términos que se utilizarán:

- **Correctos:** Aquellas predicciones que sea correcta la localización y la clasificación
- **Error Clase:** Aquellas predicciones que correspondan con la etiqueta de un glóbulo blanco por su ubicación, pero sea errónea la clasificación de este.
- **Falso Positivo:** Aquellas predicciones que no correspondan a una etiqueta y por lo tanto se determina que no había ningún glóbulo blanco en ese lugar.
- **Falso Negativo:** Aquellas etiquetas que no corresponden con una predicción. Esto significa que en ese lugar había un glóbulo blanco que el modelo no pudo detectar.

Respecto de las matrices de confusión, en las columnas se muestran las predicciones de los modelos, mientras que en las filas las etiquetas del conjunto de datos. Por lo tanto, la última fila muestra los falsos positivos predichos por el modelo y en la columna final los falsos negativos.

Comparando las matrices de confusión, se puede observar que las diferencias entre el Modelo 1 Variante 1 y Modelo 2 Variante 1 (Modelo 1 y Modelo 2, respectivamente de ahora en más) se encuentran en la distribución de los errores de clase, pero especialmente en la distribución de los falsos negativos y los falsos positivos. A continuación, un gráfico de barras comparativos de la cantidad de elementos perteneciente a cada categoría para todo el conjunto de datos y para cada clase en particular.

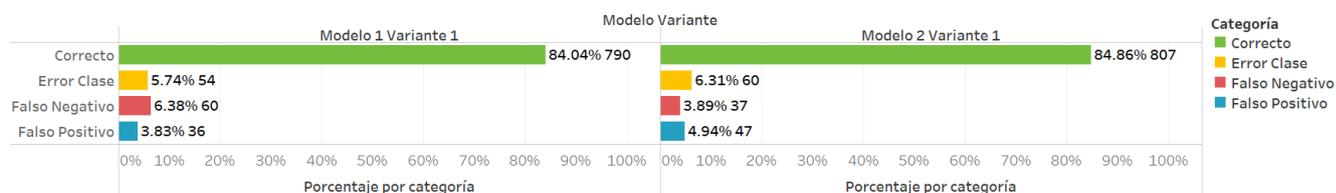


Ilustración 46: Resultados de la detección de ambos modelos para el total de las imágenes del conjunto de validación

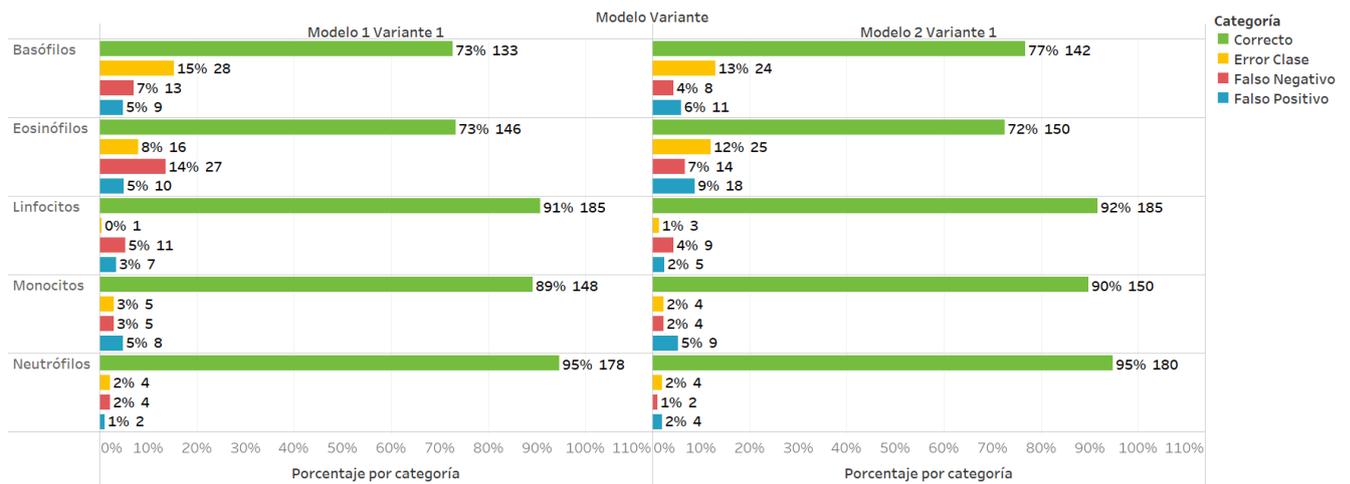


Ilustración 47: Resultados de la detección de ambos modelos por clase para las imágenes del conjunto de validación

Los gráficos anteriores muestran el porcentaje de predicciones que pertenecen a cada clase. Se decidió utilizar el porcentaje ya que cada modelo tiene un número diferente de predicciones y por lo tanto podía ser más difícil de comparar los resultados entre sí. En el segundo gráfico, se evaluaron los totales por clase. La clase tiene que ver con la clase de glóbulos blancos que realmente tenía la imagen, no lo predicho.

En el primer gráfico se puede observar que el total de predicciones correctas es similar, aunque es mayor para el Modelo 2. Se observan mayores diferencias en la cantidad de Falsos Negativos y de Falsos Positivos. Siendo los primeros mayores para el Modelo 1, mientras que los segundos es mayor para el Modelo 2. Parecería que el mejor desempeño en  $F_2$  score del Modelo 2 es a costa de un mayor número de Falsos Positivos. Esto se puede ver también en la matriz de confusión más claramente en la cantidad de Falsos Positivos ya que para todas las clases es mayor que los del Modelo 1.

El segundo gráfico aporta una información adicional para entender las diferencias en el rendimiento de ambos modelos. Se puede observar que ambos modelos tienen un desempeño peor para las imágenes que contienen basófilos y eosinófilos, donde se ve repetido el patrón antes descrito sobre el comportamiento comparado de ambos modelos.

Para entender si el rendimiento era similar para todas las imágenes de esa clase, se hizo el mismo gráfico, pero a nivel imagen original. El término imagen original refiere a la imagen que venía en el set original. Todas las imágenes obtenidas a partir de las transformaciones realizadas por el proceso de data augmentation corresponden a una imagen original. Esto quiere decir que cada imagen transformada tiene una correspondencia con solo una imagen original y es esa la que se utiliza para agrupar todas las imágenes obtenidas por el proceso descrito anteriormente.

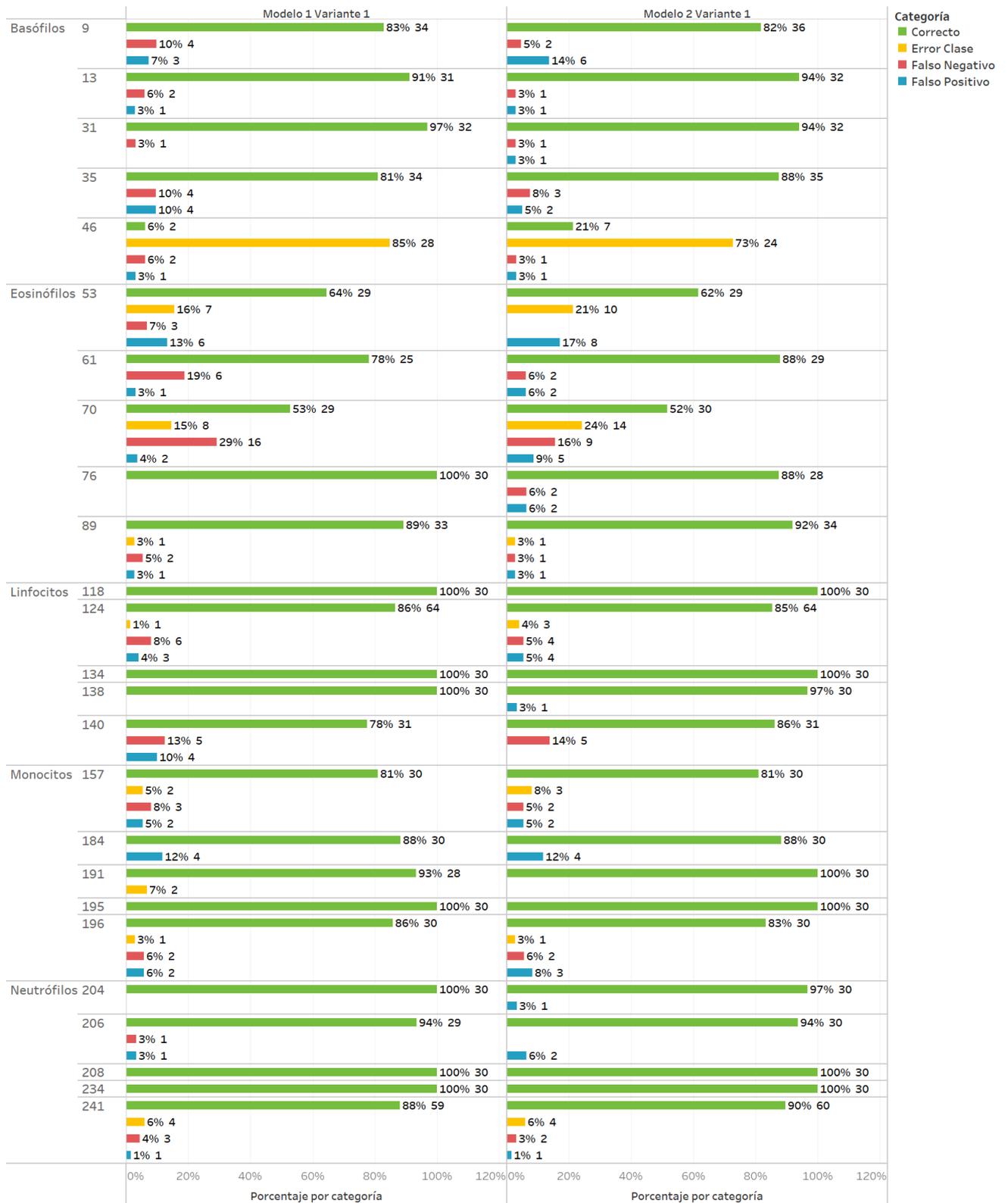


Ilustración 48: Resultados de la detección de ambos modelos agrupados por imagen original y clase del conjunto de validación

En este gráfico se puede observar que en general el patrón es similar ya que las predicciones hechas por el Modelo 1 suelen tener un número mayor de Falsos Negativos y las del Modelo 2 mayores Falsos Positivos. Sin embargo, se observan ciertas diferencias y un rendimiento despereado entre las diferentes imágenes que se abordará a continuación.

Más allá de las diferencias que se pueden observar comparando ambos lados del gráfico, hay dos imágenes que resaltan al analizarlo que son la imagen 46 y la imagen 70.

La imagen 46 es la única imagen para la cual el porcentaje de predicciones con error de clase supera a las predicciones correctas. Además, analizando qué clases predice en esos casos se observa que son todos eosinófilos y que es la única imagen de basófilos donde cualquiera de los modelos predice otra clase. A continuación, la imagen 46 original.

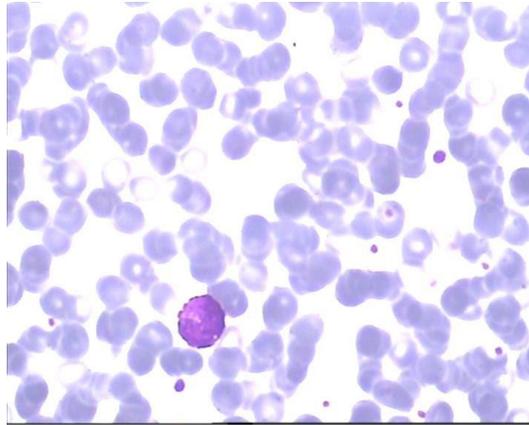


Ilustración 49: Imagen 46 del set original

Por otro lado, se muestran algunas de las imágenes de basófilos de entrenamiento.

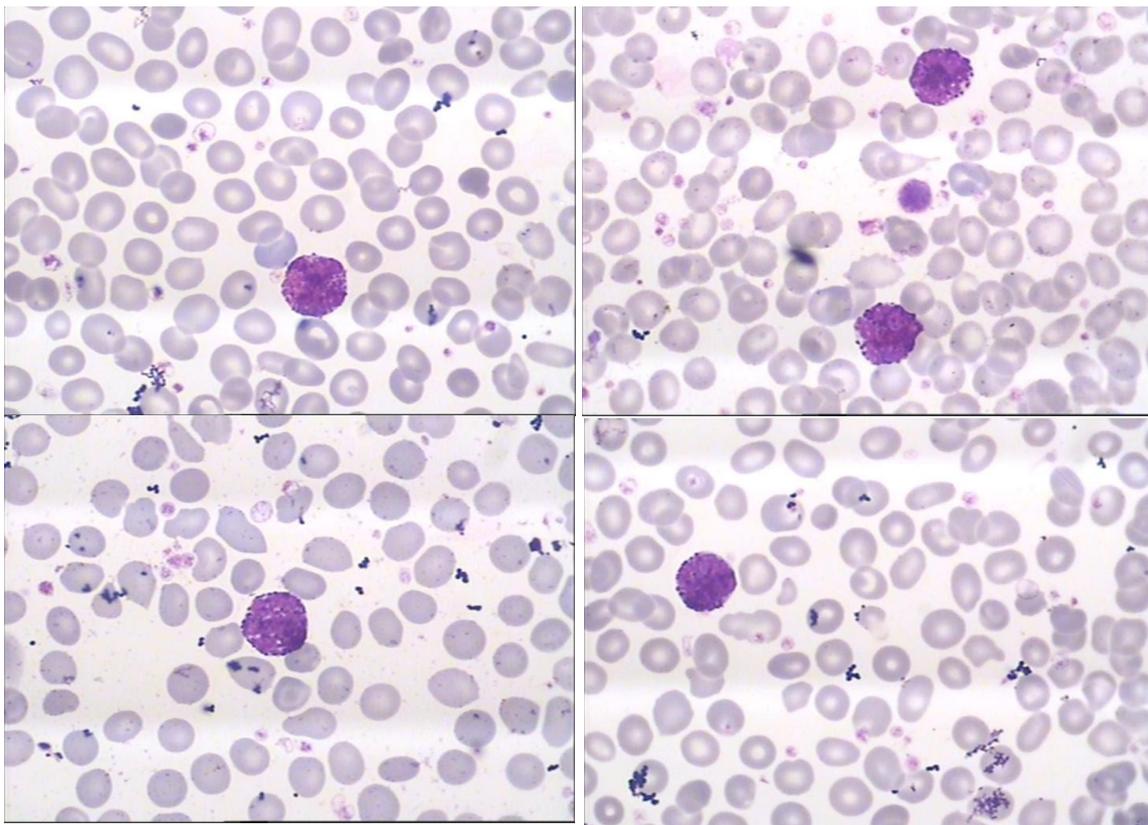


Ilustración 50: De izq. a der. Imágenes 5 a 8 del conjunto original

Una de las cosas que se puede ver distinto entre las imágenes es que aquellas que formaban parte del set de entrenamiento parecerían tener mayor definición en los gránulos del citoplasma, lo que podría ser la razón por la cual ninguno de los modelos logra una correcta clasificación de las imágenes. Este tema se desarrollará en mayor detalle en la sección de futuros trabajos.

Respecto de la imagen 70, surgen dos problemas: por un lado, el número de predicciones mal clasificadas y por el otro el número de Falsos Negativos.

Sobre aquellas predicciones que cometen un error en la clasificación, resulta interesante que son todos clasificados como basófilos. A continuación, se muestra la imagen 70 y algunas de las imágenes de entrenamiento de esta clase.

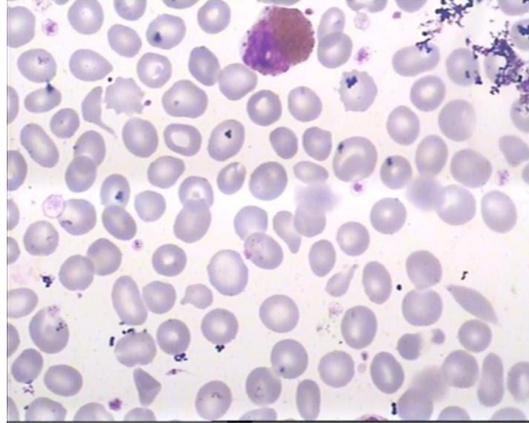


Ilustración 51: Imagen 70 del conjunto de datos original

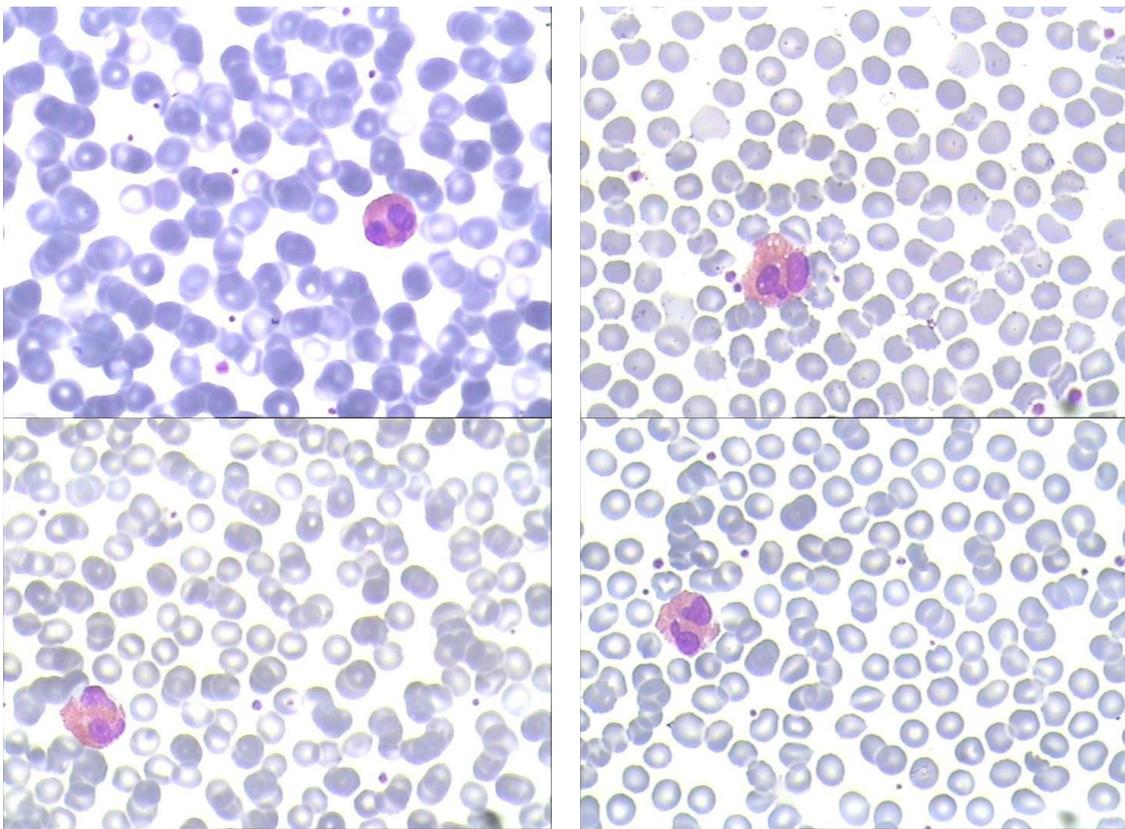
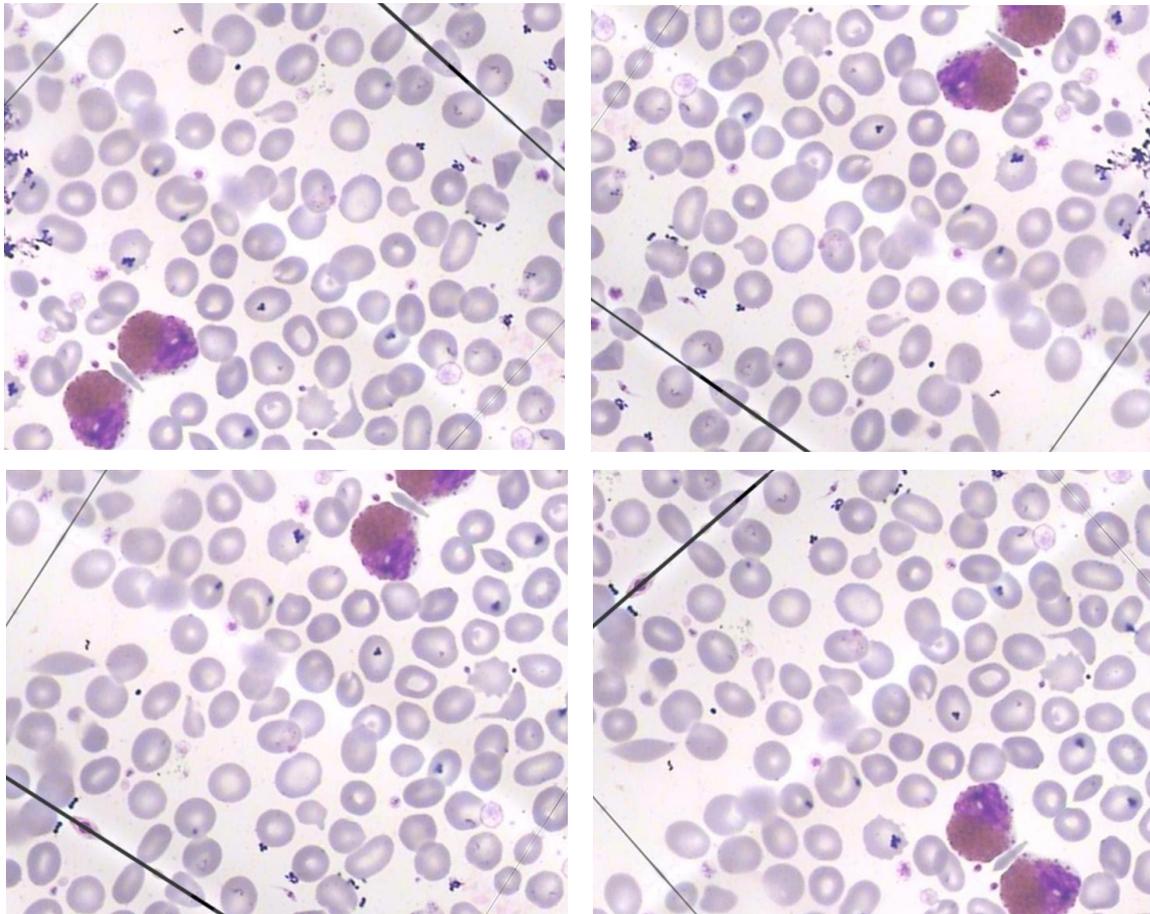


Ilustración 52: De izq a der. Imágenes 54, 56, 57, 59 del conjunto de imágenes originales

En las imágenes de entrenamiento se puede ver el citoplasma de color rosa por los gránulos característicos de los eosinófilos y el núcleo de color violeta. En la imagen 70 se ve el citoplasma de un color rosa más oscuro y parecería similar al rosáceo que se ve en toda la imagen, lo que podría llegar a ser una de las razones por la cual hay errores en la clasificación.

Por otro lado, respecto a la cantidad de Falsos Negativos parecería estar relacionado con un artefacto surgido de la forma de realizar data augmentation. Debido al reflejo que se realiza de las imágenes

durante el citado proceso, aparece un segundo glóbulo blanco en muchas de las imágenes aumentadas (a partir de la imagen 70). Al estar bastante cerca del borde el glóbulo blanco de la imagen original, los glóbulos blancos que surgen del reflejo también lo están, como se puede ver a continuación:



*Ilustración 53: Ejemplo de imágenes aumentadas con glóbulo blanco espurio producto de la reflexión de la imagen original*

Algunos de estos segundos glóbulos blancos terminan cortados por lo cual podría ser difícil para el modelo detectarlos. Por otro lado, es posible que estando tan cerca el segundo glóbulo blanco, el proceso de Non Maximum Suppression elimine dichas predicciones y por eso haya un número tan alto de Falsos Negativos.

Al analizar los resultados de ambos modelos, se buscó observar la distribución espacial de las predicciones para poder observar si existían patrones. Para eso se hizo el siguiente gráfico de dispersión que se realizó con los  $x_{medio}$  e  $y_{medio}$  de cada predicción, exceptuando en el caso de los Falsos Negativos en los que se utilizó los  $x_{medio}$  e  $y_{medio}$  de cada etiqueta. A continuación, se muestran los resultados para el set de validación.

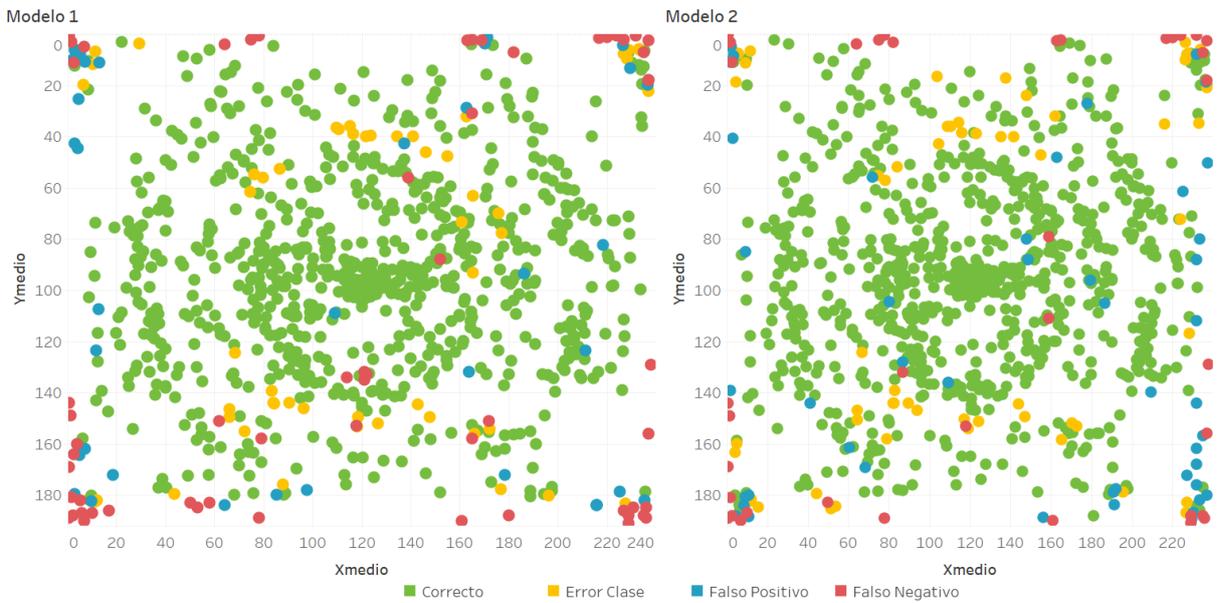


Ilustración 54: Gráfico con la ubicación de los centros de las predicciones (o de las etiquetas según corresponda) para el conjunto de datos de validación

Por un lado, se puede observar claramente la cantidad de predicciones correctas que tienen los modelos, pero al mismo tiempo se puede ver que parecerían encontrarse más concentradas en el centro de la imagen. Por otro lado, respecto de los Falsos Negativos, se puede observar una mayor concentración de estos cerca del borde de la imagen. Esto se puede deber a que esos glóbulos blancos se encuentren “cortados”, o sea que no se ve la célula completa. Este efecto se debe a la forma en que se decidió realizar el proceso de Data Augmentation. Un ejemplo de este efecto es la imagen n°9 obtenida de la imagen 70, que se mostró anteriormente. La distribución de los Falsos Negativos se puede observar con mayor detenimiento en la siguiente imagen donde también se distingue por la clase a la que pertenece el glóbulo blanco.

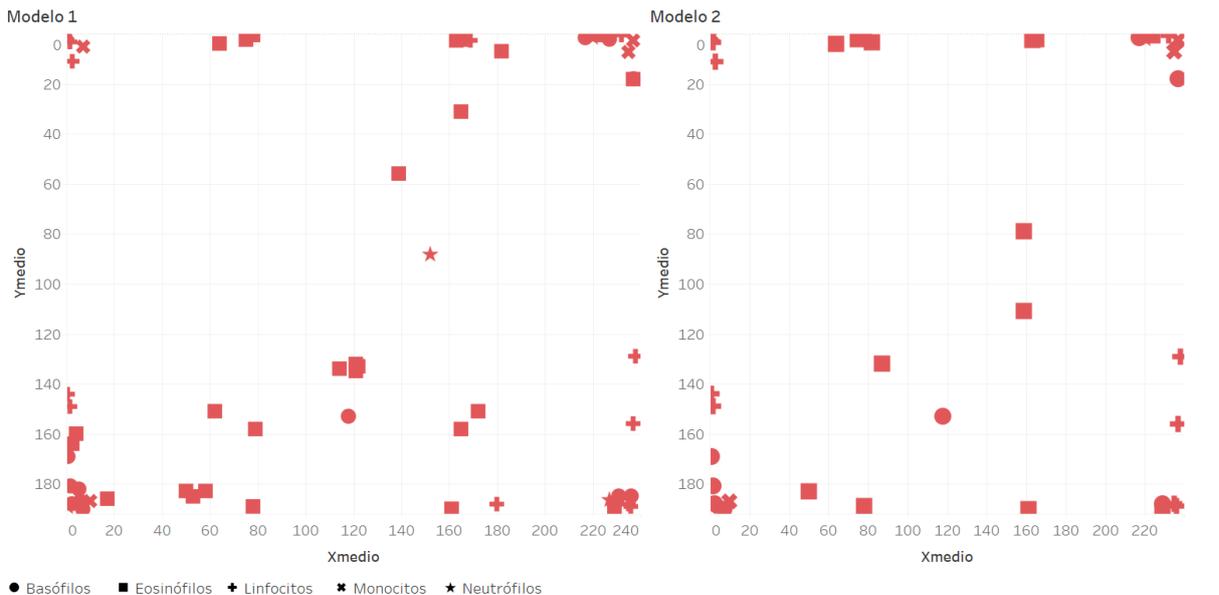


Ilustración 55: Gráfico con la ubicación específica de los falsos negativos por clase de glóbulo blanco para el conjunto de datos de validación

En este gráfico se puede observar lo antes mencionado junto al hecho que la mayoría de los glóbulos blancos que se encuentran en la parte central de ambos modelos son eosinófilos. Como se pudo observar en el gráfico de barras poseen la mayor cantidad de Falsos Negativos, en particular para el Modelo 1. De hecho, varios de esos Falsos Negativos pertenecen a la imagen 70, que ya se estudió en profundidad.

Por último, se decidió observar la distribución de los niveles de confianza de cada modelo que en los gráficos se denominarán score. Por un lado, se graficó la distribución del nivel de confianza general (el que se utiliza para definir si hay o no un objeto) que está compuesto por el nivel de confianza en la clase predicha y el nivel de confianza en sí hay un objeto o no en ese punto. A continuación, los resultados del nivel de confianza general para ambos modelos:

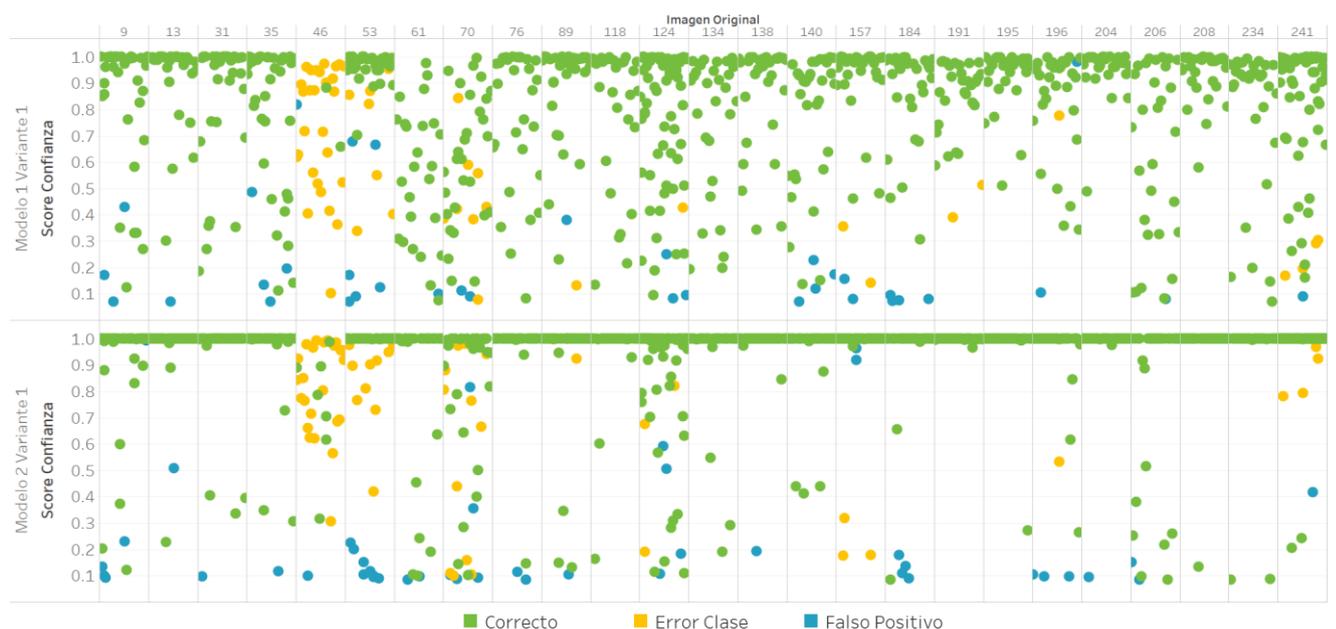


Ilustración 56: Gráfico con la distribución del nivel de confianza general por imagen original

Como ya se definió anteriormente, el nivel de confianza denominado general es la multiplicación entre el nivel de confianza en que haya un objeto y el nivel de confianza en la clase de dicho objeto. Lo interesante de este gráfico es la diferencia en la distribución de dichos valores. Es por esto que la curva del  $F_2$  score respecto del umbral se mantiene más constante para el Modelo 2 que para el Modelo 1. Al estar tan disperso el score para el Modelo 1, hace que a medida que aumenta el umbral se pierden mayor cantidad de detecciones. En el Modelo 2, se pierden detecciones a medida que se aumenta el umbral, pero el daño al desempeño del modelo es mucho menor.

Por otro lado, resulta interesante evaluar ambos componentes del score por separado para poder ver si el comportamiento es similar.

Primero se muestra un gráfico del score de clase:



Ilustración 57: Gráfico con la distribución del nivel de confianza de clase por imagen original

En el gráfico se puede observar que la distribución es bastante similar, excepto para ciertos casos como la imagen 70 o 191. Esto conforma un hallazgo interesante ya que parecería que la dispersión del nivel de confianza general tendría su origen en el nivel de confianza de objeto (Score Objeto), como se puede ver a continuación.

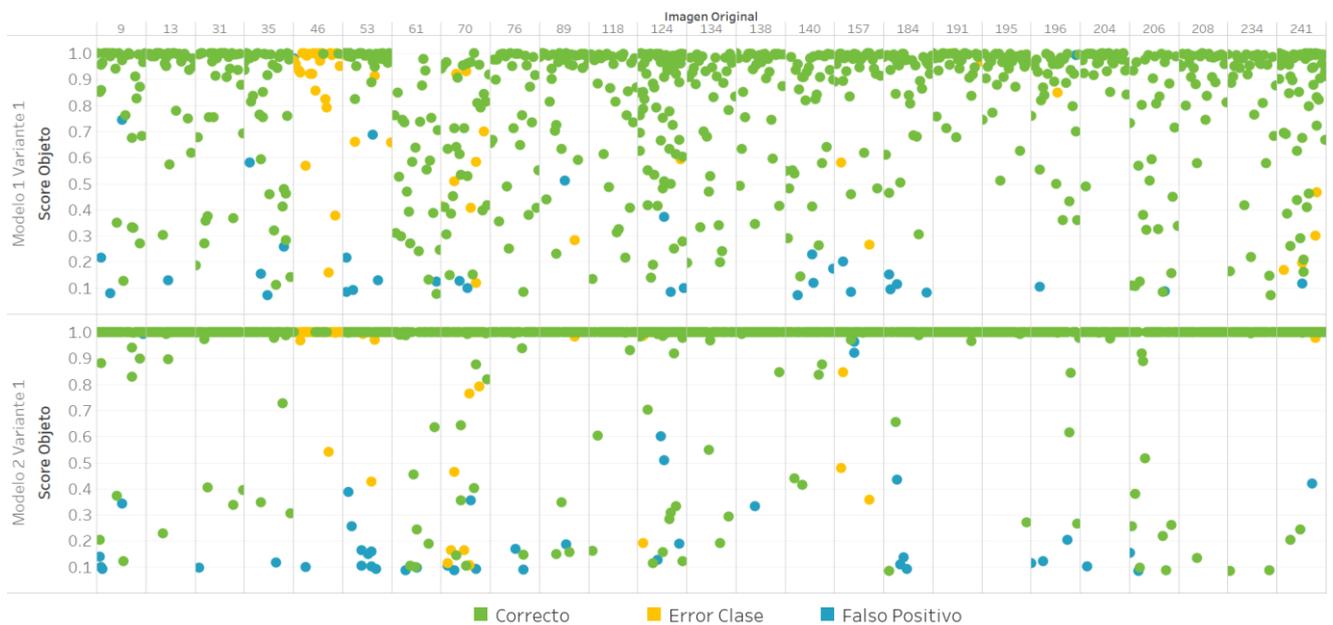


Ilustración 58: Gráfico con la distribución del nivel de confianza de objeto por imagen original

En este gráfico se puede ver una mayor dispersión del score, en ambos modelos, pero en particular para el Modelo 1 y parecería ser el originario de la dispersión del nivel de confianza general. Otro hecho interesante es que en ciertos casos en el que el nivel de confianza en que haya un objeto es realmente bajo, necesita un nivel de confianza en la clase realmente alto para que pueda superar el umbral. Sin embargo, esto podría también ser el origen de algunos de los Falsos Positivos que se encuentran.

A continuación, se pueden ver los tres gráficos juntos para poder observar el efecto antes mencionado y las diferencias en la distribución.



Ilustración 59: Gráfico con la distribución del nivel de confianza general, de clase y objeto por imagen original

En este gráfico se puede observar claramente la diferencia en la distribución del Score Objeto y el Score Clase en ambos modelos. También se pueden observar algunos casos en los que el Score Objeto es muy bajo, pero el Score Clase es tan alto que el nivel de confianza general termina superando el umbral. Esto se puede ver para casos correctos (imagen original 13 en el Modelo 2) como para falsos positivos (imagen 9 del mismo modelo).

Luego de haber analizado y definido los hiperparámetros de cada modelo con el conjunto de imágenes de validación, es necesario evaluar el rendimiento de los modelos para el conjunto de imágenes de evaluación. Para esto se analizarán los resultados de ambos  $F_2$  score para los conjuntos de validación y evaluación.

Modelo	Variante	Umbral	Set de Validación		Set de evaluación	
			$F_2$ Score Localización	$F_2$ Score Detección	$F_2$ Score Localización	$F_2$ Score Detección
Modelo 1	Variante 1	0.0692	0.9386	0.8776	0.9424	0.9156
Modelo 2	Variante 1	0.0832	0.9570	0.8902	0.9362	0.9314

Tabla 5: Comparativa entre los  $F_2$  score de cada modelo para los diferentes conjuntos de datos

Se puede observar que en ambos casos el valor es similar, aunque son mayores las métricas para el set de evaluación.

También se muestran las matrices de confusión para los umbrales seleccionados.

Etiqueta \ Predicción	Basófilo	Eosinófilo	Linfocito	Monocito	Neutrófilo	No hay objeto
Basófilo	129	15	0	1	0	19
Eosinófilo	0	153	0	2	1	10
Linfocito	2	2	172	0	0	15
Monocito	0	0	0	150	0	1
Neutrófilo	0	0	0	0	152	4
No hay objeto	10	1	14	11	6	

Tabla 6: Matriz de confusión del Modelo 1 Variante 1 utilizando el conjunto de evaluación para el umbral óptimo de 0.0692

Etiqueta \ Predicción	Basófilo	Eosinófilo	Linfocito	Monocito	Neutrófilo	No hay objeto
Basófilo	145	0	0	0	0	19
Eosinófilo	1	155	0	3	0	7
Linfocito	2	0	176	0	0	13
Monocito	0	0	0	145	0	6
Neutrófilo	0	1	0	2	149	4
No hay objeto	21	5	4	14	1	

Tabla 7: Matriz de confusión del Modelo 2 Variante 1 utilizando el conjunto de evaluación para el umbral óptimo de 0.0832

Al igual que con el set de validación, el Modelo 1 parecería presentar un mayor número de Falsos Negativos y el Modelo 2 un mayor número de Falsos Positivos. Uno de los casos que más llama la atención es la cantidad de errores de clase del Modelo 1 comparado del Modelo 2. Esto se explorará con los siguientes gráficos.

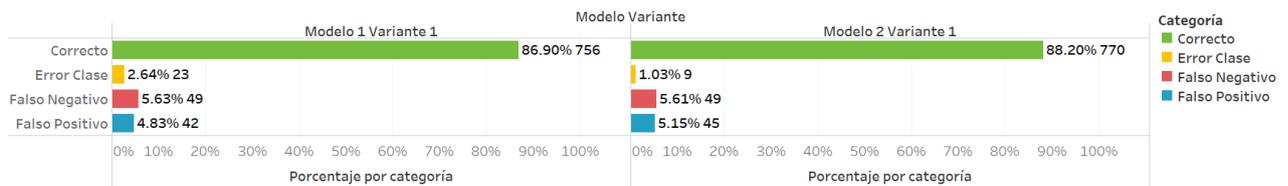


Ilustración 60: Resultados de la detección de ambos modelos para el total de las imágenes del conjunto de evaluación

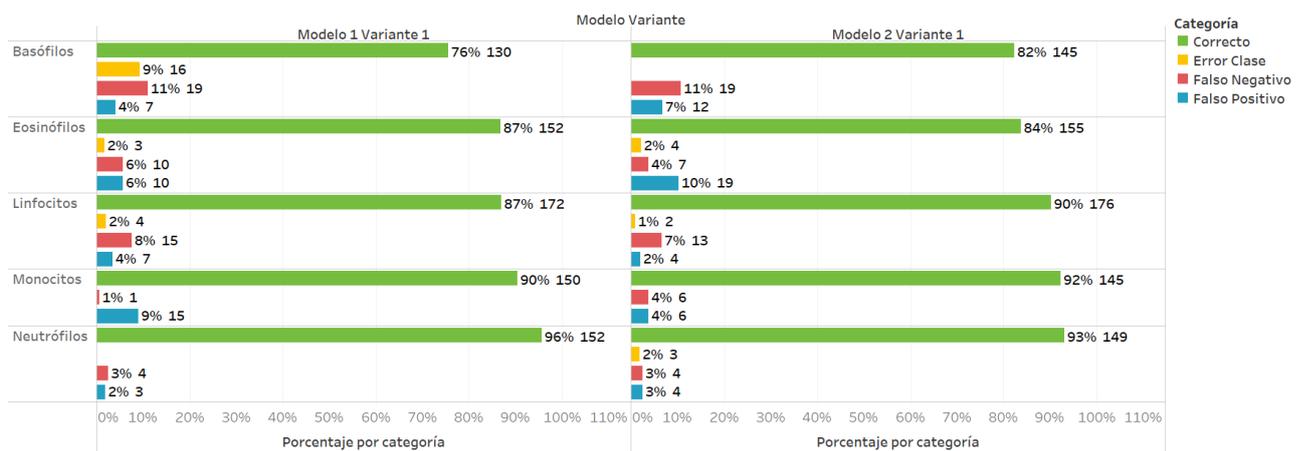


Ilustración 61: Resultados de la detección de ambos modelos por clase para las imágenes del conjunto de evaluación

En el primer gráfico se puede observar que finalmente el número de Falsos Negativos de ambos modelos es igual y la cantidad de Falsos Positivos es ligeramente mayor en el Modelo 2. La diferencia más notoria se puede ver en la cantidad de predicciones que fallaron en la clasificación en el Modelo 1.

En el segundo gráfico por clase, se puede observar que la principal diferencia en las predicciones con clase errónea es dentro de las imágenes con basófilos, situación que se abordará más adelante cuando se evalúe cada imagen en particular.

A diferencia de lo que sucedía con el set de validación, donde las diferencias entre ambos modelos se veían a través de todas las clases, en este caso, el rendimiento de ambos modelos resulta similar y varía respecto de la clase. Por ejemplo, el Modelo 2 presenta un mayor número de predicciones Correctas para los basófilos, eosinófilos y linfocitos, mientras que ocurre lo mismo para el Modelo 1 para los monocitos y neutrófilos.

A continuación, se muestra el mismo gráfico, pero por imagen original del set de evaluación:

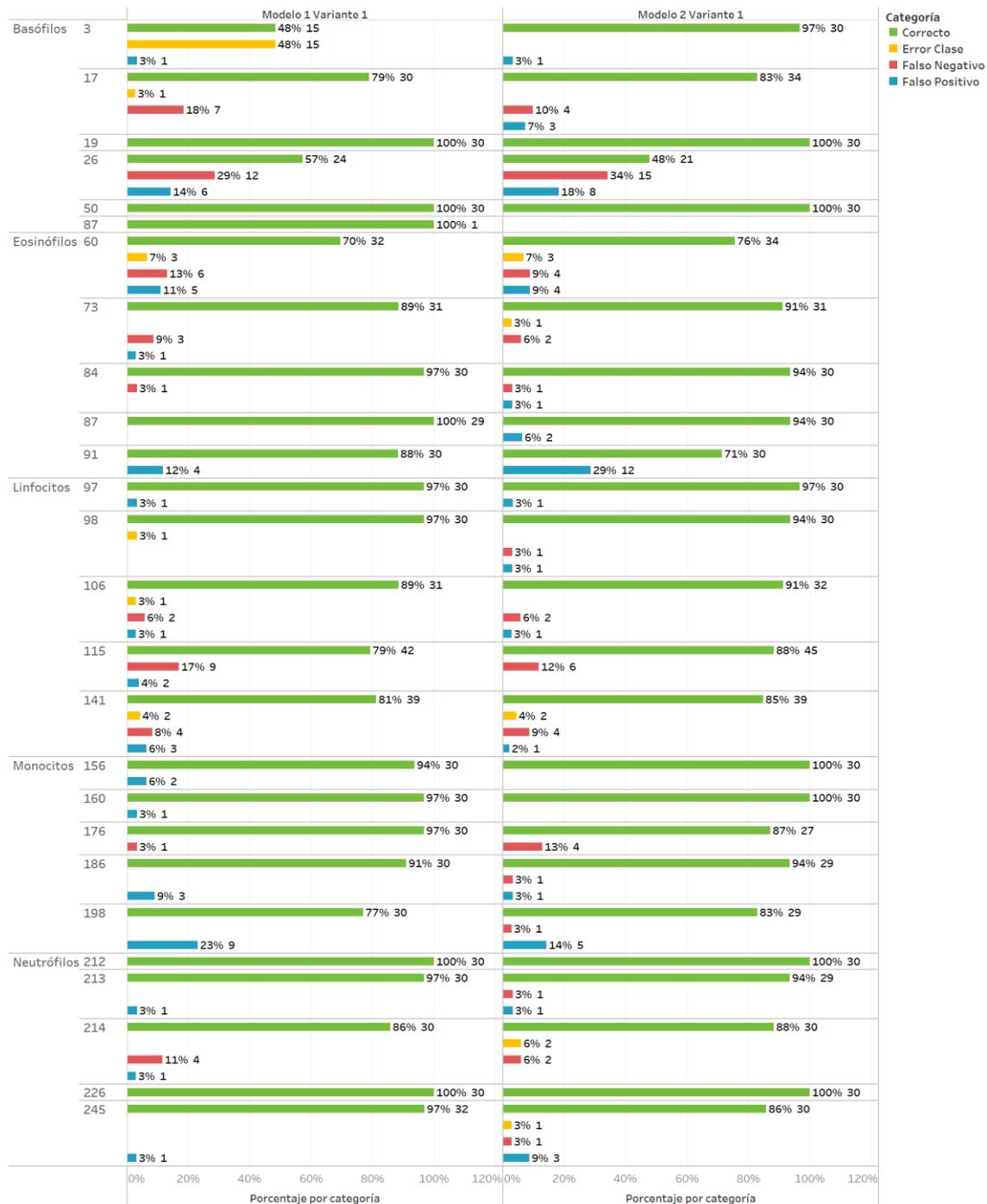


Ilustración 62: Resultados de la detección de ambos modelos agrupados por imagen original y clase del conjunto de evaluación

Uno de los hallazgos más relevantes de este gráfico, es encontrar que la mayoría de los errores de clasificación del Modelo 1 se concentran en una sola imagen que es la imagen 3. Lo llamativo, es que el Modelo 2 logra identificar correctamente todas las imágenes. En el resto de las imágenes se ve un comportamiento similar, aunque con ciertas diferencias menores en la cantidad de predicciones en cada categoría por imagen.

Al igual que sucedió con el set de validación, resulta importante entender la ubicación de los Falsos Negativos

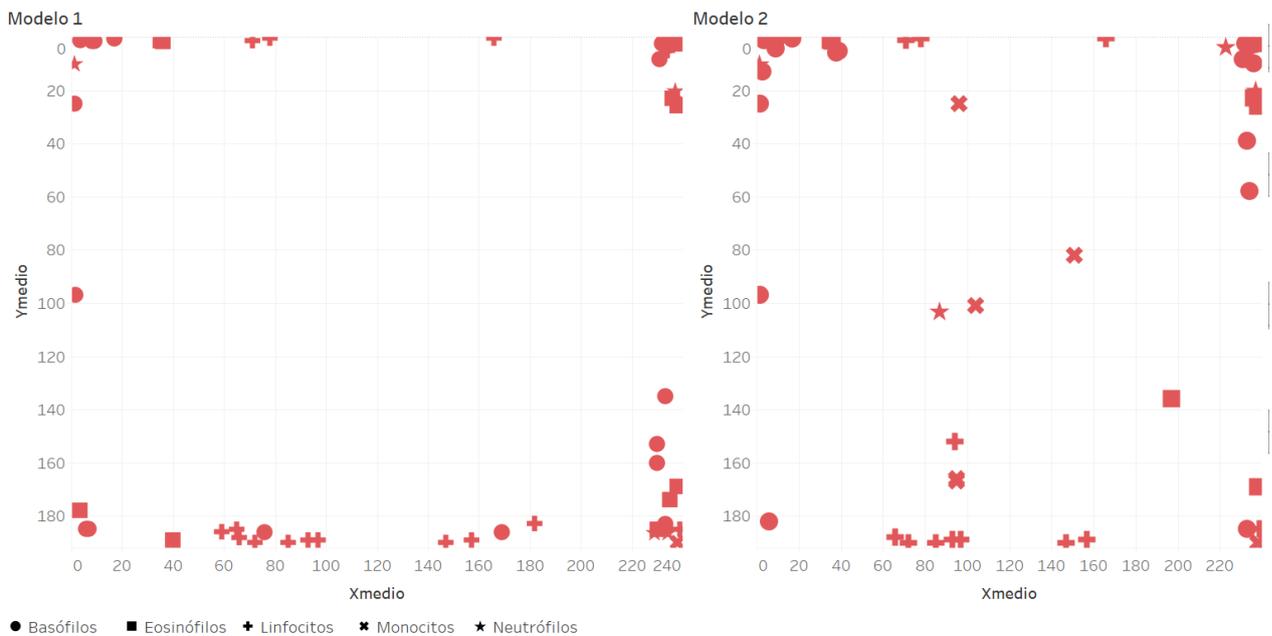


Ilustración 63: Gráfico con la ubicación específica de los falsos negativos por clase de glóbulo blanco para el conjunto de datos de evaluación

En este gráfico se puede observar que el Modelo 1 presenta solo Falsos Negativos en los bordes de la imagen, a diferencia de lo que sucedía con el conjunto de validación, lo que podría deberse a imágenes de glóbulos blancos cortados. Por otro lado, en el gráfico del Modelo 2 se observa un número de Falsos Negativos en el centro de la imagen, en su mayoría son monocitos, lo que también marca una diferencia con lo observado sobre el set de validación ya que en ese caso eran mayoría eosinófilos.

## Discusión

A partir de los resultados, se puede observar que ambos modelos lograron más de un 70% de detecciones correctas para cada una de las clases tanto para los conjuntos de validación como de evaluación. Además, superan el 80% de las detecciones correctamente localizadas dentro de cada clase. Estos resultados se ven replicados cuando se evalúa la totalidad de cada uno de los conjuntos de datos, siendo cerca del 85% y 90% de las detecciones correctas y correctamente localizadas, respectivamente.

Para poder evaluar estos resultados, es necesario ponerlos en contexto y esto es posible mediante la comparación de los mismos respecto de los resultados reflejados en otras publicaciones. Uno de los problemas para realizar este análisis es que muchos de los artículos existentes, encaran este problema como uno de clasificación y/o segmentación. En el primer caso, cuando se trata solamente el problema de clasificación, se busca asignar solo una etiqueta a la imagen sin importar la localización del objeto

y esta se encuentra limitada a que haya solo una clase de objeto dentro de la imagen [39]. En el segundo caso, la segmentación es una tarea en la que se busca a cada píxel asignar si pertenece a un objeto o no. Muchas veces la segmentación resulta en un paso intermedio para lograr la clasificación [40], [41], [42]. Esto resulta en que los artículos que se enfocan en esta temática resulten en una comparación errónea respecto de la presente tesis.

En particular, existe un artículo de Wang et. al. [43] que realiza un análisis de diferentes detectores unificados, entre los que se incluye YOLO v3, para realizar la tarea de detección de glóbulos blancos. En el mismo se usa una técnica denominada transfer learning que consiste en utilizar una red pre entrenada con algún dataset genérico, como puede ser ImageNet, y después utilizar un conjunto específico para terminar de ajustar los parámetros del modelo [18]. Este enfoque es distinto del planteado en la presente tesis donde los modelos son entrenados de cero y además no se utiliza la arquitectura particular de YOLO [29], sino que se usó otro número de capas y de filtros por capa.

El número de capas y de filtros de los modelos de la presente tesis son comparativamente menor que las de YOLO v3. Este último tiene 53 capas convolucionales y capas con hasta 512 filtros, los modelos aquí presentados tienen 7 capas y un máximo de 96 filtros por capa. Estas diferencias de diseño implican una menor complejidad de los modelos planteados, así como un número menor de parámetros. Estas diferencias en diseño fueron pensadas adrede basándose en la complejidad del problema a resolver que se refleja en la complejidad del conjunto de datos. Por ejemplo, mientras que LISC presenta entre 39 y 53 imágenes por clase, otros conjuntos, como ImageNet [44], presenta cerca de 1000 imágenes por clase. Además, hay una diferencia sustancial en la cantidad de clases que poseen ambos conjuntos de datos. En el caso de Wang et. al. se constaba de 11 categorías con un promedio de 1330 imágenes por categoría mientras que LISC presenta 5 clases. También en Wang et. al. [43], el conjunto de datos incluye clases de glóbulos blancos que se encuentran fuera del alcance de la presente tesis como pueden ser inmaduros y anormales (en sangre periférica).

La métrica utilizada para medir la eficacia en la detección en el artículo seleccionado es la media de la precisión promedio (mAP). Para su cálculo se busca la precisión promedio (AP por sus siglas en inglés) para cada clase a partir de obtener el área bajo la curva de precisión y sensibilidad. Luego, se obtiene la media de la precisión promedio de cada clase. Para poder realizar la comparación entre los modelos del artículo y los de la presente tesis, se decidió recalcular la media de la precisión promedio para incluir solo las clases que se encuentran en ambos modelos y así poder realizar una comparación más razonable. A continuación, se puede ver la comparación de las métricas.

Modelo	mAP	Basófilos	Eosinófilos	Linfocitos	Monocitos	Neutrófilos
<b>Modelo 1</b>	0.913	0.803	0.893	0.904	0.990	0.974
<b>Modelo 2</b>	0.944	0.909	0.941	0.938	0.971	0.961
<b>YOLOv3_320x320</b>	0.959	0.993	0.999	0.937	0.912	0.955
<b>YOLOv3_416x416</b>	0.955	0.987	0.999	0.947	0.894	0.947
<b>YOLOv3_416x416</b>	0.956	0.987	0.999	0.930	0.904	0.961
<b>YOLOv3-tiny_416x416*</b>	0.856	0.982	0.993	0.871	0.674	0.758

\*YOLOv3\_tiny refiere a una arquitectura de YOLO con menos capas

Tabla 8: Métricas de mAP y AP por clase para los modelos de la presente tesis y los expuestos en Wang et. al.

A partir de la comparación de las métricas, se puede observar que el desempeño de los modelos de la presente tesis es comparable con la obtenida en Wang et. al. Además, se observa que a nivel general el desempeño es mejor que YOLOv3\_tiny que es una versión de menor complejidad de YOLO al tener menos parámetros. Sin embargo, no se va a ahondar en mayor profundidad en la comparativa entre

los resultados obtenidos ya que el conjunto de datos de Wang et. [43] al era más complejo, como se explicó antes, y esto probablemente repercute en el desempeño ya que tiene una mayor cantidad de clases a identificar y es una de las posibles razones por las que se necesite modelos más complejos para resolver dicho problema.

Una de las técnicas utilizadas para poder combatir la falta de profundidad del conjunto de datos de LISC fue data augmentation. Es importante mencionar que esta técnica permite mostrar de forma diferente la información a la red neuronal, pero no es nueva información y eso está directamente relacionado con la complejidad del problema. Por otro lado, cuando esta herramienta se utilizó para este proyecto se siguieron las normas presentes en diversos libros [16], [18], [30],, pero un posible problema que no se había considerado al aplicar las transformaciones es el hecho de que varios de los glóbulos blancos queden solo una parte en la imagen. Como se observó en la sección de resultados, esta es una posible fuente de varios de los falsos negativos que se encontraron en los modelos. Por otro lado, para evitar perder completamente los glóbulos blancos en las imágenes, las traslaciones realizadas fueron pequeñas. Esto llevó a que el rango de movimiento de los leucocitos sea limitado. Esta decisión podría haber ido en contra del espíritu de esta técnica haciendo que la información presentada a los modelos no sea tan diversa como se esperaba.

Desde el diseño de los modelos también se incorporaron otros elementos distintos de YOLO como es el uso de cyclical Learning rate para el ajuste de la tasa de aprendizaje a lo largo del entrenamiento. Aunque esta es una práctica utilizada por otros modelos, en los artículos originales de YOLO se usa un calendario para configurar la tasa de aprendizaje. Uno de los hallazgos preliminares más interesantes en esta área fue el paso utilizado en los presentes modelos. En el artículo original [34] de esta técnica se menciona que el paso óptimo sería entre 2 y 10 epochs. Sin embargo, en los presentes modelos cuando se utilizó un paso de 4 epochs los modelos subajustaron, mientras que los modelos con paso de 0.08 epochs lograron un correcto ajuste (no sobreajustaban ni subajustaban). Obviamente, estos son hallazgos preliminares y serían necesarios mayores estudios sobre los hiperparámetros y otros posibles factores incidentes, pero es un primer paso. Además, este valor del parámetro podría estar relacionado con la redundancia de los datos antes mencionado.

Una de las ventajas de los modelos propuestos, es el hecho que solo está constituida por capas convolucionales y no presenta las capas densas del final de YOLO. Esto permite que se pueda potencialmente utilizar estos modelos para imágenes de otros tamaños modificando la capa de entrada y de salida con la limitación de que las imágenes deben tener dimensiones múltiplos de 8. Es importante rescatar que, aunque las imágenes sean más grandes es importante que el tamaño de los glóbulos blancos sea similar al conjunto de datos con el que fue entrenado ya que si no es probable que no los detecte correctamente.

Respecto de la detección, ya se abordaron diferentes métricas de desempeño, pero otro de los rasgos que se observó de los presentes modelos es que tienen una mayor efectividad en la localización de objetos respecto de la detección. En sí esto resulta esperable en cierta medida ya que la tarea de detección incluye a la de localización y suma la clasificación. Otro rasgo observable es que el desempeño para la tarea de localización es más consistente con los tres conjuntos de datos (entrenamiento, prueba y validación) que para la de detección. Esto quiere decir que cuando se evalúan las curvas del  $F_2$  score de localización entre los tres conjuntos, es menor la diferencia que cuando se hace lo mismo para el  $F_2$  score de detección como se muestra a continuación.

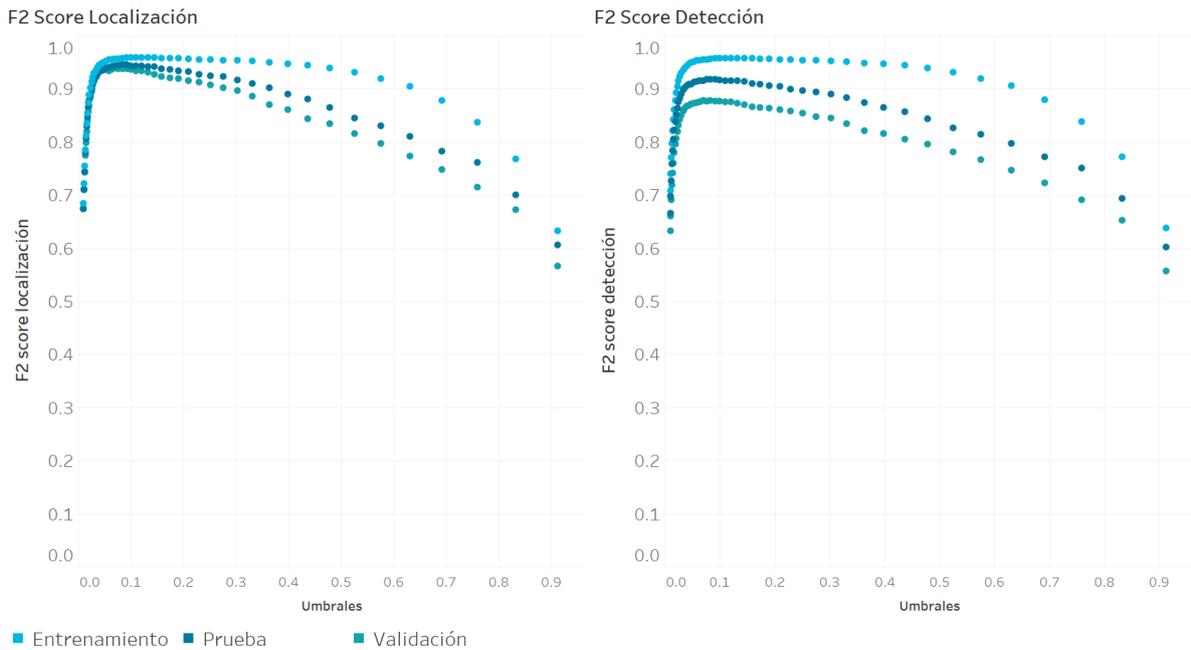


Ilustración 64: Gráfico comparativo de los F2 score para los tres conjuntos de datos para el Modelo 1

Este podría ser un aspecto para mejorar en futuros trabajos y aunque se van a mencionar en una sección subsiguiente posibles ideas, es importante recalcar que es probable que sea multifactorial la causa de estas diferencias observadas en la tarea de localización y detección.

Respecto al entrenamiento de estas redes, el tiempo de ejecución de esta fase solía estar entre 3 y 5 horas dependiendo del modelo. En Wang et. al, el tiempo de entrenamiento de YOLO y YOLO v3 tiny era alrededor de 60 hs y 9 hs, respectivamente. El problema de comparar entre ambos tiempos es que habría que empezar a comparar hardware y ambas implementaciones para poder encontrar si estas diferencias son reales o producto de los equipos utilizados. Sin embargo, el hecho que fueran modelos rápidos para entrenar y la disponibilidad en la nube de Colab facilitaba la experimentación con los mismos.

## Futuros trabajos y conclusiones

Dentro de la presente tesis, se mostraron dentro de los dos modelos diferentes variantes que obedecían a cambios en los hiperparámetros. Uno de los primeros puntos respecto de posibles trabajos a futuro es el hecho de una búsqueda de hiperparámetros optimizada de forma bayesiana [45]. Por un lado, para poder encontrar los parámetros óptimos para los diferentes modelos. Por otro lado, para poder evaluar si los comportamientos observados respecto de la comparación de los modelos responden realmente a algo intrínseco del uso de diferentes componentes como, por ejemplo, la Leaky ReLU como función de activación en los Modelo 2 respecto del Modelo 1.

Dentro de la búsqueda de hiperparámetros deberían incluirse las constantes que se utilizan en la función de costo que modulan el peso que se le da a los errores de localización respecto de los errores de clasificación. Es importante resaltar que, aunque es clave la tarea de clasificación dentro de la detección de objetos, en este caso existe un cuidado especial en la tarea de localización y no perder glóbulos blancos por la naturaleza semi automatizada de la tarea de conteo de glóbulos blancos hoy en día.

Otro de los hiperparámetros que fue constante en la presente tesis, es el umbral del IOU utilizado en Non Maximum Suppression. Al utilizar diferentes umbrales podría mejorar la tasa de Falsos Negativos, en especial en casos donde se encontrasen muy cercanos los glóbulos blancos y existiera superposición de las cajas delimitadoras. Por otro lado, se podrían probar otros métodos de Non Maximum Suppression como Soft Non Maximum Suppression [35] que quizás podrían solucionar este problema.

Además, se podría evaluar si el uso del nivel de confianza general y un único umbral sobre ese valor es lo óptimo para este problema particular. Podría ser que sea más beneficioso el utilizar dos umbrales, uno para cada nivel de confianza (de objeto y de clase).

También podría ser útil probar diferentes tamaños de grilla. Por ejemplo, al utilizar celdas de mayor tamaño, podría ser más fácil la tarea de localización del centro al mismo tiempo que puede ser que existan un menor número de predicciones alrededor de la predicción “correcta” (aquella celda que predice que hay un glóbulo blanco cuando su centro se encuentra ahí).

Asimismo, dentro de las métricas también existen posibilidades que permitirían entender mejor dónde está fallando el modelo para comprender su funcionamiento y oportunidades de mejora. Por un lado, sería interesante lograr “rescatar” las predicciones en caso de Falsos Negativos. Esto significaría ver las predicciones de la celda que debería ser responsable de esa predicción y poder también evaluar los valores. A partir de eso, quizás, se podría encontrar si esa predicción no pasó a ser una detección por los valores de los niveles de confianza o por el proceso de Non Maximum Suppression. Por otro lado, sería interesante emparejar la celda del centro de la etiqueta con la de la predicción y así poder evaluar si la predicción la está realizando la celda esperada respecto del funcionamiento de YOLO.

Desde el lado de las imágenes, se podrían entrenar los mismos modelos, pero con las imágenes de tamaño original para poder entender cómo impacta en el desempeño la diferencia de resolución. En Wang et. al. [43] se observa diferencias en el desempeño de los diferentes modelos dependiendo del tamaño de las imágenes, pero no es necesariamente extrapolable. Esto es importante para poder establecer también la relevancia de la resolución para este problema. También serviría para entender la resolución mínima necesaria para el correcto entrenamiento del modelo.

Relacionado con las imágenes utilizadas, se encuentran algunos de los problemas que se mencionaron en la sección anterior. Otra posible área de prueba sería aprovechar la libertad que proveen las imágenes de microscopía donde no existe arriba, abajo, izquierda o derecha, para probar técnicas de Data Augmentation más complejas que permitan evitar los problemas ya mencionados. Por ejemplo, que como parte del proceso se verifique la ubicación final de los glóbulos blancos y que no queden partidos.

Además, se podrían evaluar otras transformaciones que se le podrían realizar a las imágenes. Por ejemplo, teniendo en cuenta los errores de clasificación de la imagen 46, sería interesante indagar si existen otras transformaciones que sean compatibles con imágenes reales de microscopía. Por ejemplo, cambios en el brillo o el contraste, entre otros. Sin embargo, es importante mencionar que antes de realizar dichas transformaciones, se debería contar con el consejo de un técnico o bioquímico especialista en estas imágenes para entender qué artefactos son verosímiles y cuáles no. También es necesario la consulta con estos profesionales en caso de querer probar los modelos con imágenes distintas que no se encuentren etiquetadas.

Por último, es necesario volver a mencionar que estos modelos se entrenaron con un conjunto de imágenes de glóbulos blancos de morfología normal. Esto quiere decir que no se incluyen imágenes de microscopía de pacientes que presentan una morfología anómala a partir de alguna patología como, por ejemplo, leucemia donde pueden existir otras clases de glóbulos blancos donde la forma de estos se ve modificada. Es importante tener esto en cuenta pensando en un posible futuro sistema de detección de glóbulos blancos. En dicho sistema resulta un problema fundamental detectar aquellos casos de glóbulos blancos de morfología anormal.

Además, las imágenes provienen de un solo microscopio y una sola tinción. Sería interesante poder entrenar los modelos con una mayor variabilidad en las imágenes ya que eso podría aportar claridad sobre la robustez de estos. Así se podría abrir la puerta a probar, por ejemplo, si es posible con un número reducido de muestras reentrenar los modelos para que funcionen con otro microscopio y/o tinción en un futuro.

Como cierre, esta tesis se plantea como un punto de partida en la que se establecía el desafío de la detección de glóbulos blancos en las imágenes de microscopía. Como se puede concluir de los futuros trabajos, es un campo muy amplio con una gran cantidad de terreno por desarrollar, pero que es de vital importancia por el tiempo y recursos que podría ahorrar en un laboratorio.

## Bibliografía

- [1] K. S. Saladin, R. McFarland, C. A. Gan y H. N. Cushman, *Essentials of Anatomy & Physiology*, 2° ed., New York: McGraw-Hill Education, 2018, pp. 389 - 401.
- [2] E. N. Marieb y S. M. Keller, *Essentials of Human Anatomy & Physiology*, 12° ed., New York: Pearson Education, 2018, pp. 363 - 370.
- [3] G. J. Tortora, B. R. Funke y C. L. Case, *Introducción a la Microbiología*, 12° ed., Editorial Médica Panamericana, 2017.
- [4] H. Thieml, H. Diem y T. Haferlach, *Color Atlas of Hematology*, 2° revised ed., Stuttgart - New York: Thieme, 2004, p. 2.
- [5] J. H. Carr y B. F. Rodak, *Atlas de Hematología Clínica*, 5ta ed., Editorial Médica Panamericana, 2017, pp. 1-10.
- [6] B. J. Bain, I. Bates y M. A. Laffan, *Dacie and Lewis Practical Haematology*, 12° ed., Elsevier, 2017, pp. 25, 36-38.
- [7] J. P. Greer, *Wintrobe's clinical hematology*, 13th ed., J. P. Greer, D. A. Arber, B. Glaser, A. F. List, R. T. Means Jr., F. Paraskevas, G. M. Rodgers y J. Foerster, Edits., Philadelphia: Wolters Kluwer Lippincott Williams & Wilkins Health, 2014, pp. 2-10.
- [8] N. E. Fink, «Automatización en hematología,» *Hematología*, vol. 9, pp. 4-16, 2005.
- [9] A. Kratz, S.-h. Lee, G. Zini, J. A. Riedl, M. Hur y S. Machin, «Digital morphology analyzers in hematology: ICSH review and recommendations,» *International Journal of Laboratory Hematology*, vol. 41, nº 4, pp. 437-447, August 2019.
- [10] M. Leach, M. Drummond y A. Doig, *Practical Flow Cytometry in Haematology Diagnosis*, Wiley-Blackwell, 2013, pp. 3-19.
- [11] S. J. Kussick, *Hematopathology*, 3rd ed., E. D. Hsi, Ed., Elsevier, 2018, pp. 686-711.
- [12] M. Brown y C. Wittwer, «Flow Cytometry: Principles and Clinical Applications in Hematology,» *Clinical Chemistry*, vol. 46, nº 8, pp. 1221-1229, 1 August 2000.
- [13] E. Cornet, J. Perol y X. Troussard, «Performance evaluation and relevance of the CellaVision DM96 system in routine analysis and in patients with malignant hematological diseases,» *International Journal of Laboratory Hematology*, vol. 30, nº 6, pp. 536-542, December 2008.
- [14] «Introducing CellaVision DC-1,» 23 Jan 2019. [Online]. Disponible: <https://www.cellavision.com/images/pdf/2020/MM-139-01.pdf>. [Accedido 16 Agosto 2020].
- [15] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu y M. Pietikäinen, «Deep Learning for Generic Object Detection: A Survey,» *International Journal of Computer Vision*, vol. 128, nº 2, pp. 261-318, February 2020.

- [16] S. Khan, H. Rahmani, S. A. A. Shah y M. Bennamoun, *A Guide to Convolutional Neural Networks for Computer Vision*, Morgan & Claypool, 2018, pp. 1,53-56,75-76, 80-85.
- [17] N. Buduma y N. Lacascio, *Fundamentals of Deep Learning*, 1st ed., M. Loukides y S. Cutt, Edits., Sebastopol: O'Reilly, 2017, pp. 15, 69-75, 85, 88, 90-92.
- [18] C. C. Aggarwal, *Neural Networks and Deep Learning*, New York: Springer, 2018, p. 316.
- [19] G. Dougherty, *Digital Image Processing for Medical Applications*, 1st ed., New York: Cambridge University Press, 2009, pp. 170, 273-283, 311, 340-342.
- [20] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. Velasco-Hernandez, L. Krpalkova, D. Riordan y J. Walsh, «Deep Learning vs. Traditional Computer Vision,» de *2019 Computer Vision Conference*, 2020.
- [21] M. Everingham, L. Van-Gool, C. K. I. Williams, J. Winn y A. Zisserman, «The Pascal Visual Object Classes (VOC) Challenge,» *International Journal of Computer Vision*, vol. 88, nº 2, pp. 303-338, Jun 2010.
- [22] J. Patterson y A. Gibson, *Deep Learning*, 1st ed., M. Loukides y T. McGovern, Edits., Sebastopol: O'Reilly Media, 2017, pp. 4, 5, 30, 43-50, 71.
- [23] J. H. Mathews y K. D. Fink, *Métodos Numéricos con MATLAB*, I. Capella, Ed., Madrid: Prentice Hall, 2000, p. 447.
- [24] S. Ruder, *An overview of gradient descent optimization algorithms*, 2017, arXiv: 1609.04747.
- [25] S. Ren, K. He, R. Girshick y J. Sun, «Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,» de *Advances in neural information processing systems*, 2015.
- [26] K. He, G. Gkioxari, P. Dollár y R. Girshick, «Mask R-CNN,» de *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, 2017.
- [27] J. Redmon, S. Divvala, R. Girshick y A. Farhadi, «You Only Look Once: Unified, Real-Time Object Detection,» de *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 2016.
- [28] J. Redmon y A. Farhadi, «YOLO9000: Better, Faster, Stronger,» de *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, 2017.
- [29] J. Redmon y A. Farhadi, «YOLOv3: An Incremental Improvement,» 2018, arXiv: 1804.02767.
- [30] I. Goodfellow, Y. Bengio y A. Courville, *Deep Learning*, MIT Press, 2016, pp. 96-113.
- [31] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid y S. Savarese, «Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression,» de *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, 2019.
- [32] H. Jeon y S. Cho, «Brief Paper: Drivable Area Detection with Region-based CNN Models to Support Autonomous Driving,» *Journal of Multimedia Information System*, vol. 7, nº 1, pp. 41-44, 2020.

- [33] A. Tharwat, «Classification assessment methods,» *Applied Computing and Informatics*, 2018.
- [34] L. N. Smith, «Cyclical Learning Rates for Training Neural Networks,» de *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Santa Rosa, 2017.
- [35] R. Atienza, *Advanced Deep Learning with TensorFlow 2 and Keras*, J. Gonsalves, Ed., Birmingham: Packt Publishing, 2020, p. 408.
- [36] S. H. Rezatofghi y H. Soltanian-Zadeh, «Automatic recognition of five types of white blood cells in peripheral blood,» *Computerized Medical Imaging and Graphics*, vol. 35, nº 4, pp. 333-343, 2011.
- [37] Chollet, Francois, *Keras*, <https://keras.io>, 2015.
- [38] m. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, R. Jozefowicz, Y. Jia, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, M. Schuster, R. Monga, S. Moore, D. Murray, C. Olah, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu y X. Zheng, *TensorFlow: Large-scale machine learning on heterogeneous systems*, 2015.
- [39] W. Yu, J. Chang, C. Yang, L. Zhang, H. Shen, Y. Xia y J. Sha, «Automatic classification of leukocytes using deep neural network,» de *12th International Conference on ASIC (ASICON)*, Gulyang, 2017.
- [40] J. Zhao, M. Zhang, Z. Zhou, J. Chu y F. Cao, «Automatic detection and classification of leukocytes using convolutional neural networks,» *Medical & Biological Engineering & Computing*, vol. 55, pp. 1287-1301, 2017.
- [41] M. I. Razzak y S. Naz, «Microscopic Blood Smear Segmentation and Classification Using Deep Contour Aware CNN and Extreme Machine Learning,» de *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, 2017.
- [42] M. Z. Othman, T. S. Mohammed y A. B. Ali, «Neural Network Classification of White Blood Cell using Microscopic Images,» *International Journal of Advanced Computer Science and Applications*, vol. 8, nº 5, pp. 99-104, 2017.
- [43] Q. Wang, S. Bi, M. Sun, Y. Wnag, D. Wang y S. Yang, «Deep learning approach to peripheral leukocyte recognition,» *PLOS ONE*, vol. 14, nº 6, 25 June 2019.
- [44] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li y L. Fei-Fei, «Imagenet: A large-scale hierarchical image database,» de *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [45] A. H. Victoria y G. Maragatham, «Automatic tuning hyperparameters using Bayesian optimization,» *Evolving Systems*, 2020.
- [46] S. Ren, K. He, R. Girshick y J. Sun, «Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137-1149, 2015.

## Anexo A

El siguiente anexo busca incluir algunas de las imágenes abordadas en los resultados en los que se utilizó el Modelo 1 Variante 1 a modo de ejemplo. A continuación, se ofrecen los mismos gráficos, pero para los otros modelos.

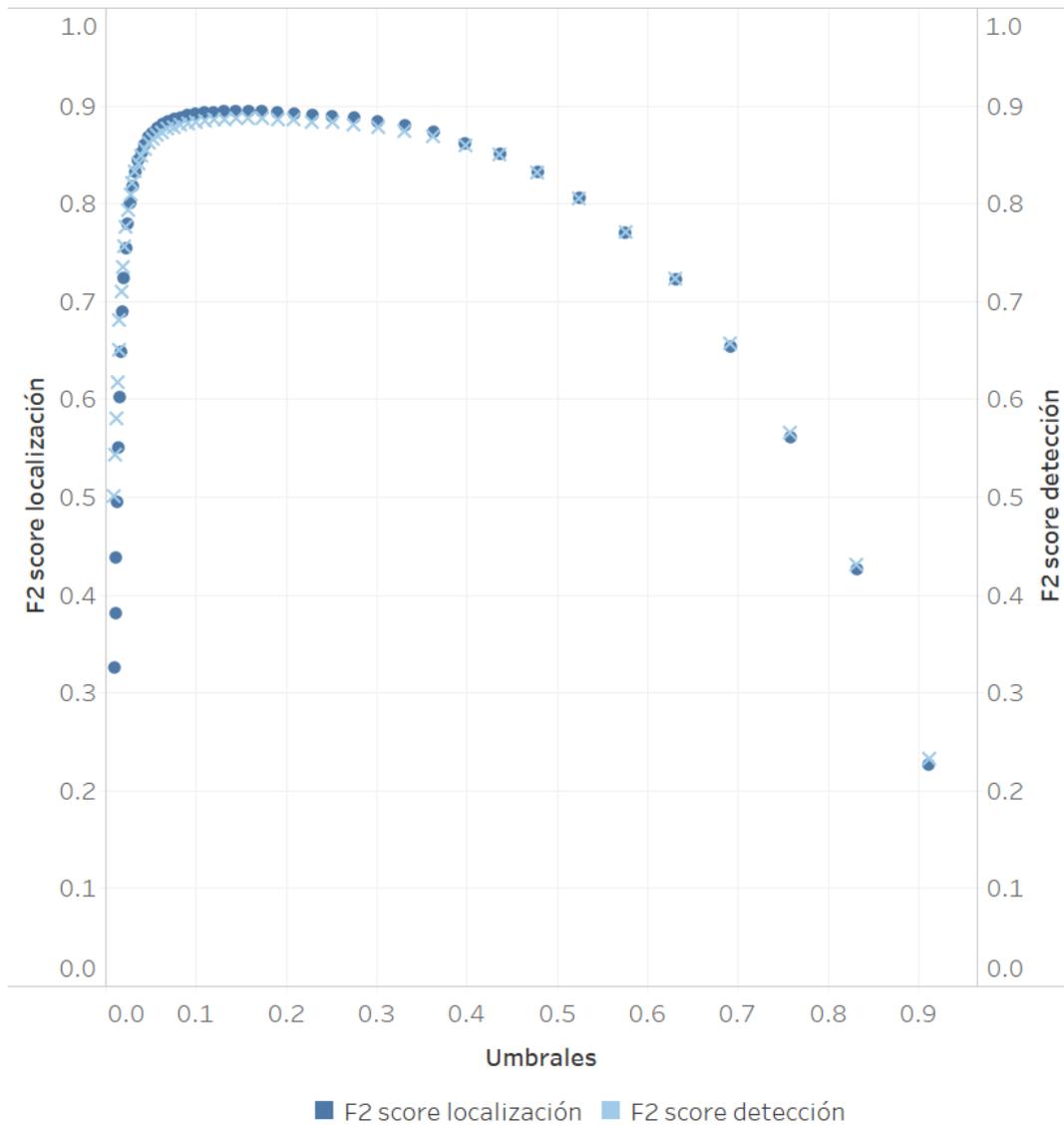


Ilustración 65:  $F_2$  score de localización y de detección en función del umbral para el Modelo 1 Variante 2 para el conjunto de entrenamiento

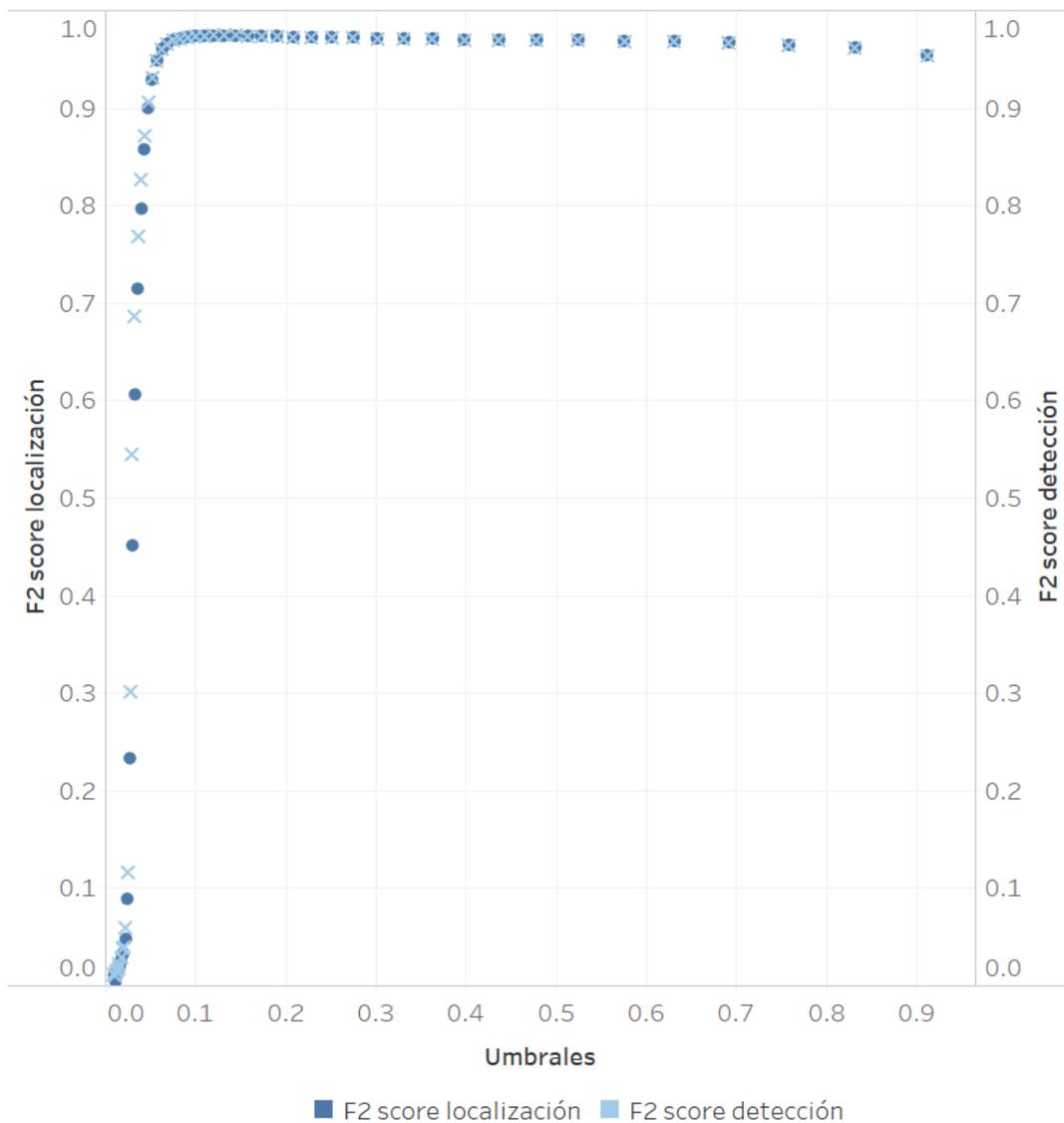


Ilustración 66:  $F_2$  score de localización y de detección en función del umbral para el Modelo 2 Variante 1 para el conjunto de entrenamiento

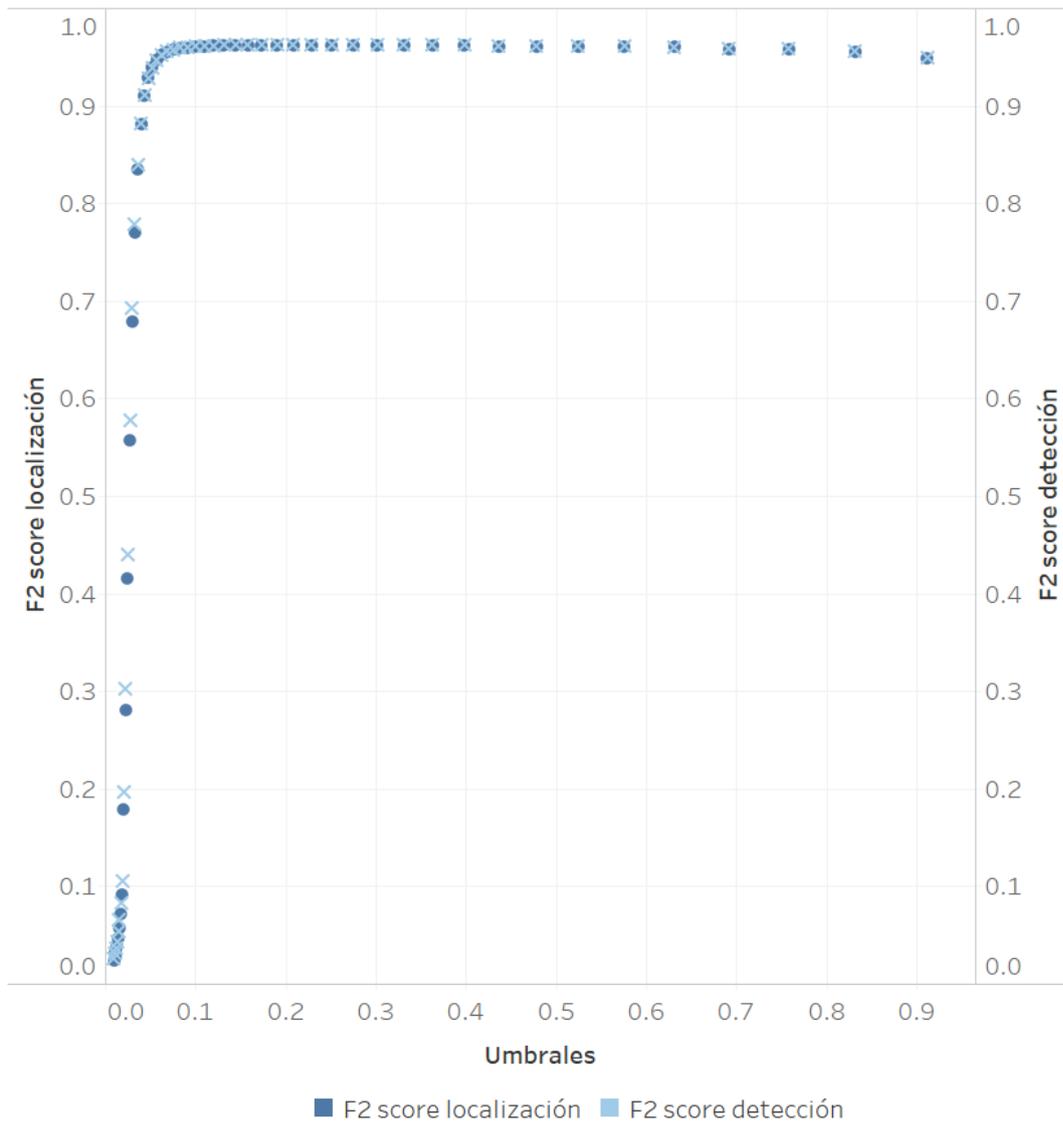


Ilustración 67:  $F_2$  score de localización y de detección en función del umbral para el Modelo 2 Variante 2 para el conjunto de entrenamiento

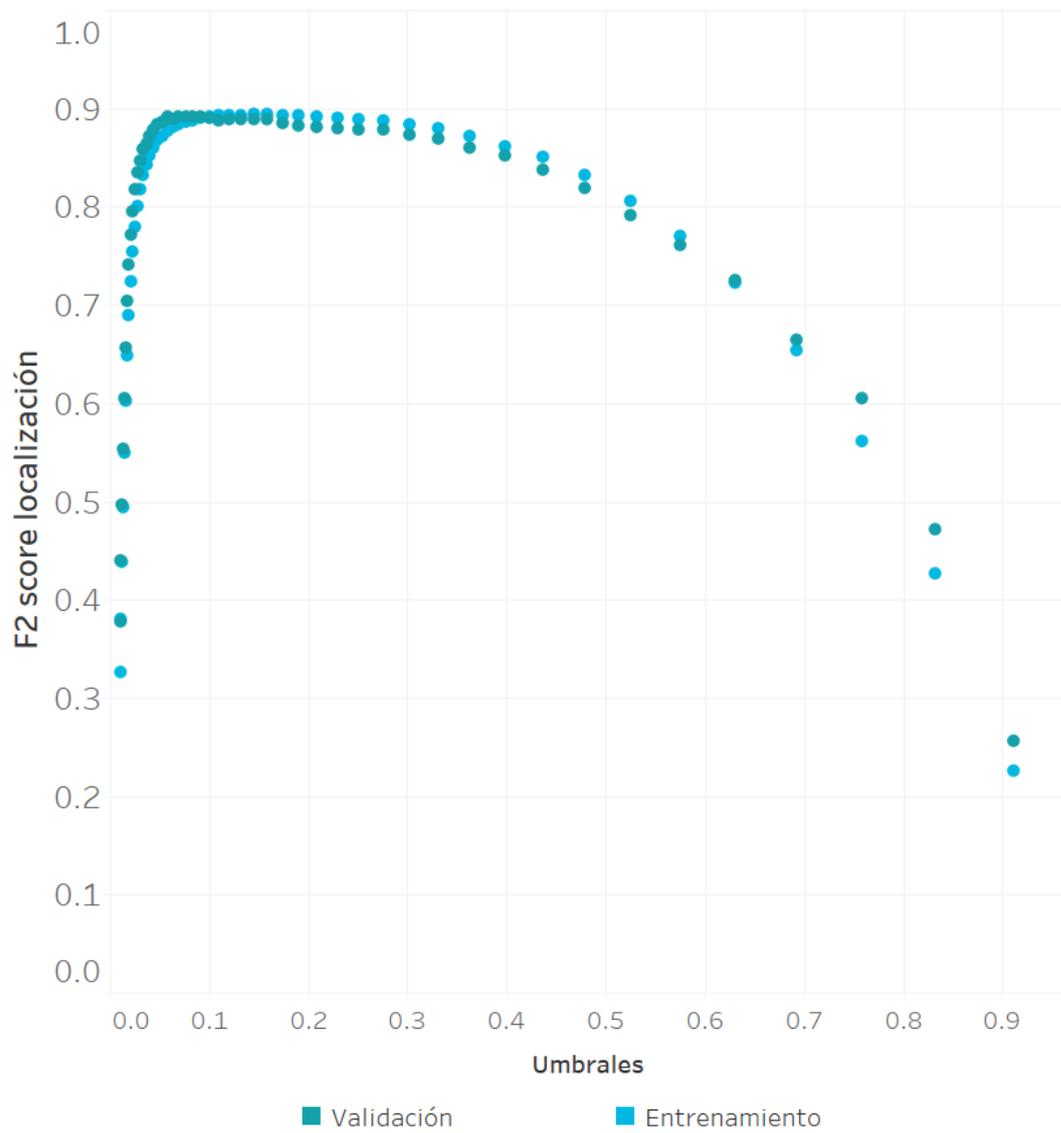


Ilustración 68: Comparación de las curvas del F<sub>2</sub> score de localización en función del umbral entre el conjunto de entrenamiento y el de validación para el Modelo 1 Variante 2

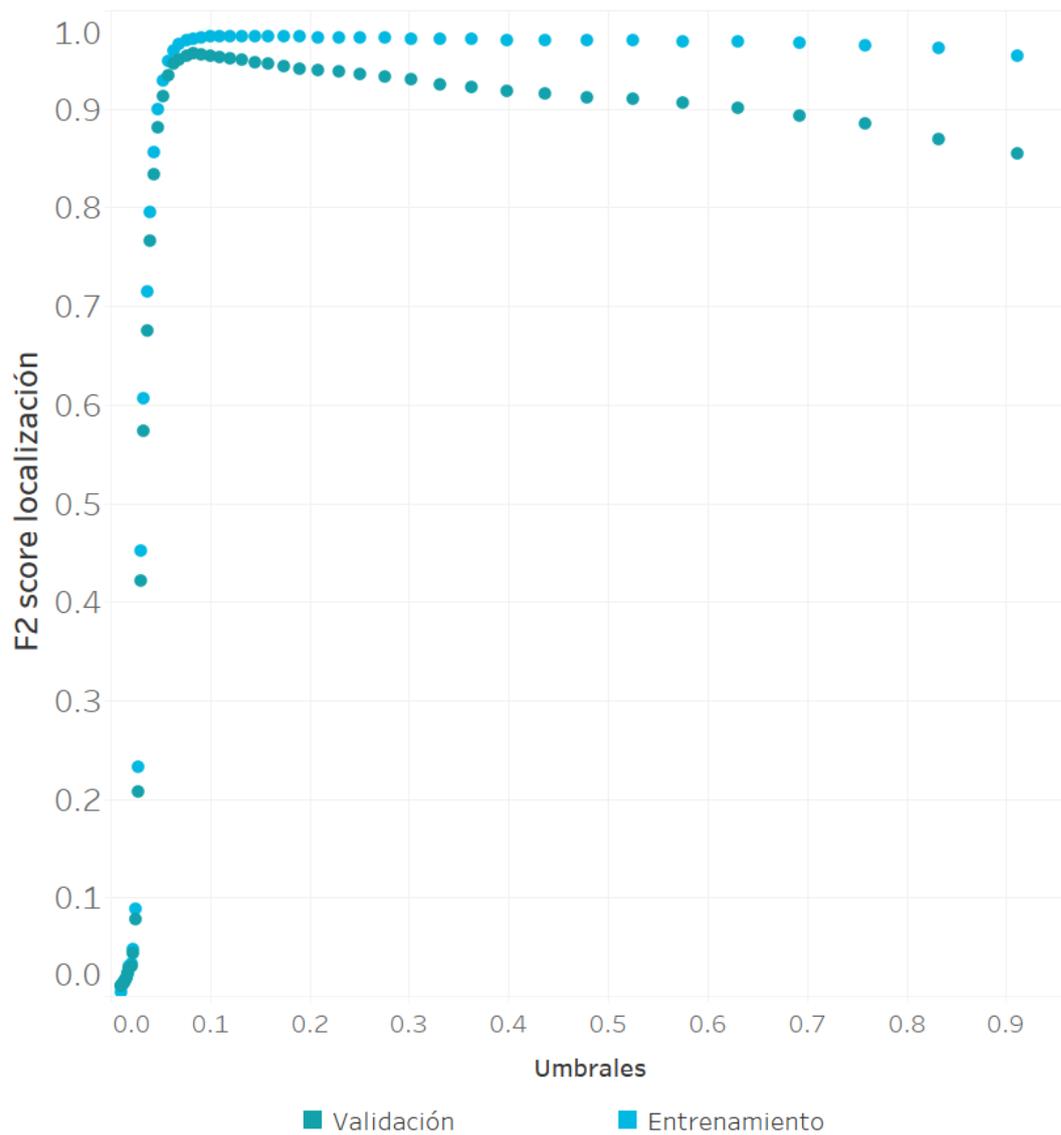


Ilustración 69: Comparación de las curvas del F<sub>2</sub> score de localización en función del umbral entre el conjunto de entrenamiento y el de validación para el Modelo 2 Variante 1

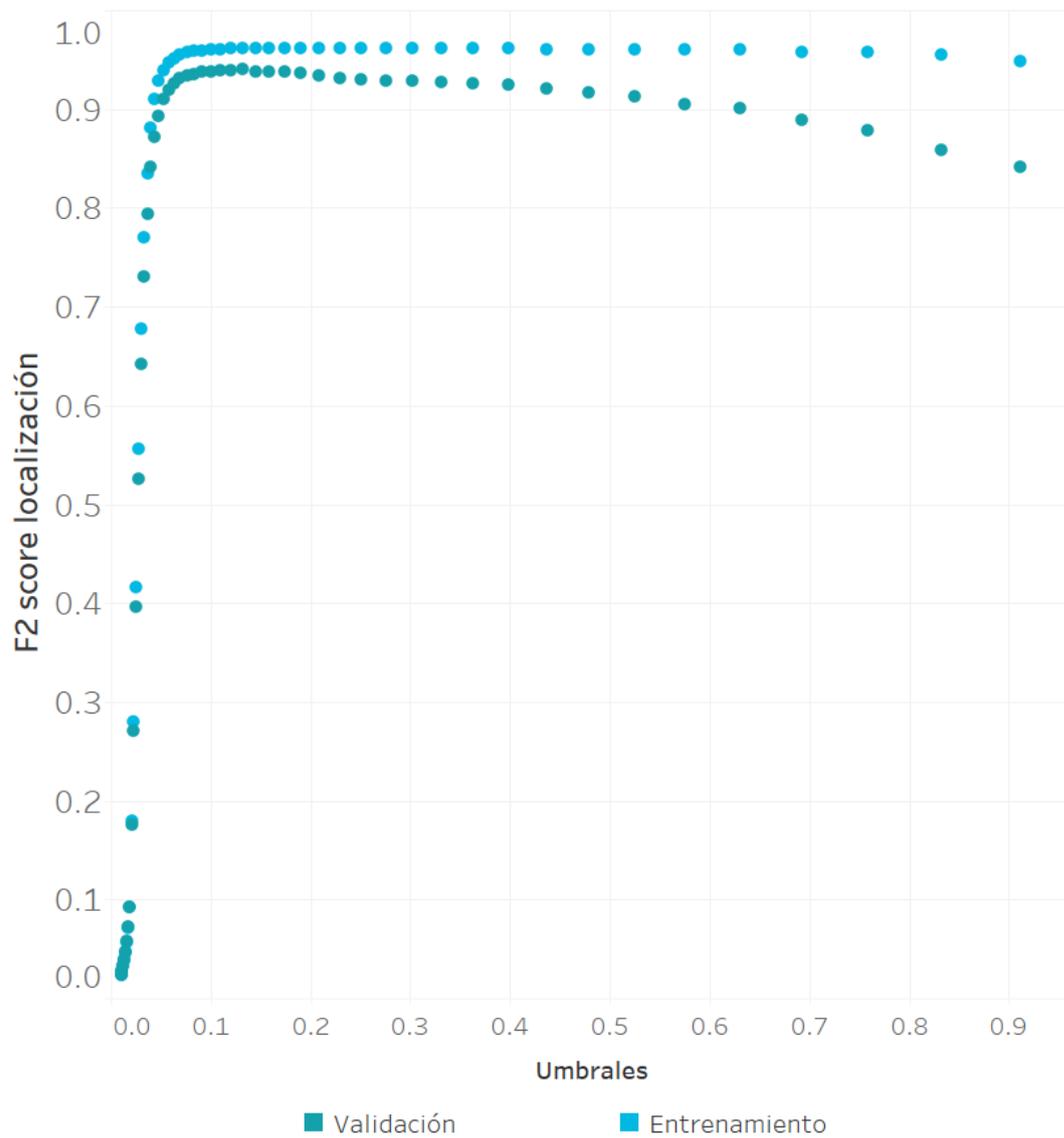


Ilustración 70: Comparación de las curvas del  $F_2$  score de localización en función del umbral entre el conjunto de entrenamiento y el de validación para el Modelo 2 Variante 2

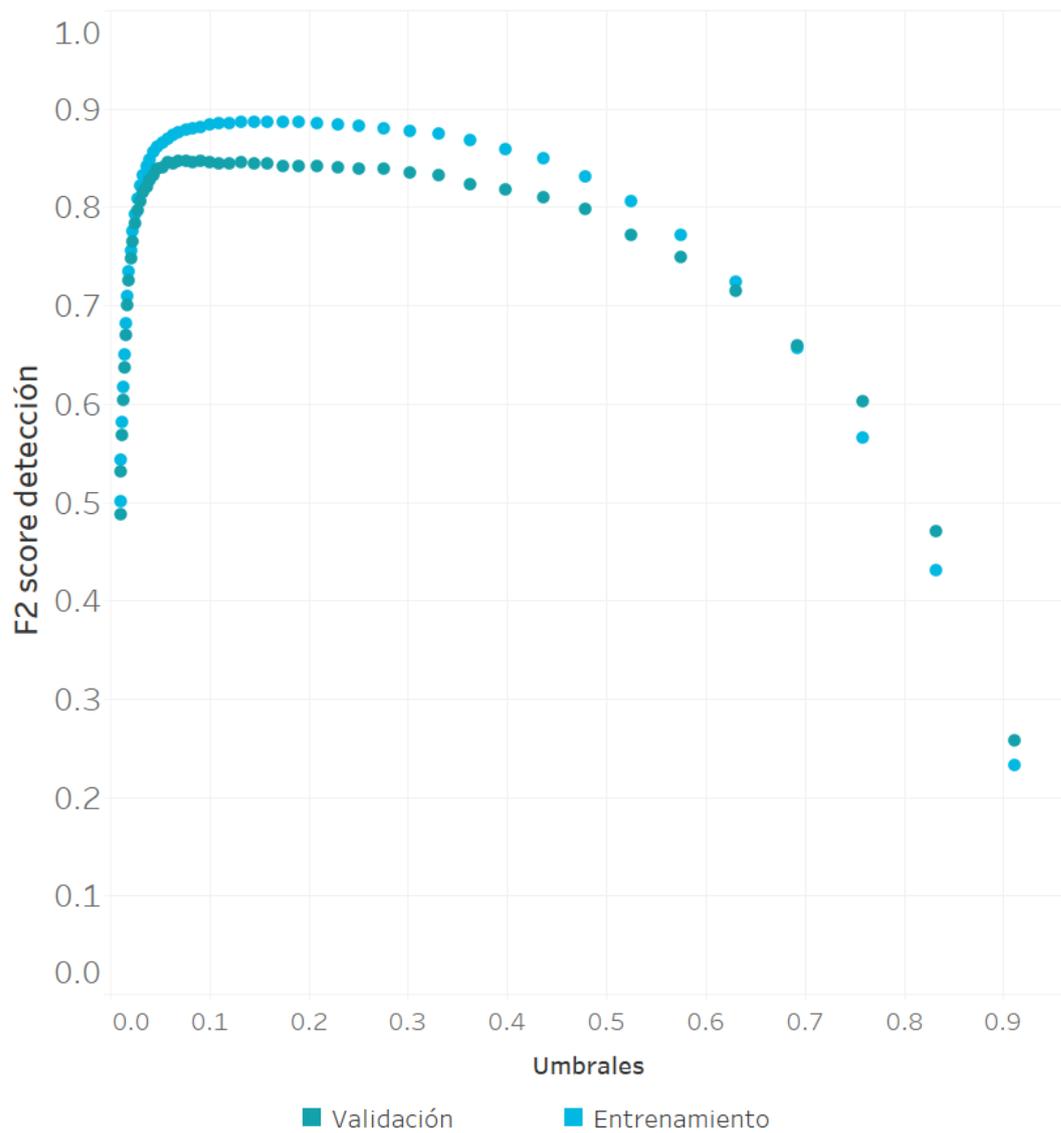


Ilustración 71: Comparación de las curvas del  $F_2$  score de detección en función del umbral entre el conjunto de entrenamiento y el de validación para el Modelo 1 Variante 2

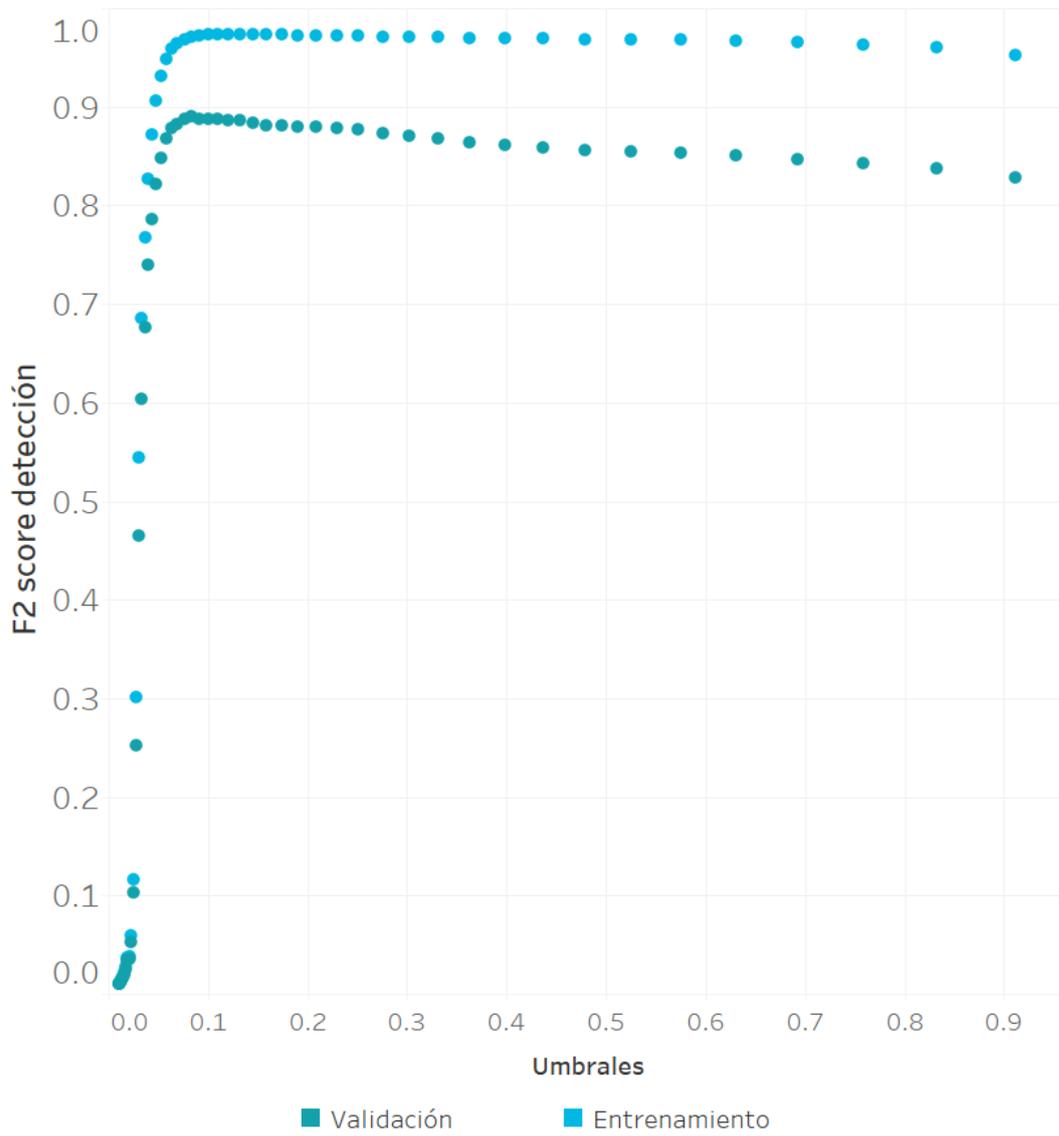


Ilustración 72: Comparación de las curvas del  $F_2$  score de detección en función del umbral entre el conjunto de entrenamiento y el de validación para el Modelo 2 Variante 1

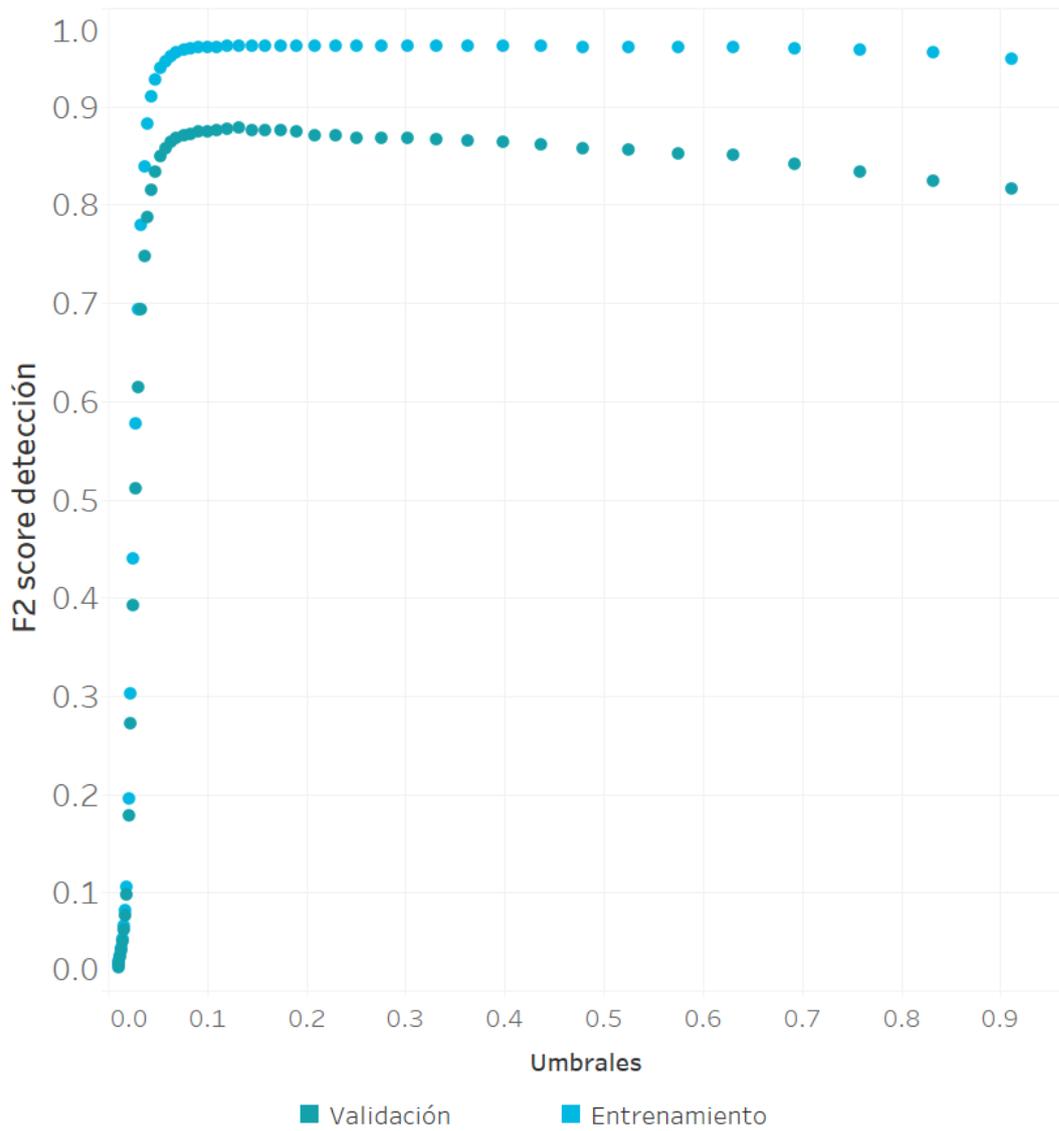


Ilustración 73: Comparación de las curvas del F<sub>2</sub> score de detección en función del umbral entre el conjunto de entrenamiento y el de validación para el Modelo 2 Variante 2

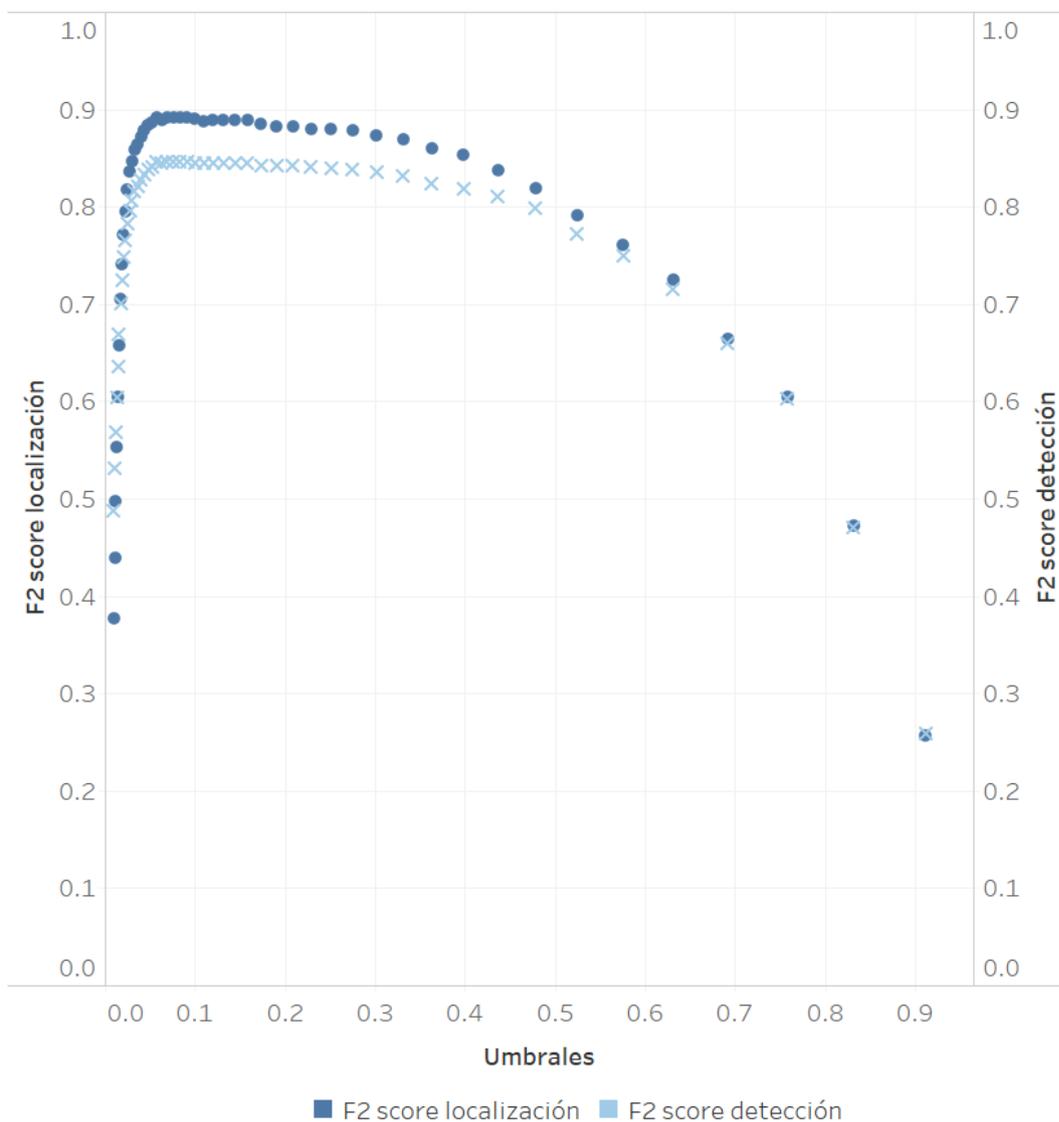


Ilustración 74: F<sub>2</sub> score de localización y de detección en función del umbral para el Modelo 1 Variante 2 para el conjunto de validación

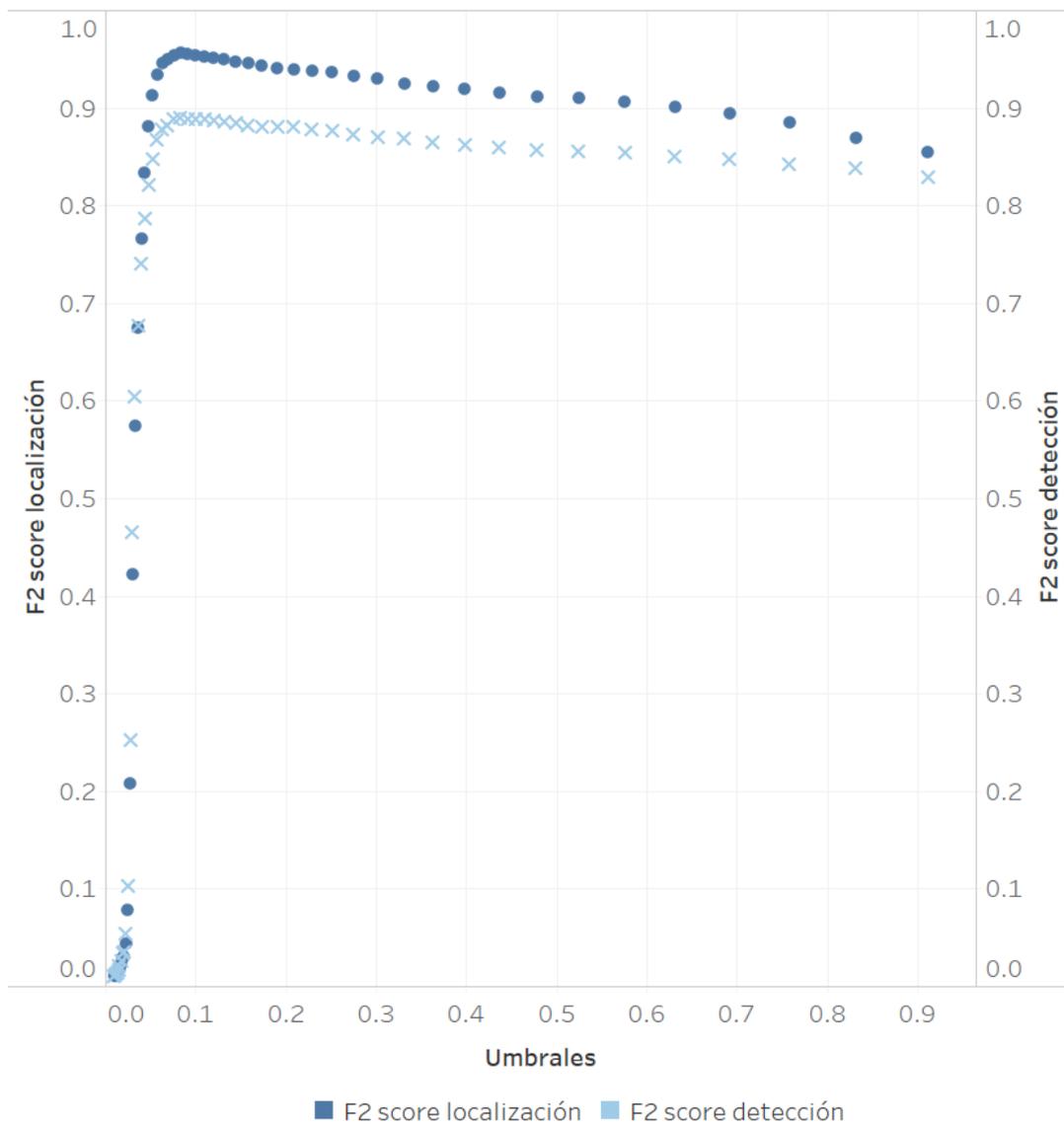


Ilustración 75: F<sub>2</sub> score de localización y de detección en función del umbral para el Modelo 2 Variante 1 para el conjunto de validación

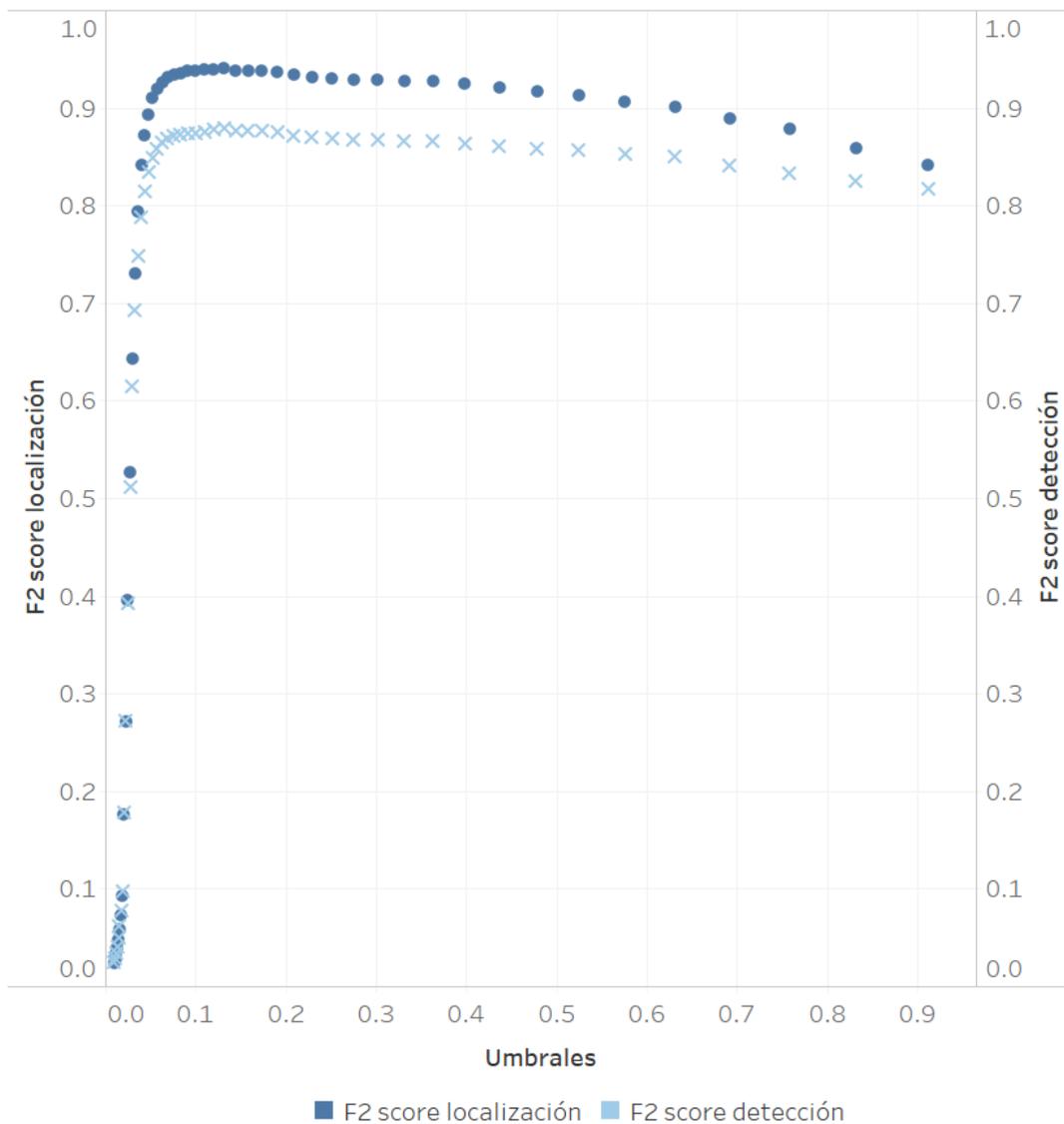


Ilustración 76: F<sub>2</sub> score de localización y de detección en función del umbral para el Modelo 2 Variante 2 para el conjunto de validación