UNIVERSIDAD NACIONAL DE CÓRDOBA

Detección de escamas laterales de la cabeza de tortugas verdes *(Chelonia mydas)* para su posterior recorte y uso en foto-identificación

Tesis

para obtener el título de

MAGÍSTER EN ESTADÍSTICA APLICADA

BIÓL. CANDELA BUTELER

Directora: ANA GEORGINA FLESIA

Deep Learning Works, I Don't Know Why

Agradecimientos

Quisiera agradecer a mi directora, la Dra. Ana Georgina Flesia por su apoyo y acompañamiento en este etapa.

A la directora científica de la ONG Karumbé, la Dra. Gabriela Vélez-Rubio por su apoyo a lo largo de toda la investigación de foto-identificación. A Alejandro Fallabrino, director de Karumbé que siempre ha sido muy generoso conmigo y con las oportunidades. A todos los integrantes de Karumbé que me enseñaron tanto.

A los integrates del tribunal evaluador de tesis, la Dra. Valeria S. Rulloni, el Dr. Angel M. Segura Castillo y el Dr. Julio Di Rienzo por sus contribuciones para el entendimiento y mejoras del documento. A cada profesor de la Maestría por los conocimientos trasmitidos y la buena onda en las clases.

A Gustavo, un genio total que siempre estuvo para ayudarme. A mis compañerxs, que hicieron más amenas las largas horas de cursado.

A Francisco Funes, por la ayuda con el código y la paciencia por explicarme algo cada vez que lo necesité. Sin él, esta tesis hubiese sido imposible.

Al Dr. Diego Sebastián Pérez por su generosidad y ayuda en el software que usamos para etiquetar las imágenes.

Sobre todo, a mi mamá (La Gringa), papá (Pepe), hermano (Juan) por el acompañamiento en cada paso que doy y a mi compañero, Juan Cruz mejor conocido como Lima por impulsarme cada vez que necesitaba ese empujón y estar en cada momento.

Resumen

La identificación de individuos es un requisito previo para estudiar poblaciones y estimar, por ejemplo, el tiempo que las tortugas marinas pasan en distintas zonas a lo largo de vida. Para ello, suele ser común marcarlos a través de marcas artificiales, las cuales suelen ser invasivas, o puede reconocerse patrones fenotípicos propios del individuo lo que implica técnicas menos invasivas. Estos procedimientos se conocen como métodos captura-marca-recaptura. Sin embargo, estos métodos en ocasiones no son confiables en la detección o es complicado el seguimiento de individuos a lo largo de los años. Las tortugas marinas pueden identificarse por el patrón de escamas del cuerpo, en especial la de los costados de la cabeza se ha usado ampliamente para distinguirlas a través de fotos, método que se conoce como foto-identificación. Este estudio propone un enfoque combinado que identifica automáticamente las escamas de la cabeza de las tortugas marinas, *Chelonia mydas*, para que luego puedan reconocerse las coincidencias entre individuos mediante el uso de foto-identificación. Mostramos aquí que la combinación de algoritmos de aprendizaje automatizado permiten detectar las escamas y formular una función de decisión que indique la precisión de esta detección. Este procedimiento reduce los tiempos de detección y recorte del área de interés y puede ser muy ventajoso cuando se tienen una base de datos muy grande. Este enfoque también podría ser aplicable a una amplia gama de especies y contribuir significativamente a la conservación de especies en peligro de extinción al proporcionar la detección automática del patrón de interés a través de técnicas no invasivas.

Índice general

Tesis de Maestría en Estadística Aplicada

7 de noviembre de 2022

1 Introducción

El aprendizaje automático, o como comúnmente se utiliza en inglés, *Machine learning*, es un campo dentro de la inteligencia artificial dedicado a comprender y crear métodos que aprovechen datos para mejorar el rendimiento en un conjunto de tareas. Los algoritmos de aprendizaje automático construyen un modelo basado en datos de muestra, conocidos como datos de entrenamiento, para hacer predicciones o tomar decisiones. Los algoritmos de aprendizaje automático actualmente se utilizan en una amplia variedad de aplicaciones, como en medicina, reconocimiento de voz, en desarrollo de autos autónomos e incluso en biología. En esta sección introduciremos las causas que inspiraron esta investigación y qué problemática pretendemos abordar.

1.1. Motivación

Reconocer animales silvestres es de gran utilidad a la hora de realizar estudios poblacionales. Para este reconocimiento es debido marcarlos (con marcas artificiales) o poder reconocer patrones fenotípicos propios del individuo (un fenotipo es cualquier característica o rasgo observable de un organismo). Estos procedimientos se conocen como métodos captura-marca-recaptura (CMR) y son los más precisos para modelar tasas de supervivencia, de permanencia y el tamaño poblacional. Los modelos basados en CMR dan cuenta de

las probabilidades de detección individual y, por lo tanto, brindan una inferencia confiable para las tasas vitales [40]. Dichos estudios de población son esenciales para planificar los esfuerzos de conservación de varios taxones y las tortugas marinas no son una excepción, ya que la estimación de las tasas de supervivencia han brindado información valiosa para acciones de conocimiento y protección de sus poblaciones. Por ejemplo, un estudio [19] realizado en tortuga verde realizado en Brasil demostró a través de CMR que el tiempo medio de residencia en un archipiélago fue de 2,4 años (con una residencia a largo plazo de hasta 11,2 años) y que la abundancia osciló entre 420 y 1148 individuos. Otro trabajo [39] en donde se utilizó información proveniente de CMR realizado con tortugas adultas y anidantes de tortugas carey logró calcular la probabilidad de supervivencia de 0,935 $\pm$ 0,01 (estimación $\pm$ error estándar).

Comúnmente, los individuos de tortugas marinas capturados se marcan artificialmente [3, 5, 50] a través de marcas de metal o plástico que se colocan sobre las aletas [46, 48, 15]. Teniendo en cuenta los inconvenientes éticos, ya que son procedimientos que pueden estresar a los animales y afectar el bienestar animal [54], considerando cuestiones monetarias de la aplicación de marcas (las marcas artificiales son costosas en términos de material y tiempo de aplicación) y que pueden tener un alto porcentaje de pérdida [46], se propusieron técnicas menos invasivas, por ejemplo, a través del uso de imágenes.

Este cambio se vio favorecido por el desarrollo tecnológico del procesamiento de imágenes y generó un incremento en el uso de técnicas de foto-identificación (PID por su referencia en inglés a *Photo identification*) [25]. Las técnicas de PID aprovechan los patrones físicos naturales para la identificación de individuos y, así encontrar sucesivos encuentros de un mismo animal a través de la comparación de imágenes. En tortugas marinas, por ejemplo, el patrón de las escamas de los laterales de la cabeza es único para cada individuos por lo que permite la identificación de cada tortuga a lo largo de los años [15].

Los recientes avances en fotografía digital y algoritmos de reconocimiento de patrones [8] han permitido la creación de análisis rápidos y eficientes de grandes bases de datos

fotográficas [28]. El uso de la tecnología ha favorecido mucho a este tipo de técnicas, de esta manera PID asistido por software reduce el tiempo de procesamiento de datos en comparación con las técnicas de PID visual [24, 4], en dónde el observador debe revisar una a una todas las fotos y encontrar la coincidencia entre los individuos. La técnica de PID para tortugas marinas puede realizarse a través de fotografías de los laterales de la cabeza (Figura1.1) que luego son analizadas y comparadas en algún software como por ejemplo WILD ID ([8], Buteler *et. al.* En revisión). Los resultados provistos por éste método son muy eficaces y hasta más precisos que los obtenidos por marcas colocadas en las aletas (Buteler *et. al.* En revisión).

Estos estudios han sido realizados en otras especies distantes a las tortugas como jirafas, sapos, tritones y focas [8, 23, 42] y al mismo tiempo han permitido la identificación exitosa de varias especies de testudines, entre ellos: tortugas laúd *(Dermochelys coriacea)*, tortugas carey *(Eretmochelys imbricata)* y tortugas verdes *(Chelonia mydas)* [2, 13, 15, 28, 46].

Los procesos de PID requieren generalmente de tres pasos luego de que las fotografías de los animales han sido obtenidas y estos pasos a veces pueden requerir mucho tiempo de procesamiento por parte del usuario o investigador. El primero es la selección manual o recorte del área de interés del animal. Luego, el siguiente paso consiste en un algoritmo de comparación entre la imagen focal (imagen que se compara) y la librería con las demás imágenes en dónde usualmente se asigna una puntuación que indica la probabilidad de coincidencia. El paso final es la comparación visual por parte de los observadores de los pares de candidatos para confirmar las coincidencias positivas [12] y determinar que es el mismo individuo. El primer paso, el recorte y edición de cada fotografía, puede llevar mucho tiempo si la base de datos que se desea emplear es grande. Buehler et al., (2019) [12] desarrollaron una manera automática de encontrar jirafas en fotografías en la sabana Africana y recortar automáticamente cada foto a través de algoritmos que

Figura 1.1: Individuo de verde Chelonia mydas capturada en Uruguay en 2008 en donde se puede ver el patrón de escamas de los laterales de la cabeza que se utilizaron en este estudio para la detección automática.

son capaces de encontrar características propias que identifican estos ejemplares. Estos métodos automáticos de detección se basan en algoritmos de aprendizaje automatizado, en especial de, aprendizaje profundo que tienen como objetivo que las computadoras aprovechen datos para automatizar procesos. Cuando el conjunto de datos que utilizamos para el entrenamiento se conoce de qué categoría pertenece, es decir, tenemos la etiqueta del dato, se conocen como aprendizaje automático supervisado. El entendimiento de estos conceptos se tratará con mayor profundidad en el capítulo 2.

De esta manera en este trabajo proponemos una técnica que acorta los tiempos de selección manual y recorte del área de interés a través de la detección automática de las escamas de los laterales de la cabeza, aplicado a las tortugas verdes y su cálculo del error

asociado.

Este estudio se realizó con tortugas marinas de las costas uruguayas, las cuales son principalmente juveniles de la especie *Chelonia mydas*, llamada comúnmente tortuga verde [52] y se enmarca en un proyecto a largo plazo desarrollado por la ONG Karumbé sobre la caracterización de la agregación de tortugas verdes juveniles de las aguas uruguayas [43]. La identificación individual de las tortugas marinas, como ya hemos mencionado, es esencial para comprender la dinámica de la población y para planificar esfuerzos de conservación para las especies. Si bien la tortuga verde se encuentra categorizada En Peligro a nivel global (IUCN, 2004), actualmente la población del Atlántico Sur se encuentra categorizada como de Preocupación menor [11]. Sin embargo, hay estudios que demuestran que los números de varamientos para la especie han aumentado [14] los cuales tendrían impactos negativos en el estado de conservación de las tortugas verdes en la región y podrían afectar la dinámica de la población en los próximos 10 a 20 años.

1.1.1. Objetivos

El objetivo principal del presente trabajo es contribuir al desarrollo de técnicas no invasivas de identificación en juveniles de *Chelonia mydas* que llegan a las costas del Atlántico Sur Occidental. Los objetivos particulares son:

- desarrollar una técnica de identificado automático que genere un recuadro en el área de interés a través de redes neuronales para un problema de regresión;
- generar una función a posteriori que permita definir cuándo una detección predicha por la red neuronal puede definirse como aceptable o no;
- calcular el error empírico de detección de escamas generado a través de la red neuronal en el recorte de un nuevo conjunto de imágenes de tortugas.

1.2. Esquema de trabajo

En este estudio se propone un enfoque que combina algoritmos de aprendizaje profundo para identificar automáticamente las escamas de la cabeza de las tortugas verdes a través de un problema de regresión y luego, a través de técnicas de aprendizaje supervisado, formular una función de decisión que indique si la detección puede considerarse como aceptable o no para luego ser utilizada como foto-identificación.

En el capitulo 2 se introducen ideas generales de aprendizaje automático, en especial sobre la red neuronal convolucional ResNet que hemos utilizado para el problema de detección de escamas.

En el capítulo 3 se introduce información sobre la metodología utilizada. Primero se detallará cómo generamos la base de datos y el entrenamiento de la ResNet. Luego, se definen los parámetros de evaluación de la detección y se introduce un árbol de decisión que permite definir una función para clasificar una detección como aceptable o no. Por último, se calculan los errores asociados a la detección a través de la red neuronal en un nuevo conjunto de datos.

En el capítulo 4 esquematizamos los resultados obtenidos sobre nuestro conjunto de datos en función de los objetivos. En el capítulo 5 establecemos las conclusiones y por último tenemos un apartado de perspectivas de esta investigación.

2 Fundamentos teóricos

Aquí hablaremos de los fundamentos teóricos generales de inteligencia artificial y aprendizaje automatizado en los que se fundamenta la detección de objetos y cómo hemos hecho uso de ellos para resolver nuestro problema de detección de escamas aplicado a tortugas marinas.

Primero, es importante mencionar que la detección de objetos es un área cada vez más importante dentro de la ciencia de la inteligencia artificial que está cobrando cada vez más importancia en los problemas biológicos y ecológicos. El crecimiento de la tecnología ha favorecido el desarrollo de nuevos algoritmos para facilitar el trabajo de biólogos y ecólogos a la hora de procesar un gran volumen de datos.

2.1. Inteligencia artificial

El término *Inteligencia artificial* nació en la década de 1950. En aquel momento se formuló la idea de que las mentes humanas y las computadoras digitales modernas eran parecidas, en el sentido de que ambos procesaban información simbólica como entrada, la manipulan de acuerdo con un conjunto de reglas formales y, al hacerlo, podían resolver problemas, formular juicios y tomar decisiones [27]. En ese momento los investigadores de inteligencia artificial se propusieron identificar los procesos formales que constituían el

comportamiento humano inteligente en el diagnóstico médico, el ajedrez, las matemáticas, el procesamiento del lenguaje, etc., con la esperanza de reproducir ese comportamiento mediante medios automatizados [27], hasta que apareció el aprendizaje automatizado.

2.2. Aprendizaje automatizado

El aprendizaje automatizado es un sub-campo de la inteligencia artificial que tiene como objetivo que las computadoras aprovechen datos y puedan realizar tareas sin ser programadas directamente cada vez que éstas se requieran [7] enfatizando en la predicción y optimización. Podríamos decir que las computadoras "aprenden" a través de la "experiencia", pero esto en realidad significa tomar información de los datos y ajustarse a ellos; por lo tanto, a veces parece no existir un límite claro entre el aprendizaje automático y los enfoques estadísticos. Sin embargo, hay algunos trabajos que persiguen el propósito de analizar las relaciones entre metodologías de aprendizaje automatizado y modelos estadísticos tradicionales y concluyen en que los conceptos a los que se apela son los mismos [44]. Aún así, a pesar de las similitudes metodológicas, el aprendizaje automático es filosófica y prácticamente distinguible: el aprendizaje automático enfatiza la precisión predictiva sobre la inferencia basada en hipótesis, generalmente enfocándose en un conjunto grande de datos y de alta dimensión [9]. Independientemente de la distinción precisa entre enfoques, en la práctica, el aprendizaje automático ofrece herramientas importantes para la simplificación de tareas o automatización de procesos [7, 35, 16, 45].

Los métodos de aprendizaje automatizado pueden dividirse en técnicas supervisadas (los algoritmos trabajan con datos etiquetados, es decir que se conoce su variable a predecir, intentado encontrar una función que, dadas las variables de entrada, le asigne la etiqueta de salida adecuada), no supervisadas (cuando no se dispone de datos etiquetados para el entrenamiento, por lo que sólo podemos describir la estructura de los datos, para intentar encontrar algún tipo de organización que simplifique el análisis) o por refuerzo

(este tipo aprendizaje se basa en mejorar la respuesta del modelo usando un proceso de retroalimentación).

Entre los métodos de aprendizaje automático supervisado más conocidos se encuentran los árboles de clasificación [10]. Un árbol de clasificación es una estructura de árbol similar a un diagrama de flujo en donde cada nodo denota una prueba en un atributo, cada rama representa un resultado de la prueba y los nodos finales representan clases o distribución de clases [32]. Otro ejemplo de método de aprendizaje supervisado son las redes neuronales dentro del aprendizaje profundo.

2.2.1. Aprendizaje profundo

El aprendizaje profundo utiliza múltiples capas de procesamiento no lineal para descubrir patrones y estructuras en conjuntos de datos muy grandes [44] e intenta emular el modo en que el cerebro humano aprende. Dentro de los métodos más conocidos se encuentran las redes neuronales artificiales, inspiradas en las células del sistema nervioso conectadas entre sí por extensiones nerviosas que transmiten señales electroquímicas a través de la red. Cuando se estimula la primera neurona de la red, se procesa la señal de entrada y, si supera un umbral determinado, la neurona se activa y transmite la señal a las neuronas con las que está conectada. A su vez, estas neuronas se pueden activar y transmitir la señal a través del resto de la red. Con el entrenamiento, las conexiones entre las neuronas se refuerzan por el uso frecuente a medida que aprende a responder de forma eficaz.

Entre las propiedades que se destacan en el uso de las redes neuronales con respecto a los modelos estadísticos clásicos, es su aplicación sin la necesidad de considerar el cumplimiento de supuestos teóricos [17]. De esta manera, los modelos de redes neuronales pueden ser considerados como nuevos paradigmas para el análisis estadístico de regresión lineal [17] debido a la similitud de la variable de salida (variable respuesta Y) que se relaciona aplicando la función de activación (función identidad) sobre una combinación lineal de

pesos (coeficientes) con las variables de entrada (variables predictoras) [17].

Consideremos un ejemplo simple de modelo que expresa una relación lineal entre características y niveles. El modelo podría expresarse de la siguiente forma:

$$y = b + wx$$

Donde y es el nivel o etiqueta de un ejemplo de entrada, x es la característica de ese ejemplo de entrada, w es la pendiente de la recta o peso y es uno de los dos parámetros que tiene que aprender el modelo durante el proceso de entrenamiento para poder usarlo luego para inferencia y por ultimo, b es el punto de intersección de la recta en el eje y o sesgo y es el otro de los parámetros que deben ser aprendidos por el modelo.

Aunque en este modelo simple que hemos representado solo tenemos una característica de entrada, en el caso del aprendizaje profundo veremos que tenemos muchas variables de entrada, cada una con su peso w_i por lo que podemos generalizar:

$$Y = b + \sum_i w_i x_i$$

Los valores de entrada a la red se suelen identificar como x (Fig. 2.1). Cuando hay más de un valor de entrada, x se considera vector con elementos denominados x_1, x_2, y así sucesivamente. A su vez, asociado con cada valor x hay un valor de ponderación w, que se usa para fortalecer o debilitar el efecto del valor x a fin de optimizar el aprendizaje, es decir, representan la influencia relativa de la entrada por la cual se multiplica x_i. Además, se agrega otra entrada, el sesgo b (en inglés *bias*), que controla qué tan predispuesta está la neurona a activarse o no independiente de los pesos. Un sesgo alto hace que la neurona requiera una entrada más alta para generar una activación y un sesgo bajo lo hace más fácil.

Durante el proceso de entrenamiento, se ajustarán los valores w y b para optimizar la red de modo que "aprenda" a generar los resultados correctos [47]. La neurona calcula

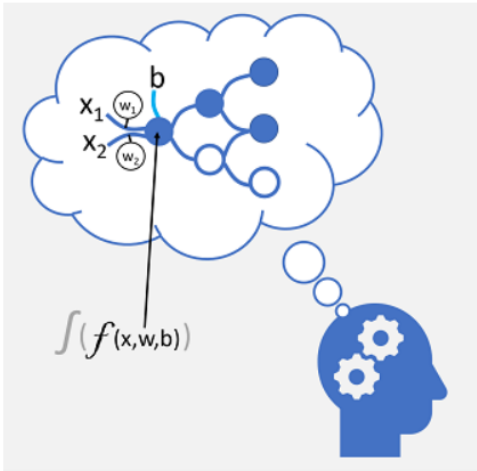


Figura 2.1: El aprendizaje profundo emula el proceso biológico que ocurre en el cerebro mediante redes neuronal artificiales que procesan entradas numéricas en lugar de estímulos electroquímicos. En las redes neuronales, los valores de entrada, x_1, x_2, tienen asociados ciertos valores de ponderación w y además un valor de sesgo b, que controla qué tan posible es que una neurona se active o no. Imagen extraída de [26]

una suma ponderada de x, w y b, la cual a su vez tiene una función de activación que restringe el resultado (por lo general, a un valor entre 0 y 1) a fin de determinar si la neurona transmite o no un resultado a la capa siguiente de neuronas de la red [26].

En cada capa se aprende un patrón de los datos sobre los que se basan las capas posteriores [47]. Por ejemplo, una red neuronal profunda encargada de interpretar formas aprendería a reconocer bordes simples en la primera capa y luego agregaría el reconocimiento de formas más complejas compuestas por esos bordes en capas posteriores [47].

Entrenamiento de una red neuronal

El proceso de entrenamiento de una red neuronal se compone de varias iteraciones o ciclos, llamadas épocas y consta de dos etapas: una etapa hacia adelante y una etapa hacia atrás. La época es un hiperpárametro de la red. Los hiperparámetros de un modelo son los valores de las configuraciones utilizadas durante el proceso de entrenamiento. Son

valores que generalmente se no se obtienen de los datos (como sí lo hacen los parámetros) sino que son indicados por el usuario.

Entonces, en la primera época o ciclo se comienza por asignar valores de ponderación (w) y de sesgo (b) que son totalmente aleatorios. Luego, las características de las observaciones de los datos con valores de etiqueta (*ground true*) conocidos se envían a la capa de entrada de a un dato por vez o pueden agruparse en lotes o grupos. Después, las neuronas aplican su función y si se activa, pasan el resultado hacia adelante a la capa siguiente y así sucesivamente hasta que la capa de salida genera una predicción. Esta predicción es comparada con el valor de etiqueta y se calcula la varianza entre los valores predichos y verdaderos, lo que se conoce como la pérdida. Una función de pérdida nos dice qué tan lejos está el modelo de algoritmo de realizar el resultado esperado.

En función de los resultados, la etapa hacia atrás, calcula gradientes de cada parámetro (lo que implica derivadas parciales) con reglas de cadena, es decir, se modifican los valores de ponderación y sesgo a fin de reducir la pérdida y estos ajustes se retro-propagan a la neuronas de las capas iniciales de la red. En la época siguiente, el entrenamiento se repite con propagación hacia adelante con los valores de ponderación y sesgo corregidos, a fin de mejorar la precisión del modelo mediante la reducción de la pérdida [31, 26]. Este proceso se repite la cantidad de épocas que hayamos indicado.

Hay otros hiperparámetros que debemos mencionar. Entre ellos, los los optimizadores que se utilizan para determinar en qué dirección se deben ajustar los parámetros de ponderación y sesgo para disminuir la pérdida en el modelo. El más famoso el es descenso por el gradiente, entre ellos, el más usado se llama optimizador de Adam.

Además, la tasa de aprendizaje es otro hiper parámetro que controla la tasa o la velocidad a la que aprende el modelo. Ésta, es un valor positivo que a menudo varía entre 0.0 y 1.0. Durante el entrenamiento, en la etapa hacia atrás, en lugar de actualizar el peso con la cantidad total, se escala según la tasa de aprendizaje. Esto significa que utilizando una tasa de aprendizaje de 0.1 (un valor predeterminado usual) los pesos en la red se

actualizan un 10 % del error del peso estimado cada vez que se actualizan. Esto podemos pensarlo como la búsqueda del mínimo global en un espacio multidimensional: si el valor es muy pequeño la actualización se puede quedar atrapada en un mínimo local y los valores de los pesos y del sesgo no cambiarían correctamente, igualmente la red neuronal tardara mucho más tiempo en optimizar. Por otro, lado si los valores son muy altos la actualización puede pasarse del punto perfecto y nunca encontrarlo y aunque el aprendizaje sea más rápido nunca llegara al mínimo global [30, 26].

2.2.2. Visión por computadora

La visión por computadora es el nombre que se le da a las tareas que incluyen métodos para adquirir, procesar, analizar y comprender imágenes o videos digitales. Como su nombre indica, el principal objetivo de la visión por computador es intentar imitar la funcionalidad del sistema de visión. Esto es una tarea compleja la cual ha llevado años de investigación para que un ordenador sea capaz de imitarlo con resultados razonables. La puesta en escena de la redes neuronales profundas y más concretamente de las redes neuronales convolucionales (las cuales tienen un tipo de capa característica en donde se produce una operación de convolución, más detalle en 2.2.3) han dado paso a este salto de nivel. El reconocimiento facial, de señales de tráfico o los coches autónomos son algunos de los ejemplos de las aplicaciones que se están consiguiendo.

Las máquinas interpretan las imágenes como un arreglo de elementos, los cuales pueden ser vectores, matrices o tensores, ordenados de modos específicos. Consideremos una imagen como una matriz de números de n columnas y m filas como podemos ver en la Figura 2.2. Luego, la intersección entre filas y columnas define los píxeles a los que se les asigna un valor que determina el color en esa posición de la imagen [29]. En el caso de imágenes en tonalidades de grises, el valor del píxel es un escalar (Fig. 2.3 a); mientras que para el caso de imágenes a color el valor de cada píxel está compuesto por información de tres matrices, cada una de las cuales especifica el grado de influencia de los colores.

Existen distintos modos de representación de imágenes a color como puede ser la RGB por el rojo (red "R"), verde (green "G") y azul (blue "B") (Fig. 2.3 b) o la CMYK (cían, magenta, amarillo y negro)[29]. Típicamente se emplean escalas para el rango de valores que podemos usar para representar colores en 2^N niveles. Es decir, para el caso más común de 8 niveles, la escala es [0 , 255], ya que por costumbre se define el rango como [0 , $2^N - 1$], en donde 0 es el valor para el negro y 255 representaría el color blanco.

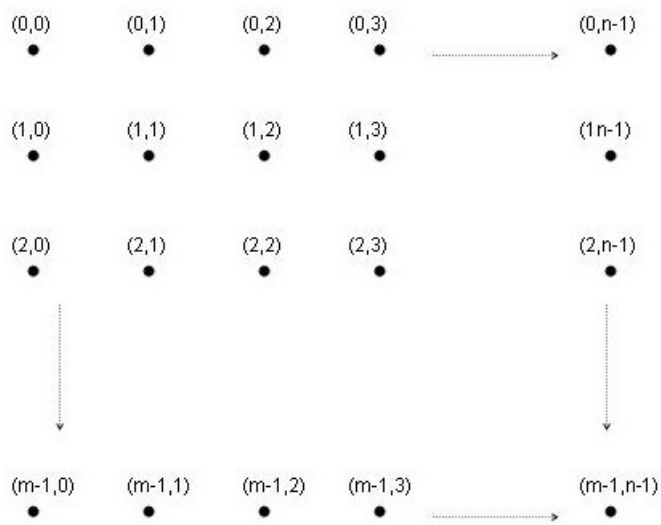
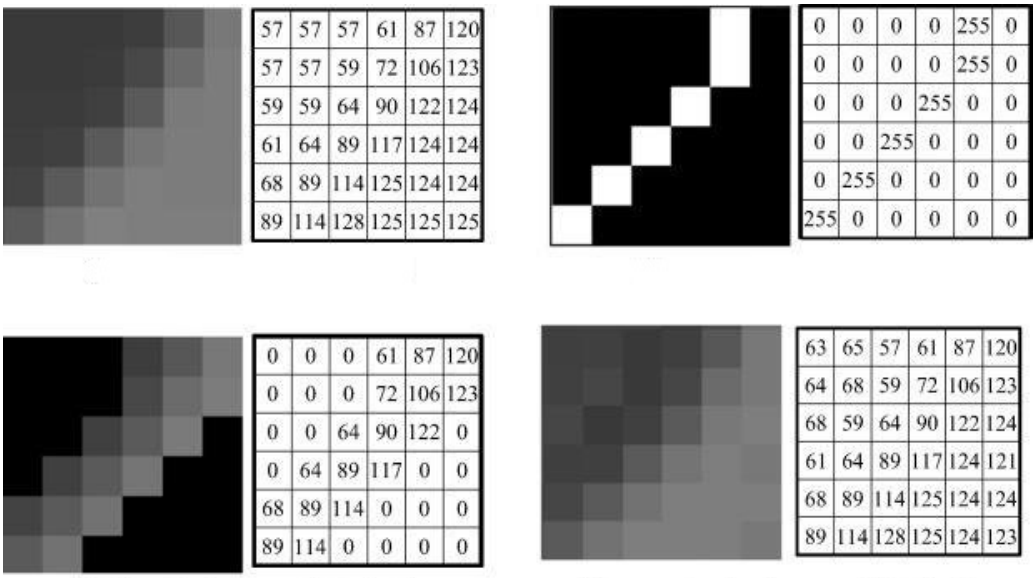


Figura 2.2: Ilustración de la disposición matricial de los valores de los píxeles de una imagen en n columnas y m filas. Arriba a la izquierda podemos observar el píxel definido por la fila 0 y la columna 0. En la esquina inferior derecha observamos el píxel definido por la la columna $n-1$ y la fila $m-1$



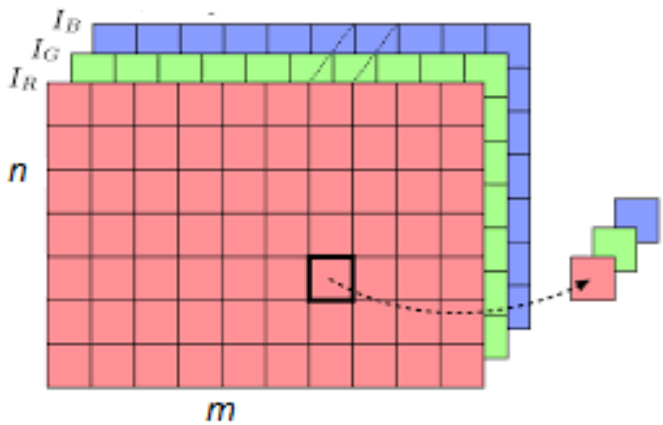
57	57	57	61	87	120
57	57	59	72	106	123
59	59	64	90	122	124
61	64	89	117	124	124
68	89	114	125	124	124
89	114	128	125	125	125

0	0	0	0	255	0
0	0	0	0	255	0
0	0	0	255	0	0
0	0	255	0	0	0
0	255	0	0	0	0
255	0	0	0	0	0

0	0	0	61	87	120
0	0	0	72	106	123
0	0	64	90	122	0
0	64	89	117	0	0
68	89	114	0	0	0
89	114	0	0	0	0

63	65	57	61	87	120
64	68	59	72	106	123
68	59	64	90	122	124
61	64	89	117	124	121
68	89	114	125	124	124
89	114	128	125	124	123

(a) Representación matricial de imágenes en tonalidades de gris



(b) Representación matricial correspondientes a imágenes a color en RGB

Figura 2.3: Representación matricial de imágenes. Para imágenes 2D los píxeles se representan como valores que determinan la intensidad en esa posición de la imagen. a) Imágenes en escala de grises la izquierda y a la derecha la matriz de valores correspondientes. Aquí se utiliza una escala en donde el 0 representa el negro y 255, el color blanco. b) En imágenes a color el valor de cada elemento de matriz es un vector de tres coordenadas, cada una de las cuales especifica el valor de los colores rojo, verde y azul (cuando se utiliza la representación RGB).

2.2.3. Redes neuronales convolucionales (CNNs)

El motivo del éxito del aprendizaje profundo en el área de la visión por computadora es un tipo de modelo denominado red neuronal convolucional (CNN) [18, 47, 31]. Por lo general, una CNN funciona extrayendo características de las imágenes y enviándolas a una red neuronal completamente conectada para generar una predicción. Así, en general, una CNN consta de tres capas neuronales principales (Fig. 2.4): capas convolucionales, capas de agrupación, y capas completamente conectadas.

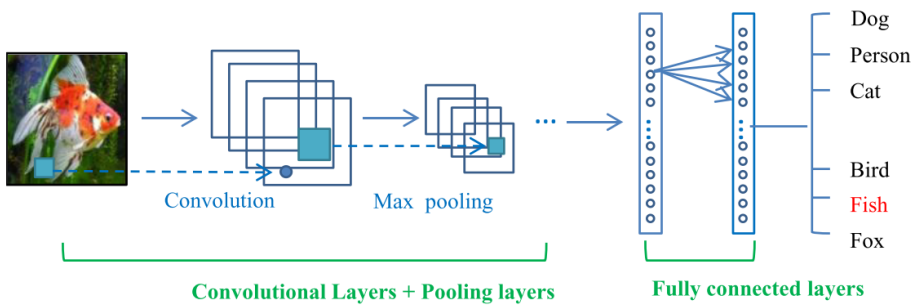


Figura 2.4: Arquitectura clásica de una red neuronal convolucional para un problema de clasificación de imágenes. Podemos ver que primero tienen capas de convolución, luego de agrupación y por último, capas totalmente conectadas. En las primeras capas, se extraen las características que definen a la imagen, como bordes y formas de un pez y las últimas capas son las encargadas de darles probabilidades a las categorías de salida para predecir qué imagen ha sido procesada por la red. Imagen extraída de [31].

Tipos de capas

Las redes neuronales están compuestas de diversas capas y cada una de ellas cumple un rol distintivo. Las más conocidas y comúnmente utilizadas son las capas convolucionales, las de agrupación y las totalmente conectadas.

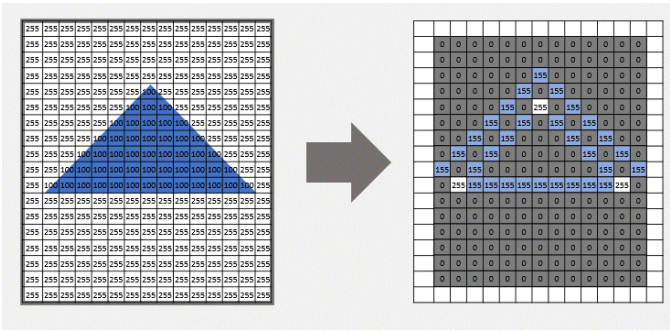
- Capas convolucionales

Una capa convolucional funciona aplicando un filtro a las imágenes (Fig. 2.5) que está

definido por un kernel que consta de una matriz de valores de ponderación. Luego, para aplicar el filtro, debe superponerlo en una imagen y calcular una suma ponderada de los valores de píxeles de la imagen correspondientes bajo el kernel del filtro. Después, el filtro se mueve (se convoluciona) por toda la imagen. Por lo general, se utiliza un tamaño de paso de 1 (es decir, se desplaza un píxel a la derecha) y se calcula el valor para el píxel siguiente. El proceso se repite hasta que se aplica el filtro en toda la imagen a fin de generar una matriz de valores nueva. La salida de la convolución generalmente se pasa a una función de activación, que a menudo es una función de unidad lineal rectificada (ReLU) que garantiza que los valores negativos se establezcan en 0.

Debido a los beneficios introducidos por la operación de convolución, algunos trabajos de investigación la utilizan como reemplazo de las capas totalmente conectadas para acelerar el proceso de aprendizaje [51].

(a) Filtro aplicado para el caso de ejemplo



(b) Imagen de entrada (derecha) y de salida (izquierda) luego de aplicar una capa de convolución

Figura 2.5: Aplicación de una operación de convolución. Podemos observar que la aplicación de un filtro de tamaño tres extrae las características de la imagen, en este caso, el borde del triángulo, apaga todos los pixeles que no le aportan información y al mismo tiempo disminuye el tamaño de la imagen. Imagen extraída de [26].

- Capas de Agrupación

En general, las capas de agrupación, o también llamadas de sub muestreo, son las capas que le siguen a las capas de convolución y se usan para disminuir la cantidad de valores

de características y con ello la dimensión, a la vez que se conservan las características diferenciales clave que se han extraído. Igual que las capas de convoluciones, son invariantes a la traslación porque sus cálculos tienen en cuenta los píxeles vecinos. Las estrategias más utilizadas son Max Polling (en donde la imagen es dividida en regiones del mismo tamaño, y para cada región se extrae el valor máximo que corresponderá a un píxel en la imagen resultante) y Average Polling (en donde se calcula el promedio de cada región).

- Capas totalmente conectadas

Las capas totalmente conectadas funcionan como una red neuronal tradicional (Fig. 2.6) y contienen alrededor del 90 % de los parámetros de una red [31]. El inconveniente de estas capas es que contienen muchos parámetros, lo que resulta en necesidad de cómputo para entrenarlas. Estas capas son las que se encargan de la predicción. Por lo tanto, una dirección prometedora y a veces aplicada es eliminar estas capas o disminuir las conexiones con un método determinado. Por ejemplo, GoogLeNet [51] diseñó una red profunda y amplia manteniendo constante el presupuesto computacional, cambiando de arquitecturas completamente conectadas a arquitecturas escasamente conectadas.

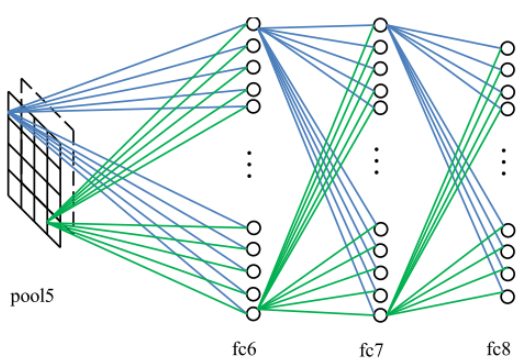


Figura 2.6: Operación de capas totalmente conectadas. Las capas totalmente conectadas funcionan como una red neuronal tradicional por lo que contienen muchos parámetros [31]. Estas capas son las que se encargan de generar la predicción. Imagen extraída de [31].

2.2.4. Redes convolucionales para regresión

Al usar redes neuronales aplicadas a detección de objetos en imágenes a través de un problema de regresión lo que se busca es predecir valores continuos que representen los bordes del área de detección en la que estamos interesados. En este sentido, la red no va a predecir una categoría como lo haría en un problema de clasificación, sino que la salida será uno o varios valores continuos, en nuestro caso, un vector de cuatro componentes que indicarán los puntos para la construcción del recuadro del área de interés definido por el extremo superior izquierdo (x_{min}, y_{min}) y el inferior derecho (x_{max}, y_{max}). En este punto nos encontramos la dificultad que genera clasificar la detección como correcta o incorrecta y de esta manera calcular los errores de predicción, ya que la predicción de estos cuatro puntos podría no ser igual a la colocada originalmente a mano pero de todas formas considerarse como aceptable ya que contiene el área de interés.

Red residual (ResNet)

Al rededor del año 2014, entre los investigadores la intención general era aumentar la profundidad de las redes neuronales para obtener un mayor nivel de rendimiento. Luego, se dieron cuenta que, agregar más capas a la red hacia que el valor de precisión se saturara o comenzara a disminuir abruptamente. El culpable de la disminución de la precisión fue el efecto de la degradación del gradiente [34]. Esto ocurre porque durante la etapa de retro-propagación, se calcula el error y se determinan los valores del gradiente para actualizar los pesos, lo que implica derivadas parciales. El proceso de determinación del gradiente y su envío a la siguiente capa oculta continúa hasta que se alcanza la capa de entrada. Así, el gradiente se vuelve cada vez más pequeño a medida que llega al principio de la red y los pesos de las capas iniciales se actualizarán muy lentamente o permanecerán iguales. En otras palabras, las capas iniciales de la red no aprenderán de manera efectiva. Así, el entrenamiento profundo de la red no convergerá y la precisión comenzará a disminuir o

saturarse en un valor particular [34, 49].

De esta manera, debido a que las redes neuronales más profundas son más difíciles de entrenar, se ha propuesto la utilización de redes neuronales residuales [35] (Fig. 2.7 adaptada a este trabajo). Estas redes fueron diseñadas para mejorar la precisión de redes neuronales profundas, de más de 100 capas debido a que con el aumento de la profundidad de las redes, la precisión se satura y aparece el problema de degradación del gradiente [35]. Las redes neuronales residuales evitan el sobre-ajuste mediante el uso de conexiones de salto o atajos para saltar sobre algunas capas [35].

Esto ocurre porque mediante el uso de una conexión de salto, proporcionamos una ruta alternativa para el gradiente en la propagación hacia atrás. La idea central es propagar hacia atrás a través de la función de identidad, simplemente usando una suma de vectores. Entonces el gradiente simplemente se multiplicaría por uno y su valor se mantendría en las capas anteriores. Esta es la idea principal detrás de las redes residuales: usar una función de identidad para preservar el gradiente [1].

Además de la degradación del gradiente, hay otra razón por la que los usamos comúnmente. Para una gran cantidad de tareas hay cierta información que se captura en las capas iniciales y es necesario permitir que las capas posteriores también aprendan de ella. Se ha observado que en capas iniciales las características aprendidas corresponden a información semántica inferior que se extrae de la entrada. Si no se utiliza la conexión de salto, esa información se podría volverse demasiado abstracta.

Resnet ha sido entrenada en el conjunto de datos ImageNet, un proyecto que contiene una gran base de datos visuales diseñados para su uso en la investigación de reconocimiento de objetos. El conjunto de datos contiene más de 14 millones de imágenes para indicar qué objetos se representan y en al menos un millón de imágenes, también se proporcionan cuadros delimitadores, con 1000 clases. Si bien esta red ha sido entrenada para un problema de clasificación, podemos utilizarla para regresión cambiando los parámetros de salida. De esta manera, podemos utilizar la red con parámetros pre-entrenados en donde solo se

actualizarán los pesos de las capas finales totalmente conectadas de las que derivamos las predicciones. Éste procedimiento se llama extracción de características en dónde puede usarse una red pre-entrenada como un extractor de características fijo y solo cambiar la capa de salida, un método de aprendizaje por transferencia. Se ha comprobado que el aprendizaje por transferencia produce resultados favorables en términos de tiempo de entrenamiento y precisión frente a un modelo entrenado desde cero[36].

Así, ResNet consiste en una combinación secuencial de algunos componentes básicos de éstos tipos de redes, como los saltos, y otros que mencionamos anteriormente:

- Capa Conv2d: calcula una convolución;
- BatchNorm2d: lo que hace esta capa es normalizar los valores. Esta normalización consiste en operaciones para que los valores tengan promedio 0 y desvío estándar 1 [37];
- ReLu: es la función de activación que devuelve cero si la entrada es menor a 0, y el valor de la entrada si es mayor a 0;
- MaxPool2d: agrupa varios píxeles de la imagen extrayendo el máximo de cada parche;
- AdaptiveAvgPool2d: esta sirve para que la red funcione sin importar el tamaño de la imagen de entrada. Es similar a MaxPool2d en el sentido de que agrupa varios píxeles, pero en este caso extrae el valor promedio de cada parche. Además, acomoda la cantidad y el tamaño de los parches para tener a la salida un tensor con el mismo tamaño, sin importar cuál fuese el tamaño de los tensores que venían propagándose por la red en las capas anteriores.;
- Capas totalmente conectadas que calculan una combinación lineal de la salida de todas las capas anteriores, y es la que determina la dimensión de la salida de la red, la cual corresponde con la cantidad de valores que debe predecir la red, en nuestro

caso va a predecir cuatro valores: (x_{min}, y_{min}, x_{max}, y_{max}), como veremos más adelante.

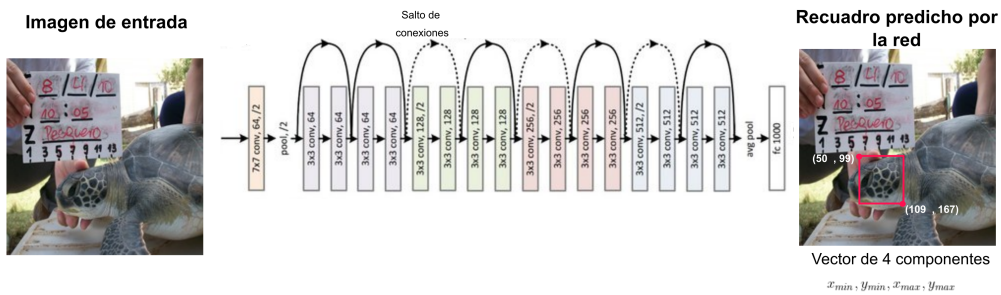


Figura 2.7: Diagrama de la red residual (ResNet) utilizada en la detección de escamas de tortugas marinas, Chelonia mydas, utilizando un problema de regresión. La red recibe como entrada una imagen y la salida es la predicción que consiste en un vector de cuatro componentes (x_{min}, y_{min}, x_{max}, y_{max}) que forman los bordes del rectángulo (marcado con rojo). Se indica la cantidad de filtros (filtros de 3x3) aplicados en las capas de convolución. Las líneas punteadas indican que se aumenta la dimensión. Imagen adaptada de [35].

3 Metodología

En este capítulo describiremos la metodología de trabajo. En la Fig 3.1 se muestra un diagrama de la metodología resumida para favorecer el entendimiento de los pasos seguidos y en las siguientes secciones se describe en detalle cada uno de ellos. Toda la tesis se realizó utilizando el lenguaje de Python en Jupyter notebook. El conjunto de datos que se uso para el entrenamiento así como el código se disponibilizarán una vez que la tesis esté publicada.

3.1. Generación de la base de datos

3.1.1. Área de estudio

La costa uruguaya consta de 710 km de longitud forma parte de un sistema hidrológico complejo que incluye la zona frontal del estuario del Río de la Plata y el Océano Atlántico [53]. La ONG local Karumbé realiza diversas tareas de investigación en la zona, entre ellas el estudio a largo plazo sobre la abundancia de tortugas marinas juveniles de la especie *Chelonia mydas* el cuál se realiza principalmente en el departamento de Rocha, en Cerro Verde e Islas de La Coronilla, un área protegida costero-marina. Para el estudio de la abundancia de individuos y el monitoreo de la agregación que reside en éstas aguas,

Karumbé ha tomado fotografías a tortugas desde el año 2000 hasta la actualidad. Para este estudio se tomaron en cuenta imágenes de tortugas verdes provienen de los departamentos de Canelones, Maldonado y Rocha. Para más detalles sobre el área de estudio referirse al trabajo de López-Mendilaharsu *et. at.* 2016 [43] y Vélez-Rubio *et. al.* 2013 [52].

3.1.2. Recopilación de datos

Este trabajo de tesis está enmarcado en un proyecto de foto-identificación que la ONG Karumbé realiza desde 2000. Para este estudio no se utilizó la base de fotos completa, sino imágenes entre 2001 y 2013 (n = 992) para el entrenamiento de la red y para construir la función de clasificación (objetivo 1 y 2) y otro conjunto obtenido entre 2017 y 2020 para el cálculo del error empírico (n = 98, objetivo 3). Cada imagen obtenida de una tortuga proviene de un registro que pudo provenir de uno de los siguientes eventos:

1. registros de animales varados durante censos de playas realizados por la ONG Karumbé o registros de personas que dieron aviso del evento de varamiento (ver Vélez-Rubio et al., 2013 [52]);

2. tortugas capturadas vivas mediante redes científicas mientras se alimentaban en áreas rocosas y arenosas; este método de captura y manejo de tortugas fue diseñado por técnicos de la ONG Karumbé en base a su experiencia en el comportamiento de las tortugas verdes en el área [43].

La técnica de fotografiado consistió en colocar al animal dorso-ventralmente sobre una superficie plana, extraer manualmente la cabeza para exponer las escamas faciales, retirar epibiontes (organismos sésiles que vive encima de otro ser vivo) si los hubiera y luego, limpiar la cabeza con agua de mar y tomar las fotografías a ambos lados de la cabeza. Todas las tortugas capturadas que se encontraban en buenas condiciones de salud después del muestreo fueron liberadas en el sitio de captura.

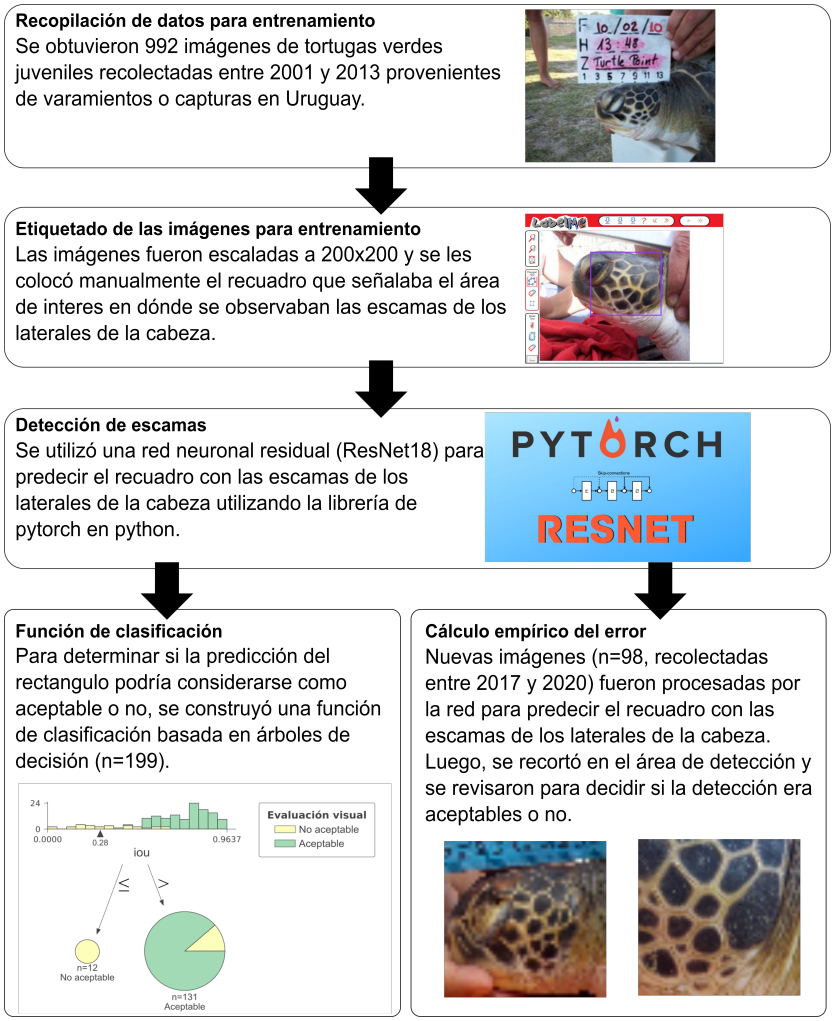


Figura 3.1: Diagrama del flujo de trabajo. Se comenzó por la generación de la base de datos proveniente de tortugas marinas juveniles de la especie Chelonia mydas en Uruguay. A 992 fotos se les colocó manualmente un recuadro que señalaba el área de interés para la detección automática utilizando un software en línea (LabelMe). El conjunto de datos se dividió en conjunto de entrenamiento (n=694) y de validación (n=99) que se utilizaron para entrenar y testear la red neuronal residual (ResNet18) con validación cruzada en 50 épocas y con 3 iteraciones. Luego, el conjunto de datos de test (n=199) se empleó para la función a posteriori para evaluar la aceptabilidad de la red. Por último, se calculó el error empíricamente procesando por nuestra red entrenada 98 imágenes que la red nunca había visto.

3.1.3. Curado de la base de datos

Para éste trabajo se tuvieron en cuenta imágenes del perfil izquierdo. Primero, las imágenes se escalaron para tener tamaño de 200x200 píxeles. Luego, se procedió a colocar manualmente un recuadro a 992 fotos que señale el área de interés, es decir, en donde se visualicen las escamas del lateral de la cabeza, utilizando el software LabelMe, una herramienta de anotación en línea para construir bases de datos de imágenes para la investigación de visión por computadora. Las anotaciones generadas se guardan en un archivo *.xml* en el que se encuentra la información de los vértices del recuadro. Como no tuvimos en cuenta que el orden en que se marcaban los rectángulos podría afectar en la generación del archivo, es decir, que si uno lo realizada de izquierda a derecha o de derecha a izquierda el orden de los puntos se cambiaría, tuvimos que utilizar funciones de mínimos y máximos para obtener el siguiente formato (x_{min}, y_{min}, x_{max}, y_{max}) para cada imagen (Fig. 3.2).

3.2. Detección de escamas

En este trabajo se utilizó la red neuronal llamada ResNet con 18 capas utilizando la librería de pytorch. Recordemos que la capa completamente conectada que calcula una combinación lineal de la salida de todas las capas anteriores determina la dimensión de la salida de la red predice cuatro valores: (x_{min}, y_{min}, x_{max}, y_{max}) que forman el recuadro deseado.

Entrenamiento de la red

ResNet18 se entrenó utilizando validación cruzada en 50 épocas y con 3 iteraciones y para ello el conjunto de datos se dividió de forma aleatoria en conjunto de datos en entrenamiento (70 %, n=694), validación (n=99, 10 %) y test (n=199, 20 %). El conjunto de datos de entrenamiento y validación se emplearon para ajustar los parámetros del

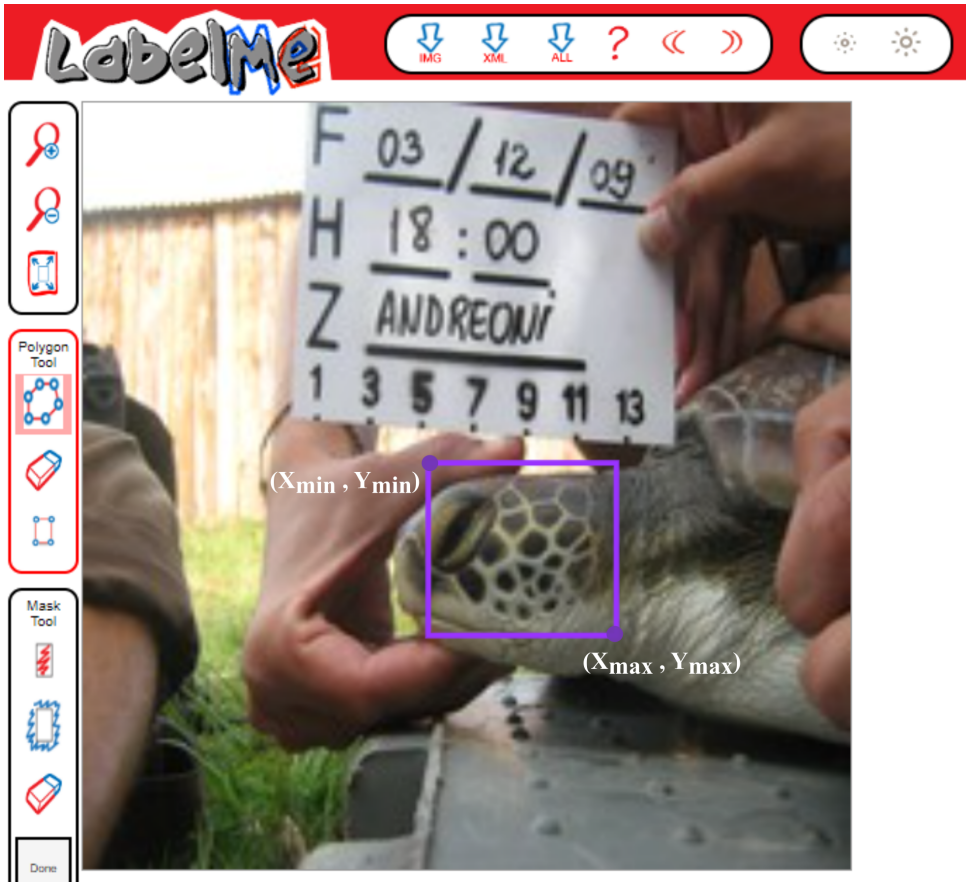


Figura 3.2: Visualización de LabelMe, la herramienta de anotación en línea que se utilizó para colocar los recuadros de las escamas de los laterales de la cabeza. Los puntos utilizados como referencia para la construcción del recuadro son el extremo superior izquierdo (x_{min}, y_{min}) y el inferior derecho (x_{max}, y_{max}).

modelo (pesos de las conexiones entre neuronas de las últimas capas). El conjunto de datos de test se empleó para evaluar la red final, ya entrenada y ajustada que luego se utilizó para la construcción de la función de clasificación que permite determinar si la detección es aceptable, que se verá en la sección 3.3.

En nuestro caso, como el objetivo de la red era predecir valores continuos, se trató como una tarea de regresión para lo que fue necesario modificar la capa de salida de

la red para que en lugar de 1000 tuviera 4 valores de salida, ya que esta red ha sido creada para predecir 1000 categorías pertenecientes al proyecto de Imagenet. Se utilizó como función de pérdida el error cuadrático medio (MSE) [55], que es el promedio de la diferencia al cuadrado entre los valores estimados y el valor real, entonces se puede usar como una medida de la calidad de un estimador. Como se deriva del cuadrado de la distancia euclidiana, siempre es un valor positivo que disminuye a medida que el error se aproxima a cero. Su fórmula matemática es:

$$MSE(\gamma, \widehat{\gamma}) = \frac{1}{n} \sum_{0}^{n-1} (\gamma - \widehat{\gamma})^2 \tag{3.2.1}$$

siendo γ para nuestro caso, un vector de dimensión cuatro con los vértices de los recuadros colocados manualmente y $\widehat{\gamma}$ un vector de dimensión cuatro con los vértices de los recuadros predichos por la red.

La red recibió como entrada un elemento de tamaño 5x3x200x200 que es el tamaño del lote de imágenes, el canal de color (rojo, verde, azul o RGB), el ancho y el alto de la imagen. Se utilizó un optimizador de Adam [41] y la velocidad de aprendizaje fue 0.001.

3.3. Función a posteriori que clasifique las detecciones como aceptables o no aceptables

Luego del entrenamiento, el mejor modelo en función de la menor pérdida se guardó en una carpeta para poderlo levantar y utilizarlo con los parámetros ya entrenados y optimizados para nuestro conjunto de imágenes. Las imágenes de test (n=199) se pasaron por la red entrenada y se calcularon las siguientes medidas de error:

- Error cuadrático medio (MSE)

Definido en 3.2.1

- Error absoluto medio (MAE)

Como su nombre indica, el error absoluto medio es un promedio de los errores absolutos $|e_i| = |\gamma - \widehat{\gamma}|$. Su fórmula matemática es:

$$MAE(\gamma, \widehat{\gamma}) = \frac{1}{n}\sum_{0}^{n-1} |\gamma - \widehat{\gamma}|$$

- Divergencia de Kullback-Leibler (D)

Esta métrica cuantifica cuánto una distribución de probabilidad difiere de otra distribución [38].

$$D(\gamma, \widehat{\gamma}) = \sum_{i=1}^{n} \gamma.(log\gamma - log\widehat{\gamma})$$

- Pérdida de probabilidad logarítmica negativa gaussiana (GLoss)

γ y $\widehat{\gamma}$ se tratan como muestras de distribuciones gaussianas con esperanzas y variaciones predichas por la red neuronal. Para un $\widehat{\gamma}$ modelado con una distribución gaussiana con un tensor de entrada de esperanzas y un tensor de varianzas positivas *var*, la pérdida es:

$$GLoss(\gamma, \widehat{\gamma}) = \frac{1}{2}(log(max(var, eps)) + \frac{(\gamma - \widehat{\gamma})^2}{max(var, eps)})$$

donde *eps* se usa para estabilidad [20].

- Pérdida de Huber (H)

La pérdida de Huber es una función de pérdida utilizada en la regresión robusta, que es menos sensible a los valores atípicos en los datos que la pérdida por error cuadrático [21]. Consideramos necesario incluir esta medida ya que podría haber algunas fotos con ruido (outliers) en la muestra de entrenamiento.

$$H = \begin{cases} \mathbf{\frac{1}{2}(\gamma - \widehat{\gamma})^2} & si \mid \gamma - \widehat{\gamma}| \leq \delta \\ \\ \boldsymbol{\delta(|\gamma - \widehat{\gamma}| - \frac{1}{2}\delta)} & c.c. \end{cases}$$

Esta función es cuadrática para valores pequeños de $(\gamma - \widehat{\gamma})$ y lineal para valores grandes, en donde δ especifica el umbral en el que cambia entre la pérdida MAE y MSE.

- Pérdida de Smooth (S)

La pérdida Smooth se puede interpretar como una combinación de pérdida de MAE y pérdida de MSE. Se comporta como MAE cuando el valor absoluto del argumento es alto y se comporta como MSE cuando el valor absoluto del argumento es cercano a cero [22]. la ecuación es:

$$S = \begin{cases} \mathbf{\frac{0,5}{\beta}(\gamma - \widehat{\gamma})^2} & si(\gamma - \widehat{\gamma}) < \beta \\ \\ \boldsymbol{|\gamma - \widehat{\gamma}| - 0{,}5\beta} & c.c. \end{cases}$$

La pérdida de Smooth está estrechamente relacionada con la pérdida de Huber, siendo equivalente a $\frac{H(\gamma - \widehat{\gamma})}{\beta}$ (el hiperparámetro β de Smooth también se conoce como δ para Huber), sin embargo tienen algunas diferencias:

Cuando β tiende a 0, la pérdida de Smooth converge a MAE, mientras que la de Huber converge a una pérdida constante de 0. Cuando beta es 0, la pérdida de Smooth es equivalente a la pérdida de MAE. Por otro lado, cuando β tiende a $+\infty$, la pérdida de Smooth converge a una pérdida constante de 0, mientras que la de Huber converge a MSE.

- Intersección sobre la unión (IoU)

Si denotamos como A al área comprendida por el rectángulo colocado manualmente y B

al área comprendida por el rectángulo predicho, denotamos

$$IoU = \frac{|A \bigcap B|}{|A \bigcup B|}$$

3.3.1. Elección de la función de clasificación

Para la construcción de nuestra variable a predecir (*target*) se revisó una por una las 199 imágenes del conjunto de test pasadas por la mejor red para decidir si los rectángulos predichos se consideraban como aceptables o no (1=aceptable; 0=no aceptable), es decir que el criterio para decidir si el recuadro predicho es aceptable, es la evaluación visual por parte del usuario. Luego, se construyeron árboles de decisión para caracterizar la detección a través de un vector de características construido con las medidas de error nombradas en 3.3 (MSE, MAE, D, GLoss, H, S) y se agregó otra variable que indica la relación entre el área del rectángulo colocado a mano y la imagen total (200 píxeles de ancho y 200 de alto):

$$área = \frac{(x_{max} - x_{min}).(y_{max} - y_{min})}{40000}$$

Esta medida se creyó importante incluir para evaluar fotos tomadas de cerca o lejos podrían introducir un mayor error.

Este conjunto de datos se dividió en conjunto de entrenamiento (70 %, n=143), validación (20 %, n=40) y test (10 %, n=16) nuevamente. El conjunto de entrenamiento y validación se utilizaron para el entrenamiento de los árboles y el de test para compararlo con otros criterios de detección de objetos (ver Criterios basados en umbrales de intersección sobre la unión 3.3.1).

Se utilizó un árbol de clasificación con los parámetros obtenidos a través de optimización para el criterio de decisión (entropía/gini), la estrategia para la división en cada nodo (el mejor/aleatorio), el número mínimo de muestras requeridas para dividir un nodo interno (enteros de 0 a 14), el número mínimo de muestras requeridas para estar en un

nodo hoja (1, 2, 5) y el control de la aleatoriedad del estimador (enteros del 0 al 14). La optimización de parámetros se realizó a través de validación cruzada con 5 iteraciones y teniendo en cuenta área bajo la curva para escoger el mejor estimador. Los parámetros del árbol elegido fueron gini como criterio de entropía, estrategia de decisión para cada nodo aleatoria, dos muestras mínimas para dividir un nodo interno, dos muestras mínimas para estar en un nodo hoja, control de la aleatoriedad del estimador igual a cuatro profundidad de tres nodos.

Se calculó la exactitud, la precisión, la sensibilidad y el F1-score para el árbol elegido. Las métricas se calculan utilizando verdaderos y falsos positivos, verdaderos y falsos negativos. Positivo y negativo en este caso son nombres genéricos para las clases predichas. De esta forma se define:

- TP (True Positive): cuando un caso fue positivo y se predijo positivo,
- TN (True Negative): cuando un caso fue negativo y se predijo negativo,
- FP (Falso Positivo): cuando un caso fue negativo pero predicho positivo,
- FN (Falso Negativo): cuando un caso fue positivo pero se predijo negativo

con el fin de calcular :

$$Exactitud = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Precisión = \frac{TP}{TP + FP}$$

$$Sensibilidad = \frac{TP}{TP + FN}$$

$$F1 - score = \frac{2TP}{2TP + FP + FN}$$

Se decidió trabajar con un árbol que tuviese como máximo tres nodos, debido a que con la profundidad se incrementaría la comprensión de la función de clasificación. Luego,

se evaluó la posibilidad de recortar el árbol elegido a uno o dos nodos para simplificar la función de clasificación, evaluando la exactitud.

En este punto se utilizó la librería de Sklearn de Python.

Criterios basados en umbrales de intersección sobre la unión

Esta sección tuvo como objetivo comparar la función de clasificación que proporcionaba en árbol de clasificación con la evaluación visual a través del usuario experimentado y con los criterios basados en IoU comúnmente usados en detección de objetos. Se usó el conjunto de datos de test (n=16) para predecir la clasificación del árbol de decisión con mayor exactitud (aquel con 3 nodos, ver Resultados) y se obtuvo la cantidad de fotos con predicción aceptable o no aceptable. Este resultado se comparó con el criterio de evaluación visual, es decir revisar con el ojo humano cómo se clasificaba esa detección, y con criterios usualmente utilizados en detección de objetos. Estos criterios se basan en establecer un umbral de aceptación a través de la métrica de IoU. Los umbrales comúnmente utilizados son 50 %, 75 % o 95 % [33]. Entonces, a partir de estos criterios se pueden tomar como aceptable cuando IoU $\geq$ umbral o; no aceptable cuando IoU $<$ umbral

3.4. Cálculo empírico del error basado en inspección

Esta parte de la investigación tuvo como objetivo evaluar si la red entrada era capaz de generalizar la detección para otro conjunto de datos obtenidos en otros años con el objetivo de poder utilizarse en la práctica (recordemos que en la ONG Karumbé participan muchas personas entonces las personas que toman las fotografías pueden cambiar de un año a otro). Entonces, para determinar el error empírico de la detección de escamas de tortugas se pasaron por la red entrenada un nuevo conjunto de datos de 98 imágenes de tortugas que la red de detección nunca había visto y se predijo el recuadro con las escamas de los laterales de la cabeza para cada una de las imágenes. Estas imágenes fueron

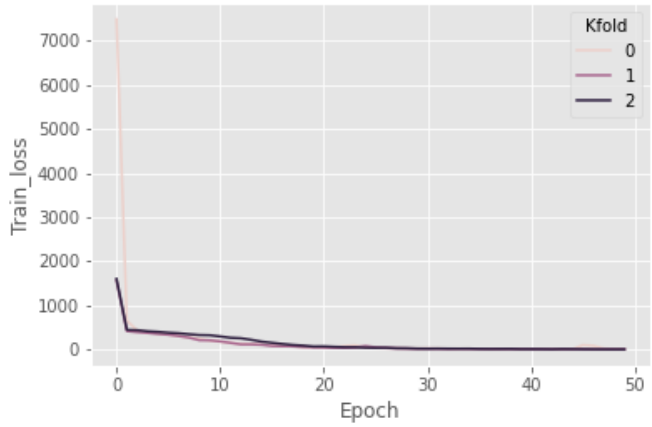
completamente independientes de las utilizadas para el entrenamiento, fueron obtenidas entre 2017 y 2020 y se eligieron aleatoriamente. Luego, se recortó en el área de detección de la foto original y se guardó el resultado del recorte en otra carpeta para su posterior revisión, simulando lo que se haría si estuviésemos utilizando la detección para su posterior análisis en un software de foto-identificación. Estas imágenes se revisaron por el mismo usuario que anteriormente y se anotó si la detección era aceptable o no. El error de la detección se determinó como la división entre la cantidad de detecciones consideradas como no aceptables y la cantidad de imágenes evaluadas.

4 Resultados

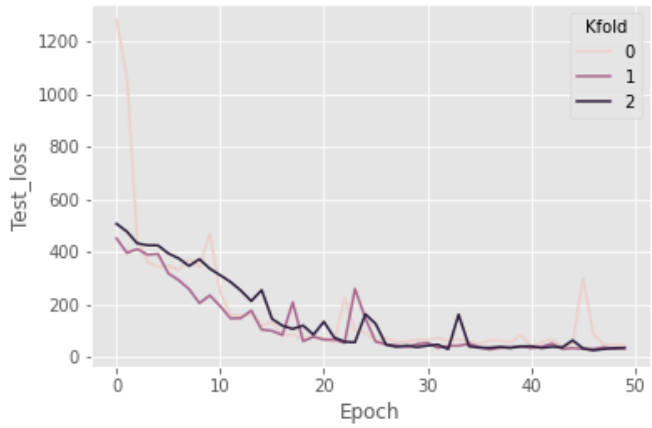
4.1. Detección de escamas

Durante el entrenamiento y el validación de la red ResNet18 aplicada a la detección de escamas de tortugas marinas se pudo observar que el valor de la pérdida disminuyó a través de las épocas y no se observó sobre-ajuste (Fig. 4.1). El entrenamiento de la red tuvo en promedio un MSE de 170.9 (desviación estándar = 616.6) y 154.6 (desviación estándar = 199.6) para el conjunto de validación. Estos datos podemos interpretarlos con más claridad al calcular la raíz cuadrada del MSE y obtener un resultado en unidades de píxeles. De esta manera podemos decir que como $\sqrt{170{,}9} = 13{,}07$, en promedio los recuadros predichos por la red en el entrenamiento se diferenciaban por 13.07 píxeles y en la validación de 12.43 ($\sqrt{154{,}6} = 12{,}43$).

Podemos observar ejemplos de imágenes donde la predicción se consideró aceptable, las cuales se encuentran en la Figura 4.2, en ellas se observa que los rectángulos colocados a mano son similares a los predichos por la red. Por otro lado, podemos ver ejemplos de imágenes consideradas no aceptables que se encuentran en la Figura 4.3 en donde se observa que los rectángulos predichos presentan variaciones con respecto a los colocados a mano.



(a) Conjunto de entrenamiento



(b) Conjunto de validación

Figura 4.1: Evolución de la pérdida, en nuestro caso, del error cuadrático medio (MSE) para el conjunto de datos de entrenamiento y validación utilizando ResNet18, con validación cruzada de 3 iteraciones y durante 50 épocas para la detección de escamas de tortugas verdes juveniles.

Figura 4.2: Ejemplos en los que la predicción de ResNet18 sobre la detección de escamas que se consideró como aceptable. Se puede observar que los recuadros colocados a mano (rojo) y los predichos (azul) son similares.

Figura 4.3: Ejemplos en los que la predicción de ResNet18 sobre la detección de escamas se consideró como no aceptable. Se puede observar que los recuadros colocados a mano (rojo) y los predichos (azul) no son similares, detectando por ejemplo menos porción de escamas o detectando el cuelo.

4.2. Función a posteriori que clasifique las detecciones como aceptables o no aceptables

4.2.1. Elección de la función de clasificación

La muestra de entrenamiento se compuso por 159 registros considerados como aceptables y 40 considerados como no aceptables. A la hora de elegir la función de clasificación construida con árboles de clasificación, la exactitud para el entrenamiento fue del 94.41 % y para la validación fue del 95.0 %. Tomando como referencia las predicciones aceptables, encontramos una precisión del 96.50 % (Tabla 4.1, la matriz de confusión se encuentra en 4.2), la cual es mayor a si tomamos como referencia a las detecciones no aceptables, es decir que es más sensible a encontrar predicciones aceptables. Lo mismo ocurre con los valores de sensibilidad y F1-score.

La función de clasificación con mayor exactitud obtenida se encuentra en la Figura 4.4 a. Dentro de las variables predictoras incluidas en nuestro modelo, la variable más importante fue la intersección sobre la unión, seguida por el error cuadrático medio, y la pérdida gausiana (Figura 4.4 b). Ésta función de clasificación es capaz de diferenciar dos nodos finales puros en donde se agrupa el 80 % de los datos (103+12=115 de 143 de la muestra de entrenamiento).

El umbral que divide el primer nodo en referencia a IoU es de 0.28, es decir, por debajo de ese valor se consideran todas la detecciones como no aceptables. Luego, por debajo de 80.95 para MSE se divide una rama que toma como referencia a la pérdida gausiana en donde si el valor se encuentra por debajo de 26.3 puede interpretarse las detecciones como aceptables. Los demás nodos finales no son nodos puros.

Cuando se recortaron las ramas a dos nodos la exactitud no cambió con respecto a los 3 nodos (Tabla 4.3): en el entrenamiento fue del 94.40 % y en el validación del 95.0 % (Fig. 4.5 a) pero sin embargo, pudimos observar que obteníamos un único nodo final puro

con predicciones no aceptables (n=12). Por otro lado, el árbol con un nodo obtuvo una exactitud menor en ambos conjunto de datos: en el entrenamiento del 89.51 % y en el validación, 82.5 % (Imagen 4.5 b) y también obteníamos un único nodo final puro.

	precisión	sensibilidad	F1-score
No aceptable	0.909	0.909	0.909
Aceptable	0.965	0.965	0.965

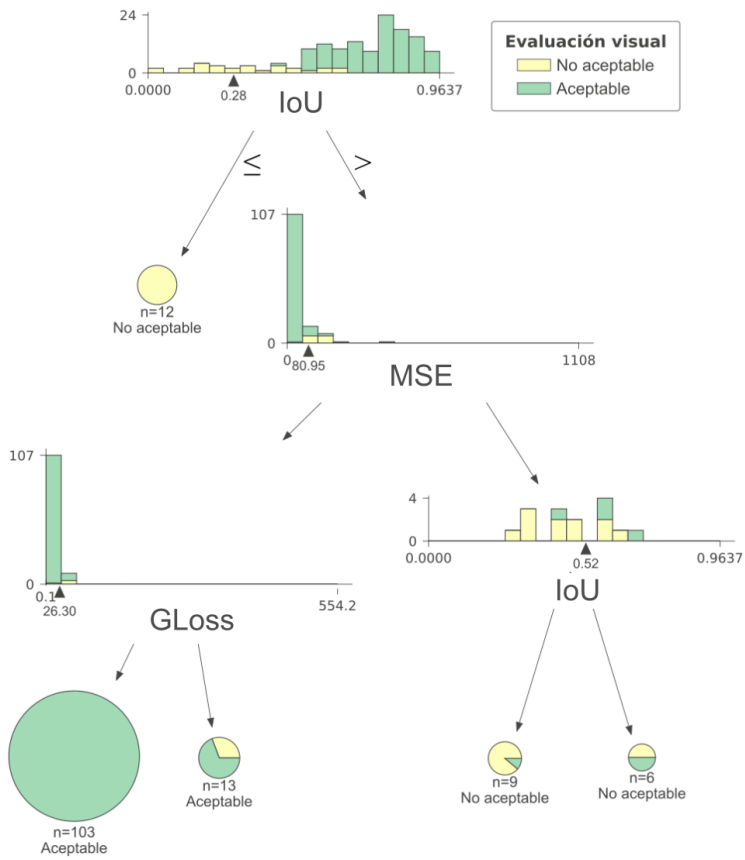
Tabla 4.1: Métricas de precisión para el árbol de clasificación para evaluar la detección de escamas de tortugas marinas como aceptables o no

		Predicho	
		Aceptable	No Aceptable
Valores verdaderos	Aceptable	28	1
	No aceptable	1	10

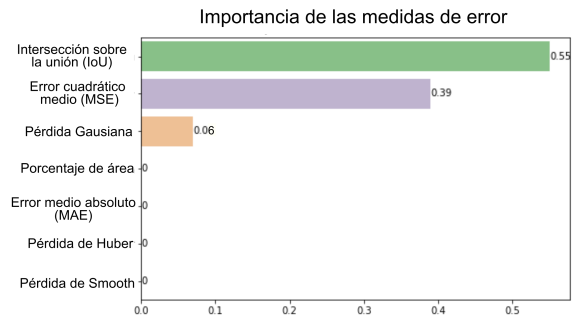
Tabla 4.2: Matriz de confusión para el árbol de clasificación con tres nodos para evaluar la detección de escamas de tortugas marinas como aceptables o no

	Árbol 3 nodos	Árbol 2 nodos	Árbol 1 nodo
Exactitud en entrenamiento	95.41 %	94.40 %	89.51 %
Exactitud en testeo	95.00 %	95.00 %	82.50 %
Cantidad de métricas implicadas	3	2	1
Cantidad de nodos puros	2	1	1

Tabla 4.3: Comparación de los árboles de clasificación con tres, dos y un nodo para evaluar la detección de escamas de tortugas marinas como aceptables o no

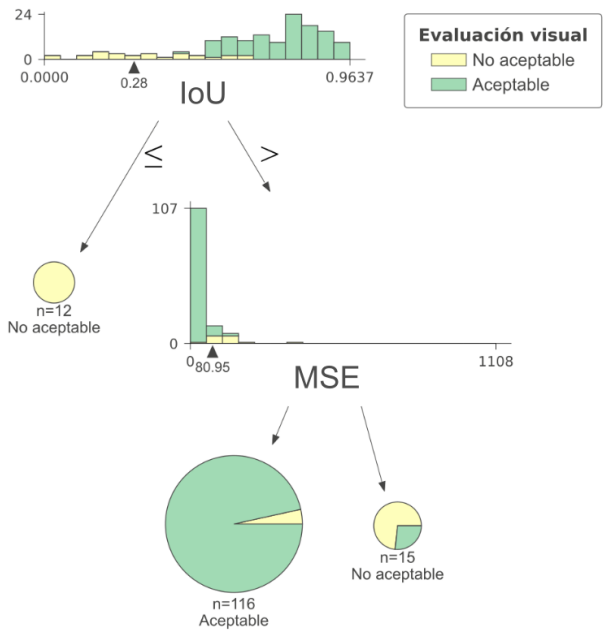


(a) Visualización del árbol con tres nodos

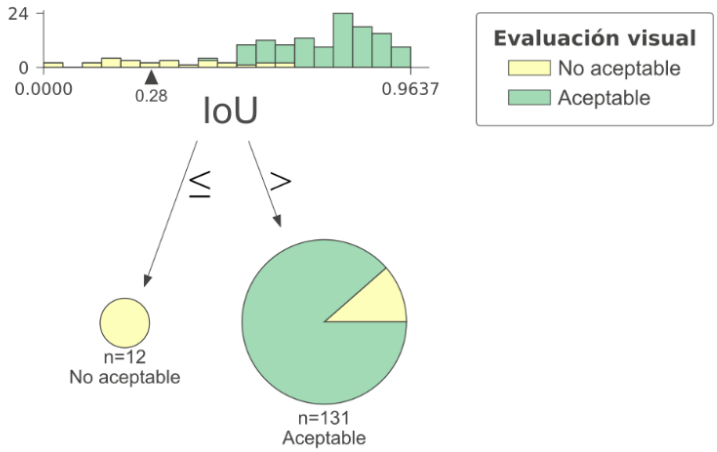


(b) Importancia de las variables

Figura 4.4: Árbol de clasificación con tres nodos para la determinación de detecciones aceptables de escamas de tortugas. En a) podemos ver la función de clasificación con una exactitud del 95.0% para la validación. En b) se encuentran las variables que tienen más importancia en la clasificación.



(a) Visualización del árbol con dos nodos



(b) Visualización del árbol con un nodo

Figura 4.5: Árboles de clasificación con dos y un nodo para la determinación de detecciones aceptables de escamas de tortugas. a) Cuando se recortaron las ramas a dos nodos la exactitud fue 95.0% para la validación. b) El árbol con un único nodo obtuvo una exactitud de 82.5% para la validación.

Criterios basados en umbrales de intersección sobre la unión

Se pudo observar que los criterios de evaluación visual, de árbol de clasificación y de IoU del 50 % resultaron en la misma proporción de detecciones aceptables y no aceptables, considerando el 88 % de las imágenes (n=14) como aceptables (Fig. 4.6). Es decir que tanto el criterio del árbol de clasificación como el de IoU del 50 % proporcionan la misma clasificación que la definida como referencia con la observación visual por parte del usuario. El criterio de IoU del 75 % consideró el 62 % de las imágenes (n=10) como aceptables y 38 % (n=6) como no aceptables. El criterio de IoU del 95 % fue muy estricto por lo que consideró a todas las detecciones como no aceptables.

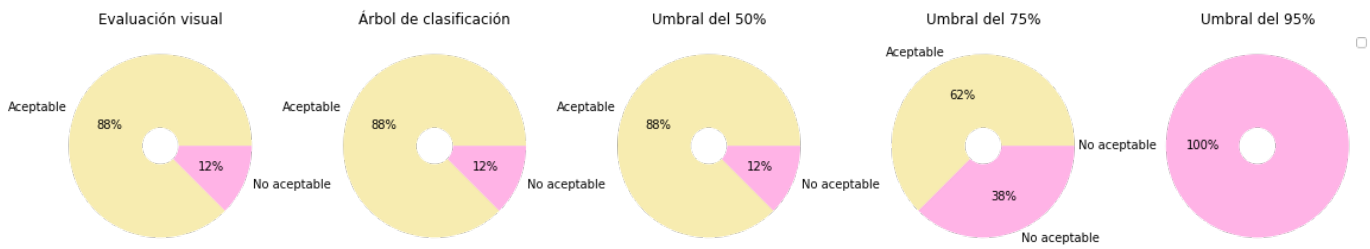
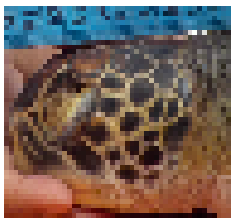


Figura 4.6: Proporción de detecciones de imágenes de escamas (n=16) consideradas como aceptables (amarillo) y no aceptables (rosa) teniendo en cuenta la evaluación visual por parte del usuario, el árbol de clasificación con tres nodos y la intersección sobre la unión con umbrales de 0.5, 0.75 y 0.95.

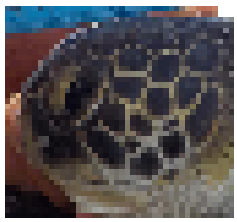
4.3. Cálculo empírico del error basado en inspección

Luego de la evaluación visual por parte del usuario de la predicción sobre 98 imágenes que la red nunca había visto, obtuvimos un total de 10 imágenes en las que se consideró que la detección de las escamas era no aceptable y 88 en los que la predicción se consideró aceptable. Esto arroja como resultado un error del 10.2 %, es decir, el 89.8 % de las imágenes se predijo con un recuadro que podría ser utilizado para foto-identificación.

En la Fig. 4.7 se pueden observar los recortes de las imágenes que podrían ser utilizadas por un software de foto-identificación. Observamos en a) y en b) ejemplos de imágenes que podrían utilizarse directamente y en c) y d) se visualizan imágenes que deberían volverse a recortar manualmente porque no se consideran aceptables ya que la zona de interés en donde se observan las escamas, no se ha detectado.



(a) Recorte clasificado como aceptable



(b) Recorte clasificado como aceptable

(c) Recorte clasificado como no aceptable

(d) Recorte clasificado como no aceptable

Figura 4.7: Recortes de imágenes provenientes de la detección predicha por la red ResNet18 entrenada para la detección de las escamas de los laterales de la cabeza de Chelonia mydas en individuos juveniles. En los recortes a) y b) vemos ejemplos de detecciones clasificadas como aceptables que pueden ser utilizadas para foto-identificación. En los recortes c) y d) vemos ejemplos de recortes no aceptables en los que se debería recortar a mano el área de interés para poder utilizarse en foto-identificación.

5 Conclusiones

Este estudio aporta a disminuir la ardua tarea de recortado que se requiere para realizar foto-identificación, en especial cuando se tiene un conjunto de datos grande. Este es el caso de la ONG Karumbé la cual recolecta fotos desde 2001 y posee un volumen de imágenes que se multiplica año a año debido a la tarea de conservación que se realiza en toda la costa uruguaya con el aporte de muchas personas.

Cuando realizamos la evaluación de la red en fotos nuevas, fue posible recortar 98 imágenes en menos de un minuto, trabajo que puede demorar un promedio de 40 segundos por imagen (un total de aproximadamente una hora, estimaciones propias), por lo que podemos dimensionar que la tarea del recorte automático reduce su tiempo notablemente. Al hacer este recorte automático no solo obtuvimos mejoras en el tiempo de procesado sino que la red neuronal detectó las escamas de las tortugas como aceptables en un 89.8 % de las imágenes. Es decir, a la hora de aplicar la red para el recorte de imágenes será necesario revisar un 10.2 % de las imágenes y recortarlas a mano pero habrá una reducción considerable del trabajo. En este punto es muy importante tener en cuenta que las imágenes utilizadas en el trabajo provenían de distintas cámaras, así como operarios o en diversos fondos pero de todas maneras el algoritmo de detección fue muy eficaz. Esta diferencia en las imágenes es debida al propio trabajo de conservación e investigación que se realiza en Karumbé en donde intervienen muchas personas y circunstancias diversas.

Esta precisión podría mejorarse en el futuro entrenando con un mayor conjunto de datos, ampliando el tamaño de las imágenes (recordar que para este estudio todas las fotos se escalaron a un tamaño de 200x200 píxeles) o estandarizando el tipo de fotografías. Sin embargo, de esta manera hemos obtenido resultados valiosos.

Pudimos obtener una función de clasificación sencilla, construida a través de árboles de decisión con tres nodos que nos permitió clasificar las detecciones de las escamas de predichas por ResNet como aceptables o no. Esta función es fácil de comprender porque está construida por tres métricas: la intersección sobre la unión (IoU), el error cuadrático medio (SME) y la pérdida gausiana, las cuales ayudan a la segmentación de las detecciones. De esta manera se pueden identificar rápidamente un conjunto de imágenes no aceptables como un nodo puro del árbol si la intersección sobre la unión es menor o igual a 0.28 (n=12). Luego, por encima de ese valor de IoU, el error cuadrático medio con un umbral de 80.95 continúa la segmentación. Por debajo de 80.95 de SME y por debajo de 26.30 de GLoss encontramos otro nodo puro en donde las detecciones de escamas se consideran aceptables (n=103) y de esta manera, la función permite clasificar el 80 % de los datos. Así mismo cuando se recortaron las ramas a dos nodos la exactitud no cambió con respecto a los tres nodos y las interpretaciones del árbol se restringen a las métricas de IoU y MSE, por lo que podría utilizarse por su simpleza pero no se garantizaría un nodo puro. Por último, el árbol con un nodo obtuvo una exactitud menor en ambos conjunto de datos, pero podríamos sintetizar la función de clasificación y asegurar una exactitud del 82.5 % tomando sólo la intersección sobre la unión, lo cual es un resultado muy simple de utilizar y fácil de interpretar: una intersección menor al 28 % entre recuadros colocados a mano y predichos, clasifica las detecciones de escamas en tortugas marinas como no aceptables.

La importancia de IoU obtenida en este estudio aporta sustento a la utilidad de ésta métrica ampliamente usada en la evaluación de detección de objetos [6]. En nuestro caso, nuestro árbol de tres nodos fue equivalente a considerar una superposición del 50 % para determinar como aceptable la detección. Los criterios del 75 % y 95 % de solapamiento

evidentemente son muy estrictos y para este problema no es necesario considerar una similitud entre la precisión y el recuadro original tan riguroso. Incluso, en nuestro caso de aplicación se puedo demostrar que con un solapamiento menos al 28 % ya podemos descartar detecciones no aceptables, luego para determinar las aceptables es necesario recurrir a las medidas de MSE y GLoss. En el futuro podría evaluarse la posibilidad de utilizar la medida de IoU como función de coste para el entrenamiento de una nueva red.

Este trabajo proporciona el primer informe del uso de detección automática de las escamas de tortugas marinas juveniles para su posterior uso en PID para datos a largo plazo en áreas de alimentación del Atlántico Sur Occidental. Este trabajo aporta valor a la tarea de PID de tortugas marinas, en especial de la tortuga verde, permitiendo la detección de las escamas de los laterales de la cabeza para la posterior comparación por un software de identificación. Nuestro trabajo prueba que la detección a través de la red neuronal ResNet18 es efectivo y tiene una precisión que ayuda a reducir el tiempo de recorte de las imágenes.

6 Perspectivas

Este trabajo es de gran utilidad debido a que el producto final es fácil de utilizar ya que no se necesita conocimiento de aprendizaje automatizado o de programación para su implementación. No solo va a ser de gran utilidad para seguir trabajando en la base de datos de Karumbé sino que también es aplicable a imágenes de otros proyectos, incluso de distintos lugares del mundo en donde se tienen las mismas dificultades a la hora de realizar estudios de marca-recaptura.

Los pasos a seguir para su implementación serían los siguientes: primero, la persona debería obtener imágenes digitales de tortugas marinas similares a las que hemos utilizado en este proyecto, esto implica que sean tomadas a una distancia media en la que además de la cabeza se observen otras partes del cuerpo y del fondo, es decir, que no sean solamente de la cabeza. Luego, estas imágenes deberían separarse en lados, por un lado las izquierdas y por otro las derechas. En este trabajo hemos comprobado que la red detecta las imágenes del perfil izquierdo pero no hay razón a priori para suponer que no podría funcionar para el otro lado también. Posteriormente, las imágenes pueden procesarse por la red utilizando la notebook de Jupyter simplemente al cambiar en el código de python, la ruta de la carpeta con las fotos que queremos recortar. Finalmente, en una carpeta separa, se guardarán las imágenes recortadas.

En el futuro proponemos utilizar este sistema de detección para un conjunto de datos

de tortugas capturadas en el Sur Brasil, en una agregación de tortugas verdes juveniles que pertenecen a la misma población de tortugas verdes utilizada para este estudio.

7. Referencias

[1] Nikolas Adaloglou. *Intuitive Explanation of Skip Connections in Deep Learning*. https://theaisummer.com/skip-connections/. 2020.

[2] Gonzalo Araujo y col. "Using minimally invasive techniques to determine green sea turtle Chelonia mydas life-history parameters". En: *Journal of Experimental Marine Biology and Ecology* 483 (2016), págs. 25-30. ISSN: 00220981. DOI: 10.1016/j.jembe.2016.06.004. URL: http://dx.doi.org/10.1016/j.jembe.2016.06.004.

[3] J. W. Arntzen y col. "Cost comparison of marking techniques in long-term population studies: PIT-tags versus pattern maps". En: *Amphibia-Reptilia* 25.3 (2004), págs. 305-315. ISSN: 0173-5373.

[4] Cecilia Bardier y col. "Performance of visual vs. software-assisted photo-identification in mark-recapture studies: a case study examining different life stages of the Pacific Horned Frog (Ceratophrys stolzmanni)". En: *Amphibia-Reptilia* (2020), págs. 1-12. ISSN: 0173-5373. DOI: 10.1163/15685381-bja10025.

[5] Cecilia Bardier y col. "Quantitative Determination of the Minimum Body Size for Photo-identification of Melanophryniscus montevidensis (Bufonidae)". En: *Herpetological Conservation and Biology* 12 (2017), págs. 119-126.

[6] Floris van Beers y col. “Deep Neural Networks with Intersection over Union Loss for Binary Image Segmentation.” En: *ICPRAM*. 2019, págs. 438-445.

[7] Qifang Bi y col. “What is machine learning? A primer for the epidemiologist”. En: *American Journal of Epidemiology* 188.12 (2019), págs. 2222-2239. ISSN: 14766256. DOI: 10.1093/aje/kwz189.

[8] Douglas T. Bolger y col. “A computer-assisted system for photographic mark-recapture analysis”. En: *Methods in Ecology and Evolution* 3.5 (2012), págs. 813-822. ISSN: 2041210X. DOI: 10.1111/j.2041-210X.2012.00212.x.

[9] Leo Breiman. “Statistical modeling: The two cultures”. En: *Statistical Science* 16.3 (2001), págs. 199-215. ISSN: 08834237. DOI: 10.1214/ss/1009213726.

[10] Leo Breiman y col. *Classification and regression trees*. Routledge, 1984.

[11] A Broderick y A Patricio. “Chelonia mydas (South Atlantic subpopulation)”. En: *The IUCN Red List of Threatened Species* (2019), págs. 2019-2.

[12] Patrick Buehler y col. “An automated program to find animals and crop photographs for individual recognition”. En: *Ecological Informatics* 50.February 2018 (2019), págs. 191-196. ISSN: 15749541. DOI: 10.1016/j.ecoinf.2019.02.003. URL: https://doi.org/10.1016/j.ecoinf.2019.02.003.

[13] Bruna Calmanovici y col. “I3S Pattern as a mark?recapture tool to identify captured and free-swimming sea turtles: An assessment”. En: *Marine Ecology Progress Series* 589 (feb. de 2018), págs. 263-268. ISSN: 01718630. DOI: 10.3354/meps12483.

[14] Mauricio Cantor y col. “High incidence of sea turtle stranding in the southwestern Atlantic Ocean”. En: *ICES Journal of Marine Science* 77.5 (2020), págs. 1864-1878.

[15] Alice S. Carpentier y col. “Stability of facial scale patterns on green sea turtles Chelonia mydas over time: A validation for the use of a photo-identification method”. En: *Journal of Experimental Marine Biology and Ecology* 476 (2016), págs. 15-21.

ISSN: 00220981. DOI: 10.1016/j.jembe.2015.12.003. URL: http://dx.doi.org/10.1016/j.jembe.2015.12.003.

[16] María Castellano Mendez. "Modelización estadística con Redes Neuronales. Aplicaciones a la Hidrología, Aerobiología y Modelización de Procesos". En: (2009), pág. 161.

[17] Cesar Higinio Menacho Chiok. "Modelos de regresión lineal con redes neuronales". En: *Anales Cientıficos*. Vol. 75. 2. Universidad Nacional Agraria La Molina. 2014, págs. 253-260.

[18] Kyung Suk Choi y col. "In vitro trans-differentiation of rat mesenchymal cells into insulin-producing cells by rat pancreatic extract". En: *Biochemical and Biophysical Research Communications* 330.4 (2005), págs. 1299-1305. ISSN: 0006291X. DOI: 10.1016/j.bbrc.2005.03.111.

[19] Liliana P Colman y col. "Long-term growth and survival dynamics of green turtles (Chelonia mydas) at an isolated tropical archipelago in Brazil". En: *Marine biology* 162.1 (2015), págs. 111-122.

[20] Torch Contributors Copyright 2019. *Documentación de pytorch, GAUSSIANNLLLOSS*. https://pytorch.org/docs/stable/generated/torch.nn.GaussianNLLLoss.html. 2022.

[21] Torch Contributors Copyright 2019. *Documentación de pytorch, HuberLoss*. https://pytorch.org/docs/stable/generated/torch.nn.HuberLoss.html. 2022.

[22] Torch Contributors Copyright 2019. *Documentación de pytorch, SmoothL1Loss*. https://pytorch.org/docs/stable/generated/torch.nn.SmoothL1Loss.html. 2022.

[23] Matthew D. Cross y col. "Pattern-recognition software as a supplemental method of identifying individual eastern box turtles (Terrapene c. carolina)". En: *Herpetological Review* 45.4 (2014), págs. 584-586. ISSN: 0018084X.

[24] Sam S. Cruickshank y Benedikt R. Schmidt. "Error rates and variation between observers are reduced with the use of photographic matching software for capture-recapture studies". En: *Amphibia Reptilia* 38.3 (2017), págs. 315-325. ISSN: 15685381. DOI: 10.1163/15685381-00003112.

[25] Li Deng. "A tutorial survey of architectures, algorithms, and applications for deep learning". En: *APSIPA Transactions on Signal and Information Processing* 3 (2014). ISSN: 20487703. DOI: 10.1017/ATSIP.2013.99.

[26] *Desafío de conocimientos en la nube, Azure Data Scientist.* https://docs.microsoft.com/es-mx/users/cloudskillschallenge/collections/. 2021.

[27] Stephanie Dick. "Artificial Intelligence". En: *Harvard Data Science Review* 1.1 (jul. de 2019). https://hdsr.mitpress.mit.edu/pub/0aytgrau.

[28] S. G. Dunbar y col. "Recognition of juvenile hawksbills Eretmochelys imbricata through face scale digitization and automated searching". En: *Endangered Species Research* 26.2 (2014), págs. 137-146. ISSN: 16134796. DOI: 10.3354/esr00637.

[29] Universidad Nacional de Córdoba Famaf. *Fundamentos básicos del procesamiento de imágenes.* https://www.famaf.unc.edu.ar/~pperez1/manuales/cim/cap2.html. 2022.

[30] Ian Goodfellow, Yoshua Bengio y Aaron Courville. "Deep learning (adaptive computation and machine learning series)". En: *Cambridge Massachusetts* (2017), págs. 321-359.

[31] Yanming Guo y col. "Deep learning for visual understanding: A review". En: *Neurocomputing* 187 (2016), págs. 27-48. ISSN: 18728286. DOI: 10.1016/j.neucom.2015.09.116.

[32] DL Gupta, AK Malviya y Satyendra Singh. "Performance analysis of classification tree learning algorithms". En: *International Journal of Computer Applications* 55.6 (2012).

[33] JiaboO He y col. “Alpha-IoU: A Family of Power Intersection over Union Losses for Bounding Box Regression”. En: *Advances in Neural Information Processing Systems*. Ed. por M. Ranzato y col. Vol. 34. Curran Associates, Inc., 2021, págs. 20230-20242. URL: https://proceedings.neurips.cc/paper/2021/file/a8f15eda80c50adb0e71943adc8015cf-Paper.pdf.

[34] Kaiming He y Jian Sun. “Convolutional Neural Networks at Constrained Time Cost Kaiming”. En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, págs. 5353-5360. ISBN: 1573-2835(Electronic);0091-0627(Print). eprint: arXiv:1412.1710v1. URL: https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/He_Convolutional_Neural_Networks_2015_CVPR_paper.pdf.

[35] Kaiming He y col. “Deep residual learning for image recognition”. En: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2016-Decem (2016), págs. 770-778. ISSN: 10636919. DOI: 10.1109/CVPR.2016.90. eprint: 1512.03385.

[36] Nathan Inkawhich. *FINETUNING TORCHVISION MODELS*. https://pytorch.org/tutorials/beginner/finetuning_torchvision_models_tutorial.html. 2017.

[37] Sergey Ioffe y Christian Szegedy. “Batch normalization: Accelerating deep network training by reducing internal covariate shift”. En: *32nd International Conference on Machine Learning, ICML 2015* 1 (2015), págs. 448-456. arXiv: 1502.03167.

[38] James M. Joyce. “Kullback-Leibler Divergence”. En: *International Encyclopedia of Statistical Science*. Ed. por Miodrag Lovric. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, págs. 720-722. ISBN: 978-3-642-04898-2. DOI: 10.1007/978-3-642-04898-2_327. URL: https://doi.org/10.1007/978-3-642-04898-2_327.

[39] William L Kendall y col. "A multistate open robust design: population dynamics, reproductive effort, and phenology of sea turtles from tagging data". En: *Ecological Monographs* 89.1 (2019), e01329.

[40] Marc Kéry y Michael Schaub. *Bayesian Population Analysis using WinBUGS*. Elsevier, 2012, pág. 538. ISBN: 9780123870209. URL: http://store.elsevier.com/Bayesian-Population-Analysis-using-WinBUGS/Marc-Kery/isbn-9780123870216/.

[41] Diederik P Kingma y Jimmy Ba. "Adam: A method for stochastic optimization". En: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.* (2015).

[42] Meeri Koivuniemi y col. "Photo-ID as a tool for studying and monitoring the endangered Saimaa ringed seal". En: *Endangered Species Research* 30.1 (2016), págs. 29-36. ISSN: 16134796. DOI: 10.3354/esr00723.

[43] Milagros López-Mendilaharsu y col. "Demographic and tumour prevalence data for juvenile green turtles at the Coastal-Marine Protected Area of Cerro Verde, Uruguay". En: *Marine Biology Research* 12.5 (2016), págs. 541-550.

[44] Alfonso Pitarque, Juan Carlos Ruiz y Juan Francisco Roy. "Las redes neuronales como herramientas estadísticas no paramétricas de clasificación". En: *Psicothema* 12.SUPPL. 2 (2000), págs. 459-463. ISSN: 02149915.

[45] Karun K. Rao y col. "Sea Turtle Facial Recognition Using Map Graphs of Scales". En: (2021).

[46] Júlia Reisser y col. "Photographic identification of sea turtles : method description and validation , with an estimation of tag loss". En: *Endangered Species Research* 5 (2008), págs. 73-82. DOI: 10.3354/esr00113.

[47] Nicole Rusk. "Deep learning". En: *Nature Methods* 13.1 (2015), pág. 35. ISSN: 15487105. DOI: 10.1038/nmeth.3707.

[48] Gail Schofield y col. "Investigating the viability of photo-identification as an objective tool to study endangered sea turtle populations". En: *Journal of Experimental Marine Biology and Ecology* 360 (2008), págs. 103-108. DOI: 10.1016/j.jembe.2008.04.005.

[49] Rupesh Kumar Srivastava, Klaus Greff y Jürgen Schmidhuber. "Highway Networks". En: (2015). arXiv: 1505.00387. URL: http://arxiv.org/abs/1505.00387.

[50] Thanchira Suriyamongkol e Ivana Mali. "Feasibility of Using Computer-Assisted Software for Recognizing Individual Rio Grande Cooter (Pseudemys gorzugi)". En: *Copeia* 106.4 (2018), págs. 646-651. ISSN: 0045-8511. DOI: 10.1643/ch-18-101.

[51] Christian Szegedy y col. "Going Deeper With Convolutions". En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Jun. de 2015.

[52] Gabriela M Vélez-Rubio y col. "Marine turtle threats in Uruguayan waters: Insights from 12 years of stranding data". En: *Marine Biology* 160.11 (2013), págs. 2797-2811. ISSN: 00253162. DOI: 10.1007/s00227-013-2272-y.

[53] Gabriela M. Vélez-Rubio y col. "Pre and post-settlement movements of juvenile green turtles in the Southwestern Atlantic Ocean". En: *Journal of Experimental Marine Biology and Ecology* 501 (2018), págs. 36-45. ISSN: 00220981. DOI: 10.1016/j.jembe.2018.01.001.

[54] Miriam A. Zemanova. "More training in animal ethics needed for European biologists". En: *BioScience* 67.3 (2017), págs. 301-305. ISSN: 15253244. DOI: 10.1093/biosci/biw177.

[55] Ning Zhang y col. "Investigation on Performance of Neural Networks Using Quadratic Relative Error Cost Function". En: *IEEE Access* 7 (2019), págs. 106642-106652. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2930520.