



UNIVERSIDAD DE LOS LAGOS

**DETECCIÓN DE PRESENCIA DE PARÁSITOS EN EXAMEN PARASITOLÓGICO**

**SERIADO DE DEPOSICIONES CON VISIÓN POR COMPUTADORA**

DEPARTAMENTO DE CIENCIAS DE LA INGENIERÍA

INGENIERÍA CIVIL EN INFORMÁTICA

CAMPUS PUERTO MONTT, CHILE

[diegoignacio.munoz@alumnos.ulagos.cl](mailto:diegoignacio.munoz@alumnos.ulagos.cl)

Diego Ignacio Muñoz Viveros

Profesor Guía: Joel Sebastian Torres Carrasco

Co-guía: Carlos Dupré Alvarado

31 de diciembre de 2021



**ACREDITADA 4 AÑOS**  
Diciembre 2016 - Diciembre 2020  
Gestión Institucional  
Docencia de Pregado  
Vinculación con el Medio

[www.ulagos.cl](http://www.ulagos.cl)



## **Resumen**

El examen parasitológico seriado de deposiciones es un examen realizado para la detección de presencia parasitológica en pacientes. El examen se lleva a cabo con la toma de muestras de deposiciones del paciente y posteriormente se hace observación de las mismas por medio de microscopio, en la realización de la observación se documenta la confirmación y clasificación de presencia parasitológica en orden de dar con un tratamiento certero, eficaz y acorde a las detecciones.

El uso de la visión por computadora espera demostrar el aumento en la precisión y velocidad de la detección y clasificación de parásitos con objetivo de reducir incertidumbre y error humano introducido en la observación manual ejercida en el proceso de observación del examen parasitológico seriado de deposiciones. La visión por computadora es la combinación de tecnologías que permite a las computadoras generar inferencia respecto a imágenes estáticas o en movimiento emulando la visión humana por medio de un aprendizaje basado en conjuntos de datos utilizados como ejemplos iniciales de los cuales el modelo generado ajustará sus parámetros.

En el presente trabajo, se realizó una búsqueda de conjuntos de datos sobre parásitos, se desarrolló un modelo de clasificación de imágenes basados en visión por computadora y se probaron diferentes alternativas para mejorar la calidad de la identificación y la clasificación.

Finalmente, este trabajo aporta un marco experimental sobre un conjunto de datos acotado de parásitos, que puede ser liberado a producción, ampliando la cantidad de parásitos a identificar.

**Palabras Clave**— Parasitología, Serializado, Deposiciones, Visión por Computadora, Modelo, Conjunto de Datos, *Underfitting*, *Overfitting*

# Índice general

<b>1. Introducción</b>	<b>1</b>
<b>2. Marco Teórico</b>	<b>4</b>
2.1. Parasitología . . . . .	4
2.1.1. Definición . . . . .	4
2.1.2. Exámenes Parasitológicos . . . . .	5
2.1.3. Procedimiento de exámenes . . . . .	5
2.1.4. Examen Parasitológico Seriado de Deposiciones (EPSD) . . . . .	6
2.2. Visión por Computadora . . . . .	9
2.2.1. Definición . . . . .	9
2.2.2. Modelos de Clasificación . . . . .	11
2.2.3. Ajuste del Modelo Probabilístico . . . . .	11
2.2.4. Conjunto de Datos . . . . .	17
2.2.5. Segmentación . . . . .	22
2.2.6. Procesamiento de Imágenes y Extracción de Características . . . . .	24
2.2.7. Modelo Faster RCNN Inception Resnet . . . . .	26
2.2.8. Modelo Centernet . . . . .	26
2.2.9. Tensorflow API . . . . .	27
<b>3. Formulación del Proyecto</b>	<b>29</b>
3.1. Objetivo General . . . . .	29
3.2. Objetivos Específicos . . . . .	29
3.3. Metodología . . . . .	30
3.3.1. Planificación . . . . .	31

<b>4. Modelo de Identificación y Clasificación de Parásitos</b>	<b>32</b>
4.1. Conjunto de Datos . . . . .	32
4.1.1. Preprocesamiento y Limpieza . . . . .	33
4.2. Modelo . . . . .	35
<b>5. Entrenamiento</b>	<b>37</b>
<b>6. Resultados</b>	<b>38</b>
<b>7. Conclusión</b>	<b>41</b>
<b>Referencias</b>	<b>42</b>
<b>A. Archivo XML</b>	<b>46</b>

# Índice de figuras

2.1.	Frascos para almacenamiento de muestras de deposiciones . . . . .	7
2.2.	Muestra centrifugada de deposiciones . . . . .	7
2.3.	Ejemplos de documentación del examen parasitológico seriado de deposiciones.	8
2.4.	Sistema de seguimiento corporal para dibujar el contorno de la persona en la imagen	9
2.5.	Tipos de clasificadores de objetos; clasificación, clasificación + localización, detección de objetos y segmentación intensa . . . . .	10
2.6.	Gráfica de función sigmoidea donde los valores de $c_i$ controlan la pendiente de la función en $c_2$ . . . . .	12
2.7.	a) representación de la probabilidad de un estado del mundo $w$ y su conjugado, b) función lineal de la distribución y c) función sigmoidea sobre la función lineal representando como los factores de la suma ponderada se anulan en torno a punto de diferenciación haciendo que la función defina áreas donde trabajan . . . . .	13
2.8.	Gráfico de funciones de datos repartidos en dos estados del mundo $w$ , y el área de la función $Pr(x^* \theta)$ para dos estimadores (círculos en rojo y azul). . . . .	14
2.9.	Árbol de clasificación definido para el ejemplo donde los datos fluyen desde la raíz hasta las hojas. El área en gris representa el área de las variables que pueden ser desarrolladas juntas en un entrenamiento incremental. . . . .	16
2.10.	N instancias de árboles generando inferencia sobre un dato $x$ y ponderando su resultado en una votación para generar la predicción final. . . . .	17
2.11.	Ejemplo de aumento de datos, las transformaciones incluyen la imagen original, reflejo horizontal, reflejo vertical, rotaciones de $90^\circ$ , $-90^\circ$ y $60^\circ$ , reescalado a una menores resoluciones, recortes, aumento y disminución de exposición y agregación de ruido aleatorio. . . . .	20

2.12. <i>Original dataset</i> muestra el conjunto de datos original, <i>RandomOverSampler</i> muestra el aumento de datos de manera aleatoria, <i>SMOTE</i> muestra el aumento de datos más localizado y <i>ADASYN</i> muestra el aumento de muestras para definir zonas más marcadas. . . . .	21
2.13. En <i>Under-sampling</i> se eliminan datos del grupo más cuantioso para igualar proporciones, en cambio, en <i>Over-sampling</i> se aumentan de manera artificial los datos del grupo con menos representación. . . . .	22
2.14. Ejemplo de segmentación de instancias donde se puede ver que cada elemento reconocido recibe un color diferente. Elementos como los carteles o árboles del fondo al no contar con etiqueta son ignorados por el modelo. . . . .	23
2.15. Ejemplo de segmentación semántica donde los objetos similares son agrupados bajo la misma etiqueta . . . . .	23
2.16. Ejemplo de segmentación panóptica donde la totalidad de la imagen recibe una clase que diferencia los distintos objetos contenidas en esta. En la parte superior el cielo, los árboles de fondo, el vehículo en primer plano el suelo debajo de este.	24
2.17. Transformación de imagen en su escala de grises. . . . .	25
2.18. Filtro con <i>kernel</i> para resaltar contornos de objetos en la imagen. La matriz se usa para saber las ponderaciones de los píxeles en la suma para el nuevo valor de píxeles en la imagen. Así obtenemos un mapa de características que en este ejemplo nos entrega la información de los límites del animal de la imagen de entrada. . . . .	25
2.19. Ejemplo de detección en su contexto general de Faster RCNN Inception Resnet. .	26
2.20. Primero se puede apreciar la búsqueda de puntos clave, luego la consideración de espacio tridimensional para delimitar las proporciones, luego la profundidad de los objetos, finalmente determina la orientación de movimiento de los objetos detectados. . . . .	27
2.21. Arquitectura de red Centernet. . . . .	27
2.22. Interfaz de monitoreo de <i>Tensorflow API</i> . . . . .	28
3.1. Carta Gantt para desarrollo del proyecto . . . . .	31
4.1. Ejemplo de imagen donde se aprecian dos <i>Hookworms</i> . . . . .	33

4.2. Ejemplo de modificación de archivo <i>xml</i> de imagen. . . . .	33
4.3. Meta data del conjunto de elementos con más de una detección de parásitos. . .	36
4.4. Ejemplo de detección de objetos con <i>Centernet</i> . . . . .	36
6.1. Gráfica de pérdida general del modelo. . . . .	38
6.2. Gráfica de pérdida en detección de centros. . . . .	39
6.3. Gráfica de pérdida en alineamiento delimitador. . . . .	39
6.4. Detección de multiples parásitos contenidos en una sola muestra. . . . .	40
6.5. Muestra de parásito Hookworm obstruido por material indeseado. . . . .	40

# Capítulo 1

## Introducción

Los exámenes, pruebas o mediciones médicas son la base para la evaluación médica y estudios clínicos que permitan elaborar diagnósticos correctos y, consecuentemente, plantear tratamientos adecuados. Los exámenes médicos son procedimientos rigurosos que permiten observar el comportamiento humano siguiendo el método científico para recolectar evidencias, en este caso, del cuerpo humano. Para poder medir de mejor manera todos estos comportamientos, se han desarrollado diferentes técnicas a lo largo del tiempo, como la observación, análisis químico o genético. Por ende, la calidad de los exámenes médicos son un factor clave para que los tratamientos dados tengan efectos positivos y, finalmente, mejorar la calidad de vida de los pacientes.

En la actualidad, los laboratorios médicos son los encargados de realizar una gran cantidad de exámenes relacionados a la observación microscópica de muestras. Estos exámenes son realizados bajo observación humana de manera totalmente manual considerando a un experto llamado tecnólogo médico.

La tecnología médica en el ámbito del conocimiento abarca el estudio e investigación que tiene como objetivo la aplicación de diferentes tipos de tecnologías para mejorar la salud de las personas durante su diagnóstico, desarrollo de la enfermedad y tratamiento aplicado. En el contexto médico, la tecnología médica es la rama de la salud que involucra a profesionales que se forman en los avances tecnológicos para aplicarlos a la medicina y las ciencias de la salud.

Normalmente los exámenes que involucran esta observación microscópica suelen demandar bastante tiempo por muestra; lo cual aumenta la demanda habitual en el área de la salud y, sobre todo, en la salud pública lo que se refleja en demoras en la entrega de resultados.

Otra característica inherente de los exámenes basada en observaciones de imágenes está relacionada al error humano introducido en las observaciones, dependiendo totalmente de la expertiza del tecnólogo médico. En consecuencia, las observaciones resultante contienen clasificaciones erróneas, influyendo en un mal diagnóstico médico. Es así que, en ocasiones puede haber una gran diferencia entre las observaciones de distintos tecnólogos médicos sobre la misma muestra, lo que genera una gran incertidumbre a los estudios clínicos y que obliguen a realizar observaciones adicionales o complementarias.

Uno de estos exámenes basados en observaciones de imágenes es el examen parasitológico seriado de deposiciones que es un examen realizado para la detección de presencia parasitológica en pacientes. Este examen se lleva a cabo con la toma de muestras de deposiciones del paciente y, posteriormente, se hace observación de las muestras por medio de observación microscópica. En la realización de la observación se documenta la confirmación y clasificación de presencia parasitológica en orden de dar con un tratamiento certero, eficaz y acorde a las detecciones. La denominación seriado refiere a las condiciones en las que estas muestras, tres muestras enfrascadas por separado, serán tomadas con la finalidad de tener muestras en distintos ciclos larvales.

Por lo tanto, la problemática presentada en este trabajo radica en lo altamente susceptible al sesgo del observador del examen, es decir, es muy sensible a pérdida o mala clasificación de avistamientos llevando a malos diagnósticos médicos.

En este trabajo se propone una solución inteligente para automatizar el examen parasitológico seriado de deposiciones para poder mejorar la calidad de los resultados y así mejorar la calidad de vida de las personas.

Para realizar este proyecto se pretende usar tecnologías que permitan aprovechar la experiencia de los tecnólogos médicos para mejorar los resultados de las observaciones. Es por ello, que se pretende utilizar tecnologías de visión por computadora para automatizar la observación de muestras y, a su vez, reducir el grado de error. Además, se espera que la implementación permita simplificar y agilizar el procedimiento de observación del examen.

La visión por computadora es la combinación de tecnologías que permite a las computadoras generar inferencia respecto a imágenes estáticas o en movimiento emulando la visión humana por medio de un aprendizaje basado en conjuntos de datos utilizados como ejemplos iniciales de los

cuales el modelo generado ajustará sus parámetros. El uso de la visión por computadora espera demostrar el aumento en la precisión y velocidad de la detección y clasificación de parásitos con objetivo de reducir incertidumbre y error humano introducido en la observación manual ejercida en el proceso de observación del examen parasitológico seriado de deposiciones.

El desafío técnico encontrado está relacionado a la estructuración de un conjunto de datos de imágenes microscópicas que contengas la información suficiente para generar un modelo robusto y de altas prestación con gran capacidad de generalización para el examen parasitologico seriado de deposiciones.

En adelante, el documento se estructura de la siguiente manera. El capítulo 2, presenta los conocimientos básicos para entender el desarrollo de este proyecto de título. El capítulo 3, define los objetivos, metodología y planificación del proyecto. Finalmente, el capítulo 7, entrega algunas ideas a desarrollar a futuro en el proyecto.

# Capítulo 2

## Marco Teórico

### 2.1. Parasitología

#### 2.1.1. Definición

La parasitología es la rama de las ciencias biológicas dedicada a el estudio de organismos, denominados parásitos, que dependen de otro para poder sobrevivir y que ocasionan grandes daños a las especies de las cuales dependen, relación llamada parasitismo.

La parasitología es una disciplina con aplicación en campos variados como medicina, farmacología y veterinaria. Es utilizada en la investigación de parásitos que pueden producir enfermedades en plantas y animales con objeto de analizar, diagnosticar y posteriormente establecer un tratamiento óptimo para poder curarlas y erradicarlas (FE., 2002).

Gran parte de los parásitos más difícil de tratar son los que se alojan en el interior del organismo, lo que puede ingresar por vía oral o fluidos y gran parte de estos pueden alojarse en el sistema digestivo, principalmente en estomago e intestino (ISP, 2021).

En el contexto de la medicina, la área de **Tecnología Médica** en su especialización de parasitología esta encargada de la realización y análisis de exámenes con la finalidad de diagnosticar amenazas relacionadas a la disciplina (Weitz, 2013).

## 2.1.2. Exámenes Parasitológicos

Existen muchos tipos de análisis de laboratorio para diagnosticar enfermedades parasitarias. El tipo de análisis que solicite el médico se basará en sus signos y síntomas presentados durante la consulta médica, cualquier otra afección médica que pueda tener y sus antecedentes de viajes.

El análisis de laboratorio se lleva a cabo con las observaciones de muestras entregadas al laboratorio por el médico tratante. Estas muestras dependen de la búsqueda de los parásitos sospechados y sus posibles ubicaciones, siendo estas muestras de la forma de sangre, heces, muestras urogenitales, esputo, aspirados o biopsias. La especificidad de los exámenes puede variar en la capacidad de detectar diferentes especies o realizar búsquedas de manera particular (Huapaya, 2002).

Estos exámenes se pueden dividir en dos categorías:

- Invasivos: la adquisición de la muestra requiere intervención quirúrgica algún tipo como las biopsias.
- No invasivos: la toma de la muestra presenta un método de obtención que no involucra una intervención invasiva al paciente como serían muestras de sangre o heces.

## 2.1.3. Procedimiento de exámenes

Para la realización de los exámenes se procede de las siguientes formas: (CDC, 2016)

1. **Exámenes de muestra de sangre:** la muestra es tintada y analizada por goteo grueso y/o fino con un microscopio. El goteo fino es una forma de repartir la muestra en un portaobjeto<sup>1</sup> a manera de dejar una capa delgada y uniforme en la cual realizar observaciones, el goteo grueso por otro lado, consiste en soltar una gota de muestra de forma que la tensión superficial de la muestra mantenga su forma circular para dejar decantar las células contenidas en la muestra al fondo. La tinción es el proceso en el cual se suman compuestos a la muestra que reaccionan a componentes conocidos con el fin de teñir componente para facilitar la visualización.

<sup>1</sup>Placa de acrílico transparente usada para manejo de muestras para microscopio

2. **Endoscopia/Colonoscopia:** Consiste en la inserción en la boca (endoscopia) o el recto (colonoscopía) de una sonda con la cual el médico, normalmente un gastroenterólogo, examina las cavidades en busca de presencia parasitaria y/o lesiones de relacionadas a presencia parasitaria.
3. **Exámenes parasitológico seriado de deposiciones:** consiste en el análisis de tres muestras seriadas<sup>2</sup> de heces tintadas con observación por microscopio . La observación se realiza por goteo fino.
4. **Resonancia Magnética (RM), Tomografía axial computarizada (TAC):** Pruebas realizadas para buscar enfermedades parasitarias que pueden provocar lesiones en los órganos.

#### **2.1.4. Examen Parasitológico Seriado de Deposiciones (EPSD)**

Como se menciona anteriormente, el examen parasitológico seriado de deposiciones (EPSD) es un examen que busca estudiar la presencia parasitaria intestinal por medio de observaciones de muestras de heces. La realización de este examen es llevado a cabo por un tecnólogo médico y su principal objetivo es documentar las observaciones realizadas.

La realización del examen en detalle se muestra a continuación:(Jercic, 2017)

1. Una vez se reconoce la necesidad de la realización del examen, el médico tratante pedirá la preparación de tres frascos de muestras con 10-20 ml de líquido fijador<sup>3</sup>.
2. Se hace entrega al paciente de los frascos junto a una nota con las instrucciones para la toma de muestras, conservación de las mismas y precauciones a tomar 2.1.

---

<sup>2</sup>Muestras tomadas con intervalos de tiempo equidistantes con objetivo de muestrear sin que se pierdan ciclos larvarios evitando excluir avistamientos

<sup>3</sup>Líquido conservantes, normalmente formaldehído



Figura 2.1: Frascos para almacenamiento de muestras de deposiciones

3. Cuando se reciben las muestras y se valida su tamaño, los tiempos en los que se tomaron y la forma en que se conservaron, estas pasaran a ser entregadas al laboratorio donde serán almacenadas y/o examinadas.
4. Las muestras son retiradas del área de refrigeración, en el caso que se encuentren en conservación, y se proceden a introducir una porción de cada frasco en tres probetas diluidas en el líquido fijador.
5. Se centrifugan las muestras en orden de separar el material fecal del líquido fijador para poder extraerlo 2.2.



Figura 2.2: Muestra centrifugada de deposiciones

6. En caso de que el tecnólogo lo considere necesario, tintar las muestras con soluciones reactivas que resalten componentes específicos para facilitar las observaciones.

7. Dejar caer una gota de cada una de las muestras en un portaobjeto y esparcir para observación de goteo fino.
8. Hacer observación de las muestras con gran detalle en el microscopio y documentar hallazgos en caso de haber 2.3.

Examen Parasitológico Seriado de Deposiciones. (EPSD)  
- Método de Burrows Modificado.  
- (x) Número de muestras procesadas.  
- Resultado: **No se observan elementos parasitarios.**  
- Observaciones:

Examen Parasitológico Seriado de Deposiciones. (EPSD)  
- Método de Burrows Modificado.  
- (x) Número de muestras procesadas.  
- Resultado: **Se deben informar la presencia de todos los elementos observados en la muestra.**  
- Observaciones:

- (a) Contenido textual en caso de no hallar presencia parasitológica (b) El informe debe contener un listado detallado de las observaciones incluyendo nombre del parásito y cantidad de avistamientos.

Figura 2.3: Ejemplos de documentación del examen parasitológico seriado de deposiciones.

9. En caso de no detectar presencia parasitaria, documentar que no se observa presencia en la muestra.

Para la correcta realización del examen existen ciertas consideraciones expuestas a continuación:

- La toma de muestras debe abarcar un período mínimo de cinco días.
- La duración de la observación microscópica puede durar alrededor de 45 minutos y es dependiente de la habilidad y experiencia del tecnólogo.
- En caso de encontrar alguna anomalía en la recepción de las muestras por parte del paciente, se debe pedir que este haga la toma nuevamente en frascos diferentes.
- La muestra que no se utilice durante la observación deberá ser conservada en refrigeración.
- En caso de existir dudas sobre la observación de las muestras, se puede realizar una segunda observación.

## 2.2. Visión por Computadora

### 2.2.1. Definición

La visión por computadora es el campo de la inteligencia artificial (IA) que permite a los computadores y sistemas derivados a extraer información útil de imágenes digitales, videos y otras entradas visuales y tomar acciones o hacer recomendaciones basadas en esta información. Si la IA permite a los computadores pensar, la visión por computadora les permite ver, observar y entender (IBM, 2020).

La visión por computadora, así como muchas otras ramas de la disciplina de IA, logra la tarea de inferir gracias al aprendizaje de máquinas. El aprendizaje de maquinas consiste en el entrenamiento de modelos de IA en base a conjuntos de datos que harán que los distintos modelos aprendan a reconocer patrones y generen la lógica interna necesaria para permitirles inferir de manera correcta.

Para definir formalmente el problema de la visión por computadora (Prince, 2012) partimos tomando un dato visual  $x$  y lo usamos para inferir un estado del mundo  $w$ . El estado del mundo  $w$  puede ser de naturaleza continua (e.g. La posición tridimensional del modelo de un cuerpo 2.4) o discreto (e.g. La presencia o ausencia de un objeto particular 2.5). Cuando el estado es continuo, la inferencia es llamada regresión. Cuando el estado es discreto, la inferencia es llamada clasificación.

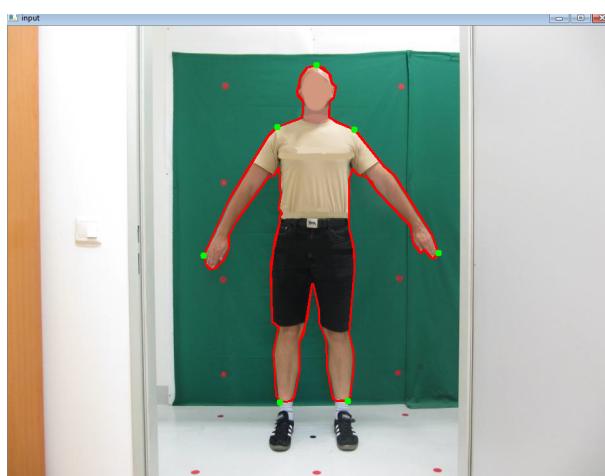


Figura 2.4: Sistema de seguimiento corporal para dibujar el contorno de la persona en la imagen

Esta definición nos permite definir el proceso de inferencia como  $Pr(w|x)$ , es decir, la proba-

bilidad de que cierto estado del mundo  $w$  dado  $x$  dato visual. Se puede notar rápidamente que a partir de esta definición solo se puede inferir sobre estados  $w$  conocidos. Esto en práctica se resuelve utilizando un estimador  $\hat{w}$  que resulte suficientemente cercano.

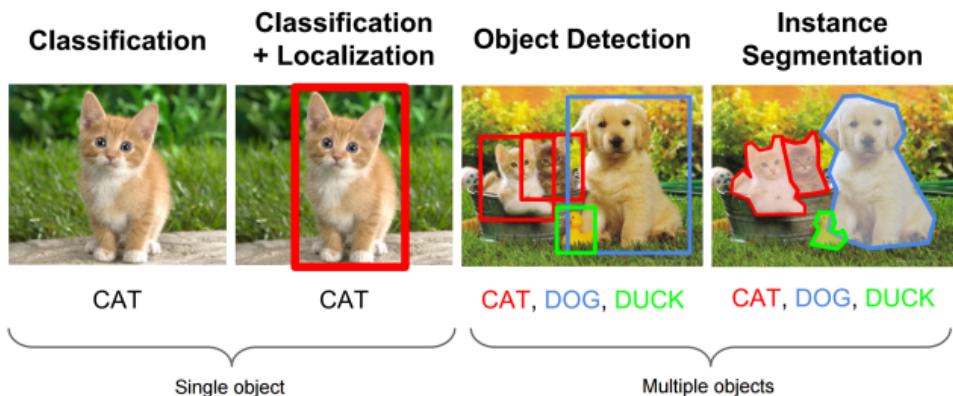


Figura 2.5: Tipos de clasificadores de objetos; clasificación, clasificación + localización, detección de objetos y segmentación intensa

Para resolver un problema de visión de esta naturaleza se necesitan tres componentes:

- Un modelo matemático que represente los datos visuales  $x$  y los estados del mundo  $w$ . El modelo representa una familia de posibles relaciones entre  $x$ ,  $w$  y la relación particular es determinada por el modelo de parámetros  $\theta$ .
- Un algoritmo de aprendizaje que nos permita ajustar los parámetros  $\theta$  usando ejemplos de pares de entrenamiento  $\{x_i, w_i\}$ , donde sabemos ambas medidas y el estado relacionado.
- Un algoritmo de inferencia que tome una nueva observación  $x$  y use el modelo para calcular  $Pr(w|x, \theta)$  sobre el estado del mundo  $w$ .

Los modelos que relacionan los datos  $x$  a el mundo  $w$  caen en una de dos categorías:

1. modelar la contingencia del estado del mundo en los datos  $Pr(w|x)$  o
2. modelas la contingencia de los datos sobre el estado del mundo  $Pr(x|w)$ .

El primer tipo de modelos es denominado discriminativo. El segundo es denominado generativo; aquí, construimos un modelo de probabilidad sobre los datos y esto puede ser usado para generar nuevas observaciones.

Para efectos de este informe se estudiara el primer tipo de modelo.

## 2.2.2. Modelos de Clasificación

Al modelar  $Pr(w|x)$ , elegimos una forma de distribución apropiada para  $Pr(w)$  sobre los estados del mundo  $w$  y hacer la distribución de los parámetros una función de los datos  $x$ . Si el mundo de estados es continuo, lo definimos como una distribución normal con media  $\mu$  en función de  $x$  (Fuchs, 2017).

El valor que nos entrega la función también en un conjunto de parámetros definidos,  $\theta$ . Dado que la distribución depende tanto de los datos  $x$  y los parámetros  $\theta$ , escribimos que la función es  $Pr(w|x, \theta)$  y nos referimos a ella como distribución posterior.

El objetivo del algoritmo de aprendizaje es el de ajustar los parámetros  $\theta$  usando pares de entrenamiento  $\{x_i, w_i\}$ .

## 2.2.3. Ajuste del Modelo Probabilístico

Nos referimos como aprendizaje al proceso de ajustar el modelo probabilístico al conjunto de datos  $\{x_i\}$  ajustando los parámetros  $\theta$  y es de vital importancia que el modelo pueda calcular la probabilidad de un nuevo dato  $x^*$  (Tiño, 2016).

### Regresión Logística

Para los modelos de clasificación consideraremos la regresión logística (Geert, 2020), la cual independientemente de su nombre es un modelo que puede ser aplicado a clasificación. La regresión logística es un modelo discriminativo; nosotros seleccionamos una distribución de probabilidad sobre un estado del mundo  $w \in \{0, 1\}$  y hacemos estos parámetros consistentes en los datos observados  $x$ . Como los estados del mundo son binarios, lo describimos con una distribución de Bernoulli y hacemos el parámetro  $\lambda$  (que indica la probabilidad de que el estado del mundo tome el valor  $w = 1$ ) una medida en función de  $x$ .

Es importante destacar, no podemos simplemente hacer el parámetro  $\lambda$  una función lineal  $\phi_0 + \phi^T x$  de las medidas; una función lineal puede retornar cualquier valor, pero el parámetro  $\lambda$  debe caer entre 0 y 1. Consecuentemente, primero debemos calcular la función lineal y luego pasarl a través de la función sigmoidea que evalúa el rango  $[-\infty, \infty]$  a  $[0, 1]$ . El modelo final queda

2.7

$$Pr(\omega|\phi_0, \phi, x) = Bern_{\omega}[\text{sig}[\alpha]],$$

donde  $\alpha$  es denominado la activación y es dado por la función lineal

$$\alpha = \phi_0 + \phi^T x$$

y donde la función sigmoidea (Wood, 2020) es de la forma 2.6

$$\text{sig}(x) = \frac{1}{1 + e^{-x}}$$

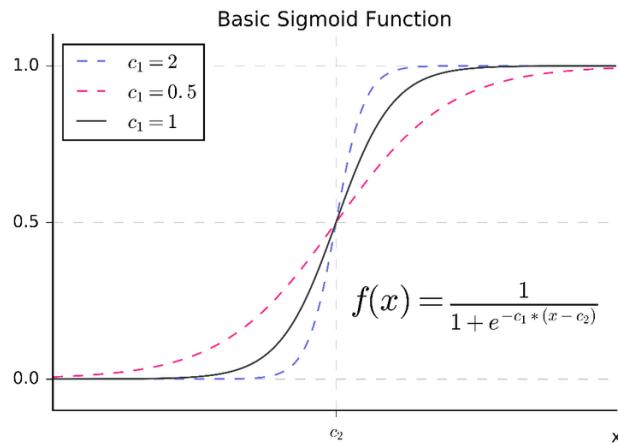


Figura 2.6: Gráfica de función sigmoidea donde los valores de  $c_i$  controlan la pendiente de la función en  $c_2$

Así como la activación  $\alpha$  tiene a infinito esta función tiende a uno. Y así como tiende a menos infinito tiende a cero. Cuando  $\alpha$  es cero, la función logística sigmoidea retorna un valor de un medio. Para datos unidimensionales de  $x$ , el conjunto de efectos de esta transformación es la de describir una curva sigmoidea relacionando  $x$  con  $\lambda$ . La posición horizontal de la sigmoidea es determinada por el lugar donde la función lineal  $\alpha$  cruza cero y su pendiente depende del gradiente  $\phi_1$ .

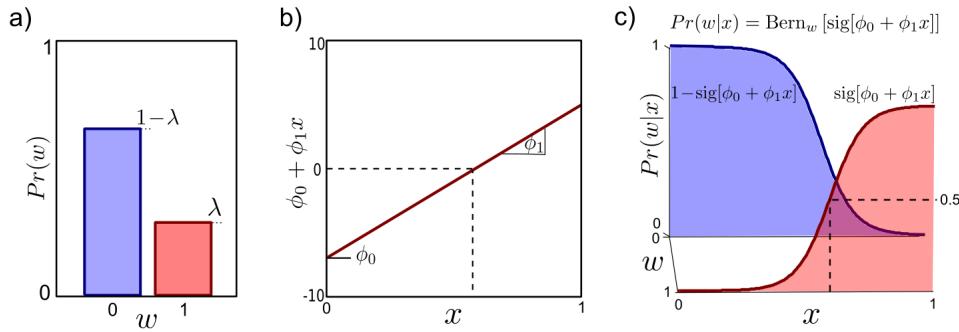


Figura 2.7: a) representación de la probabilidad de un estado del mundo  $w$  y su conjugado, b) función lineal de la distribución y c) función sigmoidea sobre la función lineal representando como los factores de la suma ponderada se anulan en torno a punto de diferenciación haciendo que la función defina áreas donde trabajan

### Método Bayesiano

En el método bayesiano (Rennie, 2003; Backlund, 2017) dejamos de intentar estimar un solo valor de los parámetros  $\theta$  y aceptamos lo obvio; habrán muchos valores de parámetros que serán compatibles con los datos. Calculamos la distribución de probabilidad  $Pr(\theta|x_{1\dots I})$  sobre los parámetros  $\theta$  basados en los datos  $\{x_i\}_{i=1}^I$  usando el teorema de Bayes

$$Pr(\theta|x_{1\dots I}) = \frac{\prod_{i=1}^I Pr(x_i|\theta)Pr(\theta)}{Pr(x_{1\dots I})}$$

Evaluando la distribución predictiva es más difícil para el caso bayesiano dado que no tenemos ningún modelo estimado pero en su lugar encontramos una distribución de probabilidad sobre posibles modelos. Así, calculamos

$$Pr(x_*|x_{1\dots I}) = \int Pr(x_*|\theta)Pr(\theta|x_{1\dots I})d\theta,$$

esto puede ser interpretado como: el término  $Pr(x_*|\theta)$  es la predicción para un valor dado de  $\theta$ . Así, la integral puede entenderse como la suma ponderada de los diferentes parámetros  $\theta$ , donde la ponderación está dada por la distribución de probabilidad posterior  $Pr(\theta|x_{1\dots I})$  sobre los parámetros (asumiendo que los parámetros son diferentes).

Los cálculos de densidad predictiva para el bayesiano, máximo a posteriori u máxima verosimilitud puede unificarse si consideramos que estos dos últimos son distribuciones de probabili-

dades especiales sobre los parámetros donde toda la densidad está en  $\hat{\theta}$ . Más formalmente, los consideramos funciones delta centradas en  $\hat{\theta}$ . Una función delta  $\delta(z)$  es una función que integra a uno, y retorna cero en todo su dominio excepto  $z = 0$ .

Entonces escribimos

$$Pr(x^* | x_{1 \dots I}) = \int Pr(x^* | \theta) \delta[\theta - \hat{\theta}] d\theta,$$

$$= Pr(x^* | \theta),$$

Finalmente vemos calculamos las probabilidades evaluando la probabilidad de los datos bajo el modelo con el que estimamos los parámetros 2.8.

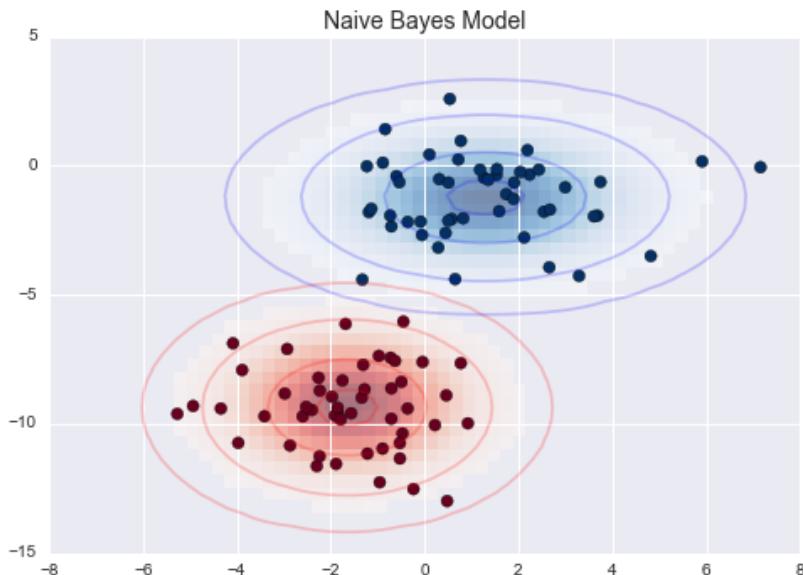


Figura 2.8: Gráfico de funciones de datos repartidos en dos estados del mundo  $w$ , y el área de la función  $Pr(x^* | \theta)$  para dos estimadores (círculos en rojo y azul).

## Árboles de Decisión

Para implementar ese modelo (Rokach, 2007) comenzamos partiendo el espacio de datos en distintas regiones y aplicamos diferentes clasificaciones a cada región.

El modelo de regresión ramificada logística tiene activaciones,

$$\alpha_i = (1 - g[x_i, \omega])\phi_0^T x_i + g[x_i, \omega]\phi_1^T x_i$$

El término  $g[x, \omega]$  es una función de activación que retorna un número entre 0 y 1. Si esta función de activación retorna 0, entonces la activación será  $\phi_0 x_i$ , por otro lado si retorna 1, la activación será  $\phi_1 x_i$ . Si la activación retorna un valor intermedio, entonces la activación será la suma ponderada de los dos componentes. La propia función de activación depende de los datos  $x_i$  y toma parámetros  $\omega$ . Este modelo induce un complejo límite de decisión no linear donde las dos funciones lineares  $\phi_0 x_i$  y  $\phi_1 x_i$  son especializadas en diferentes regiones del espacio de datos.

La función de activación puede tomar muchas formas, pero una obvia posibilidad es la de usar un segundo modelo de regresión lineal. Vale decir, calculamos una función lineal  $\omega^T x_i$  de los datos que son pasados a través de una sigmoidea logística, es decir,

$$g[x_i, \omega] = \text{sig}[\omega^T x_i]$$

Para que este modelo aprenda debemos maximizar  $L = \sum_i \log[Pr(w_i|x_i)]$  una probabilidad logarítmica de los pares de datos de entrenamiento  $\{x_i, w_i\}_{i=1}^I$  con respecto a todos los parámetros  $\theta = \{\phi_0, \phi_1, w\}$ . Normalmente esto puede lograr usando métodos de optimización no lineal.

Podemos extender esta idea para crear una estructura de árbol jerárquica anidando funciones de activación. Por ejemplo,

$$\begin{aligned} \alpha_i &= (1 - g[x_i, w]) [\phi_0^T x_i + (1 - g[x_i, w_0])\phi_{00}^T x_i + g[x_i, w_0]\phi_{01}^T x_i] \\ &\quad + g[x_i, w] [\phi_1^T + (1 - g[x_i, w_1])\phi_{01}^T x_i + g[x_i, w_1]\phi_{11}^T x_i] \end{aligned} \tag{2.1}$$

Esto es un ejemplo de árbol de clasificación 2.9.

Para aprender los parámetros  $\theta = \{\phi_0, \phi_1, \phi_{00}, \phi_{01}, \phi_{10}, \phi_{11}, w, w_0, w_1\}$  podemos tomar una aproximación incremental. En la primera etapa ajustamos la parte superior del árbol, ajustando los parámetros  $w, \phi_0, \phi_1$ . Luego ajustamos la rama de la izquierda, ajustando los parámetros  $w_0, \phi_{00}, \phi_{01}$  y así con la rama de la derecha, ajustando  $w_1, \phi_{01}, \phi_{11}$  y así sucesivamente.

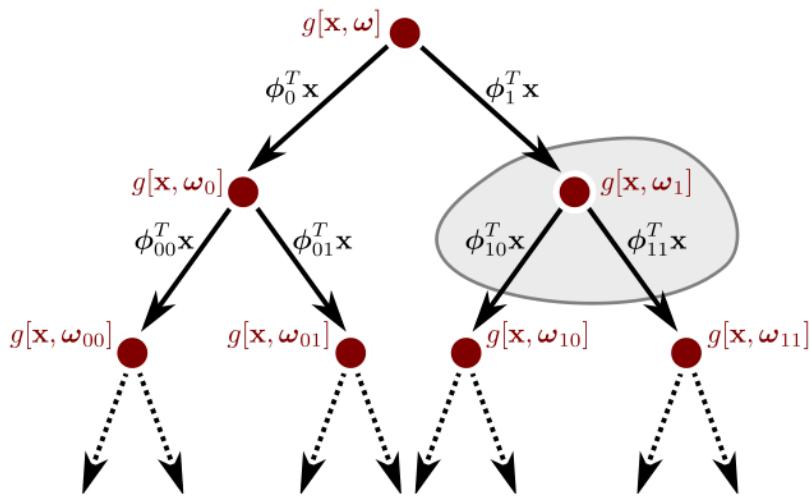


Figura 2.9: Árbol de clasificación definido para el ejemplo donde los datos fluyen desde la raíz hasta las hojas. El área en gris representa el área de las variables que pueden ser desarrolladas juntas en un entrenamiento incremental.

### Bosque de Árboles Aleatorio

Esta idea se volvió popular en la problemas de clasificación multiclas usando la idea de árbol de decisión definida anteriormente.

Un bosque aleatorio (Ho, 2016) es una colección de árboles aleatorios, cada uno de estos usa un conjunto de funciones elegidas de manera aleatoria. Cuando se promedian todas las probabilidades de cada árbol, es decir,  $Pr(x * | w_i)$  predicha por cada árbol, se produce una clasificación mucho más robusta. Una forma de pensarla es aproximarla al método bayesiano; donde construimos una respuesta final tomando una suma ponderada de las predicciones propuestas por los distintos conjuntos de parámetros 2.10.

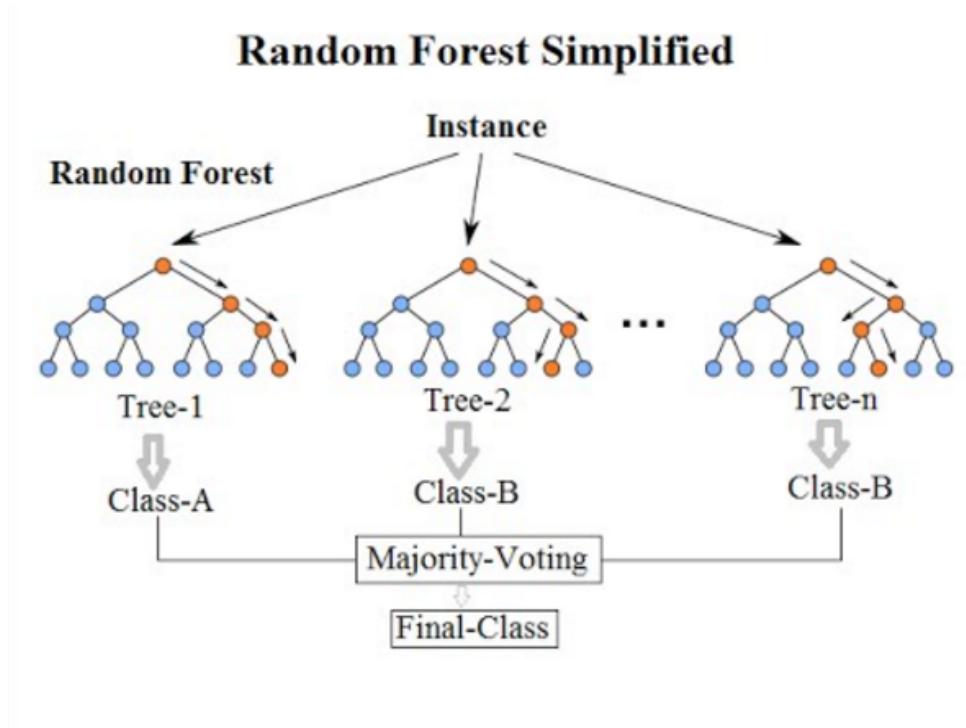


Figura 2.10: N instancias de árboles generando inferencia sobre un dato  $x$  y ponderando su resultado en una votación para generar la predicción final.

## 2.2.4. Conjunto de Datos

El manejo de los conjuntos de datos (Byjus, 2012b) es una de las tareas más importantes a la hora de diseñar un modelo de IA, ya que la información y como esté clasificada marcará como el modelo podrá desempeñarse en un entorno de uso general o específico que no contenga información vista con anterioridad.

La estructura general de un conjunto de datos en modelos de visión por computadora suele consistir en un conjunto de imágenes o videos junto a etiquetas que muestran la información contenida en estas, normalmente en la forma de coordenadas que delimitan el área en donde se localiza en la imagen el objeto a identificar y/o clasificar, junto a la etiqueta que identifica que es o que acción esta ocurriendo.

Es importante contar con un conjunto de datos que contenga gran cantidad de ejemplos relacionados a la tarea que se espera lograr, esto con objetivo de generalizar los patrones e información obtenida por parte del modelo durante el proceso de entrenamiento. La cantidad de información que se requiera estará dada por la complejidad en el modelo a entrenar y la complejidad de la

tarea a desarrollar, y el análisis de estos parámetros dependerá del equipo de desarrollo. Cuando en entrenamiento carece de un conjunto de datos contundente o el modelo es muy simple para la tarea ocurre *underfitting*, esto es, cuando el modelo realiza inferencias escazas y/o erróneas.

Para un hacer buen provecho de la información existen diferentes tratamientos realizables a los conjuntos de datos como lo son el limpieza de datos, aumento de datos y el balanceo de datos.

### Limpieza de Datos

La limpieza de datos es un procedimiento que nos ayuda a filtrar datos erróneos y manipular información faltante para no causar una mayor pérdida de datos (Byjus, 2012a). Este procedimiento es descrito como la intuición de los datos, dado que no existen muchas más formas de entender un conjunto de datos que analizándolo de manera manual. Generalmente el manejo de la limpieza de datos suele ser una de las partes más demandantes en el proceso de desarrollo de un modelo de inteligencia artificial.

Los problemas que se presentan en conjuntos de datos que debes ser solventados son:

- **Datos faltantes:** Existen situaciones en las que no todos los datos de una entrada tendrán asignados un valor. Si bien es factible eliminar todas las entradas que contengan datos perdidos, en la mayoría de los casos esto termina descartando excesiva cantidades de entradas (hay conjuntos de datos que pueden tener datos perdidos en todas sus entradas).

Una alternativa para cuando los datos faltantes tienen estar agrupados en su mayoría en un campo en particular es eliminar el campo por completo. Esto valido dado que el campo de por si mostraba no tener mucha utilidad.

La forma más común de manejar datos faltantes es usar un valor por defecto que en ultima instancia servirá para que el modelo reconozca que existen situaciones donde los datos no se encuentran disponibles. Esto es básicamente porque siempre se puede extraer información del echo de que no existen entradas en un campo en particular (Tatman, 2020a).

- **Distribución de datos inconsistentes:** Esto ocurre cuando la escala o el formato de los datos no esta bien definido en los campos, vale decir, tener números en campos que describen texto o números en rangos que no corresponden.

Cuando se habla de inconsistencia en tipo de datos solo basta con reconocer el tipo de dato del campo para hacer la corrección del resto de los datos, revisando que la transformación de los datos sea consistente al objetivo final.

Cuando la inconsistencia corresponde a rangos se implementan dos alternativas, escalado y normalización.

El escalado corresponde a una transformación en el rango de los datos, esto convierte los datos tomando el mínimo y máximo valor en los campos y escalándolos a valores entre 0 y 1.

La normalización es una transformación que convierte la distribución de los datos en una distribución normal Gaussiana. Este método es utilizado en datos destinados a modelos de predicción estadística (Tatman, 2020c).

- **Datos inconsistentes:** Es el tipo de dato que requiere manipulación acorde a las necesidades, estos datos pueden ser desde texto que registra entradas diferentes por tener su versión iniciando con mayúsculas y su versión en minúsculas hasta tipos de datos diferentes en las mismas columnas.

Para la reparación de estos datos se debe hacer estudio de los datos y la información que se espera de ellos y es totalmente sujeta a quien manipula los datos (Tatman, 2020b).

## Aumento de Datos

Muchos de los desarrollos de modelos actuales presentan una característica en común, fueron entrenados con cantidades de datos que pasan el orden de magnitud de los cientos milos o millones. Esto es un importante dato a considerar cuando se tienen modelos que no cuentan con un conjunto de datos lo suficientemente amplio 2.11 (J. Wang, 2017).

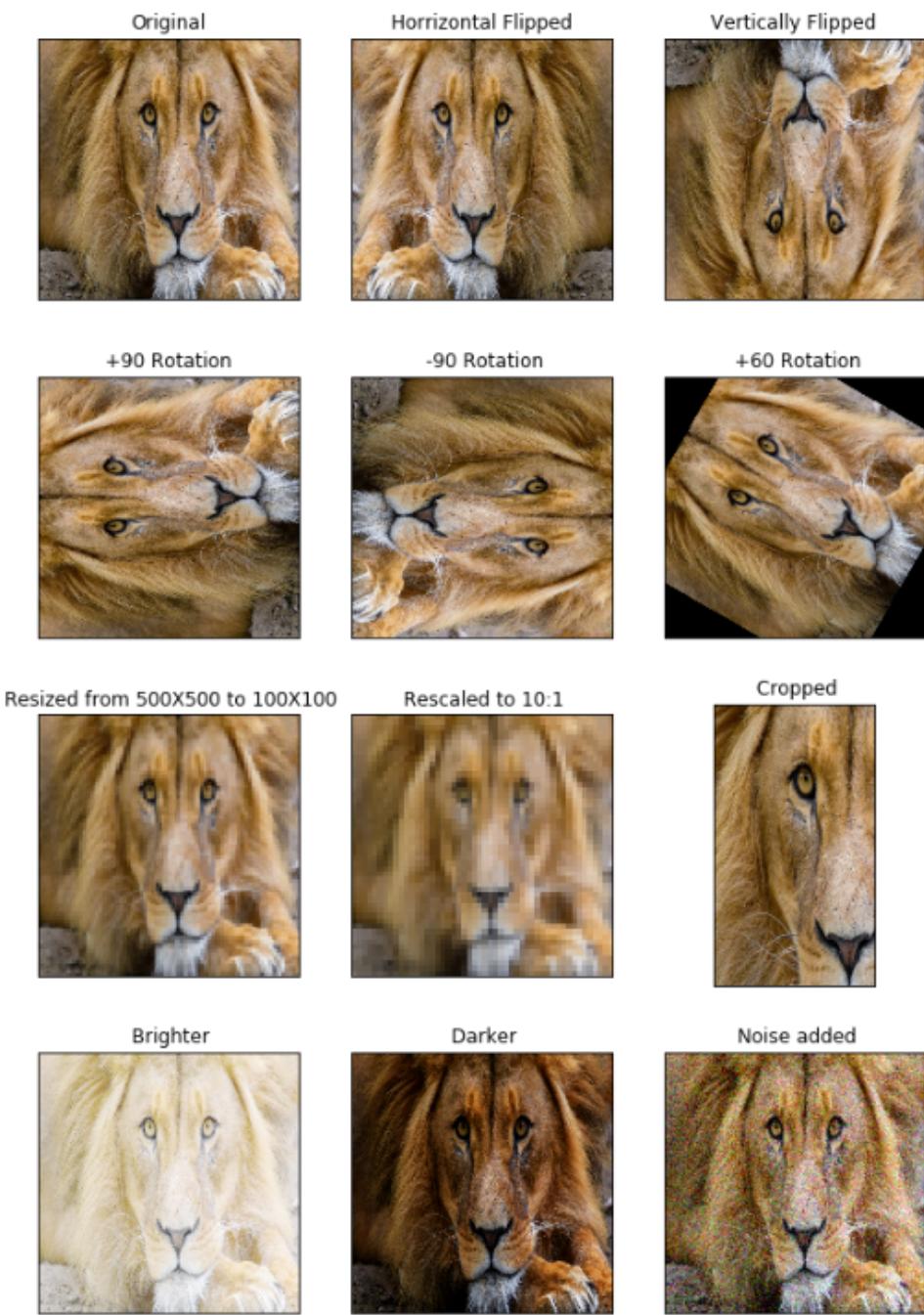


Figura 2.11: Ejemplo de aumento de datos, las transformaciones incluyen la imagen original, reflejo horizontal, reflejo vertical, rotaciones de 90°, -90° y 60°, reescalado a una menores resoluciones, recortes, aumento y disminución de exposición y agregación de ruido aleatorio.

Para esto es que se hacen estudios respecto el aumento artificial de datos, que consisten en resumidas cuentas en el uso de datos pre-existentes en el conjunto de datos actual para crear nuevas entradas que representen información similar. Hay estudios que muestran aumentos de rendimiento sobre el 30 % (K, 2020) sobre modelos que no la utilizan.

### Aumento de datos numéricos

Las técnicas de aumento utilizadas (Khare, 2011) en las aplicaciones de aprendizaje profundo dependen del tipo de datos. Para aumentar los datos numéricos simples, son populares técnicas como *SMOTE*, *ADASYN* o *RandomOverSampler*. Estas técnicas se utilizan generalmente para abordar el problema del desequilibrio de clases en las tareas de clasificación.

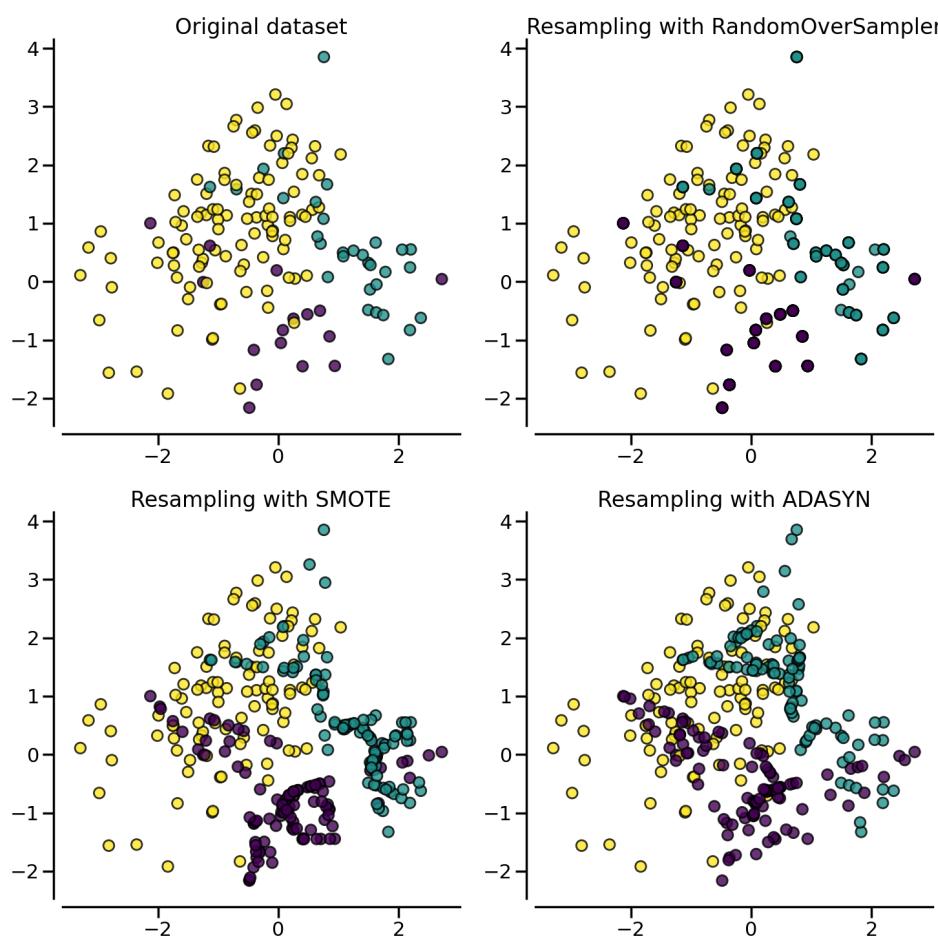


Figura 2.12: *Original dataset* muestra el conjunto de datos original, *RandomOverSampler* muestra el aumento de datos de manera aleatoria, *SMOTE* muestra el aumento de datos más localizado y *ADASYN* muestra el aumento de muestras para definir zonas más marcadas.

Para los datos no estructurados, como las imágenes y el texto, las técnicas de aumento varían desde simples transformaciones hasta datos generados por redes neuronales, en función de la complejidad de la aplicación (Nayak, 2020).

### Balonceo de Datos

En cierto tipo de conjuntos de datos ocurre que cierta concentración de datos sobrepasa de sobre manera a otra, causando desbalanceo de datos. Una forma de sobre llevar esto es particionando el segmento de datos más abundante para poder representar toda la información con el mismo nivel de importancia 2.13 (Dey, 2017).

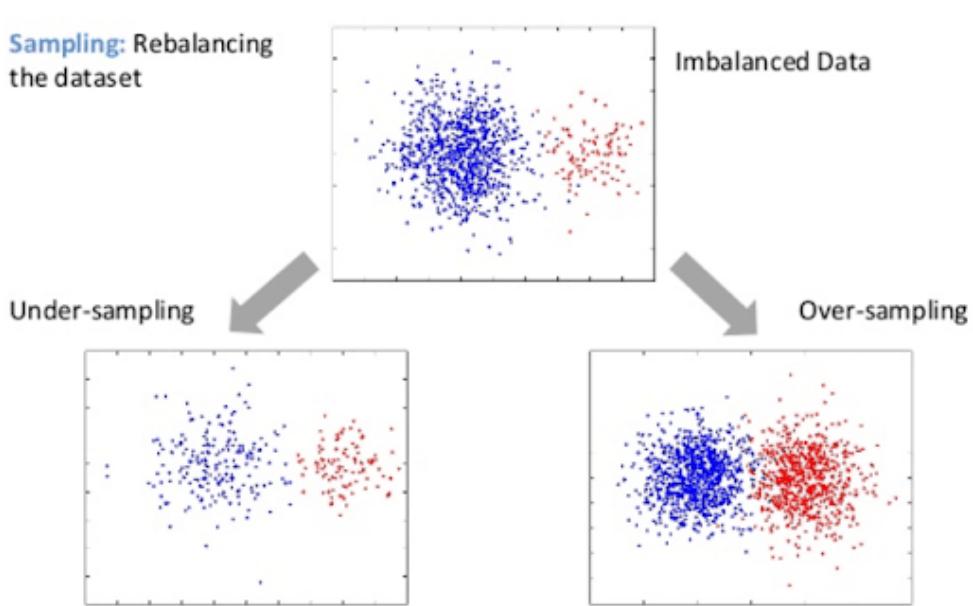


Figura 2.13: En *Under-sampling* se eliminan datos del grupo más cuantioso para igualar proporciones, en cambio, en *Over-sampling* se aumentan de manera artificial los datos del grupo con menos representación.

Es importante reconocer las desventajas a lo que esto puede conllevar; esto hace dejar de lado gran cantidad de datos, esto es proporcional al conjunto que represente la minoría de datos en el conjunto, esto culminando en *underfitting*.

### 2.2.5. Segmentación

La segmentación (Labelbox, 2020) de imágenes es el tipo de modelo que delimita los contornos de los elementos contenidos en la entrada visual, en este modelo cada pixel pertenece a una clase. Es usado para identificar imágenes para aplicaciones que requieren gran nivel de precisión. La salida es una máscara que delimita la forma del objeto en la imagen. La segmentación de imágenes viene en muchas formas (segmentación semántica, intensa, panóptica, etc.), la práctica de

segmentación de imágenes generalmente describe la necesidad de anotar cada pixel de la imagen con una clase.

- **Segmentación de instancias:** cada instancia de un objeto es etiquetada por separado y los segmentos sin instancia son ignorados.



Figura 2.14: Ejemplo de segmentación de instancias donde se puede ver que cada elemento reconocido recibe un color diferente. Elementos como los carteles o árboles del fondo al no contar con etiqueta son ignorados por el modelo.

- **Segmentación semántica:** Aquí se consideran todas las regiones de la imagen y los objetos del mismo tipo reciben la misma etiqueta.



Figura 2.15: Ejemplo de segmentación semántica donde los objetos similares son agrupados bajo la misma etiqueta

- **Segmentación panóptica:** es una combinación de segmentación semántica y segmentación

de instancias, de modo que a todos los píxeles se les asigna una etiqueta de clase y todas las instancias de objetos están segmentadas de forma única.



Figura 2.16: Ejemplo de segmentación panóptica donde la totalidad de la imagen recibe una clase que diferencia los distintos objetos contenidas en esta. En la parte superior el cielo, los árboles de fondo, el vehículo en primer plano el suelo debajo de este.

### 2.2.6. Procesamiento de Imágenes y Extracción de Características

Es importante saber reducir la complejidad del análisis de imágenes sabiendo reconocer que tipo de solución necesitamos para nuestro problema. Para ello existen multiples transformaciones que permiten simplificar la entrada visual que recibirá el modelo en orden de facilitar la tarea de extracción de características claves (Prince, 2012).

#### Transformación por Pixel

Es la transformación más directa de imágenes (Y. Wang, 2020), consisten en la modificación de píxeles individuales asumiendo la estructura bidimensional. Se dice que la transformación modifica el pixel  $p_{ij}$  donde  $i, j$  son las coordenadas del pixel en el arreglo de la imagen  $P$ .

#### Escala de Grises

Es convertir la imagen de entrada a color a su versión a escala de grises. Este filtro es especialmente útil para reducir el arreglo de píxeles de valores rgb a un simple valor que representa el nivel de blanco del pixel, esto es útil en los casos donde los colores no son relevantes en la extracción de características (Y. Wang, 2020).



(a) Versión de imagen a color.

(b) Versión de imagen en escala de grises.

Figura 2.17: Transformación de imagen en su escala de grises.

### Filtros Lineales

Son transformaciones de píxeles  $x_{ij}$  que consideran una suma ponderada del valor de los píxeles adyacentes (Ludwig, 2020), es decir, siendo una imagen  $P$  a la cual se aplicará un filtro denominado *kernel*  $F$ , el cual tendrá entradas  $f_{mn}$  donde  $m \in \{-M, M\}$  y  $n \in \{-N, N\}$ . Finalmente quedamos con un filtro de la forma 2.18

$$x_{ij} = \sum_{m=-M}^M \sum_{n=-N}^N p_{i-m, j-n} f_{m,n}$$

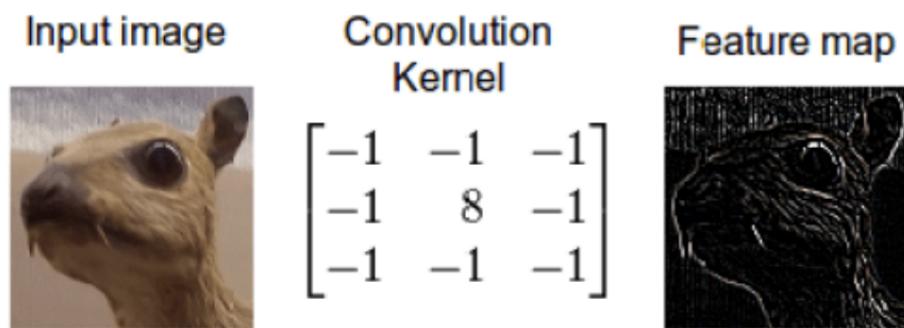


Figura 2.18: Filtro con *kernel* para resaltar contornos de objetos en la imagen. La matriz se usa para saber las ponderaciones de los píxeles en la suma para el nuevo valor de píxeles en la imagen. Así obtenemos un mapa de características que en este ejemplo nos entrega la información de los límites del animal de la imagen de entrada.

### 2.2.7. Modelo Faster RCNN Inception Resnet

El modelo *Faster RCNN Inception Resnet ??* es una red neuronal pre-entrenada para propósito general, es decir, se centra en el reconocimiento de 600 clases diferentes. Para esto el modelo hace uso de una red convolucional y una red de extracción de características que apunta a precisión sobre las detecciones que realiza para su clasificación, está centrada en la búsqueda exhaustiva de elementos diferenciables en una imagen. Esto otorga al modelo la capacidad de reconocer gran cantidad de objetos de la misma imagen como se puede apreciar en la figura ??.

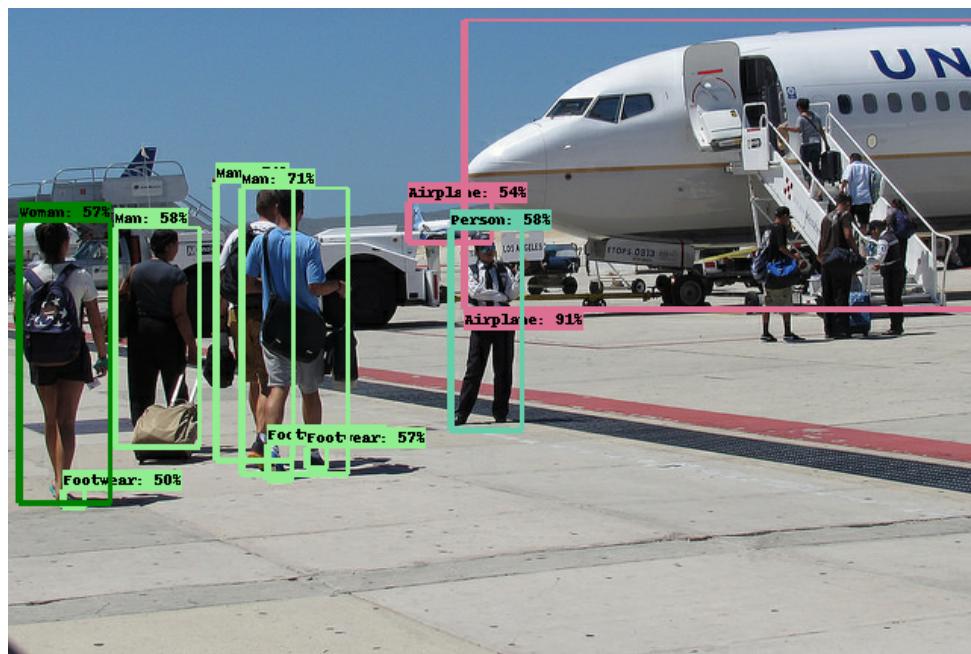


Figura 2.19: Ejemplo de detección en su contexto general de Faster RCNN Inception Resnet.

### 2.2.8. Modelo Centernet

El modelo *Centernet ??* de detección identifica los objetos como cuadros alineados con el eje en una imagen. La mayoría de los detectores de objetos que tienen éxito enumeran una lista casi exhaustiva de posibles ubicaciones de objetos y clasifican cada una de ellas. Esto es un desperdicio, ineficiente y requiere un post-procesamiento adicional. El modelo, adapta un enfoque diferente. Modela un objeto como un único punto, el punto central de su caja delimitadora. El modelo utiliza la estimación del punto clave para encontrar los puntos centrales y hace una regresión a todas las demás propiedades del objeto, como el tamaño, la ubicación 3D, la orientación e inclu-

so la postura (figura 2.20). Este enfoque basado en el punto central es diferenciable de extremo a extremo, más simple, más rápido y más preciso que los detectores correspondientes basados en el cuadro delimitador. CenterNet consigue la mejor relación velocidad-precisión en el conjunto de datos *MS COCO*.

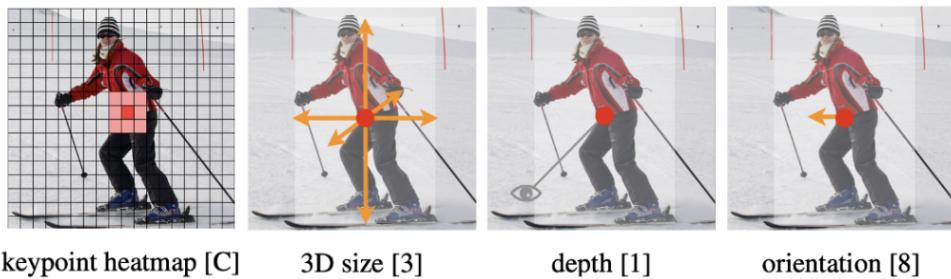


Figura 2.20: Primero se puede apreciar la búsqueda de puntos clave, luego la consideración de espacio tridimensional para delimitar las proporciones, luego la profundidad de los objetos, finalmente determina la orientación de movimiento de los objetos detectados.

Su arquitectura es una red convolucional que aplica extracción de esquinas en cascada (*Cascade corner pooling*) y extracción de centros (*Center pooling*) para generar dos mapas; de calor y características respectivamente. La detección de esquinas son utilizadas para la detección temprana de posibles delimitantes de objetos. Entonces los puntos correspondientes a centros detectados son usados para determinar el tamaño real de las áreas delimitantes (figura 2.21).

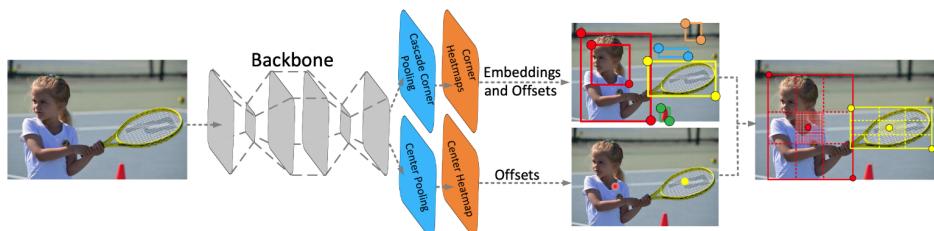


Figura 2.21: Arquitectura de red Centernet.

## 2.2.9. Tensorflow API

La API de detección de objetos de *TensorFlow* es un marco de trabajo de código abierto construido sobre *TensorFlow* con la finalidad de facilitar la construcción, el entrenamiento y el despliegue de modelos de detección de objetos creando una instancia de seguimiento en tiempo real

para las métricas de rendimiento de los modelos (figura 2.22) y permitiendo empaquetar resultados junto a sus configuraciones para etapas posteriores de producción.

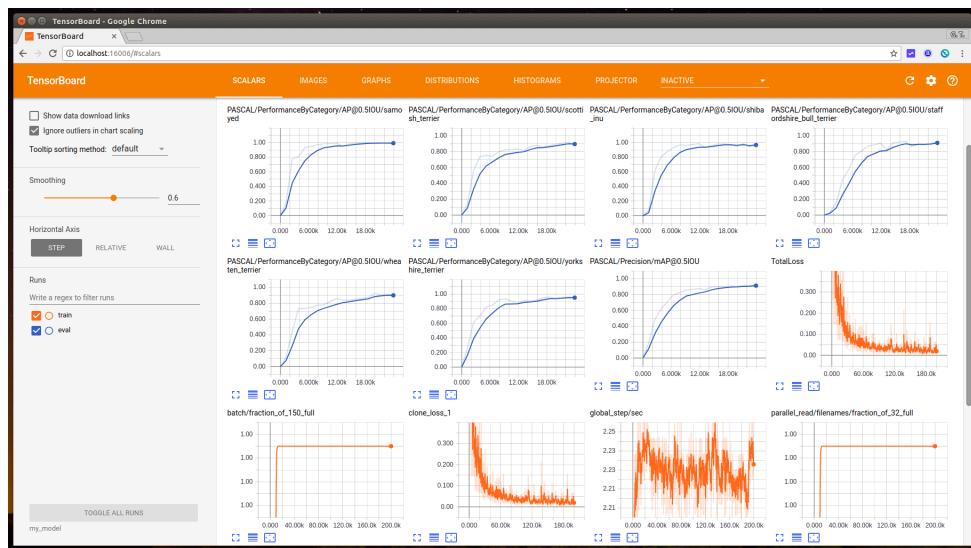


Figura 2.22: Interfaz de monitoreo de *Tensorflow API*.

## Capítulo 3

# Formulación del Proyecto

### 3.1. Objetivo General

Crear un modelo de aprendizaje de maquinas para la detección de y clasificación de parásitos en el análisis parasitológico seriado de deposiciones.

### 3.2. Objetivos Específicos

1. Recopilar y estructurar un conjunto de datos de imágenes de muestras para el entrenamiento y testeо del modelo de detección de parásitos.
2. Definir un modelo de detección de parásitos para la automatización de resultados del examen parasitológico seriado de deposiciones basado en visión por computadora.
3. Analizar y validar la calidad de las predicciones de la detección de parásitos en las muestras, a través de experimentación con el conjuntos de datos recopilado y con muestras obtenidas desde procedimientos reales del examen parasitológico seriado de deposiciones, utilizando imágenes microscópicas.

### 3.3. Metodología

1. Recopilar y estructurar un conjunto de datos de imágenes de muestras para el entrenamiento y testeo del modelo de detección de parásitos.
  - a) Formulación del proyecto.
  - b) Revisión de la literatura relacionada al proyecto de título.
  - c) Desarrollo del marco teórico.
  - d) Escritura del informe.
  - e) Revisión de conjunto de datos disponibles relacionados al examen parasitológico seriado de deposiciones.
  - f) Definir una estructura estándar para almacenar imágenes microscópicas de examen parasitológico seriado deposiciones.
  - g) Implementar un procedimiento de extracción y limpieza de imágenes de examen parasitológico seriado de deposiciones.
  - h) Realizar un procedimiento de análisis exploratorio de datos.
2. Definir un modelo de detección de parásitos para la automatización de resultados del examen parasitológico seriado de deposiciones basado en visión por computadora.
  - a) Adoptar una metodología (elección de experimento, métricas, configuración y *kfolds*) de proyectos de visión por computadora para realizar una comparación de la calidad de los modelos.
  - b) Proponer distintos modelos para la detección de parásitos en imágenes microscópicas.
  - c) Seleccionar el o los modelos con mejor calidad de resultado.
3. Analizar y validar la calidad de las predicciones de la detección de parásitos en las muestras, a través de experimentación con el conjuntos de datos recopilado y con muestras obtenidas desde procedimientos reales del examen parasitológico seriado de deposiciones, utilizando imágenes microscópicas.

- a) Implementar la metodología elegida o adaptada para el tratamiento de imágenes microscópicas.
- b) Implementar el procedimiento de predicción de parásitos detectada en las imágenes microscópicas.
- c) Implementar el reporte final de parásitos detectados en las muestras a través del modelo de visión por computadora.
- d) Recopilación de nuevas muestras, sus resultados, su validación y la retroalimentación del tecnólogo médico.

### 3.3.1. Planificación

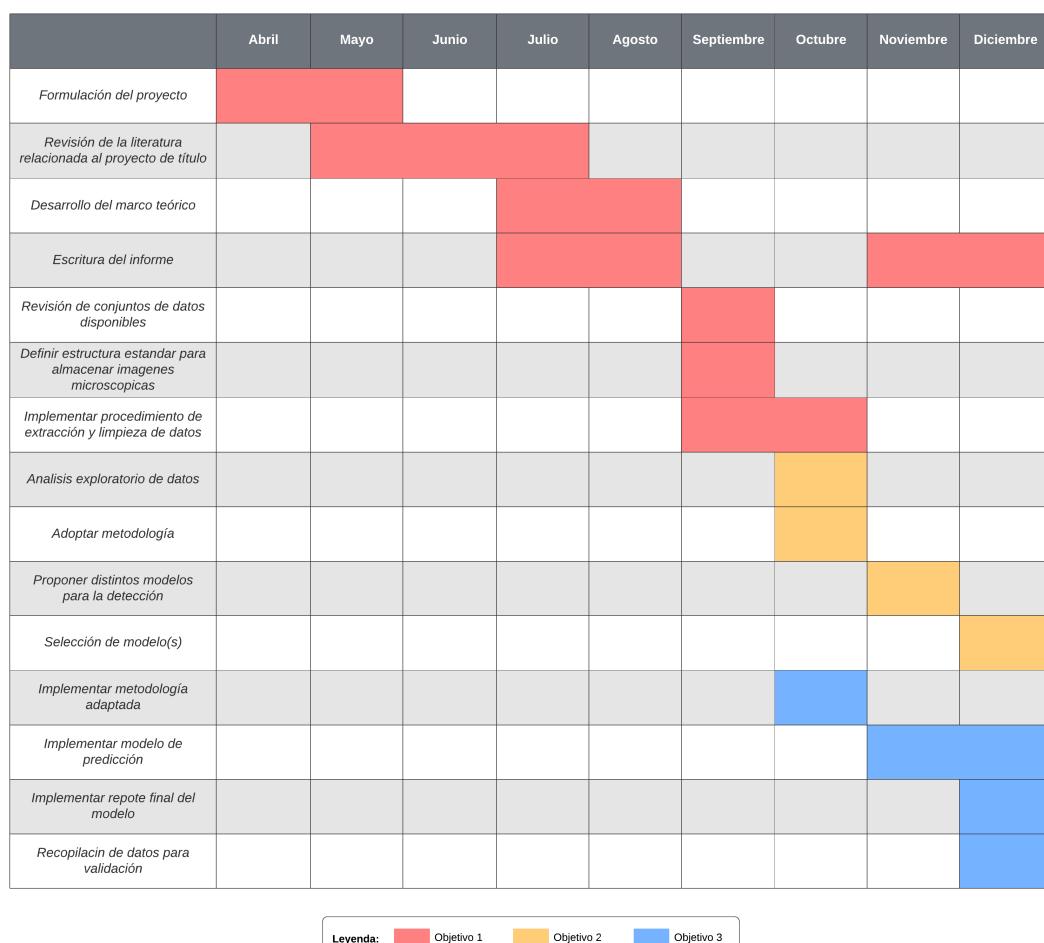


Figura 3.1: Carta Gantt para desarrollo del proyecto

## Capítulo 4

# Modelo de Identificación y Clasificación de Parásitos

En el presente capítulo, se describe el desarrollo de un modelo para indentificar y clasificar los parásitos contenidos en en conjunto de dato utilizado para el examen parasitológico seriado de deposiciones.

### 4.1. Conjunto de Datos

Para el desarrollo del proyecto de título se utiliza el conjunto de datos *Makerere Automated Lab Diagnostics Database (MALDD)* en cuyas característica encontramos 1217 imágenes clasificadas entre tres distintas clases de parásitos:

- Hookworm
- Hymenolepsis Nana
- Taenia

El conjunto se compone de imágenes microscópicas de muestras de deposiciones con etiquetas de parásitos presentes (anexo A) en la misma como se meuestra en la figura 4.1. La estructura de los datos se conforma en una carpeta raíz con las imágenes y su meta data en un archivo en formato *xml*, ambos de mismo nombre.

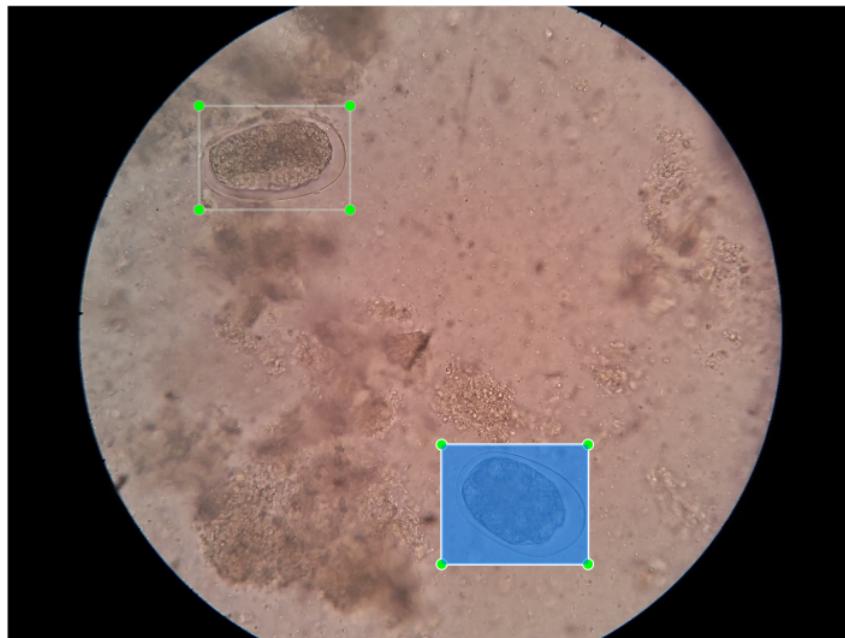


Figura 4.1: Ejemplo de imagen donde se aprecian dos *Hookworms*.

#### 4.1.1. Preprocesamiento y Limpieza

El proceso comprende la reestructuración de las carpetas separando el conjunto en entrenamiento (*train*) y testeо (*test*), a su vez se requiere de una modificación en la meta data de los archivos *xml* que incluya si la carpeta del archivo corresponde a uno de entrenamiento o testeо, agregar una etiqueta con el nombre del archivo de la imagen que le corresponde y su ruta absoluta al mismo mostrado en la figura 4.2. El conjunto de datos se separa con 60 % de las imágenes para entrenamiento del modelo y 40 % para testeо de manera aleatoria uniforme.

```

  <annotation>
    <filename>intestinalparasites-phone-1100.jpg</filename>
    <folder>test/train</folder>
    <source>
      <database>Makerere Automated Lab Diagnostics Database</database>
      <annotation>Makerere University</annotation>
    </source>
    <image>Mulago National Referral Hospital</image>
    <size>
      <width>2064</width>
      <height>1161</height>
    </size>
  </annotation>

```

Figura 4.2: Ejemplo de modificación de archivo *xml* de imagen.

La utilidad de conjunto de datos está especialmente orientada a la detección de las clases anteriormente descritas. En el contexto del examen parasitológico seriado de deposiciones, el conjunto de datos cubre una pequeña parte de los parásitos que se esperan reconocer siendo la lista de parásitos faltantes los siguientes:

- *Aedes aegypti*
- *Anisaki* spp
- *Anopheles*
- *Ascaris lumbricoide*
- *Balantidium coli*
- *Blastocystis hominis*
- *Chilomastix mesnili*
- *Cryptosporidium* spp
- *Cyclospora cayetanensis*
- *Cystoisospora belli*
- *Diphyllobothrium* spp
- *Endolimax nana*
- *Entamoeba coli*
- *Entamoeba gingivalis*
- *Entamoeba histolytica / dispar*
- *Enterobius vermicularis*
- *Fasciola hepática*
- *Giardia lamblia*

- Hymenolepis diminuta
- Iodamoeba butschlii
- Latrodectus mactans: viuda negra
- Loxosceles laeta: rincon
- Schistosoma mansoni
- Toxoplasma gondii
- Tribolium confusum
- Trichomonas tenax
- Trichomonas vaginalis
- Trichuris trichiuria
- Trypanosoma cruzi
- Pediculus humanus
- Plasmodium falciparum
- Phthirus pubis
- Pulex irritans
- Schistosoma haematobium

## 4.2. Modelo

Se revisaron las opciones de modelos; *Faster RCNN Inception Resnet v2 (FRRN v2)* y *Centernet* durante la fase de investigación con el objetivo de un modelo general que pueda especificarse en la tarea de reconocimiento de parásitos a través de imágenes microscópicas.

	filename	width	height		name	quantity	xmin	ymin	xmax	ymax
184	intestinalparasites-phone-0184.xml	1632	1224		[Hookworm, Hookworm]	2	[1043, 982]	[413, 714]	[1235, 1189]	[714, 976]
215	intestinalparasites-phone-0215.xml	1632	1224		[Hookworm, Taenia]	2	[373, 510]	[437, 336]	[632, 665]	[668, 492]
271	intestinalparasites-phone-0271.xml	1632	1224		[Hookworm, Hookworm]	2	[379, 492]	[406, 583]	[681, 772]	[632, 760]
393	intestinalparasites-phone-0393.xml	1632	1224	[Hookworm, Hookworm, Hookworm]		3	[458, 863, 406]	[248, 251, 811]	[696, 1104, 690]	[489, 537, 1022]
492	intestinalparasites-phone-0492.xml	1632	1224	[Hookworm, Hookworm]		2	[668, 836]	[251, 461]	[891, 1040]	[507, 720]
523	intestinalparasites-phone-0523.xml	1632	1224	[Hookworm, Taenia]		2	[367, 513]	[878, 766]	[623, 665]	[1110, 939]
558	intestinalparasites-phone-0558.xml	1632	1224	[Hookworm, Hookworm]		2	[942, 434]	[665, 0]	[1146, 629]	[948, 166]
594	intestinalparasites-phone-0594.xml	1632	1224	[Hookworm, Hookworm]		2	[187, 1085]	[492, 352]	[492, 1289]	[684, 650]
665	intestinalparasites-phone-0665.xml	1632	1224	[Hookworm, Hookworm]		2	[1134, 202]	[285, 227]	[1445, 516]	[495, 416]
747	intestinalparasites-phone-0747.xml	1632	1224	[Hookworm, Hookworm]		2	[370, 839]	[193, 848]	[662, 1122]	[394, 1079]
954	intestinalparasites-phone-0954.xml	1632	1224	[Hookworm, Hookworm]		2	[824, 614]	[175, 629]	[1046, 848]	[458, 878]
979	intestinalparasites-phone-0979.xml	1632	1224	[Hookworm, Hookworm]		2	[656, 860]	[263, 580]	[891, 1052]	[498, 836]
1063	intestinalparasites-phone-1063.xml	1632	1224	[Hookworm, Hookworm]		2	[507, 881]	[193, 547]	[723, 1073]	[431, 811]
1082	intestinalparasites-phone-1082.xml	1632	1224	[Hookworm, Hookworm]		2	[976, 327]	[802, 199]	[1232, 562]	[1037, 480]

Figura 4.3: Meta data del conjunto de elementos con más de una detección de parásitos.

Se descarta *FRIR v2* por su lento desempeño general en inferencias sobre conjuntos de datos de imágenes de contexto general.

Finalmente, se decide por *Centernet* por su característica forma de realizar una inferencia (figura 4.4), *Centernet* busca identificar ejes alineados a un rectángulo correspondiente a un objeto dentro de una imagen. Esta diferencia nace de su forma de buscar regiones que contengan objetos e identificándolos como un punto, dando así con un modelo de muy rápida inferencia y gran nivel de precisión en casos particulares.

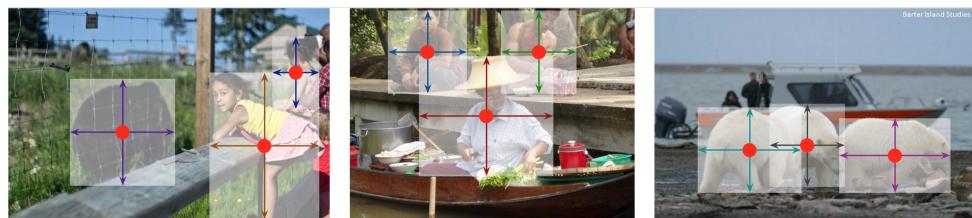


Figura 4.4: Ejemplo de detección de objetos con *Centernet*.

# Capítulo 5

## Entrenamiento

El entrenamiento del modelo *Centernet* con el conjunto de datos de *MALDD* se realiza de manera local con la API de *TensorFlow* para un seguimiento en tiempo real de las mediciones de pérdida del modelo durante el entrenamiento. En las configuraciones del modelo se aceptan como parámetros; máxima cantidad de predicciones de 100 para no dejar fuera elementos de las muestras, mínimo grado de solapamiento de cajas de reconocimiento de 0,7 dado que las muestras son tomadas en un porta objeto que aplana la muestra para evitar solapamiento de antemano, también se usa aumento de datos con giros horizontales y verticales de las imágenes para cubrir el pequeño tamaño del conjunto de datos. El modelo es entrenado con el optimizador Adam y un ratio de aprendizaje de 0,001 constante durante todo el entrenamiento y 40000 iteraciones por generación, especificando el modelo en detección sin *kfolds*, validación cruzada y sin optimización de hiperparámetros.

Se ocupó para la etapa de entrenamiento un computador con las siguientes características:

- Procesador: AMD 2600X (6 núcleos)
- Tarjeta Gráfica: NVidia RTX 2060 (6GB)
- Cantidad de Memoria RAM: 16 GB

Finalmente, el entrenamiento se lleva a cabo durante 10 horas y 31 minutos.

# Capítulo 6

## Resultados

En la gráfica de pérdida final de la figura 6.1 se muestra el seguimiento de pérdida general en función del tiempo en el eje horizontal y la precisión en el eje vertical. En el análisis de la curva de pérdida general se puede apreciar el error inicial cercano al 30 % y disminuye rápidamente hasta el 10 % para la iteración 15000, manteniendo una disminución casi constante a partir de ese punto. La gráfica atenuada detrás es la curva de pérdida real sin suavizado se puede apreciar entre las iteraciones 15000 y 20000 un gran nivel de error que pueden haber sido causadas por imágenes mal etiquetadas, finalmente ese ruido causado en el entrenamiento no repercute en desajuste del modelo y el error en entrenamiento se normalizó en un valor cercano al 30 % (70 % de precisión).

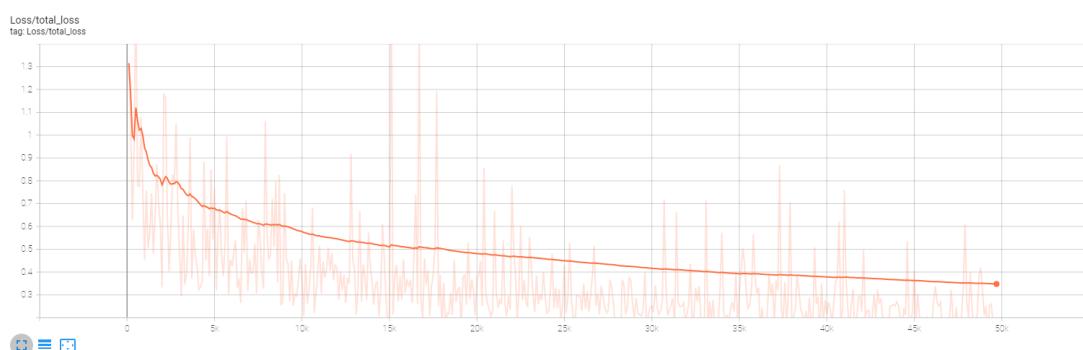


Figura 6.1: Gráfica de pérdida general del modelo.

Así mismo en la figura 6.2 de mismos ejes y separando el error en precisión del modelo en identificación de centros de objetos se aprecia una curva similar a la anteriormente descrita. Los valores de error disminuyen con el mismo comportamiento general y sin recibir mayor aumento de error por ruido en el conjunto de datos. En resumen, se termina en un error del 2,5 % (98,5 %

de precisión).

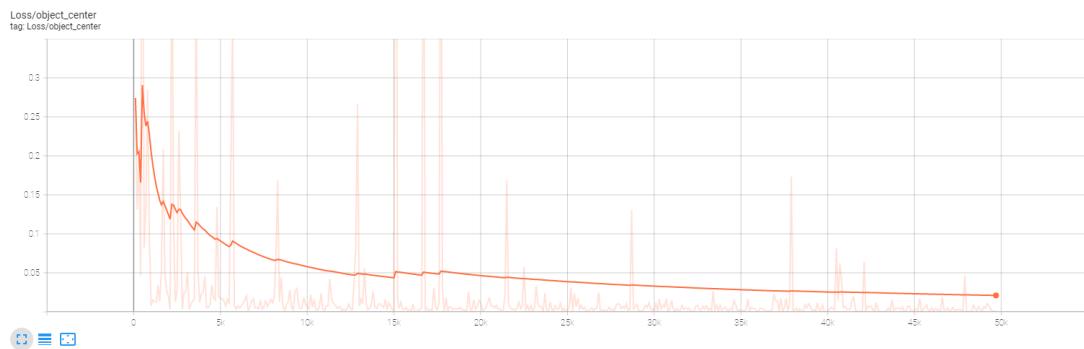


Figura 6.2: Gráfica de pérdida en detección de centros.

Por último, tenemos la figura 6.3 con los resultados de error en lo que comprende a la delimitación del área de detección de cada error. Se puede apreciar que a diferencia del error de centros, la gráfica muestra una disminución de pérdida más lineal, lo que da a entender que la pérdida en la delimitación de las áreas donde se localizan los objetos es un error que disminuye en razón a una constante.

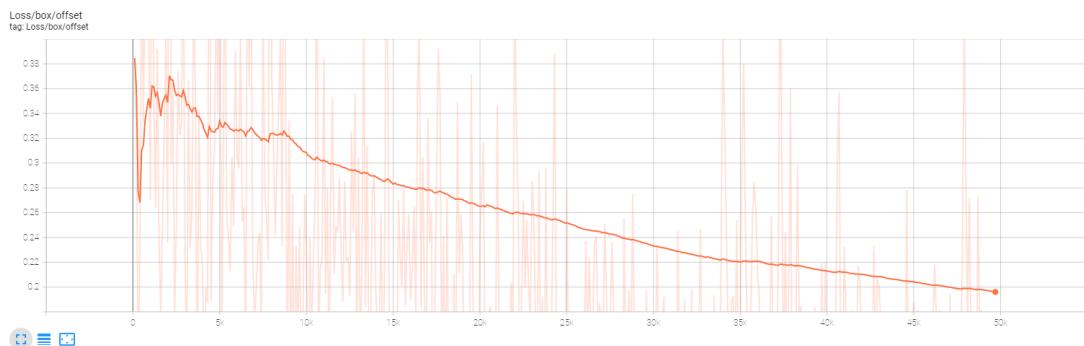
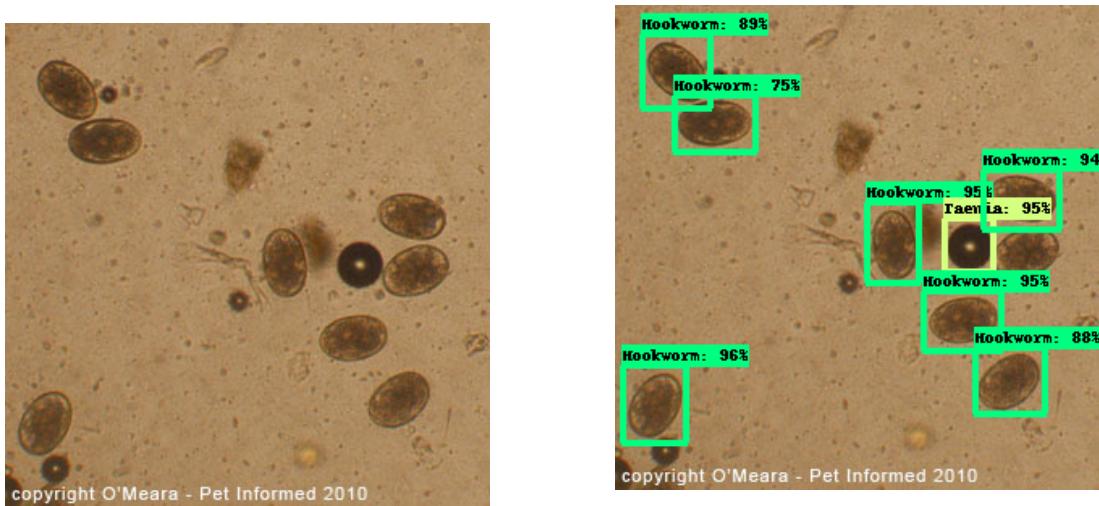


Figura 6.3: Gráfica de pérdida en alineamiento delimitador.

Los resultados del modelo concluyen en un tiempo de espera por inferencia de 7,84 segundos con clasificaciones mostradas a continuación en 6.4.



(a) Imagen original (300x300).

(b) Imagen con etiquetas de reconocimiento, clases reconocidas; Hookworm y Taenia.

Figura 6.4: Detección de multiples parásitos contenidos en una sola muestra.

Se destaca la especialidad del modelo en la detección de presencia parasitaria en casos de material indeseado obstruyendo la muestra como se puede apreciar en la figura 6.5.

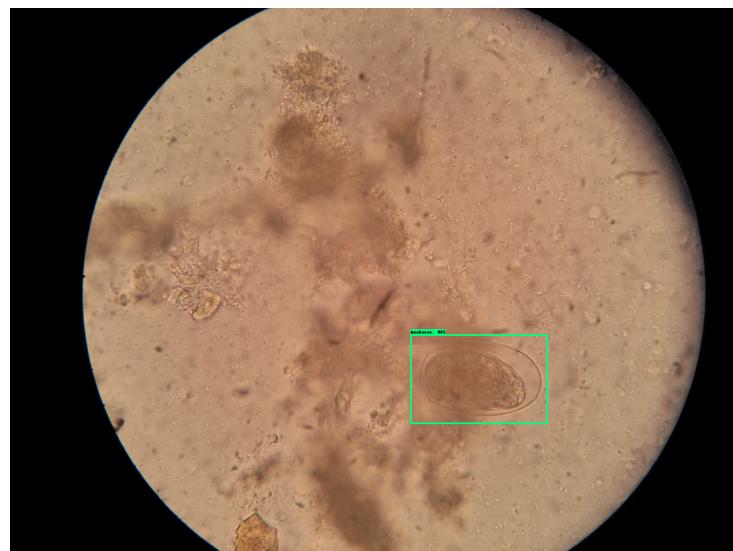


Figura 6.5: Muestra de parásito Hookworm obstruido por material indeseado.

Finalmente, se aprecia que los resultados obtenidos en las detecciones es altamente sensible a la resolución de la imagen, dejando un rango de resoluciones de  $200 \times 200$  hasta  $1000 \times 1000$  píxeles aproximadamente. Cualquier imagen fuera de ese rango causa errores en interpretaciones que pueden variar desde objetos no reconocidos, hasta objetos clasificados de manera errónea.

# Capítulo 7

## Conclusión

Durante este informe se vieron las tecnologías relacionadas a la visión por computadora que permiten hacer uso de una inferencia en base al entrenamiento de modelos de aprendizaje de máquinas para la detección y clasificación de presencia parasitológica en el examen seriado de deposiciones. Se vió el estado de los procedimientos para este examen y reconocimos el punto donde el uso de estas tecnologías puede jugar un gran rol en la mejora de resultados y la velocidad de análisis.

En términos de los objetivos planteados al inicio del proyecto de título, se recopiló un conjunto de datos orientado a imágenes microscópicas de muestras de deposiciones con y sin presencia parasitaria con la finalidad del entrenamiento del modelo. El conjunto de datos cumplió con las características de poseer tres clases de parásitos intestinales, así como también imágenes sin presencia alguna, haciendo una totalidad de 1217 imágenes de muestras.

Se entrenó un modelo llamado *Centernet* con la finalidad de hacerlo específico al reconocimiento y clasificación de especies de parásitos presentes en el sistema digestivo humano para la ayuda en el área de la tecnología médica, que es en donde se realizan los estudios médicos y se emplea el uso de las tecnologías existentes para facilitar y respaldar el diagnóstico médico.

Al analizar la calidad de las predicciones en la detección de parásitos se puede apreciar la gran precisión que logra el modelo dentro de las ventanas de tiempo y tamaños de muestras con las que se contó en el momento, esto muestra una gran ventana al desarrollo de la visión por computadora en el área de la medicina, más específico, al área del análisis de imágenes microscópicas.

# Referencias

- Backlund, G. (2017). *An overview of the data augmentation algorithm*. Department of Statistics, University of Florida. Descargado March 2017, de <https://people.clas.ufl.edu/grantback21/files/An-Overview-of-the-Data-Augmentation-Algorithm.pdf>
- Byjus. (2012a). *Data sets meaning*. Descargado 2012, de <https://byjus.com/maths/data-sets/> (Datasets)
- Byjus. (2012b). *What is data management?* Descargado 2012, de <https://byjus.com/maths/data-management/> (Data Management)
- CDC. (2016). *Diagnóstico de enfermedades parásitarias*. Descargado 18 Octover 2016, de [https://www.cdc.gov/parasites/es/references\\_resources/diagnosis.html](https://www.cdc.gov/parasites/es/references_resources/diagnosis.html) (¿Cómo diagnostican las enfermedades parásitarias?)
- Dey, T. K. (2017). *Undersampling and oversampling in sample based shape modeling*. Ohio State University Columbus. Descargado 2017, de [https://graphics.stanford.edu/courses/cs468-03-fall/Papers/deygiessen\\_undersampling.pdf](https://graphics.stanford.edu/courses/cs468-03-fall/Papers/deygiessen_undersampling.pdf)
- FE., C. (2002). *History of human parasitology*. Portland State University. Descargado 16 January 2003, de <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC126866/>
- Fuchs, K. (2017). *Machine learning: Classification models*. Descargado 28 March 2017, de <https://medium.com/fuzz/machine-learning-classification-models-3040f71e2529> (Classification Models)
- Geert, R. (2020). *Logistic regression*. Descargado 2020, de <https://www.spss-tutorials.com/logistic-regression/#simple-logistic-regression-equation> (Logistic Regression)
- Ho, T. K. (2016). *Random decision forests*. AT&T Bell Laboratories. Descargado 17 April

- 2016, de <https://web.archive.org/web/20160417030218/http://ect.bell-labs.com/who/tkh/publications/papers/odt.pdf>
- Huapaya, P. (2002). *Utilidad de exámenes parasitológicos y serológicos como métodos de diagnóstico de estreñiloidiosis humana.* Instituto de Medicina Tropical, Universidad Nacional Mayor de San Marcos. Descargado 2002, de [https://sisbib.unmsm.edu.pe/brevistas/anales/v63\\_n1/pdf/utilidad\\_examenes\\_parasitologico.pdf](https://sisbib.unmsm.edu.pe/brevistas/anales/v63_n1/pdf/utilidad_examenes_parasitologico.pdf)
- IBM. (2020). *What is computer vision?* Descargado 10 August 2020, de <https://www.ibm.com/topics/computer-vision> (Computer definition by IBM)
- ISP. (2021). *Vigilancia parasitología.* Ministerio de Salud. Descargado 2021, de <https://www.ispch.cl/biomedico/vigilancia-de-laboratorio/ambitos-de-vigilancia/vigilancia-parasitologia/>
- Jercic, M. I. (2017). *Recomendaciones para la realización del examen parasitológico seriado de deposiciones.* Instituto de Salud Pública. Descargado 2017, de <https://www.ispch.cl/sites/default/files/RECOMENDACIONES%20PARA%20LA%20REALIZACION%20DEL%20EXAMEN%20PARASITOLOGICO%20SERIADO%20DE%20DEPOSICIONES.pdf>
- K, A. (2020). *Understanding data augmentation | what is data augmentation & how it works?* Descargado 5 August 2020, de <https://www.mygreatlearning.com/blog/understanding-data-augmentation/#numericaldataaugmentation> (Data Augmentation)
- Khare, K. (2011). *A spectral analytic comparison of trace-class data augmentation algorithms and their sandwich variants.* Descargado Octover 2011, de <https://www.jstor.org/stable/41713589> (Spectral Analytic and Comparations)
- Labelbox. (2020). *Image segmentation 101.* Descargado 2020, de <https://labelbox.com/image-segmentation-overview> (Segmentation)
- Ludwig, J. (2020). *Image convolution.* Portland State University. Descargado 2020, de [http://web.pdx.edu/~jduh/courses/Archive/geog481w07/Students/Ludwig\\_ImageConvolution.pdf](http://web.pdx.edu/~jduh/courses/Archive/geog481w07/Students/Ludwig_ImageConvolution.pdf)
- Nayak, R. (2020). *Data augmentation in natural language processing for text classification.* Descar-

- gado 18 May 2020, de <https://medium.com/swlh/data-augmentation-in-natural-language-processing-756e942b42f3> (Improve the performance of your model by generating data on the go)
- Prince, S. (2012). *Computer vision: models, learning and inference*. Cambridge University Press. Descargado 10 August 2020, de <http://web4.cs.ucl.ac.uk/staff/s.prince/book/book.pdf>
- Rennie, J. D. (2003). *Tackling the poor assumptions of naive bayes text classifiers*. Artificial Intelligence Laboratory; Massachusetts Institute of Technology; Cambridge, MA 02139. Descargado 2003, de <https://people.csail.mit.edu/jrennie/papers/icml03-nb.pdf>
- Rokach, L. (2007). *Decision trees*. Tel-Aviv University. Descargado 2007, de <https://www.ise.bgu.ac.il/faculty/liorr/hbchap9.pdf>
- Tatman, R. (2020a). *Handling missing values*. Descargado 2020, de <https://www.kaggle.com/alexisbcook/handling-missing-values> (Missing Data)
- Tatman, R. (2020b). *Inconsistent data entry*. Descargado 2020, de <https://www.kaggle.com/alexisbcook/inconsistent-data-entry> (Inconsistent Data)
- Tatman, R. (2020c). *Scaling and normalization*. Descargado 2020, de <https://www.kaggle.com/alexisbcook/scaling-and-normalization> (Scaling and Normalize Data)
- Tiño, P. (2016). *Probabilistic modelling in machine learning*. Birmingham University, UK. Descargado 10 August 2020, de [https://s4d.sciencesconf.org/data/program/prob\\_mod\\_ml\\_caen\\_1.pdf](https://s4d.sciencesconf.org/data/program/prob_mod_ml_caen_1.pdf)
- Wang, J. (2017). *The effectiveness of data augmentation in image classification using deep learning*. Standford University and Google. Descargado 2017, de <http://cs231n.stanford.edu/reports/2017/pdfs/300.pdf>
- Wang, Y. (2020). *Geometric transformations: Warping, registration, morphing*. Polytechnic University, Brooklyn. Descargado 2020, de [https://eeweb.engineering.nyu.edu/~yao/EL5123/lecture12\\_ImageWarping.pdf](https://eeweb.engineering.nyu.edu/~yao/EL5123/lecture12_ImageWarping.pdf)
- Weitz, J. C. (2013). *Diagnóstico y tratamiento de las enfermedades digestivas*. Sociedad Chilena de Gastroenterología. Descargado Octover 2013, de <http://www.sociedadgastro.org>

- .cl/gastroweb/documentos/Diagnostico-y-tratamiento-ED.pdf  
Wood, T. (2020). *What is the sigmoid function?* Descargado 27 September 2020, de  
[https://deepai.org/machine-learning-glossary-and-terms/  
sigmoid-function \(Sigmoid Function\)](https://deepai.org/machine-learning-glossary-and-terms/sigmoid-function)

## Anexos A

### Archivo XML

```
1 <annotation>
2   <source>
3     <database>Makerere Automated Lab Diagnostics Database</←
4       database>
5     <annotation>Makerere University</annotation>
6     <image>Mulago National Referral Hospital</image>
7   </source>
8   <size>
9     <width>1632</width>
10    <height>1224</height>
11  </size>
12  <object>
13    <name>Hookworm</name>
14    <pose>Unspecified</pose>
15    <truncated>0</truncated>
16    <occluded>0</occluded>
17    <bndbox>
18      <xmin>903</xmin>
        <ymin>735</ymin>
```

```
19      <xmax>1198</xmax>
20      <ymax>921</ymax>
21      </bndbox>
22  </object>
23 </annotation>
```