UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

RECONOCIMIENTO DE ESPECIES DE PECES APLICANDO EXTRACCIÓN DE CARACTERÍSTICAS MORFOLÓGICAS MEDIANTE FEATURE ENGINEERING Y DEEP LEARNING

POR

Vincenzo Alexo Caro Fuentes

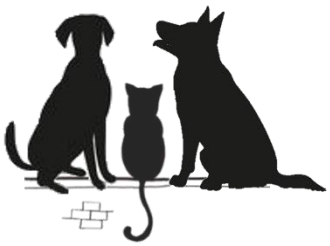
Tesis presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al grado académico de Magíster en Ciencias de la Ingeniería con Mención en Ingeniería Eléctrica

Profesor Guía
Dr. Sebastián Godoy Medel

Profesor Co-Guía
Dr. Mauricio Urbina Foneron

Marzo, 2024
Concepción, Chile

“A mis padres, Ruth y Edgar, a mi hermanito, Matías, y a mi pareja, Ignacio, por su apoyo incondicional y una vida llena de mágicos misterios. A mis hijos peludos, Chimuelo, Nayla y Suter, por su reconfortante compañía.”



AGRADECIMIENTOS

Este trabajo está dedicado a la memoria del profesor Jorge Pezoa, QEPD, quien me encaminó en este mundo del reconocimiento de peces y que fue un pilar fundamental para el profesional en el que me he convertido hoy en día. Tampoco puedo dejar de lado al profesor Sebastián Godoy, a quien agradezco por darme el impulso necesario para continuar cada vez que algo no salía como lo esperado, por sus almuerzos inesperados, por sus consejos de vida, por su confianza y por su paciencia infinita esperando que lograra terminar esta tesis.

También quisiera agradecer a mis familiares y amigos. A mis padres, Ruth y Edgar, quienes siempre me han dado su apoyo incondicional para todo lo que me propongo y que me inspiran a seguir creciendo como persona y como profesional. A mi hermanito, Matías, por siempre estar ahí presente, por apañarme en mis locuras y por ser la mente maestra detrás de los diagramas más bonitos de esta tesis. A mi pareja, Ignacio, por llegar en un momento especial en mi vida, y por hacerme enormemente feliz. Al Ariel, mi amigo y colega de pescados, por apañarme en estos últimos meses mientras terminaba la tesis, por siempre tentarme con juegos nuevos, y por no dejarme caer en la locura gracias a eso.

También agradezco a mis alumnos de los cursos de TIC, por inspirarme cada día para ser un mejor docente, y quienes no salieron perjudicados tras la realización de esta tesis.

Agradezco al Servicio Nacional de Pesca y Acuicultura (Sernapesca), por proporcionar muestras de peces y a SICPA, por su compromiso a la hora de impulsar el uso de la inteligencia artificial en la industria pesquera. Esta investigación fue financiada por la Agencia Nacional de Investigación y Desarrollo (ANID), FONDEF IT20I0032 y ANILLO ACT210073, y por la beca ANID-Subdirección de Capital Humano/Magíster Nacional/2022 - 22221382.

RESUMEN

En esta tesis de magíster se apunta a entregar una solución que contribuya en la incorporación de tecnología y trazabilidad al proceso pesquero llevado a cabo en Chile, respecto de la discriminación automática y en línea de la composición de especies en los desembarques utilizando técnicas de visión computacional y *deep learning*. Adicionalmente, se entregan las bases necesarias para refinar a los modelos de discriminación, incorporando un análisis morfológico y geométrico de las características de los peces para mejorar la clasificación entre las especies de interés que son visualmente parecidas entre sí. La etapa clave del sistema de reconocimiento implementado fue la discriminación de peces, donde se utilizó el modelo YOLOv7 para la detección y clasificación automática de 6 especies pelágicas en tiempo real. Este modelo se adaptó para operar dentro del ambiente industrial en dos plantas pesqueras de la región del Biobío. Posteriormente, en un ambiente de laboratorio y con el objetivo de aumentar la precisión en la identificación de especies problemáticas como la anchoveta, la caballa, el jurel y la sardina común, se experimentó con la detección de *keypoints* utilizando el modelo Keypoint R-CNN para automatizar la extracción de características morfológicas de los peces. Esto condujo al desarrollo de un modelo de clasificación jerárquico que adapta una taxonomía en base a las características más específicas de las especies de peces problemáticas. Los resultados obtenidos probaron ser sobresalientes para ambos ambientes de trabajo. En el ambiente industrial, a pesar de que el sistema se enfrentó a desafíos como la detección en condiciones ambientales variables y una alta aparición de oclusiones entre los peces, se lograron precisiones sobre el 90% para 4 de las 6 especies, logrando un sistema que puede operar en promedio a 54.1 imágenes por segundo, ajustándose a los estándares mínimos requeridos por el personal de Sernapesca para disponer de una herramienta que sirva de apoyo para llevar a cabo su labor. En el entorno de laboratorio, el sistema logró precisiones de hasta 1.00 para predecir especies de peces pequeñas, y 0.98 para predecir especies de peces grandes, demostrando que la identificación de *keypoints* es la clave para enfrentar problemas donde se puede explotar de mejor forma la taxonomía de las especies de interés.

ABSTRACT

This master's thesis aims to provide a solution that contributes to the incorporation of technology and traceability to the fishing process carried out in Chile, regarding the automatic and online discrimination of species composition in landings using computer vision and deep learning techniques. In addition, the necessary bases are provided to refine the discrimination models, incorporating a morphological and geometric analysis of the characteristics of the fish to improve the classification between the species of interest that are visually similar to each other. The key stage of the implemented recognition system was fish discrimination, where the YOLOv7 model was used for the automatic detection and classification of 6 pelagic species in real time. This model was adapted to operate within the industrial environment of two fishing plants in the Biobío region. Subsequently, in a laboratory setting and with the aim of increasing the accuracy in identifying problematic species such as anchovy, mackerel, jack mackerel, and sardine, experiments were conducted with keypoint detection using the Keypoint R-CNN model to automate the extraction of morphological features of the fish. This led to the development of a hierarchical classification model that adapts a taxonomy based on more specific characteristics of the problematic fish species. The results obtained proved to be outstanding for both work environments. In the industrial environment, despite the fact that the system faced challenges such as detection in variable environmental conditions and a high occurrence of occlusions among fishes, precisions of over 90% were achieved for 4 of the 6 species, achieving a system that can operate on average at 54.1 images per second, adjusting to the minimum standards required by Sernapesca inspectors to have a tool that serves as support to carry out their work. In the laboratory environment, the system achieved accuracies of up to 1.00 to predict small fish species, and 0.98 to predict big fish species, demonstrating that the identification of keypoints is the key to facing problems where the taxonomy of the species of interest can be better exploited.

Tabla de Contenidos

LISTA DE TABLAS

LISTA DE FIGURAS

LISTA DE ECUACIONES

CAPÍTULO 1. INTRODUCCIÓN

1.1. Introducción General

A modo de contexto general, en Chile, la pesca y la acuicultura son unas de las principales actividades económicas, ya que contribuye al empleo, al ingreso nacional y al abastecimiento de alimentos. Según la FAO, Chile es uno de los diez países pesqueros más importantes del mundo, con capturas anuales de más de 3 millones de toneladas de peces [1]. Por ende, es de vital importancia para las entidades gubernamentales a cargo, como el Servicio Nacional de Pesca y Acuicultura (Sernapesca), regular constantemente la actividad pesquera artesanal e industrial, para así garantizar el cumplimiento de las cuotas de extracción de pesca, y fiscalizar otras actividades ilícitas, como la pesca de especies vedadas, el subreporte de los desembarques para una o varias especies, o la pesca ilegal de especies protegidas; todo con el fin de proteger los recursos hidrobiológicos [2]. Sin embargo, el método de fiscalización actual utilizado es precario, tanto por el lado del personal, que no da abasto para controlar adecuadamente todos los desembarques a nivel nacional, disminuyendo los porcentajes efectivos de fiscalización; como por el lado del método en sí utilizado, el cual se encuentra limitado desde el punto de vista técnico y humano, volviéndolo un proceso actualmente no trazable, ineficiente, y carente de tecnología.

En respuesta a lo anterior, como parte del trabajo conjunto entre Sernapesca, SICPA y la Universidad de Concepción, se han propuesto y desarrollado diversas técnicas para tecnologizar este proceso de fiscalización. Esto considera las memorias de título de Diego Ramírez [3] y Alonso Díaz [4], así como la memoria de título realizada por quien les escribe [5], y también la tesis de maestría de Diego Ramírez [6] respaldada por el proyecto FONDEF 17I10184, donde se adoptaron técnicas de visión computacional (CV) y *machine learning* (ML) para el proceso completo de la reconocimiento e identificación

de peces, gracias en su mayor medida a la gran disponibilidad de datos y avances en hardware computacional (GPUs, TPUs), y al auge en que esto se deriva para la utilización de modelos de aprendizaje automático basado en redes neuronales profundas. Inicialmente, el desarrollo de los sistemas de visión se enfocó a partir de la aplicación de imagenología híperespectral para la clasificación de especies pelágicas, mediante la obtención de una huella espectral para cada especie de pescado de interés, tanto en el espectro visible como en el infrarrojo cercano [3], [4], [6]. Luego, durante desarrollos posteriores, se descartó el uso de la información hiperespectral, considerando únicamente un análisis espacial RGB de las imágenes utilizando una arquitectura de redes neuronales convolucionales de dos etapas, contando con una primera etapa de detección, donde se reconocen y se segmentan los peces presentes en una imagen, para luego entrar a una segunda etapa de clasificación, donde a cada pescado segmentado se le asigna una etiqueta correspondiente a su especie en función de la base de datos disponible [5].

Sin embargo, el sistema aún presenta fallos, siendo muy susceptible ante cambios en el entorno (iluminación, ruido, vibración), a la superposición de muestras (completas o parciales), o también, a la aparición de elementos ajenos a los conocidos por los modelos en ambas etapas. Debido a esto, se dificulta tanto el proceso de construcción de la base de datos (por el número y la calidad de las muestras representativas con los que se debe contar, junto con el alto tiempo que toma en realizar la tarea), como también el proceso de entrenamiento y validación de los modelos (por el sesgo que impone la calidad de la base de datos sobre la extracción de características relevantes), donde se genera un bajo número de detecciones, se influencia la aparición de falsas detecciones y, en consecuencia, se afecta la precisión total obtenida por el sistema.

Inspirado en esta problemática, se espera que en este trabajo se enfrenten dichas complicaciones realizando un estudio más profundo sobre cómo se está realizando cada etapa del proceso, siendo el foco principal, que los modelos presten más atención a los

detalles específicos de los peces que a los de su propio entorno (el cual se supone conocido). En función de aquello, las principales contribuciones del trabajo serán desarrollar una solución multietapa, en base a algoritmos de *deep learning* (DL), que sea capaz de:

- mejorar la calidad del reconocimiento de peces y su capacidad de respuesta ante perturbaciones externas, como cambios de iluminación, ruido u oclusiones.
- extraer características morfológicas, geométricas o espaciales en especies pelágicas en base a su información taxonómica estándar de manera automatizada.
- adaptar estas características morfológicas, geométricas o espaciales para incorporar una mayor cantidad de información relevante para cada especie y mejorar su clasificación.

Siguiendo la estrategia ya mencionada, se espera lograr una *macro-average precision* (MP) mínima alcanzable del 80% en la predicción de las especies de interés, considerando que en resultados previos obtenidos con el sistema de reconocimiento se han logrado métricas MP del 93% para especies de peces grandes, y del 85% para especies de peces pequeñas (revisar Sección 3.3.1A ..1 para mayores detalles), siendo imprescindible introducir el menor error posible dentro de un proceso crítico como lo es la fiscalización, llegando a ser útil para apoyar a este proceso en la práctica.

1.2. Estado del Arte

El análisis del estado del arte relacionado con el problema de reconocimiento de peces y la disponibilidad de tecnología para llevarlo a cabo se muestra a continuación. Este se decidió dividir en dos tópicos principales:

- Respecto del análisis de características morfológicas u otras características propias de peces con inteligencia artificial (AI).

- Respecto del tratamiento del ruido u otras variables adversas que afectan a los sistemas de reconocimiento.

1.2.1 Análisis de Características de Peces con IA

Con el paso de los años, en el marco del área de la acuicultura y la pesca, la implementación de la Inteligencia Artificial (IA) ha abierto nuevas puertas hacia el entendimiento y la gestión de las especies pelágicas de peces. Según propone Barbedo, la capacidad de la IA para analizar grandes conjuntos de datos visuales, tales como imágenes y videos, ha sido crucial para la identificación de especies y la observación de su comportamiento en su hábitat natural, especialmente en condiciones de visibilidad subóptima y en profundidades inaccesibles para los observadores humanos [7]. La IA, a través de técnicas como el aprendizaje profundo y las redes neuronales convolucionales, ha permitido a los científicos superar varios desafíos asociados con el estudio de especies pelágicas, proporcionando medios para analizar comportamientos, patrones migratorios y características morfológicas de estas especies de manera no invasiva y a gran escala [8].

De manera más específica, es posible encontrar estudios que se enfocan principalmente en la utilización de características morfológicas para alimentar los sistemas de reconocimiento. Por ejemplo, Duran *et al*. realizan un análisis espectral de los peces, aplicando una técnica de aprendizaje reforzado sobre un set de datos obtenido mediante el uso de varios métodos de espectroscopia multimodal para la clasificación de imágenes en función de ciertas reglas, como el rango de sensibilidad y varianza por clase [9]. También, en Spampinato *et al*. se dedicaron a combinar diferentes características de texturas y formas sobre 10 especies de peces, logrando precisiones promedio del 92% al entrenar clasificadores en base a análisis de discriminante [10]. Similar al problema anterior, Coelho-Caro *et al*. presentan un método para reconocer automáticamente mejillones en una imagen basado en la extracción de características morfológicas, alcanzando precisiones de hasta 95% [11].

También es posible encontrar aplicaciones que utilizan *deep learning* como técnica para generar un sistema de reconocimiento de peces a partir de sus características morfológicas. En Pornpanomchai *et al*. implementan un sistema de cinco etapas para el reconocimiento de peces en una imagen [12]. Dentro de estas etapas, se destaca la tercera donde realizan una extracción de características que, a diferencia de otras propuestas, se basa únicamente en la incorporación de las características físicas más relevantes de un pez, como las relaciones de largo y ancho y el promedio de cada canal de color, las cuales sirven de entrada hacia el modelo de clasificación escogido (uno basado en distancia Euclideana, y otro basado en redes neuronales). Respecto del último, los autores alcanzan hasta un 99% de precisión para un total de 30 especies de pescado.

En Alsmadi *et al*. realizan un trabajo similar, cuyo propósito fue reconocer patrones de interés en peces utilizando una combinación de diferentes características en base a la localización de puntos relevantes, *landmarks* o *keypoints*, entre las que se destacan: mediciones de distancias anatómicas (relativas y angulares); análisis de textura del color; y otros parámetros geométricos [13]. Las características mencionadas luego sirven de entrada para una red neuronal, la cual logra hasta un 86% de precisión para un total de 20 familias de peces. Gowda *et al*. también se muestran la importancia del espacio de color a la hora de clasificar imágenes, obteniendo diferencias significativas en la precisión del modelo cuando se escogen espacios de color particulares en función del tipo de imagen del problema en cuestión [14].

Ahondando un poco más en la línea de detección de *keypoints*, se encontraron diversos estudios que se enfocan principalmente en incorporar este paradigma en el reconocimiento de peces. Principalmente, en Saitoh *et al.* se explora el reconocimiento de imágenes de peces basado en *keypoints*, utilizando diversos grupos de características, incluyendo características geométricas (relativas y angulares), un modelo de tipo "*Bags of visual words model*" (BoVW), y características a partir de texturas. Para la clasificación de

sus datos emplearon un modelo *Random Forest* probando todas las combinaciones posibles entre los tres grupos de características escogidas, alcanzando una precisión de hasta 96.3% en el mejor de los casos, proporcionando un enfoque eficiente para la identificación de 129 especies de peces en imágenes [15]. En Saleh *et al*. se presenta un modelo basado en redes neuronales convolucionales (CNN) denominado "*Mobile fish landmark detection network*" (MFLD-net), el cual está específicamente diseñado para ser ligero y eficiente en dispositivos móviles y embebidos, demostrando ser altamente efectivo en la detección automática de *keypoints* en imágenes de la especie Barramundi, incluso con conjuntos de datos relativamente pequeños [16]. El modelo, pese a no utilizar pesos preentrenados, alcanza una precisión promedio (AP) de 96.7%. Por otro lado, Suo *et al.* introduce un sistema de monitoreo de ecología de peces que utiliza una arquitectura de red neuronal profunda para detectar peces utilizando el modelo Faster R-CNN, incorporando una etapa adicional para estimar con gran precisión la longitud de cada muestra detectada mediante la localización de *keypoints* específicos, utilizando para ello un modelo basado en la arquitectura *Stacked Hourglass*, mostrando ser una herramienta eficiente y precisa para el monitoreo de ecología de peces en línea [17]. Finalmente, Lin *et al.* proporcionan una exploración preliminar sobre la estimación de la postura de los peces utilizando diversos modelos para la detección de objetos con *bounding boxes* rotativos, abordando los desafíos de la detección de peces en tiempo real en entornos subacuáticos y logrando presiones de hasta 90.61% [18]. Para la etapa de estimación de postura, los autores se centran particularmente en el modelo de dos etapas, DeepPose, alcanzando un porcentaje de *keypoints* correctos (PCK) del 97.8%.

Relacionado con tecnologías capaces de segmentar automáticamente diferentes especies de peces en una imagen, Yu *et al*. presentan un modelo para segmentar y medir peces utilizando Mask R-CNN [19], un modelo de segmentación por instancias que genera máscaras dentro de la imagen para cada instancia de algún objeto en particular para el cual este haya sido entrenado [20]. La solución de los autores nace en respuesta a la actual forma de medir ciertas características morfológicas de los peces, la cual es compleja y se

realiza de forma manual. Ellos, entonces, adaptan el modelo de Mask R-CNN para la tarea en particular, obteniendo como resultado los peces segmentados junto con sus medidas más relevantes, con un error menor del 2.8% para imágenes sin fondo, y menor al 3% para imágenes en condiciones más adversas. Desde otra perspectiva, la propuesta de Liang *et al*. presenta PolyTransform, un modelo pensado para identificar todos los objetos observables en una imagen mediante la asignación de una máscara o polígono a cada objeto encontrado [21]. La técnica utilizada consta de 2 partes importantes: la asignación de polígonos que mejor representen a un objeto, explotando su naturaleza geométrica; y la aplicación de un *deforming model* para ajustar el polígono al contorno real del objeto, incluso cuando este se encuentra dividido por partes al ser ocluido por otro. De esta forma, los autores logran resultados sobresalientes con una precisión promedio (AP) de 40.1, superando en 8.1 puntos a los resultados alcanzados por Mask R-CNN entrenado bajo un contexto similar.

Otras técnicas que se concentran en el entendimiento de contornos, formas o ambientes se pueden encontrar en [22], [23]. Particularmente, Husain *et al*. presentan una estrategia para comprender de mejor forma el entorno visto por robots de interiores, combinando características semánticas a nivel de píxel y características geométricas para mejorar la segmentación de múltiples clases de objetos [22]. Hayder *et al*., por su parte, dirigen su investigación a la corrección de los candidatos de objetos encontrados por modelos de detección utilizando como base para ello la información sobre su contorno o forma, prediciendo máscaras que van más allá del alcance de su *bounding box* respectivo [23].

1.2.2 Ruido u Otras Variables Adversas

Dejando de lado las características morfológicas de los peces, es posible enfrentarse a otro problema común relacionado a cómo afecta el entorno (fondo) de las imágenes en el rendimiento de los sistemas de reconocimiento ¿Existirán diferencias si se prepara, por ejemplo, una base de datos bajo el agua y otra base de datos sobre el agua involucrando un mismo espécimen? Las referencias disponibles aportan bastante información sobre dicho punto, porque es uno de los aspectos más influyentes a la hora de diseñar un sistema de reconocimiento (ya sea o no de peces), determinando los alcances que se deben tomar en cuenta y las consideraciones que se pueden tomar para su buen y correcto funcionamiento.

En particular para imágenes que se toman bajo el agua, existen numerosas técnicas de ML que se han aplicado para entrenar modelos de alto rendimiento bajo esas condiciones. Si bien no es ni será el foco de este trabajo, es importante tomarlo en cuenta debido a las condiciones adversas que se pueden presentar en estos entornos subacuáticos como, por ejemplo, el ruido inherente del ambiente, objetos extraños de carácter incontrolable, entre otras condiciones adversas que pueden extrapolarse a condiciones que se generen en sistema sobre el agua.

En Zhang *et al*. es posible encontrar uno de estos sistemas, donde aportan con una solución para aquellos problemas donde el ruido del fondo de una imagen no es despreciable, evitando que modelos tradicionales para el reconocimiento de peces puedan proporcionar resultados suficientemente precisos [24]. Se propone el uso de la técnica por aprendizaje de características discriminantes (DFL), con la idea de generar un entrenamiento que contraste las características más relevantes (*discriminative features*) para objetos de diferentes clases versus las características comunes entre objetos de una misma clase, estableciendo límites de decisión que mantengan separadas las características para elementos de distintas clases, y, al mismo tiempo, siendo capaz de ignorar el ruido mediante una selección de parámetros que representen el nivel de atención sobre el objeto de

interés. El método propuesto alcanza precisiones de hasta ~99%, superando otras estrategias del estado del arte disponibles a la fecha.

En Lee *et al*. se presenta un sistema para detectar, realizar seguimiento y reconocer peces en una imagen, particularmente al interior de un acuario [25]. Respecto de la detección, esta primero se realiza en base al cálculo previo del fondo de referencia que se mantendrá estático en todo momento, sirviendo como base posterior para la remoción de fondo y otros efectos propios de la iluminación del acuario, como sombras. La clasificación posterior se realiza con el algoritmo de *k-Nearest Neighbors* (K-NN), alcanzando precisiones de hasta un 97%.

A su vez, en Jianping *et al*. se muestra la respuesta a problemas como atenuación de la iluminación o distorsión en entornos acuáticos submarinos, donde el reconocimiento de peces se dificulta enormemente [26]. Los autores presentan un método para realizar reconocimiento de peces preciso y robusto. Para ello, diseñan PCANet, un modelo de 2 etapas que se encarga de extraer características abstractas a partir de imágenes de peces, logrando precisiones de hasta 95.18%. En Tamou *et al*., por su parte, se presenta una técnica para reconocimiento de peces utilizando una base de datos disponible de peces únicos en ambientes submarinos para entrenar un extractor de características basado en AlexNet y un clasificador de tipo *support vector machine* (SVM) adicional [27]. Se alcanzan presiones de hasta 99.45%.

Bajo condiciones aún más adversas, Qin *et al*. proponen una solución para el reconocimiento de peces captados desde una cámara ubicada en el océano [28]. El proceso consta de una etapa de extracción del fondo mediante una descomposición matricial por matrices dispersas, cuya salida pasa por un extractor de características en base al análisis de componentes principales (PCA) y hashing binario para las capas no-lineales, e histogramas para las capas de pooling. Luego, se utiliza un kernel de pooling espacial piramidal (SPP) para extraer información que no depende de la posición de las muestras. Finalmente,

se utiliza un clasificador SVM para la etapa final de clasificación, alcanzando precisiones de hasta 98.64%. Se da una buena explicación del funcionamiento de las capas intermedias del modelo. En Chuang *et al*. se presenta una alternativa que permite extraer las características de peces a partir de una técnica no-supervisada de aprendizaje que se basa en el mapeo de las características físicas de los peces antes de ser entregados a un clasificador [29]. El método también introduce una técnica para evaluar la clase de una imagen ambigua, introduciendo el concepto de clasificación parcial cuando hay indecisión al momento de asignar una etiqueta determinada. (el método asigna etiquetas por etapa, en lugar de una única etiqueta al final del procesamiento). Se alcanzan presiones de hasta un 98.4%.

Otras alternativas similares se pueden encontrar en el trabajo de Li *et al*., donde utilizan una técnica para reconocimiento de peces bajo el agua que busca acelerar el comportamiento general de Faster R-CNN, reemplazando su arquitectura original por una etapa de generación de regiones, la cual entrega propuestas de alta calidad que resulta en mayores precisiones y, principalmente, menor tiempo de procesamiento [30]. Posteriormente también presentan otra alternativa que utiliza Fast R-CNN para el reconocimiento de peces bajo el agua [31], y en Shortis *et al*. se presenta un compilado de técnicas utilizadas para la identificación, medición y conteo de peces [32]. Con el propósito de automatizar el proceso, en el trabajo se estudian todo tipo de técnicas para detectar, identificar, medir, contar y realizar seguimiento en base a procesos o métodos automatizados.

Dejando de lado los sistemas bajo el agua, se han presentado otras soluciones en respuesta a otro tipo de problemáticas, como la superposición de objetos (iguales o diferentes), o que presentan otras situaciones alternativas para el reconocimiento de objetos en imágenes.

Esto es el caso de Jia *et al*., donde nuevamente se utilizan técnicas de aprendizaje profundo para detectar y segmentar frutas superpuestas mediante Mask R-CNN, adaptando el modelo para su uso por un robot cosechador [33]. El método fue mejorado para facilitar el reconocimiento de áreas superpuestas dentro de una imagen, logrando preci-

siones sobre el 97% durante el testeo del modelo. Como contraparte, en Yu *et al*. se muestra otra propuesta para mejorar el rendimiento de un robot cosechador de frutillas en base al uso de Mask R-CNN, donde se presentan algunos problemas a la hora de diferenciar objetos sencillos para los humanos, pero complejos para las máquinas [34]. Su propuesta modifica el segmento original destinado a la extracción de características, y lo reemplazan por uno que genera una buena relación performance/tiempo de cálculo, alcanzando precisiones de precisión de hasta el 95%. Si bien el modelo se comporta bien para frutas superpuestas o con variadas condiciones de iluminación, este presenta ciertos fallos en la predicción al confundir hojas u otros elementos con frutillas (por tener contextura o color similar).

Por otro lado, en Luo *et al*. se presenta un método para el reconocimiento y conteo de peces [35]. Para ello, divide el algoritmo en 3 etapas: eliminación de ruido, reconocimiento de peces entre regiones con y sin peces mediante un modelo de formas a partir de estadística (SSM), y finalmente un reconocimiento fino de peces (se separa por categorías en base a lo que se muestra en la imagen) junto con un conteo respectivo. Finalmente, en Vannoy *et al*. se presenta una propuesta para reducir el tiempo de etiquetado manual que requiere una base de datos para identificar peces al interior de imágenes [36]. En este se demuestra el impacto que tiene evaluar un mismo modelo de aprendizaje supervisado para dos sets de datos tomados en temporadas no consecutivas, y además ingresa el problema sobre contar con datos sobre peces ubicados en distintas regiones dentro de una imagen, alterando las características comunes (como iluminación o posiblemente, distorsión), y la distribución uniforme de información relevante, afectando el performance del sistema completo.

1.3. Discusión

La revisión bibliográfica realizada muestra algunas de las aplicaciones desarrolladas hasta la fecha sobre sistemas de reconocimientos de peces en una gran variedad de entornos, demostrando que sí existe tecnología capaz de aportar nueva información para desenvolverse dentro de la problemática que se presenta en este trabajo.

Es importante destacar que, si bien ya se han creado soluciones en respuesta a la detección y clasificación de especies de peces utilizando información morfológica, el foco de esta investigación recae en determinar u obtener aquella información que tenga mayor relevancia para llevar a cabo una discriminación robusta. En ese sentido, el trabajo presentado en [15] es el que más se acerca a este objetivo, puesto que todas las características utilizadas para entrenar los modelos de clasificación se extraen a partir de la referencia entregada por *keypoints*, los cuales se escogieron de manera general (pero representativa) para acomodarse a todas las familias de peces involucradas en el proceso. Sin embargo, como los mismos autores indican, su principal falencia es que estos *keypoints* deben ser seleccionados manualmente, donde una ligera perturbación en su posición puede generar resultados inesperados. En [20] la estrategia se resuelve de manera inversa, puesto que es el mismo modelo de detección el que se encarga de generar una predicción y segmentar las partes del cuerpo de interés de los peces, como la ubicación de ojos o la cola, presentando el inconveniente de que la base de datos se vuelve mucho más engorrosa de elaborar al tener que segmentar inicialmente de manera manual todas las partes de interés, pero siendo muy robusto ante las perturbaciones del fondo u otros objetos desconocidos para el modelo. En este trabajo, la solución presentada será un punto intermedio de ambas estrategias, donde inicialmente un modelo de detección será el encargado de asignar automáticamente los *keypoints* para cada pez utilizando imágenes con fondos aleatorios para mitigar la perturbación del fondo, y luego un modelo de clasificación utilizará la información morfológica extraída para generar la predicción final.

Desde el punto de vista de las arquitecturas utilizadas por los distintos autores, la mayoría de las soluciones encontradas modifican partes de la arquitectura interna de otros modelos ya existentes para adecuarlos al problema que intentan resolver, basándose principalmente en técnicas como *transfer learning* para liberar el diseño de los modelos de la alta carga computacional que requiere realizar un entrenamiento desde cero, resultando ventajoso respecto de la dimensión que debe tener la base de datos (no requiriendo tantos datos para lograr resultados adecuados), y entregando flexibilidad a la hora de combinar ciertas etapas de interés de uno o más modelos para lograr un objetivo más específico. Siguiendo este lineamiento, el sistema de reconocimiento que se diseñará en este trabajo contemplará múltiples etapas, cada una con una tarea y un modelo específicos para resolverla, particularmente la etapa de clasificación, donde las características morfológicas se dividirán en diferentes ramas de manera jerárquica para generar una etiqueta de clasificación de la misma forma que operan las claves dicotómicas para diferenciar una especie animal de otra.

1.4. Hipótesis

La hipótesis principal de investigación es que el sistema de reconocimiento de especies de peces puede mejorar su *macro-average precision* en reconocer especies, logrando valores superiores o igual al 80% si se incorpora información morfológica relevante estandarizada en base a la taxonomía de los peces, y otros parámetros geométricos y/o espaciales.

Durante el proceso de desarrollo de la propuesta, se busca responder preguntas, como:

1. ¿Qué características mínimas debe cumplir la base de datos para representar correctamente la problemática? ¿Cuáles son las mejores variables de entrada para los modelos?

2. ¿Cómo se puede enseñar a los modelos a reconocer especies de peces por sus características propias, y no en función de su entorno?

3. ¿Qué tipo de arquitecturas de redes neuronales profundas son las más apropiadas? ¿Deben considerarse estrategias multietapa o de una única para detectar y clasificar cada especie de pez?

4. ¿Qué tipo de técnicas o modelos pueden contrarrestar de mejor forma características poco controlables del entorno industrial, como cambios de iluminación, cambios de velocidad durante la toma de imágenes, o el solapamiento constante de las muestras?

1.5. Objetivos

1.5.1 Objetivo General

El objetivo general de este trabajo consiste en desarrollar y aplicar una estrategia supervisada para la detección de peces, centrada en la extracción de características robustas y la segmentación por instancias. Esta estrategia busca incorporar información morfológica, geométrica y espacial de las especies objetivo durante las fases de entrenamiento e inferencia de los modelos, cuyo enfoque se centrará en mejorar la clasificación entre especies de gran interés a nivel industrial, como la anchoveta y la sardina, así como el jurel y la caballa, principalmente, por ser parte de los recursos pesqueros extraídos en la región del Biobío.

1.5.2 Objetivos Específicos

- Reestructurar la base de datos existente para entrenar modelos de DL de naturaleza supervisada en función de las especies de peces y su información morfológica disponibles.
- Analizar y adaptar algoritmos de detección de objetos que extraigan parámetros morfológicos, geométricos o espaciales para el reconocimiento de peces de manera automatizada.

- Analizar y adaptar algoritmos de clasificación jerárquica que incorporen parámetros morfológicos, geométricos o espaciales para la clasificación específica de especies de peces problemáticas.
- Analizar y adaptar estrategias que mitiguen el efecto del entorno durante la etapa de entrenamiento de los modelos.
- Evaluar y contrastar los resultados obtenidos mediante la utilización de las curvas de pérdida y las métricas de desempeño estándar para los modelos de detección y clasificación. Esto incluye: *precision* (P), *macro-average precision* (MP), *mean average precision* (mAP), *percentage of correct keypoints* (PCK), *recall* (R) y *F-score*.

1.6. Alcances y Limitaciones

En el contexto del trabajo desarrollado en esta tesis, se describen las etapas mínimas requeridas para desarrollar un sistema de reconocimiento de especies pelágicas, tanto en ambiente industrial como en laboratorio, y se entrega una revisión de los algoritmos y modelos aplicados en cada etapa.

La adquisición de los datos fue realizada con dos versiones de un sistema optomecánico (pórtico), los cuáles cuentan a su vez con dos modelos de cámaras IP Hikvision diferentes, ambas operando en el espectro visible (RGB). Adicionalmente, la disponibilidad de muestras de peces (potencialmente muertas) se diferencia en función del ambiente de trabajo. Para la industria, esto se limita a las seis especies visualizadas, registradas y exhaustivamente revisadas a través de los pórticos, mientras que, para el laboratorio, esto se limita a las especies que pueden ser proporcionadas por algún ente externo (Sernapesca), siendo solo cuatro de estas de interés para esta investigación.

Todo el proceso de programación fue realizado en lenguaje Python, utilizando bibliotecas y software de terceros de código abierto. Esto incluye, pero no se limita a, mo-

delos preentrenados en plataformas de desarrollo como Tensorflow y PyTorch, herramientas para etiquetar bases de datos y aplicar aumento de datos y la plataforma para el entrenamiento de modelos en la nube, Google Colab [37].

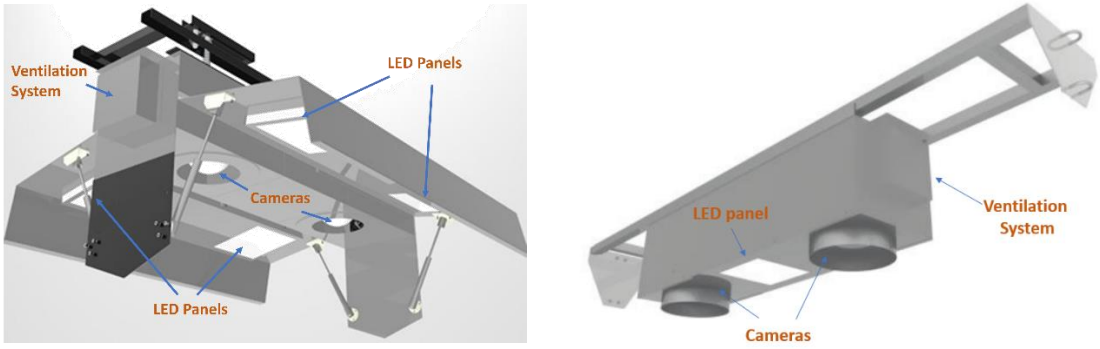
CAPÍTULO 2. MATERIALES Y MÉTODOS

2.1. Hardware

Para implementar el sistema de reconocimiento de peces, ya sea instalado sobre una cinta transportadora en plantas pesqueras o en ambiente de laboratorio, se diseñaron los dispositivos optomecánicos mostrados en la Figura N° 2.1. El componente principal del dispositivo es un pórtico que consta de una estructura de acero inoxidable, dos cámaras RGB, paneles LED sin parpadeo (*anti-flicker*) para evitar problemas visuales de parpadeo en las cámaras, y un sistema de ventilación.

Particularmente, se diseñaron e implementaron dos versiones del pórtico. La primera versión cuenta con cuatro paneles LED distribuidos alrededor de dos cámaras IP Hikvision de 2MP (modelo DS-2CD2721G0-IZS), que operan en el espectro visible. La conexión a las cámaras se puede establecer mediante protocolos HTTP y RTSP, y la gestión y el seguimiento se pueden realizar utilizando el software de control de Hikvision [38]. Este software permite la extracción de transmisiones de video con resolución variable, tasa de captura, tasa de bits y otras características modificables por el usuario. El modelo seleccionado para las cámaras del diseño del segundo pórtico es Hikvision DS-2CD1653G0-IZ. Este modelo de 5MP puede funcionar en condiciones de poca luz y tiene la opción de ajustar el campo de visión para acercar o alejar cómodamente, según el tamaño de la especie a clasificar. Esta versión de la puerta tiene un único panel LED ubicado entre las dos cámaras. La primera versión de diseño del pórtico fue instalada en la planta industrial pesquera Orizon, Coronel, mientras que la segunda versión de diseño del portón se instaló en la planta Blumar, San Vicente; ambas ubicadas en la zona centro-sur de Chile. El apartado a de la Figura N° 2.1 muestra una representación del pórtico instalado en la planta Orizon, mostrando la estructura de acero inoxidable montada sobre una viga soporte, junto con el sistema de iluminación con 4 paneles LED y el sistema de ventilación.

En contraste, el apartado b muestra una vista esquemática del portón instalado en la planta Blumar, destacando la presencia de un único panel LED colocado en la parte inferior de la estructura.



(a) (b)

Figura N° 2.1 Esquemáticos de los pórticos instalados en ambiente industrial. a) Pórtico Orizon. b) Pórtico Blumar. Fuente: [Elaboración Propia]

Adicionalmente, todas las pruebas de laboratorio fueron realizadas utilizando el pórtico de Blumar, antes de su instalación en la planta.

2.2. Captura de Imágenes (Bases de Datos)

Para el entrenamiento de los modelos es necesario crear una base de datos con capturas de las especies que se desea que el sistema reconozca. Las capturas de las especies se realizaron tanto en laboratorio como en la industria gracias a los pórticos, y, adicionalmente, se utilizaron fuentes externas para complementar el entrenamiento de algunos modelos con imágenes extraídas de bases de datos encontradas en internet. En total, se construyeron 6 bases de datos diferentes, las cuales se detallan en las subsecciones siguientes.

2.2.1 Bases de Datos B01 y B02

Para las bases de datos construidas a partir de imágenes en la industria, la captura de datos se realizó de forma remota estableciendo una conexión directa desde un ordenador externo a las cámaras instaladas en cada pórtico. Para la planta Orizon, las capturas consistieron en imágenes con una resolución de 1920x1080 píxeles, tomadas a una velocidad de fotogramas fija de 8 imágenes por segundo (FPS). Esta velocidad de fotogramas fue sincronizada con la velocidad del movimiento de los peces sobre la correa transportadora, para así evitar posibles duplicaciones de un mismo pez en múltiples fotogramas. Los datos fueron capturados desde abril de 2021 a julio de 2021, abarcando sesiones durante la mañana, la tarde y la noche. De manera similar, para la planta Blumar, las imágenes capturadas tenían una resolución de 1920x 1080 píxeles, pero fueron capturadas a una velocidad de fotogramas menor de 1 FPS debido a la velocidad más lenta de la correa transportadora de la planta. Las capturas se realizaron desde noviembre de 2022 hasta marzo de 2023, contemplando también días y horarios variados. Teniendo en cuenta el gran volumen de datos, la captura de datos para ambas plantas no se realizó de forma continua durante largos períodos de tiempo para minimizar los requisitos de almacenamiento.

Las muestras consideradas se dividieron por cada planta, con las especies anchoveta (*Engraulis ringens*), jurel (*Trachurus murphyi*), caballa (*Scomber japonicus*) y sardina (*Strangomera bentincki*) para la planta Orizon, y jurel, caballa, jibia (*Dosidicus gigas*) y sierra (*Thyrsites atun*) para la planta Blumar. La cantidad final de muestras etiquetadas para cada especie, para cada planta, se resume en la Tabla N° 2.1. En adelante, la base de datos de la planta Orizon se denominará **B01**, mientras que para la planta Blumar, esta se denominará **B02**.

Tabla N° 2.1 Distribución de imágenes capturadas por especie para las plantas Orizon y Blumar.

Clase	Cantidad de especies Planta Orizon (B01)	Cantidad de especies Planta Blumar (B02)
Anchoveta	549	-
Caballa	139	764
Jibia	-	22
Jurel	1339	7377
Sardina	480	-
Sierra	-	177

Finalmente, se aplicaron técnicas de *data augmentation* (ver Anexo A.5.1) para mejorar la capacidad de detección de peces de los modelos, especialmente para reconocer la fauna acompañante, que tiene una baja probabilidad de ocurrencia en un desembarque pesquero común. Las técnicas de *data augmentation* utilizadas incluyen volteo horizontal y vertical, rotaciones máximas de 30 grados, cambios de brillo y contraste, adición de ruido y desenfoque.

2.2.2 Base de Datos B03

Para la base de datos construida a partir de imágenes en el laboratorio, la captura de datos se realizó de forma local estableciendo una conexión directa desde un ordenador a las cámaras instaladas del pórtico. En este caso, las capturas también consistieron en imágenes con una resolución de 1920x1080 píxeles, pero esta vez sin movimiento.

Las muestras consideradas incluyen las especies anchoveta, jurel y sardina, cuya cantidad final se resume en la Tabla N° 2.2. Debido a la baja cantidad de muestras por especie, también se aplicaron técnicas de *data augmentation* para incrementar digitalmente las variantes posibles para cada imagen, incluyendo volteo horizontal y vertical, rotaciones máximas de 30 grados, cambios de brillo y contraste, adición de ruido y desenfoque. En adelante, la base de datos de laboratorio se denominará **B03**, la cual presenta variantes con múltiples peces en una imagen y con un único pez por imagen, dependiendo del requerimiento de los modelos.

Tabla N° 2.2 Distribución de imágenes capturadas por especie para laboratorio.

Clase	Cantidad de especies (sin data augmentation)	Cantidad de especies (sin data augmentation) B03
Anchoveta	21	172
Jurel	15	112
Sardina	27	279

2.2.3 Bases de Datos B04, B05 y B06

Adicionalmente, también se utilizaron fuentes externas desde donde se descargaron nuevas imágenes para complementar el análisis llevado a cabo en el ambiente de laboratorio al momento de entrenar los modelos.

Tal es el caso de las muestras de caballa, las cuales son necesarias para la mayoría de los entrenamientos presentados en la Sección 3.4, pero que no estaban disponibles al momento de preparar la base de datos **B03**, y que tampoco pudieron ser reutilizadas directamente de las bases de datos **B01** o **B02** debido a su baja resolución y a la baja frecuencia de ejemplares enteros. Para solventar esta falta, se utilizaron las imágenes disponibles en [39], la cual es una base de datos, perteneciente a la plataforma Kaggle[1], que contiene 20 especies del mar mediterráneo (entre ellas, la caballa). El resumen de las muestras extraídas a partir de esta base de datos se muestra en la Tabla N° 2.3, donde nuevamente fue necesario aplicar *data augmentation* para incrementar digitalmente las variantes posibles para cada imagen. En adelante, esta base de datos se denominará **B04**.

Tabla N° 2.3 Distribución de imágenes descargadas de caballa.

Clase	Cantidad de especies (sin data augmentation)	Cantidad de especies (sin data augmentation) B04
Caballa	78	161

[1] Plataforma de competencia de ciencia de datos y una comunidad en línea de científicos de datos y profesionales del aprendizaje automático de Google LLC.

Otra de las bases de datos construidas contempla la utilización de imágenes de peces y otros objetos dentro de un contexto lo más general posible, lo cual es necesario para entrenar el modelo de la Sección 3.4.1. Las imágenes en esta oportunidad fueron descargadas directamente de internet, siguiendo la distribución que se muestra en la Tabla X. Particularmente, las imágenes de objetos *random* se dividieron entre: agua, correas transportadoras, crustáceos, escobas, mesas, metal, moluscos, pasto, personas y plástico; elementos a fines a lo que uno se puede encontrar en el ambiente industrial.

Tabla N° 2.4 Distribución de imágenes descargadas de peces y objetos random.

Clase	Cantidad de especies (sin data augmentation) B05
Fish (*random*)	800
No Fish (*random)*	800

También se cuenta con una segunda base de datos descargada de la plataforma Kaggle, la cual consta cerca de 4000 imágenes para 468 especies de peces diferentes [40]. A partir de esta base de datos, se extrajeron 908 imágenes repartidas entre peces con y sin manchas, según se muestra en la Tabla N° 2.5, necesarias para el entrenamiento del modelo presentado en la Sección 3.4.5A. En adelante, esta base de datos en su versión reducida se denominará **B06**.

Tabla N° 2.5 Distribución de imágenes capturadas por esp.

Clase	Cantidad de especies (sin data augmentation) B06
Peces con manchas	475
Peces sin manchas	433

2.3. Algoritmos de Detección y Clasificación

Para llevar a cabo la tarea de detectar y clasificar automáticamente diversas especies de peces a partir de imágenes, se aplicaron técnicas avanzadas de procesamiento de imágenes y reconocimiento de patrones que pueden identificar características relevantes

(formas, tamaños, texturas, colores, etc.) mediante la utilización de herramientas de inteligencia artificial y *deep learning*.

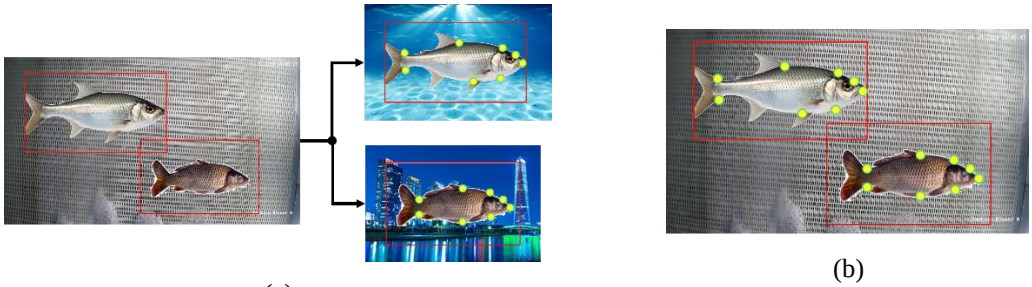
Para la tarea de detección, se utilizaron algoritmos pertenecientes al paradigma de detección de objetos y dos de sus ramas principales: segmentación por instancias y detección de *keypoints*. Para la tarea de clasificación, se utilizaron algoritmos de clasificación multiclase y de clasificación jerárquica. Para esta última, el análisis se dividió entre tres variantes: clasificación jerárquica local, clasificación jerárquica plana y clasificación jerárquica multidimensional.

Particularmente para la detección de *keypoints*, existen dos modalidades que pueden aplicarse: *top-down* y *bottom-up*. En general, la primera entrega resultados más precisos y robustos, pero más lentos; mientras que la segunda tiende a alcanzar mayores velocidades de inferencia, pero a costa de una menor precisión al momento de asignar los *keypoints* [18].

La modalidad de detección *top-down* se puede apreciar de forma sistemática en el apartado a de la Figura N° 2.2. En primer lugar, el modelo realiza una representación global de la imagen, identificando los objetos de interés junto con sus características mediante la detección de objetos. En segundo lugar, se mejoran las predicciones generadas por el modelo a través de una representación local de cada *bounding box* detectado, lo que permite ajustar la ubicación precisa de cada *keypoint*.

Por otro lado, la modalidad de detección bottom-up se evidencia en el apartado b de la Figura N° 2.2. En este enfoque, el modelo comienza por detectar *keypoints* individuales en la imagen sin tener en cuenta una estructura global, resaltando regiones potenciales de objetos o características relevantes de forma local. Luego, en una segunda fase, el modelo busca establecer conexiones y relaciones globales entre los *keypoints* detectados, asociando cada grupo de *keypoints* al objeto más cercano que mejor los represente.

Considerando que bajo una modalidad *bottom-up* es extremadamente fácil dividir los *keypoints* incorrectamente cuando hay objetos muy cercanos entre sí (como se da en el caso de los peces), en esta tesis solo se probaron modelos bajo una modalidad *top-down*. Esto también aplica para los modelos de segmentación por instancias, donde se utilizan máscaras en lugar de *keypoints.*



(a) (b)

Figura N° 2.2 Modalidades de detección de keypoints. a) top-down. b) bottom-up. Fuente: [Elaboración propia].

A continuación, se entrega una descripción de cada tipo de algoritmo o modelo principal utilizado a lo largo de esta tesis.

2.3.1 YOLO

El algoritmo YOLO (*You Only Look Once*) es una técnica de código abierto de una única etapa para la detección de objetos en imágenes y videos en tiempo real a partir de CNN, siendo concebido como parte del estado del arte para este paradigma desde el lanzamiento de su primera versión, en 2015.

Su funcionamiento general se describe en la Figura N° 2.3. Para llevar a cabo la detección, primero se divide la imagen en una cuadrícula con celdas de tamaño predefinido (imagen izquierda). Luego, en cada celda se predicen los posibles candidatos de objeto con su *bounding box* y su confianza respectivos (imagen central). Finalmente, se eli-

minan los *bounding boxes* con una confianza bajo un cierto umbral, y se aplica un algoritmo de supresión no máxima (NMS) para eliminar las detecciones redundantes o duplicadas (imagen derecha).

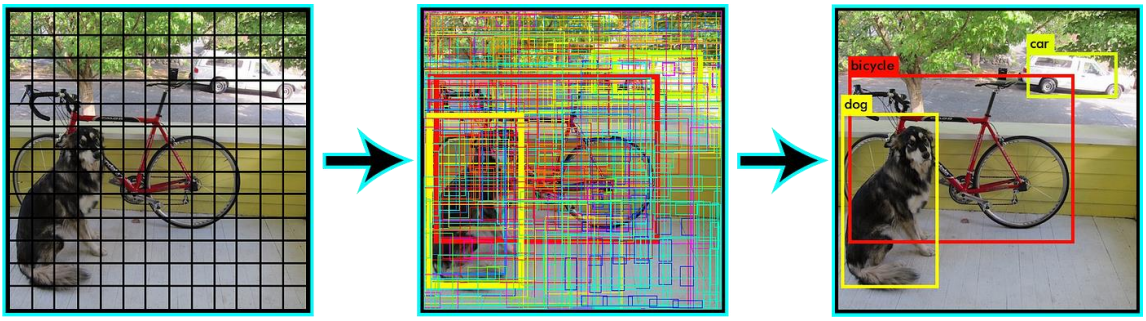


Figura N° 2.3 Proceso de detección de objetos llevado a cabo por YOLO. Fuente: [41]

A lo largo del tiempo, se han desarrollado distintas variantes de YOLO que han ido mejorado tanto su precisión como su la velocidad de detección. Para esta tesis, se entrenaron principalmente modelos utilizando YOLOv7 [42], pero también se mencionan algunos resultados obtenidos utilizando una versión más antigua, YOLOv4 [43].

2.3.2 Mask R-CNN

Mask R-CNN es una extensión del algoritmo Faster R-CNN (*Region-based Convolutional Neural Network*) que agrega la capacidad de segmentación de instancias a la detección de objetos en una modalidad compuesta por 3 etapas [19]. Este algoritmo permite identificar no solo la presencia y la ubicación de objetos en una imagen mediante *bounding boxes*, sino también segmentar píxeles individuales formando máscaras correspondientes a cada instancia de objeto detectado.

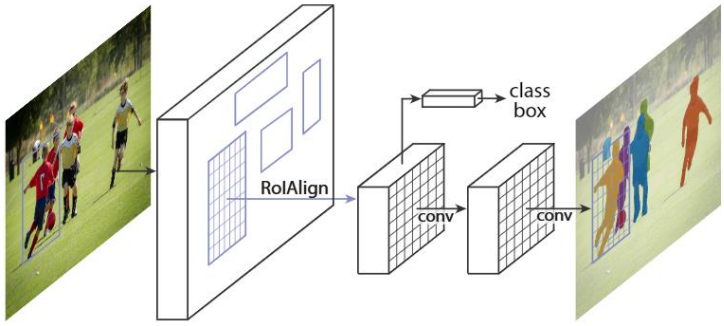


Figura N° 2.4 Arquitectura de MaskR-CNN. Fuente: [19]

El proceso de detección y segmentación de objetos en Mask R-CNN se visualiza de forma general en la Figura N° 2.4. Para llevar a cabo esta tarea, la primera etapa del modelo utiliza un *backbone*[2] de extracción de características basado en CNNs para generar propuestas de regiones de interés (RoIs) que pueden contener un objeto[3]. Luego, el modelo integra una operación denominada RoIAlign, la cual, a grandes rasgos, se encarga de alinear las características extraídas para distintos RoI colindantes y combinarlos en función del *ground truth* de la entrada. Finalmente, el modelo divide su flujo de operaciones en dos etapas: la primera es una etapa de clasificación que refina los *bounding boxes* generados por el modelo mediante un algoritmo NMS, y la segunda es una etapa de CNN que se encarga de predecir una máscara de segmentación binaria de forma precisa para cada RoI propuesto.

Si bien la segmentación lograda por este modelo es sumamente precisa, llegando a formar parte del estado del arte desde su lanzamiento, en 2017, el agregar una etapa extra al modelo original de Faster R-CNN lo vuelve uno de los modelos detección de objetos más lentos de esta gama.

---

[2] Las capas o el modelo encargados de aprender los patrones de características de las imágenes, desde los más simples a los más abstractos.
[3] Esta etapa también se conoce como Region Proposal Network (RPN).

2.3.3 Keypoint R-CNN

A. Fundamentos del Modelo

Keypoint R-CNN es un modelo que pertenece al paradigma de detección de *keypoints*, el cual es una extensión tanto del modelo Faster R-CNN como Mask R-CNN [44]. Este algoritmo permite identificar tanto *bounding boxes* como los *keypoints* asociados a las zonas relevantes de cada objeto detectado, siendo en realidad un subconjunto más acotado de las máscaras o polígonos entregados por Mask R-CNN.

Una de las primeras versiones de la arquitectura del modelo Keypoint-RCNN se muestra en la Figura N° 2.6. En primer lugar, se utiliza una red de tipo RPN para generar las propuestas de regiones que pueden contener un objeto. En segundo lugar, el modelo acopla una cuarta etapa a la arquitectura de Faster R-CNN, la cual se encarga de predecir los *keypoints* en cada región de interés propuesta. En caso de que la salida del modelo sea un *heatmap*, como el que se muestra en la Figura N° 2.5, esta entrega cuán probable un píxel en una imagen puede contener un *keypoint* (subconjunto de la máscara predicha del objeto). En caso de que la salida del modelo sean directamente las coordenadas (x, y) de cada *keypoint*, típicamente el modelo integra una capa adicional de regresión para transformar las características extraídas en la ubicación aproximada de los *keypoints* en la imagen para cada objeto detectado (subconjunto del polígono o contorno predicho del objeto).

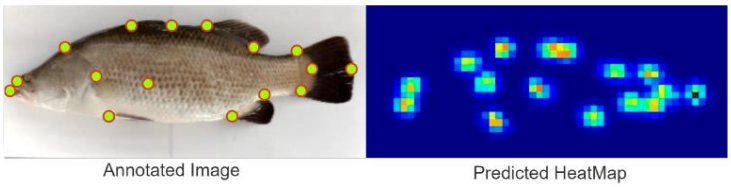


Figura N° 2.5 Ejemplo de anotaciones de keypoints en un pez y el respectivo heatmap generado por un modelo de detección de keypoints. Fuente: [16]

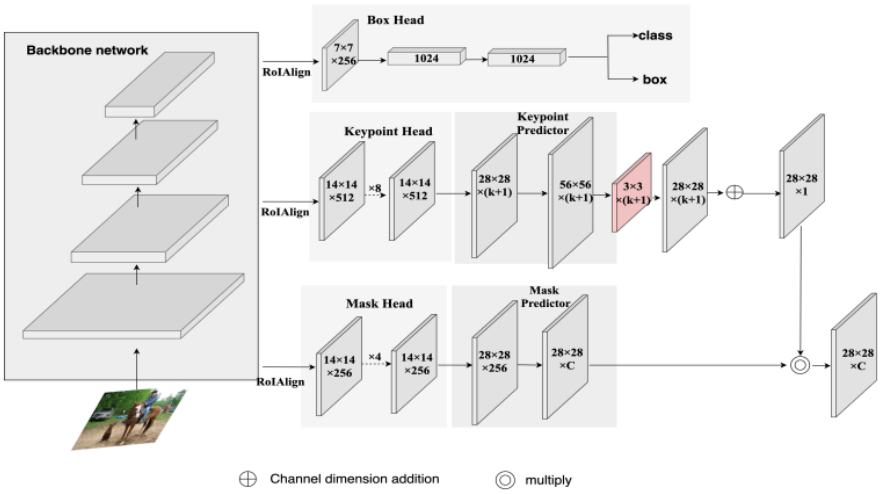


Figura N° 2.6 Arquitectura de Keypoint R-CNN. Fuente: [45]

B. Limitaciones Técnicas

Este paradigma de detección en particular presenta limitaciones técnicas relacionadas con la capacidad de los modelos para trabajar con un número diferente de *keypoints* para las clases de interés, lo cual es particularmente problemático en aplicaciones donde las diferentes clases pueden requerir naturalmente un número variable de *keypoints*, como es común en el estudio de la taxonomía de peces u otras áreas de la biología marina.

Desde el punto de vista de la distribución de clases, en vista de la limitación de los modelos de *keypoints*, todas las bases de datos utilizadas durante el entrenamiento de Keypoint R-CNN se simplificaron para trabajar con una única clase, “peces”, asignando a cada muestra la misma cantidad de *keypoints* (8), según fueron definidos en la Sección 2.5.1.

2.3.4 Clasificadores Multiclase

Los clasificadores multiclase son algoritmos de *machine learning* diseñados para asignar una categoría o clase a un grupo de datos con características en común, ya sean datos unidimensionales o multidimensionales, como imágenes. Para los modelos que fueron entrenados y testeados a lo largo de esta tesis, se utilizaron tanto clasificadores del estado del arte, como *Support Vector Machine* (SVM), árboles de decisión y k-*Nearest Neighbors* (KNN); como también clasificadores basados en redes neuronales, específicamente en redes neuronales completamente conectadas (FCNN) y redes neuronales convolucionales, como VGG16 [46]. Respecto de estas últimas, estas redes son particularmente adecuadas para manejar grandes volúmenes de datos y extraer una gran variedad de características (desde las más simples a las complejas o abstractas), lo que las hace ideales para tareas relacionadas con el procesamiento de imágenes y visión computacional.

A continuación, se describen brevemente los dos tipos de redes neuronales utilizados, junto con el uso de modelos pre-entrenados basados en CNNs.

A. Redes Neuronales Completamente Conectadas

Las redes de tipo FCNN son una clase de redes neuronales artificiales que consisten en múltiples capas de nodos (neuronas[4]), donde cada nodo de una capa está conectado a todos los nodos de la capa siguiente (ver Figura N° 2.7), siguiendo una estructura de tipo *Feedforward*. A este tipo de capas se les conoce comúnmente como capas *fully connected* o capas densas, las cuales pueden ser de tres tipos:

- Capa de entrada: La capa que recibe un vector de características unidimensionales de entrada.

[4] La unidad básica de cálculo de una red neuronal, donde se aplica una combinación lineal de sus entradas y luego aplica una función de activación.

- Capas ocultas: Una o más capas donde cada neurona está conectada a todas las neuronas de la capa adyacente.
- Capa de Salida: La capa que produce el resultado final del modelo en base a una función de activación en particular (típicamente *softmax*), que puede ser una clasificación multiclase o una regresión.

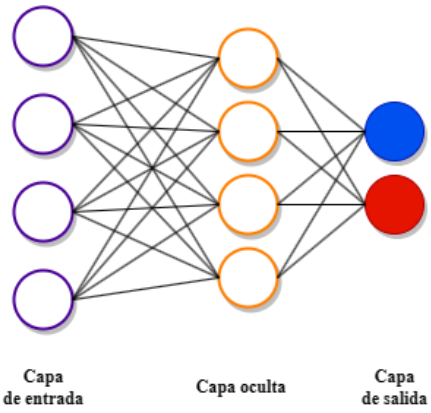


Figura N° 2.7 Ejemplo de operación de capas fully connected. Fuente: [Elaboración propia]

Particularmente, la función de activación de tipo *softmax* proporciona como resultado un conjunto de probabilidades que reflejan la pertenencia de la entrada a las posibles clases del modelo, lo cual se utilizó para todos los modelos de clasificación multiclase a lo largo de esta tesis.

B. Redes Neuronales Convolucionales

Las redes de tipo CNN son una categoría de redes neuronales que han demostrado ser extremadamente efectivas en tareas relacionadas con el procesamiento de imágenes, aunque también se utilizan en otras aplicaciones. Las CNN utilizan operaciones de convolución para procesar datos en forma de matrices, lo que les permite aprovechar de mejor forma la espacialidad de los datos para detectar patrones locales y características espaciales en las imágenes. Un ejemplo de esta operación de convolución, entre otras de las capas

comúnmente utilizadas con las redes neuronales convolucionales, se muestran en el Anexo A.4.

C. Modelos Pre-Entrenados

De forma similar a como YOLO o Mask R-CNN forman parte del estado del arte en modelos de detección de objetos, también es posible encontrar modelos del estado del arte para la clasificación de imágenes basados en CNNs, como ResNet50 [47] y VGG16 (generalmente utilizados como *backbones* de los modelos de detección). Estos modelos fueron desarrollados y entrenados en grandes conjuntos de datos, como ImageNet [48], y han demostrado una capacidad excepcional para extraer características y patrones relevantes a partir imágenes. Al emplear estos modelos preentrenados, se aprovecha el conocimiento adquirido en tareas de ámbito general, lo que permite una inicialización más efectiva de los pesos de una red para tareas de ámbito más específico. Este enfoque se conoce como *transfer learning*, donde las capas iniciales de los modelos, que generalmente capturan características visuales básicas, se mantienen, mientras que las capas finales se adaptan o reentrenan para la clasificación específica de múltiples clases en un dominio particular. Esta técnica no solo mejora la precisión de la clasificación, especialmente cuando se dispone de datos limitados, sino que también reduce significativamente el tiempo y los recursos computacionales necesarios para llevar a cabo los entrenamientos.

A lo largo de esta tesis, se utilizaron exclusivamente modelos pre-entrenados basados en la arquitectura de VGG16 para llevar a cabo la clasificación multiclase de imágenes de peces. Esta elección nació como resultado de la prueba de validación llevada a cabo por Sernapesca (ver Sección 3.3.1A ..1), donde se compararon al menos cinco tipos diferentes de modelos de clasificación pre-entrenados, siendo VGG16 el que logró los mejores resultados en la mayoría de los casos.

2.3.5 Clasificadores Jerárquicos

Los Clasificadores Jerárquicos son un enfoque de *machine learning* que se utiliza para manejar problemas de clasificación con múltiples clases que tienen una estructura jerárquica o de árbol previamente definida, donde cada nodo representa una categoría y las ramas representan las relaciones jerárquicas entre las categorías. Esta metodología permite una clasificación más precisa que los clasificadores multiclase convencionales, ya que cada nodo del árbol se especializa en la detección de un conjunto particular de características.

Respecto de esta última afirmación, cabe notar que la estructura de los árboles de decisión es muy similar a un modelo de clasificación de tipo jerárquico, siendo su principal diferencia el método de elección y ordenamiento de los nodos del árbol, los cuales se escogen de manera recursiva para lograr la mejor separación entre las clases en función de una métrica (como la entropía). Sin embargo, de manera similar a lo que ocurre con las capas de extracción de características de los modelos pre-entrenados, esta modalidad de elección de los nodos se comporta como una caja negra difícil de manipular, por lo que no es trivial reconocer qué nodo o qué características en particular es necesario modificar para lograr un mejor rendimiento del modelo. Por este motivo, la utilización de los árboles de decisión directamente como modelo de clasificación final para el sistema de reconocimientos de peces no es viable.

La clasificación jerárquica puede ser de dos tipos: "*top-down*" (de arriba hacia abajo), donde la clasificación comienza desde la raíz del árbol y se mueve hacia abajo hasta llegar a la categoría correcta, y "*bottom-up*" (de abajo hacia arriba), donde la clasificación comienza desde las hojas y se mueve hacia arriba hasta encontrar la categoría raíz. Para esta tesis, se usará principalmente la clasificación jerárquica de tipo "*top-down*", la cual se orientó en tres ramas distintas en base al análisis llevado a cabo en [49]: clasificación jerárquica local por nodo padre, clasificación jerárquica plana y clasificación jerárquica multidimensional.

A. Clasificador Jerárquico Local por Nodo Padre

El clasificador jerárquico local por nodo padre es un enfoque en la clasificación jerárquica que se centra en que cada nodo del árbol jerárquico (excepto las hojas) tiene asociado un clasificador que se entrena para distinguir entre sus nodos hijos directos, sin considerar otras clases o nodos intermedios evaluados a lo largo de la jerarquía.

Siguiendo el diagrama de la Figura N° 2.8, cuando se realiza una nueva predicción sobre un dato, el proceso comienza en la raíz del árbol y el clasificador asociado a ese nodo decide cuál de sus hijos es la categoría más probable. Luego, el proceso se repite, moviéndose hacia abajo en la jerarquía y utilizando los clasificadores locales en cada nodo, hasta llegar a un único nodo hoja que proporciona la clasificación o etiqueta final.

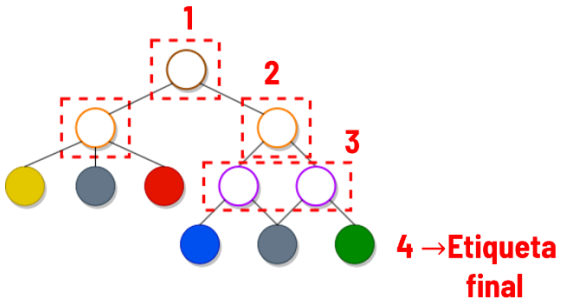


Figura N° 2.8 Árbol de clasificación jerárquica local por nodo padre. Los cuadros punteados denotan a todos los minimodelos que realizan una clasificación. Fuente: [Elaboración propia]

Este enfoque tiene la ventaja de descomponer un problema de clasificación multi-clase potencialmente complejo en varios problemas de clasificación más simples y manejables. Además, al utilizar clasificadores locales, el método puede ser más robusto frente a la variabilidad de los datos, ya que cada clasificador se entrena para distinguir entre un número menor de clases. Sin embargo, una desventaja potencial es que los errores en las decisiones de los clasificadores en los niveles superiores de la jerarquía no pueden corregirse en los niveles inferiores, lo que podría propagar los errores a través de las predicciones finales (inconsistencia).

B. Clasificador Jerárquico Plano

Un clasificador jerárquico plano es un enfoque de clasificación donde todas las clases o etiquetas son consideradas independientes entre sí y no se establecen relaciones de subordinación o jerarquía entre ellas, comportándose, en esencia, como los algoritmos de clasificación tradicionales. Típicamente, todo el proceso de clasificación se realiza entrenando un único clasificador multiclase, donde el modelo solo se encarga de clasificar los nodos hoja de la jerarquía. Sin embargo, en esta tesis se presenta una estrategia combinada entre en clasificador jerárquico local y el clasificador jerárquico plano tradicional. Siguiendo el diagrama presentado en la Figura N° 2.9, el proceso de clasificación reutiliza todas las etiquetas intermedias generadas por los nodos padre que realizan una clasificación (círculos de color sin relleno). Luego, estas etiquetas se combinan en un único vector de características (similar a una capa *flatten* de una red neuronal), y son entregadas al clasificador multiclase (círculos de color con relleno) para generar la etiqueta de clasificación final.

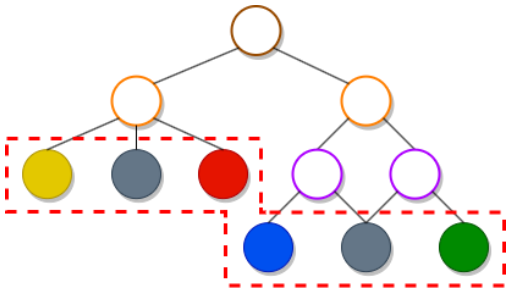


Figura N° 2.9 Ejemplo de árbol de clasificación jerárquica plano. El cuadro punteado representa a un único clasificador multiclase. Fuente: [Elaboración propia]

Si bien el modelo jerárquico plano tradicional es comúnmente utilizado en problemas de clasificación multiclase y multietiqueta donde las relaciones entre las clases no son evidentes o no aportan información adicional significativa para el proceso de clasificación, la idea de la variante propuesta es mostrar cómo la combinación de todas las etiquetas intermedias puede ayudar al modelo a crear relaciones más flexibles entre las clases, versus una clasificación donde la jerarquía entre las clases sí podría ser explotada de

forma más ordenada para mejorar la precisión del modelo. Además, a diferencia del clasificador jerárquico local, esta variante no sufre del problema de inconsistencia gracias a que una clasificación intermedia errónea se puede compensar internamente en el modelo con el resto de las clasificaciones correctas.

C. Clasificador Jerárquico Multidimensional

Un clasificador jerárquico multidimensional es un tipo de modelo de clasificación que utiliza una estructura jerárquica para realizar decisiones de clasificación en múltiples niveles o dimensiones. En este enfoque, en lugar de realizar una clasificación para una única ruta o rama del árbol, el modelo toma decisiones de clasificación en varios niveles de la jerarquía, tomando en consideración la probabilidad predicha por cada nodo de cada rama del árbol y con ello considerar a todas las ramas como posibles caminos para un mismo dato de entrada, lo cual ayuda a mejorar la precisión y eficiencia del modelo, especialmente en casos donde las categorías son desbalanceadas o cuando algunas decisiones de clasificación son más críticas que otras.

Siguiendo el diagrama de la Figura N° 2.10, cuando se realiza una nueva predicción sobre un dato, este debe pasar por todos los clasificadores de la jerarquía, generando con ello las probabilidades predichas por cada clasificador para todos sus nodos hijo. Luego, la decisión final de clasificación se toma en base a la rama que obtenga la mayor probabilidad. De las variantes propuestas en [49], se eligió el método basado en el producto de probabilidades. De esta forma, el producto de probabilidades se realiza en cada rama de la jerarquía, eligiendo la rama que logra el mayor producto de todos. Para el ejemplo, esto ocurre para la rama que termina en el círculo ●, con una probabilidad final de 0.56.

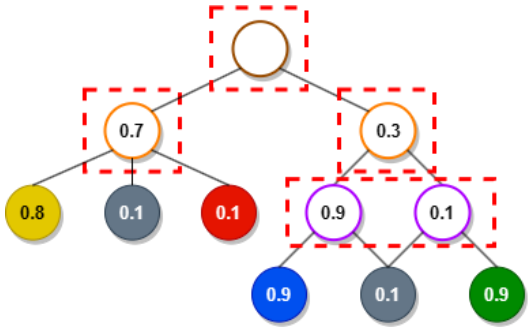


Figura N° 2.10 Árbol de clasificación jerárquica multidimensional. Los cuadros punteados denotan a todos los minimodelos que realizan una clasificación. Fuente: [Elaboración propia]

2.3.6 Métricas de Desempeño

Para la evaluación de los modelos de detección y clasificación entrenados a lo largo de esta tesis, se utilizaron las métricas de desempeño: precision (P), recall (R), F1, mean-average precision (mAP), macro-average precision (MP), mean absolute error (MAE) y percentage of correct keypoints (PKC); las cuales se describen con mayor detalle en el Anexo A.7.

2.4. Etiquetado de Imágenes

El etiquetado es un proceso clave a la hora de entrenar de manera supervisada modelos de *deep learning* que trabajan con imágenes. Dentro del paradigma de detección de objetos, este proceso tiene por objetivo tanto asociar categorías o clases a cada objeto de interés presente en una imagen, como también entregar información respecto de su localización mediante diversos tipos de etiquetado, entre los que se encuentra el *bounding box*, el más común de todos, y también las máscaras o *keypoints*, propios de paradigmas más complejos. A continuación, se muestra una breve explicación para los tres tipos de etiquetado utilizados.

2.4.1 Bounding Box

Un *bounding box*, o caja delimitadora, es un rectángulo que se utiliza para delimitar la ubicación y el tamaño de un objeto de interés en una imagen. Es comúnmente utilizado en todas las tareas de detección de objetos, incluyendo la primera etapa de los modelos de segmentación por instancias y detección de *keypoints*. Este rectángulo se define típicamente mediante cuatro coordenadas: las coordenadas del vértice superior izquierdo (x_{min}, y_{min}) y las coordenadas del vértice inferior derecho (x_{max}, y_{max}), aunque también puede definirse en función de su ancho (x_{max} - x_{min}) y de su largo (y_{max} - y_{min}). La Figura N° 2.11 muestra en ejemplo de este tipo de anotación.

Figura N° 2.11 Ejemplo de anotación con bounding box. Fuente: [Elaboración propia]

2.4.2 Polígonos

En el contexto del etiquetado de imágenes, los polígonos se utilizan para trazar contornos precisos alrededor de objetos para complementar la información entregada por un *bounding box*, siendo capaz de definir una máscara que segmente completamente un objeto de su fondo. Típicamente, se utilizan en tareas de detección de objetos relacionadas con segmentación por instancias o segmentación semántica[5]. La Figura N° 2.12 muestra en ejemplo de este tipo de anotación.

[5] Algoritmo de deep learning que asocia y agrupa a cada píxel presente en una imagen para una etiqueta o categoría en particular, sin ser capaz de diferenciar entre varias instancias de un mismo objeto.

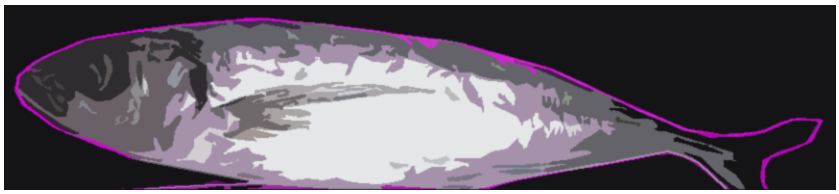


Figura N° 2.12 Ejemplo de anotación por máscara o polígono. Fuente: [Elaboración propia]

2.4.3 Keypoints

Tal y como hace referencia su nombre, los *keypoints* son puntos específicos en una imagen que representan zonas de interés con características visualmente distintivas de algún objeto en particular. Típicamente, se utilizan en tareas de detección de *keypoints*, seguimiento (*tracking*) de objetos o estimación de poses humanas. La Figura N° 2.13 muestra en ejemplo de este tipo de anotación.

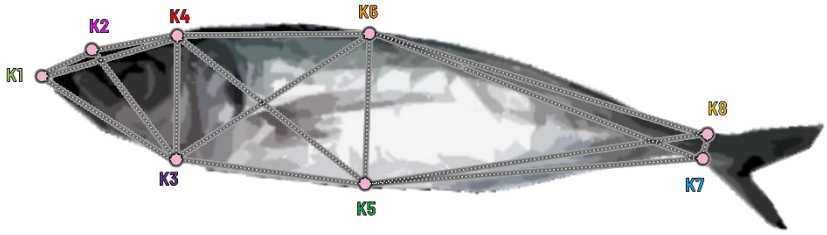


Figura N° 2.13 Ejemplo de anotación con keypoints. Fuente: [Elaboración propia]

2.4.4 Formato de Etiquetado

Para trabajar con alguno de los tipos de etiquetado o anotaciones ya descritos, es necesario adoptar un formato o estructura de datos estandarizada que se mantenga consistente a la largo de todas las anotaciones realizadas en las imágenes. Para esta tesis, se utilizaron dos formatos en concreto: el formato COCO (*Common Object in Context*) y el formato propio de YOLO.

El formato COCO forma parte de un estándar ampliamente utilizado en el área de visión computacional y detección de objetos. Es fácil de adaptar para una gran cantidad

de modelos, lo cual permite el desarrollo de entrenamientos y la generación de métricas de desempeño de manera más consistente y eficiente. Un ejemplo de la estructura de este formato se muestra en el Anexo A.1, donde en un único archivo se incluye principalmente información sobre la ubicación de los objetos (a través de *bounding boxes*, máscaras y/o *keypoints*), las categorías seleccionadas para los distintos objetos y la ubicación de las imágenes que fueron anotadas.

Respecto del formato YOLO, este es en esencia mucho más simple que el formato COCO, donde las anotaciones se realizan generalmente en un archivo de texto simple para cada imagen, donde cada línea del archivo representa un objeto detectado en la imagen correspondiente. Un ejemplo de este formato se muestra en el Anexo A.2, donde se incluye primero el identificador de la clase del objeto, luego las coordenadas del *bounding box* normalizadas respecto de la dimensión original de la imagen, y finalmente las coordenadas de la máscara o polígono del objeto, también normalizadas.

2.4.5 Herramientas de Etiquetado

Para llevar a cabo el etiquetado de las imágenes, se utilizaron dos plataformas en concreto que son compatibles con al menos uno de los formatos de anotaciones.

La primera fue Roboflow[6] [50], la cual permite gestionar bases de datos para la preparación de modelos de clasificación, detección de objetos y segmentación por instancias; tanto en formato COCO como en formato YOLO. Las principales características de su versión gratuita radican en la incorporación del modelo *Segment Anything* (SAM) [51], implementado por Facebook AI Research (FAIR), el cual es capaz de generar automáticamente la máscara más representativa de un objeto solamente con un *click*; y también en la incorporación de *data augmentation* en línea (hasta un máximo de 3 veces del tamaño

[6] Servidor de inferencia fácil de usar y listo para producción para la visión artificial que admite la preparación de bases de datos y la implementación de muchas arquitecturas de modelos populares y modelos personalizados.

de la base de datos original). La Figura Anexo A.1 muestra una visual de la plataforma en pleno proceso de etiquetado de una imagen de jureles. Esta plataforma fue utilizada para generar la primera versión de las bases de datos **B01**-**B04**.

La segunda plataforma utilizada fue COCO Annotator [52]. Esta plataforma, si bien no cuenta con las mismas características principales que Roboflow, permite gestionar bases de datos para modelos de detección de objetos únicamente en formato COCO, extendiendo sus herramientas de etiquetado para la detección de *keypoints* (no disponible en Roboflow). La Figura Anexo A.2 muestra una visual de la plataforma en pleno proceso de etiquetado de una imagen de anchovetas. Esta plataforma fue utilizada para generar la versión con *keypoints* de las bases de datos **B03** y **B04**.

2.5. Taxonomía de peces

La taxonomía es la ciencia que se dedica a la clasificación de los seres vivos, basándose en sus relaciones naturales y características compartidas. En el caso particular del estudio de los peces, la taxonomía es esencial para comprender la vasta biodiversidad que existe dentro de este grupo de vertebrados acuáticos. La anatomía de los peces juega un papel crucial dentro de la taxonomía, puesto que cada especie tiene características anatómicas únicas que las distinguen unas de otras. Estas características pueden ser de tipo morfológicas, donde se incluyen la forma y posición de las aletas, la estructura de las escamas, la forma del cuerpo, entre otras; y también pueden ser de tipo merísticas (cuantificables), como el número de aletas, espinas, radios o escamas. Para los fines de esta tesis, el análisis se enfocará únicamente en las características morfológicas para llevar a cabo la identificación de los peces.

2.5.1 Box Truss y Elección de Keypoints

Para seleccionar las características morfológicas, se hizo uso de un protocolo geométrico denominado *box truss* (ver Figura N° 2.14), según se describe en [53], página 11. Este es un protocolo que garantiza una cobertura sistemática de la forma de un pez y que

dirige de manera exhaustiva y redundante su configuración en base a puntos de referencia o *keypoints*. La denominación particular como *truss* se debe a que se utiliza para superar ciertas limitaciones en la morfometría, permitiendo la descomposición de un individuo en formas geométricas simples y compuestas (triángulos, principalmente), lo cual es útil para describir diferencias de forma entre poblaciones al estudiar las distorsiones globales o locales en la red o esqueleto del *truss*.

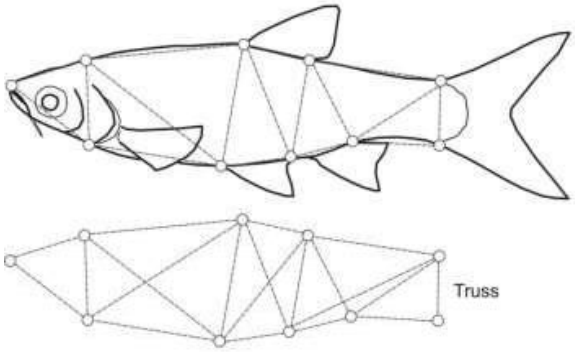


Figura N° 2.14 Mediciones comunes entre puntos clave para construir patrones que cuantifiquen la varianza morfométrica entre especies. Fuente: [53]

A partir del *box truss*, es posible reconstruir una muestra de forma o de distancia utilizando los *keypoints* como coordenadas cartesianas. Luego, estás muestras pueden ser promediadas o estandarizadas a uno o más tamaños de referencia comunes, seguidos de una reconstrucción de la forma o distancia real utilizando una medida corporal o un método de calibración estándar. Además, la red del *box truss* permite encontrar patrones de alometría[7] o diferencias de forma dentro de un grupo de individuos, la cual se ve beneficiada por el uso de medidas cruzadas redundantes dentro de la misma red.

[7] Cambios de la dimensión relativa de las partes corporales, correlacionados con los cambios en el tamaño total.

Para el análisis morfológico que fue realizado en esta tesis, se consideraron 8 *keypoints* que describen de la manera más general posible a las especies de peces mostradas en la Sección 2.5.2. Esta decisión se tomó en base a que no todas las características diferenciadoras entre una especie u otra se puede apreciar correctamente a partir de una imagen al estar estas muertas y fuera del agua, particularmente, la posición de las aletas dorsales o ventrales. En función de esta consideración, los *keypoints* escogidos se muestran en la Figura N° 2.15, tomando en cuenta la siguiente distribución:

- K_1: extremo anterior de la boca [54].
- K_2: proyección del extremo posterior de la boca [54].
- K_3: margen anterior del istmo[8] [55].
- K_4: nuca (borde antero-dorsal de la cabeza) [54].
- K_5: borde ventral más bajo del cuerpo [54].
- K_6: borde dorsal más alto del cuerpo [54].
- K_7: borde posterior dorsal del pedúnculo caudal[9] [42] [55].
- K_8: borde posterior ventral del pedúnculo caudal [42] [55].

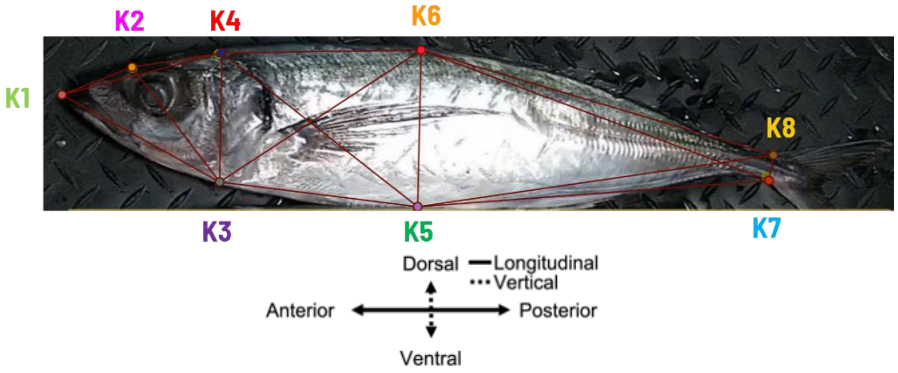


Figura N° 2.15 Box truss con la localización de los 8 keypoints escogidos para una muestra de jurel. Fuente: Adaptado de [54].

La elección de estos *keypoints* en particular se describe a continuación:

[8] El área estrecha de la superficie ventral de la garganta de un pez que separa las dos aberturas operculares (las cubiertas de las branquias) entre sí.
[9] Parte angosta que une el cuerpo de un pez con la cola.

- El *keypoint* K_1 se necesita para conocer el punto de inicio anterior de la cabeza. Esto sirve de base para calcular, por ejemplo, la longitud de la cabeza, la longitud estándar o la longitud total del pez, según se muestra en la Figura N° 2.14.
- El *keypoint* K_2 se necesita para conocer el largo total de la boca, pero se utilizó su proyección sobre el contorno de la cabeza para preservar la forma del *box truss*.
- Los *keypoints* K_3 y K_4 se necesitan para calcular algunas dimensiones de la cabeza, como la altura o su longitud, y también para dividir el *box truss* entre la cabeza y el cuerpo del pez. Particularmente, el *keypoint* K_3 se tomó como referencia desde el istmo[10] en lugar del borde posterior del opérculo[11], puesto que es más fácil ubicar el istmo sobre el contorno del pez, a pesar de que el opérculo entrega una referencia más acotada de la longitud de la cabeza.
- Los *keypoints* K_5 y K_6 se necesitan para conocer la altura máxima del cuerpo del pez.
- Los *keypoints* K_7 y K_8 se necesitan para conocer el término del cuerpo del pez y el inicio de la cola, lo cual sirve de base para calcular la longitud estándar, según se muestra en la Figura N° 2.14.

2.5.2 Familias de las Especies Problemáticas

Para el estudio en base a características morfológicas, se trabajó con las 4 especies involucradas en la prueba de validación llevada a cabo por Sernapesca: anchoveta, caballa, jurel y sardina común (ver Sección 3.3.1A ..1 para mayores detalles). En dicha prueba, la identificación de las imágenes entre caballa y jurel presentaba problemas a la hora de diferenciar correctamente a las caballas, lo cual es problemático para la industria, debido a que la caballa pertenece a la fauna acompañante del jurel y aparece con una marcada minoría dentro de los desembarques. Lo mismo aplica para la identificación de las imágenes entre anchoveta y sardina, pero esta vez los fallos en la clasificación se veían afectados

[10] Área estrecha de la superficie ventral de la garganta que separa las dos aberturas operculares entre sí.
[11] Hueso que cubre las branquias de los peces.

por una mala resolución para las imágenes con especies muy pequeñas, disminuyendo en gran medida las características visuales distintivas para estas dos especies.

En definitiva, dado que los modelos de reconocimiento confundían con una muy alta confianza especies de peces que son fácilmente diferenciables para el personal experto, se decidió evaluar la hipótesis sobre este grupo particular de 4 especies de peces, las que en adelante se denominarán especies problemáticas.

A continuación, se muestra una descripción general de las características morfológicas relacionadas con las familias de las 4 especies problemáticas, las que en adelante servirán como una base para llevar a cabo la diferenciación entre las mismas.

A. Engraulidae (anchoveta)

La familia *Engraulidae*, conocida comúnmente como anchoas o boquerones, se caracteriza por tener cuerpos pequeños y delgados, generalmente no superando los 15 cm de longitud. Estos peces tienen una forma corporal alargada y comprimida lateralmente, con una boca grande y protráctil que se abre hacia arriba, adaptada para la alimentación por filtración. Las anchoas tienen una sola aleta dorsal y una aleta caudal bifurcada. Su coloración suele ser plateada, con un brillo metálico (especular) que les ayuda a camuflarse en su entorno pelágico. Son conocidas por formar grandes bancos, lo que les permite protegerse de los depredadores y ser eficientes en la captura de plancton, su principal fuente de alimento.

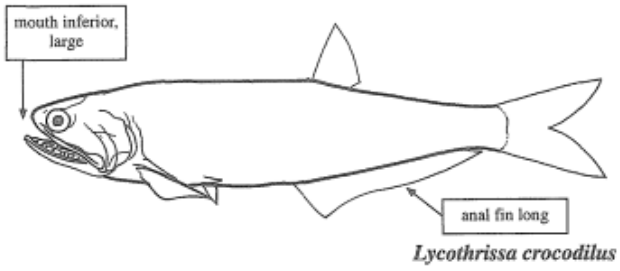


Figura N° 2.16 Ejemplar genérico de pez perteneciente a la familia Engraulidae. Fuente: [56]

B. Scombridae (caballa)

La familia *Scombridae,* que incluye especies como atunes, caballas y bonitos, se caracteriza por tener cuerpos esbeltos y aerodinámicos, perfectamente adaptados para la natación rápida y eficiente. Estos peces tienen una forma fusiforme, con una sección transversal casi circular, y una piel lisa con escamas pequeñas. Poseen dos aletas dorsales, la primera con espinas y la segunda seguida de aletillas. La aleta caudal es profundamente bifurcada, proporcionando una propulsión poderosa. Muchas especies tienen una línea lateral prominente y una coloración que varía desde patrones de rayas hasta tonos azules metálicos y plateados, lo que les ayuda en el camuflaje en el agua abierta. Estos peces son conocidos por su capacidad para mantener una alta actividad metabólica y temperatura corporal más alta que la del agua circundante, lo que les permite una caza eficiente en diferentes condiciones marinas.

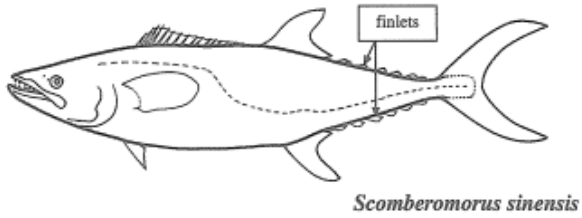


Figura N° 2.17 Ejemplar genérico de pez perteneciente a la familia Scombridae. Fuente: [56]

C. Carangidae (jurel)

La familia *Carangidae*, que incluye jureles, pampanos y chicharros, se caracteriza por tener cuerpos comprimidos lateralmente y fusiformes, adaptados para la natación rápida. Presentan dos aletas dorsales separadas, aletas pectorales largas y puntiagudas, y una aleta caudal bifurcada y potente. Sus escamas suelen ser pequeñas y ajustadas, y la cabeza es grande con una boca prominente y mandíbulas fuertes. Su línea lateral posee grandes escamas en forma de escudos, muy notorios, formando una "quilla" a ambos lados del pedúnculo caudal. La coloración varía entre especies, a menudo con tonos metálicos y

patrones distintivos. El tamaño de estos peces es diverso, reflejando su adaptación a un estilo de vida pelágico donde la velocidad y agilidad son esenciales.

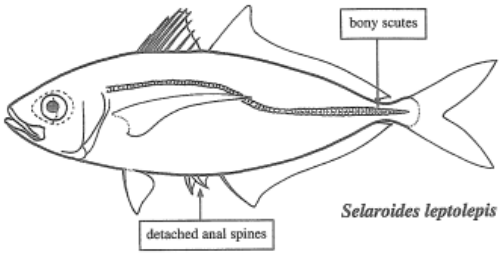


Figura N° 2.18 Ejemplar genérico de pez perteneciente a la familia Carangidae. Fuente: [56]

D. Clupeidae (sardina común)

La familia *Clupeidae*, que incluye especies como sardinas, arenques y alacha, se distingue por sus cuerpos pequeños a medianos, generalmente delgados y alargados, con una forma típicamente comprimida lateralmente. Estos peces tienen una boca pequeña a moderadamente grande, a menudo con una mandíbula inferior prominente. Una característica distintiva es la presencia de escamas grandes y brillantes, que a menudo se desprenden fácilmente. Tienen una sola aleta dorsal y una aleta caudal profundamente bifurcada, que les proporciona una natación rápida y ágil. Su coloración varía, pero comúnmente presentan tonos plateados y brillantes, lo que les ayuda en el camuflaje dentro de su hábitat pelágico. Los *Clupeidae* son conocidos por formar grandes bancos y se alimentan principalmente de plancton, utilizando su método de filtración para capturar pequeñas partículas de alimento.

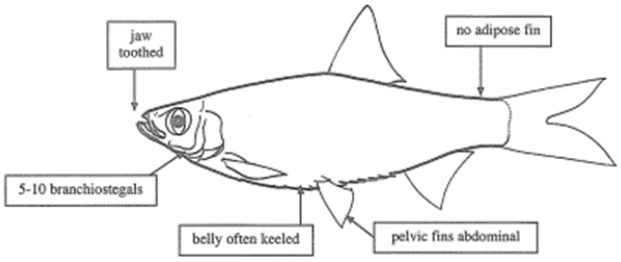


Figura N° 2.19 Ejemplar genérico de pez perteneciente a la familia Clupeidae. Fuente: [56]

2.5.3 Clave Dicotómica para las Especies Problemáticas

En esta sección se presenta, de forma general y sintética, una clave dicotómica construida en base a las descripciones morfológicas de la sección anterior para poder diferenciar un individuo entre las 4 especies problemáticas y una categoría adicional para los peces que no correspondan a ninguna de las descripciones sugeridas: "especie desconocida". Esta se muestra a continuación:

¿El pez es grande?	..... Sí. N° 2
	..... No. (**pequeño**) N° 3
¿El pez tiene manchas en forma de zigzag en la parte superior de su cuerpo?	..... Sí. (**Caballa**)
	..... No, solo presenta un cambio de coloración. (**Jurel**)
	..... No, no presenta patrones de manchas. (**Especie desconocida**)
¿El pez es pequeño y tiene un cuerpo alargado?	..... Sí. N° 4
	..... No. (r**obusto**) N° 5
¿El pez tiene una boca que se proyecta detrás del borde posterior del ojo?	..... Sí. (**Anchoveta**)
	.....No. (**Especie desconocida**)
¿El pez tiene una mandíbula inferior que protruye?	..... Sí. (**Sardina**)
	.....No. (**Especie desconocida**)

Como se puede observar, la clave dicotómica se basa en preguntas que engloban características propias de los peces en base a su tamaño, forma, manchas y boca. Las siguientes subsecciones muestran un mayor detalle respecto a cada tipo de característica.

A. Tamaño

La característica con la que inicia la clave dicotómica está relacionada con el tamaño de las especies problemáticas. Esto se debe principalmente a que el tamaño es una de las características más importantes a nivel biológico, fácil de medir y distintiva en muchas especies [57]. Además, entrega la flexibilidad necesaria para que la clasificación pueda ser diseñada para distinguir entre distintas clases, basadas en un rango de tallas específicas, siempre y cuando se tenga en consideración que muchas especies de peces

pueden variar significativamente en tamaño a medida que crecen y se desarrollan, volviéndose necesario concentrar el análisis en una edad o estadio de desarrollo relativamente estacional para una misma especie.

En función de lo anterior, la noción más general que se puede establecer para dividir a las 4 especies problemáticas es la siguiente:

i. Caballa y jurel como especies grandes (con un largo estándar de hasta 36cm para **B03**, sin considerar caballa, puesto que pertenece a **B04** y no se cuenta con una referencia para estimar su tamaño real). En adelante, "*big*".
ii. Anchoveta y sardina como especies pequeñas (con un largo estándar de hasta 15cm para **B03**). En adelante, "*small*".

B. Forma

Siguiendo el camino de las especies pequeñas, la característica relacionada con la forma es clave para poder distinguir una imagen entre una anchoveta de una sardina. Esto se debe a que, a pesar de que ambas especies son fusiformes, las anchovetas tienen una forma corporal mucho más alargada y comprimida lateralmente que las sardinas, siendo estas últimas más robustas, presentando una mayor relación de aspecto o ancho del cuerpo.

En función de lo anterior, la clasificación de las especies pequeñas en base a su forma es la siguiente:

i. Anchoveta como especie alargada. En adelante, "*elongated*".
ii. Sardina como especie robusta. En adelante, "*robust*".

C. Boca

Avanzando un nivel más profundo en la clasificación de las especies pequeñas, la siguiente diferencia importante entre una anchoveta y una sardina se encuentra en la disposición de su boca.

La Figura N° 2.20 muestra los 3 tipos más comunes de bocas que existen, los cuales son perfectamente aplicables a las especies problemáticas. Particularmente, la caballa

y el jurel poseen bocas de tipo terminal, mientras que la anchoveta posee una boca de tipo subterminal inferior que se extiende detrás de su ojo, y la sardina posee una boca subterminal superior que protruye o sobrepasa la mandíbula superior.

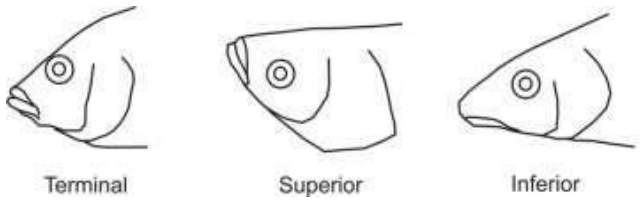


Figura N° 2.20 Tipos comunes de boca presentes en al menos una de las especies problemáticas. Fuente: [53]

Considerando que no es de interés clasificar en específico ninguna especie desconocida, una situación particular se da a lugar con el nodo de boca, donde las posibilidades de clasificación de los peces pequeños se pueden reducir a:

1. Alargado y con una boca que se extiende hacia atrás (anchoveta). En adelante, "*no_protrude*".
2. Robusto y con una boca que se extiende hacia atrás (desconocido).
3. Alargado y con una boca que se extiende hacia adelante (desconocido).
4. Robusto y con una boca que se extiende hacia adelante (sardina). En adelante, "*protrude*".

De esta forma, se decidió por simplemente clasificar si la boca se extiende hacia adelante (protruye) o hacia atrás, replicando este nodo tanto para la rama de clasificación de anchoveta como de sardina.

D. Manchas

Siguiendo el camino de las especies grandes, dado que una caballa y un jurel son muy parecidos tanto en forma como en tamaño, la principal característica distintiva entre

ambas se encuentra en los patrones de manchas presentes en su zona dorsal y en el opérculo del jurel.

La Figura N° 2.21 muestra aproximadamente los tipos de manchas más comunes que se pueden encontrar en las especies problemáticas. Particularmente, la caballa presenta a lo largo de todo su dorso el patrón de la izquierda, denominado zigzag o "*painted*", mientras que la anchoveta, el jurel y la sardina también presentan un patrón de manchas a lo largo de todo su dorso similar al patrón de la derecha sin las zonas blancas intermedias, denominado ensillado o "*saddled*", el cual también se puede interpretar de manera más simple como un cambio de coloración. Además, la anchoveta y el jurel presentan un tipo de mancha adicional de tipo ocelado o "*eye-like*", la cual se presenta en la zona superior del opérculo en el caso de la anchoveta, y en la zona posterior del opérculo en el caso del jurel.

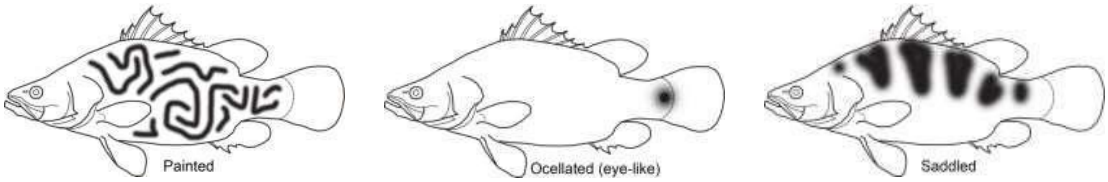


Figura N° 2.21 Patrones corporales o manchas comunes presentes en al menos una de las especies problemáticas. Fuente: [53]

En función de lo anterior, considerando que no es posible identificar correctamente las manchas presentes en el opérculo del jurel con los *keypoints* escogidos, la clasificación de las especies grandes en base a sus patrones de manchas es la siguiente:

i. Caballa como especie con patrones de manchas en zigzag. En adelante, "*patches*" o "*painted*".
ii. Jurel como especie con cambios de coloración en su zona dorsal. En adelante, "*full_color*".

iii. Partes adicionales del cuerpo o especies sin presencia de manchas. En adelante, "*no_patches*".

2.5.4 De Clave Dicotómica a Modelo de Clasificación

Para facilitar el entendimiento del problema y, de la misma forma, para llevar la clave dicotómica a una estructura que sea análoga de la construcción de un sistema de clasificación en base a preguntas o reglas, se elaboró el diagrama de la Figura N° 2.22.

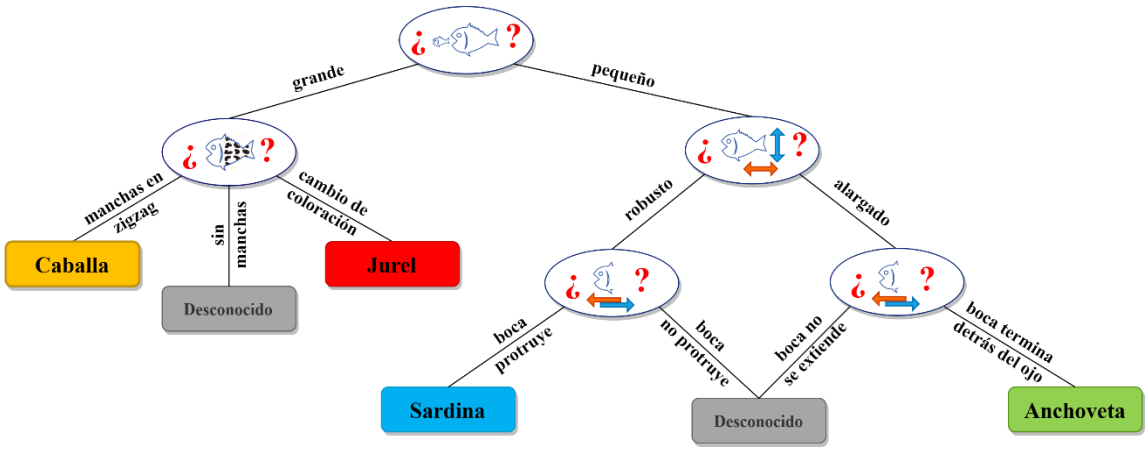


Figura N° 2.22 Árbol de clasificación jerárquica para las especies de interés. Fuente: [Elaboración propia]

El diagrama en particular posee una estructura de árbol o jerarquía de categorías, donde cada nivel de la jerarquía se enfoca en las características específicas que son relevantes para la identificación de las especies problemáticas. Por ejemplo, para este caso particular, se crearon un máximo de 3 niveles utilizando información acerca del tamaño, la forma, la boca y la presencia de manchas en los peces.

Luego, la idea es adaptar cada nivel de la jerarquía o cada pregunta de la clave dicotómica a un minimodelo de clasificación que se construya de manera específica para

extraer la característica que se está evaluando, aplicando para ello *feature engineering*[12]. Esta técnica busca reemplazar el entrenamiento característico de los modelos de clasificación, los cuales extraen características de todo tipo (actuando como una caja negra global), dividiendo la extracción de características en etapas más específicas (actuando como cajas negras locales), y con ello flexibilizar el entrenamiento y depuración de la discriminación de especies en función de su necesidad, facilitando además su adaptación a un mayor número de escenarios en un futuro.

Además, considerando que un tipo de clasificación (ver Sección 2.3.5C) requiere que todas las especies problemáticas sean clasificadas por todos los minimodelos, se debieron realizar las siguientes adaptaciones:

- Para el minimodelo de forma, tanto caballa como jurel pertenecen a la clase "*robust*".
- Para el minimodelo de boca, tanto caballa como jurel pertenecen a la clase "*protrude*". Sin bien ambas especies debiesen pertenecer a la clase "especie desconocida" por tener una boca de tipo terminal, esto implicaría entrenar minimodelos de boca diferentes para las ramas "*robust*" y "*elongated*", lo cual es poco práctico considerando que la probabilidad de que una caballa o un jurel sean clasificados como una especie pequeña es muy baja. En vista de que esto no afectó los resultados finales, se prefirió proceder de esta forma.
- Para el minimodelo de manchas, tanto anchoveta como sardina pertenecen a la clase "*full_color*".

[12] Proceso de seleccionar, modificar o crear características relevantes a partir de datos crudos para mejorar la eficacia de los modelos de aprendizaje automático.

2.5.5 Preparación de los Datos

A. Mediciones Morfométricas

Para garantizar buenos resultados al momento de utilizar mediciones morfométricas, es importante considerar que todas las imágenes utilizadas deben preservar la relación de aspecto original de los peces. Esto no necesariamente implica conocer la medida real de los mismos, como se analizará más adelante en la Sección 0, pero si evitar aplicar técnicas de *data augmentation* que alteren la relación de aspecto, como *crops* (recortes), *zoom* o estiramientos horizontales o verticales. Por ejemplo, la Figura N° 2.23 muestra una comparación entre una imagen de jurel y otra de anchoveta que fueron utilizadas durante el entrenamiento de los modelos. Si bien se puede apreciar que ambas especies están a una misma escala de tamaño (para permitir una mejor visualización de sus características), el reescalado de todas las imágenes se realizó usando el método de *letterboxing*[13], por lo que su relación de aspecto no fue alterada. Lo anterior produce un efecto inverso en la sensación visual que se tiene respecto del tamaño de las especies, puesto que las especies grandes tienden a disminuir en tamaño para adaptarse a la nueva resolución de la imagen, mientras que las especies pequeñas tienden a aumentar su tamaño.

(a) (b)

*Figura N° 2.23 Comparación entre una especie grande y una especie pequeña de la base de datos **B03**. (a) jurel, especie grande; (b) anchoveta, especie pequeña. Fuente: [Elaboración propia]*

[13] Este algoritmo se encarga de añadir bandas negras a los lados o en la parte superior e inferior de la imagen para llenar el espacio restante, lo que garantiza que se conserve su relación de aspecto original.

Respecto de la extracción de las características morfométricas, se tomaron como referencia algunos de los datos morfométricos comunes mostrados en la Figura N° 2.24, particularmente, el ancho del cuerpo (*body depth*) y la longitud estándar (*standard length*). Luego, el resto de las características fueron extraídas directamente de los 8 *keypoints* identificados para todas las especies. Por un lado, se consideraron las 16 distancias morfométricas entre pares de *keypoints* (en píxeles) conformadas por el esqueleto del *box truss*, cuyas proyecciones se muestran en la Figura N° 2.25.Por otro lado, siguiendo las mediciones realizadas en [15], se consideraron adicionalmente el ángulo de la cabeza y los 4 ángulos conformados entre los extremos superior, inferior, lateral izquierdo y lateral derecho del *box truss*, según se muestra en la Figura N° 2.26. Un resumen con las 21 características medidas se muestra en la Tabla N° 2.6, donde se incorporó el *keypoint* auxiliar, $K_{\overline{78}}$, el cual representa el punto medio entre K_7 y K_8.

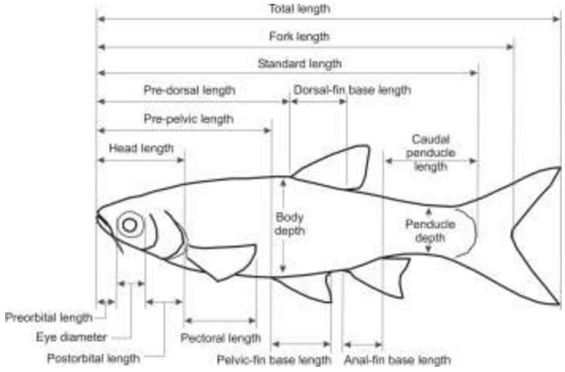


Figura N° 2.24 Datos morfométricos comunes recopilados para la identificación de peces. Fuente: [53]

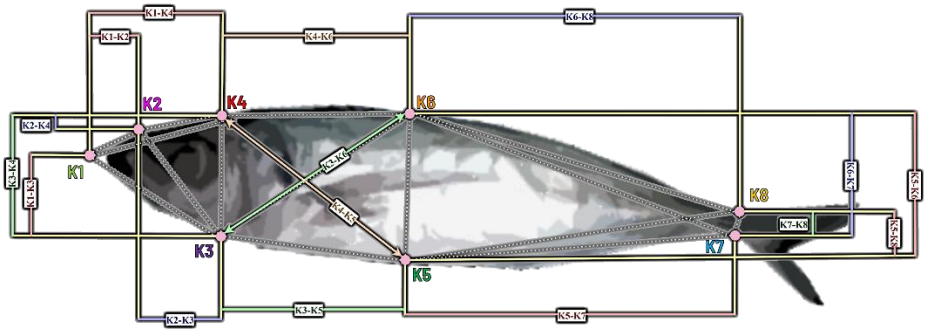


Figura N° 2.25 Distancias morfométricas seleccionadas para la identificación de las especies de interés. Fuente: [Elaboración propia]

Adicionalmente, todas las mediciones morfométricas se utilizarán como una proporción del largo estándar, el cual está definido como la distancia desde el extremo anterior del hocico o maxila (K_1) hasta el extremo posterior del pedúnculo caudal ($K_{\overline{78}}$). Esto es necesario porque proporciona una manera estandarizada y consistente de comparar y analizar el tamaño y la forma de peces entre diferentes especies, poblaciones y estudios [58]. Particularmente, es útil para mitigar las desviaciones de tamaño presentes en individuos de una misma especie, y elimina la necesidad de conocer el tamaño real en centímetros de un individuo en particular.

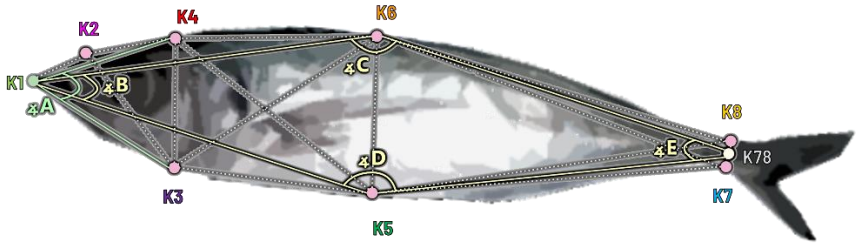


Figura N° 2.26 Ángulos morfométricos seleccionados para la identificación de las especies de interés. Fuente: [Elaboración propia]

Tabla N° 2.6 Resumen de las mediciones morfométricas extraídas de los keypoints, tanto distancias como ángulos.

ID	Distancia de referencia
D_1	K_1-K_2
D_2	K_1-K_3
D_3	K_1-K_4
D_4	K_2-K_3
D_5	K_2-K_4
D_6	K_3-K_4
D_7	K_3-K_5
D_8	K_3-K_6
D_9	K_4-K_5
D_{10}	K_4-K_6
D_{11}	K_5-K_6
D_{12}	K_5-K_7
D_{13}	K_5-K_8
D_{14}	K_6-K_7
D_{15}	K_6-K_8
D_{16}	K_7-K_8

ID	Ángulo de referencia
∡A	∡K_4-K_1-K_3
∡B	∡K_6-K_1-K_5
∡C	∡K_1-K_6-$K_{\overline{78}}$
∡D	∡ $K_{\overline{78}}$-K_5-K_1
∡E	∡ K_6-$K_{\overline{78}}$-K_5

B. Segmentación de Manchas

La Figura N° 2.27 muestra el proceso de segmentación de manchas aplicado a todas las imágenes pertenecientes a las bases de datos **B03** y **B04**, donde el triángulo superior más cercano a la zona dorsal del pez enfoca una zona con manchas, mientras que el triángulo inferior más cercano a la zona ventral del pez enfoca una zona sin manchas. Luego, considerando que los triángulos se encuentran dentro de la segunda sección del *box truss* conformado por los *keypoints* K_3 al K_6, si se define el *keypoint* auxiliar, K_0, como el punto de intersección de los segmentos $\overline{K_4K_5}$ y $\overline{K_3K_6}$, el triángulo superior queda determinado por ΔK_0-K_4-K_6, mientras que el triángulo inferior queda determinado por ΔK_0-K_5-K_3.

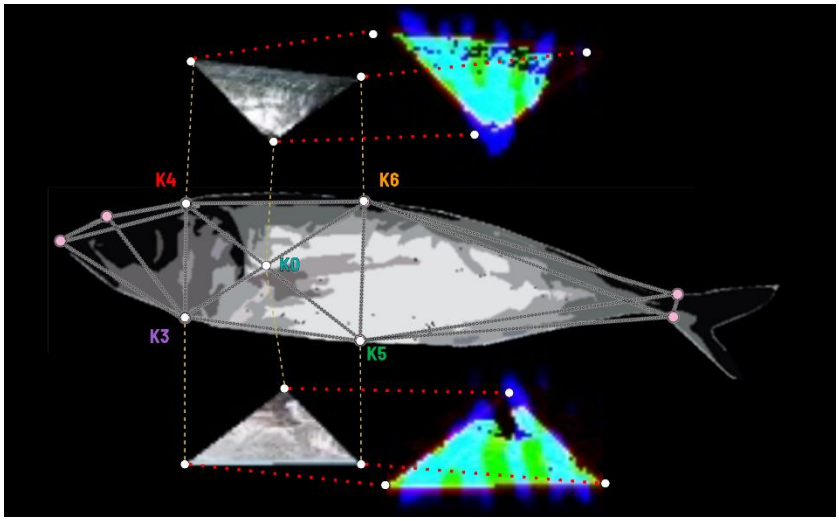


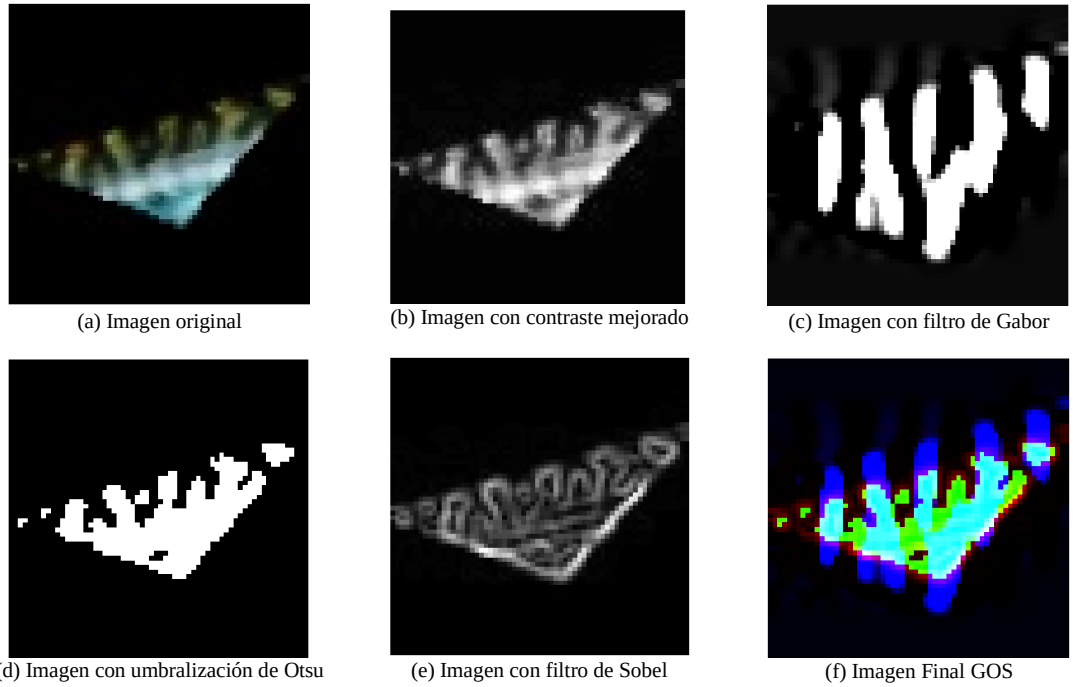
Figura N° 2.27 Extracción de los triángulos definidos por los keypoints K_0-K_4-K_6 y K_0-K_5-K_3. Una vez extraídos, estos son pasados por los filtros GOS, resultando en los triángulos de colores a la derecha de la imagen. Fuente: [Elaboración propia]

En la imagen anterior también se revelan las dos variantes de imágenes que fueron utilizadas durante el entrenamiento del minimodelo, todas con una resolución de 64x64 escogida experimentalmente en función del tamaño original de los triángulos para todas

las especies. Por un lado, se tienen las imágenes RGB de cada triángulo segmentado, reescalado y luego puesto sobre un fondo negro. Por otro lado, se tienen las mismas imágenes anteriores, solo que estas fueron pasadas adicionalmente por una combinación de filtros denominada GOS; sigla que representa las iniciales de los filtros de Gabor, Otsu y Sobel (ver Anexo A.6.4).

Respecto de las imágenes GOS, la elección de estos filtros se debe a que, en muchas de las imágenes presentes en las bases de datos **B03** y **B04**, los patrones de manchas se ven distorsionados debido a la baja resolución de las imágenes con los triángulos. Si bien esto generalmente no suele ser un problema para entrenar un modelo de redes neuronales, donde es común utilizar imágenes con resoluciones pequeñas con detalles distorsionados, se prefirió probar con uno o más filtros que fueran capaces de realzar o segmentar las manchas para lograr, en teoría, un mejor rendimiento por parte de los modelos. Particularmente, la elección del filtro de Gabor fue incentivada por el trabajo desarrollado por Ha *et al.*, donde aplicaron este filtro para segmentar manchas de suciedad en los lentes de una cámara, las cuales se caracterizan por ser muy sutiles y hasta difíciles de apreciar visualmente [59].

La Figura N° 2.28 muestra todas las transformaciones aplicadas en las imágenes de los triángulos para generar su variante GOS. Primero, la imagen pasa por un filtro de realce de contraste mediante el método de ecualización de histograma adaptativo limitado por contraste (CLAHE, por sus siglas en inglés), priorizando una mejor calidad y una menor amplificación del ruido inherente a la baja resolución de las imágenes (apartado b). Luego, la imagen con contraste mejorado pasa por los 3 filtros ya mencionados. En el apartado c se muestra el resultado luego de pasar la imagen por un filtro de Gabor. En el apartado d se muestra el resultado luego de pasar la imagen por una umbralización de Otsu. En el apartado e se muestra el resultado luego de pasar la imagen por un filtro de Sobel. Finalmente, el apartado f muestra la imagen final GOS, donde todos los filtros se unieron simulando ser un nuevo espacio de colores, como en una imagen RGB tradicional.



(a) Imagen original (b) Imagen con contraste mejorado (c) Imagen con filtro de Gabor

(d) Imagen con umbralización de Otsu (e) Imagen con filtro de Sobel (f) Imagen Final GOS

Figura N° 2.28 Aplicación de diferentes filtros a una imagen recortada con las manchas características de la zona dorsal de una caballa. Fuente: [Elaboración propia]

2.5.6 Estimación de Talla y Peso

La expresión para estimar el peso de una especie de pez en relación con su talla es la siguiente:

$$W = aL^b$$

Ecuación *2.1*
Curva talla/peso

En esta expresión, W representa el peso del pez, en gramos, y L representa la longitud estándar del pez, en centímetros. Además, se consideran dos parámetros, a y b, los cuales pueden variar dependiendo de factores como la ubicación geográfica, la temporada y otras variables específicas de cada especie de pez. En consecuencia, las estimaciones pueden diferir de un año a otro y entre distintas zonas de pesca. La Tabla N° 2.7 muestra los parámetros ajustados solo para las 4 especies problemáticas, puesto que no se cuenta con la información necesaria para estimar las curvas de las especies restantes de la base

de datos **B02**. Los parámetros a y b se determinaron mediante mediciones experimentales de laboratorio llevada a cabo en las dependencias de la Universidad de Concepción.

Tabla N° 2.7 Parámetros estimación de talla/peso para cada clase.

Clase	a	b
Anchoveta	0.003653	3.23344
Caballa	0.0090147	3.1067
Jurel	0.007921	3.1085
Sardina	0.0064065	3.1299

CAPÍTULO 3. EVALUACIÓN Y RESULTADOS

3.1. Arquitectura del Sistema de Reconocimiento

El proceso completo propuesto para el funcionamiento del *back-end* del sistema de reconocimiento de peces se resume en el diagrama de la Figura N° 3.1, donde se muestra la arquitectura macro del *software* con sus cinco etapas más importantes: captura de imágenes, preprocesamiento, discriminación de peces, postprocesamiento y despliegue de los resultados obtenidos. Adicionalmente, se muestran algunas etapas secundarias, como el etiquetado de imágenes y la estimación de la talla y el peso de los peces detectados por el sistema.

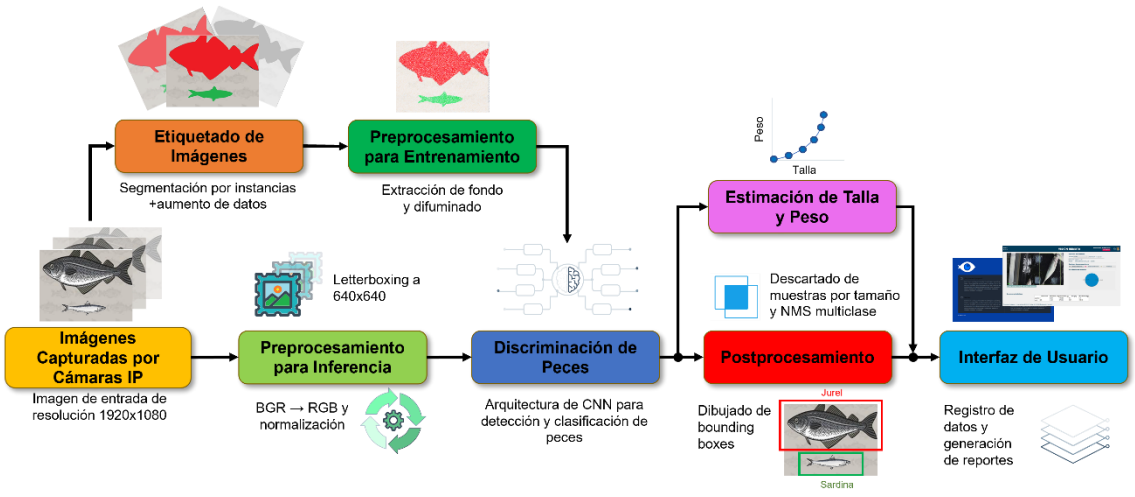


Figura N° 3.1 Arquitectura del sistema de reconocimiento de peces. Fuente: [Elaboración propia]

Antes de entrar en el detalle de la descripción del sistema de reconocimiento, es importante mencionar que este sistema está pensado para operar en línea y conectarse remotamente a alguno de los pórticos instalados en las plantas Blumar u Orizon, según fue mencionado en la Sección 2.1. Sin embargo, dado que en un ambiente industrial no es fácil llevar a cabo experimentación *in situ* con el pórtico, todas las pruebas de laboratorio relacionadas con el estudio de las características morfológicas de las especies de interés se realizaron dentro de las dependencias de la Universidad.

Ahondando con mayor profundidad en la operación de cada etapa, la primera etapa se encarga de recibir inicialmente una imagen con una resolución fija de 1080p de la fuente actual seleccionada mediante protocolo RTSP. A continuación, la imagen de entrada se preprocesa en la segunda etapa.

Particularmente, el módulo de preprocesamiento consta de tres funciones principales que se aplican antes de la etapa de discriminación. En primer lugar, el formato de composición de color de las imágenes se transforma de BGR a RGB. Esta conversión es necesaria por el modo de operación de OpenCV, biblioteca de Python utilizada principalmente para el procesamiento de imágenes, la cual utiliza por defecto el formato BGR para leer imágenes a partir de una cámara. En segundo lugar, las imágenes se redimensionan mediante el algoritmo de *letterboxing* para ajustar su resolución de 1080p a 640x640, según se muestra en la Figura N° 3.2. Por último, el módulo de preprocesamiento incluye una función para normalizar los valores de intensidad de píxeles de las imágenes desde el rango original de [0, 255] hasta el rango de [0, 1].

La tercera etapa contempla el foco principal del trabajo que se desarrolló en esta tesis, el sistema de discriminación de peces. Particularmente, esta etapa debe cumplir con tres funciones: la primera se encarga de detectar las especies de peces presentes en cada imagen, la segunda se encarga de encontrar los *keypoints* de cada especie detectada, y la tercera se encarga de asignarles una clase o etiqueta final, la cual se compara con la clase

entregada por el modelo de detección. Un diagrama representando cada función de esta etapa se muestra en la Figura N° 3.3.

Figura N° 3.2 Algoritmo de letterboxing aplicado a una imagen del pórtico Blumar para ajustar su resolución a 640x640 píxeles. Fuente: [Elaboración propia]

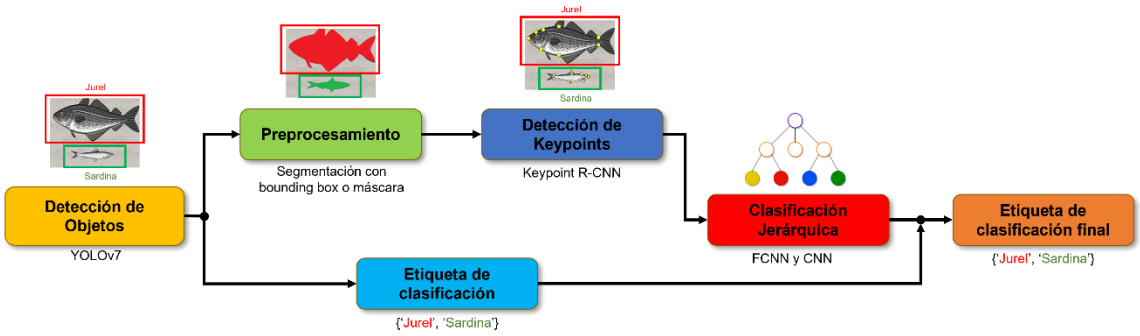


Figura N° 3.3 Arquitectura de la etapa de discriminación de peces. Fuente: [Elaboración propia]

Para su desarrollo, el lineamiento se divide entre la industria y el laboratorio. Para la industria, la discriminación de peces solo depende del modelo de detección de objetos, el cual entregará un *bounding box*, una máscara y una etiqueta de clasificación para cada pez encontrado en la imagen de entrada. Para el laboratorio, además de contar con la información generada por el modelo de detección, se agregaron dos fases adicionales que servirán como una doble verificación de la clasificación realizada por el detector. Esto incluye una fase de detección de *keypoints* para automatizar la extracción de las características morfológicas de los peces (ver Sección 3.3.2), la cual puede o no depender de un

módulo de preprocesamiento para segmentar las detecciones de su fondo; y, además, una fase de clasificación jerárquica que se alimenta de las características morfológicas para generar la etiqueta final con la especie del pez correspondiente (ver Sección 3.4).

Dentro de la cuarta etapa, el algoritmo de postprocesamiento incorpora tres módulos. El primer módulo se encarga de descartar las detecciones en función de su tamaño, lo que permite un filtrado más preciso de los resultados. El segundo módulo implementa un algoritmo de NMS multiclase, que ayuda a eliminar las detecciones redundantes o superpuestas entre distintas clases de peces. Por último, el tercer módulo se encarga de dibujar los *bounding boxes* y/o máscaras resultantes de cada detección en la imagen original. También se incluye la etiqueta con el nombre de la especie, la talla y el peso de la detección y el nivel de confianza entregado por el modelo, lo que proporciona una representación visual y anotaciones informativas, tal y como se muestra en la Figura N° 3.4.

Particularmente, el módulo para NMS multiclase desempeña un papel crucial en el filtrado de los resultados de detección, puesto que se comprueba la proximidad entre *bounding boxes* pertenecientes a diferentes clases (operación no realizada internamente por el modelo, a diferencia del NMS para una misma clase, el cual si está considerado). De esta forma, si se encuentra que dos *bounding boxes* están muy cerca uno del otro, el módulo elimina el *bounding box* con el nivel de confianza más bajo. Este paso garantiza que solo se conserven las detecciones más seguras y distintas, lo que evita la redundancia y mejora la precisión general del sistema.

Respecto de la estimación de la talla de un pez, se aplicó un factor de conversión para obtener su valor de longitud real en centímetros, que corresponde a la distancia representada por un solo píxel en el mundo real. Este factor de conversión se determinó de antemano para cada pórtico de su respectiva planta, y es importante calibrarlo correctamente cada vez que la cámara cambia de posición o alguna de sus configuraciones, como el nivel de zoom.

(a)

(b)

Figura N° 3.4 Detecciones de especies en la industria. (a) Planta Orizon; (b) Planta Blumar. Fuente: [Elaboración propia]

Por último, el sistema incluye una etapa de almacenamiento de datos que facilita la organización y gestión de los datos procesados. En esta se incluye la entrega del tamaño y peso estimados para cada pez detectado a partir de su curva talla/peso, junto con un resumen completo de todos los datos procesados. Finalmente, el sistema genera un reporte final en formatos Excel y PDF, lo que permite un cómodo acceso y análisis de la información recopilada.

3.2. Requerimientos Mínimos

Antes de llevar a cabo el entrenamiento de los modelos de detección y clasificación, es importante tener en consideración los requerimientos de *hardware* mínimos necesarios para realizar dicha tarea. Los modelos basados en CNN, particularmente arquitecturas más avanzadas como las utilizadas por YOLO o las derivadas de R-CNN, son computacionalmente demandantes debido a poseer múltiples capas y una gran cantidad de parámetros, los cuales intervienen a su vez con una gran cantidad de operaciones matriciales y convoluciones internas. A lo anterior también se le suma el hecho de que estos modelos procesan lotes o *batches* de múltiples imágenes al mismo tiempo a lo largo de varias épocas, lo cual implica almacenar en memoria grandes cantidades de datos y, generalmente, conlleva la implementación de diversas fases de preprocesamiento, como las ya mencionadas en la sección anterior. Este enfoque de entrenamiento, en el que el modelo es expuesto repetidamente al conjunto completo de datos, intensifica aún más la demanda de recursos computacionales.

Debido a lo anterior, es importante contar con una unidad de procesamiento gráfico (GPU). Las GPU están diseñadas para realizar operaciones en paralelo, lo que es ideal para los cálculos matriciales de una CNN. Sin embargo, para aprovechar plenamente este *hardware*, es necesario estimar correctamente la cantidad de memoria virtual (VRAM) que se necesita, lo que generalmente depende del o los modelos que se utilizarán, y del tamaño del conjunto completo de datos.

Por ejemplo, la Figura N° 3.5 muestra las estimaciones del tamaño en MB que ocupa el modelo YOLOv7 en una tarjeta gráfica (GPU), donde "*Params size*" representa el tamaño de los parámetros cargados en el modelo (152MB), y "*Forward/backward pass size*" representa el tamaño de todas las operaciones internas entre capas que realiza el modelo al procesar una única imagen (1687MB). Esto determina que el espacio mínimo en memoria requerido por el modelo es de aproximadamente 1.8GB, lo cual irá escalando

en función de la tarea que se desea realizar (entrenamiento o inferencia) y en función del número de imágenes que se deben cargar adicionalmente para ser procesadas al mismo tiempo (generalmente más influyente durante el entrenamiento).

```
Total params: 37,882,274S
Trainable params: 0
Non-trainable params: 37,882,274
Total mult-adds (G): 70.96
==========================================
Input size (MB): 4.92
Forward/backward pass size (MB): 1530.34
Params size (MB): 151.53
Estimated Total Size (MB): 1686.78
```

Figura N° 3.5 Espacio ocupado por YOLOv7 en la memoria de la GPU. Fuente: [Elaboración propia]

En función del análisis anterior, y considerando que las bases de datos que se utilizarán para entrenar a los modelos son relativamente pequeñas, es recomendable tener una GPU con al menos 8 GB de memoria dedicada para garantizar un entrenamiento eficiente y sin interrupciones. Además, también es recomendable contar con un procesador de al menos cuatro núcleos y 16 GB de memoria RAM para manejar todas las operaciones paralelas necesarias y almacenar temporalmente los datos durante el proceso de entrenamiento.

Específicamente hablando del *hardware* utilizado durante todas las pruebas a lo largo de esta tesis, todos los modelos de detección fueron entrenados en la nube utilizando la plataforma Google Colab [37]. Si bien los recursos solicitados podían variar dependiendo de la demanda y disponibilidad del usuario, una configuración común era la siguiente: CPU Intel(R) Xeon(R) @ 2.20GHz, 2 núcleos, 12.68GB de RAM y GPU Tesla P100-PCIE-16GB.

Respecto de todas las pruebas realizadas con los modelos de clasificación, estas fueron realizadas en un computador local con las siguientes características: CPU Ryzen 7 5800H de 8 núcleos/16 hilos, GPU Nvidia RTX3070 Laptop 8 GB de VRAM y 32 GB de RAM DDR4.

3.3. Diseño de los Modelos de Detección

3.3.1 Detección de Objetos y Segmentación por Instancias

En esta sección se mostrará el diseño y los resultados obtenidos para la primera función de la etapa de discriminación de especies, la cual está encargada de encontrar todas las instancias de peces de interés presentes en una imagen, para luego generar la información de su localización en formato de *bounding box* y/o máscara junto con una etiqueta de clasificación con el nombre de la especie detectada, y con ello alimentar las siguientes etapas del sistema de reconocimiento.

Teniendo en mente el proceso de entrenamiento y testeo, el diseño de este modelo se realizó para adaptarse a los siguientes entornos en la industria:

i. Planta Orizon, utilizando la base de datos **B01**.
ii. Planta Blumar, utilizando la base de datos **B02**.

A. Resultados Planta Orizon

Inicialmente, durante una fase de prueba, se utilizaron varios modelos de CNN que realizan de manera conjunta tanto detección de objetos como clasificación, entre los que se incluyen YOLOv4 y Mask R-CNN, presentados en la Sección 2.3. La elección de estos modelos se basó en las características distintivas de sus arquitecturas y sus capacidades específicas para detectar objetos en imágenes. Por una parte, YOLOv4 es un modelo solo capaz de generar *bounding boxes*, siendo en promedio 17 veces más rápido que Mask R-CNN, a costa de una precisión menor al momento de encontrar los distintos objetos en

una imagen. Por otro lado, Mask R-CNN es un modelo capaz de generar tanto *bounding boxes* como máscaras, siendo mucho más preciso que YOLOv4, a costa de un tiempo de inferencia mucho más lento, cercano a los 300ms.

A ..1 *Resultados Validación Sernapesca*

En el caso particular de la planta Orizon, al ser la primera planta donde se implementó una prueba piloto del sistema de reconocimiento, se realizó una validación manual de los mejores modelos entrenados por inspectores de Sernapesca. Para llevar a cabo este proceso, los inspectores analizaron algunas imágenes obtenidas para distintas descargas, considerando tanto especies de peces pequeñas como especies de peces grandes, y ayudaron en tareas como la identificación de especies y la corrección de las detecciones erróneas a partir de los resultados de inferencia de los modelos, lo que luego permitió realizar los ajustes finales.

Sin entrar en mayores detalles respecto del entrenamiento de dichos modelos, que corresponden a una etapa previa de lo desarrollado dentro del Proyecto FONDEF IT20I0032, los mejores resultados fueron obtenidos para el modelo Mask R-CNN, los cuales se resumen en la Tabla N° 3.1.

Tabla N° 3.1 Resumen de las métricas MP obtenidas para la prueba de validación realizada por Sernapesca en la planta Orizon.

	Modelo Mask R-CNN – 120 imágenes de prueba		
Clase	**mínimo**	**máximo**	**promedio**
Anchoveta	0.58	0.86	0.75
Caballa	0.80	0.90	0.87
Jurel	**0.98**	0.99	**0.98**
Sardina	0.85	**1.00**	0.94

A partir de esta tabla, es posible concluir que:

- El modelo en particular ya era capaz de lograr métricas MP de al menos 0.9 para 3 de las 4 especies, siendo el jurel el que alcanza el mayor valor en promedio, con un 0.98.

- El modelo alcanza un promedio de 0.93 para reconocer a especies de peces grandes en imágenes, y un promedio de 0.85 para reconocer a especies de peces pequeñas.

A ..2 ***Resultados tras el Cambio de Modelo***

Posteriormente, con el lanzamiento de YOLOv7 a mediados de 2022, surgió un nuevo modelo que combina los beneficios de obtener información similar a la de Mask R-CNN, pero con una velocidad de inferencia significativamente más rápida que YOLOv4. Como resultado, YOLOv7 se destacó como la opción más adecuada entre los modelos evaluados y fue el seleccionado para presentar los resultados de los modelos de detección de objetos en esta tesis.

En consecuencia, el entrenamiento del modelo YOLOv7 para la planta Orizon fue realizado con las siguientes configuraciones:

- Distribución 70%-30% para los sets de entrenamiento y validación, respectivamente.
- Imágenes RGB de tamaño 640x640, reescaladas con el algoritmo de *letterboxing*.
- Preprocesamiento de las imágenes aplicando *blurring* o reemplazo del fondo, según se indica en al Anexo A.6.1 y en el Anexo A.6.2.1, respectivamente.
- *Learning rate* variable de tipo decaimiento coseno con calentamiento, siguiendo el *scheduler* mostrado en el Anexo A.5.3, con un mínimo de 1e-3 y un máximo de 1e-4.
- Número total de épocas[14] ajustado manualmente hasta 500, considerando un *Early Stopping* ajustado para 100 épocas, según lo mostrado en el Anexo A.5.2.

[14] Una época representa un ciclo en la que todos los ejemplos de entrenamiento son utilizados una vez para actualizar los parámetros del modelo.

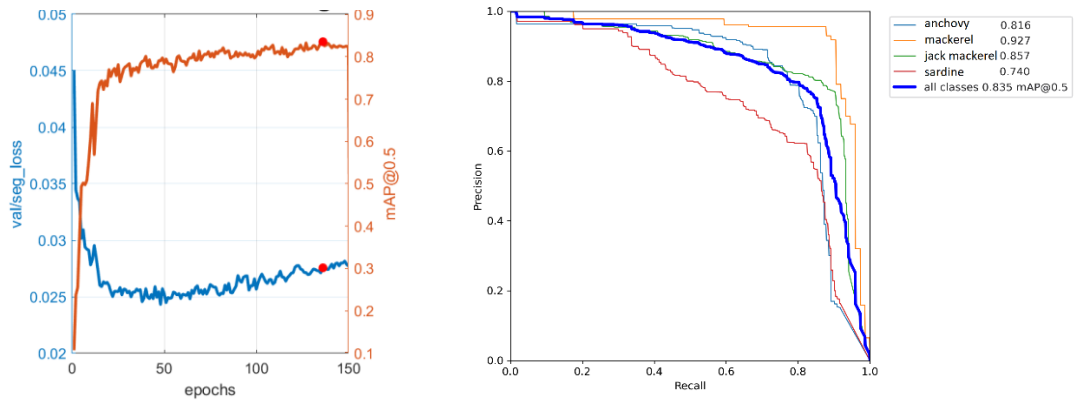
- *Batch*[15] de 16 imágenes por iteración.
- Umbral de *Intersection Over Union* (IOU)[16] ajustado en 0.2.

Para la planta Orizon, el modelo se evaluó para 752 muestras de peces. Los resultados obtenidos se resumen en la matriz de confusión de la Figura N° 3.6. El sistema logró una métrica P de 0.9 para la anchoveta, 0.93 para el jurel, 0.95 para la caballa y 0.83 para la sardina, lo que da como resultado un valor de 0.9 para la métrica MP. La precisión de falsos positivos en el fondo (*background* FP), que corresponde a elementos del fondo detectados erróneamente como pertenecientes a una de las clases entrenadas, tuvo un valor máximo de 0.48 para el jurel. Además, la precisión de falsos negativos en el fondo (*background* FN), que corresponde a las especies que no fueron detectadas por el modelo y que fueron clasificadas erróneamente como parte del fondo, tuvo un valor máximo de 0.16 para la sardina. También se dieron algunos casos de clasificación errónea entre clases, por ejemplo, con el jurel clasificado erróneamente como caballa con un valor de 0.04 y la anchoveta clasificada erróneamente como sardina, y viceversa, a una tasa de 0.01.

Figura N° 3.6 Matriz de confusión obtenida durante la validación del modelo para la planta Orizon. Fuente: [Elaboración propia]

[15] Los *batches* o lotes en el entrenamiento son subconjuntos del conjunto de datos total utilizados para actualizar los parámetros del modelo. Se utilizan para hacer un entrenamiento computacionalmente más eficiente con actualizaciones más frecuentes de los parámetros dentro de una misma época.
[16] IOU es una métrica que mide la superposición entre el área de un *bounding box* predicho y el área de un *bounding box* real. Entre más cercano a 1.0, más restrictivo se vuelve el modelo para generar detecciones.

Por un lado, el apartado a de la Figura N° 3.7 muestra el progreso tanto de la función de pérdida como de la métrica mAP a lo largo de la fase de entrenamiento de la planta Orizon. El punto rojo representa la época 137, que corresponde a los pesos donde el modelo logró sus mejores resultados, donde se obtiene el valor máximo de la métrica mAP de 0.838. Estos pesos son los utilizados en etapas posteriores durante el proceso de inferencia con el modelo. Por otro lado, el apartado b muestra la curva de *precision-recall* para las máscaras generadas por el algoritmo YOLOv7 durante el proceso de validación en la planta Orizon. Esta curva muestra el rendimiento del modelo en términos de la métrica P para diferentes valores o umbrales de la métrica R, y revela que la caballa alcanza el valor de la métrica AP más alto de 0.927, mientras que la sardina obtiene el valor de la métrica AP más bajo de 0.740. Teniendo en cuenta todas las clases, el valor de la métrica mAP es de 0.835, siendo muy similar al máximo valor obtenido en la curva mAP de entrenamiento.

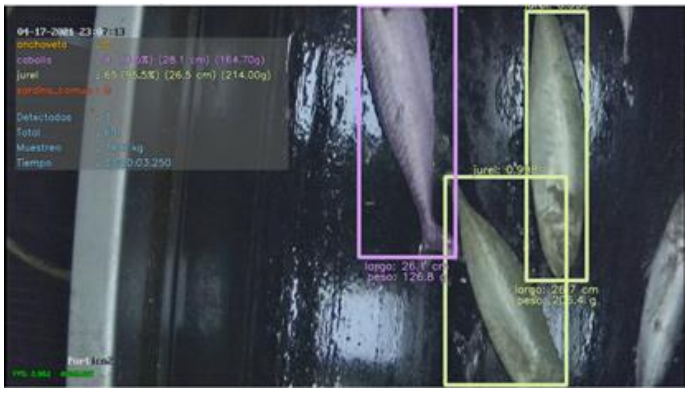


(a) (b)

Figura N° 3.7 Resultados para la planta Orizon utilizando el modelo YOLOv7. (a) Evolución de la función de pérdida y mAP. (b) Curva precision-recall para máscaras. Fuente: [Elaboración propia]

La Figura N° 3.8 muestra un ejemplo de inferencia con el modelo. En el apartado a se muestra específicamente la detección de especies grandes, donde se encontró una

caballa (*bounding box*) y dos jureles (*bounding box*), lo que demuestra que estas especies se detectaron con precisión sin ningún problema para esta imagen en particular. Las especies más desafiantes para el sistema son las especies más pequeñas, como la sardina y la anchoveta. En el apartado b se muestra específicamente la detección de especies pequeñas, donde es habitual encontrar imágenes en las que está presente un número importante de estas especies, pero se revela la gran dificultad para detectarlas a todas debido a su baja calidad. En este caso particular, el modelo solo logró detectar 6 sardinas (*bounding box*).



(a)

(b)

Figura N° 3.8 Detección de especies en la planta Orizon. a) Especies grandes. b) Especies pequeñas.

En la Figura N° 3.8 también se muestra la funcionalidad del módulo de estimación de longitud y peso implementado en su versión antigua, mostrando los valores estimados

debajo de cada *bounding box*. Este módulo proporciona aproximaciones en tiempo real de la longitud y el peso de las especies detectadas. Sin embargo, hay ciertos desafíos que aún deben abordarse, por ejemplo, la orientación del pez y si el modelo detecta el pez entero o solo una parte de él; detalles que pueden afectar la precisión de las estimaciones.

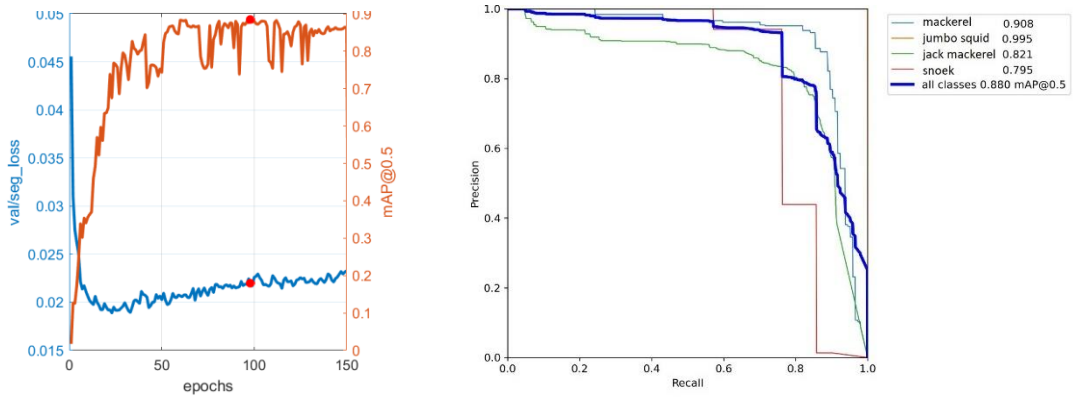
B. Resultados Planta Blumar

El entrenamiento del modelo YOLOv7 para esta planta se realizó con las mismas configuraciones que para la planta Orizon.

Para la planta Blumar, el modelo fue evaluado para la detección de jurel, caballa, jibia y sierra utilizando un total de 1.771 muestras de validación. Los resultados de la validación se resumen en la matriz de confusión de la Figura N° 3.9. Aquí se observa que se logró una precisión de 0.98 para el jurel, 0.84 para la caballa, 1.0 para jibia y 0.4 para sierra, lo que da como resultado un valor de 0.805 para la métrica MP. Por un lado, la precisión de *background* FP solo tiene un valor de 0.01 para el jurel, mientras que el resto de las clases tienen un valor de 0. Por otro lado, la precisión de *background* FN tiene un valor máximo de 0.52 para el jurel y un valor mínimo de 0 para jibia.

Figura N° 3.9 Matriz de confusión obtenida durante la validación del modelo para la p1anta Blumar. Fuente: [Elaboración propia]

Por un lado, el apartado a de la Figura N° 3.10 presenta la evolución de la función de pérdida y mAP durante el proceso de entrenamiento del modelo en la planta Blumar. El punto rojo en el gráfico indica la época 97, que corresponde a los pesos donde el modelo logró sus mejores resultados. En esta época, el modelo alcanza un valor máximo de la métrica mAP de 0.88, lo que indica su rendimiento óptimo en términos de precisión de detección. Por otro lado, el apartado b muestra la curva de *precision-recall* del modelo en esta planta, donde se alcanza el valor máximo de la métrica AP de 0.995 para la jibia, lo que indica una alta precisión en la detección de esta clase. Además, la sierra tiene el valor de la métrica AP más bajo de 0.795, lo que sugiere que todavía puede lograrse un margen de mejora en su rendimiento de detección, como se observa en la matriz de confusión durante la validación, donde el modelo confunde un 60% de las detecciones de sierra como si estas fueran jurel. Teniendo en cuenta todas las clases, el valor de la métrica mAP de validación es de 0.888, lo que representa una medida general del rendimiento del modelo, siendo también muy similar al mejor valor de la métrica mAP durante el entrenamiento.



(a) (b)

Figura N° 3.10 Resultados para la planta Blumar utilizando el modelo YOLOv7. (a) Evolución de la función de pérdida y mAP. (b) Curva precision-recall para máscaras. Fuente: [Elaboración propia]

Los resultados obtenidos de la planta Blumar ponen de relieve los desafíos en la detección de la sierra, ya que distinguir visualmente una sierra de una caballa o un jurel

no es sencillo a partir de las imágenes con las que se cuenta actualmente. Sin embargo, la jibia sí es fácilmente reconocible por el sistema debido a su gran tamaño y color distintivo, lo que la hace destacar en comparación con otras especies.

(a)

(b)

Figura N° 3.11 Detección de especies en la planta Blumar. a) Detección de múltiples especies de forma simultánea. b) Detección de caballa y jibia.

En el apartado a de la Figura N° 3.11 se observa un ejemplo de identificación, con una presencia significativa de jurel (indicado por el *bounding box* 🟠), en la correa transportadora. En la zona derecha de la imagen, se identifican dos caballas (indicadas por el *bounding box* 🔴), caracterizadas por un patrón de manchas en zigzag en su zona dorsal. En el lado izquierdo de la imagen, se detecta una sierra (indicada por el *bounding box* 🟡). Cabe señalar que la sierra es de mayor tamaño en comparación con las otras especies, pero es altamente probable que no aparezca completa dentro de la captura de la imagen.

En el apartado b se muestra la detección de una jibia (indicada por el *bounding box*), con algunos jureles a su alrededor. Debido a su tamaño, similar a la sierra, es poco probable que se capture por completo en una sola imagen con la configuración de zoom actual. Además, la jibia tienda a "esconderse" debajo del jurel, lo que hace que su tamaño visible mucho más variable que el resto de las especies.

3.3.2 Detección de *Keypoints*

En esta sección se mostrará el diseño y los resultados obtenidos para la segunda función de la etapa de discriminación de especies, la cual está encargada de identificar automáticamente los 8 *keypoints* seleccionados para cada instancia de pez detectada por el modelo de detección de objetos, para luego ser entregados al modelo de clasificación jerárquica final.

A. Entrenamiento de Modelos Fallidos

Antes de entrar en el detalle de los resultados, es importante recalcar que encontrar un modelo de detección de *keypoints* que se adapte de manera flexible a una base de datos personalizada y que sea viable para realizar inferencia, fuera de su entorno de entrenamiento original, representa un desafío no menor.

Desde el punto de la adaptabilidad del modelo a una base de datos personalizada, es crucial que el modelo sea capaz de trabajar y operar correctamente ante datos nuevos que se enfoquen en objetos más específicos. Hoy en día, muchos modelos preentrenados están optimizados para bases de datos estándar, principalmente, para reconocer la postura de un humano, por lo que no es posible que rindan igualmente bien con datos personalizados sin recibir una recalibración o reentrenamiento previo. Tal fue el caso de los modelos que se enlistan a continuación, los cuales no fue posible reentrenar:

- CenterNet, un modelo de detección de objetos que funciona mediante la identificación del centro de un objeto y sus *keypoints* asociados [60][17].
- Detectron2, perteneciente a FAIR, el cual incluye implementaciones de alta calidad de algoritmos de detección de objetos de última generación, incluidas algunas variantes de la familia de modelos Mask R-CNN y estimación de pose humana[61].
- YOLOv5 – *keypoints*, perteneciente a la familia de modelos de YOLO, pero añadiendo una rama adicional para la identificación de *keypoints* para cada *bounding box* detectado por el modelo[18].

No obstante, se encontró una versión estable del modelo Keypoint R-CNN (ver Sección 2.3.3 para mayores detalles), la cual está incluida dentro del *model zoo*[19] de la biblioteca PyTorch[20], y cumple tanto con ser flexible para realizar un reentrenamiento del modelo con una nueva base de datos, como con ser adaptable ante nuevos entornos de programación, siendo sencillo guardar los pesos del modelo y exportarlos a un entorno local. Este modelo en particular entrega las coordenadas (x, y) de cada *keypoint* detectado.

De esta forma, tomando en cuenta la utilización de Keypoint R-CNN para el diseño de los modelos de detección de *keypoints*, el entrenamiento y testeo de esta etapa se realizó siguiendo tres estrategias principales:

i. Utilizando la versión original de la base de datos **B03**, con múltiples peces por imagen.

[17] No contaba con un código reproducible para realizar el reentrenamiento.
[18] Si contaba con un código, pero solo estaba preparado para trabajar con *keypoints* de poses humanas y, por ende, no fue posible adaptarlo a la base de datos de peces.
[19] Repositorio que contiene modelos previamente entrenados para diversas tareas de aprendizaje automático. Estos modelos se entrenan en grandes conjuntos de datos y están listos para implementarse o ajustarse para tareas específicas.
[20] Biblioteca de Python utilizada para aplicaciones de aprendizaje automático y visión artificial, desarrollada por FAIR.

ii. Utilizando una versión de la base de datos **B03** con un único pez por imagen, segmentado en un fondo negro.
iii. Utilizando una versión de la base de datos **B03** con un único pez por imagen, segmentado en un fondo *random*.

B. Primera Estrategia de Entrenamiento

La primera estrategia de entrenamiento consiste en probar la capacidad del modelo para detectar *keypoints* en una imagen donde se cuenta con múltiples individuos de peces a la vez. Luego, una vez entrenado el modelo, interesa probar la capacidad de generalización de este para predecir *keypoints* en imágenes diferentes a las ya utilizadas durante el entrenamiento, considerando para ello las bases de datos de las estrategias ii) y iii).

El entrenamiento fue realizado con las siguientes configuraciones:

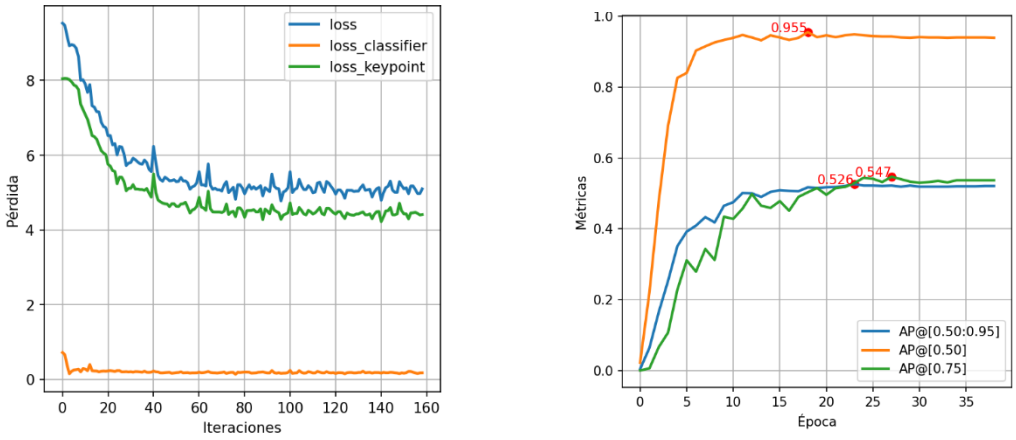
- Distribución 70%-30% para los sets de entrenamiento y validación, respectivamente.
- Imágenes RGB de tamaño 960x960, reescaladas usando *letterboxing*.
- *Anchors* de tamaños [32, 64, 128, 256, 512] y proporciones [0.25, 0.5, 0.75, 1.0, 2.0, 3.0, 4.0], ajustados experimentalmente para encontrar peces de diferentes tamaños.
- *Learning rate* variable en forma de escalón, siguiendo el *scheduler* mostrado en el Anexo A.5.3, con un mínimo de 1e-5 y un máximo de 1e-3.
- Número total de épocas ajustado manualmente hasta 40 (no cuenta con *Early Stopping*).
- No se aplica *data augmentation* en línea.

Particularmente, respecto de la última configuración, la técnica de *data augmentation* funciona bien para entrenar modelos que solo dependen de imágenes, pero, al trabajar con keypoints, se corre el riesgo de perder la referencia correcta de algunos de ellos

al aplicar transformaciones que los lleven fuera del límite máximo, dado por las dimensiones de la imagen. Por ende, es preferible no utilizarla en esta oportunidad.

El resumen de los resultados obtenidos durante la validación del modelo se presenta a continuación. La Figura N° 3.12 muestra las curvas de pérdida generadas con el *set* de entrenamiento en el apartado a y las curvas de la métrica AP[21] generadas con el *set* de validación en el apartado b. Se puede observar que el modelo alcanza un punto óptimo de la función de pérdida general (curva 🔵) alrededor de las iteraciones 40-80 (equivalente a las épocas 10-20 en el gráfico de las curvas AP), siendo esta la suma de la función de pérdida de clasificación (curva 🟠) y la función de pérdida de *keypoints* (curva 🟢). Particularmente, la función de pérdida de clasificación es mucho más baja que la de *keypoints* debido a que el modelo logra diferenciar fácilmente a los peces de su fondo. Respecto de la función de pérdida de *keypoints*, si bien se aprecia que esta se estanca en un valor por sobre 4 (potencialmente debido a un *learning rate* muy pequeño), la métrica AP@[0.5][22] (curva 🟠) muestra que se alcanza un máximo de 0.955, demostrando un buen rendimiento del modelo para un IOU de 0.5. Sin embargo, dado que la métrica AP@[0.5:0.95] (curva 🔵) solo alcanza un máximo de 0.526, esto indica que el modelo disminuye su rendimiento cuando se somete a IOU altos, lo cual es un indicador de que el modelo no opera correctamente cuando los peces se encuentran muy próximos entre sí, donde existe el riesgo de que una predicción del modelo abarque más de un pez a la vez o se generen demasiados *bounding boxes* intermedios entre los peces más cercanos.

[21] Equivalente a la métrica mAP dado que el modelo solo trabaja con una clase.
[22] Notación que se usa frecuentemente en paradigmas de detección de objetos para calcular la métrica AP para diferentes valores o rangos del parámetro IOU (lo que sigue del @). Cuando se trabaja con rangos, la curva AP se muestra como el promedio para todos los IOU evaluados.



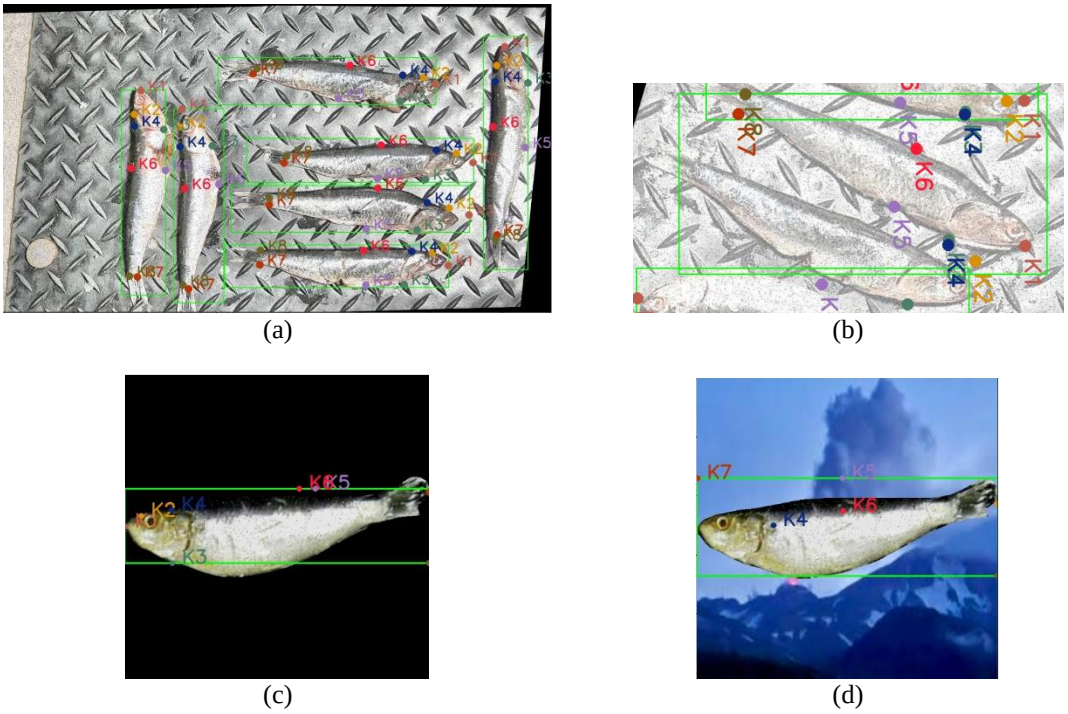
(a) (b)

*Figura N° 3.12 Resultados de entrenamiento y validación para la primera estrategia del modelo Keypoint R-CNN, considerando la base de datos **B03**. a) Evolución de la función de pérdida. b) Evolución de las métricas AP para bounding box. Fuente: [Elaboración propia]*

De manera adicional, en la Figura N° 3.13 se muestran 4 imágenes con los resultados de inferencia del modelo. En el apartado a se muestra una imagen del set de validación que cuenta con 7 anchovetas, donde todas fueron identificadas correctamente junto con sus respectivos *keypoints*, mientras que en el apartado b se muestra un ejemplo menos favorable, donde el *bounding box* generado para una anchoveta abarca otros dos ejemplares, confundiendo los *keypoints* K_2-K_4. Además, en los apartados c y d se incluyen dos variantes para una sardina que fue segmentada de su fondo original. Si bien el apartado c muestra una imagen en un entorno más simple, con un fondo negro, tanto la generación del *bounding box* como de los *keypoints* no se ajustan a lo esperado. Lo mismo ocurre en el apartado d, considerando un fondo más complejo, donde el *bounding box* sí es generado correctamente, pero no así los *keypoints*.

Si bien los resultados anteriores muestran el rendimiento general alcanzado por el modelo durante el entrenamiento, la métrica AP no entrega información suficiente para concluir sobre qué tan bien o mal se están generando los *keypoints* respecto del *ground truth* (la ubicación real de los peces y sus respectivos *keypoints*). Para ello, la Tabla N°

3.2 incluye un resumen con las métricas MAE y PKC evaluadas sobre distintos sets de validación. Respecto de la métrica PKC, el umbral de desviación máxima de los *keypoints* se fijó experimentalmente en un valor de 40 píxeles en todas las direcciones, para todas las estrategias.



(a) (b) (c) (d)

Figura N° 3.13 Ejemplo de imágenes procesadas con Keypoint R-CNN para la primera estrategia de entrenamiento. a) Imagen con múltiples muestra de anchoveta. b) Detección de anchoveta que involucra 3 muestras dentro de un mismo bounding box. c) Imagen de una sardina segmentada en un fondo negro. d) Imagen de una sardina segmentada en un fondo random. Fuente: [Elaboración propia]

Por un lado, la sección ● de la tabla muestra las métricas logradas para el set de validación original del modelo, donde la predicción del *keypoint* K_6 alcanza la menor distancia promedio respecto del *ground truth*, con un MAE de 45.29, y la predicción del *keypoint* K_2 alcanza la mayor distancia promedio, con un MAE de 63.82. Respecto de la predicción general de los *keypoints* (métrica PKC), el modelo predice correctamente el 70.17% de la totalidad de los *keypoints*.

Por otro lado, las secciones 🟡 y 🟠 muestran las métricas logradas para los sets de validación con los peces individuales, donde el *keypoint* **K_6** se mantiene respecto del caso anterior como aquel que logra los MAE más pequeños. Sin embargo, para ambos casos, se da que la métrica PKC disminuye bajo el 30% para una misma desviación máxima de 40 píxeles, indicando que el modelo generaliza pobremente los *keypoints* para la mayoría de las imágenes de peces individuales.

*Tabla N° 3.2 Resumen de las métricas de desempeño obtenidas para la validación de la primera estrategia del modelo Keypoint R-CNN, considerando la base de datos **B03**.*

	Modelo *keypoints* múltiple – imagen original							
Métrica	**K_1**	**K_2**	**K_3**	**K_4**	**K_5**	**K_6**	**K_7**	**K_8**
MAE	62.35	63.82	61.03	50.06	61.98	**45.29**	46.44	45.85
MAE promedio	54.60							
PKC	70.17%							
	Modelo *keypoints* múltiple – peces segmentados en fondo negro							
Métrica	**K_1**	**K_2**	**K_3**	**K_4**	**K_5**	**K_6**	**K_7**	**K_8**
MAE	133.33	104.86	58.61	69.27	59.71	**56.94**	82.79	69.31
MAE promedio	76.85							
PKC	27.05%							
	Modelo *keypoints* múltiple – peces segmentados en fondo *random*							
Métrica	**K_1**	**K_2**	**K_3**	**K_4**	**K_5**	**K_6**	**K_7**	**K_8**
MAE	130.41	120.07	74.21	72.86	56.87	**54.31**	95.44	81.87
MAE promedio	85.76							
PKC	25.85%							

Lo anterior demuestra que, si bien es posible entrenar un modelo de detección de *keypoints* que pueda detectar múltiples especies de peces en una misma imagen, la cercanía entre los mismos es potencialmente un riesgo que puede generar resultados erróneos al traslapar la información tanto respecto de los *bounding boxes* como de los *keypoints* entre distintas muestras, lo cual debe evitarse para no generar resultados erróneos en el modelo de clasificación. Si bien esto puede resolverse modificando la fase de detección global que se aplica dentro de la modalidad de detección *top-down* (ver Sección 2.3**¡Error! No se encuentra el origen de la referencia.**), considerando que YOLOv7 ya e s capaz de entregar buenos resultados de peces segmentados individualmente, el modelo de *keypoints* no es lo suficientemente generalizable aún como para detectar correctamente

los *keypoints* en imágenes de peces individuales, puesto que son imágenes desconocidas para el modelo. No obstante, esta estrategia en particular es útil para construir una herramienta de etiquetado automático de *keypoints* para imágenes de múltiples peces en condiciones similares, facilitando así un proceso que manualmente puede tardar entre 1 a 5 minutos por imagen, dependiendo de la cantidad de objetos de interés y su distribución.

C. Segunda Estrategia de Entrenamiento

En base a los resultados anteriores, la idea detrás de esta segunda estrategia es entrenar un modelo de detección de *keypoints* que se adapte de mejor manera a los peces que ya fueron detectados y segmentados por el modelo detección de objetos, utilizando para ello una base de datos más específica con los peces separados individualmente y luego dispuestos sobre un fondo negro. Adicionalmente, también interesa probar la capacidad de generalización del modelo para imágenes diferentes a las utilizadas durante el entrenamiento, considerando particularmente las imágenes de la estrategia iii).

Respecto de la segmentación de los peces a partir de las imágenes, esta puede realizarse tanto a partir del *bounding box* como de la máscara utilizados para etiquetar la base de datos. Teniendo en mente que con la segmentación desde un *bounding box* se corre el riesgo de que partes de otros peces se infiltren dentro de una misma imagen, se prefirió utilizar la máscara para llevar a cabo la segmentación, a pesar de que esta operación es más lenta y menos directa que utilizando un *bounding box*.

Respecto de la elección del fondo negro como tal, esta es una estrategia común que se utiliza para eliminar la atención de los modelos por sobre el fondo y así aislar los objetos de interés. Si bien, en teoría, cualquier color podría utilizarse para este propósito, se eligió el color negro para generar un alto contraste respecto de los colores naturales de los peces, lo cual ayuda a que el modelo se centré en los atributos locales de los objetos (los más específicos), facilitando su aprendizaje y fomentando una mejor capacidad de generalización.

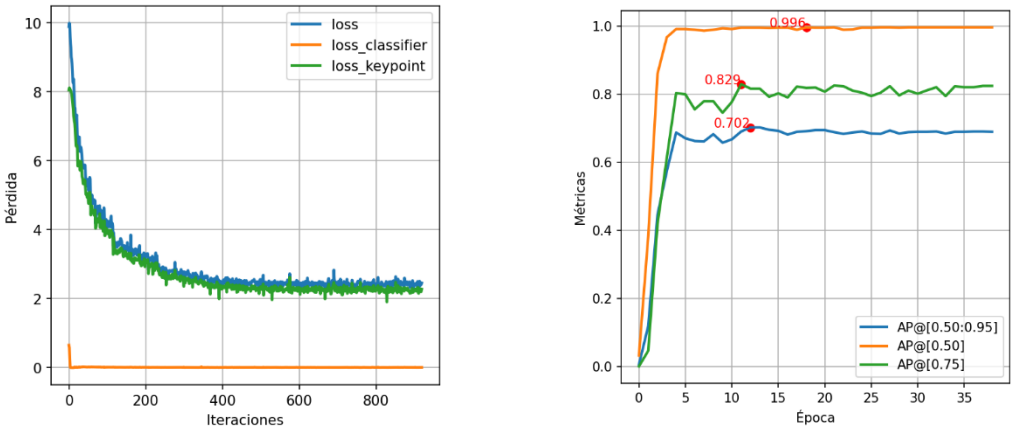
Es entonces como el entrenamiento fue realizado utilizando las mismas configuraciones de la estrategia anterior, pero considerando las siguientes modificaciones:

- Imágenes RGB de tamaño 256x256, reescaladas utilizando *letterboxing*.
- *Anchors* de tamaños [4, 8, 16, 32, 64] y proporciones [0.25, 0.5, 0.75, 1.0, 2.0].

El resumen de los resultados obtenidos durante la validación del modelo se presenta a continuación. La Figura N° 3.14 muestra las curvas de pérdida generadas con el *set* de entrenamiento en el apartado a y las curvas de la métrica AP generadas con el *set* de validación en el apartado b. Se puede notar que, a diferencia de la primera estrategia, la curva de pérdida general (curva ●) tarda un mayor número de iteraciones en alcanzar el punto óptimo (debido a que el número de imágenes aumentó), pero, a su vez, la curva de la métrica AP@[0.5:0.95] (curva ●) alcanza valores cercanos a su máximo (0.702) en un menor número de épocas (desde la época 4). Considerando que el máximo de la métrica AP@[0.5:0.95] resultó ser 0.176 puntos mayor que en la primera estrategia, esto demuestra que la simplificación de la base de datos mejora el rendimiento del modelo al reducir la posibilidad de la generación de detecciones intermedias, aunque es importante notar que este comportamiento no es posible eliminarlo del todo, debido a que es el comportamiento natural de los modelos de detección que utilizan *anchors* para encontrar a los objetos dentro de una imagen.

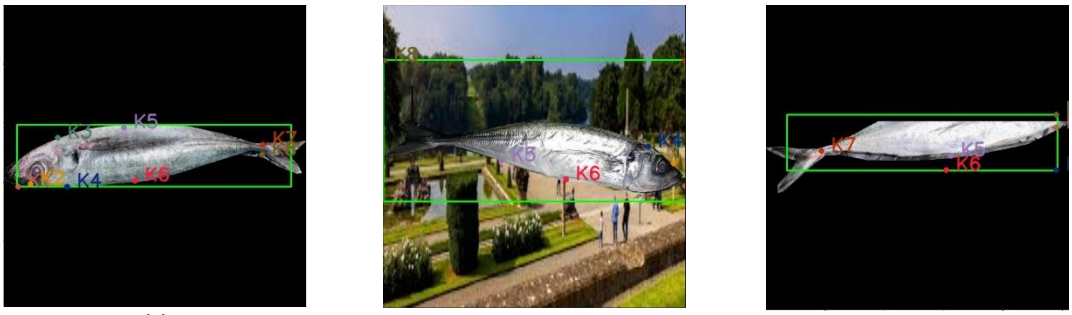
La Figura N° 3.15 muestra algunos resultados obtenidos con la inferencia del modelo. En el apartado a se muestra una imagen de jurel presente en el set de validación de la base de datos, donde se aprecia que 7 de los 8 *keypoints* fueron detectados correctamente, pero el *bounding box* no logra encerrar al pez completo. El *keypoint* mal identificado en este caso fue **K_1**, el cual se acopló en el extremo inferior izquierdo del *bounding box* intentando buscar el punto más lejano donde comienza la boca del pez (lo cual es técnicamente correcto, puesto que un *keypoint* no puede estimarse fuera del *bounding box*). En el apartado b se muestra una imagen con un jurel segmentado en un fondo *random*, donde se da el caso de que solo 2 de los 8 *keypoints* fueron detectados correctamente,

principalmente debido a que el modelo todavía no es capaz de generalizar correctamente imágenes con fondos tan complejos, a pesar de que la forma, texturas y colores de los peces se mantienen intactos. En el apartado c se muestra un caso más extremo de la situación mostrada en el apartado a, donde es la cabeza del pez la cual se encuentra fuera de los límites de la imagen original. Debido a la restricción del modelo a siempre generar una predicción con la localización de todos los *keypoints*, la distribución de estos se ve forzada a alejarse del esqueleto original del *box truss* para adaptarse a la información disponible, errando 6 de los 8 *keypoints*. En general, la elección de los *keypoints* es ambigua para imágenes donde no se aprecia la morfología completa de un pez, por lo que es recomendable evitar este tipo de imágenes en la base de datos o, en su defecto, filtrar los esqueletos para que estos no sean entregados al modelo de clasificación, utilizando por ejemplo la silueta del esqueleto como referencia. Por simplicidad, se eligió la primera alternativa.



(a) (b)

Figura N° 3.14 Resultados de entrenamiento y validación para la segunda estrategia del modelo Keypoint R-CNN, considerando la base de datos ***B03****. (a) Evolución de la función de pérdida. (b) Evolución de las métricas AP para bounding box. Fuente: [Elaboración propia]*



(a) (b) (c)

Figura N° 3.15 Ejemplo de imágenes procesadas con Keypoint R-CNN para la segunda estrategia de entrenamiento. a) Imagen de un jurel segmentado en un fondo negro. b) Imagen parcial de un jurel segmentado en un fondo negro. c) Imagen de un jurel segmentado en un fondo random. Fuente: [Elaboración propia]

De manera complementaria, la Tabla N° 3.3 muestra las métricas obtenidas respecto de la distribución de los *keypoints*. Al igual que para la estrategia anterior, la métrica PKC obtenida durante la validación del modelo para imágenes con fondo negro (sección 🔵) se encuentra muy cercana al 70%, siendo esta vez el *keypoint* **K_2** el que logra el menor distanciamiento respecto del *ground truth*, con un MAE de 20.11; el más bajo obtenido hasta el momento. Respecto de la prueba del modelo para imágenes con fondo *random* (sección 🟡), los resultados fueron aún peores que la estrategia anterior, aumentando el MAE promedio (considerando todos los *keypoints*) de 85.76 a 159.22, y disminuyendo la métrica PKC por debajo del 23%.

*Tabla N° 3.3 Resumen de las métricas de desempeño obtenidas para la validación de la segunda estrategia del modelo Keypoint R-CNN, considerando la base de datos **B03**.*

	Modelo *keypoints* individual – peces segmentados en fondo negro							
Métrica	**K_1**	**K_2**	**K_3**	**K_4**	**K_5**	**K_6**	**K_7**	**K_8**
MAE	22.21	**20.11**	27.26	25.39	33.57	32.70	33.21	24.42
MAE promedio	27.36							
PKC	70.59%							
	Modelo *keypoints* individual – peces segmentados en fondo *random*							
Métrica	**K_1**	**K_2**	**K_3**	**K_4**	**K_5**	**K_6**	**K_7**	**K_8**
MAE	169.45	117.52	200.16	118.89	160.99	**100.45**	199.99	206.32
MAE promedio	159.22							
PKC	22.79%							

Lo anterior demuestra que el modelo sí es capaz de detectar *keypoints* de manera satisfactoria cuando se segmentan los peces individualmente sobre un fondo negro, pero todavía existe espacio de mejora para diseñar un modelo robusto que se adapte de mejor forma a imágenes con un fondo dentro de un contexto más real, como lo sería en el ambiente industrial de no contar con un modelo de detección de objetos que realice segmentación por instancias.

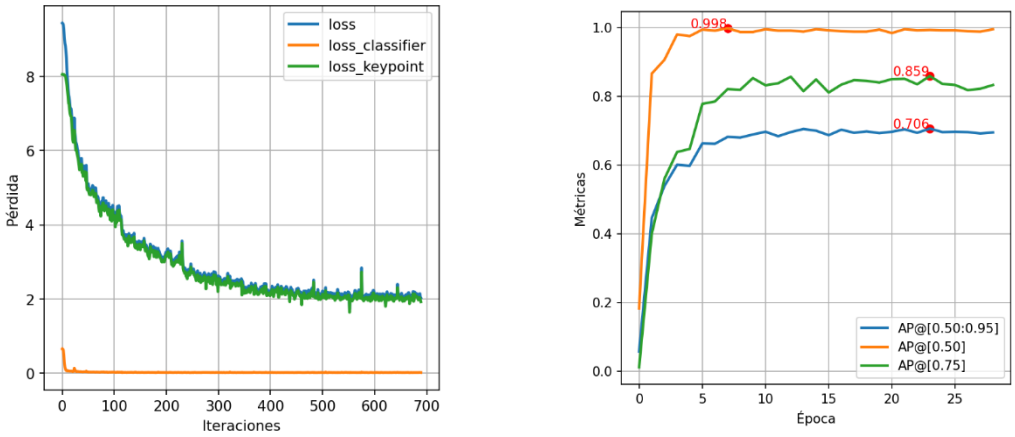
D. Tercera Estrategia de Entrenamiento

La idea detrás de esta tercera estrategia es probar qué tanto mejora el rendimiento de un modelo de detección de *keypoints* entrenado utilizando imágenes de peces individualmente segmentados sobre un fondo *random*. Adicionalmente, también interesa probar la capacidad de generalización del modelo en imágenes diferentes a las utilizadas durante el entrenamiento, considerando para ello las imágenes de la base de datos **B04**, también en su variante con los peces segmentados individualmente sobre un fondo *random*.

La elección de utilizar los fondos *random* (presentados en el Anexo A.6.2.2.) por sobre fondos de un único color o fondos sintéticos creados digitalmente, radica en mejorar la capacidad del modelo para generalizar situaciones del mundo real, donde los fondos pueden variar enormemente y es difícil asegurar que todas las condiciones aparezcan en las imágenes [62]. De esta forma, exponer al modelo a diferentes contextos o escenarios durante el entrenamiento facilita su capacidad para enfocarse en los atributos relevantes de los objetos de interés (que guardan mayor similitud entre sí) y a no depender en exceso de las características específicas del fondo, considerándolas como un ruido [63]. Esto, a su vez, aumenta tanto la robustez del modelo, permitiéndole adaptarse mejor a diversas condiciones, como su capacidad para evitar el sobreajuste al prevenir que el modelo "memorice" los patrones o características de un único fondo en específico, como ocurre con los fondos negros.

Particularmente, el entrenamiento del modelo fue realizado utilizando las mismas configuraciones de la estrategia anterior, únicamente modificando los fondos de las imágenes, considerando un fondo diferente para todos los casos.

El resumen de los resultados obtenidos durante la validación del modelo se presenta a continuación. La Figura N° 3.16 muestra las curvas de pérdida generadas con el *set* de entrenamiento en el apartado a y las curvas de la métrica AP generadas con el *set* de validación en el apartado b, desde donde se puede observar que son muy similares a las obtenidas para la estrategia anterior, para todos los casos. Esto es esperable, debido a que la posición de los peces no cambia realmente dentro de las imágenes, pero demuestra que los fondos aleatorios no generan el problema de detecciones intermedias, como sí ocurría en el entrenamiento de la primera estrategia.

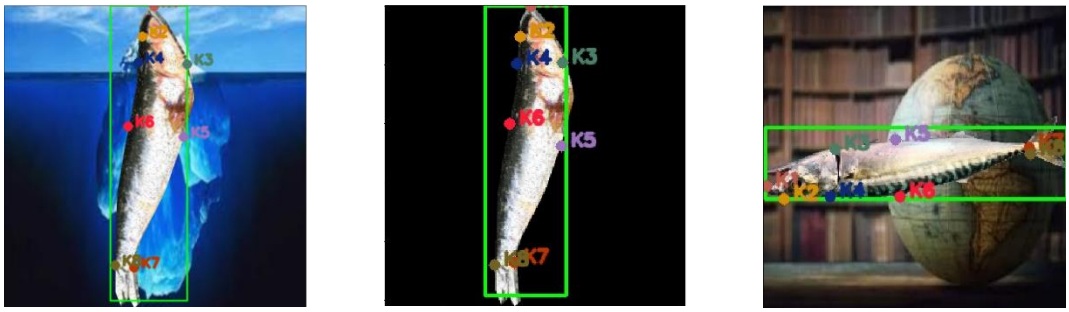


(a) (b)

*Figura N° 3.16 Resultados de entrenamiento y validación para la tercera estrategia del modelo Keypoint R-CNN, considerando la base de datos **B03**. (a) Evolución de la función de pérdida. (b) Evolución de las métricas AP para bounding box. Fuente: [Elaboración propia]*

La Figura N° 3.17 muestra algunos resultados obtenidos con la inferencia del modelo. En los apartados a y b se muestran imágenes de una misma anchoveta donde únicamente se varía su fondo, desde donde se puede observar que los 8 *keypoints* son identificados correctamente para ambos casos, a pesar de que el modelo no conoce imágenes con

fondo negro. Lo mismo ocurre para la imagen del apartado c, donde el modelo logra identificar correctamente los 8 *keypoints* para una caballa, también desconocida para el modelo.



(a) (b) (c)

Figura N° 3.17 Ejemplo de imágenes procesadas con Keypoint R-CNN para la tercera estrategia de entrenamiento. a) Imagen de una anchoveta segmentada en un fondo random. b) Imagen de una anchoveta segmentada en un fondo negro. c) Imagen de una caballa, desconocida para la base de datos, segmentada en un fondo random. Fuente: [Elaboración propia]

El excelente comportamiento del modelo ante imágenes desconocidas para la base de datos se puede respaldar a partir de las métricas que se incluyen en la Tabla N° 3.4, donde se observa que las métricas PKC alcanzan valores que rondan el 90% para todas las variantes de imágenes con las cuáles fue testeado el modelo, demostrando ser más robusto que las dos estrategias anteriores. Respecto de las métricas MAE, se observa como el MAE promedio para los peces con fondo negro (sección) alcanza el valor más bajo de todos, con un 15.83, lo cual se puede justificar debido a que los fondos negros son naturalmente más simples de identificar que los fondos *random*, afectando en menor proporción a la localización de los *keypoints.*

Otra prueba adicional para demostrar la robustez del modelo consistió en testearlo únicamente con las imágenes *random* utilizadas para definir los fondos de las bases de datos, resultando en solo un 1% de imágenes con detecciones con un umbral de confianza sobre el 70% para un total de 1680 imágenes diferentes.

*Tabla N° 3.4 Resumen de las métricas de desempeño obtenidas para la validación de la tercera estrategia del modelo Keypoint R-CNN, considerando la base de datos **B03**.*

	Modelo *keypoints* individual *random* – peces segmentados en fondo *random*							
Métrica	**K_1**	**K_2**	**K_3**	**K_4**	**K_5**	**K_6**	**K_7**	**K_8**
MAE	**9.19**	12.64	20.54	11.99	23.91	17.21	29.91	21.36
MAE promedio	18.35							
PKC	89.89%							
	Modelo *keypoints* individual *random* – peces segmentados en fondo negro							
Métrica	**K_1**	**K_2**	**K_3**	**K_4**	**K_5**	**K_6**	**K_7**	**K_8**
MAE	13.19	13.01	11.78	13.46	26.43	19.85	18.41	**10.48**
MAE promedio	15.83							
PKC	90.58%							
	Modelo *keypoints* individual *random* – caballas							
Métrica	**K_1**	**K_2**	**K_3**	**K_4**	**K_5**	**K_6**	**K_7**	**K_8**
MAE	22.87	20.45	13.66	**8.65**	24.24	18.35	11.21	21.55
MAE promedio	17.62							
PKC	90.97%							

E. Resumen de los Resultados

El modelo de detección de *keypoints* resultó ser exitoso tanto para imágenes con múltiples peces como para imágenes con un único pez. Como ya se mencionó anteriormente, si bien se tiene un modelo de detección de objetos que es mejor detectando la localización de múltiples peces en una imagen, siendo robusto incluso cuando los peces se solapan unos con otros, el modelo de *keypoints* para múltiples especies es útil como una herramienta dedicada para etiquetar *keypoints* en peces, la cual se puede reentrenar para ajustarse a nuevas bases de datos. Respecto del modelo para un único pez, el modelo entrenado con fondos *random* demuestra ser más robusto que el resto, entregando una mayor flexibilidad a la hora de manipular el flujo de las detecciones a lo largo de la etapa de discriminación. Por ejemplo, el modelo de *keypoints* puede entrenarse con fondos *random*, pero la inferencia puede realizarse sin problemas con un fondo negro utilizando como base la segmentación de los peces a partir de su *bounding box* o de su máscara, considerando que la segmentación por *bounding box* es mucho más rápida que la segmentación por máscaras, y estas a su vez son más rápidas de integrar para un fondo negro que para un fondo *random*.

3.4. Diseño de los Modelos de Clasificación

En esta sección se mostrará el diseño y los resultados obtenidos para la tercera función de la etapa de discriminación de especies, la cual está encargada de realizar una doble verificación de la clasificación que ya viene incorporada con el modelo de detección de objetos.

La idea detrás de esta estrategia recae en el hecho de que muchos de los modelos del estado del arte, como VGG16 o YOLO, ya han sido entrenados previamente en conjuntos de datos extensos y diversificados, lo que les permite reconocer una amplia gama de características de diferentes objetos. Si bien esto puede ser una ventaja en términos de la versatilidad y la capacidad de generalización resultantes, también puede convertirse en un inconveniente cuando se intenta realizar un ajuste fino con una base de datos específica y más limitada. En situaciones donde los entrenamientos con un conjunto de datos particular no arrojan resultados satisfactorios, como lo es el caso de las especies de peces problemáticas, la complejidad de las características y patrones aprendidos durante el entrenamiento inicial del modelo original no necesariamente se alinea con las características únicas y particulares de la nueva base de datos con la que se está trabajando, y se torna complejo saber qué capa o parámetros específicos es necesario modificar del modelo original.

Una posible respuesta a la problemática planteada se encuentra en reemplazar la etapa de extracción de características de los modelos tradicionales por una que responda a una taxonomía mucho más específica para la clasificación de especies de peces, estableciendo una jerarquía desde las características más generales hacia las características más específicas. En este sentido, y a diferencia de un clasificador multiclase tradicional, el modelo de clasificación busca combinar algunas de las características más relevantes que se pueden extraer a partir de los 8 *keypoints* seleccionados para las 4 especies problemáticas, y con ello construir un clasificador jerárquico que tenga la estructura de árbol mostrada anteriormente en la Sección 2.5.3.

A continuación, se muestra el detalle de los cuatro nodos o minimodelos del árbol que dan respuesta a una pregunta de clasificación específica para poder diferenciar finalmente una imagen entre anchoveta, caballa, jurel o sardina. Para simplificar el diseño de estos clasificadores, únicamente se consideraron modelos basados en FCNN o CNN. Adicionalmente, se añadió un minimodelo para diferenciar si una imagen corresponde a un pez o no, el cual servirá de condición para determinar cuándo una imagen debe o no ingresar al clasificador jerárquico. Finalmente, se incluyen los resultados obtenidos luego de combinar todos los minimodelos para generar la etiqueta de clasificación final del árbol a nivel de hojas (no considerando clasificaciones en niveles intermedios). Las pruebas realizadas incluyen una comparación entre las tres modalidades del clasificador jerárquico presentadas en la Sección 2.3.5 y un clasificador multiclase tradicional, considerando para ello tanto los modelos de clasificación del estado del arte mencionados en la Sección 2.3.4 como modelos basados en FCNN y CNN.

3.4.1 Minimodelo Fish/No Fish

Este primer minimodelo tiene como función distinguir entre objetos que son peces de aquellos que no lo son, sirviendo como un filtro para garantizar que solo las imágenes de peces auténticos se sometan al proceso de clasificación jerárquica. Si bien este fenómeno es reducido significativamente por los modelos de detección, siempre existe una pequeña probabilidad de que objetos ajenos, desconocidos para la base de datos, puedan infiltrarse al sistema. Por lo tanto, es necesario incorporar este filtro, no solo para incrementar la precisión en la clasificación final, sino también para alivianar la carga de procesamiento mediante la eliminación de imágenes no pertinentes.

Teniendo en mente el proceso de entrenamiento, el diseño de este minimodelo se realizó siguiendo dos estrategias:

iii. Utilizando una base de datos local de peces con fondos aleatorios (**B03**).
iv. Utilizando una base de datos de peces lo más general posible (**B05**).

Para las dos estrategias planteadas, y dada la complejidad inherente a las imágenes de la base de datos, se decidió implementar un modelo utilizando la técnica de *transfer learning* (TL) aplicada a una arquitectura pre-entrenada de VGG16 (ver Figura N° 3.18). En este proceso, se mantiene el *backbone* de extracción de características y se reemplazaron las capas originales de clasificación de VGG16 por una versión personalizada constituida por una capa *Global Average Pooling*, una capa de *dropout* y una capa densa con una función de activación de tipo *softmax* (ver Figura N° 3.19).

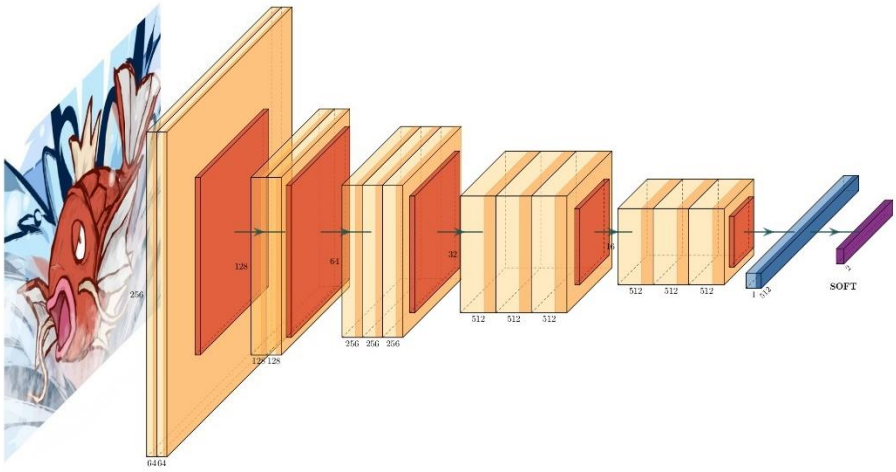
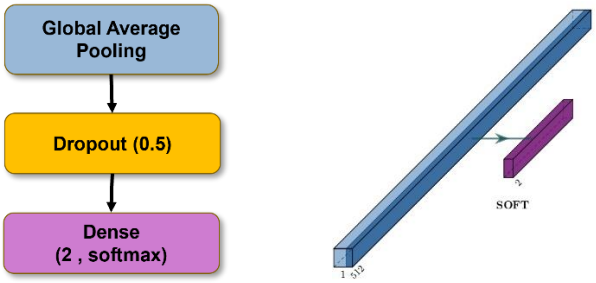


Figura N° 3.18 Arquitectura de VGG16 con etapa de clasificación modificada. Fuente: [Elaboración propia]



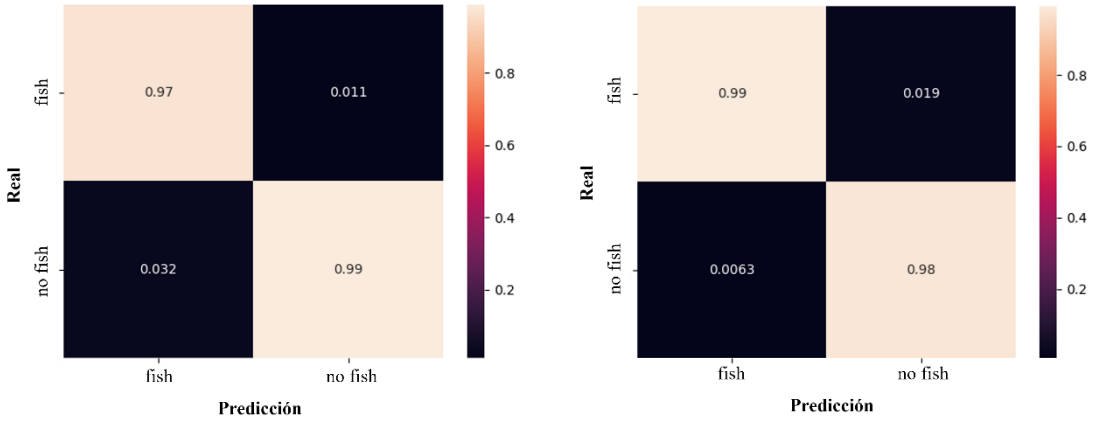
(a) (b)

Figura N° 3.19 Capas de Clasificación modificadas de VGG16. (a) Esquema 2D; (b) Esquema 3D; no se considera la capa dropout. Fuente: [Elaboración propia]

El entrenamiento de los minimodelos se realizó considerando las siguientes configuraciones para ambas estrategias:

- Distribución 70%-30% para los sets de entrenamiento y validación, respectivamente.
- Imágenes RGB de tamaño 256x256 utilizando fondos aleatorios.
- *Data augmentation* en línea relacionado con cambios de brillo, desplazamientos horizontales y verticales y rotaciones; aplicados a cada imagen durante el entrenamiento.
- *Batch* de 32 imágenes por iteración.
- *Early Stopping* ajustado para un total de 150 épocas.
- *Learning rate* constante, ajustado experimentalmente dentro del rango óptimo [1e-3, 5e-3], según el método propuesto en el Anexo A.5.3.

El resumen de los resultados obtenidos durante la validación de los minimodelos se muestra tanto en las matrices de confusión normalizadas por columnas de la Figura N° 3.20 como en la Tabla N° 3.5. Se puede observar que en ambos casos el modelo alcanza valores de la métrica P notablemente altos, aun cuando la variabilidad de los objetos que no corresponden a un pez es muy diferente, por ejemplo, considerando objetos tan comunes como una mesa o una botella de plástico. Particularmente, se destaca la métrica P de 0.99 alcanzada por la clase "*no fish*" para la base de datos local, y de 0.99 alcanzada por la clase "*fish*" para la base de datos global, mientras que para la métrica R los mejores modelos se invierten por una diferencia mínima. Respecto de la métrica MP, el modelo para la base de datos global logra el mejor resultado, siendo este de 0.99.



(a) (b)

*Figura N° 3.20 Matrices de confusión para las dos estrategias del minimodelo Fish/No Fish. (a) Base de datos local, **B03**; (b) Base de datos global, **B05**. Fuente: [Elaboración propia]*

Sin embargo, una prueba adicional fue realizar una validación cruzada entre ambas bases de datos, testeándolas con el modelo contrario que no las utilizó durante su entrenamiento. Tal y como se puede apreciar en la Tabla N° 3.6, esta prueba resulta en un descenso de la métrica MP para ambos modelos, alcanzando un valor máximo de 0.81 para el minimodelo entrenado originalmente con la base de datos local. Si bien para ambos modelos se observa que para predecir un pez en una imagen la métrica P se mantiene en valores sobre 0.90, el valor de la métrica R es mucho más bajo, particularmente para el modelo entrenado con la base de datos global, con un R de 0.39. Esto es un claro indicador de que la mayoría de las imágenes donde si hay pescados son clasificadas erróneamente, lo cual repercute negativamente en la métrica P para predecir objetos distintos a un pez, siendo más baja para el modelo entrenado originalmente con la base de datos global, con un P de 0.62. Este efecto es interesante debido a que ambos modelos utilizan exactamente las mismas imágenes para representar la clase "*no fish*", por lo que se esperaría una precisión mucho más alta en ambos casos, lo cual no se cumple. Esto entrega un indicio de que los modelos están generalizando mucho mejor las características de los peces por sobre la del resto de objetos, lo cual puede deberse al hecho que la clase "*no fish*" en concreto es demasiado general respecto de clase "*fish*", que es más específica. Para fines prácticos,

el minimodelo entrenado con la base de datos local es el que puede potencialmente entregar mejores resultados que el minimodelo global, siempre y cuando este se vaya reentrenando constantemente para adaptarse correctamente ante nuevas imágenes desconocidas para la base de datos.

*Tabla N° 3.5 Resumen de las métricas de desempeño obtenidas para la validación del minimodelo Fish/No Fish considerando las bases de datos **B03** y **B05**.*

	Minimodelo local con base de datos **B03**			
Clase	**precision**	**recall**	**F1**	**MP**
Fish	0.97	**0.99**	0.98	0.98
No Fish	**0.99**	0.97	0.98	

	Minimodelo global con base de datos **B05**			
Clase	**precision**	**recall**	**F1**	**MP**
Fish	**0.99**	0.98	0.99	0.99
No Fish	0.98	**0.99**	0.99	

*Tabla N° 3.6 Resumen de las métricas de desempeño obtenidas para la validación del minimodelo Fish/No Fish considerando las bases de datos **B03** y **B05** intercambiadas.*

	Minimodelo local testeado con base de datos **B05**			
Clase	**precision**	**recall**	**F1**	**MP**
Fish	**0.90**	0.65	0.75	0.81
No Fish	0.73	**0.93**	0.81	

	Minimodelo global testeado con base de datos **B03**			
Clase	**precision**	**recall**	**F1**	**MP**
Fish	**0.94**	0.39	0.56	0.78
No Fish	0.62	**0.98**	0.76	

3.4.2 Minimodelo Tamaño

Este minimodelo tiene por objetivo clasificar a los peces en función de su tamaño, sirviendo como nodo raíz del árbol de clasificación jerárquica, tal y como se muestra en la Figura N° 3.21.

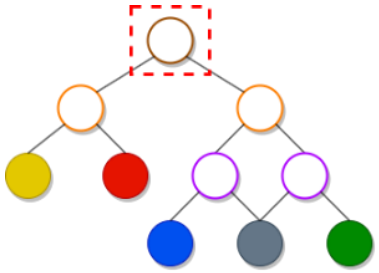
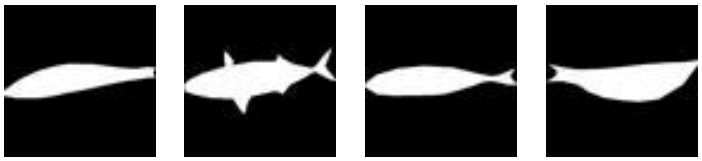


Figura N° 3.21 Árbol de clasificación jerárquica simplificado. El cuadro punteado denota el minimodelo de interés, en este caso el de tamaño. Fuente: [Elaboración propia]

Entrando en el detalle de la preparación de los datos, para este minimodelo se utilizarán dos tipos de entradas: algunas características morfométricas que pueden extraerse a partir de los *keypoints*, y una silueta con la forma de cada especie de pez.

En total se utilizarán 21 mediciones morfométricas, entre distancias y ángulos, las cuáles se describen con mayor detalle en la Sección 2.5.5A.

Respecto de las imágenes que contienen la silueta de los peces, un ejemplo de las imágenes que se utilizarán se muestra en la Figura N° 2.26. Si bien el uso de siluetas de objetos puede ser un método factible para enseñar a un modelo a reconocer la forma de los objetos [63], también es interesante probar si acaso el modelo es capaz de aprender sobre el tamaño de los objetos a partir de esta información. Debido a que las imágenes fueron reescaladas, eliminando su relación de tamaño original, se espera que el modelo solo se base en la sensación visual inversa de tamaño resultante del reescalado para realizar la clasificación, donde las especies pequeñas tienden a ocupar más espacio en la imagen que las especies grandes.



(a) (b) (c) (d)

*Figura N° 3.22 Comparación entre las siluetas de los peces tanto de la base de datos **B03** como **B04**. a) anchoveta; b) caballa; c) jurel; d) sardina. Fuente: [Elaboración propia]*

Teniendo en mente el proceso de entrenamiento, el diseño de este minimodelo se realizó siguiendo tres estrategias:

i. Utilizando solo las 21 características morfométricas extraídas de los peces tanto de las bases de datos **B03** y **B04**.
ii. Utilizando solo las imágenes con las siluetas de los peces tanto de las bases de datos **B03** y **B04**.
iii. Utilizando ambos tipos de características.

Para la primera estrategia, dado que solo se están utilizando datos numéricos unidimensionales, basta con utilizar una arquitectura de tipo FCNN para entrenar al minimodelo. Particularmente, luego de varias pruebas para definir experimentalmente el número apropiado de neuronas y capas, se utilizará la arquitectura mostrada en la Figura N° 3.23, donde se cuenta con dos capas densas intercaladas con capas de *dropout* y una capa densa con capa de activación *softmax* para generar la clasificación final.

Para la segunda estrategia, además de probar con la arquitectura pre-entrenada de VGG16 utilizada para el minimodelo anterior, también se probó con la arquitectura de tipo CNN simplificada mostrada en la Figura N° 3.24, la cual cuenta solo con 3 capas convolucionales intercaladas con capas de *pooling*. Esta prueba decidió realizarse debido a que las imágenes de siluetas son mucho más simples que las imágenes reales de peces que presentan objetos, formas y texturas más complejas, por lo que, en teoría, no se necesita una gran cantidad de filtros o capas para extraer características relevantes.

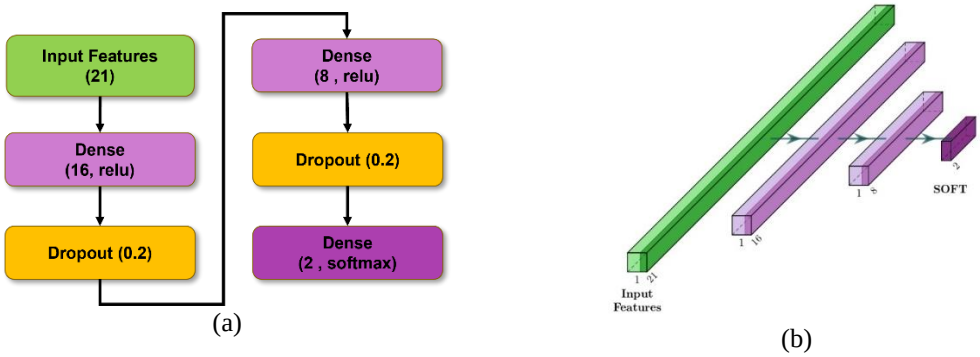


Figura N° 3.23 FCNN para la clasificación de tamaño considerando como entrada las 21 características morfológicas extraídas de un pez. (a) Esquema 2D; (b) Esquema 3D; no se consideran las capas dropout. Fuente: [Elaboración propia]

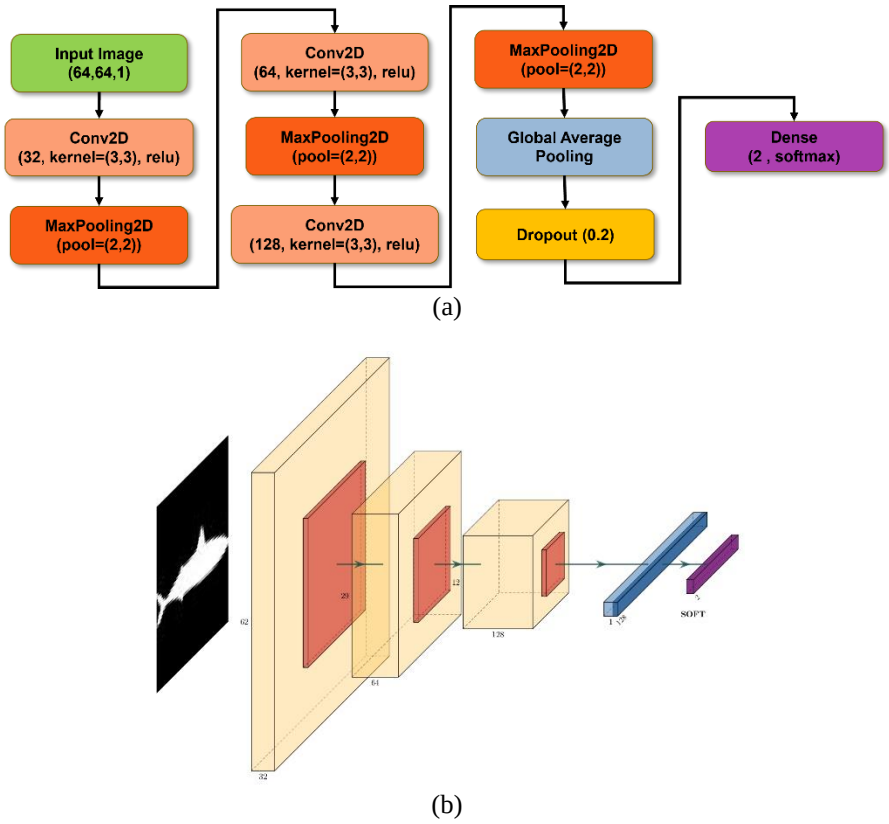
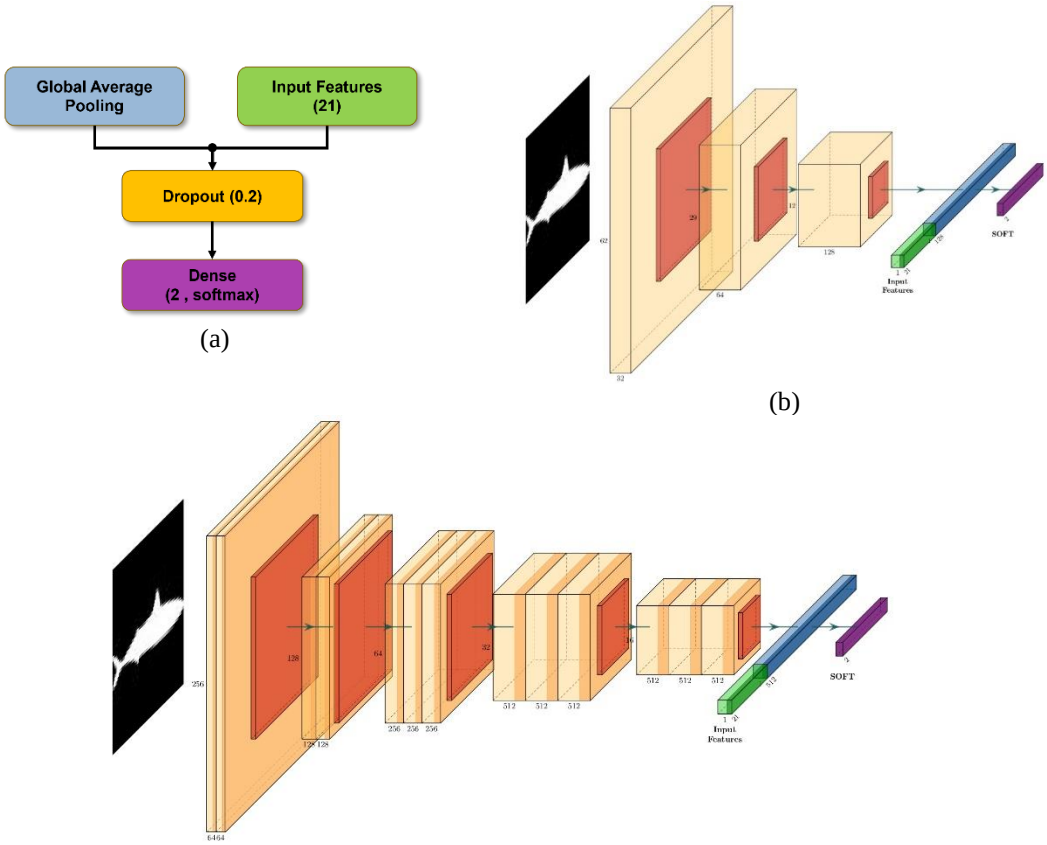


Figura N° 3.24 CNN para la clasificación de tamaño considerando como entrada la silueta de los peces. (a) Esquema 2D; (b) Esquema 3D; no se consideran las capas de dropout. Fuente: [Elaboración propia]

Para la tercera estrategia, dado que es necesario combinar dos tipos de entradas diferentes, fue necesario implementar un generador de datos personalizado para poder facilitar la carga y manipulación de los datos de manera dinámica y en tiempo real durante el entrenamiento del minimodelo. Respecto de las arquitecturas utilizadas, se cuenta con las dos variantes utilizadas para la estrategia anterior con las modificaciones mostradas en los apartados b y c de la Figura N° 3.25, donde se concatena el vector con las características morfológicas (en verde) con el vector de características de la imagen generadas por la CNN respectiva (en azul), según se muestra en el apartado a.



(a)

(b)

(c)

Figura N° 3.25 CNNs para la clasificación de tamaño. a) esquema 2D con la unión de los vectores de características; b) esquema 3D con entradas combinadas utilizando CNN simplificada; c) esquema 3D con entradas combinadas utilizando VGG16. Fuente: [Elaboración propia]

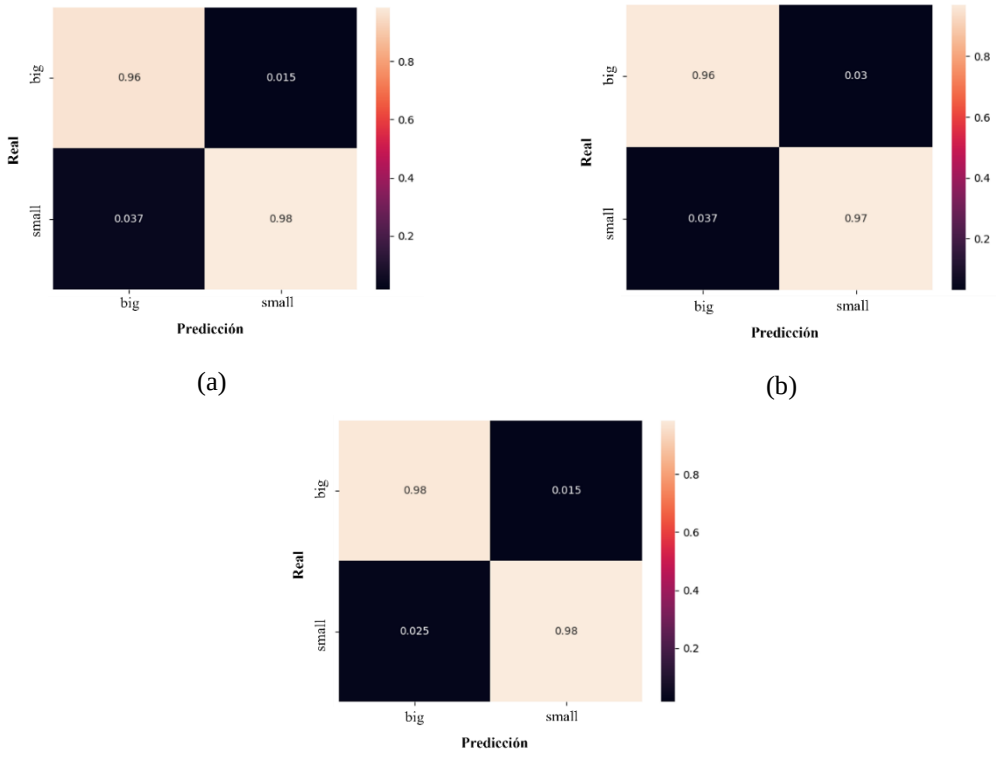
Finalmente, el entrenamiento de los minimodelos se realizó considerando las siguientes configuraciones:

- Distribución 70%-30% para los sets de entrenamiento y validación, respectivamente.
- Imágenes binarias de tamaño 64x64 para las estrategias ii) y iii). Para CNN con VGG16, fue necesario replicar la imagen de la silueta para poder simular una imagen de 3 canales.
- *Data augmentation* en línea relacionado con cambios de brillo, desplazamientos horizontales y verticales y rotaciones; aplicados a cada imagen durante el entrenamiento.
- *Batch* de 32 datos o imágenes por iteración.
- *Early Stopping* ajustado para un total de 150 épocas.
- *Learning rate* constante, ajustado experimentalmente dentro del rango óptimo [3e-3, 5e-3] para las estrategias i y iii y [2e-3, 2e-2] para la estrategia ii.

El resumen de los resultados obtenidos durante la validación de los minimodelos para las tres estrategias planteadas se muestra tanto en las matrices de confusión normalizadas por columnas de la Figura N° 3.26 como en la Tabla N° 3.7. Se puede observar que los mejores resultados son logrados por la clase "*small*", alcanzando valores de la métrica P por sobre 0.96 en todos los casos. Si se considera la métrica MP, el mejor minimodelo de todos es aquel que utiliza las entradas combinadas junto con el modelo VGG16, con un MP de 0.98. Además, las métricas P y R presentan un buen balance, lo cual indica que el modelo clasifica correctamente un gran porcentaje de los datos y no genera demasiados falsos positivos.

Respecto del rendimiento general de los minimodelos, se observa que cada entrada por separado si aporta la información suficiente para generar buenos resultados de clasificación, confirmando que a pesar de que las imágenes de las siluetas no mantienen su relación de tamaño original, si contienen ciertas características útiles para el aprendizaje

del modelo. En caso contrario, esto hubiera afectado tanto a los minimodelos que utilizan las siluetas como las entradas combinadas. En relación con las arquitecturas de CNN utilizadas, contrario a lo esperado, la CNN simplificada logró peores resultados que VGG16 a pesar de que las imágenes son simples, lo cual demuestra que las características ya aprendidas por VGG16 marcaron la diferencia a la hora de adaptar el modelo a la nueva base de datos. Esto puede deberse a que, sin una referencia de tamaño clara, el modelo debe adaptarse a otro tipo de características locales para lograr hacer la clasificación como, por ejemplo, las formas de la cola o aletas que únicamente están presentes en las especies grandes. Sin embargo, de no considerar las máscaras completas con el contorno de los peces, sino que únicamente el contorno formado por los *keypoints* más generales, las formas de las siluetas serían muy similares entre sí, y no hay garantía de que los modelos puedan rendir correctamente en tal situación.



(a)

(b)

(c)

Figura N° 3.26 Mejores matrices de confusión para las tres estrategias del minimodelo de tamaño: a) solo características morfométricas; b) solo imágenes con la silueta de los peces; c) entradas combinadas. Fuente: [Elaboración propia]

*Tabla N° 3.7 Resumen de las métricas de desempeño obtenidas para la validación del minimodelo de tamaño considerando las bases de datos **B03** y **B04**.*

	Minimodelo solo con características morfométricas			
Clase	**precision**	**recall**	**F1**	**MP**
Big	0.96	**0.98**	0.97	0.97
Small	**0.98**	**0.98**	0.98	
	Minimodelo solo con imágenes de siluetas – CNN simplificada			
Clase	**precision**	**recall**	**F1**	**MP**
Big	0.79	**0.98**	0.87	0.89
Small	**0.98**	0.84	0.91	
	Minimodelo solo con imágenes de siluetas – VGG16			
Clase	**precision**	**recall**	**F1**	**MP**
Big	0.96	0.95	0.96	0.97
Small	**0.97**	**0.98**	0.97	
	Minimodelo con entradas combinadas – CNN simplificada			
Clase	**precision**	**recall**	**F1**	**MP**
Big	0.85	**1.00**	0.92	0.93
Small	**1.00**	0.89	0.94	
	Minimodelo con entradas combinadas – VGG16			
Clase	**precision**	**recall**	**F1**	**MP**
Big	**0.98**	**0.98**	0.98	0.98
Small	**0.98**	**0.98**	0.98	

3.4.3 Minimodelo Forma

Este minimodelo tiene por objetivo clasificar a los peces en función de su forma, conectándose únicamente con la rama del árbol de clasificación jerárquica para peces pequeños, tal y como se muestra en la Figura N° 3.27.

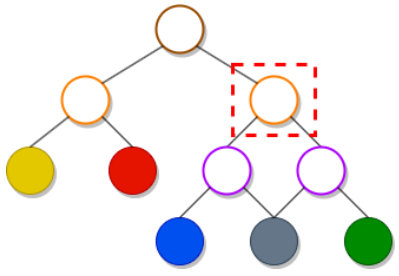


Figura N° 3.27 Árbol de clasificación jerárquica simplificado. El cuadro punteado denota el minimodelo de interés, en este caso el de forma. Fuente: [Elaboración propia]

De manera similar al minimodelo de tamaño, es importante tener en cuenta que todas las imágenes deben preservar su relación de aspecto original para no alterar la forma de los peces y, por ende, el rendimiento de los modelos. No obstante, en contraste con el tamaño, que puede ser menos intuitivo de extraer a partir de objetos en imágenes mostrados a una escala uniforme, la forma representa una característica distintiva fundamental de los objetos, y es la primera intuición que se busca capturar mediante la utilización de modelos basados en CNN.

En el trabajo de Bakerid *et al.* se discute esta última afirmación, puesto que la forma muchas veces es opacada debido a la fuerte influencia de los patrones de texturas o colores presentes en una imagen, siendo fácil alterar la clasificación de un modelo variando estos patrones, aun cuando la forma es visualmente distintiva para el ojo humano [64]. Teniendo en mente este fenómeno, la forma de un objeto puede ser fielmente representada por su silueta al eliminar la textura de la superficie y la información de color, sirviendo como una representación simplificada del contorno de un objeto, y permitiendo a los modelos aprender y extraer características importantes relacionadas con la curvatura de algunas secciones del contorno u otras relaciones entre las diferentes partes de un objeto.

En base a lo anterior, el entrenamiento de los minimodelos de forma se realizó considerando los mismos datos de entrada y las mismas configuraciones propuestas para los 3 experimentos del minimodelo de tamaño, únicamente modificando la etiqueta de clasificación de los peces a "*elongated*" o "*robust*", según corresponda.

El resumen de los resultados obtenidos durante la validación de los minimodelos se muestra tanto en las matrices de confusión normalizadas por columnas de la Figura N° 3.28 como en la Tabla N° 3.8. Se puede observar que la clase "*robust*" alcanza valores de la métrica P de al menos 0.90 en todos los casos, pero la clase "*elongated*" solo lo hace para 4 de los 5 casos, siendo 0.88 su valor más bajo para la estrategia solo con siluetas y el modelo de CNN basado en VGG16. Respecto de la métrica MP, el mejor minimodelo

es el que utiliza solo las distancias morfométricas, con un MP de 0.98, siendo secundado por el minimodelo con entradas combinadas, con un MP de 0.97.

En general, el rendimiento de este minimodelo es más bajo que para el minimodelo de tamaño, ocurriendo un efecto inverso en las métricas al usar la CNN simplificada vs VGG16, donde la primera variante alcanza mejores precisiones en esta oportunidad. Una posible explicación puede ser que las características ya aprendidas por VGG16 son demasiado complejas para centrarse únicamente en el ancho de la silueta, el cual es el elemento visual distintivo entre un pez más alargado, como la anchoveta, y otro más robusto, como la sardina. No así la variante con la CNN simplificada, puesto que esta debe aprender todas las características desde cero y no posee un *bias* inicial, lo cual apoya la teoría de que esta arquitectura si es capaz de extraer características robustas a partir de imágenes simples. Una explicación similar puede darse para la variante del minimodelo que utiliza directamente las distancias morfométricas, donde la relación de ancho viene explícitamente incluida dentro de los datos de entrada, permitiendo al minimodelo ajustarse rápidamente al punto óptimo durante el entrenamiento.

Por otro lado, también se da el caso de imágenes donde las anchovetas no aparecen lo suficientemente alineadas en su eje longitudinal, o donde los peces vienen maltratados y se ven partes de sus entrañas, distorsionando la silueta original del pez y, por ende, aumentando la probabilidad de que el minimodelo se equivoque al momento de asignar la etiqueta de clasificación correspondiente.

En función de las observaciones planteadas, y en vista de que es preferible mantener la simplicidad de los minimodelos y la uniformidad de sus entradas para garantizar buenos resultados a la hora de evaluar el árbol de clasificación jerárquica en su totalidad, tanto para el minimodelo de tamaño como para el minimodelo de forma es preferible optar por la primera estrategia de entrenamiento, descartando la utilización de las siluetas.

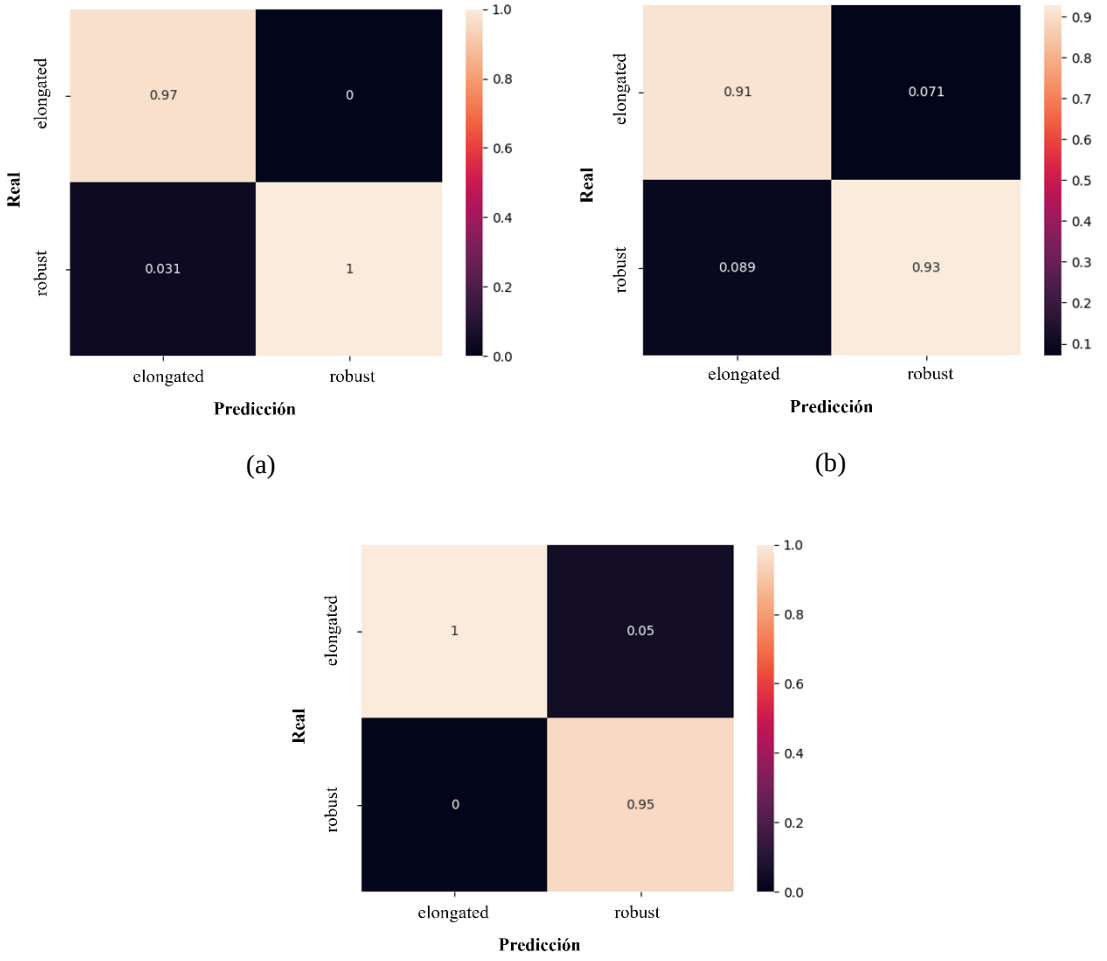


Figura N° 3.28 Mejores matrices de confusión para las tres estrategias del minimodelo de tamaño: a) solo características morfométricas; b) solo imágenes con la silueta de los peces; c) entradas combinadas. Fuente: [Elaboración propia]

*Tabla N° 3.8 Resumen de las métricas de desempeño obtenidas para la validación del minimodelo de tamaño considerando las bases de datos **B03** y **B04**.*

	Minimodelo solo con características morfométricas			
Clase	**precision**	**recall**	**F1**	**MP**
Elongated	0.97	**1.00**	0.98	0.98
Robust	**1.00**	0.99	0.99	
	Minimodelo solo con imágenes de siluetas – CNN simplificada			
Clase	**precision**	**recall**	**F1**	**MP**
Elongated	0.91	0.82	0.86	0.92
Robust	**0.93**	**0.97**	0.95	
	Minimodelo solo con imágenes de siluetas – VGG16			
Clase	**precision**	**recall**	**F1**	**MP**
Elongated	0.88	0.84	0.86	0.91
Robust	**0.93**	**0.95**	0.94	
	Minimodelo con entradas combinadas – CNN simplificada			
Clase	**precision**	**recall**	**F1**	**MP**
Elongated	**1.00**	0.87	0.93	0.97
Robust	0.95	**1.00**	0.97	
	Minimodelo con entradas combinadas – VGG16			
Clase	**precision**	**recall**	**F1**	**MP**
Elongated	**0.92**	0.74	0.82	0.91
Robust	0.90	**0.97**	0.94	

3.4.4 Minimodelo Boca

Este minimodelo tiene por objetivo clasificar a los peces en función de la proyección de su boca, conectándose un nivel más bajo que el nodo para clasificación de forma dentro de la rama del árbol de clasificación jerárquica para peces pequeños, tal y como se muestra en la Figura N° 3.29.

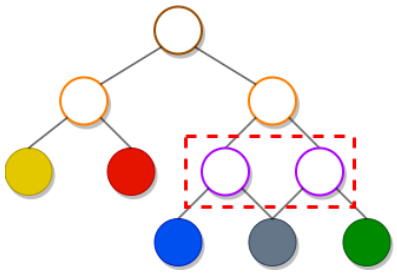


Figura N° 3.29 Árbol de clasificación jerárquica simplificado. El cuadro punteado denota el minimodelo de interés, en este caso el de boca. Fuente: [Elaboración propia]

A diferencia de los minimodelos anteriores, donde la característica clave que se desea clasificar es relativamente fácil de interpretar a partir de una imagen, la proyección de la boca es una característica mucho más sutil que es fácil de omitir ante cambios de brillo o contraste demasiado intensos (como ocurre con muchas imágenes de la base de datos), perdiendo la referencia del punto donde termina la boca, el cual es clave para este minimodelo. De esta forma, es preferible no depender de la imagen, pero sí de las características morfométricas que pueden ser extraídas a partir de los *keypoints*, las cuales se asumen que están correctamente distribuidas a lo largo del contorno del pez.

Considerando que, de todos los *keypoints*, **K_2** se escogió directamente como la proyección del término de la boca de los peces sobre el contorno de su cabeza, se presume que todas las distancias morfométricas relacionadas con este *keypoint* y la cabeza del pez contienen la información suficiente para realizar la clasificación, particularmente la distancia D_1, la cual representa la distancia entre el punto donde comienza la boca y donde esta termina.

Teniendo en mente los detalles ya mencionados, el diseño de este minimodelo se realizó siguiendo dos estrategias:

i. Utilizando solo 6 características morfométricas extraídas de la cabeza de los peces tanto de las bases de datos **B03** y **B04**, donde se consideran las distancias D_1-D_5 y el ángulo ∢A.

ii. Utilizando las 21 características morfométricas extraídas de los peces tanto de las bases de datos **B03** y **B04**.

Para ambas estrategias, se utilizarán las mismas arquitecturas de tipo FCNN utilizadas para los minimodelos de tamaño y forma, únicamente modificando el tamaño de la entrada para la primera estrategia (ver Figura N° 3.30) y las clases de salida del modelo como "*no_protrude*" y "*protude*", según corresponda.

Respecto de los entrenamientos, estos se realizaron con las siguientes configuraciones:

- Distribución 70%-30% para los sets de entrenamiento y validación, respectivamente.
- *Batch* de 32 imágenes por iteración.
- *Early Stopping* ajustado para un total de 150 épocas.
- *Learning rate* constante, ajustado experimentalmente dentro del rango óptimo [1e-3, 2-e2].

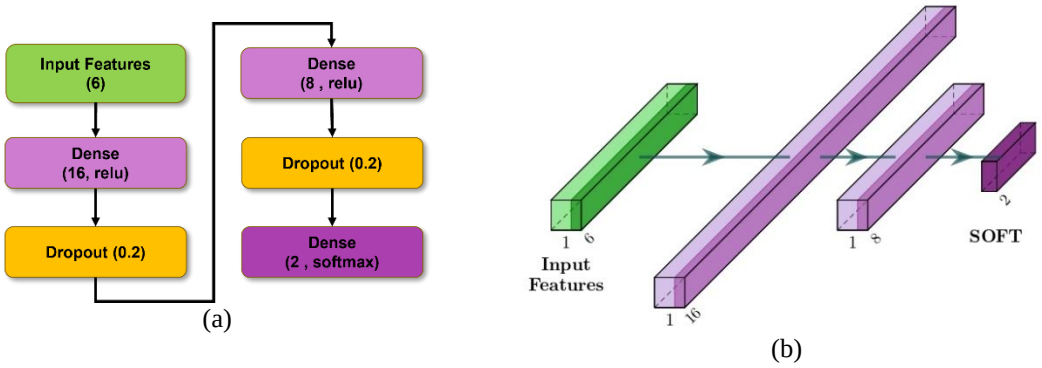
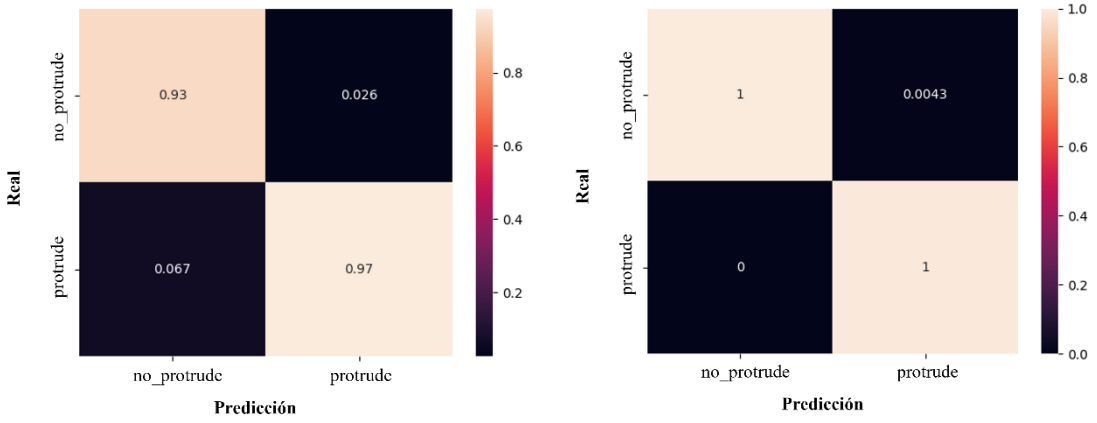


Figura N° 3.30 FCNN para la clasificación de boca considerando como entrada las 6 características morfológicas extraídas de la cabeza de un pez. (a) Esquema 2D; (b) Esquema 3D; no se consideran las capas dropout. Fuente: [Elaboración propia]

El resumen de los resultados obtenidos durante la validación de los minimodelos se muestra tanto en las matrices de confusión normalizadas por columnas de la Figura N° 3.31 como en la Tabla N° 3.9. Se puede observar que ambas estrategias logran métricas con un valor de al menos 0.90 para ambas clases, siendo la clase "*no_protrude*" aquella que presenta los resultados más bajos, con una métrica P de 0.93 y una métrica R de 0.90 en el peor de los casos. Esto puede deberse a que, dentro de la base de datos, la anchoveta es la única especie que pertenece a esta clase, por lo que la base de datos está fuertemente desbalanceada hacia la clase "*protrude*" (un 76% de los datos pertenecen a esta clase). Sin embargo, en vista de los resultados logrados para la segunda estrategia, donde se incluyen todas las distancias morfométricas, este desbalance es compensado por una mejor extracción de características, elevando las métricas hasta lograr un F1 de 0.99. Por otro lado, si bien el mejor minimodelo se obtiene para la segunda estrategia, con un MP de 1.00, la primera estrategia, que utiliza menos datos, solo alcanza un MP ligeramente más bajo de 0.95, demostrando que al menos las distancias medidas de la cabeza de los peces si pueden contener la suficiente información para describir correctamente la proyección de la boca, lo cual podría mejorarse si se incluyeran un mayor número de muestras u otras especies que presenten una boca similar a la anchoveta.



(a) (b)

Figura N° 3.31 Matrices de confusión para las dos estrategias del minimodelo de boca: a) todas las especies de peces; 6 características. c) todas las especies de peces; 21 características. Fuente: [Elaboración propia]

*Tabla N° 3.9 Resumen de las métricas de desempeño obtenidas para la validación del minimodelo de boca considerando las bases de datos **B03** y **B04**.*

	Minimodelo para todas las especies de peces – 6 características			
Clase	**precision**	**recall**	**F1**	**MP**
No Protrude	0.93	0.90	0.92	0.95
Protrude	**0.97**	**0.98**	0.98	
	Minimodelo para todas las especies de peces – 21 características			
Clase	**precision**	**recall**	**F1**	**MP**
No Protrude	1.00	0.98	0.99	1.00
Protrude	**1.00**	**1.00**	1.00	

3.4.5 Minimodelo Manchas

Este minimodelo tiene por objetivo clasificar a los peces en función de los patrones de manchas presentes en su zona dorsal, conectándose únicamente con la rama del árbol de clasificación jerárquica para peces grandes, tal y como se muestra en los diagramas de la Figura N° 3.32, donde se presentan las variantes de dos y tres clases.

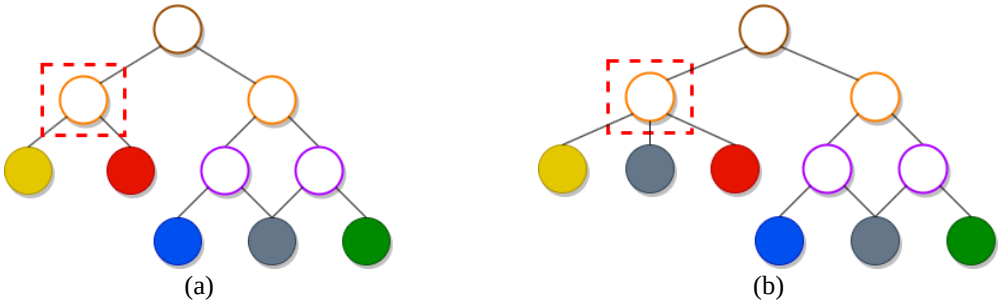


Figura N° 3.32 Árbol de clasificación jerárquica simplificado. El cuadro punteado denota el minimodelo de interés, en este caso el de manchas. Fuente: [Elaboración propia]

Teniendo en mente el proceso de entrenamiento, el diseño de este minimodelo se realizó siguiendo tres estrategias principales:

i. Utilizando una base de datos de peces lo más general posible con diferentes patrones de manchas (**B06**).

ii. Utilizando una base de datos exclusivamente de caballas (**B04**).

iii. Utilizando una combinación entre las bases de datos **B03** y **B04**.

A. Primera Estrategia de Entrenamiento

Para la primera estrategia, la idea es comprobar qué tan factible es enseñarle a un modelo de manera general a distinguir entre peces con y sin manchas utilizando imágenes de la figura completa del pez, considerando para ello imágenes diferentes a las cuatro especies problemáticas. Además, siguiendo la línea de trabajo presentada por Saitoh *et al.*, también se probó con una variante adicional que no utiliza las imágenes directamente, sino que se aplican diferentes filtros de extracción de texturas del estado del arte para construir un vector de características [15]. Particularmente, a lo largo de todas las pruebas realizadas para este minimodelo, se utilizarán los 4 algoritmos que lograron los mejores resultados de clasificación para los autores: GLCM, HOG, DCT y LBP; los cuáles se describen brevemente en el Anexo A.6.3.

Manteniendo el enfoque de entrenamiento utilizado para los minimodelos anteriores, se utilizó una arquitectura pre-entrenada de VGG16 para la variante que utiliza imágenes, únicamente modificando el tamaño de las imágenes RGB a 512x512.

Para la variante que utiliza las características de texturas extraídas a partir de las imágenes, se cuenta con un total de 32.795 datos de entrada que se distribuyen de la siguiente manera:

- GLCM: 14 características.
- HOG: 32.768 características.
- DCT: 4 características.
- LBP: 9 características.

Luego, para el entrenamiento de esta variante, se utilizó una red de tipo FCNN similar a las pruebas anteriores (ver Figura N° 3.33), la cual incorpora una capa de *batch*

normalization para normalizar los datos antes de ingresar a las capas ocultas y capas densas con un mayor número de neuronas en ambos casos (64) dada la alta dimensionalidad de los datos de entrada.

Adicionalmente, se modificaron el rango óptimo del *learning rate* a [3e-4, 3e-3] para la variante con imágenes y a [1e-4, 2e-4] para la variante con texturas, además de las etiquetas de clasificación de los peces a "*no_patches*" o "*patches*", según corresponda.

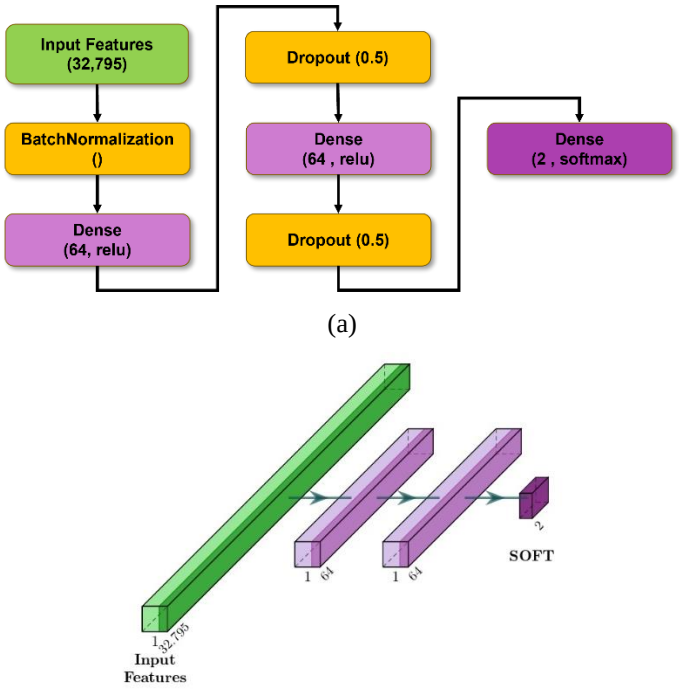


Figura N° 3.33 FCNN para la clasificación de manchas considerando como entrada las características de texturas. (a) Esquema 2D; (b) Esquema 3D; no se consideran las capas de batchNormalization ni dropout. Fuente: [Elaboración propia]

El resumen de los resultados obtenidos durante la validación de la primera estrategia del minimodelo se muestra tanto en la matriz de confusión normalizada por columnas de la Figura N° 3.34 como en las secciones de color ● de la Tabla N° 3.10. Se puede observar que la variante de texturas no rinde tan bien como la variante de imágenes, puesto

que a pesar de que logra una métrica P mayor para la clase "*no_patches*", de 0.96, su métrica R se estanca en solo 0.70, lo cual es un indicador que el minimodelo confunde muchas muestras de peces sin manchas como si las tuvieran. A esto se le suma el hecho que su *learning rate* es particularmente el más pequeño y lento comparado con todo el resto de los minimodelos que se entrenaron a la fecha (siendo potencialmente influenciado por la alta dimensionalidad y complejidad de los datos de texturas), lo cual limita en gran cantidad la capacidad de aprendizaje del modelo, más aún sin utilizar pesos pre-entrenados.

Respecto de la variante con imágenes, si bien se logran métricas sobre 0.9, el modelo no es capaz de generalizar al mismo nivel que el minimodelo *fish/no fish*, lo cual se espera cuando se tienen bases de datos numerosas y balanceadas para todas sus clases. Esto se puede corroborar con la prueba adicional que se muestra en la sección de color , donde el minimodelo fue testeado con imágenes provenientes de las bases de datos **B03** y **B04** con las 4 especies de peces problemáticas (seleccionando únicamente a la caballa dentro de la clase "*patches*"), alcanzando una métrica MP bajo 0.5, demostrando no ser mejor que el lanzamiento de una moneda.

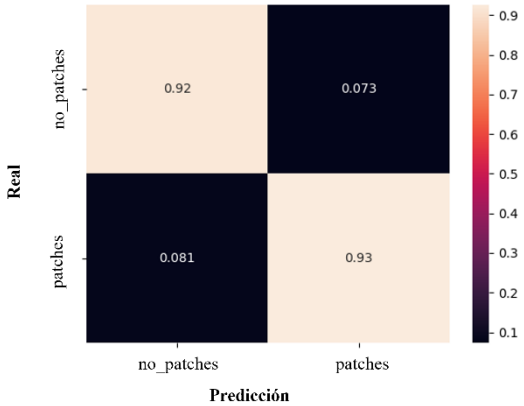


Figura N° 3.34 Mejor matriz de confusión para la primera estrategia del minimodelo de manchas. Fuente: [Elaboración propia]

*Tabla N° 3.10 Resumen de las métricas de desempeño obtenidas para la validación de la primera estrategia del minimodelo de manchas considerando la base de datos **B06**. El ■ indica que se utilizó la imagen completa del pez.*

	Minimodelo para base de datos **B06** – imágenes ■ RGB			
Clase	**precision**	**recall**	**F1**	**MP**
No patches	0.92	0.92	0.92	0.92
Patches	**0.93**	**0.93**	0.93	
	Minimodelo para base de datos **B06** – texturas ■ RGB			
Clase	**precision**	**recall**	**F1**	**MP**
No patches	**0.96**	0.70	0.81	0.87
Patches	0.78	**0.98**	0.87	
	Prueba con base de datos **B03** y **B04** - imágenes ■ RGB			
Clase	**precision**	**recall**	**F1**	**MP**
No patches	**0.80**	0.23	0.36	0.48
Patches	0.15	**0.71**	0.24	

Los malos resultados obtenidos pueden explicarse tomando como base el ejemplo de la base de datos **B06** que se muestra en la Figura N° 3.35, desde donde se pueden apreciar diferentes patrones de manchas que pueden encontrarse en un mismo pez cuando se analiza su figura completa. Esto es sin dudas indeseable, debido a que las imágenes presentan una alta variabilidad y complejidad natural en sus patrones, lo cual puede entrelazarse fácilmente con otros elementos visuales que dificultan la tarea de decidir con exactitud qué constituye una mancha y lo que no (incluso para un humano), más aún cuando dichos patrones pueden variar ampliamente entre individuos de una misma especie.

De esta forma, al igual como se recomienda en la literatura, los patrones de manchas deben ser analizados localmente para garantizar un contexto visual específico donde sea fácil identificarlos [15]. Lo anterior va de la mano con la construcción de la base de datos, donde es preciso definir cuidadosamente lo que se quiere clasificar de lo que no para ser consistente con el aprendizaje del modelo y su capacidad de generalización a *posteriori* para imágenes desconocidas.

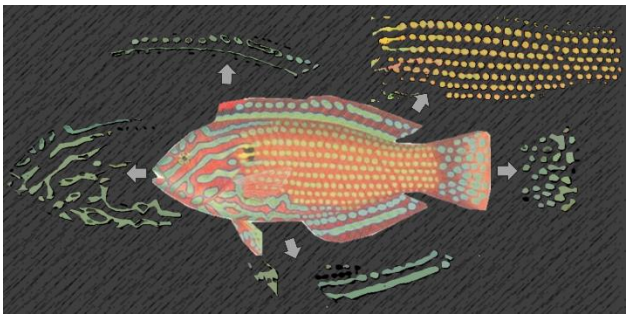


*Figura N° 3.35 Diferentes patrones de manchas segmentados de un pez de la especie Macropharyngodon Meleagris presente en la base de datos **B06**. Fuente: [Elaboración propia]*

B. Segunda Estrategia de Entrenamiento

Esto da pie a la segunda estrategia de entrenamiento, donde se probó la capacidad de los modelos para identificar correctamente los patrones de manchas de interés de las caballas, segmentando dos áreas locales donde se aprecia claramente la diferencia entre una zona con y sin manchas.

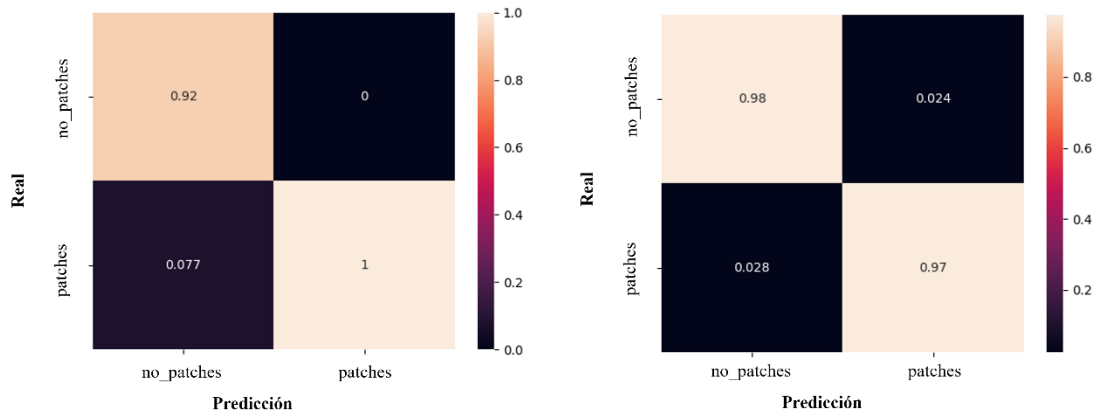
Considerando lo descrito en la Sección 2.5.5B, el entrenamiento de esta segunda estrategia mantiene las variantes con imágenes y texturas de la estrategia anterior junto con las arquitecturas utilizadas para los minimodelos, extendiendo el análisis tanto para los ▲RGB como para los ▲GOS, modificando las siguientes configuraciones:

- Distribución 70%-30% para los sets de entrenamiento y validación, respectivamente.
- Imágenes de tamaño 64x64. Para las imágenes de ▲RGB se utilizó VGG16, mientras que para las imágenes de ▲GOS se utilizó la CNN simplificada.
- *Data augmentation* aplica en línea a las imágenes de ▲GOS para mejorar el rendimiento de los modelos, generando al menos 12 combinaciones distintas por imagen variando los parámetros *kernel* (15 y 31), *theta* (0° y 45°) y *gamma* (0.5, 1 y 1.5) del filtro de Gabor.

- Para los ▲RGB, debido a que únicamente el filtro HOG depende de la resolución de la imagen, el total de características de texturas extraídas se reduce a 539 datos.
- Para los ▲GOS, las características también se ven reducidas, pero las texturas fueron extraídas para cada filtro GOS por separado, siendo concatenadas finalmente en un único vector de características con un total de 1617 datos.
- Rango óptimo del *learning rate* para imágenes de ▲RGB entre [3e-3, 1e-2].
- Rango óptimo del *learning rate* para texturas de ▲RGB entre [1e-3, 2e-3].
- Rango óptimo del *learning rate* para imágenes de ▲GOS entre [2e-4, 3e-3].
- Rango óptimo del *learning rate* para texturas de ▲GOS entre [3e-4, 3e-3].

El resumen de los resultados obtenidos durante la validación de esta estrategia se muestra tanto en las matrices de confusión normalizadas por columnas de la Figura N° 3.30 como en la Tabla N° 3.11. Se puede observar que para todas las variantes se obtienen métricas de al menos 0.92, siendo los ▲GOS para los cuales se logra el mejor caso por una diferencia mínima respecto de los ▲RGB, con una métrica MP de 0.97 para la variante que utiliza imágenes como entrada. Respecto de los minimodelos que utilizan la información de texturas, queda claro que al focalizar la extracción de estas características en una región de interés más acotada se obtienen resultados más coherentes y consistentes, a diferencia de la estrategia anterior, donde fue complejo obtener una iteración del entrenamiento que lograra mejores resultados.

Sin embargo, al igual que para la estrategia anterior, se sigue manteniendo una baja capacidad de generalización para las imágenes de la base de datos **B03**, lo cual se muestra en la sección de color 🟠 de la Tabla N° 3.11. Si bien los resultados son mejores, alcanzando una métrica MP de 0.60, las métricas R y F1 demuestran que el minimodelo todavía confunde muchas de las imágenes sin manchas de la base de datos como si las tuvieran, recordando que solo las caballas presentan el patrón distintivo sobre el cual se desea realizar la clasificación.



(a) (b)

Figura N° 3.36 Mejores matrices de confusión para la segunda estrategia del minimodelo de manchas: a) imágenes de triángulos RGB. b) imágenes de triángulos GOS. Fuente: [Elaboración propia]

*Tabla N° 3.11 Resumen de las métricas de desempeño obtenidas para la validación de la segunda estrategia del minimodelo de manchas considerando las bases de datos **B04**.*

	Minimodelo para base de datos **B04** - imágenes ▲RGB			
Clase	**precision**	**recall**	**F1**	**MP**
No patches	0.92	**1.00**	0.96	0.96
Patches	**1.00**	0.92	0.96	
	Minimodelo para base de datos **B04** – texturas ▲RGB			
Clase	**precision**	**recall**	**F1**	**MP**
No patches	**0.94**	**0.94**	0.94	0.94
Patches	**0.94**	**0.94**	0.94	
	Minimodelo para base de datos **B04** – imágenes ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
No patches	0.97	**0.98**	0.97	0.97
Patches	**0.98**	0.97	0.97	
	Minimodelo para base de datos **B04** – texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
No patches	**0.98**	0.95	0.96	0.96
Patches	0.95	**0.98**	0.96	
	Prueba con base de datos **B03** y **B04** – imágenes ▲GOS - ΔK_0-K_4-K_6			
Clase	**precision**	**recall**	**F1**	**MP**
No patches	**1.00**	0.28	0.44	0.60
Patches	0.21	**1.00**	0.35	

En vista de que las dos estrategias probadas no funcionan adecuadamente para clasificar las manchas a partir de las bases de datos **B03** y **B04** combinadas, es necesario analizar con mayor profundidad la raíz del problema. La Figura N° 3.37 muestra una comparación de los triángulos ΔK_0-**K_4**-**K_6** para un individuo de cada especie problemática, donde se observa claramente que todas las especies presentan un cambio de coloración más oscuro en su zona dorsal. Considerando además que la clase "*no_patches*" fue entrenada hasta el momento solo con imágenes de la zona ventral de las caballas, definida por los triángulos ΔK_0-**K_5**-**K_3** (ver apartado b de la Figura N° 3.38), los cambios de coloración son totalmente desconocidos para los modelos y pueden ser fácilmente interpretados como una mancha en la mayoría de los casos.

(a) (b) (c) (d)

Figura N° 3.37 Comparación de los triángulos ΔK_0-K_4-K_6 de las 4 especies problemáticas. a) anchoveta. b) caballa. c) jurel. d) sardina. Fuente: [Elaboración propia]

(a) (b) (c) (d)

Figura N° 3.38 Comparación de los triángulos ΔK_0-K_5-K_3 de las 4 especies problemáticas. a) anchoveta. b) caballa. c) jurel. d) sardina. Fuente: [Elaboración propia]

C. Tercera Estrategia de Entrenamiento

La tercera estrategia de entrenamiento nace en función de esta problemática. En primer lugar, el entrenamiento del minimodelo fue realizado directamente con las bases de datos **B03** y **B04** para lograr la mejor capacidad de generalización posible. En segundo lugar, se modificaron las clases de salida del minimodelo para abarcar la tercera variante con los cambios de coloración. Dentro de este nuevo orden, la clase "*patches*" fue dividida

entre la clase "*painted*", la cual refleja el patrón zigzagueante característico de las caballas, y la clase "*full_color*", la cual engloba los cambios de coloración presentes en las anchovetas, jureles y sardinas.

En base a lo anterior, el entrenamiento de los minimodelos se realizó considerando las mismas 4 variantes que para la estrategia anterior, modificando las siguientes configuraciones:

- Rango óptimo del *learning rate* para imágenes de ▲RGB entre [1e-3, 6e-3].
- Rango óptimo del *learning rate* para texturas de ▲RGB entre [3e-3, 2e-2].
- Rango óptimo del *learning rate* para imágenes de ▲GOS entre [3e-4, 3e-3].
- Rango óptimo del *learning rate* para texturas de ▲GOS entre [4e-4, 3e-3].

El resumen de los resultados obtenidos durante la validación de esta estrategia se muestra tanto en las matrices de confusión normalizadas por columnas de la Figura N° 3.40 como en la Tabla N° 3.12. Por un lado, se puede observar que el rendimiento general de los minimodelos disminuyó en casi todos los casos, siendo la clase "*painted*" la que logra los peores resultados, con una métrica R más baja de 0.81 para la variante con texturas a partir de ▲RGB, y una métrica P más baja de 0.72 para la variante con imágenes de ▲RGB. Por otro lado, para esta estrategia si marcó más la diferencia el utilizar imágenes GOS. Dicha variante logró los mejores resultados, alcanzando una métrica MP de 0.95, y presentando además un buen balance para la predicción de todas las clases que le permitió lograr métricas F1 de al menos 0.93.

Tomando como base la matriz de confusión del minimodelo de texturas extraídas de ▲RGB (apartado a), es posible darse cuenta de que la clase "*painted*" se tiende a confundir en mayor proporción con la clase "*full_color*" que con la clase "*no_patches*", ocurriendo un efecto parecido en la matriz del apartado b. Esto puede ser un indicador de la existencia de imágenes donde el patrón de manchas de las caballas no predomina por sobre el cambio de coloración de su piel, lo cual puede ser un problema a futuro debido a que

de momento es su única característica distintiva respecto de un jurel dentro del árbol de clasificación. La Figura N° 3.39 corrobora esta afirmación, donde se muestran dos ejemplos de imágenes dentro de la clase "*painted*" que no poseen un patrón de manchas distintivo a simple vista, lo cual puede ocurrir cuando las caballas se encuentran ladeadas en las imágenes, asemejándose mucho a un jurel.

(a) (b)

Figura N° 3.39 Ejemplos de triángulos ΔK_0-K_4-K_6 de dos caballas donde no se aprecia en gran medida su patrón de manchas característico. Fuente: [Elaboración propia]

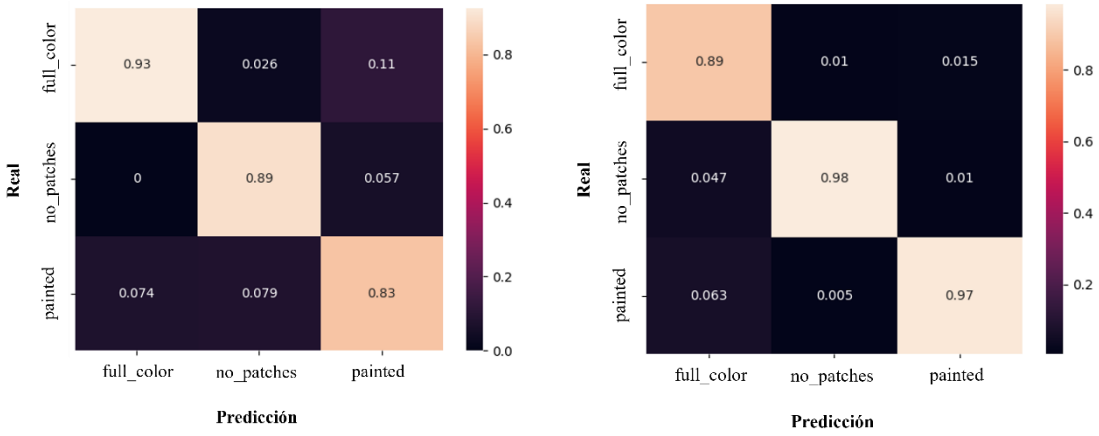


Figura N° 3.40 Mejores matrices de confusión para la tercera estrategia del minimodelo de manchas: a) texturas de ▲RGB. b) imágenes de ▲GOS. Fuente: [Elaboración propia]

En función de los resultados finales obtenidos, y considerando que al menos dos tipos de clasificación jerárquica requieren evaluar todas las entradas del modelo para cada minimodelo en la jerarquía, no limitándose solo a jurel y caballa para el minimodelo de manchas, se prefirió optar por los minimodelos de 3 clases que dependen tanto de las imágenes GOS como de sus texturas para la etapa final, teniendo en mente como objetivo

el obtener los mejores resultados posibles, y al mismo tiempo reducir la cantidad de variantes que deberán evaluarse.

*Tabla N° 3.12 Resumen de las métricas de desempeño obtenidas para la validación de la tercera estrategia del minimodelo de manchas considerando las bases de datos **B03** y **B04**. El ▲ indica que se utilizaron los triángulos segmentados de los peces.*

	Minimodelo para base de datos **B03** y **B04** - imágenes ▲ RGB			
Clase	**precision**	**recall**	**F1**	**MP**
Full color	0.90	**0.96**	0.93	
No Patches	**1.00**	0.69	0.82	0.87
Painted	0.72	0.86	0.78	
	Minimodelo para base de datos **B03** y **B04** – texturas ▲ RGB			
Clase	**precision**	**recall**	**F1**	**MP**
Full color	**0.93**	0.91	0.92	
No Patches	0.89	**0.94**	0.92	0.88
Painted	0.83	0.81	0.82	
	Minimodelo para base de datos **B03** y **B04** – imágenes ▲ GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Full color	0.89	**0.98**	0.94	
No Patches	**0.98**	0.91	0.95	0.95
Painted	0.97	0.89	0.93	
	Minimodelo para base de datos **B03** y **B04** – texturas ▲ GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Full color	0.89	0.84	0.87	
No Patches	**0.94**	**0.95**	0.94	0.87
Painted	0.78	0.82	0.80	

3.4.6 Modelo Jerárquico Final

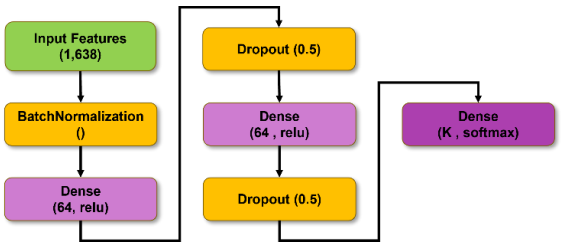
Este modelo tiene por objetivo combinar algunos, o bien, todos los minimodelos para generar una nueva etiqueta de clasificación en base a la jerarquía o taxonomía presentada en la Sección 2.5.3, la cual es luego comparada con la clasificación realizada por el modelo de detección (YOLOv7) y, con ello, se toma la decisión final en base al mejor umbral de confianza u otra medida de comparación que se estime conveniente.

Tomando como base las distintas configuraciones del clasificador jerárquico planteadas en la Sección 2.3.5, el diseño y testeo de esta etapa se realizó siguiendo cuatro estrategias principales:

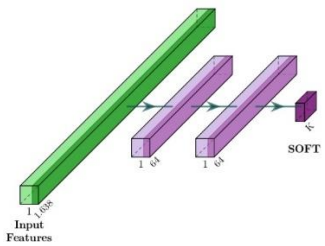
i. Utilizando un clasificador multiclase, sin considerar una jerarquía de los datos ni los minimodelos.
ii. Utilizando un clasificador jerárquico plano, sin considerar una jerarquía de los datos, pero si los minimodelos.
iii. Utilizando un clasificador jerárquico local por nodo padre (en adelante, *Top*/*Down*), considerando un único camino posible a través de los minimodelos.
iv. Utilizando un clasificador jerárquico multidimensional, considerando todos los caminos posibles a través de los minimodelos.

A. Primera Estrategia de Entrenamiento

La primera estrategia tiene como finalidad el mostrar qué tan capaz es el modelo para clasificar correctamente a las especies problemáticas sin depender de ninguna jerarquía, utilizando para ello una arquitectura pre-entrenada de VGG16 (al igual que para el entrenamiento de algunos minimodelos), junto con las imágenes originales de las bases de datos **B03** y **B04**. Adicionalmente, también se probó una variante que solo utiliza las mediciones morfométricas (21) y las texturas de los ▲GOS (1617) como un único vector unidimensional para entrenar la arquitectura de tipo FCNN mostrada en la Figura N° 3.41, inspirada originalmente en la arquitectura utilizada en el minimodelo de manchas.



(a)



(b)

Figura N° 3.41 FCNN para la clasificación multiclase utilizando un único vector de características sin depender de una jerarquía o de los minimodelos. (a) Esquema 2D. (b) Esquema 3D; no se consideran las capas de batch normalization ni dropout. Fuente: [Elaboración propia]

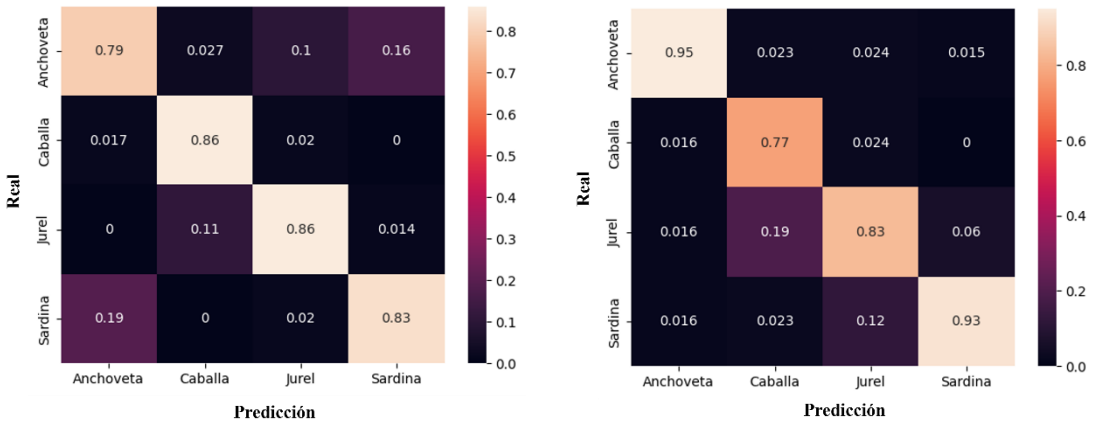
De esta forma, los entrenamientos se realizaron considerando las siguientes configuraciones:

- Distribución 70%-30% para los sets de entrenamiento y validación, respectivamente.
- Imágenes RGB de tamaño 256x256 utilizando fondos aleatorios y 1638 datos entre morfometría y texturas de ▲GOS. Las texturas fueron extraídas a partir del triángulo de la zona dorsal de los peces, definido por ΔK_0-K_4-K_6.
- *Data augmentation* en línea relacionado con cambios de brillo, desplazamientos horizontales y verticales y rotaciones; aplicados a cada imagen durante el entrenamiento.
- *Batch* de 32 imágenes o datos por iteración.
- *Early Stopping* ajustado para un total de 150 épocas.
- Rango óptimo del *learning rate* para imágenes RGB entre [1e-3, 1e-2].
- Rango óptimo del *learning rate* para morfometría + texturas entre [6e-4, 7e-3].

El resumen de los resultados obtenidos durante la validación de los modelos se muestra tanto en las matrices de confusión normalizada por columnas de la Figura N° 3.42 como en la Tabla N° 3.13. Se puede observar que ambos modelos no logran superar un valor de 0.9 para la métrica MP, notándose una cierta tendencia del modelo en base a imágenes a clasificar mejor a las especies grandes, destacándose la métrica R de la caballa, con un 0.94, mientras que el modelo en base a morfometría + texturas clasifica mejor a las especies pequeñas, también siendo la métrica P la cual alcanza el valor más alto, con un 0.95 para anchoveta. Además, si se observa con mayor detalle la matriz de confusión del modelo en base a imágenes (apartado a), este tiende a confundir en mayor proporción anchoveta/sardina y caballa/jurel por separado, lo cual es esperable debido a que son los

pares de especies que más se parecen entre sí. Sin embargo, también se da un porcentaje no menor del 10% de anchovetas que se confunden con jureles, lo cual puede ser un efecto negativo proveniente del reescalamiento de las imágenes, considerando que estas no conservan su relación de tamaño original. Respecto de la matriz de confusión del modelo en base a texturas (apartado b), si bien se mantiene la tendencia a confundir jurel con caballa en un 19%, también existe una confusión no menor entre jurel y sardina, errando un 12% de la clasificación de sardinas como si estas fueran jureles, pero no así entre sardina/anchoveta, donde la proporción de errores se mantiene bajo el 1.6%.

Adicionalmente, en la Tabla Anexo B.1 se muestran los resultados obtenidos con tres clasificadores del estado del arte (*decision tree*, SVM y KNN) para las variantes de clasificación entre 4 especies utilizando únicamente morfometría + texturas (debido a la restricción inherente de los modelos para trabajar con imágenes). Si bien para los clasificadores SVM y KNN no se lograron resultados con una métrica MP superior a 0.72, el modelo de *decision tree* logró mejores resultados que la variante de VGG16 con imágenes RGB, alcanzando una métrica MP de 0.87, donde nuevamente se repite la tendencia de clasificar mejor a las especies de peces pequeñas, con una métrica F1 de 0.94.



(a) (b)

Figura N° 3.42 Matrices de confusión obtenidas para el modelo de clasificación multiclase. a) imágenes RGB. b) morfometría + texturas ▲GOS. Fuente: [Elaboración propia]

*Tabla N° 3.13 Resumen de las métricas de desempeño obtenidas para la validación del modelo de clasificación multiclase para 4 especies considerando las bases de datos **B03** y **B04**.*

	Modelo clasificación multiclase VGG16 – imágenes ■ RGB			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	0.79	0.73	0.76	0.84
Caballa	**0.86**	**0.94**	0.90	
Jurel	**0.86**	0.89	0.88	
Sardina	0.83	0.83	0.83	
	Modelo clasificación multiclase FCNN – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**0.95**	**0.95**	0.95	0.87
Caballa	0.77	0.94	0.85	
Jurel	0.83	0.72	0.77	
Sardina	0.93	0.90	0.91	

De manera complementaria a los resultados anteriores, también se entrenaron modelos a partir de algunas de las combinaciones posibles entre solo dos especies de peces; particularmente, especies grandes, especies pequeñas y especies combinadas (anchoveta y jurel).

La Tabla N° 3.14 muestra un resumen con los mejores resultados obtenidos entre las variantes con imágenes o con morfometría + texturas, mientras que la totalidad de las pruebas se muestran en las tablas de color ○ presentadas en el Anexo B.2 y Anexo B.3, respectivamente. Desde aquí es posible observar que, al separar manualmente las bases de datos, los resultados mejoran considerablemente respecto de las variantes de 4 especies, particularmente para el modelo que utiliza morfometría + texturas para clasificar anchoveta/sardina, logrando un MP casi ideal de 0.99, mientras que la clasificación de especies grandes logra un MP máximo de 0.93 para el modelo que utiliza imágenes RGB. Si se comparan ambos modelos respecto de sus variantes con el segundo tipo de entrada, ninguna de las dos variantes logra elevar su métrica MP por sobre 0.86, enfatizando el hecho que las especies pequeñas se clasifican mejor utilizando morfometría y texturas, mientras que las especies grandes se clasifican mejor utilizando imágenes. Respecto del modelo

con clases combinadas, se logran resultados muy similares utilizando ambos tipos de entradas, lo cual muestra que los modelos sí son capaces de diferenciar correctamente entre especies que no son tan similares entre sí con relativa facilidad.

*Tabla N° 3.14 Resumen de las mejores métricas de desempeño obtenidas para la validación del modelo de clasificación multiclase para 2 especies considerando las bases de datos **B03** y **B04**.*

	Modelo clasificación multiclase FCNN – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	0.98	**1.00**	0.99	0.99
Sardina	**1.00**	0.99	0.99	
	Modelo clasificación multiclase VGG16 – imágenes ■ RGB			
Clase	**precision**	**recall**	**F1**	**MP**
Caballa	**0.94**	0.88	0.91	0.93
Jurel	0.92	**0.96**	0.94	
	Modelo clasificación multiclase VGG16 – imágenes ■ RGB			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	0.95	**1.00**	0.98	0.98
Jurel	**1.00**	0.94	0.97	

Los resultados de la primera estrategia demuestran dos cosas:

- El sistema sí es capaz de superar los resultados de la prueba de validación realizada por Sernapesca utilizando enfoques diferentes para la clasificación entre especies grandes y especies pequeñas. Esto es un buen indicador para el análisis posterior, donde se decidió que el minimodelo de tamaño tiene que ser el nodo pivote o raíz desde el cual se divide la clasificación jerárquica de las especies problemáticas.
- El sistema todavía presenta espacio de mejora para clasificar a las 4 especies problemáticas o solo a las especies grandes dentro de un único modelo.

B. Segunda Estrategia de Entrenamiento

Es entonces como el análisis mostrado da pie a la segunda estrategia de diseño, utilizando para ello un modelo jerárquico plano que incorpora las mejores variantes seleccionadas de los minimodelos de tamaño, forma, manchas y boca diseñados en las secciones anteriores. La idea detrás de esta estrategia es utilizar a los minimodelos como un *backbone* más adaptable a los requerimientos específicos de cada clase del modelo, siendo más flexible que un *backbone* de un modelo de CNN, el cual suele ser una arquitectura fija que se entrena para un conjunto de datos mucho más grande y diverso.

En el apartado a de la Figura N° 3.43 se muestra un diagrama con el funcionamiento del modelo jerárquico plano para la clasificación de las especies problemáticas. Primero, se toma como entrada a los datos morfométricos y las imágenes o las texturas de ▲GOS, según corresponda a cada minimodelo. Luego, debido al aplanamiento de la jerarquía, el modelo transforma todas las características de entrada en un único vector de características con las 9 posibles salidas (clases) de los 4 minimodelos combinados (2 salidas para los minimodelos de tamaño, forma y boca, y 3 salidas para el minimodelo de manchas). Finalmente, el modelo realiza la clasificación final en base al número de especies deseado y al tipo de clasificador que se desee utilizar. Particularmente para la imagen de ejemplo, la clasificación final es realizada utilizando la arquitectura de tipo FCNN que se muestra en los apartados b y c.

Respecto del entrenamiento del modelo, es importante destacar que este puede realizarse permitiendo o no el reentrenamiento de alguno o de todos los minimodelos. La elección de este escenario depende de varios factores, como: la naturaleza de los datos, por ejemplo, si estos cambian significativamente respecto del entrenamiento original de un minimodelo; los recursos disponibles, considerando que el modelo jerárquico final es computacionalmente más costoso que cada minimodelo por separado; y del rendimiento deseado para el modelo final, el cual puede estar sujeto a cambios tanto respecto de su jerarquía como del número de clases de cada nodo.

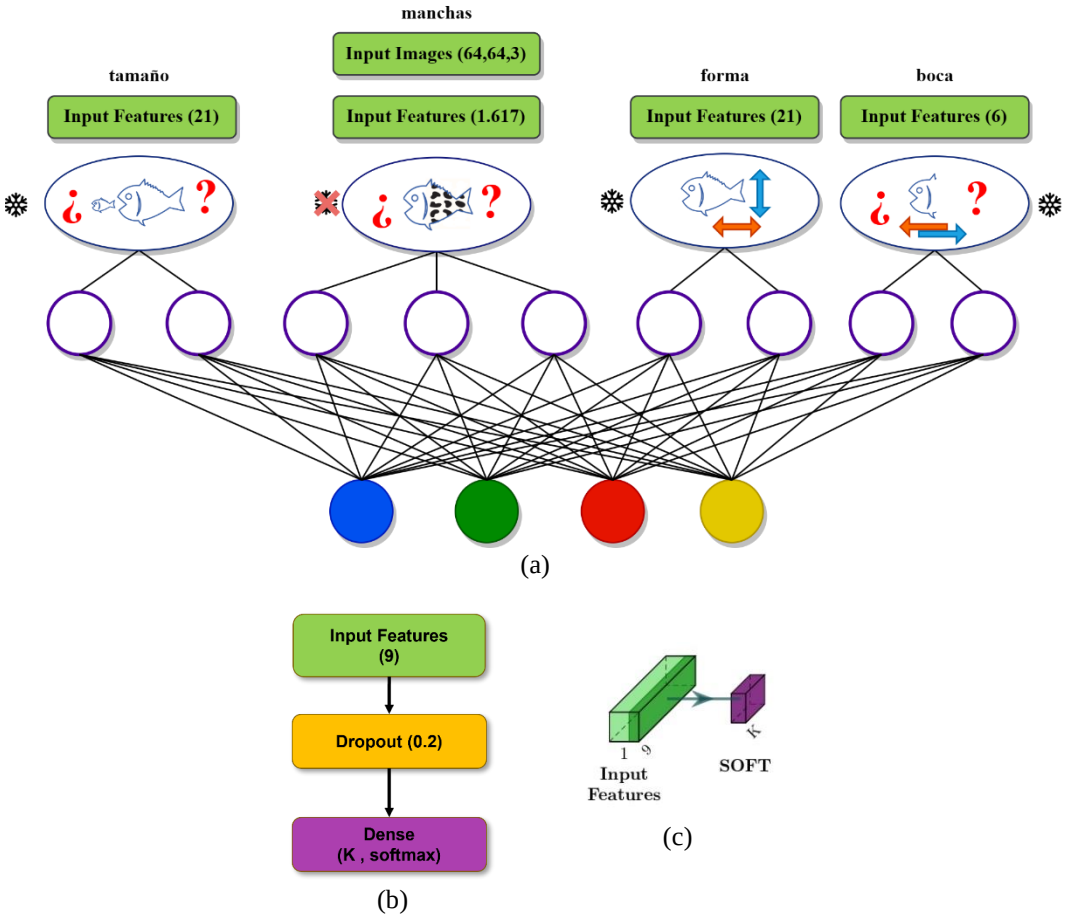


Figura N° 3.43 Ejemplo del funcionamiento del modelo jerárquico plano con o sin reentrenamiento, utilizando una arquitectura de tipo FCNN. a) diagrama del modelo considerando todas las características de entrada; b) esquema 2D de las capas de clasificación del modelo; c) esquema 3D de las capas de clasificación del modelo. Fuente: [Elaboración propia]

Por un lado, si se permite el reentrenamiento de los minimodelos, este puede realizarse "en línea" o de forma incremental, iterando sobre cada *batch* de datos a través de los minimodelos durante cada época del entrenamiento del modelo final (solo disponible para redes neuronales). También, este puede realizarse por separado para cada minimodelo que se desee reajustar, y luego únicamente actualizar sus pesos durante el entrenamiento del modelo final (disponible para todos los tipos de clasificadores), siendo un enfoque intermedio entre las modalidades con y sin reentrenamiento.

Por otro lado, si no se permite el reentrenamiento de los minimodelos, la estructura del modelo es similar al caso anterior si se realiza un entrenamiento incremental, congelando los pesos de los minimodelos para evitar su actualización durante el entrenamiento. Esto también se puede apreciar en el diagrama del apartado a de la Figura N° 3.43, donde el símbolo ❄ indica qué minimodelos se mantuvieron congelados durante el entrenamiento del clasificador final. Sin embargo, también es posible simplificar la estructura del modelo para reducir los tiempos de entrenamiento y facilitar su integración con otro tipo de clasificadores, distintos a las redes neuronales, si se realiza un entrenamiento compacto. En este enfoque de entrenamiento, en lugar de utilizar las características morfológicas, se puede usar directamente el vector de características compactado con las 9 posibles salidas de los minimodelos, y con ello generar un nuevo conjunto de entrenamiento que solo necesite pasar por el clasificador final.

De esta forma, los entrenamientos con FCNN se realizaron considerando las siguientes configuraciones:

- Imágenes de ▲GOS de tamaño 64x64 y 1638 datos entre morfometría y texturas de ▲GOS. Las texturas fueron extraídas a partir del triángulo de la zona dorsal de los peces, definido por ΔK_0-**K_4**-**K_6**.
- Reentrenamiento activado únicamente para el minimodelo de manchas en sus dos variantes, utilizando el enfoque incremental[23].
- *Data augmentation* en línea relacionado con cambios de brillo, desplazamientos horizontales y verticales y rotaciones; aplicados a cada imagen GOS durante el entrenamiento.
- *Batch* de 32 imágenes o datos por iteración.
- *Early Stopping* ajustado para un total de 150 épocas.

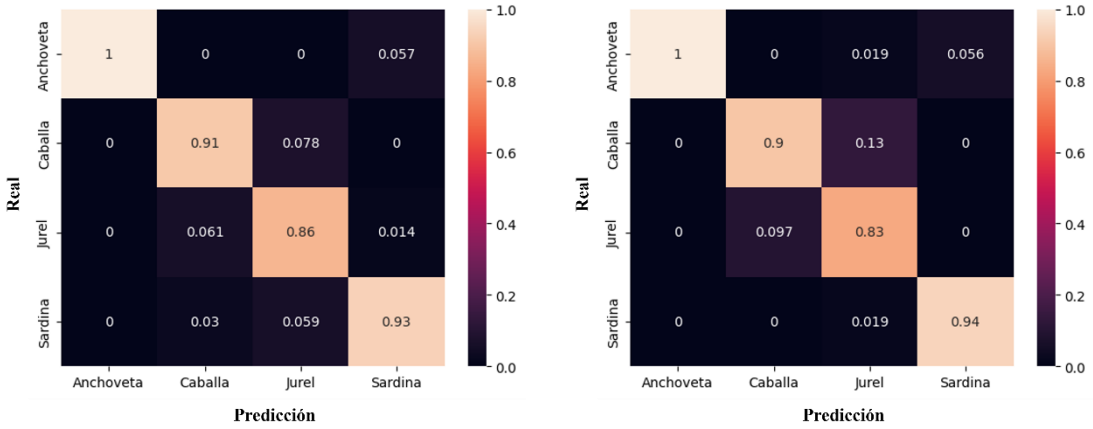
[23] Esto se debe a que no se considera el *data augmentation* aplicado originalmente a la base de datos durante el entrenamiento del minimodelo de manchas. Además, le permite al minimodelo ajustarse mejor ante ciertas imágenes donde los patrones de manchas o cambios de coloración no son tan notorios.

- Rango óptimo del *learning rate* considerando morfometría + imágenes ▲GOS entre [1e-3, 3e-3].
- Rango óptimo del *learning rate* considerando morfometría + texturas ▲GOS entre [1e-3, 1e-2].

El resumen de los resultados obtenidos durante la validación de los modelos para las 4 especies problemáticas se muestra tanto en las matrices de confusión normalizada por columnas de la Figura N° 3.44 como en la Tabla N° 3.15. Se puede observar que ambos modelos mejoran en términos de rendimiento respecto de su análogo correspondiente de la primera estrategia, alcanzando valores para la métrica MP de al menos 0.92, siendo superior el modelo que utiliza las imágenes de ▲GOS por un leve margen de 0.01 puntos. Individualmente hablando, la especie que mejor es clasificada corresponde a anchoveta, con una métrica P de 1.00 y una métrica F1 de 0.97; las más altas en comparación con el resto de las especies. Por el contrario, la especie con los peores resultados de clasificación es la caballa, con una métrica R de 0.88 y una métrica F1 de 0.90 en el mejor de los casos (el modelo a partir de morfometría + imágenes ▲GOS). Respecto de la tendencia de los modelos a separar la clasificación de las especies grandes respecto de las especies pequeñas, se puede apreciar fácilmente por la distribución de colores de ambas matrices que los errores más altos se logran solo entre anchoveta/sardina y jurel/caballa, siendo la clasificación de jurel/caballa la que logra los peores resultados, con un 13% de caballas que fueron confundidas con jureles y un 9.7% en el caso opuesto. Particularmente, el modelo a partir de morfometría + texturas ▲GOS es el que alcanza los errores más bajos para especies de distinto tamaño, con solo un 1.9% de confusión entre anchoveta/jurel y sardina/jurel.

Adicionalmente, en la Tabla Anexo B.2 se muestran los resultados obtenidos con los clasificadores del estado del arte para las variantes de clasificación entre 4 especies utilizando únicamente morfometría + texturas con un enfoque de entrenamiento compactado sin reentrenamiento. En esta oportunidad, los tres clasificadores lograron resultados

muy similares, alcanzando métricas MP de al menos 0.9. También se mantiene la tendencia de clasificar mejor a las anchovetas por sobre el resto de las especies, con una métrica F1 más alta de 0.98 para el clasificador SVM. Respecto de la caballa, esta logra métricas P más bajas respecto de los clasificadores FCNN, con un máximo solo de 0.79 para los clasificadores SVM y KNN. Esto demuestra que la incorporación de los minimodelos logra una mejora significativa en el rendimiento de los clasificadores finales, principalmente debido a que las características originales se simplifican a tal grado que el modelo solo debe ajustarse a las relaciones que se forman entre las distintas clases de los minimodelos, las cuáles pueden establecerse de manera más flexible en un modelo jerárquico plano al no forzar una jerarquía determinada.



(a) (b)

Figura N° 3.44 Matrices de confusión obtenidas para el modelo de clasificación jerárquico plano. a) morfometría + imágenes ▲GOS; b) morfometría + texturas ▲GOS. Fuente: [Elaboración propia]

De manera similar a la estrategia anterior, también se entrenaron modelos a partir de algunas de las combinaciones posibles entre solo dos especies de peces. La Tabla N° 3.16 muestra un resumen con los mejores resultados obtenidos entre las variantes con morfometría + imágenes de ▲GOS o morfometría + texturas de ▲GOS, mientras que la totalidad de las pruebas se muestran en las tablas de color ● presentadas en el Anexo B.2 y en el Anexo B.3, respectivamente para cada variante. Desde aquí es posible observar nuevamente que, al separar manualmente las bases de datos, las métricas mejoran para la

totalidad de las especies, logrando una clasificación ideal con una métrica MP de 1.00 para las especies pequeñas en ambas variantes del modelo, y para las especies combinadas solo en la variante con morfometría + texturas de ▲GOS. Respecto de las especies grandes, en esta oportunidad es el modelo que depende de las texturas de ▲GOS el que logra los mejores resultados, con una métrica MP de 0.98 (la más alta alcanzada durante todas las pruebas realizadas a lo largo de esta tesis), mientras que para su variante con imágenes de ▲GOS, el modelo también logra buenos resultados, con una métrica MP de 0.95, siendo la caballa la cual alcanza métricas F1 un poco más bajas que las del jurel, con un 0.94 en el peor caso.

*Tabla N° 3.15 Resumen de las métricas de desempeño obtenidas para la validación del modelo de clasificación jerárquico plano considerando las bases de datos **B03** y **B04**.*

	Modelo jerárquico plano – morfometría + imágenes ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	**0.94**	0.97	0.93
Caballa	0.91	0.88	0.90	
Jurel	0.86	**0.94**	0.90	
Sardina	0.93	**0.94**	0.94	
	Modelo jerárquico plano – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.92	0.96	0.92
Caballa	0.90	0.80	0.85	
Jurel	0.83	0.94	0.88	
Sardina	0.94	**0.99**	0.96	

Considerando a los clasificadores del estado del arte para las variantes de clasificación entre 2 especies, donde se utilizó únicamente morfometría + texturas con un enfoque de entrenamiento compactado sin reentrenamiento (ver Anexo B.1.2), los resultados presentados se muestran muy similares a los obtenidos utilizando FCNN, donde los modelos para especies pequeñas y especies combinadas logran métricas MP de 0.99 con al menos uno de los clasificadores utilizados. No obstante, la clasificación de especies gran-

des sigue logrando métricas más bajas, con un peor caso de 0.85 para la métrica MP utilizando *decision tree*, y un mejor caso de 0.9 para la métrica MP utilizando SVM y KNN. En el caso del jurel, se observa que es la especie con la tasa más alta de clasificación errónea, con métricas R que no superan el valor de 0.87.

*Tabla N° 3.16 Resumen de las mejores métricas de desempeño obtenidas para la validación del modelo de clasificación jerárquico plano para 2 especies considerando las bases de datos **B03** y **B04**.*

	Modelo jerárquico plano – morfometría + imágenes/texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	**1.00**	1.00	1.00
Sardina	**1.00**	**1.00**	1.00	
	Modelo jerárquico plano – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Caballa	**1.00**	0.94	0.97	0.98
Jurel	0.96	**1.00**	0.98	
	Modelo jerárquico plano – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	**1.00**	1.00	1.00
Jurel	**1.00**	**1.00**	1.00	

Los resultados de la segunda estrategia demuestran que la clasificación de las especies problemáticas mejora considerablemente al integrar la clasificación intermedia realizada de forma mucho más específica por los minimodelos, pero los modelos de 4 especies aun no logran alcanzar los resultados logrados por los modelos de 2 especies. Particularmente, para las especies grandes, el minimodelo de manchas si aporta la información suficiente para diferenciar una imagen entre jurel y caballa, mostrando los mejores resultados para las variantes del modelo jerárquico plano con reentrenamiento.

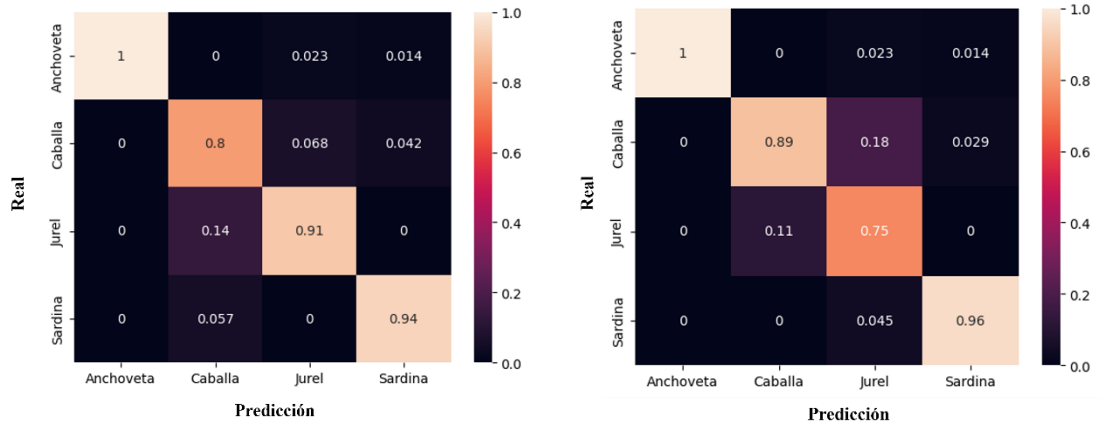
C. Tercera y Cuarta Estrategias de Entrenamiento

Para continuar con el análisis del clasificador final, la tercera y cuarta estrategias buscan comparar qué tan efectiva es la inclusión de una jerarquía más estructurada para la evaluación de los minimodelos respecto de las estrategias anteriores, construyendo un

conjunto de reglas que sigan la estructura del árbol de clasificación presentada en la Sección 2.5.3. Cabe notar además que, si bien los modelos jerárquicos *Top/Down* y multidimensional no requieren de un entrenamiento como tal para generar la clasificación final, estos si pueden adoptar las modalidades con y sin reentrenamiento para los minimodelos utilizados a lo largo de la jerarquía.

El diseño final de la arquitectura del modelo jerárquico se presenta en el ANEXO C. Particularmente, la Figura Anexo C.1 muestra el esquema 3D de la arquitectura con la variante del minimodelo de manchas que utiliza las imágenes de ▲GOS, mientras que la Figura Anexo C.2 muestra la variante que utiliza las texturas de ▲GOS.

Por un lado, el resumen de los resultados obtenidos para el modelo jerárquico Top/Down, evaluado en el conjunto de validación de las bases de datos B03 y B04 para las 4 especies problemáticas, se muestra tanto en las matrices de confusión normalizadas por columnas de la Figura N° 3.45 como en la Tabla N° 3.17. Se puede observar que los resultados logran métricas a un nivel intermedio entre las obtenidas para la primera y la segunda estrategia, siendo mejores que la primera, pero peores que la segunda por un leve margen. Respecto de las métricas obtenidas para cada especie, la anchoveta sigue logrando los mejores resultados con un buen balance entre las métricas P y R, logrando una métrica F1 de 0.98 para las dos variantes de las entradas de los modelos, mientras que la caballa se sigue manteniendo como la especie que logra los peores resultados, logrando métricas F1 que no superan el valor de 0.81. En términos de los errores de clasificación, el modelo a partir de morfometría + texturas de ▲GOS logra el mayor error de clasificación entre jurel y caballa, con un 18% de caballas mal clasificadas, pero en esta oportunidad, para el modelo a partir de morfometría + imágenes de ▲GOS, la confusión entre caballa/sardina es mayor que la confusión entre anchoveta/sardina, alcanzando un 4.2% de caballas mal clasificadas.



(a) (b)

Figura N° 3.45 Matrices de confusión obtenidas para el modelo de clasificación jerárquico Top/Down. a) morfometría + imágenes ▲GOS; b) morfometría + texturas ▲GOS. Fuente: [Elaboración propia]

*Tabla N° 3.17 Resumen de las métricas de desempeño obtenidas para la validación del modelo de clasificación jerárquica considerando las bases de datos **B03** y **B04**.*

	Modelo jerárquico Top/Down – morfometría + imágenes ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.96	0.98	0.91
Caballa	0.80	0.82	0.81	
Jurel	0.91	0.89	0.90	
Sardina	0.94	**0.97**	0.96	
	Modelo jerárquico Top/Down – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	**0.97**	0.98	0.90
Caballa	0.89	0.71	0.79	
Jurel	0.75	0.92	0.83	
Sardina	0.96	**0.97**	0.96	

Por otro lado, el resumen de los resultados obtenidos para el modelo jerárquico multidimensional se muestra en la Tabla N° 3.18, donde se puede apreciar que este es idéntico al obtenido para el modelo jerárquico *Top/Down*. Este resultado es interesante debido a que, según la teoría, el modelo multidimensional es estadísticamente mejor que un modelo jerárquico evaluado siguiendo una estrategia clásica *Top/Down* [49]. Sin embargo, para en este caso particular, dado que todos los minimodelos logran resultados con

métricas MP sobresalientes, sobre 0.95, las probabilidades entregadas por cada minimodelo a la hora de realizar la predicción para un nuevo dato son muy certeras, siendo muy cercana a 1 para la clase final estimada por el modelo, y muy cercanas a 0 para el resto de las clases. Luego, considerando que la rama escogida por el modelo es aquella que logra la mayor multiplicación de probabilidades, la rama final siempre logrará una probabilidad muy cercana a 1, mientras que todo el resto de las ramas opcionales lograrán probabilidades muy cercanas a 0. Con esto se obtiene el mismo comportamiento que el modelo jerárquico *Top/Down*, el cual escoge una única rama en función de las clases con mayor probabilidad para cada nodo del árbol.

*Tabla N° 3.18 Resumen de las métricas de desempeño obtenidas para la validación del modelo de clasificación jerárquica considerando las bases de datos **B03** y **B04**.*

	Modelo jerárquico multidimensional – morfometría + imágenes ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.96	0.98	0.91
Caballa	0.80	0.82	0.81	
Jurel	0.91	0.89	0.90	
Sardina	0.94	**0.97**	0.96	
	Modelo jerárquico multidimensional – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	**0.97**	0.98	0.90
Caballa	0.89	0.71	0.79	
Jurel	0.75	0.92	0.83	
Sardina	0.96	**0.97**	0.96	

Al igual que para las estrategias anteriores, también se probaron las variantes de los modelos con solo 2 especies. La Tabla N° 3.19 muestra un resumen con los mejores resultados obtenidos entre las variantes con morfometría + imágenes de ▲GOS o morfometría + texturas de ▲GOS, mientras que la totalidad de las pruebas se muestran en las tablas de color ◐ presentadas en el Anexo B.2 y en el Anexo B.3, respectivamente para cada variante. A partir de los resultados se puede observar que la clasificación de especies pequeñas y combinadas también logra métricas MP casi ideales, de 0.99, demostrando

que no es estrictamente necesario entrenar un clasificador adicional para lograr bueno resultados. No así para la clasificación de especies grandes, donde solo se logró una métrica MP máxima de 0.89 para la variante a partir de imágenes de ▲GOS, lo cual indica que sí es necesario reforzar la clasificación de este modelo con un reentrenamiento del minimodelo de manchas, con un clasificador final utilizando un modelo jerárquico plano, o ambas opciones, como ya se mostró en la segunda estrategia.

*Tabla N° 3.19 Resumen de las mejores métricas de desempeño obtenidas para la validación del modelo de clasificación jerárquico Top/Down y multidimensional para 2 especies considerando las bases de datos **B03** y **B04**.*

	Modelo jerárquico Top/Down – morfometría + imágenes/texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.98	0.99	0.99
Sardina	0.99	**1.00**	0.99	
	Modelo jerárquico multidimensional – morfometría + imágenes ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Caballa	0.85	**0.90**	0.88	0.89
Jurel	**0.93**	0.89	0.91	
	Modelo jerárquico Top/Down – morfometría + imágenes/texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.98	0.99	0.99
Jurel	0.97	**1.00**	0.99	

3.5. Timing de los Modelos

Otra prueba que se realizó de manera adicional fue medir los tiempos de inferencia para los mejores modelos obtenidos en cada función de la etapa de discriminación de especies.

Para los modelos de detección de objetos de ambas plantas industriales utilizando YOLOv7, el tiempo de inferencia promedio obtenido al procesar una imagen completa fue de 18.5 ms, lo que corresponde a una velocidad de fotogramas de 54.1 FPS.

Para los modelos de detección de *keypoints* utilizando el modelo Keypoint R-CNN, el tiempo de inferencia promedio obtenido al procesar una imagen completa con múltiples peces fue de 142.3ms (7.03 FPS), mientras que el tiempo de inferencia promedio obtenido al procesar una imagen con un único pez fue de 88.4ms (11.3 FPS).

Para los modelos de clasificación, la Tabla N° 3.20 muestra un resumen de los tiempos obtenidos en promedio para las 4 estrategias planteadas. La "Etapa extracción" contempla la etapa de extracción de características morfológicas. La "Etapa inferencia" contempla la etapa de inferencia de los modelos. Para el modelo jerárquico *Top/Down*, este proceso solo contempla la inferencia de los minimodelos para una única rama del árbol de clasificación, mientras que, para el modelo jerárquico multidimensional, este proceso contempla la inferencia de los minimodelos para todas las ramas del árbol de clasificación. La "Etapa postprocesamiento" contempla la etapa de dibujado de los resultados visuales generados durante todo el proceso de discriminación de especies, incluyendo el *bounding box*, los *keypoints* y la especie predicha para cada caso.

Tabla N° 3.20 Resumen de las mediciones de tiempo para los mejores modelos de clasificación de 4 especies.

	Etapa extracción [ms]	Etapa inferencia [ms]	Etapa postprocesamiento [ms]	Tiempo total [ms]
Modelo clasificación multiclase VGG16	-	45.6	251.3	296.9
Modelo jerárquico plano CNN	2.7	58.1	251.3	312.1
Modelo jerárquico Top/Down	2.7	116.1	251.3	370.1
Modelo jerárquico multidimensional	2.7	200.9	251.3	454.9

A partir de la tabla se observa que los tiempos de extracción de características son pequeños y constantes para todos los casos, tardando solo 2.7ms, mientras que los tiempos de postprocesamiento son constantes, pero mucho más altos, tardando 251.3ms, principalmente debido al dibujado de todos los resultados en las imágenes (lo cual es costoso computacionalmente hablando).

Respecto de los tiempos de inferencia, se observa que los tiempos van en aumento a medida que se aumenta la complejidad de los modelos. Si bien el modelo de clasificación VGG16 cuenta con un mayor número de parámetros, este tarda solo 45.6ms en procesar una imagen. Luego, el resto de las implementaciones deben cargar a todos los minimodelos antes de poder hacer la inferencia de una imagen. La diferencia más notoria se observa entre el modelo jerárquico plano y el modelo multidimensional, siendo el primero 3.46 veces más rápido que el segundo, a pesar de que internamente operan de la misma forma (ambos necesitan que los datos de entradas sean evaluados para todos los minimodelos). Esto puede deberse a que el modelo jerárquico plano condensa a todos los minimodelos en una única arquitectura de CNN, lo cual es más eficiente para una GPU que realizar la inferencia en cada minimodelo por separado.

Respecto de los tiempos finales, se observa que, en el mejor caso, el sistema puede operar a 296.9ms (3.37 FPS), mientras que, en el peor caso, este opera a 454.9ms (2.2 FPS). Sin embargo, si se retira la etapa de postprocesamiento del flujo de los datos, el modelo jerárquico plano (que logra los mejores resultados de forma global), podría reducir sus tiempos de inferencia a solo 60.8ms (16.5 FPS).

Como última observación, a pesar de que el modelo jerárquico *Top/Down* logra los mismos resultados que el modelo jerárquico multidimensional, este último requiere probar todos los caminos posibles del árbol para generar un único resultado de clasificación, lo cual crece proporcionalmente en función del número de ramas y del número de minimodelos del árbol. En consecuencia, el modelo multidimensional es naturalmente más lento que el modelo *Top/Down*, lo cual lo vuelve dispensable para los resultados finales de esta tesis.

3.6. Modificación de la Estimación de Talla/Peso

Respecto de la estimación de la talla de las especies identificadas por el sistema de reconocimiento, esta anteriormente se calculaba utilizando el ancho y el alto de los píxeles dentro del *bounding box* entregado por el modelo de detección, representando la longitud total del pez aproximadamente por la diagonal del *bounding box*. La utilización de la longitud total en lugar de la longitud estándar fue necesaria debido a la dificultad inherente a la identificación precisa de la articulación hipural[24] dentro de un *bounding box*, que sirve como base para calcular la longitud estándar. Actualmente, gracias a la integración de los *keypoints*, esta modificación ya no es necesaria, dado que solo se necesita la correcta identificación de los *keypoints* ubicados en el inicio de la cola (**K_7** y **K_8**, según el diagrama mostrado en la Sección 2.5.1), mejorándose en consecuencia la estimación del peso de las especies cuya curva talla/peso sea conocida. Además, se aplicó un factor de conversión para obtener el valor de longitud en centímetros, que corresponde a la distancia real representada por un solo píxel en el mundo real. Este factor de conversión se determinó de antemano, calibrando el sistema con una moneda de dimensiones conocidas.

[24] Articulación entre la aleta caudal y la última vértebra del pez.

CAPÍTULO 4. CONCLUSIÓN

4.1. Discusión General

En el capítulo anterior se presentaron los experimentos y resultados más importantes relacionados con la etapa de discriminación de peces del sistema de reconocimiento, los cuales se dividieron entre las aplicaciones realizadas para la industria, con los modelos de detección de peces en correas transportadoras, y las aplicaciones realizadas a nivel de laboratorio, explorando los modelos de detección de *keypoints* y los de clasificación jerárquica siguiendo una taxonomía de características morfológicas.

Respecto de los resultados obtenidos para la industria, la nueva variante del sistema utilizando YOLOv7 permite la identificación de distintas especies con altos valores de precisión, de lo cual se destaca que 5 de 6 especies alcanzan valores de la métrica MP sobre 0.8, con excepción de sierra, y 4 de 6 especies alcanzan valores de la métrica MP sobre 0.9, con excepción de sierra y sardina. Sin embargo, se mantiene el escenario donde el modelo confunde especies que son visualmente muy similares entre sí, como es el caso de las especies caballa-jurel y jurel-sierra para la Planta Blumar. Acerca de las especies pequeñas, como anchoveta y sardina, los resultados obtenidos para la Planta Orizon indican que el modelo si logró disminuir la confusión entre ambas especies, pero se mantiene el fenómeno donde hay un porcentaje significativo de detecciones que se confunden con el fondo de las imágenes para ambas plantas (*i.e.* como parte de la correa transportadora), a pesar de las técnicas de preprocesamiento utilizadas para disminuir el foco del fondo por sobre las especies de peces.

En términos de consideraciones prácticas, cuando se utiliza YOLOv7 para la clasificación en tiempo real de especies de peces con diferentes tamaños, existe un delicado

equilibrio entre la velocidad de fotogramas (FPS) y la resolución de la imagen. Respecto de los FPS, estos fueron ajustados de tal forma que se sincronizaran con la velocidad de las correas transportadoras, asegurando que un mismo pez no aparezca en dos imágenes consecutivas. Esta sencilla optimización permite una mejor utilización de los recursos computacionales para procesar varias imágenes simultáneamente. En cuanto a la resolución de la imagen, esta juega un papel crucial en la distinción de especies más pequeñas o de tamaño similar, como las anchovetas o las sardinas. En este sentido, la decisión fue mantener la resolución de la imagen de entrada predeterminada de YOLOv7 (640x640). Esta elección da como resultado unos FPS de inferencia 6,75 veces más rápidos que los FPS de la cámara en la Planta Orizon y 54 veces más rápidos que los FPS de la cámara en la Planta Blumar. Si bien optar por una resolución más alta podría mejorar las capacidades de extracción de características de los modelos, esto produciría una disminución en los FPS de inferencia, lo que limitaría la capacidad del sistema para procesar múltiples imágenes de una misma planta al mismo tiempo.

Por otro lado, es importante mencionar las limitaciones inherentes de los pórticos en ambientes industriales. Entre estas limitaciones se encuentran: la capacidad del sistema para detectar únicamente las especies que se encuentran en la capa superior de la cinta transportadora, la sensibilidad del sistema a objetos superpuestos desconocidos para las bases de datos, y la incapacidad del sistema para clasificar a los peces que se presentan en un estado deteriorado, como aquellos que están seccionados o molidos. Además de estas consideraciones, se han observado fenómenos que generan fuentes de error ambientales, como la luz solar intensa, que puede provocar una sobresaturación de píxeles en las imágenes, así como condiciones de humedad que pueden dar lugar al empañamiento de las cámaras; siendo factores que pueden reducir significativamente la capacidad de detección de los modelos y, en algunos casos, requerir intervenciones manuales por parte de los operadores de planta para solucionarlos.

Abordando los resultados obtenidos para el laboratorio, las pruebas relacionadas con los modelos para la detección automática de *keypoints* se mostraron prometedoras a pesar de la baja cantidad de imágenes que se tenían disponibles originalmente (solo 3 imágenes con múltiples ejemplares por especie, abarcando hasta un máximo de 30 muestras, según se puede observar en la segunda columna de la Tabla N° 2.2). Los mejores resultados se lograron para el modelo de *keypoints* entrenado con peces individuales en fondos *random*, el cual alcanzó un valor máximo de 0.706 de la métrica AP@[0.5:0.95] y, además, logró el valor más alto de la métrica PCK en relación con el resto de variantes, con un 89.89%, probando ser robusto tanto para imágenes desconocidas para el modelo (caballas) como ante las mismas imágenes de los fondos *random* (sin peces). Esto revela que la métrica PCK tiene más peso que la métrica AP sobre el rendimiento alcanzado por los modelos, pero es muy susceptible a los cambios en la distribución de los *keypoints*, por ejemplo, debido al error humano introducido al momento de etiquetar la base de datos, o debido a una imagen donde solo se aprecie a uno o varios peces de forma parcial, lo cual puede desviar, acumular en un solo punto u omitir uno o más *keypoints*.

Respecto de la localización en sí de los *keypoints*, para las distintas pruebas realizadas, los resultados revelan que los *keypoints* **K_3** y **K_4** lograron los errores más bajos, con un MAE menor a 10. Esto puede deberse a que ambos representan la ubicación de la cabeza de los peces, la cual es fácil de observar a simple vista. De la misma forma, el *keypoint* que podría ser más difícil de reconocer es **K_2**, puesto que este es una proyección del término de la boca de los peces sobre su propia cabeza, y no apunta directamente a una característica en particular, como si ocurre en el resto de los casos. Sin embargo, contrario a lo esperado, los resultados muestran que el MAE de este *keypoint* en particular se encuentra dentro de la media, mientras que los más difíciles de detectar en general son aquellos que representan la relación de ancho del cuerpo (**K_5** y **K_6**) o la ubicación del inicio de la cola (**K_7** y **K_8**). Particularmente, para **K_5** y **K_6**, su desviación del *ground truth* puede estar relacionada con la selección de estos *keypoints* como los puntos más altos y bajos en el cuerpo del pez (proporcionando una estimación aproximada del ancho), siendo

una medida que puede derivarse de diversas áreas del cuerpo de un pez al mismo tiempo sin pérdida de generalidad, especialmente en especies de peces con morfologías alargadas, como la anchoveta o la merluza.

En base al diseño dual escogido para la detección de *keypoints*, la combinación de modelos de detección, como YOLOv7, seguidos de modelos de detección de *keypoints*, como Keypoint R-CNN, puede ofrecer una mejora significativa en la comprensión y descripción de la anatomía de los objetos en una imagen, a costa de aumentar los recursos computacionales utilizados de la GPU y los tiempos de procesamiento. Si bien la estrategia pudiera parecer redundante al ser ambos modelos capaces de detectar objetos en imágenes, por un lado, YOLOv7 es excepcionalmente rápido y preciso para llevar a cabo esta tarea, incorporando además la generación de máscaras que son útiles para segmentar a cada pez antes de ingresar al modelo de *keypoints*, mientras que, por otro lado, Keypoint R-CNN es un modelo versátil a la hora de adaptarse a nuevas bases de datos, requiriendo pocas imágenes para lograr buenos resultados, y es capaz de generar *keypoints* con una gran precisión dentro de una región de interés focalizada en un único pez.

En caso de que esta modalidad de operación sea llevada a un ambiente industrial, fácilmente se podría acoplar la salida de los modelos de detección ya entrenados para cada planta con un modelo de detección de *keypoints*, los cuales podrían ser reentrenados con fines generales (preparados para identificar *keypoints* en una gran variedad de especies de peces) o con fines locales (preparados para identificar *keypoints* en las especies de interés de la planta). No obstante, es importante tener en consideración los tiempos de operación del sistema como un todo. Esto debido a que, actualmente, los modelos de *keypoints* son capaces de procesar una imagen con un único pez entre 50 y 100 ms; tiempo que debe multiplicarse por cada pez detectado por los modelos de detección. En caso de que se quisiera mantener la condición de operación en tiempo real del sistema de reconocimiento, utilizar esta estrategia en particular no sería viable, al menos dentro del flujo principal del programa. Esto se refuerza por el hecho de que el funcionamiento del modelo de *keypoints*

en imágenes obtenidas de la industria es un campo no explorado en esta tesis, por lo que puede estar sujeto a modificaciones, especialmente debido a que el modelo necesita una muestra completa de un pez para operar correctamente, lo cual puede verse perturbado debido a la condición de la pesca o al grado de solape entre los peces (que depende de la forma de operación de cada planta y de cada correa transportadora).

Considerando las pruebas relacionadas con los modelos de clasificación jerárquica, queda demostrado que los modelos de clasificación multiclase sin jerarquía generalizan de manera deficiente (métrica P bajo 0.85) al menos 2 de las 4 especies de peces problemáticas. El mejor resultado fue logrado por el modelo de tipo FCNN, que toma directamente todas las características extraídas a partir de las imágenes de los peces (distancias morfométricas + texturas) como datos de entrada, destacando su capacidad para clasificar anchoveta y sardina. Esto es contrario a lo esperado, debido a que las diferencias claves entre anchoveta y sardina se suelen esconder más detrás de su morfología propia. No así las diferencias entre jurel y caballa, que son especies muy visualmente diferenciables entre sí debido al patrón de manchas en zigzag que solo presentan las caballas en su zona dorsal.

Si bien lo anterior puede atribuirse al tamaño de la base de datos, donde 3 de las 4 especies problemáticas cuentan con menos de 30 muestras originales (menos de 200 si se considera las imágenes con *data augmentation*), los resultados obtenidos para cada nivel o nodo del clasificador jerárquico muestran que no es necesario contar con una base de datos numerosa para lograr un buen rendimiento de los modelos. Particularmente, los 4 minimodelos seleccionados para construir el árbol de clasificación (tamaño, forma, boca y manchas) alcanzan valores de la métrica MP notablemente altos, con un máximo entre 0.95 y 1.00 para al menos una de las estrategias propuestas en cada caso. De entre todas las variantes de los minimodelos que fueron probadas, las más desafiantes fueron sin lugar a duda las del minimodelo de manchas. Esto se debió principalmente a que fue complejo enseñarles a los modelos a diferenciar entre las manchas características de las caballas

respecto de otras características presentes en la piel de los peces, como zonas con cambios de coloración, lunares, escamas o sangre (en algunos casos); aun focalizando la clasificación dentro de una región de interés determinada por lo *keypoints*. Por esta razón, se utilizó el filtro GOS (Gabor-Otsu-Sobel) para realzar las manchas respecto de otras texturas más sutiles presentes en la imagen, logrando resultados de hasta un 0.95 para la métrica MP. Sin embargo, en caso de que se introduzcan nuevas especies de peces a la base de datos con patrones de manchas diferentes, tanto las etapas de preprocesamiento como el número de clases del minimodelo de manchas necesarias pueden diferir de la propuesta utilizada en esta tesis. Otra posibilidad es no solo ampliar el número de clases para cubrir los distintos patrones de manchas, sino que también extender la profundidad del árbol para abarcar un mayor número de características de manera más específica y focalizada en función de alguna región de interés seleccionada. Por ejemplo, la rama de clasificación orientada para el jurel podría extenderse en un nivel si se considerara un modelo para identificar la mancha característica que se presenta en el borde dorsal de su opérculo.

Finalmente, a partir de las pruebas que involucran los modelos jerárquicos, es posible concluir que, para clasificar entre las 4 especies problemáticas al mismo tiempo, el modelo jerárquico plano es el que entrega los mejores resultados, presentando una métrica MP por sobre 0.93 para todos los clasificadores, siendo el clasificador con FCNN el mejor de todos (por un pequeño margen). Este resultado es interesante, puesto que el modelo jerárquico plano realmente ignora la jerarquía interna establecida entre los minimodelos, actuando como una red de extracción de características mucho más específica que los *backbones* tradicionales, donde la idea es cubrir un amplio rango de características, desde las más generales hasta las más abstractas. Otra ventaja respecto de utilizar FCNN en lugar de algoritmos de clasificación del estado del arte más tradicionales, es que el entrenamiento se puede adaptar para reentrenar los pesos obtenidos para algún minimodelo en particular, lo cual es útil cuando se tiene un minimodelo que fue entrenado para datos muy generales (por ejemplo, para todos los patrones de manchas existentes), pero solo se desea

clasificar datos más específicos (por ejemplo, un único patrón de manchas), siguiendo una lógica similar a la técnica de *transfer learning*.

De manera adicional, también se probaron otras variantes de los modelos jerárquicos para clasificar solamente entre algunos pares de clases, como especies grandes (caballa y jurel), especies pequeñas (anchoveta y sardina) y especies combinadas (anchoveta y jurel). De aquí se destacan los resultados obtenidos para los modelos de especies pequeñas y especies combinadas, donde todas las variantes de clasificadores jerárquicos, incluyendo FCNN y los clasificadores tradicionales para los modelos jerárquicos planos, alcanzaron valores de MP entre 0.97 y 1.00 para ambos casos. Esto es un claro reflejo de que las principales diferencias entre anchoveta y sardina o anchoveta y jurel si pueden abordarse desde una perspectiva morfológica, utilizando para ello solo mediciones morfométricas extraídas a partir de *keypoints* generales para muchas especies de peces, superando con creces los resultados obtenidos durante la etapa de validación que realizó Sernapesca, donde el MP para peces pequeños solo fue de 0.85. Desde el punto de vista de las especies grandes, sin bien los resultados de validación de Sernapesca lograron una métrica MP de 0.93, estos fueron superados por el clasificador jerárquico plano utilizando FCNN, alcanzando un MP de 0.98. Sin embargo, todavía se percibe una gran resistencia por parte de los modelos para aprender de manera efectiva las características necesarias para diferenciar una imagen entre jurel y caballa. Esto implicaría, en un futuro, abordar otras diferencias morfológicas existentes entre ambas especies, como las diferencias de largo de la aleta pectoral o la visibilidad de la línea lateral, para lo cual si se necesita agregar una mayor cantidad de *keypoints* que no sean tan generales como los 8 que fueron utilizados durante esta tesis.

Como un último detalle a destacar sobre los modelos jerárquicos locales y multidimensionales, es importante mencionar que su rendimiento fue exactamente el mismo en todos los casos, aun cuando el modelo multidimensional es estadísticamente más preciso que el modelo local. Este fenómeno puede explicarse debido a que todos los minimodelos

entregaban probabilidades muy certeras para su clasificación final (por ejemplo, [0.9999, 0.0001] para un modelo de dos clases), lo cual reduce en gran medida las probabilidades de las ramas opcionales del árbol de clasificación que se calculan para el modelo multidimensional.

4.2. Conclusiones Generales

En este trabajo se ha presentado un sistema de visión computacional para el reconocimiento de especies pelágicas pequeñas de la costa chilena, el cual combina los paradigmas de detección de objetos, detección de *keypoints* y clasificación jerárquica utilizando modelos de *deep learning*. El trabajo realizado se dividió entre una aplicación llevada a cabo en la industria pesquera, ajustada para dos plantas de desembarque, con el fin de implementar un modelo de detección de objetos capaz de operar en tiempo real; y una aplicación llevada a cabo en ambiente controlado de laboratorio, con el fin de realizar una discriminación robusta de una especie de pez a través de la extracción de características morfológicas relevantes de forma automatizada. Los resultados obtenidos demuestran la capacidad del sistema para detectar y distinguir diferentes especies con altos niveles de precisión, así como los desafíos asociados con ciertas especies que exhiben similitudes visuales, y las limitaciones asociadas al entrenamiento de modelos con bases de datos pequeñas.

Por el lado de la industria, la implementación del sistema de detección en tiempo real representa una mejora significativa en comparación con los procedimientos de inspección actuales, proporcionando un método automatizado para la identificación de peces en línea junto con la estimación de su talla y su peso. El sistema es capaz de proporcionar una trazabilidad más precisa y fiable de los desembarques de pescado a los organismos de ordenación pesquera, lo que podría contribuir a la sostenibilidad de los recursos marinos y a la viabilidad a largo plazo de la industria pesquera.

Por el lado de las pruebas de laboratorio, la extracción de características morfológicas ha demostrado ser fundamental para mejorar significativamente la capacidad de los modelos para diferenciar entre especies de peces muy similares entre sí, siendo la clasificación jerárquica una estrategia precisa y mucho más adaptable que los enfoques de clasificación tradicionales para abordar la identificación de características morfológicas de manera específica.

En el contexto de este trabajo, es importante enfatizar que la complejidad de los modelos de *machine learning* no necesariamente se traduce en un mejor desempeño, ya que esto puede dar lugar a problemas de generalización y a una ineficiencia computacional que no van de la mano con la operación de un sistema en tiempo real. Más aún, considerando que la visibilidad de la inteligencia artificial se ha ido propagando con fuerza durante los últimos años, mejorar la transparencia y la comprensión de los métodos utilizados es clave para que estas herramientas se puedan utilizar con mayor confianza y no sean percibidas como cajas negras por el personal experto en otras áreas que no van tradicionalmente de la mano con este tipo de soluciones, como lo es en este caso particular la biología marina y el ambiente industrial pequero.

4.3. Trabajo Futuro

Algunas de las ideas que pueden abordarse dentro de un trabajo futuro se enlistan a continuación:

- Contemplar la instalación y el manejo de un mayor número de pórticos en plantas industriales, lo cual permitiría conocer nuevos entornos y, potencialmente, incorporar un mayor número de especies en las bases de datos.
- Mejora del rendimiento del módulo de estimación de peso y talla.
- Probar otras variantes de modelos de detección de *keypoints* que permitan el procesamiento del sistema en tiempo real. Esto no solo implica encontrar el modelo adecuado, sino que también implica manejar de forma más eficiente la disposición de los peces dentro de una imagen para mejorar los resultados obtenidos.

- Probar a entrenar modelos que extraigan un mayor número de *keypoints* para poder obtener características morfológicas más específicas para ciertas especies de peces. Si bien hoy en día no se han dado a conocer modelos multiclase que permitan una cantidad diferente de *keypoints* para cada clase, no se descarta que esto sea posible en un futuro.
- Extraer características morfológicas o mediciones morfométricas de interés para el personal experto, en lugar de las mediciones extraídas únicamente a partir del *box truss*.
- Ampliar las posibilidades de los modelos jerárquicos al testear un mayor número de clasificadores para los minimodelos, con tal de encontrar un mejor balance entre los tiempos de procesamiento y la complejidad de estos.
- Extender el modelo jerárquico para clasificar entre un mayor número de especies de peces.
- Extender o crear nuevos modelos jerárquicos para otro tipo de especies de animales marinos de interés, como moluscos o crustáceos.

4.4. Logros de Investigación

1. Participación en Proyecto FONDEF IT20I0032 y en el Proyecto ANILLO ACT210073, como tesista de postgrado.

2. Elaboración del *paper*: "*Vincenzo Caro Fuentes, Ariel Torres Sánchez, Jorge E. Pezoa, Sergio N. Torres, Rosario Castillo, Rubén Escribano, Mauricio Urbina, "SAFE: a deep-learning-based software for catch-control of small-scale fishing boats in Chile," Proc. SPIE 12227, Applications of Machine Learning 2022, 1222706 (3 October 2022); https://doi.org/10.1117/12.2628826*"

3. Asistencia a la Conferencia SPIE OPTICS and PHOTONICS 2022 en San Diego, EE. UU., como presentador del *paper* mencionado en el punto anterior.

4. Elaboración del *paper*: "*Caro Fuentes, V.; Torres, A.; Luarte, D.; Pezoa, J.E.; Godoy, S.E.; Torres, S.N.; Urbina, M.A. Digital Classification of Chilean Pelagic Species in Fishing Landing Lines. Sensors 2023, 23, 8163. https://doi.org/10.3390/s23198163*", donde se detalla la primera mitad del trabajo presentado en esta tesis, abarcando todo lo relacionado con los modelos de detección de objetos en tiempo real en la industria pesquera.

5. *Paper* de título tentativo "*Morphological Feature Extraction based on Deep Learning and Feature Engineering Application for Discriminating Small Pelagic Species in Chile*", el cual contempla la segunda mitad del trabajo presentado en esta tesis, abarcando todo lo relacionado con los modelos de *keypoints* y la clasificación jerárquica a partir de características morfológicas. Actualmente en proceso de redacción.

ABREVIACIONES

AI	: Artificial Intelligence
ANID	: Agencia Nacional de Investigación y Desarrollo
AP	: Average Precision
API	: Application Programming Interface
CNN	: Convolutional Neural Network
COCO	: Common Objects in Context
CPU	: Central Processing Unit
CV	: Computer Vision
DCT	: Discrete Cosine Transformation
DL	: Deep Learning
FAIR	: Facebook AI Research
FC	: Fully Connected
FCNN	: Fully Connected Neural Network
FPS	: Frames per Second
HOG	: Histogram of Oriented Gradients
GLCM	: Gray-level Cooccurrence Matrix
GOS	: Gabor-Otsu-Sobel
GPU	: Graphics Processing Unit
IA	: Artificial Intelligence
IDE	: Integrated Development Environment
KNN	: K-Nearest Neighbors
LBP	: Local Binary Patterns
MAE	: Mean Absolute Error
mAP	: Mean Average Precision
ML	: Machine Learning
MP	: Macro-average Precision
NMS	: Non-maximum Suppression
P	: Precision
PCK	: Percentage of Correct Keypoints
R	: Recall
R-CNN	: Region-Based Convolutional Neural Network
R-FCN	: Region-Based Fully Convolutional Network

RoI	: Region of Interest
RPN	: Region Proposal Network
SAFE	: Sistema de Apoyo a Fiscalización, Información Espacial
SAFES	: Sistema de Apoyo a Fiscalización, Información Espacio-Espectral
SAM	: Segment Anything Model
SERNAPESCA	: Servicio Nacional de Pesca y Acuicultura
SVM	: Support Vector Machine
YOLO	: You Only Look Once

BIBLIOGRAFÍA

[1] FAO, “Chile comparte experiencias para aumentar el consumo de pescado”, FAO en Chile - Noticias. Accedido: 16 de octubre de 2023. [En línea]. Disponible en: https://www.fao.org/chile/noticias/detail-events/es/c/1542532/#:~:text=Seg%C3%BAn%20cifras%20de%20la%20FAO%2C%20Chile%20se%20encuentra,cinco%20principales%20exportadores%20de%20pescado%20y%20productos%20pesqueros.

[2] “Servicio Nacional de Pesca y Acuicultura”. Accedido: 20 de diciembre de 2021. [En línea]. Disponible en: http://www.sernapesca.cl/

[3] D. Ramírez, “Clasificación Multiclase de especies pelágicas pequeñas basadas en huellas espectrales”, Memoria de Título, Ingeniería Civil en Telecomunicaciones, Universidad de Concepción, Concepción, 2018.

[4] A. Díaz, “SAFE: Software de asistencia para los procesos de fiscalización de especies pelágicas de peces mediante la clasificación espacial”, Memoria de Título, Ingeniería Civil en Telecomunicaciones, Universidad de Concepción, Concepción, 2020.

[5] V. Caro, “ SAFE 2.0: Software de Clasificación de Especies y Estimación de Tallas de Pescado en la Banda Visible usando Algoritmos de Deep Learning”, Memoria de Título, Ingeniería Civil Electrónica, Universidad de Concepción, Concepción, 2021.

[6] D. Ramírez, “Deep learning e imagenología híperespectral para la clasificación de especies pelágicas pequeñas”, Tesis de maestría. Magíster en Ciencias de la Ingeniería con mención en Ingeniería Eléctrica, Universidad de Concepción, Concepción, 2019.

[7] J. G. A. Barbedo, “A Review on the Use of Computer Vision and Artificial Intelligence for Fish Recognition, Monitoring, and Management”, *Fishes*, vol. 7, nº 6. MDPI, 1 de diciembre de 2022. doi: 10.3390/fishes7060335.

[8] A. S. Abangan, D. Kopp, y R. Faillettaz, “Artificial intelligence for fish behavior recognition may unlock fishing gear selectivity”, *Frontiers in Marine Science*, vol. 10. Frontiers Media S.A., 2023. doi: 10.3389/fmars.2023.1010761.

[9] R. B. Duran *et al.*, “Multimode hyperspectral data fusion for fish species identification using supervised and reinforcement learning”, en *Proceedings Volume*

11421, Sensing for Agriculture and Food Quality and Safety XII; 114210L (2020), SPIE-Intl Soc Optical Eng, abr. 2020, p. 22. doi: 10.1117/12.2559226.

[10] C. Spampinato, D. Giordano, R. Di Salvo, Y.-H. Chen-Burger, R. B. Fisher, y G. Nadarajan, "Automatic Fish Classification for Underwater SpeciesBehavior Understanding", en *ARTEMIS '10: Proceedings of the first ACM international workshop on Analysis and retrieval of tracked events and motion in imagery streams*, Association for Computing Machinery, oct. 2010, p. 92. doi: 10.1145/1877868.1877881.

[11] P. A. Coelho-Caro, C. E. Saavedra-Rubilar, J. P. Staforelli, M. J. Gallardo-Nelson, V. Guaquin, y E. Tarifeño, "Mussel Classifier System Based on Morphological Characteristics", *IEEE Access*, vol. 6, pp. 76935–76941, 2018, doi: 10.1109/ACCESS.2018.2884394.

[12] C. Pornpanomchai, B. Lurstwut, P. Leerasakultham, y W. Kitiyanan, "Shape-and Texture-Based Fish Image Recognition System", *Kasetsart Journal - Natural Science*, vol. 47, pp. 624–634, 2013.

[13] M. Khalil Sari Alsmadi, P. Khairuddin Bin Omar, P. Azman Noah, y I. Almarashdah, "FISH RECOGNITION BASED ON THE COMBINATION BETWEEN ROBUST FEATURES SELECTION, IMAGE SEGMENTATION AND GEOMETRICAL PARAMETERS TECHNIQUES USING ARTIFICIAL NEURAL NETWORK AND DECISION TREE", *International Journal of Computer Science and Information Security*, vol. 6, nº 2, pp. 215–221, 2009, Accedido: 23 de mayo de 2022. [En línea]. Disponible en: https://doi.org/10.48550/arXiv.0912.0986

[14] S. N. Gowda y C. Yuan, "ColorNet: Investigating the importance of color spaces for image classification", feb. 2019, [En línea]. Disponible en: http://arxiv.org/abs/1902.00267

[15] T. Saitoh, T. Shibata, y T. Miyazono, "Feature Points based Fish Image Recognition", 2016. [En línea]. Disponible en: www.mirlabs.net/ijcisim/index.html

[16] A. Saleh, D. Jones, D. Jerry, y M. R. Azghadi, "A lightweight Transformer-based model for fish landmark detection", sep. 2022, [En línea]. Disponible en: http://arxiv.org/abs/2209.05777

[17] F. Suo, K. Huang, G. Ling, Y. Li, y J. Xiang, "Fish Keypoints Detection for Ecology Monitoring Based on Underwater Visual Intelligence", en *16th IEEE International Conference on Control, Automation, Robotics and Vision, ICARCV 2020*, Institute of Electrical and Electronics Engineers Inc., dic. 2020, pp. 542–547. doi: 10.1109/ICARCV50220.2020.9305424.

[18] B. Lin *et al.*, “Feasibility research on fish pose estimation based on rotating box object detection”, *Fishes*, vol. 6, nº 4, dic. 2021, doi: 10.3390/fishes6040065.

[19] K. He, G. Gkioxari, P. Dollár, y R. Girshick, “Mask R-CNN”, *IEEE Trans Pattern Anal Mach Intell*, vol. 42, nº 2, pp. 386–397, mar. 2017, doi: 10.48550/arxiv.1703.06870.

[20] C. Yu *et al.*, “Segmentation and measurement scheme for fish morphological features based on Mask R-CNN”, *Information Processing in Agriculture*, vol. 7, nº 4, pp. 523–534, dic. 2020, doi: 10.1016/j.inpa.2020.01.002.

[21] J. Liang, N. Homayounfar, W.-C. Ma, Y. Xiong, R. Hu, y R. Urtasun, “PolyTransform: Deep Polygon Transformer for Instance Segmentation”, dic. 2019, [En línea]. Disponible en: http://arxiv.org/abs/1912.02801

[22] F. Husain, H. Schulz, B. Dellen, C. Torras, y S. Behnke, “Combining Semantic and Geometric Features for Object Class Segmentation of Indoor Scenes”, *IEEE Robot Autom Lett*, vol. 2, nº 1, pp. 49–55, ene. 2017, doi: 10.1109/LRA.2016.2532927.

[23] Z. Hayder, X. He, y M. Salzmann, “Boundary-aware Instance Segmentation”, dic. 2016, [En línea]. Disponible en: http://arxiv.org/abs/1612.03129

[24] Z. Zhang, X. Du, L. Jin, D. Han, C. Li, y X. Liu, “Discriminative feature learning for underwater fish recognition”, *J Electron Imaging*, vol. 30, nº 02, abr. 2021, doi: 10.1117/1.jei.30.2.023020.

[25] J. H. Lee, M. Y. Wu, y Z. C. Guo, “A tank fish recognition and tracking system using computer vision techniques”, en *Proceedings - 2010 3rd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2010*, 2010, pp. 528–532. doi: 10.1109/ICCSIT.2010.5563625.

[26] Y. Jianping, J. Dong, X. Sun, C. Wanga, y X. Wang, “Low-contrast underwater living fish recognition using PCANet”, SPIE-Intl Soc Optical Eng, abr. 2018, p. 63. doi: 10.1117/12.2302695.

[27] A. Ben Tamou, A. Benzinou, K. Nasreddine, y L. Ballihi, “Underwater live fish recognition by deep learning”, en *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2018, pp. 275–283. doi: 10.1007/978-3-319-94211-7_30.

[28] H. Qin, X. Li, J. Liang, Y. Peng, y C. Zhang, “DeepFish: Accurate underwater live fish recognition with a deep architecture”, *Neurocomputing*, vol. 187, pp. 49–58, abr. 2016, doi: 10.1016/j.neucom.2015.10.122.

[29] M. C. Chuang, J. N. Hwang, y K. Williams, “A feature learning and object recognition framework for underwater fish images”, *IEEE Transactions on Image Processing*, vol. 25, nº 4, pp. 1862–1872, abr. 2016, doi: 10.1109/TIP.2016.2535342.

[30] X. Li, M. Shang, J. Hao, y Z. Yang, “Accelerating fish detection and recognition by sharing CNNs with objectness learning”, en *OCEANS 2016 - Shanghai*, Institute of Electrical and Electronics Engineers Inc., jun. 2016. doi: 10.1109/OCEANSAP.2016.7485476.

[31] X. Li, M. Shang, H. Qin, y L. Chen, “Fast accurate fish detection and recognition of underwater images with Fast R-CNN”, en *OCEANS 2015 - MTS/IEEE Washington*, Institute of Electrical and Electronics Engineers Inc., feb. 2016. doi: 10.23919/oceans.2015.7404464.

[32] M. R. Shortis *et al.*, “A review of techniques for the identification and measurement of fish in underwater stereo-video image sequences”, en *Videometrics, Range Imaging, and Applications XII; and Automated Visual Inspection*, SPIE, may 2013, p. 87910G. doi: 10.1117/12.2020941.

[33] W. Jia, Y. Tian, R. Luo, Z. Zhang, J. Lian, y Y. Zheng, “Detection and segmentation of overlapped fruits based on optimized mask R-CNN application in apple harvesting robot”, *Comput Electron Agric*, vol. 172, may 2020, doi: 10.1016/j.compag.2020.105380.

[34] Y. Yu, K. Zhang, L. Yang, y D. Zhang, “Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN”, *Comput Electron Agric*, vol. 163, ago. 2019, doi: 10.1016/j.compag.2019.06.001.

[35] S. Luo, X. Li, D. Wang, J. Li, y C. Sun, “Automatic Fish Recognition and Counting in Video Footage of Fishery Operations”, en *Proceedings - 2015 International Conference on Computational Intelligence and Communication Networks, CICN 2015*, Institute of Electrical and Electronics Engineers Inc., ago. 2016, pp. 296–299. doi: 10.1109/CICN.2015.66.

[36] T. C. Vannoy *et al.*, “Machine learning-based region of interest detection in airborne lidar fisheries surveys”, *J Appl Remote Sens*, vol. 15, nº 03, jul. 2021, doi: 10.1117/1.jrs.15.038503.

[37] E. Bisong, “Google Colaboratory”, en *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, Berkeley, CA: Apress, 2019, pp. 59–64. doi: 10.1007/978-1-4842-4470-8_7.

[38] “HikVision”. Accedido: 22 de noviembre de 2023. [En línea]. Disponible en: https://www.hikvision.com/en/

[39] G. Georgiou, “Fish Species”, Kaggle Datasets. Accedido: 23 de noviembre de 2023. [En línea]. Disponible en: https://www.kaggle.com/datasets/giannisgeorgiou/fish-species/data

[40] S. Srinivasan, “Fish Species Image Data”, Kaggle Datasets. Accedido: 23 de noviembre de 2023. [En línea]. Disponible en: https://www.kaggle.com/datasets/sripaadsrinivasan/fish-species-image-data/data

[41] E. A., “Detección de objetos con YOLO: implementaciones y como usarlas”. Accedido: 13 de octubre de 2023. [En línea]. Disponible en: https://medium.com/@enriqueav/detecci%C3%B3n-de-objetos-con-yolo-implementaciones-y-como-usarlas-c73ca2489246

[42] C.-Y. Wang, A. Bochkovskiy, y H.-Y. M. Liao, “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”, jul. 2022, [En línea]. Disponible en: http://arxiv.org/abs/2207.02696

[43] A. Bochkovskiy, C.-Y. Wang, y H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection”, abr. 2020, doi: 10.48550/arxiv.2004.10934.

[44] P. Potrimba, “What is Keypoint Detection?”, Computer Vision, Keypoint Detection. Accedido: 23 de noviembre de 2023. [En línea]. Disponible en: https://blog.roboflow.com/what-is-keypoint-detection/

[45] W. Zhang, C. Fu, y M. Zhu, “Joint Object Contour Points and Semantics for Instance Segmentation”, ago. 2020, [En línea]. Disponible en: http://arxiv.org/abs/2008.00460

[46] K. Simonyan y A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”, *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, sep. 2014, doi: 10.48550/arxiv.1409.1556.

[47] K. He, X. Zhang, S. Ren, y J. Sun, “Deep Residual Learning for Image Recognition”, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 770–778, dic. 2015, doi: 10.48550/arxiv.1512.03385.

[48] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge”, sep. 2014, [En línea]. Disponible en: http://arxiv.org/abs/1409.0575

[49] J. N. Hernández Torres, “Clasificación Jerárquica Multidimensional”, Tesis de Maestría. Maestro en Ciencias en la Especialidad de Ciencias Computacionales, Instituto Nacional de Astrofísica, Óptica y Electrónica, Tonantzintla, Puebla, México, 2012.

[50] “Roboflow”, Everything you need to build and deploy computer vision models. Accedido: 23 de noviembre de 2023. [En línea]. Disponible en: https://roboflow.com/

[51] A. Kirillov *et al.*, “Segment Anything”, abr. 2023, [En línea]. Disponible en: http://arxiv.org/abs/2304.02643

[52] J. Brooks, “COCO Annotator”. Accedido: 24 de noviembre de 2023. [En línea]. Disponible en: https://github.com/jsbroks/coco-annotator/

[53] K.-C. Ng C., A.-C. Ooi P., W. W.L., y K. G., “A Review of Fish Taxonomy Conventions and Species Identification Techniques”, *J Surv Fish Sci*, vol. 4, nº 1, pp. 54–93, 2017.

[54] A. Angulo, A. R. Ramírez Coghi, y M. López Sánchez, “Claves para la identificación de los peces de las aguas continentales e insulares de Costa Rica. Parte I: Familias”, *UNED Research Journal*, vol. 13, nº 1, p. 27, mar. 2021, doi: 10.22458/urj.v13i1.3145.

[55] A. F. González Acosta, “ESTUDIO SISTEMÁTICO Y BIOGEOGRÁFICO DEL GÉNERO Eugerres (PERCIFORMES: GERREIDAE)”, Tesis de Doctorado, Instituto Politécnico Nacional, La Paz, Baja California Sur, 2005.

[56] W. J. Rainboth, “Fishes of the Cambodian Mekong”, *FAO*. p. 265, 1996. Accedido: 7 de enero de 2023. [En línea]. Disponible en: http://library.enaca.org/inland/fishes-cambodian-mekong.pdf

[57] “Biodiversity and Morphology”, FishBase. Accedido: 23 de octubre de 2023. [En línea]. Disponible en: https://fishbase.mnhn.fr/fishonline/english/fol_biodiversityandmorphology.htm#tab3_2

[58] J. C. Howe, “Standard length: not quite so standard”, 2002.

[59] M. H. Ha, Y. G. Kim, y T. H. Park, “Stain Defect Classification by Gabor Filter and Dual-Stream Convolutional Neural Network”, *Applied Sciences (Switzerland)*, vol. 13, nº 7, abr. 2023, doi: 10.3390/app13074540.

[60] X. Zhou, D. Wang, y P. Krähenbühl, “Objects as Points”, abr. 2019, [En línea]. Disponible en: http://arxiv.org/abs/1904.07850

[61] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, y R. Girshick, “Detectron2”. 2019. Accedido: 14 de noviembre de 2023. [En línea]. Disponible en: https://github.com/facebookresearch/detectron2

[62] G. Chen, P. Sun, y Y. Shang, “Automatic fish classification system using deep learning”, en *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, IEEE Computer Society, jul. 2018, pp. 24–29. doi: 10.1109/ICTAI.2017.00016.

[63] N. Bakerid, H. Lu, G. Erlikhmanid, y P. J. Kellman, “Deep convolutional networks do not classify based on global object shape”, 2018, doi: 10.1371/journal.pcbi.1006613.

[64] N. Bakerid, H. Lu, G. Erlikhmanid, y P. J. Kellman, “Deep convolutional networks do not classify based on global object shape”, 2018, doi: 10.1371/journal.pcbi.1006613.

[65] V. Dumoulin y F. Visin, “A guide to convolution arithmetic for deep learning”, mar. 2016, [En línea]. Disponible en: http://arxiv.org/abs/1603.07285

[66] “Max Pooling”, Pooling Operations. Accedido: 21 de octubre de 2023. [En línea]. Disponible en: https://paperswithcode.com/method/max-pooling

[67] B. Zhang, Y. Shi, L. Hou, Z. Yin, y C. Chai, “Tsmg: A deep learning framework for recognizing human learning style using eeg signals”, *Brain Sci*, vol. 11, nº 11, nov. 2021, doi: 10.3390/brainsci11111397.

[68] J. Huber, “Batch normalization in 3 levels of understanding”, Towards Data Science. Accedido: 4 de noviembre de 2023. [En línea]. Disponible en: https://towardsdatascience.com/batch-normalization-in-3-levels-of-understanding-14c2da90a338

[69] J. Jordan, “Setting the learning rate of your neural network.”, DATA SCIENCE. Accedido: 21 de octubre de 2023. [En línea]. Disponible en: https://www.jeremyjordan.me/nn-learning-rate/

ANEXO A. INFORMACIÓN ADICIONAL

A.1 Formato de Anotaciones COCO

```
{
    "categories": [{
        "id": "category_id",
        "name": "category_name",
        "supercategory": "supercategory_name",
        "keypoints": ["K1", "K2", "K3", "K4", "K5", "K6", "K7", "K8"],
        "skeleton": [[1,2], [1,3], [1,4], [2,3],
                     [2,4], [3,4], [3,5], [3,6],
                     [4,5], [4,6], [5,6], [5,7],
                     [5,8], [6,7], [6,8], [7,8]]
    }]
,

    "images": [{
        "id": 1,
        "width": 1920,
        "height": 1080,
        "file_name": "image_name",
        "license": 1,
        "date_captured": "2023-01-01"
    }]
,
    "annotation": [{
        "id": "annotation_id",
        "image_id": "image_id",
        "category_id": "category_id",
        "segmentation": [polygon],
        "area": [area],
        "bbox": [x, y, width, length],
        "keypoints": [[x1, y1, 2], [x2, y2, 2], …, [x8, y8, 2]],
        "iscrowd": [0 or 1]
    }]
}
```

A.2 Formato de Anotaciones YOLO

```
1  <class_id> <x_center> <y_center> <width> <height> <x1> <y1> <x2> <y2> ... <xn> <yn>
```

A.3 Ejemplos de Herramientas de Etiquetado

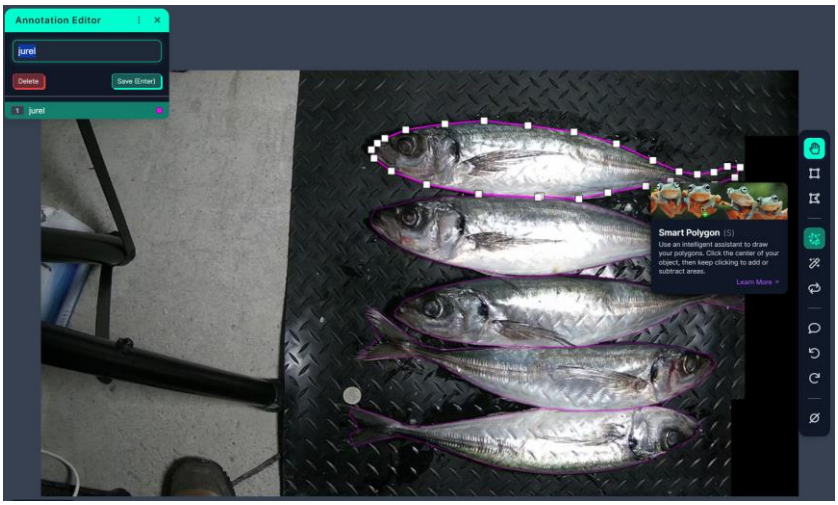


Figura Anexo A.1 Ejemplo de etiquetado de imágenes utilizando la plataforma Roboflow. Fuente: [Elaboración propia]

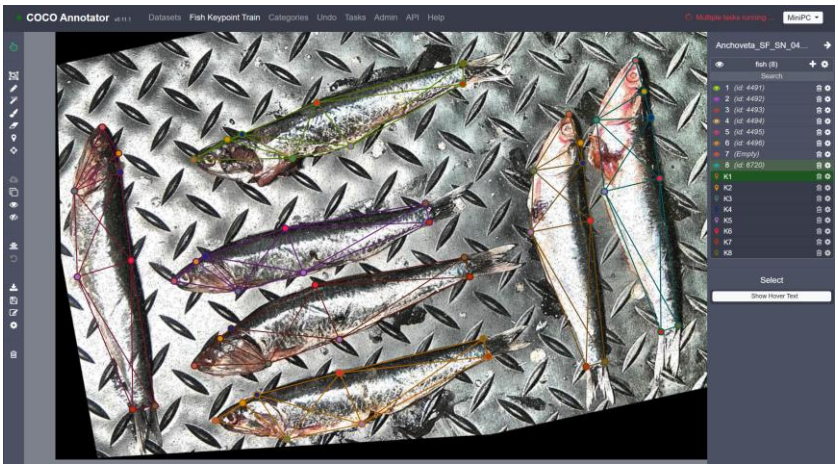


Figura Anexo A.2 Ejemplo de etiquetado de imágenes utilizando la plataforma COCO Annotator. Fuente: [Elaboración propia]

A.4 Capas Comunes de una Red Neuronal

A.4.1 Conv2D:

Capa que realiza operaciones de convolución 2-D con kernels de tamaño fijo sobre su entrada, desplazándose un cierto número de strides (o pasos), hasta la convolución siguiente.

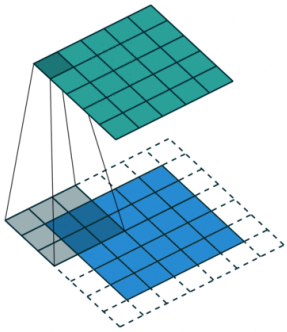


Figura Anexo A.3 Ejemplo de convolución 2D (verde) con un kernel de 3x3 (gris) sobre una entrada de 5x5 (azul) utilizando un stride de 2x2. Fuente: [65]

A.4.2 Max Pooling:

Capa que desliza un kernel de tamaño fijo sobre sus datos de entrada, generando únicamente como salida el mayor valor del bloque o pool cubierto. (Realizando así un submuestreo de la entrada).

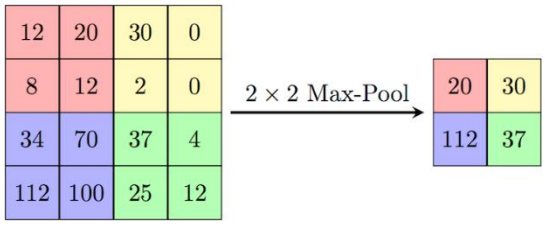


Figura Anexo A.4 Ejemplo de operación de una capa max pooling de 2x2. Fuente: [66]

A.4.3 Global Average Pooling:

Capa que se adapta al tamaño de su entrada, y calcula el promedio del bloque o pool cubierto. Suele reemplazar a las capas de flatten y fully connected.

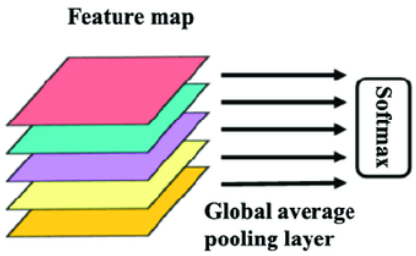


Figura Anexo A.5 Ejemplo de operación de una capa global average pooling. Fuente: [67]

A.4.4 Flatten:

Capa que toma sus datos de entrada y los “estira” de manera tal que se genera un vector unidimensional.

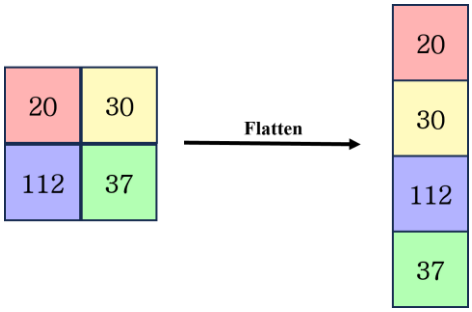


Figura Anexo A.6 Ejemplo de operación de una capa flatten. Fuente: [Elaboración propia]

A.4.5 Dropout:

Capa que previene el sobreajuste durante el entrenamiento "desactivando" aleatoriamente un porcentaje previamente determinado de sus neuronas durante cada iteración del entrenamiento.

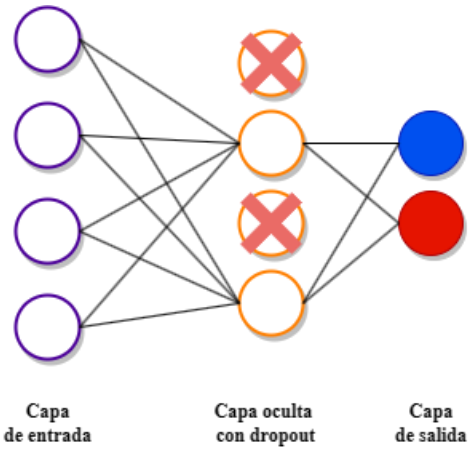


Figura Anexo A.7 Ejemplo de operación de una capa dropout. Fuente: [Elaboración propia]

A.4.6 Batch Normalization:

Capa que ayuda a que la red se entrene de manera más eficiente y rápida. Funciona ajustando los valores de salida de una capa en la red de manera que tengan una media cercana a cero y una desviación estándar cercana a uno. Esto ayuda a evitar problemas de explosión o desvanecimiento de gradientes durante el entrenamiento.

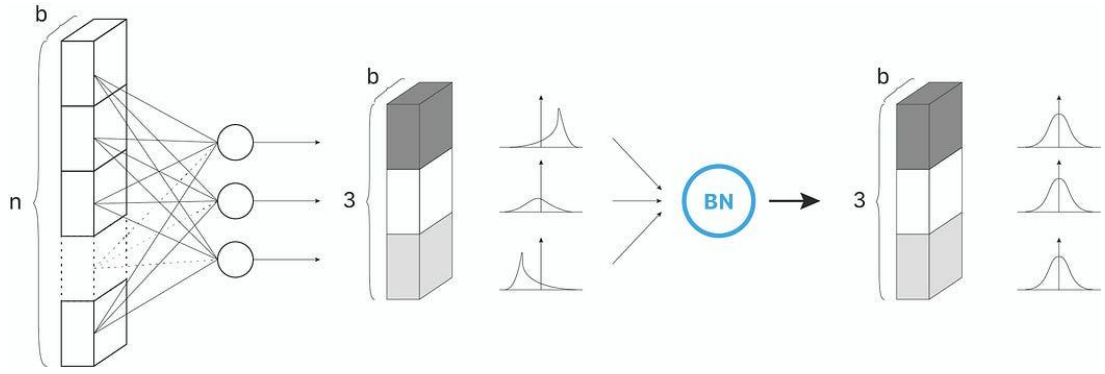


Figura Anexo A.8 Ejemplo de operación de una capa batch normalization para una capa oculta de 3 neuronas, con un batch de tamaño b. Cada neurona sigue una distribución normal estándar al terminar el proceso. Fuente: [68]

A.5 Técnicas de Regularización

A.5.1 Data Augmentation

Data augmentation o aumento de datos es una técnica que no incide directamente sobre los parámetros de la red, pero si aplica transformaciones aleatorias sobre las imágenes de entrada a la red en cada iteración durante la fase de entrenamiento. De esta forma, se garantiza que la red siempre vea imágenes levemente distintas a las originales, evitando el sobreajuste.

Algunos de los métodos aplicados durante la fase de entrenamiento de la mayoría de los modelos que utilizan imágenes como entrada fueron:

- Ajuste de *zoom* +20%
- Ajuste del rango de brillo [0.8, 1.25]
- Volteo horizontal/vertical
- Rango de rotación $\pm 45°$
- Ajuste de ancho +20%
- Ajuste de altura +20%

A.5.2 Early Stopping

Early stopping, o parada temprana, es una técnica utilizada para detener el entrenamiento de la red en función de una condición evaluada sobre alguna métrica de desempeño, como *accuracy*, en caso de que no aumente; o *loss*, en caso de que no disminuya. Si bien esta técnica tampoco incide en algún parámetro de la red, si proporciona las herramientas para no entrenar el modelo más tiempo de lo necesario.

A.5.3 Tasa de Aprendizaje

La tasa de aprendizaje (*learning rate*) es un parámetro que escala la magnitud completa de todos los pesos de la red y busca minimizar su función de pérdida. Si bien no es estrictamente una técnica de regularización, dependiendo de su valor, esta puede incidir

negativamente en el aprendizaje de la red, haciéndolo muy lento (*learning rate* demasiado bajo), o muy impredecible (*learning rate* demasiado alto).

Los métodos más comunes para escoger un *learning rate* apropiado se enumeran a continuación:

i. Decidir entre una tasa de aprendizaje que no sea ni demasiado baja ni demasiado alta, es decir, encontrar el mejor escenario.
ii. Ajustar la tasa de aprendizaje durante el entrenamiento de alta a baja para reducir la velocidad del ajuste de pesos una vez que la función de pérdida se acerque a una solución óptima.
iii. Oscilar entre tasas de aprendizaje altas y bajas para crear una función de planificación (*scheduler*).

Siguiendo lo propuesto en el método i, una forma sistemática para encontrar el rango óptimo para el *learning rate* se muestra en la Figura Anexo A.9. Según se menciona en [69], el método consiste simplemente en aumentar gradualmente la tasa de aprendizaje durante el entrenamiento del modelo, registrando la pérdida en cada incremento. Para las tasas de aprendizaje que son demasiado bajas, la pérdida puede disminuir, pero a un ritmo muy bajo. Para las tasas de aprendizaje que son demasiado altas, las actualizaciones de los pesos recibirán cambios muy extremos, aumentando la pérdida de forma oscilatoria, volviéndose inestable. Sin embargo, un punto intermedio entre ambos casos extremos ocurre al entrar en la zona de tasa de aprendizaje óptima, donde es posible observar una rápida caída en la función de pérdida en forma de pendiente.

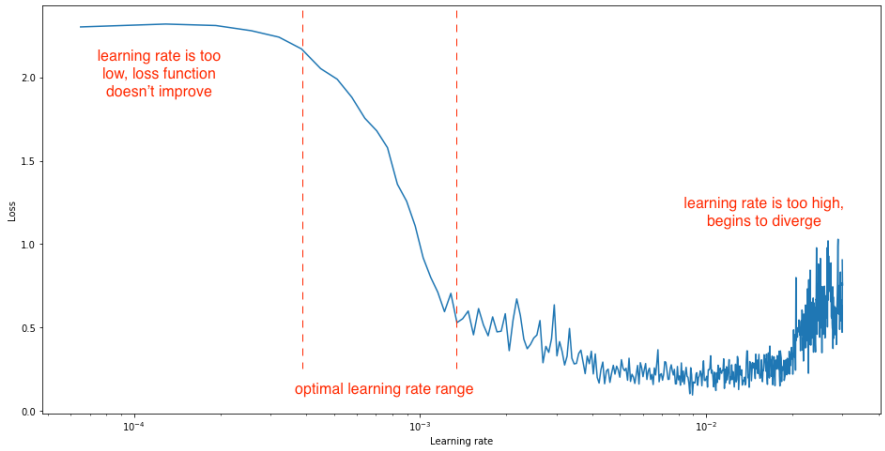
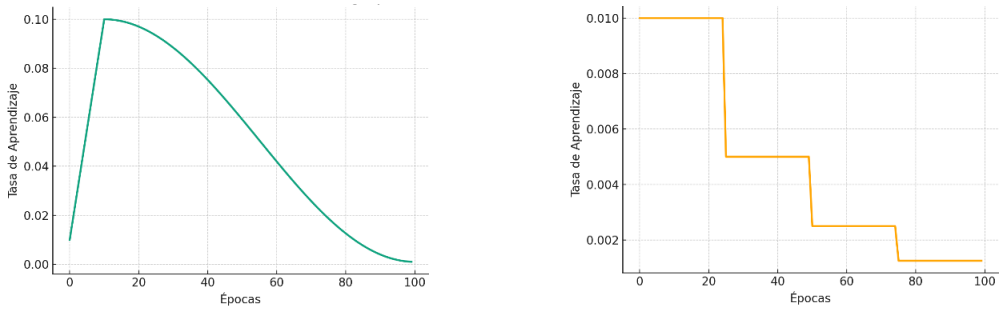


Figura Anexo A.9 Variación típica de la curva de pérdida en función del learning rate ajustado para un modelo "x". Fuente: [Elaboración Propia]

Respecto del método iii, los enfoques más utilizados mediante un *scheduler* son:

- **Decaimiento coseno con calentamiento**: Tal y como se muestra en el apartado a de la Figura Anexo A.10 este enfoque combina una fase inicial de "calentamiento" (*warming up*) donde la tasa de aprendizaje aumenta gradualmente desde un valor pequeño hasta alcanzar un valor máximo. Luego, en la fase de decaimiento coseno, la tasa de aprendizaje disminuye siguiendo la forma de una función coseno desde el máximo hasta un valor mínimo. Esto permite un ajuste por etapas ideal del aprendizaje, creciendo gradualmente al inicio para prevenir cambios bruscos en los pesos del modelo, facilitando su convergencia, y luego disminuyendo de forma gradual y suave para realizar un ajuste fino de los pesos en etapas avanzadas del entrenamiento.
- **Escalonado**: Tal y como se muestra en el apartado b de la Figura Anexo A.10Figura Anexo A.10 en este enfoque la tasa de aprendizaje disminuye en pasos predefinidos durante el entrenamiento. Primero, el entrenamiento se inicia con una tasa de aprendizaje alta y, después de un número específico de épocas o pasos, la tasa disminuye a un valor menor. Este proceso se repite varias veces, siendo útil para adaptar el aprendizaje a diferentes fases del entrenamiento, permitiendo rápidos avances al principio y ajustes más detallados a medida que avanza el entrenamiento.



(a) (b)

Figura Anexo A.10 Schedulers para la variación del learning rate. a) Decaimiento coseno con calentamiento. b) Escalonado. Fuente: [Elaboración Propia]

A.6 Preprocesamiento de Imágenes

A.6.1 Blurring

El principal problema de las imágenes obtenidas en entorno industrial es el solapamiento de las especies, lo que impide etiquetar completamente el cuerpo de la mayoría de los peces presentes en las imágenes. Este solapamiento representa un problema para el entrenamiento de los modelos, puesto que las imágenes parcialmente etiquetadas pueden potencialmente causar una disminución en las métricas de validación, lo cual se debe a que el modelo tiende a identificar más peces de los que fueron realmente etiquetados. Por este motivo, se utilizó la técnica que se muestra en la Figura Anexo A.11, la cual consiste en aplicar un blurring (difuminado) a las secciones de la imagen que no se pudieron etiquetar manualmente, convirtiéndolas en parte del fondo. De esta manera, el modelo puede interpretar estas áreas como ruido, lo que ayuda a mejorar su rendimiento en la detección y clasificación de peces.

Figura Anexo A.11 Blurring aplicado en el área no etiquetada de la imagen. Fuente: [Elaboración Propia]

A.6.2 Reemplazo del fondo con imágenes aleatorias

A.6.2.1. Industria

Como alternativa al difuminado del fondo, también se probó un enfoque que consiste en un algoritmo capaz de reemplazar aleatoriamente el fondo de la imagen original que no se etiquetó completamente, por una imagen de una correa vacía. Esta técnica busca mejorar la capacidad del modelo para distinguir entre peces y el fondo, ofreciendo un contraste más claro y reduciendo el riesgo de falsos positivos en la detección, sirviendo como una versión intermedia de los fondos *random* utilizados en la sección siguiente para el mismo propósito. Un ejemplo de la aplicación de esta técnica se muestra en la Figura Anexo A.12.

Figura Anexo A.12 Ejemplo de imagen etiquetada en la que el fondo se cambió por la imagen de una correa vacía. Fuente: [Elaboración Propia]

A.6.2.2. Laboratorio

De manera complementaria para las bases de datos de peces **B03** y **B04**, fue necesario preparar imágenes aleatorias para utilizarlas como fondos en algunas de las pruebas realizadas con los modelos, particularmente, los de la Sección 3.3.2 y la Sección 3.4. Estas imágenes fueron extraídas directamente desde internet, alcanzando un total de 1680 fondos divididos entre: playa, ciudad, desierto, tierra, bosque, jardín, iceberg, Japón, paisajes, montañas, naturaleza, paraíso, lluvia, mar, nieve, estrellas, pantanos, universo, volcanes y cascadas. La Figura N° 2.2 muestra un ejemplo con algunos de los fondos utilizados.

Figura Anexo A.13 Ejemplos de fondos random. Fuente: [Desconocida]

A.6.3 Filtros de extracción de texturas

Un filtro de extracción de texturas es una herramienta utilizada en el procesamiento de imágenes y visión computacional para identificar y aislar regiones de una imagen que tienen una textura específica. La textura en una imagen se refiere a las variaciones espaciales en la intensidad de los píxeles, y puede describirse mediante patrones, repeticiones, rugosidad, suavidad, etc. La extracción de texturas es fundamental en diversas aplicaciones, particularmente, para el minimodelo de clasificación de manchas introducido en la Sección 3.4.5.

A continuación, se describen algunos de los filtros de extracción de texturas utilizadas a lo largo de esta tesis.

A.6.3.1. Matriz de co-ocurrencia de nivel de gris (GLCM)

Es una técnica estadística que considera la relación espacial entre píxeles. La GLCM calcula cómo se distribuyen las combinaciones de pares de píxeles, considerando su intensidad, la distancia y el ángulo entre ellos, proporcionando medidas de textura como:

- *Angular second moment* (ASM)
- *Contrast*
- *Correlation*
- *Variance*
- *Inverse difference moment* (IDM)
- *Sum average*
- *Sum variance*
- *Sum entropy*
- *Entropy*
- *Difference variance*
- *Difference entropy*
- *Information measure of correlation 1*
- *Information measure of correlation 2*
- *Maximal correlation coefficient*

A.6.3.2. Histograma de gradientes orientados (HOG)

Es una técnica de visión por computadora utilizada para la extracción de características, que implica la evaluación de gradientes orientados en porciones localizadas de

una imagen para identificar formas y estructuras. El número de características de este algoritmo depende proporcionalmente de las dimensiones de la región o imagen a partir de la que se calcula el HOG, escalando rápidamente.

A.6.3.3. Transformada discreta del coseno (DCT)

Es una técnica matemática que convierte señales espaciales en frecuencia, siendo ampliamente utilizada en la compresión de imágenes, como en el formato JPEG, al preservar las características más importantes de la imagen mientras reduce la dimensión de los datos.

A.6.3.4. Patrones binarios locales (LBP)

Son un método de descripción de texturas que etiqueta los píxeles de una imagen al umbralizar la vecindad de cada píxel y considerando el resultado como un número binario, siendo eficaz para la clasificación de texturas en imágenes.

A.6.4 Filtros para la Segmentación de Manchas

A.6.4.1. Filtros de Gabor

El filtro de Gabor es un filtro lineal cuya respuesta de impulso es una función sinusoidal multiplicada por una función gaussiana. La principal ventaja que se obtiene al introducir la envolvente gaussiana es que las funciones de Gabor están localizadas tanto en el dominio espacial como en el de la frecuencia, por lo que son ampliamente utilizados para la extracción de texturas y resaltar características locales de una imagen debido a su capacidad para capturar patrones que poseen orientaciones y escalas distintas.

A.6.4.2. Umbralización de Otsu

Es un método de binarización de imágenes que determina de manera automática el umbral óptimo al minimizar la varianza intra-clase de los píxeles entre su fondo y un objeto en concreto, siendo especialmente útil cuando la imagen tiene dos modas prominentes.

A.6.4.3. Filtro de Sobel

Es un método rápido utilizado para la detección de bordes en imágenes, aplicando convoluciones con kernels específicos que calculan las derivadas aproximadas de la imagen en las direcciones x e y, resaltando así los bordes más gruesos y transiciones de la intensidad de los píxeles.

A.7 Métricas de Desempeño

A.7.1 Precision (P)

La métrica P, también conocida como precisión, mide la exactitud de las predicciones positivas realizadas por un modelo de clasificación. Es la proporción de verdaderos positivos (TP) entre la suma de verdaderos positivos y falsos positivos (FP). La ecuación para calcular la métrica P es:

$$P = \frac{TP}{TP + FP}$$

Ecuación *4.1 Métrica P*

A.7.2 Recall (R)

La métrica R, también conocida como sensibilidad, mide la capacidad del modelo para identificar y clasificar correctamente los casos positivos reales. Es la proporción de los verdaderos positivos entre la suma de verdaderos positivos y falsos negativos (FN). La ecuación para calcular la métrica R es:

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

Ecuación *4.2 Métrica R*

A.7.3 F-score (F1)

El F-score, F1-score, o métrica F1, es una métrica que combina tanto la métrica P como la métrica R en un solo número, proporcionando una medida de la calidad y la completitud de las predicciones positivas de un modelo de clasificación. La ecuación para calcular la métrica F1 es:

$$F1 = 2 \times \frac{P \times R}{P + R}$$

Ecuación *4.3 Métrica F1*

A.7.4 Mean-Average Precision (mAP)

Mean-average precision es una métrica de desempeño que se utiliza habitualmente en las tareas de detección de objetos. Esta mide la media aritmética de la métrica *Average Precision* (AP) entre varias clases y para diferentes niveles de umbrales de confianza de detección, siguiendo la fórmula:

$$mAP = \frac{1}{n} \sum_{i=1}^{n} AP_i$$

Ecuación *4.4 Métrica mAP*

Donde n es el número total de clases y AP_i es la métrica AP de la clase "i". Por su parte, la métrica AP resume el área bajo la curva *precision-recall*, la cual se construye evaluando la métrica P para diferentes valores de la métrica R, proporcionando una puntuación de rendimiento general del modelo para cada clase respectiva.

A.7.5 Macro-average Precision (MP)

La *macro-average precision* es una métrica que calcula la métrica P para cada clase individualmente y luego toma la media de estos valores, tratando todas las clases por igual, independientemente de su distribución en el conjunto de datos. Si se tienen n clases y P_i es la precisión para la clase i, la métrica MP se calcula como:

$$MP = \frac{1}{n}\sum_{i=1}^{n} P_i$$

Ecuación *4.5 Métrica MP*

A.7.6 Mean Absolute Error (MAE)

El Mean Absolute Error (MAE) es una métrica utilizada para medir la precisión de las predicciones de un modelo, típicamente en problemas de regresión. Es la media del valor absoluto de los errores entre las predicciones y los valores reales. Si se tienen un conjunto de n observaciones, donde y_i es el valor real e $\hat{y}_i$ es el valor predicho para la observación i, el MAE se calcula como:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} \left| y_i - \hat{y}_i \right|$$

Ecuación *4.6 Métrica MAE*

Esta métrica es particularmente útil para la detección de *keypoints*, puesto que proporciona una medida directa de la magnitud promedio del error entre un *keypoint* real y un *keypoint* predicho por el modelo. De esta forma, a partir de la **¡Error! No se encuentra el origen de la referencia.**, el MAE para el *keypoint* "K_j" se obtiene mediante:

$$MAE_{k_j} = \frac{1}{n}\sum_{i=1}^{n} \left| K_{j,i} - \widehat{K}_{j,i} \right|$$

Ecuación *4.7 Métrica MAE para keypoints*

Donde n pasa a ser el total de peces detectados, $K_{j,i}$ es el *keypoint* K_j real y $\widehat{K}_{j,i}$ es el *keypoint* K_j predicho para la observación i.

A.7.7 Percentage of Correct Keypoints (PCK)

El porcentaje de *keypoints* correctos (PCK) es una métrica comúnmente utilizada en la evaluación de las articulaciones del esqueleto o en los mismos *keypoints* generados en algoritmos de detección de *keypoints*. Respecto de los *keypoints*, esta métrica mide la precisión de la detección de *keypoints* al comparar la distancia entre los *keypoints* predichos y los reales con un umbral predefinido. Si esta distancia es menor que un porcentaje

especificado (como el tamaño del objeto o alguna otra medida de referencia), entonces se considera una detección correcta.

La fórmula para calcular el PCK para un conjunto de n *keypoints* es:

$$PCK = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(d_i < \alpha D)$$

Ecuación *4.8 Métrica PCK*

Donde:

- **1** es la función indicatriz, que vale 1 si la condición es verdadera y 0 si es falsa.
- d_i es la distancia entre el *keypoint* predicho y el real para el *keypoint* K_i.
- α es el porcentaje del tamaño del objeto utilizado como umbral.
- D es una medida de escala del objeto, como su tamaño o un valor predefinido.

Por simplicidad, para las pruebas realizadas con los modelos de *keypoints*, el umbral de desviación, αD, se definió como una constante, en píxeles, escogida experimentalmente para cada caso.

ANEXO B. RESULTADOS COMPLEMENTARIOS

B.1 Resultados con Clasificadores del Estado del Arte

B.1.1 Variantes de 4 Clases

*Tabla Anexo B.1 Resumen de las métricas de desempeño obtenidas para la validación del modelo de clasificación multiclase utilizando clasificadores del estado del arte y considerando las bases de datos **B03** y **B04**.*

	Modelo clasificación multiclase – morfometría + texturas ▲GOS – Decision Tree			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**0.97**	0.92	0.94	0.87
Caballa	0.67	**0.97**	0.79	
Jurel	0.91	0.64	0.75	
Sardina	0.93	0.94	0.94	
	Modelo clasificación multiclase – morfometría + texturas ▲GOS - SVM			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.59	0.74	0.72
Caballa	0.52	**0.97**	0.67	
Jurel	0.67	0.26	0.37	
Sardina	0.69	0.93	0.79	
	Modelo clasificación multiclase – morfometría + texturas ▲GOS - KNN			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	0.74	0.63	0.68	0.62
Caballa	0.44	**0.80**	0.57	
Jurel	0.54	0.43	0.48	
Sardina	**0.75**	0.65	0.70	

*Tabla Anexo B.2 Resumen de las métricas de desempeño obtenidas para la validación del modelo de clasificación jerárquico plano, utilizando clasificadores del estado del arte y considerando las bases de datos **B03** y **B04**.*

	Modelo jerárquico plano – morfometría + texturas ▲GOS – *Decision Tree*			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.94	0.97	0.91
Caballa	0.71	0.94	0.81	
Jurel	0.97	0.83	0.90	
Sardina	0.96	**0.96**	0.96	
	Modelo jerárquico plano – morfometría + texturas ▲GOS – *SVM*			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.95	0.98	0.92
Caballa	0.79	0.91	0.85	
Jurel	0.98	0.85	0.91	
Sardina	0.92	**0.97**	0.94	
	Modelo jerárquico plano – morfometría + texturas ▲GOS – *KNN*			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.92	0.96	0.92
Caballa	0.79	0.91	0.85	
Jurel	0.98	0.85	0.91	
Sardina	0.89	**0.97**	0.93	

B.1.2 Variantes de 2 Clases

*Tabla Anexo B.3 Resumen de las métricas de desempeño obtenidas para la validación del modelo de clasificación jerárquico plano para especies pequeñas, utilizando clasificadores del estado del arte y considerando las bases de datos **B03** y **B04**.*

	Modelo jerárquico plano – morfometría + texturas ▲GOS – *Decision Tree*			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.97	0.98	0.99
Sardina	0.97	**1.00**	0.99	
	Modelo jerárquico plano – morfometría + texturas ▲GOS – *SVM*			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.95	0.98	0.98
Sardina	0.96	**1.00**	0.98	
	Modelo jerárquico plano – morfometría + texturas ▲GOS – *KNN*			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.92	0.96	0.97
Sardina	0.93	**1.00**	0.97	

*Tabla Anexo B.4 Resumen de las métricas de desempeño obtenidas para la validación del modelo de clasificación jerárquico plano para especies grandes, utilizando clasificadores del estado del arte y considerando las bases de datos **B03** y **B04**.*

	Modelo jerárquico plano – morfometría + texturas ▲GOS – *Decision Tree*			
Clase	**precision**	**recall**	**F1**	**MP**
Caballa	0.78	**0.91**	0.84	0.85
Jurel	**0.93**	0.81	0.86	
	Modelo jerárquico plano – morfometría + texturas ▲GOS – *SVM*			
Clase	**precision**	**recall**	**F1**	**MP**
Caballa	0.82	**0.97**	0.89	0.90
Jurel	**0.98**	0.85	0.91	
	Modelo jerárquico plano – morfometría + texturas ▲GOS – *KNN*			
Clase	**precision**	**recall**	**F1**	**MP**
Caballa	0.84	**0.94**	0.89	0.90
Jurel	**0.95**	0.87	0.91	

*Tabla Anexo B.5 Resumen de las métricas de desempeño obtenidas para la validación del modelo de clasificación jerárquico plano para especies combinadas, utilizando clasificadores del estado del arte y considerando las bases de datos **B03** y **B04**.*

	Modelo jerárquico plano – morfometría + texturas ▲GOS – *Decision Tree*			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**0.98**	**0.98**	0.98	0.98
Jurel	**0.98**	**0.98**	0.98	
	Modelo jerárquico plano – morfometría + texturas ▲GOS – *SVM*			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	0.98	**1.00**	0.99	0.99
Jurel	**1.00**	0.98	0.99	
	Modelo jerárquico plano – morfometría + texturas ▲GOS – *KNN*			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	0.98	**1.00**	0.99	0.99
Jurel	**1.00**	0.98	0.99	

B.2 Resultados con CNN y FCNN Utilizando Imágenes de Manchas GOS

*Tabla Anexo B.6 Resumen de las métricas de desempeño obtenidas para la validación de múltiples clasificadores para especies pequeñas, utilizando clasificadores con CNN y FCNN y considerando las bases de datos **B03** y **B04**.*

	Modelo clasificación multiclase VGG16 – imágenes ■ RGB			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	0.80	**0.89**	0.84	0.84
Sardina	**0.89**	0.80	0.84	
	Modelo jerárquico plano – morfometría + imágenes ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	**1.00**	1.00	1.00
Sardina	**1.00**	**1.00**	1.00	
	Modelo jerárquico Top/Down – morfometría + imágenes ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.98	0.99	0.99
Sardina	0.99	**1.00**	0.99	
	Modelo jerárquico multidimensional – morfometría + imágenes ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.98	0.99	0.99
Sardina	0.99	**1.00**	0.99	

*Tabla Anexo B.7 Resumen de las métricas de desempeño obtenidas para la validación de múltiples clasificadores para especies grandes, utilizando clasificadores con CNN y FCNN y considerando las bases de datos **B03** y **B04**.*

	Modelo clasificación multiclase VGG16 – imágenes ■ RGB			
Clase	**precision**	**recall**	**F1**	**MP**
Caballa	**0.94**	0.88	0.91	0.93
Jurel	0.92	**0.96**	0.94	
	Modelo jerárquico plano – morfometría + imágenes ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Caballa	0.92	**0.97**	0.94	0.95
Jurel	**0.98**	0.94	0.96	

	Modelo jerárquico Top/Down – morfometría + imágenes ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Caballa	0.85	**0.90**	0.88	0.89
Jurel	**0.93**	0.89	0.91	
	Modelo jerárquico multidimensional – morfometría + imágenes ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Caballa	0.85	**0.90**	0.88	0.89
Jurel	**0.93**	0.89	0.91	

*Tabla Anexo B.8 Resumen de las métricas de desempeño obtenidas para la validación de múltiples clasificadores para especies combinadas, utilizando clasificadores con CNN y FCNN y considerando las bases de datos **B03** y **B04**.*

	Modelo clasificación multiclase VGG16 – imágenes ■ RGB			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	0.95	**1.00**	0.98	0.98
Jurel	**1.00**	0.94	0.97	
	Modelo jerárquico plano – morfometría + imágenes ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	0.98	**1.00**	0.99	0.99
Jurel	**1.00**	0.98	0.99	
	Modelo jerárquico Top/Down – morfometría + imágenes ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.98	0.99	0.99
Jurel	0.98	**1.00**	0.99	
	Modelo jerárquico multidimensional – morfometría + imágenes ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.98	0.99	0.99
Jurel	0.98	**1.00**	0.99	

B.3 Resultados con FCNN Utilizando Texturas a partir de Imágenes de Manchas GOS

*Tabla Anexo B.9 Resumen de las métricas de desempeño obtenidas para la validación de múltiples clasificadores para especies pequeñas, utilizando clasificadores con FCNN y considerando las bases de datos **B03** y **B04**.*

	Modelo clasificación multiclase FCNN – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	0.98	**1.00**	0.99	0.99
Sardina	**1.00**	0.99	0.99	
	Modelo jerárquico plano – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	**1.00**	1.00	1.00
Sardina	**1.00**	**1.00**	1.00	
	Modelo jerárquico Top/Down – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.98	0.99	0.99
Sardina	0.99	**1.00**	0.99	
	Modelo jerárquico multidimensional – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.98	0.99	0.99
Sardina	0.99	**1.00**	0.99	

*Tabla Anexo B.10 Resumen de las métricas de desempeño obtenidas para la validación de múltiples clasificadores para especies grandes, utilizando clasificadores con FCNN y considerando las bases de datos **B03** y **B04**.*

	Modelo clasificación multiclase FCNN – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Caballa	0.73	**1.00**	0.84	0.86
Jurel	**1.00**	0.72	0.84	
	Modelo jerárquico plano – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Caballa	**1.00**	0.94	0.97	0.98
Jurel	0.96	**1.00**	0.98	

	Modelo jerárquico Top/Down – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Caballa	**0.89**	0.76	0.82	0.85
Jurel	0.80	**0.92**	0.86	
	Modelo jerárquico multidimensional – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Caballa	**0.89**	0.76	0.82	0.85
Jurel	0.80	**0.92**	0.86	

*Tabla Anexo B.11 Resumen de las métricas de desempeño obtenidas para la validación de múltiples clasificadores para especies combinadas, utilizando clasificadores con FCNN y considerando las bases de datos **B03** y **B04**.*

	Modelo clasificación multiclase FCNN – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	0.95	**0.98**	0.97	0.97
Jurel	**0.98**	0.94	0.96	
	Modelo jerárquico plano – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	**1.00**	1.00	1.00
Jurel	**1.00**	**1.00**	1.00	
	Modelo jerárquico Top/Down – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.98	0.99	0.99
Jurel	0.97	**1.00**	0.99	
	Modelo jerárquico multidimensional – morfometría + texturas ▲GOS			
Clase	**precision**	**recall**	**F1**	**MP**
Anchoveta	**1.00**	0.98	0.99	0.99
Jurel	0.97	**1.00**	0.99	

ANEXO C. DIAGRAMAS DE MODELOS 3D

Figura Anexo C.1 Esquema 3D del árbol de clasificación jerárquica utilizando el minimodelo de manchas a partir de imágenes con ▲GOS. Fuente: [Elaboración Propia]

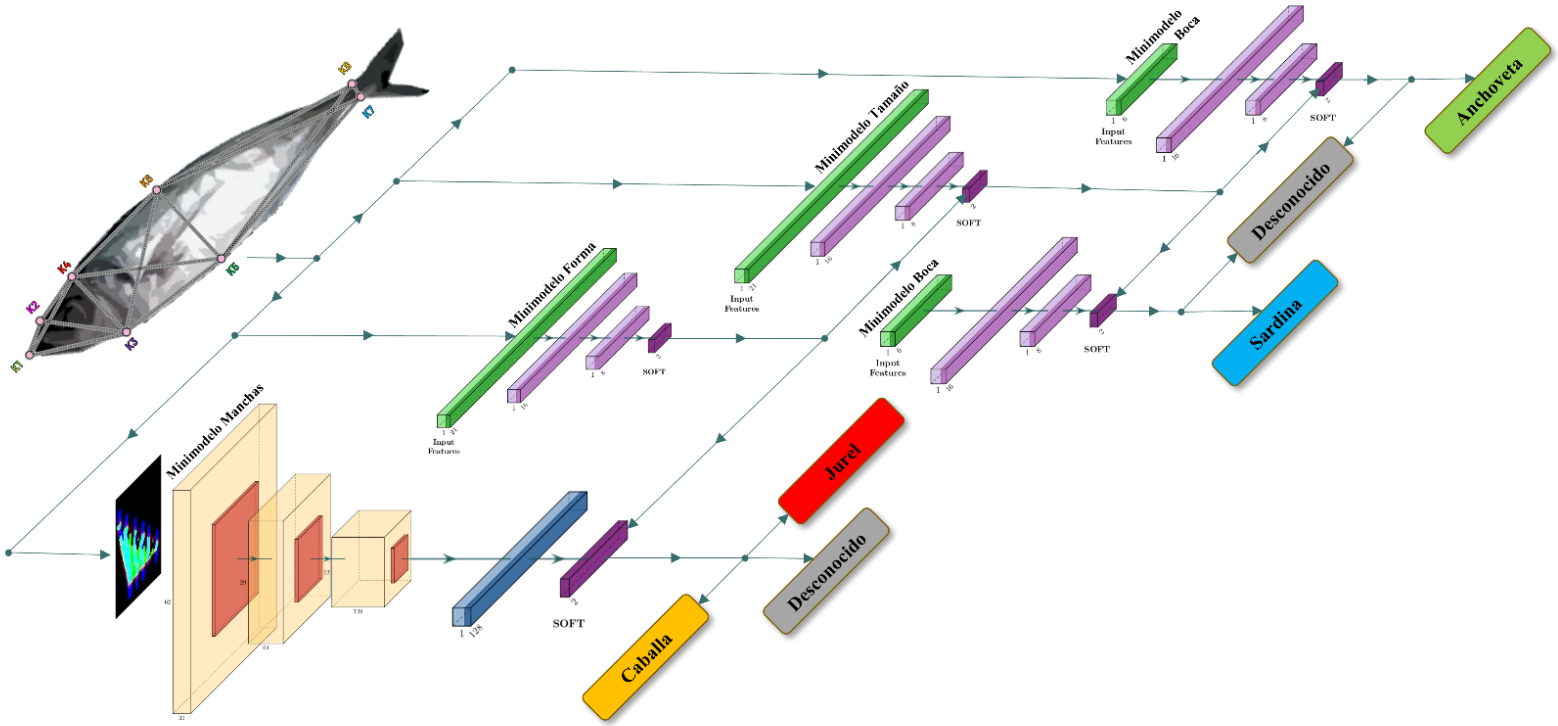


Figura Anexo C.2 Esquema 3D del árbol de clasificación jerárquica utilizando el minimodelo de manchas a partir de las texturas de imágenes con ▲GOS. Fuente: [Elaboración Propia]