



**Conocimiento y transferencia de
tecnología sobre vehículos aéreos y
acuáticos para el desarrollo transfronterizo
de ciencias marinas y pesqueras
(POCTEP 0622-KTTSEADRONES-5-E)**

Informe técnico

**Actividad 2. Desarrollo de sensores y herramientas
informáticas para vehículos aéreos y marinos**

**Informe sobre el desarrollo de sensores en sistemas fijos
(boyas) y ROVs para su uso en estuarios y explotaciones
piscícolas**

Producto final – Acción 2.3A2 | 30/06/2021



Nombre del Proyecto: KTTSeaDrones - Conocimiento y transferencia de tecnología sobre vehículos aéreos y acuáticos para el desarrollo transfronterizo de ciencias marinas y pesqueras

Código del proyecto: POCTEP 0622-KTTSEADRONES-5-E

Datos del proyecto

Localización	España y Portugal		
Fecha de inicio	01/10/2017		
Fecha de finalización	31/12/2021	Prórroga hasta	05/04/2022
Financiación total	€ 1.120.864,43		
Contribución UE	€ 840.648,32		
Cofinanciación UE (%)	75%		

Datos del beneficiario principal (BP)

Beneficiario principal	Universidad de Huelva
Persona de contacto	Vicerrector de Investigación y Transferencia
Dirección	C/ Cantero Cuadrado 6, 21004 Huelva
Teléfono	+34 959 218013
E-mail	kttseadrones@gmail.com
Página WEB (proyecto)	https://kttseadrones.wixsite.com/kttseadrones

Datos del documento

Nombre	Informe sobre el desarrollo de sensores en sistemas fijos (boyas) y ROVs para su uso en estuarios y explotaciones piscícolas
Acción	Acción 2.3A2. Desarrollo de un sistema fijo (boya) y un vehículo operado remotamente (ROV) con la capacidad de recoger, transmitir y almacenar información sobre las condiciones físico-químicas en estuarios y en balsas de engorde en instalaciones de producción acuícola
Autores	Samuel López Domínguez (UHU) Antonio Peregrín Rubio (UHU) Juan Carlos Gutiérrez Estrada (UHU) Inmaculada Pulido Calvo (UHU) Jairo Castro Gutiérrez (UHU) Alejandro Garrocho Cruz (UHU) Fernando Gómez Bravo (UHU)
Equipo técnico	Juan Antonio Gómez Galán (UHU) Raúl Jiménez Naharro (UHU) Rafael López de Ahumada (UHU) Manuel Sánchez Raya (UHU) Manuel Pedro Carrasco (UHU)

Javier Aroba Páez (UHU)
Francisco José Moreno Velo (UHU)
José Manuel Martín Ramos (UHU)
Ana María Roldán Ruiz (UHU)
Francisco Alfredo Márquez Hernández (UHU)
Antonio Ángel Márquez Hernández (UHU)
Ignacio de la Rosa Lucas (UHU)

Fecha 30/06/2021

Versión (nº) 1.0

Las opiniones presentadas en este informe comprometen únicamente a los beneficiarios del proyecto y, en consecuencia, no representan en ningún caso la opinión oficial de los órganos de gestión del POCTEP.

ÍNDICE

1. INTRODUCCIÓN	5
1.1. DEEP LEARNING.....	6
1.2. SEGMENTACIÓN DE IMÁGENES.....	8
2. MODELO PROPUESTO	12
2.1. ARQUITECTURA	12
2.2. CÁLCULO DE LA BIOMASA. PRIMERA APROXIMACIÓN.....	14
3. PRUEBAS DEL MODELO	16
3.1. CONFIGURACIÓN DE LAS PRUEBAS.....	16
3.2. MÉTRICAS DE EVALUACIÓN	18
3.3. AJUSTE DEL MODELO	18
3.4 RESULTADOS OBTENIDOS Y ANÁLISIS.....	19
4. CONCLUSIONES	24
5. REFERENCIAS	25

ÍNDICE DE FIGURAS

Fig. 1 Esquema de una DNN.	6
Fig. 2 Perceptrón recurrente vs perceptrón con sin realimentación.	7
Fig. 3 Esquema de una CNN.	7
Fig. 4 Ejemplo de segmentación semántica extraído del dataset Cityscapes.....	8
Fig. 5 Ejemplo de segmentación de instancias.....	9
Fig. 6 Generación de máscaras seguida de clasificación.	9
Fig. 7 Detección seguida de clasificación.	10
Fig. 8 Etiquetado de píxeles seguido de clustering.	10
Fig. 9 Métodos basados en el desplazamiento de ventanas.....	11
Fig. 10 Esquema de un autoencoder.....	12
Fig. 11. Arquitectura de la red propuesta.	13
Fig. 12 Clúster de GPUs usado en este trabajo	16
Fig. 13 Transductor del sonar de imagen LVS32 de Garmin, acoplado a un módulo LiveScope GLS10 (Sistema Panoptix LiveScopeTM).	17
Fig. 14 Imagen de sonar de ultrasonidos.	21
Fig. 15 Etiquetado manual de las imágenes usando el programa labelImg (https://github.com/tzutalin/labelImg).	22
Fig. 16 Etiquetado de las distintas instancias.....	23

ÍNDICE DE TABLAS

Tabla 1. Comparación de las funciones de coste según las distintas métricas de evaluación....	19
---	----

Acción 2.3 Desarrollo de un sistema fijo (boya) y un vehículo operado remotamente (ROV) con la capacidad de recoger, transmitir y almacenar información sobre las condiciones físico-químicas en estuarios y en balsas de engorde en instalaciones de producción acuícola

1. INTRODUCCIÓN

La sobrepesca, es decir, la captura de especies acuáticas en grandes volúmenes, suponen un gran peligro para los ecosistemas marinos pudiendo ser el factor desencadenante de la destrucción de arrecifes o la desaparición de ciertas especies.

Una alternativa sostenible a este tipo de pesca lo encontramos en la acuicultura, que se puede definir como el conjunto de conocimientos y técnicas aplicadas al cultivo de especies acuáticas tanto animales como vegetales. Además, este tipo de actividad puede ser desarrollada en agua dulce, agua salada y/o en tanques de tierra.

A la hora de diseñar un plan de producción óptimo en este tipo de entornos es indispensable conocer la biomasa, la cual puede ser calculada multiplicando el peso medio de los peces por el número total de individuos en la balsa [Haddon, 2010]. Hallar este índice es una operación muy costosa ya que en primer lugar hay que proceder al vaciado total o parcial de la balsa para luego mediante un muestreo manual establecer el peso medio de los peces.

En este trabajo se plantea como solución a este problema la utilización de técnicas de inteligencia artificial, concretamente de *Deep Learning*, para automatizar este proceso a través del análisis de imágenes de sonar de ultrasonidos.

1.1. DEEP LEARNING

El *Deep Learning* es un campo perteneciente a la inteligencia artificial que consiste en un conjunto de transformaciones no lineales cuyo objetivo es obtener representaciones óptimas de los datos con distintos niveles de abstracción para en un etapa posterior proceder a su detección o clasificación [LeCun et al., 2015].

Dependiendo de su estructura y de la aplicación, existen varios tipos de redes neuronales profundas:

- **Dense Neural Networks (DNN).**

Red neuronal artificial con un número de capas ocultas superior a una. En estas redes los datos fluyen desde la capa de entrada hasta la capa de salida sin retroceder y los enlaces entre las capas son unidireccionales, es decir, en dirección hacia adelante y nunca vuelven a hacia un nodo de la misma capa o anterior.

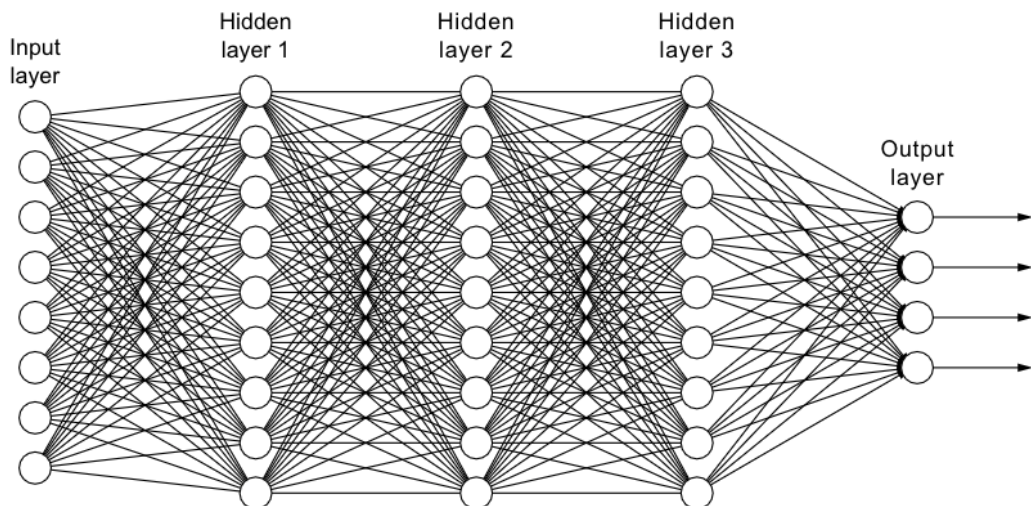


Fig. 1 Esquema de una DNN.

- **Recurrent Neural Networks (RNN).**

Las RNN son tipo de clase de redes para analizar datos de series temporales permitiendo tratar la dimensión de "tiempo". La red está formada por un conjunto de neuronas recurrentes las cuales reciben como entrada la salida de la capa anterior así como si propia salida del instante de tiempo anterior.

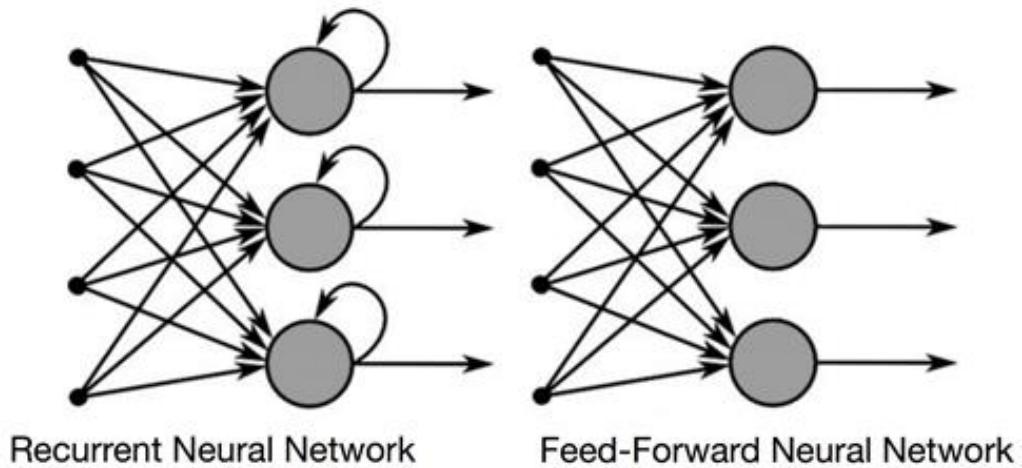


Fig. 2 Perceptrón recurrente vs perceptrón con sin realimentación.

- **Convolutional Neural Networks (CNN).**

Una CNN es un caso concreto de red neuronal cuya entrada es una imagen [LeCun at al., 1998]. El modo de funcionamiento de este tipo de red es similar a como funciona el cortex visual: primero se reconocen los patrones más pequeños, como por ejemplo los bordes; a continuación, a través de una composición de patrones se reconocen patrones más complejos como por ejemplo líneas.

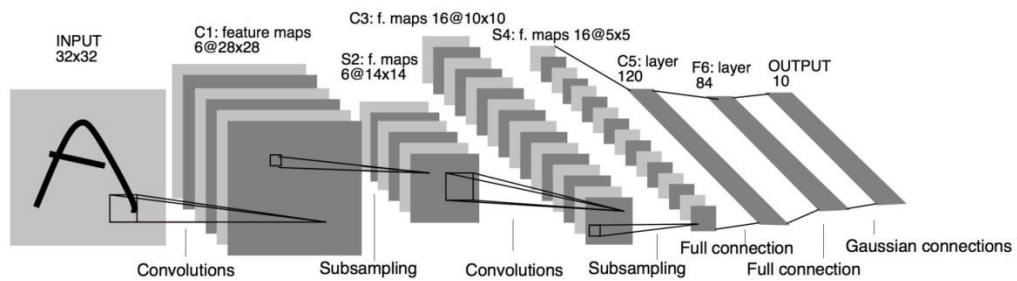


Fig. 3 Esquema de una CNN.

1.2. SEGMENTACIÓN DE IMÁGENES

La segmentación de imágenes es uno de los problemas generales dentro del campo de la visión artificial y que consiste en dividir una imagen en un conjunto de regiones denominados segmentos.

Existen varios tipos de segmentación dependiendo de su objetivo. Las más destacables son las siguientes:

- **Segmentación semántica**

Se conoce como segmentación semántica al proceso de asignar una categoría o etiqueta a cada uno de los píxeles que forman una imagen, de modo que dos píxeles pertenecientes a objetos del mismo tipo deben de tener asignada la misma etiqueta. En otras palabras, la segmentación semántica es la clasificación de una imagen a nivel de píxeles.

Este tipo de segmentación permite conocer a que clase pertenece un píxel, pero no a que objeto en concreto. En otras palabras, si en una imagen existen varios objetos del mismo tipo, todos los píxeles correspondientes a dichos objetos tendrán asignada la misma etiqueta. Un ejemplo de esto lo podemos ver en la Fig. X donde todos los peatones tienen asignada la misma etiqueta.



Fig. 4 Ejemplo de segmentación semántica extraído del dataset Cityscapes.

Este tipo de técnicas son empleadas en una gran variedad de aplicaciones, desde conducción autónoma para identificar la escena hasta aplicaciones médicas para la identificación de tumores.

- **Segmentación de instancias**

La segmentación de instancias es el proceso mediante el cual se busca detectar los objetos en una escena y por cada uno de ellos generar una máscara que permita extraerlo del conjunto de la imagen.

De forma similar a como se hace en la segmentación semántica, por cada píxel se asocia una etiqueta, pero en este caso objetos que pertenecen a la misma clase se tratan como objetos individuales.



Fig. 5 Ejemplo de segmentación de instancias.

Este problema, el de la segmentación de instancias, puede ser abordado desde usando para ello técnicas de *Deep Learning*. Para ello se suele dividir el proceso en dos subprocessos siendo los más habituales [Hafiz y Bhat, 2020]:

1. *Generación de máscaras seguida de clasificación.* En primer lugar, se genera un conjunto de máscaras delimitando los píxeles que corresponden al objeto. Acto seguido, se extrae la región correspondiente al objeto del resto de la imagen y utilizando un clasificador se infiere el tipo de objeto del que se trata. Ejemplos de este grupo serían arquitecturas como R-CNN [Girshick et al., 2014], Fast R-CNN [Girshick, 2014] y Faster R-CNN [Ren et al., 2017].

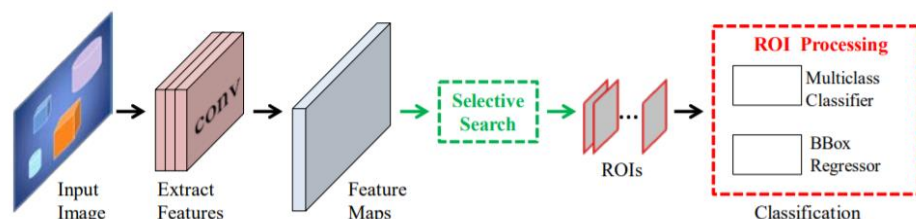


Fig. 6 Generación de máscaras seguida de clasificación.

2. *Detección seguida de segmentación.* Los algoritmos encuadrados en este grupo generan primero un conjunto de “cajas” con las coordenadas de cada uno de los objetos para en un segundo paso segmentarlo del fondo. Un ejemplo de este tipo de arquitectura sería Mask R-CNN [He et al., 2017].

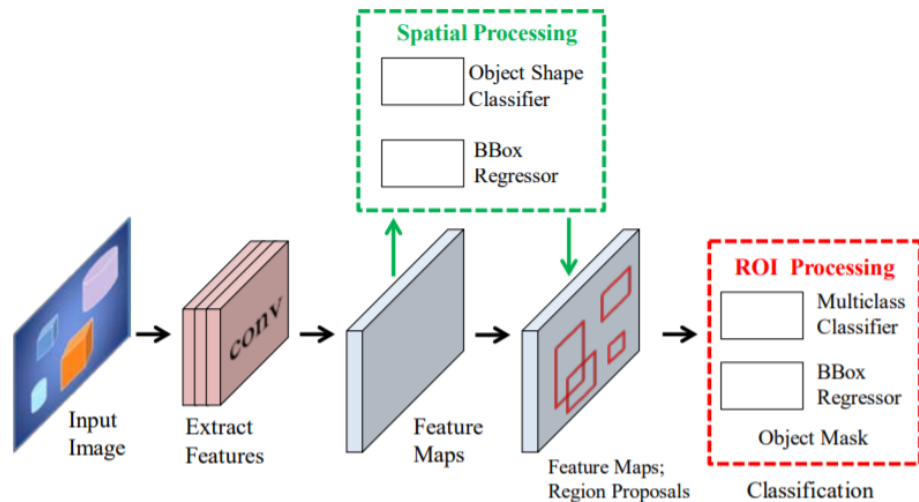


Fig. 7 Detección seguida de clasificación.

3. *Etiquetado de píxeles seguido de clustering.* Este tipo de técnicas pueden verse como una extensión de la segmentación semántica. Primero se realiza una clasificación de todos los píxeles que forman la imagen indicando su categoría. A continuación, se utilizan técnicas de *clustering* para agrupar los píxeles adyacentes con la misma etiqueta y que presentan características similares. Como ejemplos de este grupo nos encontramos con el algoritmo Deep Watershed [Bai y Urtasun, 2017] y con InstanceCut [Kirillov et al., 2017].

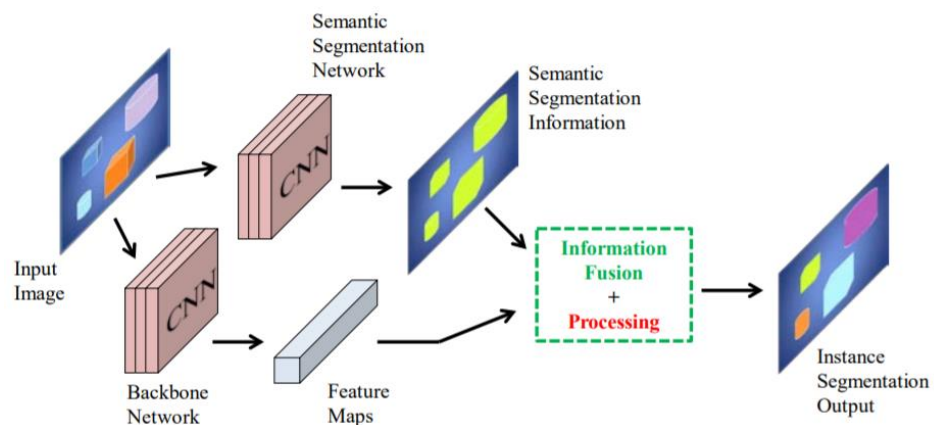


Fig. 8 Etiquetado de píxeles seguido de clustering.

4. *Métodos basados en el desplazamiento de ventanas.* Este grupo basa su funcionamiento en desplazar un conjunto de ventanas con distintas escalas y relaciones sobre la imagen generando así un conjunto de máscaras. Finalmente, estas máscaras son pasadas a un clasificador. Los métodos propuestos en [Pinheiro, 2015] y [Dai, 2016] serían ejemplos de este grupo.

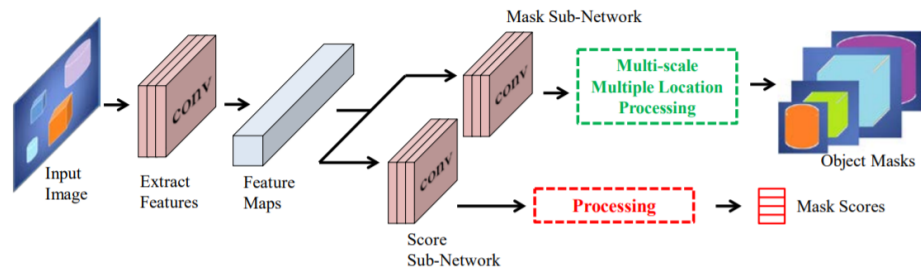


Fig. 9 Métodos basados en el desplazamiento de ventanas.

2. MODELO PROPUESTO

El proceso propuesto para hallar la biomasa en un estanque consiste en el análisis de imágenes de sonar de ultrasonidos, hallando la cantidad y peso de los individuos que aparecen en la escena. Este análisis sigue la filosofía “*etiquetado de píxeles seguido de clustering*”, es decir, en primer lugar, se realiza una segmentación semántica separando los píxeles correspondientes a pez de los píxeles correspondientes a fondo utilizando para ello una CNN, para luego aplicando técnicas de *clustering* identificar todas las instancias.

2.1. ARQUITECTURA

Un *autoencoder* [Rumelhart et al., 1975] es un tipo específico de red neuronal capaz de aprender sin etiquetas (aprendizaje no supervisado), representaciones densas de los datos de entrada llamadas representaciones latentes. Luego esta representación latente es decodificada de nuevo de manera que la entrada reconstruida sea lo más parecida posible a la salida deseada.

Los *autoencoders* se dividen en dos subredes:

- *Encoder*: es la parte encargada de comprimir los datos de entrada en un espacio latente.
- *Decoder*: es la parte encargada de reconstruir los datos de salida a partir del espacio latente.

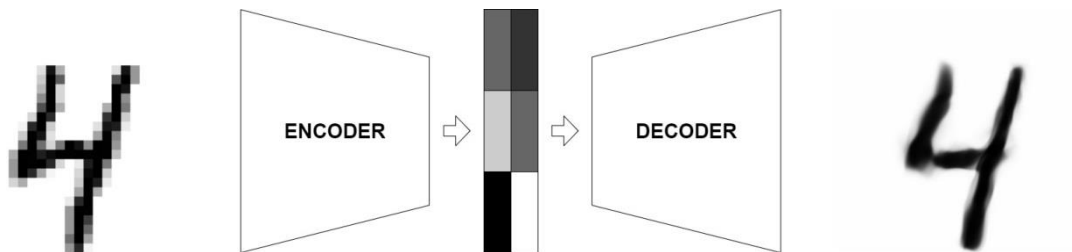


Fig. 10 Esquema de un autoencoder.

La arquitectura empleada en este trabajo está basada en un tipo de *autoencoder* denominado U-Net [Ronneberger, 2015] y que cuenta con la particularidad de contar con conexiones desde capas pertenecientes al *encoder* hacia capas pertenecientes al *decoder* permitiendo así la propagación de la información.

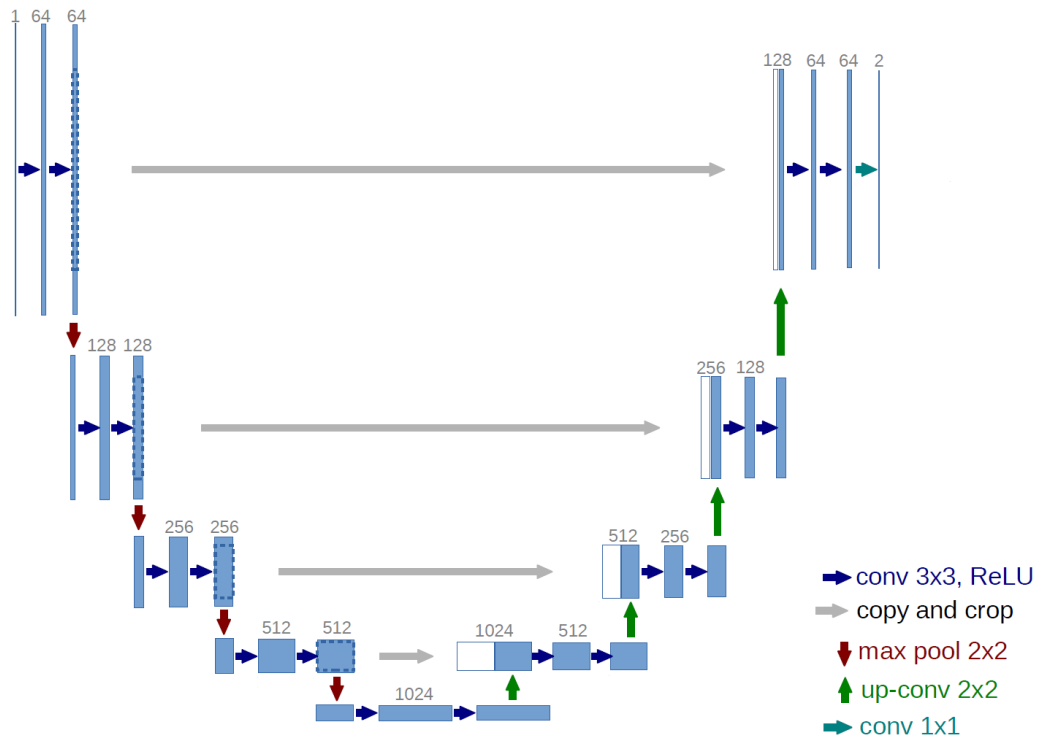


Fig. 11. Arquitectura de la red propuesta.

Una vez segmentados las regiones correspondientes al pez del fondo, se procede a detectar las distintas instancias. Esto se consigue aplicando técnicas de *clustering* como puede ser el algoritmo *watershed* [Vincent y Soille, 1991; Romero y Reinoso, 2018].

El algoritmo *watershed* es un proceso ampliamente extendido en el análisis de relieves topográfico, cuyo cometido es el de buscar hacia que mínimo o valle tendería una gota que cae sobre este relieve. De forma análoga, una imagen en escala de grises puede ser vista como un relieve topográfico donde la altura de cada píxel viene determinada por su nivel de intensidad.

El modo de funcionamiento de este algoritmo consiste en partir de un conjunto de píxeles semillas los cuales corresponden a mínimos locales, a los que de forma iterativa se van añadiendo los píxeles vecinos que presentan características similares como puede ser el color o la intensidad.

A la hora de tratar con cardúmenes, es decir, agrupaciones de peces que se desplazan juntos, este algoritmo resulta especialmente útil. Si estos peces se encuentran lo suficientemente juntos, en el mapa de segmentación pueden aparecer como un único pez. El algoritmo *watershed* permitiría delimitarlos ya que se trata de un método de segmentación orientado a regiones el cual permite extraer las fronteras de las regiones existentes en la imagen.

2.2. CÁLCULO DE LA BIOMASA. PRIMERA APROXIMACIÓN

En el cálculo de la biomasa es indispensable conocer el peso medio de los peces, así como el número de individuos con tamaño comercial.

En entornos acuícolas con una única especie, como es el caso que nos ocupa, se puede conocer el peso de un pez sabiendo su altura ya que existe una relación directa entre ambas magnitudes. Para conocer el peso de un espécimen que aparece en una imagen de sonar de ultrasonidos, basta con segmentarlo del resto de la imagen con los métodos descrito anteriormente y calcular su altura en píxeles. Posteriormente, esta medida es convertida en altura real conociendo la relación de escala proporcionada por el sonar. Sabiendo este dato se puede calcular el peso del pez a través de la fórmula:

$$L_T = 2.7711 * H + 2.3891, \quad (n = 57; r^2 = 0.9387; p < 0.05)$$
$$W = 0.054 * L_T^{3.31}, \quad (n = 57; r^2 = 0.9787; p < 0.05)$$

donde L_T es la longitud total (cm); H es la altura total entre la aleta dorsal y el comienzo de las aletas pélvicas (cm) y W es el peso fresco (g). Estas relaciones fueron obtenidas experimentalmente a partir de ejemplares capturados en la piscifactoría en estudio.

Los sonar de imagen de ultrasonidos son dispositivos que permiten medir profundidades y las distancias hasta objetos que se encuentren bajo el agua. Este tipo de dispositivos funcionan emitiendo un ultrasonido el cual rebota al encontrarse con un obstáculo; calculando el tiempo de retorno y sabiendo la velocidad de propagación podemos calcular la distancia a la que se encuentra dicho obstáculo. El haz que emite el sonar no se propaga en línea recta, sino que describe un cono, por lo que el escaneo se produce sobre un espacio 3D cuyo volumen puede ser calculado sabiendo el ángulo de apertura del dispositivo. Esta característica ha sido el punto de partida para el cálculo del número total ejemplares.

El número total de peces en un estanque de la piscifactoría se calcula adaptando la idea previa usada para el conteo de aves a partir de los datos obtenidos por un grupo de observadores (o detectores), los cuáles desde un punto fijo registraron los ejemplares detectados en bandas concéntricas de anchura prefijada [Ramsey y Scott, 1979; Reynolds et al., 1980; Fancy, 1997]. En los trabajos anteriores, esta detección se hizo a lo largo de todas las direcciones, sin embargo, en este trabajo solo se tiene en cuenta la sección cubierta por el haz del sonar, el cual está fijo y siempre orientado hacia la misma dirección.

En este método se asume que la detectabilidad de un espécimen es una función [$g(x) = 1 - cx$] la cual depende de una constante c [Robbins, 1981; McCracken, 1994]. En este método la estimación de la densidad está basada en el concepto de 'cono de detectabilidad' que establece que aproximadamente el 66% de los

especímenes localizados alrededor del sonar (dentro y fuera de detección) no van a ser detectados.

Para hallar el total de peces en el estanque, se ha calculado en primer lugar la densidad de especímenes por metro cúbico en la región cubierta por el haz del sonar. Esto se consigue tomando imágenes durante un periodo de tiempo y calculando la media de peces que aparecen en la escena. Esta densidad es calculada mediante la función:

$$\overline{D}_t = 1.66 \frac{360n_t}{\pi r^2 \alpha h}$$

donde \overline{D}_t es la densidad estimada en el momento t ; r es el radio de detección del sonar; α es el ángulo que determina la anchura del haz del sonar; y h es la altura media de la balsa. Se aplica un factor de 1.66 para corregir el número de peces no detectados. A continuación, sabiendo el volumen de la balsa, extrapola el total de especímenes multiplicando el valor de esta densidad por el volumen total del estanque.

Con las imágenes obtenidas, se calcula una distribución de los pesos de los peces y el número de individuos totales tal y como se ha descrito anteriormente. A continuación, usando estos dos valores se halla una estimación de la biomasa total existente en la balsa.

3. PRUEBAS DEL MODELO

3.1. CONFIGURACIÓN DE LAS PRUEBAS

Para el entrenamiento de nuestro modelo se ha hecho uso de un servidor con las siguientes características:

- 2 procesadores Intel Xeon Gold con 20 núcleos cada uno.
- 512GB de RAM.
- 10 GPUs NVIDIA RTX QUADRO 6000 con 24GB cada una.

Aunque se dispone de varias GPUs, el entrenamiento se ha hecho de forma local, es decir, un modelo por GPU. Esto ha permitido entrenar de forma paralela el mismo modelo con tres funciones de coste que se detallan más abajo.



Fig. 12 Clúster de GPUs usado en este trabajo

El conjunto de datos disponible está formado por un total de 4039 imágenes de sonar de ultrasonidos. Estas imágenes fueron recogidas durante dos días no consecutivos y de forma estática. En otras palabras, se recogieron con el sensor sumergido y siempre desde la misma posición y orientación se tomaron las distintas imágenes. Esto puede verse en la Fig. 13.



Fig. 13 Transductor del sonar de imagen LVS32 de Garmin, acoplado a un módulo LiveScope GLS10 (Sistema Panoptix LiveScope™).

Por otro lado, en las imágenes obtenidas se observa que la balsa tiene una pequeña pendiente. Para evitar el aprendizaje de esta zona por parte del modelo, se han usado técnicas de *data augmentation*; más concretamente, se ha rotado con respecto al eje vertical el 50% de las imágenes de modo que la pendiente aparece en ambos lados de la imagen.

El proceso de generación de las máscaras de salida ha consistido en el etiquetado de forma manual de cada una de las imágenes generando por cada pez que aparece en la escena una retícula. Posteriormente se han aplicado técnicas de *clustering* para agrupar píxeles que tienen características similares. Este agrupamiento se ha hecho usando el algoritmo *k-means* de esta forma se ha dividido la imagen en un total de 6 clústeres; la elección del número de clústeres se ha realizado siguiendo el método de *elbow* [Thorndike, 1953]. Para finalizar, se han seleccionado los píxeles que pertenecen a los clústeres que son de nuestro interés y que se encuentran dentro de las retículas

generadas previamente. Un ejemplo de este proceso puede ser visto en las Fig. 14, Fig. 15 y Fig. 16 Etiquetado de las distintas instancias. Fig. 16.

3.2. MÉTRICAS DE EVALUACIÓN

Para la evaluación de nuestro modelo se hace uso de diferentes métricas, concretamente: *Dice Coefficient*, *Sensitivity* y *Specificity* [Jadon, 2020].

- ***Dice Coefficient (DC)***. Calcula el índice de similitud entre dos muestras. Se define como dos veces la intersección de dos conjuntos entre el sumatorio de las áreas de dichos conjuntos. Esta métrica viene definida por la función:

$$DC = \frac{2TP}{2TP + FP + FN}$$

- ***Sensitivity (TPR)***. Calcula el índice de las predicciones positivas correctas entre el total de positivos. Esta métrica viene definida por la función:

$$Sensitivity(TPR) = \frac{TP}{TP + FN}$$

- ***Specificity (TNR)***. Calcula el índice de las predicciones negativas correctas entre el total de negativos. Esta métrica viene definida por la función:

$$Specificity(TNR) = \frac{TN}{TN + FP}$$

3.3. AJUSTE DEL MODELO

Para el ajuste de nuestro modelo, se ha hecho uso del optimizador *Adam* [Kingman y Ba, 2014] con los siguientes parámetros: $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ y $\epsilon = 10^{-8}$; el artículo que propuso esta técnica fijaba estos valores, y son los que hemos usado. Con la idea de buscar la función de coste que mejor se ajuste a nuestras imágenes, se ha hecho uso de tres funciones de coste distintas. Estas han sido escogidas atendiendo a las características del problema: clasificación binaria con gran desbalanceo entre clases.

- ***Binary Cross-Entropy*** [Yi-de et al., 2004]. La entropía cruzada se define como una medida de la diferencia entre dos distribuciones de probabilidad para una determinada variable aleatoria. Es ampliamente utilizada para problemas de clasificación, como es el caso de la segmentación semántica que es una clasificación a nivel de píxeles.

$$L_{BCE}(y, \hat{y}) = (y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

- ***Dice Loss*** [Sudre et al., 2017]. *Dice Coefficient* es una métrica muy utilizada para calcular la similitud entre dos imágenes. En 2016 esta métrica fue adaptada como función de coste dando lugar a la función *Dice Loss*.

$$DL(y, \hat{p}) = 1 - \frac{2y\hat{p} + 1}{y + \hat{p} + 1}$$

- **Focal Loss** [Lin et al., 2017]. Variante de Binary Cross-Entropy. Reduce la contribución de la clase mayoritaria para permitir al modelo centrarse más en la clase minoritaria. Funciona especialmente bien en problemas donde existe un gran desbalanceo de clases.

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t)$$

3.4. RESULTADOS OBTENIDOS Y ANÁLISIS

La **Tabla 1** muestra los resultados obtenidos por el modelo propuesto. Observando esta información vemos que:

- El mejor resultado es obtenido empleando la función de coste *Focal Loss*. Pensamos que esto es así debido al hecho de que esta función fue diseñada para problemas en los que existe un gran desbalanceo de clases, esto es, hay muchas más clases de una categoría que de otra. En nuestro caso, tenemos muchos más píxeles correspondientes a fondo que a pez.
- El peor resultado se obtiene para la función de coste *Dice Loss*. Esta función es una adaptación del evaluador *DC* el cual se basa en medir el nivel de solapamiento entre la fuente de verdad y el resultado obtenido. El hecho de que los resultados obtenidos para esta función de coste sean menores que la del resto pensamos que es debido a que no existen unos bordes definidos y es aquí precisamente donde se comete el error.

Función de coste	Métricas de evaluación		
	<i>DC</i>	<i>TPR</i>	<i>TNR</i>
<i>Binary Cross-Entropy</i>	0.8669	0.7850	0.9995
<i>Dice Loss</i>	0.7778	0.7442	0.9968
<i>Focal Loss</i>	0.8702	0.7888	0.9995

Tabla 1. Comparación de las funciones de coste según las distintas métricas de evaluación

Por otro lado, observando las imágenes que se obtienen a la salida del modelo, se ven regiones no etiquetadas que el modelo identifica como peces. Esto sucede principalmente en imágenes donde existen regiones de ruido relativamente grandes. No se considera esto relevante ya que a ojos de un experto es difícil discernir si realmente se trata de un pez o no.

En cuanto a la capacidad del modelo para predecir la biomasa en el estanque en el que se ha probado, el resultado estimado es de 11141 peces con un peso promedio de 238 g y una biomasa total de 2.6 tn. Estos resultados son muy próximos a los valores suministrados y estimados por los gestores de la piscifactoría (unos 15000 peces de

250 g). Debe tenerse en cuenta que es muy difícil conocer la cantidad real de peces introducidos en un estanque ya que ésta es una aproximación realizada por los gestores a partir de un peso total del lote introducido y un peso promedio. Este procedimiento por regla general sobrestima el número de ejemplares en el lote. Así mismo, hay una cantidad significativa de pérdidas no contabilizadas como consecuencia de la mortalidad natural y la causada por factores como el efecto de enfermedades, depredación por aves piscívoras y hurtos. Por tanto, se puede asumir que los resultados se aproximan significativamente al número de ejemplares presentes en el estanque.

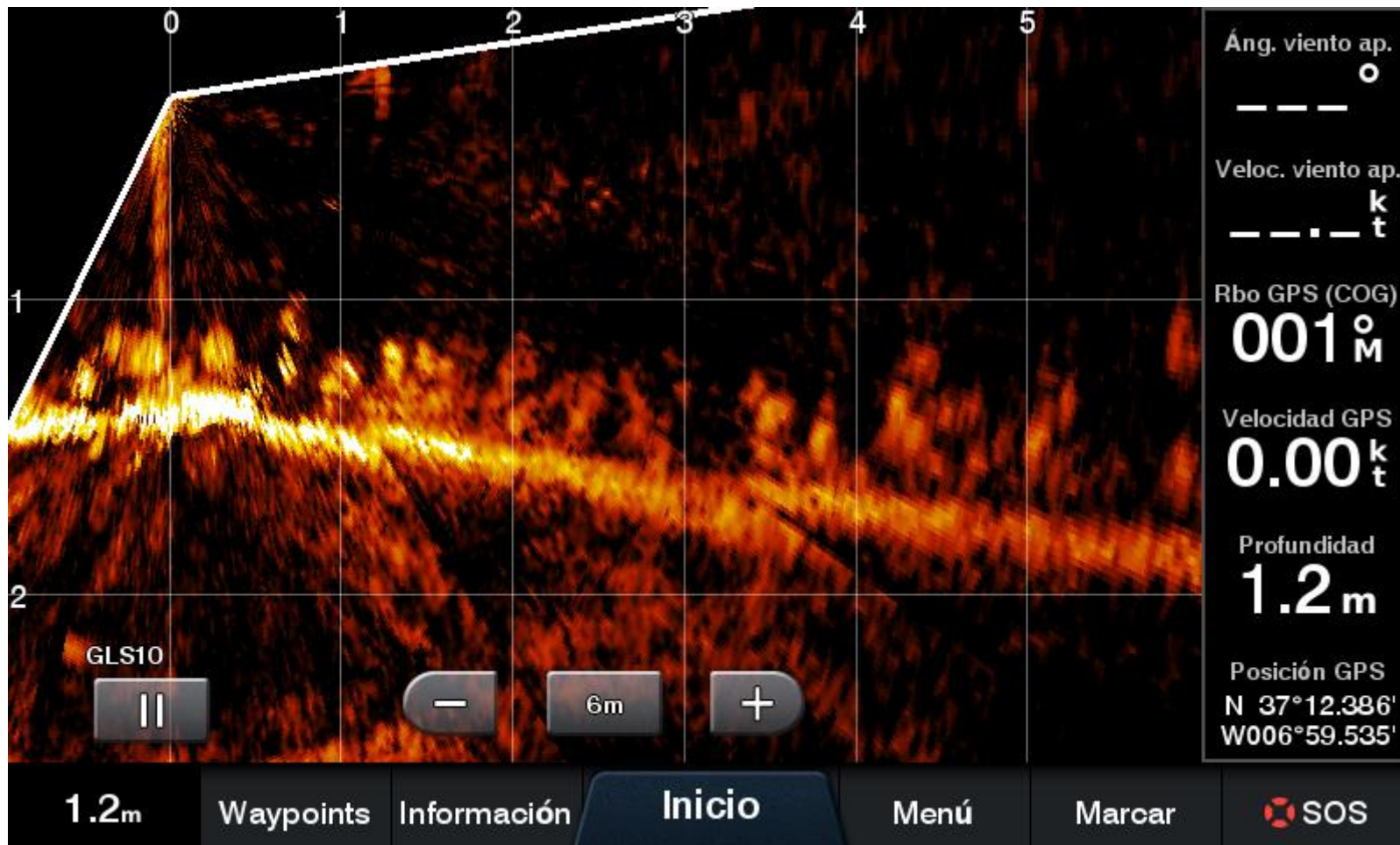


Fig. 14 Imagen de sonar de ultrasonidos.

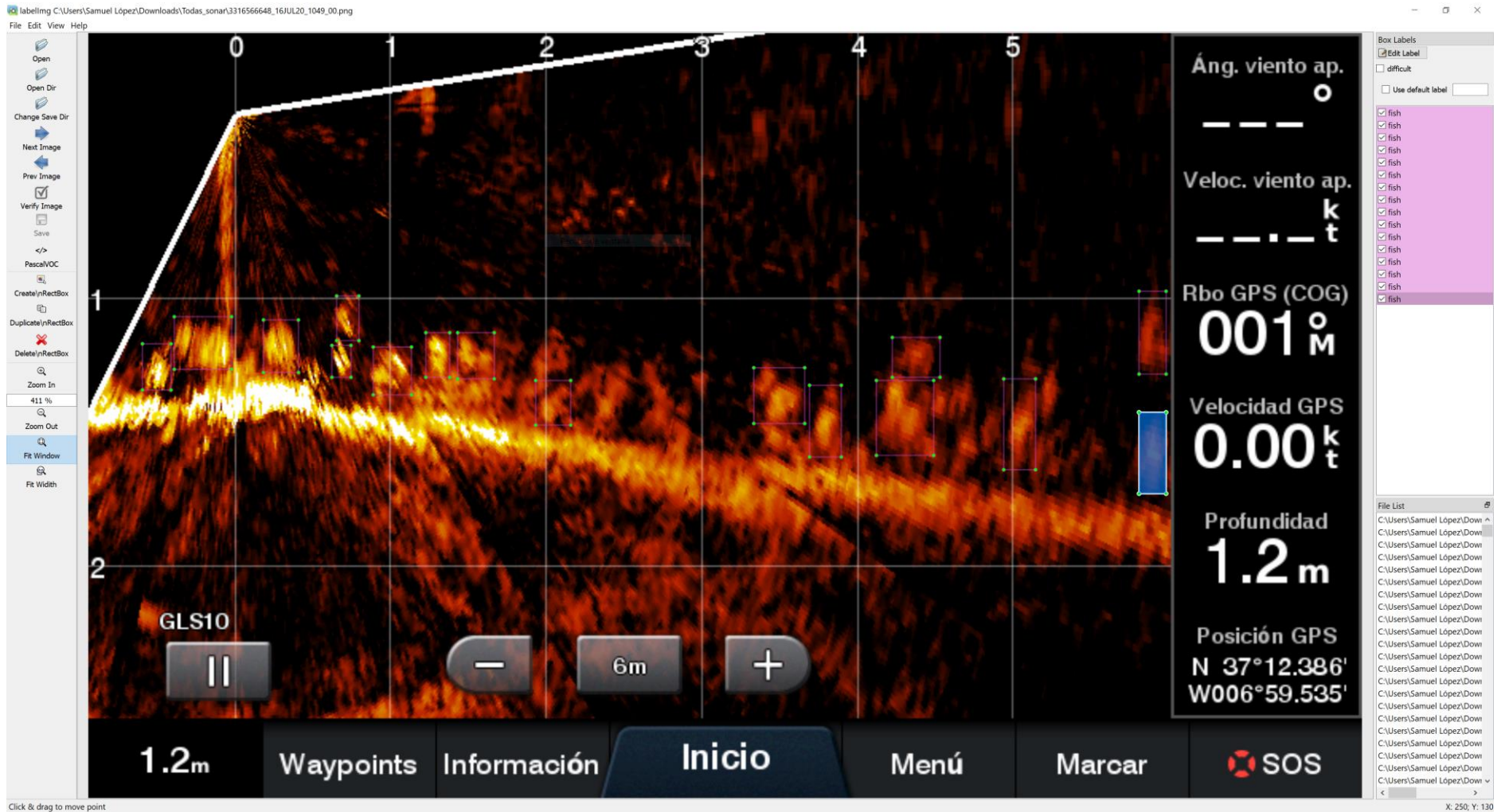


Fig. 15 Etiquetado manual de las imágenes usando el programa labellmg (<https://github.com/tzutalin/labellmg>).

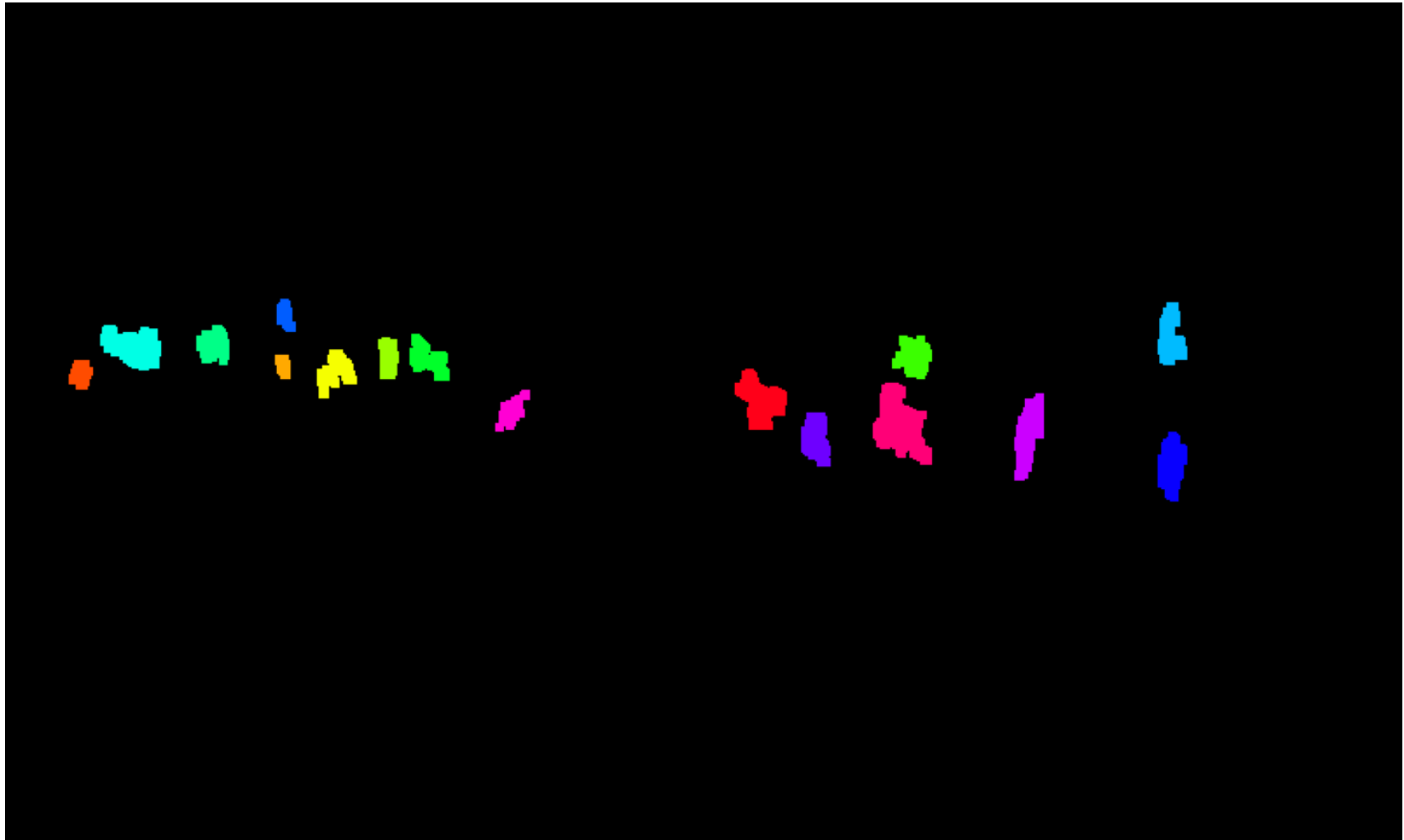


Fig. 16 Etiquetado de las distintas instancias.

4. CONCLUSIONES

El objetivo principal de este trabajo es desarrollar un sistema capaz de hallar de forma automática la biomasa existente en un estanque en cuyas aguas haya poca visibilidad, utilizando para ello imágenes de baja resolución como pueden ser las obtenidas con un sonar.

El resultado de este trabajo es positivo, es decir, se ha conseguido desarrollar un sistema capaz de segmentar los peces a partir de las imágenes de sonar. Además, el modelo consigue un gran nivel de *DC*, esto es, el nivel de solapamiento entre las imágenes obtenidas a la salida del modelo y las imágenes de referencia es bastante alto, lo que permite estimar unos pesos de los peces de una forma que se aproxima bastante a la realidad.

En adelante se plantea el contrastar la estimación obtenida con la biomasa real de modo que sea cuantificable la calidad de esta medida. La validación no se ha podido realizar ya que para ello se debería haber procedido a desembalse del estanque utilizado para este estudio con los problemas anteriormente planteados.

Por otro lado, se plantea también extender este trabajo al uso de imágenes de video en lugar de imágenes estáticas ya que se considera que gran parte de la información para la determinación de dónde hay un animal está en su propio movimiento.

5. REFERENCIAS

- M. Haddon, "Modelling and quantitative methods in fisheries". Second edition, Ed. Taylor and Francis Group, LLC, Boca Raton, 449 pp, 2011.
- Y. LeCun, Y. Bengio, y G. Hinton, "Deep learning", Nature, vol. 521, no. 7553, p. 436, 2015.
- Y. Lecun, L. Bottou, Y. Bengio y P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.
- A. M. Hafiz y G. M. Bhat, "A survey on instance segmentation: state of the art". International Journal of Multimedia Information Retrieval, 1-19, 2020.
- R. Girshick, J. Donahue, T. Darrell y J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", in IEEE Conference on Computer Vision and Pattern Recognition, pp. 580-587, 2014.
- R. Girshick, "Fast R-CNN", in IEEE International Conference on Computer Vision (ICCV), pp. 1440-1448, 2015.
- S. Ren, K. He, R. Girshick y J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149,
- K. He, G. Gkioxari, P. Dollár y R. Girshick, "Mask R-CNN", in IEEE International Conference on Computer Vision (ICCV), pp. 2980-2988, 2017.
- M. Bai y R. Urtasun, "Deep Watershed Transform for Instance Segmentation", in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2858-2866, 2017.
- A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy y C. Rother, "InstanceCut: From Edges to Instances with MultiCut", in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7322-7331, 2017.
- P. O. Pinheiro, R. Collobert y Piotr Dollár, "Learning to segment object candidates", in Proceedings of the 28th International Conference on Neural Information Processing Systems, vol. 2, pp. 1990–1998, 2015.
- J. Dai, K. He, Y. Li, S. Ren y J. Sun, "Instance-sensitive fully convolutional networks", in European Conference on Computer Vision, pp. 534-549, 2016.
- D. E. Rumelhart, G. E. Hinton y R. J. Williams, "Learning internal representations by error propagation". California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

O. Ronneberger, P. Fischer y T. Brox, "U-net: Convolutional networks for biomedical image segmentation", in International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234-241, 2015.

L. Vincent y P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations", in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 6, pp. 583-598, 1991.

R. Romero-Zaliz y J. F. Reinoso-Gordo, "An updated review on watershed algorithms" in Soft computing for sustainability science, pp. 235-258, 2018.

F. L. Ramsey y J. M. Scott, "Estimating population densities from variable circular plot surveys", in R. M. Cormack, G. P. Patil, y D. S. Robson, eds. Sampling biological populations. International Cooperative Publishing House, pp. 155-181, 1979.

R. T. Reynolds, J. M. Scott y R. A. Nussbaum. "A variable circularplot method for estimating bird numbers", Condor 82, pp. 309-313, 1980.

S. G. Fancy, "A New Approach for Analyzing Bird Densities from Variable Circular-Plot Counts", in Pacific Science, vol. 51, no. 1, pp. 107- 114, 1997.

C. L. Robbins, "Effect of time of day on bird activity", in Stud. Avian Biol. 6:275-286, 1981.

M. L. McCracken, "Factors affecting bird counts and their influence on density estimates". Ph.D. diss., Oregon State University, Corvallis, 1994

R. L. Thorndike, "Who belongs in the family?", Psychometrika, vol.18, no.4, pp.267–276, 1953.

S. Jadon, "A survey of loss functions for semantic segmentation", in IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), pp. 1-7, 2020.

D. P. Kingma y L. Ba Jimmy, "Adam: A method for stochastic optimization", 2014. arXiv preprint arXiv:1412.6980.

M. Yi-de, L. Qing y Q. Zhi-bai, "Automated image segmentation using improved PCNN model based on cross-entropy", Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, pp. 743-746, 2004.

C. H. Sudre, W. Li, T. Vercauteren, S. Ourselin y M. J. Cardoso, "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations", In Deep learning in medical image analysis and multimodal learning for clinical decision support (pp. 240-248), 2017.

T. Lin, P. Goyal, R. Girshick, K. He y P. Dollár, “Focal Loss for Dense Object Detection”, in IEEE International Conference on Computer Vision (ICCV), pp. 2999-3007, 2017.



Interreg

España - Portugal



UNIÓN EUROPEA
UNIÃO EUROPEIA

Fondo Europeo de Desarrollo Regional
Fundo Europeu de Desenvolvimento Regional