

Hyperparameters Tuning of Faster R-CNN Deep Learning Transfer for Persistent Object Detection in Radar Images

Rosa Gonzales-Martínez, Javier Machacuay, Pedro Rotta, César Chinguel

Abstract—The radars have become a key sensor for civil and military applications including air, maritime and ground traffic control and surveillance. This is due to their advantages in detecting and locating objects in long-range and all weather. In previous work, a methodology was proposed to obtain a sea surface object detection model based on the Faster R-CNN architecture using commercial navigation radar images from Sperry Marine. Unfortunately, the recall using the validation dataset was 0.7576 with a minimum score for true positives of 7% due to a network overfitting issue. In this research, the overfitting problem is solved by comparing three experiments. Each experiment consist of the combinations of different hyperparameters within the Faster R-CNN architecture. The main hyperparameters modified to improve the performance of the model were the initialization of weights and the optimizer. The results, finally achieved show a significant improvement in relation to the previous work. The improved persistent object detection model shows a recall of 0.9394 with a minimum true positive score of 98%.

Index Terms—Faster R-CNN, hyperparameters, object detection model, overfitting, persistent objects.

I. INTRODUCCIÓN

Hoy en día, la detección de objetos en el océano a partir de imágenes de radar es ampliamente utilizada en distintas actividades en el campo civil y militar. Entre estas actividades destacan el monitoreo del océano con capacidad de trabajo durante todo el día y en cualquier clima [1], control de tráfico [2], ordenación pesquera [3], protección del medio ambiente [4] así como, rescates en caso de desastres [5].

Uno de los principales problemas de la tecnología de radar es la mitigación de los ecos no deseados del entorno, denominado desorden o clutter; proveniente de colinas, edificios, aves, movimiento de las olas del mar, o concentraciones de cuerpos ligeros suspendidos en el aire (chaff) [6]. Por otra parte, la presencia de ruido, que es la materialización de interferencias de alta frecuencia, afecta la señal tomada por el receptor del radar. Por ello se puede decir, que las imágenes de radar no se componen únicamente de contactos de interés sino también, de clutter y ruido. En consecuencia, esta dispersión en las imágenes genera variabilidad en la detección de objetos, falsas

alarmas o pérdida de información principalmente en zonas cercanas a la costa [7], [8].

Recientemente, el desarrollo de radares comerciales de bajo costo han hecho posible que muchas aplicaciones civiles se beneficien de estos sensores. Los radares comerciales tienen menor alcance, resolución y confiabilidad que los radares militares. Sin embargo, el desarrollo en tecnología microondas, radiofrecuencia, métodos avanzados de procesamiento de señales, detección y clasificación son esenciales para explotarlos de manera eficiente [9], [10].

En detección de embarcaciones a partir de imágenes de radar existen dos categorías: Métodos tradicionales basados en características concretas y métodos modernos basados en características abstractas [11]. Los métodos tradicionales detectan embarcaciones a través de características que son más accesibles de discernir. Entre ellas sobresalen: Retrodispersión, polarización, caracterización geométrica, Histograma del Gradiente de Orientación (HOG) [12], co-ocurrencia de múltiples características similares a Haar [13], Patrón Binario Local (LBP) [14] y Transformada de Características de Escala Invariante (SIFT) [15].

Los métodos basados en características de retrodispersión son los más fundamentales. Normalmente, los barcos fabricados de materiales metálicos tienen características de retrodispersión más fuertes, por lo que sus valores de gris son más altos. En el filtro de Tasa Constante de Falsas Alarmas (CFAR), se propuso establecer un modelo de distribución estadística [16]. Los algoritmos de umbral adaptativo están diseñados para buscar valores de píxeles que son inusualmente brillantes en comparación con los del área circundante. Los valores que se encuentran por encima del umbral se declaran inusualmente brillantes y, por lo tanto, probablemente son muestras de un objeto [17]. Sin embargo CFAR, el método tradicional más utilizado en la detección de objetos, resulta inestable en precisión cuando se trata de embarcaciones pequeñas (multiescala) y en escenarios cercanos a la costa (multiescena) [18].

Por otra parte, los métodos modernos basados en características abstractas detectan embarcaciones mediante características profundas de los datos (sombras, contornos, texturas) [19], [20] que se extraen a través de redes neuronales convolucionales profundas (CNN) sin intervención humana, logrando un rendimiento de detección superior a los métodos tradicionales [21]. Esta propuesta de detección de embarcaciones se basa en la confianza de las características que incluyen la estimación de la densidad del núcleo, la relación longitud-ancho y el número de píxeles. Los objetos con alta confianza

Rosa Gonzales-Martínez, Departamento de Ingeniería Mecánico-Eléctrica, Universidad de Piura, Piura, Perú.

Javier Machacuay, Departamento de Ingeniería Mecánico-Eléctrica, Universidad de Piura, Piura, Perú.

Pedro Rotta, Departamento de Ingeniería Mecánico-Eléctrica, Universidad de Piura, Piura, Perú.

César Chinguel, Departamento de Ingeniería Mecánico-Eléctrica, Universidad de Piura, Piura, Perú

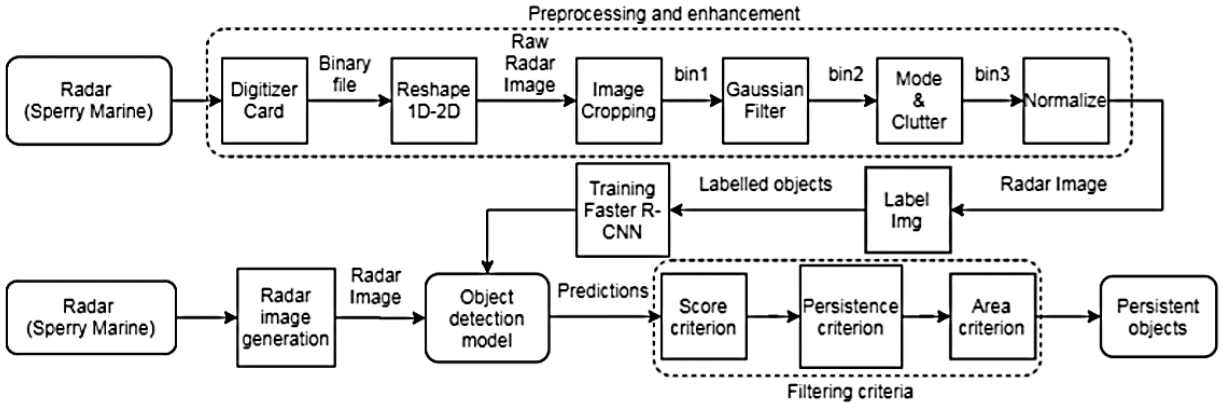


Fig. 1: Flow chart of the proposed methodology in the previous model.

en las características se interpretan como embarcaciones. Sin embargo, la existencia de ruido y clutter dificulta la extracción efectiva para la discriminación. Por ello, para obtener un mejor rendimiento de detección es necesario desarrollar un detector con una gran capacidad de extracción de características [22]. Este segundo grupo de métodos modernos basados en características abstractas se puede clasificar en detectores de dos etapas y de una etapa. Los detectores de dos etapas, representados por la serie de Redes Neuronales Convolucionales basadas en Regiones (R-CNN), que luego evolucionaron a Fast R-CNN y Faster R-CNN tienen la ventaja de reconocer con mayor precisión diferentes tipos de objetos, superando el problema de la detección a multiescala y multiescena [23], [24]. Por otro lado, los detectores de una etapa como You Only Look Once (YOLO) y Single Shot MultiBox Detector (SSD) funcionan mucho más rápido que los detectores de dos etapas sin embargo, comprometen la precisión, siendo insuficiente cuando se trata de objetos pequeños [25], [26].

Inspirados por estos métodos más avanzados intentamos introducirlos en un escenario complejo como es la detección de objetos en imágenes generadas por radares de navegación comercial Sperry Marine. En una investigación anterior desarrollamos una metodología basada en la arquitectura Faster R-CNN y generamos un modelo de detección de objetos empleando aprendizaje por transferencia (TL), este método aplicado favoreció de manera eficaz los cálculos y tiempo de cálculo [27]. Con el modelo generado obtuvimos un rendimiento aceptable en la detección de objetos persistentes [28]. Sin embargo, creemos que es posible mejorar el rendimiento de nuestro modelo anterior.

Diferentes investigaciones han logrado incrementar el rendimiento de sus modelos basados en CNN considerando un mayor volumen de datos en su entrenamiento [29], argumentando que el aumento de datos mejora el rendimiento de CNN, evita el ajuste excesivo y es fácil de implementar [30], [31]. Otros trabajos, sustentan que los hiperparámetros determinan la arquitectura de la capa en el paso de extracción de características de una CNN. La selección de hiperparámetros es muy importante para el éxito del entrenamiento de las CNN y esto afecta la precisión de la clasificación y el tiempo de aprendizaje. Sin embargo, elegir valores adecuados para

los hiperparámetros es complicado ya que no sólo depende del nivel de experiencia, sino también de la capacidad de aprendizaje de los valores obtenidos en cada iteración de prueba, además puede resultar intensivo en mano de obra y consumo de tiempo [32], [33], [34], [35].

Por lo expuesto, en este trabajo presentamos un método novedoso para ajustar hiperparámetros para la transferencia de aprendizaje profundo en el problema de detección de objetos en imágenes de radar. Realizamos una serie de experimentos para redefinir los hiperparámetros de configuración más adecuados. Evaluamos la siguiente hipótesis: Si realizamos el afinamiento de hiperparámetros podríamos mejorar el rendimiento del modelo. Los hiperparámetros que redefinimos son IoU, Batch-size, épocas, inicialización de pesos y el optimizador. Cada uno de los hiperparámetros tuvo un efecto diferente en la capacidad de detección de los modelos, siendo la combinación de los dos últimos hiperparámetros los que contribuyeron a obtener mejores resultados para el modelo.

El resto de este documento se organiza de la siguiente manera. El modelo de línea base se resume en la Sección 2. La metodología de la investigación conjuntamente con la configuración de los experimentos se presentan en la Sección 3. El análisis de los resultados se presenta en la Sección 4. Por último las conclusiones se presentan en la Sección 5.

II. MODELO DE LÍNEA BASE [28]

El modelo de línea base comprende tres fases: Preprocesamiento y mejoramiento de las imágenes de radar, entrenamiento y generación del modelo de detección de objetos, y el filtrado de las predicciones para obtener los objetos persistentes. La metodología que utilizamos se muestra en (Fig. 1).

En la primera fase, las imágenes de radar con resolución de 1024 x 2048 píxeles de tamaño, fueron sometidas a varios filtros de convolución y normalizadas para reducir el clutter y el ruido. En la segunda fase, las imágenes normalizadas pasaron por un proceso de entrenamiento empleando TL. Utilizamos la Interfaz de Programación de Aplicaciones (API) de detección de objetos de Tensorflow (API-TFOD). Se realizó el etiquetamiento, definiendo las coordenadas de los cuadros delimitadores que contienen los objetos que serán detectados y

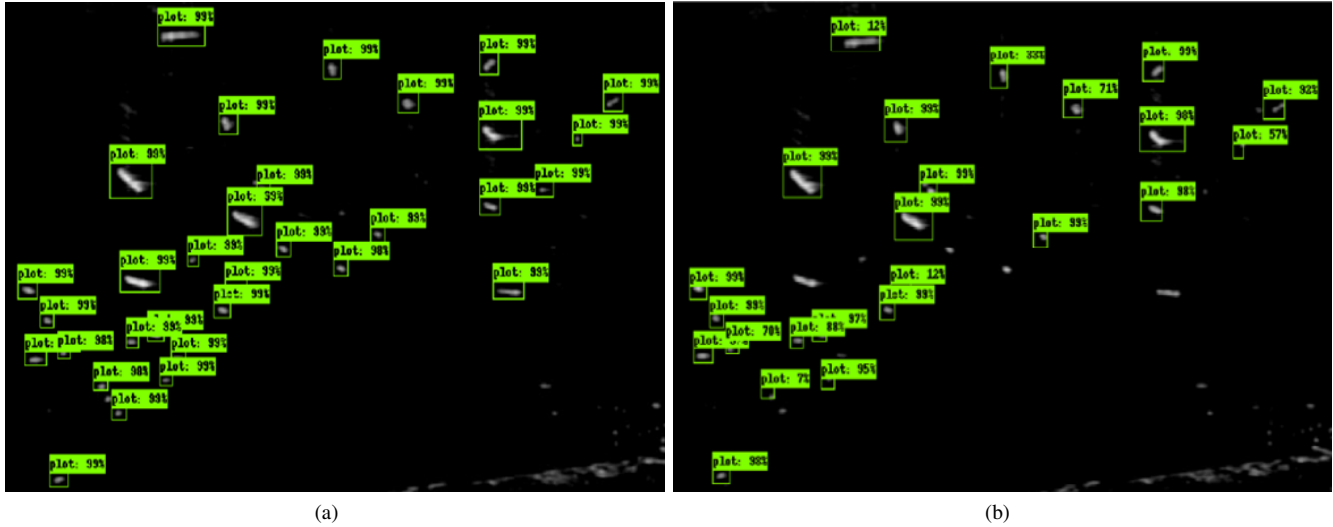


Fig. 2: Predictions obtained from previous work with the model based on Faster R-CNN in radar images on the Peruvian coast, Callao: (a) Objects detected on data set of training images, recall=0.9697; (b) Objects detected on data set of testing images, recall=0.7576.

recuperados posteriormente. Asimismo, elegimos la arquitectura de aprendizaje profundo más adecuada evaluando tanto la velocidad de detección como la precisión [36]. De las alternativas posibles, elegimos la arquitectura Faster R-CNN Inception V2, priorizando el modelo que permita un alto rendimiento en términos de precisión en la detección de objetivos, sin ser menos importante el tiempo de procesamiento.

La arquitectura Faster R-CNN comprende cuatro pasos: Extracción de características, Red de Regiones Propuestas (RPN), agrupación y clasificación de regiones de interés (ROI). La RPN genera un conjunto de propuestas de objetos rectangulares, cada una con una puntuación de objetividad. El uso de anclas o cuadros de diferentes tamaños y escalas conjuntamente con la regresión softmax y la regresión del cuadro delimitador permiten corregir las propuestas seleccionadas y obtener propuestas más precisas. Esto quiere decir, que la RPN se enlaza con la estructura convolucional de la Fast R-CNN, para generar las regiones de contorno y encontrar los objetos deseados [37], [38], [39]. Utilizamos la arquitectura Faster R-CNN con el extractor de características Inception V2. Esta versión particular de CNN se centra en la capacidad de ampliar las redes neuronales sin aumentar el costo computacional. Esto se realiza mediante la aplicación de un nuevo bloque de construcción conocido como el "módulo de inicio". Un módulo de inicio Inception, calcula múltiples transformaciones diferentes sobre el mismo mapa de entrada en paralelo, y para reducir la dimensionalidad usa convoluciones 1x1. Las convoluciones 1x1 permiten disminuir los canales del tensor de salida y a la vez, apilar más capas. De esta manera se resuelve el cuello de botella computacional [40], [41], [42]. En esta segunda fase además, cambiamos las primeras capas del modelo importado para inicializar los pesos. Asimismo, reemplazamos las últimas capas con una configuración específica de clases y reemplazamos el conjunto de datos predeterminado con imágenes de radar de navegación comercial Sperry Marine.

Finalmente, en la tercera fase las predicciones obtenidas pasaron por un proceso de filtrado para recuperar solo los objetos persistentes considerando tres criterios: Una puntuación de confianza mayor a 85%, la persistencia de objetos en dos vueltas de radar, actual y anterior, y los objetos que abarcan el área de energía reflejada del radar o Radar Cross Section (RCS) inferior a $700000 m^2$.

En el modelo de línea de base se logró un recall de 0.7576 con el conjunto de datos de evaluación en un tiempo total de procesamiento de 0.1730 segundos. Sin embargo, los resultados reflejan una brecha con relación al conjunto de datos de entrenamiento, recall=0.9697 [28]. Los resultados obtenidos con el modelo de línea base se muestran en (Fig. 2).

III. METODOLOGÍA

La metodología abarcó la selección de hiperparámetros y la ejecución de experimentos. En cada experimento se llevó a cabo la preparación de datos estableciendo diferentes combinaciones de hiperparámetros; entrenamientos y la evaluación de métricas por cada uno de los experimentos. La metodología para llevar a cabo los experimentos se muestra en (Fig.3).

A. Selección de hiperparámetros

1) Intersection over Union (IoU)

El IoU es utilizado en las dos fases que comprende la arquitectura Faster R-CNN. Precisamente para entrenar la red de regiones propuestas o RPN, se asigna una etiqueta de clase binaria (de ser un objeto o no) a cada cuadro delimitador en cada posición de la imagen sobre el mapa de características. El valor o puntaje de la etiqueta se basa en la intersección sobre unión (IoU) con la caja que contiene el objeto o caja de la verdad. Se asigna una etiqueta positiva a cualquier ancla colocada de tal manera que la IoU con la caja de verdad sea

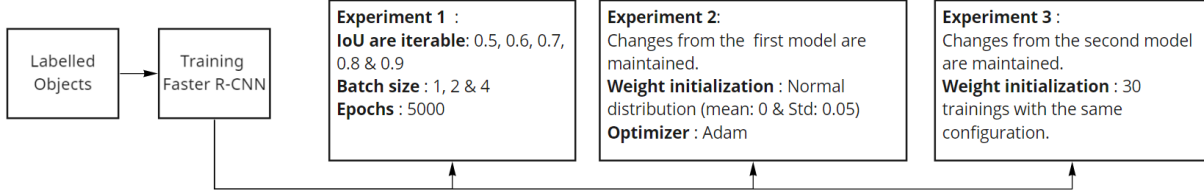


Fig. 3: Proposed methodology for the experiments.

mayor a un determinado umbral y se asigna una etiqueta con valor negativo, a los anclajes con IoU menor a un determinado umbral con respecto a la caja de verdad [43]. El cálculo de IoU se define así:

$$IoU = \frac{area(P \setminus G)}{area(P \cap G)} \quad (1)$$

Donde G es la posición real de las embarcaciones (caja de la verdad) y P es la posición predicha. Cuando IoU=1, el rendimiento de detección es el mejor. Pero, se puede considerar un resultado muy bueno cuando IoU>0,5 en la detección real [44], [45]. Consideramos que mejorar la IoU ayudaría a reducir la incertidumbre de los objetos y aumentaría la precisión de detección por ello, modificamos el valor del umbral entre 0.5 y 0.9.

2) *Epochs*

Realizamos varios experimentos modificando el número de épocas. Notamos que luego de un determinado número de épocas los resultados de la función de pérdida no mejoraba, Por ello, creemos que es necesario definir el número de épocas entre valores establecidos de manera correcta para obtener un modelo capaz de desempeñarse de manera óptima. Por tanto, para evitar un sobreajuste excesivo, limitamos la capacidad de la red mediante una parada anticipada [46]. Para todos los experimentos consideramos un valor de 5000.

3) *Batch size*

El tamaño del lote es el número de muestras de entrenamiento que se utilizan en una iteración. Al utilizar diferentes tamaños de batch size, la red no tiene que calcular todo el número de muestras en una sola iteración cuando se realiza el backpropagation, de ese modo la red ejecuta los cálculos con mayor rapidez, permitiendo mayor velocidad en el entrenamiento y la convergencia, aunque tiene ciertas desventajas. Si el batch size es igual a 1 puede ser que el proceso no llegue a tener los mejores rendimientos debido a la variabilidad en el proceso de iteración de pesos y si el batch size es igual al total de elementos del conjunto de datos, provoca mayor costo computacional. Por esta razón, generalmente se elige un valor entre 1 y el total de elementos del conjunto de datos de entrenamiento. Consideramos muestras de entrenamiento de tamaños de 1, 2 y 4 para completar la actualización de parámetros [47].

4) *Inicialización de pesos*

La inicialización de pesos afecta la probabilidad y la velocidad de convergencia y en consecuencia la general-

Table I: Hyperparameter configuration for experiment 1.

Hyperparameter	Baseline Experiment	Experiment 1
Epochs	200 000	5 000
IoU	0.7	0.5; 0.6; 0.7; 0.8; 0.9
Batch size	1	1; 2; 4

ización en redes neuronales. La inicialización de pesos es un factor clave en el entrenamiento de una red, ya que esta puede converger en un mínimo local y no en el mínimo global de la función de pérdida. Aún no hay una metodología para encontrar la mejor inicialización de pesos, pero en este trabajo elegimos algunos parámetros para redefinir el valor inicial de los pesos [48].

5) *Optimizadores*

El optimizador genera la dirección por la cual los pesos deben modificarse para garantizar la convergencia hacia el mínimo global o muchas veces local. La técnica matemática que se utiliza es la derivada parcial de los pesos que nos dará la dirección del mínimo descenso que es el gradiente (SGD). Los optimizadores permitirán encontrar mejores formas de cálculo de los pesos. Esto a la vez significa mayor rapidez para optimizar la función de pérdida. Para actualizar los pesos y sesgos en la red utilizamos al algoritmo de estimación de momento adaptativo (ADAM) similar al descenso gradiente estocástico (SGD) con las ventajas de un cálculo eficiente y consumo de menos memoria [49].

B. *Ejecución de experimentos*

1) *Experimento 1*

En el experimento 1 modificamos dos hiperparámetros: IoU y Batch-size. Consideramos 3 tamaños de Batch-size: 1, 2 y 4; los valores para IoU fueron 0.5; 0.6; 0.7; 0.8 y 0.9. El número de épocas predeterminado de 200000 fue configurado a 5000 épocas. Se realizaron 15 experimentos combinando los diferentes valores de los hiperparámetros tal como se muestra en la Tabla I.

2) *Experimento 2*

Los hiperparámetros modificados en el experimento 2 corresponden a la inicialización de pesos y el optimizador. Para la inicialización de pesos consideramos la distribución normal con $\mu=0$ y $\sigma=0.05$. Se realizaron 15 experimentos combinando los diferentes valores de los hiperparámetros tal como se muestra en la Tabla II. Posteriormente contrastamos los

resultados con el mejor modelo del experimento 1.

3) Experimento 3

Para este experimento seleccionamos los modelos que corresponden al rendimiento más alto y más bajo del experimento 2. Bajo la premisa, que no existe una metodología definida para alcanzar el mínimo global al momento de aplicar el descenso del gradiente [50]. Buscamos comprobar el nivel de aleatoriedad de los resultados de la función de pérdida. Para evaluar la validez de los valores asignados en la inicialización de pesos con $\alpha=0$ y $\beta=0.05$ consideramos un mínimo de 30 entrenamientos de 5000 épocas cada uno, como referencia válida para la distribución normal de valores.

C. Métricas de evaluación

1) Recall

Esta métrica mide la asertividad del detector de objetos para una clase determinada, representa el porcentaje de predicciones positivas correctas entre todas las verdades etiquetadas que han recuperado cada uno de los modelos. La métrica se denota así:

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

Donde TP es la suma de los verdaderos positivos en el conjunto de evaluación es decir, el número de instancias detectadas correctamente. FN representa los falsos negativos es decir, los contactos no detectados [51].

2) Función de pérdida

La función de pérdida denominada también función de costo, representa la medida de las pérdidas de clasificación y regresión del cuadro delimitador. Es deseable obtener un valor bajo, muy cercano a cero para los datos de evaluación; valores altos en el entrenamiento significaría que el modelo se encuentra sobreajustado o subajustado. En el caso de sobreajuste, la solución es encontrar los hiperparámetros correctos que puedan resolver este problema. La función de costo se define:

$$L(f; p_i; g; f; t_i; g) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i; p_i^*) + \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i; t_i^*) \quad (3)$$

Donde i , es el índice de un ancla en un minilote y p_i es la probabilidad predicha de que el ancla i contenga un objeto. La etiqueta de la caja de la verdad p_i^* es 1 si el ancla es positiva y es 0 si el ancla es negativa,

t_i es un vector que representa las 4 coordenadas parametrizadas del cuadro delimitador predicho, y t_i^* es el vector de la caja de la verdad asociada con un ancla positiva. La pérdida de clasificación L_{cls} se da por el registro de una de las dos clases (objeto versus no objeto). Para medir la pérdida de regresión, usamos $L_{reg}(t_i; t_i^*) = R(t_i - t_i^*)$ donde R es la función de costo. El término $p_i^* L_{reg}$ significa que la pérdida de regresión es activada solo para anclas positivas ($p_i^* = 1$) y es deshabilitada en caso contrario ($p_i^* = 0$). La pérdida de regresión activada se ejecuta en dos etapas: En la red RPN, se realiza una transformación de posición mediante regresión lineal para sintonizar la ubicación del ancla positiva con la caja de la verdad y en Faster R-CNN se realiza la transformación para reducir la brecha entre la posición del ROI y la caja de la verdad. Las salidas de las capas *cls* y *reg* consisten en p_i y t_i respectivamente. Los dos términos están normalizados por N_{cls} y N_{reg} y ponderado por un parámetro de equilibrio λ . Por último, la salida obtenida con el modelo se compara con el resultado deseado y los parámetros de la red se ajustan para minimizar la función de pérdida [36], [51].

3) Puntaje de confianza o Minimum Score

Es el porcentaje de confianza o certeza de detectar el objeto deseado. Los detectores de objetos predicen la ubicación de objetos de una clase determinada en una imagen con un puntaje de confianza. El cuadro delimitador identifica la posición del objeto y contiene un puntaje que indica qué tan seguro está el detector sobre esa predicción. Mientras más alto es este valor, mayor será la certeza de clasificación que logra el modelo siendo 100% el valor más óptimo. En cada uno de los resultados analizamos los porcentajes del mínimo puntaje de confianza o Minimum score.

Para todos los experimentos de este trabajo la función de costo y recall se expresaron con una exactitud de 4 decimales mientras que minimum score en porcentajes.

IV. ANÁLISIS DE RESULTADOS

La evaluación de modelos se realizó en base a: Un conjunto de cuadros delimitadores que contienen la verdad y un conjunto de detecciones predichas generadas por un modelo; cada una de las predicciones consta de un cuadro delimitador, una clase y un puntaje de confianza. Evaluamos el rendimiento de los modelos en base a las métricas y comparamos con los resultados obtenidos en el modelo de línea base.

Table II: Hyperparameter configuration for experiment 2.

Hyperparameter	Experiment 1	Experiment 2
Epochs	5 000	5 000
IoU	0.7	0.5; 0.6; 0.7; 0.8; 0.9
Batch size	2	1; 2; 4
Weight Initialization	By default	$\alpha=0$; $\beta=0.05$
Optimizer	SGDM	ADAM

Table III: Results of experiment 1.

		Baseline Experiment	Experiment 1
Hyperparameter	Epochs	200 000	5 000
	IoU	0.7	0.7
	Batch size	1	2
Metrics	Loss function	0.2000	0.0437
	Recall	25/33=0.7576	15/33=0.4545
	Minimum Score	7%	96%

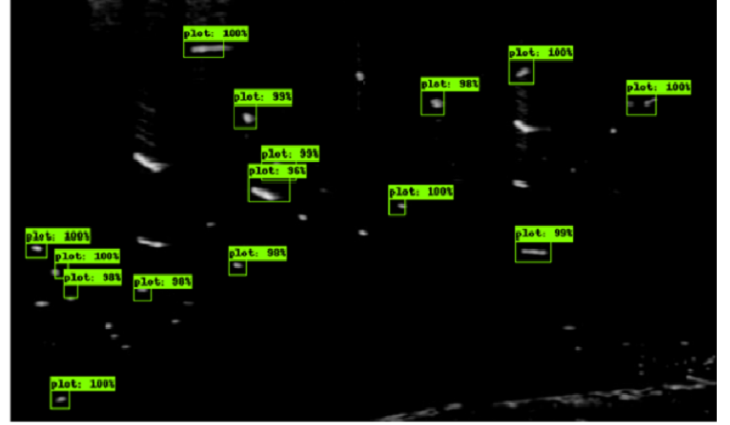
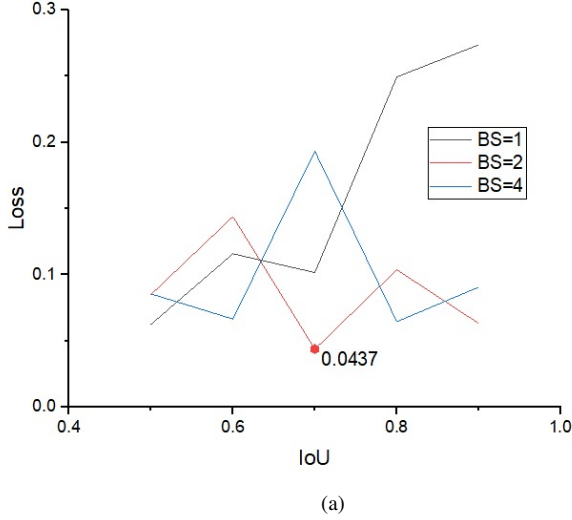


Fig. 4: Experiment 1 results with change of hyperparameters, IoU and Batch-Size in radar images on the Peruvian coast, Callao: (a) The best cost function=0.0437, when IoU=0.7 and Batch-size=2; (b) Bounding boxes annotations with Recall=0.4545 and Minimum Score=96%.

A. Experimento 1

Los resultados de la función de costo se muestran en (Fig. 4a). Cada punto representa una combinación de IoU con un valor de Batch-size. Para el caso de Batch-size=1, obtuvimos valores de función de costo más altos por el contrario, con Batch-size 2 y 4 obtuvimos valores más cercanos a 0. Observamos que la combinación IoU=0.7 y Batch-size=2 alcanzó el mejor valor para la función de costo. En cuanto a las predicciones en (Fig. 4b) observamos que solo 15 objetos fueron detectados correctamente, estos son los verdaderos positivos (TP) de un total de 33 objetos. A la vez, los 18 objetos que no fueron detectados corresponden a los falsos negativos (FN). Asimismo, apreciamos que el mínimo puntaje de confianza alcanzó un 96%. Presentamos un comparativo de los resultados con el modelo de línea base en la Tabla III, apreciamos que la función de costo mejoró significativamente logrando un valor de 0.0437. Con relación al recall observamos que no mejoró a pesar que el valor de puntaje de confianza aumentó. El criterio para elegir la mejor configuración ha sido lograr buenos resultados no solo en la función de costo y el puntaje de confianza sino también en recall. Los resultados obtenidos no cubrieron las expectativas.

B. Experimento 2

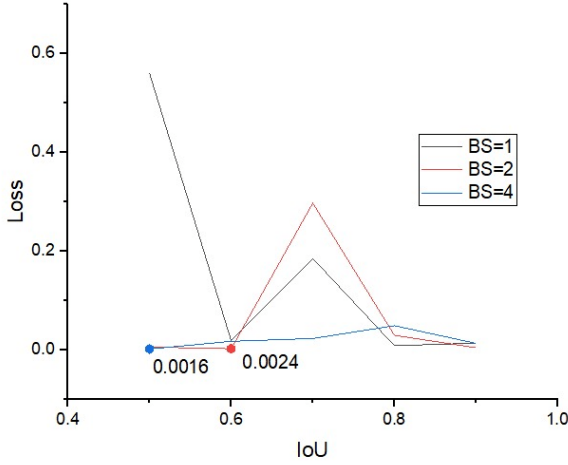
Considerando los nuevos valores para la inicialización de pesos ($\mu = 0$ y $\sigma = 0.05$) y el cambio del optimizador de SGDM a ADAM, se obtuvieron nuevos valores para la función de costo tal como se muestran en (Fig. 5a). Observamos que los resultados son aún más cercanos a 0 que el experimento anterior. La combinación IoU=0.6 con Batch-size=2 arrojó un valor 0.0024 y la configuración IoU=0.5 con Batch-size=4 alcanzó un valor mínimo de 0.0016 respectivamente. En (Fig. 5b) mostramos los resultados de las predicciones y apreciamos que la combinación de hiperparámetros IoU=0.6 con Batch-size=2, lograron detectar correctamente 31 objetos de los 33 objetos etiquetados. En la Tabla IV observamos que la combinación con IoU=0.5 y Batch-size=0.4 obtuvo el valor mínimo de 0.0016 para la función de costo; un recall de 0.8182 con una puntuación mínima de 86%. Sin embargo, según el análisis de las métricas de evaluación la mejor combinación de hiperparámetros corresponde a IoU=0.6 con Batch-size=2 ya que a pesar de tener un valor superior en función de costo de 0.0024; presenta valores más altos para Recall y el puntaje de confianza, equivalente a 0.9394 con una puntuación mínima para positivos verdaderos de 98% respectivamente. Los resultados obtenidos mejoraron significativamente y cubrieron las expectativas.

C. Experimento 3

El entrenamiento reiterativo ha sido una propuesta de experimento relacionado con el hiperparámetro de inicialización de pesos. La finalidad fue evaluar la variabilidad de los resultados de entrenamiento con la inicialización de los pesos ($\mu = 0$; $\sigma = 0.05$) y analizar el comportamiento de la función de costo en términos de mínimo local y global. Se tomaron las configuraciones de hiperparámetros del peor y el mejor modelo obtenidos en el experimento 2. Los resultados se aproximaron a una distribución normal y se analizó el comportamiento.

Table IV: Results of experiment 2.

Hyperparameter	Experiment 1	Experiment 2	Experiment 2
Epochs	5 000	5 000	5 000
IoU	0.7	0.6	0.5
Batch size	2	2	4
Weight Initialization	By default	$\mu=0$; $\sigma=0.05$	$\mu=0$; $\sigma=0.05$
Optimizer	SGDM	ADAM	ADAM
Metrics	Experiment 1	Experiment 2	Experiment 2
Loss function	0.0437	0.0024	0.0016
Recall	15/33=0.4545	31/33=0.9394	27/33=0.8182
Minimum Score	96%	98%	86%

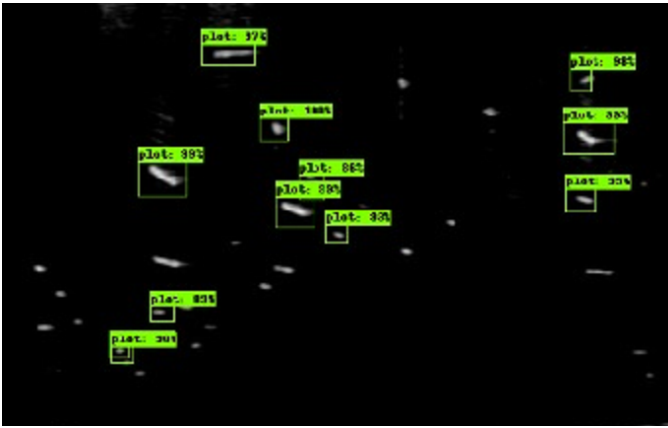


(a)



(b)

Fig. 5: Experiment 2 results with change of hyperparameters, weights initialization $\theta = 0$, $\sigma = 0.05$ and optimizer ADAM: (a) Cost function=0.0024, when IoU=0.6 and Batch-size=2; minimum value of cost function=0.0016, when IoU=0.5 and Batch-size=4; (b) Bounding boxes annotations with Recall=0.9394 and Minimum Score=98%.



(a)



(b)

Fig. 6: Prediction results with the same set of hyperparameters in different training order: (a) Bounding box annotations in the 1st training with Recall=0.3333 and Loss function=0.5598; (b) Bounding box annotations in the 24th training with Recall=0.9091 and Loss function=0.0042.

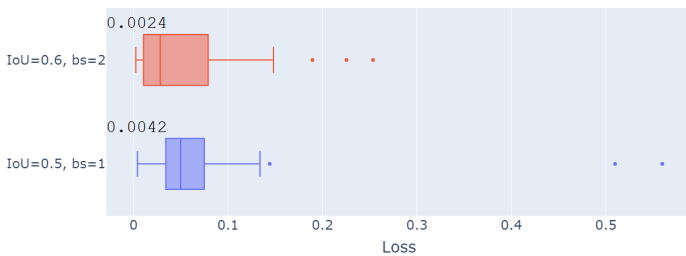


Fig. 7: Comparison of the cost function results of the two subsets of hyperparameters. The blue boxplot represents the set of hyperparameters of the worst model, when Batch-size=1 and IoU=0.5 and the red boxplot represents the set of hyperparameters of the best model, when Batch-size=2 and IoU=0.6. During 30 trainings, the worst model achieved a better cost function, close to the lowest cost function of the best model.

Los resultados reflejaron que se puede conseguir valores más bajos en la función de costo en el transcurso de cambio de pesos durante los entrenamientos es decir, la función de costo ejecuta un proceso de optimización y esto no significa que logre el mejor nivel de optimización o el mínimo global en el primer entrenamiento. Los resultados se muestran en (Fig. 6). Finalmente, observamos un comparativo de los 30 entrenamientos en términos de función de costo para los dos conjuntos de hiperparámetros en (Fig. 7).

V. CONCLUSIONES

El cambio de valores en los hiperparámetros inicialización de pesos $\theta = 0$, $\sigma = 0.05$ y optimizador Adam lograron elevar el desempeño del modelo de detección de línea base. Podemos asegurar esto en base a los resultados de los experimentos cuyo valor en función de costo descendió de 0.2000 a 0.0024; la precisión en términos de recall aumentó de 0.7576 a 0.9394 y

el puntaje mínimo de confianza de 7% a 98% respectivamente. El método de aprendizaje por transferencia resulta beneficioso para escenarios donde el conjunto de datos corresponde a imágenes de radar comercial.

El modelo de detección obtenido puede contribuir a la innovación de los algoritmos tradicionales de detección de objetos, reduciendo la tasa de falsas alarmas y pérdida de información sobre todo en zonas costeras.

Finalmente, para futuras investigaciones, realizaremos un estudio detallado sobre la adaptación de arquitecturas basadas en YOLOv4 empleando aprendizaje por transferencia y evaluaremos si garantiza una ventaja en tiempo real y precisión [52], [53].

AGRADECIMIENTOS

Rosa Gonzales-Martínez reconoce que esta investigación denominada "Hyperparameters Tuning of Faster R-CNN Deep Learning Transfer for Persistent Object Detection in Radar Images" ha sido financiada por el Proyecto Concytec - Banco Mundial "Mejoramiento y Ampliación de los Servicios del Sistema Nacional de Ciencia Tecnología e Innovación Tecnológica" 8682-PE, a través de su unidad ejecutora ProCiencia [Contrato no. 06-2018-FONDECYT/BM], ejecutado como parte del programa de Doctorado en Ingeniería con mención en Automatización, Control, y Optimización de Procesos, desarrollado en el Departamento de Ingeniería Mecánico-Eléctrica de la Universidad de Piura, Perú.

REFERENCES

- [1] U. Kanjir, H. Greidanus, K. Oštir, "Vessel detection and classification from spaceborne optical images: A literature survey," *Remote Sensing of Environment*, Volume 207, 2018, pp. 1-26, ISSN 0034-4257, <https://doi.org/10.1016/j.rse.2017.12.033>.
- [2] F. Meyer, S. Hinz, A. Laika, D. Wehling, R. Bamler, "Performance analysis of the TerraSAR-X Traffic monitoring concept," *ISPRS Journal of Photogrammetry and Remote Sensing*, Volume 61, Issues 3-4, 2006, pp. 225-242, ISSN 0924-2716, <https://doi.org/10.1016/j.isprsjprs.2006.08.002>.
- [3] M. Petit, J. Stretta, H. Farrugio and A. Wadsworth, "Synthetic aperture radar imaging of sea surface life and fishing activities," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 30, no. 5, pp. 1085-1089, Sept. 1992, doi: 10.1109/36.175346.
- [4] A.K. Mazur, A.K. Wählin, A. Krężel, "An object-based SAR image iceberg detection algorithm applied to the Amundsen Sea," *Remote Sensing of Environment*, Volume 189, 2017, pp. 67-83, ISSN 0034-4257, <https://doi.org/10.1016/j.rse.2016.11.013>.
- [5] C. N. Koyama, H. Gokon, M. Jimbo, S. Koshimura, M. Sato, "Disaster debris estimation using high-resolution polarimetric stereo-SAR," *ISPRS Journal of Photogrammetry and Remote Sensing*, Volume 120, 2016, pp. 84-98, ISSN 0924-2716, <https://doi.org/10.1016/j.isprsjprs.2016.08.003>
- [6] D. Purizaga-Céspedes, "Analysis of a New Two-Parameter Filter for Contact Detection in Marine Radar Images," M.S. thesis, Department of Mechanical-Electrical Engineering, University of Piura, Perú, 2018.
- [7] S. Kingsley and S. Quegan, "Understanding Radar Systems," in *SciTech Publishing, Inc.*, ed. 1992 originally, USA: McGraw-Hill, 1992, pp. 38-42.
- [8] Gini Fulvio, "Chapter 10 -Introduction to the Radar Signal Processing," in Academic Press, Regno Unito di Gran Bretagna, 2014. DOI: 10.1016/B978-0-12-396500-4.00010-7.
- [9] S. H. Javadi, A. Farina, "Radar networks: A review of features and challenges," *Information Fusion*, Volume 61, 2020, pp. 48-55, <https://doi.org/10.1016/j.inffus.2020.03.005>
- [10] D. Callaghan, J. Burger and A. Mishra, "A machine learning approach to radar sea clutter suppression," *Conference: 2017 IEEE Radar Conference (RadarConf17)*, pp. 1222-1227. DOI: 10.1109/RADAR.2017.7944391.
- [11] T. Zhang, X. Zhang, J. Shi, S. Wei, "HyperLi-Net: A hyper-light deep learning network for high-accurate and high-speed ship detection from synthetic aperture radar imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, Volume 167, 2020, pp. 123-153, September 2020.
- [12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 2005, pp. 886-893 vol. 1. DOI: 10.1109/CVPR.2005.177.
- [13] T. Mita, T. Kaneko and O. Hori, "Joint Haar-like features for face detection," Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, 2005, pp. 1619-1626 Vol. 2. DOI: 10.1109/ICCV.2005.129.
- [14] S. Yi-Kang and C. Ching-Te, "Local binary pattern orientation based face recognition," *ICASSP 2015 - 2015 IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2015. DOI:10.1109/ICASSP.2015.7178138.
- [15] David G. Lowe, "Distinctive Image Features from Scale-Invariant Key-points," *Computer Science Department, University of British Columbia, Vancouver, B.C., Canada International Journal of Computer Vision* 60(2), 91-110, 2004.
- [16] W. An, C. Xie and X. Yuan, "An Improved Iterative Censoring Scheme for CFAR Ship Detection With SAR Imagery," in *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 8, pp. 4585-4595, Aug. 2014. DOI: 10.1109/TGRS.2013.2282820.
- [17] D. Crisp, "The state-of-the-art in ship detection in Synthetic Aperture Radar imagery", Defence Science and Technology Organisation Salisbury (Australia) Info Sciences Lab, Edinburgh, Australia, Research Rep., Mayo, 2004. [Online]. Available: <https://apps.dtic.mil/sti/pdfs/ADA426096.pdf>
- [18] M. Kang, X. Leng, Z. Lin and K. Ji, "A modified faster R-CNN based on CFAR algorithm for SAR ship detection", *International Workshop on Remote Sensing with Intelligent Processing (RSIP)*, 2017, pp. 1-4. DOI: 10.1109/RSIP.2017.7958815.
- [19] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998. DOI: 10.1109/5.726791.
- [20] Y. LeCun, B. Yoshua, H. Geoffrey, "Deep learning," *Nature* 521, 436-444, 2015. DOI: <https://doi.org/10.1038/nature14539>.
- [21] Jayanth Koushik, "Understanding Convolutional Neural Networks," arXiv preprint, arXiv:1605.09081, 2016, <https://arxiv.org/abs/1605.09081>.
- [22] Y. Gui, X. Li and L. Xue, "A Multilayer Fusion Light-Head Detector for SAR Ship Detection," *Sensors*, 19(5), 1124, 2019. DOI:10.3390/s19051124
- [23] Y. Wang, Y. Zhang, H. Qu and Q. Tian, "Target Detection and Recognition Based on Convolutional Neural Network for SAR Image," 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), 2018, pp. 1-5. DOI: 10.1109/CISP-BMEI.2018.8633151.
- [24] J. Jiao, Y. Zhang, H. Sun, X. Yang, X. Gao, W. Hong, K. Fu, X. Sun, "A Densely Connected End-to-End Neural Network for Multiscale and Multiscene SAR Ship Detection", *IEEE Access*, vol. 6, pp. 20881-20892, 2018. DOI: 10.1109/ACCESS.2018.2825376.
- [25] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788. DOI: 10.1109/CVPR.2016.91.
- [26] W. Liu, D. Anguelov, D. Erhan, "SSD: Single Shot MultiBox Detector", *Proceedings of the Computer. Vision—ECCV 2016, Amsterdam, The Netherlands*, 8-16 October 2016; pp. 21-37.
- [27] Dey, N., Wu, Y., Wu, Q., and Sherratt, S., "Learning Models for Semantic Classification of Insufficient Plantar Pressure Images", *International Journal Of Interactive Multimedia And Artificial Intelligence*, 6(Special Issue on Soft Computing), 2020, pp. 51-61, <http://doi.org/10.9781/ijimai.2020.02.005>.
- [28] R. G. Martínez, J. M. Vera and C. C. Arrese, "Real-Time Detection Method of Persistent Objects in Radar Imagery with Deep Learning", 2020 IEEE Engineering International Research Conference (EIRCON), Lima, Peru, 2020, pp. 1-4. DOI: 10.1109/EIRCON51178.2020.9254021.
- [29] M. Dong, Y. Cui, X. Jing, X. Liu and J. Li, "End-to-End Target Detection and Classification with Data Augmentation in SAR Images," 2019 IEEE International Conference on Computational Electromagnetics (ICCEM), 2019, pp. 1-3. DOI: 10.1109/COMP.2019.8779096.
- [30] C. Shorten, T.M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J Big Data* 6, 60 (2019). <https://doi.org/10.1186/s40537-019-0197-0>

