
SISTEMA DE RECONOCIMIENTO DE BARCOS DESDE IMÁGENES DE SATÉLITE

**Blanco Pérez, Martín. González Leal, Guillermo
Lijó Sánchez, Manuel. Míguez García, Nicolás. Pérez Luis, Brandan**

**Aprendizaje Automático
Universidade da Coruña
Curso 2021-22**

Índice

1. Introducción.	1
1.1. Glosario	2
1.2. Consideraciones previas	2
1.3. Estructura de la memoria	4
2. Descripción del problema.	5
2.1. Características:	5
2.2. Restricciones:	5
2.3. Descripción de la base de datos	5
2.4. Origen de la base de datos	6
2.5. Propiedades de los datos	6
3. Análisis bibliográfico.	7
4. Desarrollo.	8
4.1. Aproximación 1	8
4.1.1. Descripción	8
4.1.2. Resultados	11
4.1.3. Discusión	14
4.2. Aproximación 2	15
4.2.1. Descripción	15
4.2.2. Resultados	16
4.2.3. Discusión	19
4.3. Aproximación 3	19
4.3.1. Descripción	20
4.3.2. Resultados	21
4.3.3. Discusión	23
4.4. Aproximación 4	24
4.4.1. Descripción	24
4.4.2. Resultados	25
4.4.3. Discusión	28
4.5. Aproximación 5	28
4.5.1. Descripción	28
4.5.2. Resultados	29
4.5.3. Discusión	34
5. Conclusiones	36
6. Trabajo futuro	39
5. Referencias.	40

1. Introducción.

El uso y control efectivo de las rutas marítimas para objetivos comerciales o militares, constituye una necesidad vital de los estados en el mundo. Algunos de los factores que ilustran la importancia de esta necesidad son:

- La vigilancia de bancos de pesca o caladeros protegidos legalmente. Los caladeros son zonas marítimas donde los pescadores tienden a calar sus redes por la existencia de condiciones favorables que facilitan la abundancia de pesca. Aquellas aguas dentro de las 200 millas marítimas desde la costa de un estado son caladeros nacionales, y es el mismo el que se encarga de regular la pesca en esas zonas. Por este razón con motivo de imponer el cumplimiento de sus leyes y normas en estos caladeros los estados necesitan medios para ejercer un control efectivo de los barcos que transiten por allí.
- Búsqueda e identificación de barcos, restos de aviones o contenedores perdidos en el mar. La detección temprana de barcos naufragados o aviones caídos en los que se ha perdido la conexión con el exterior, puede suponer la diferencia entre la vida y la muerte para sus pasajeros. Por otro lado, también puede ser importante para la recogida de mercancías perdidas, de alto valor o altamente contaminantes.
- Detección de barcos enemigos en frentes de guerra. En las guerras por mar, normalmente los barcos enemigos son muy difíciles de localizar debido a las muchas rutas que pueden seguir para llegar al frente en cuestión. Por este motivo, tener un medio para conocer la posición, tamaño y rutas de la flota enemiga permitiría prepararse para interceptar el ataque con mucha antelación, pudiendo significar esto la diferencia entre la victoria y la derrota.
- Evitar actividades de comercio ilícito, así como el tráfico ilegal de drogas por mar. El mar por su falta de control fronterizo por medio de aduanas, comparado con otros medios de transporte como los terrestres y aéreos siempre ha sido el preferido para el tráfico ilegal de todo tipo de mercancías. Por este motivo un control más exhaustivo de las costas y posibles atracaderos supondría un gran descenso de este tipo de ilegalidades.

Sin embargo, los medios existentes para ejercer este control son poco efectivos y costosos como la vigilancia por helicóptero o sensores marítimos en zonas costeras. Por este motivo surge la necesidad de poder monitorizar las actividades marítimas de un modo instantáneo y automático, en cualquier momento del día, a través de imágenes tomadas por satélite. Por tanto, el problema que nos planteamos resolver es la detección automática de barcos en imágenes tomadas por satélite.

1.1. Glosario

- AA: aprendizaje automático.
- SVM: Máquinas de soporte vectorial.
- RNA: redes de neuronas artificiales.
- DT: árboles de decisión.
- RNC: redes neuronales convolucionales.
- DL: Deep Learning.
- KNN: k vecinos más cercanos.
- RGB: formato de representación de colores en píxeles de imágenes mediante combinación de los tres colores primarios: rojo, verde y azul.
- STD: desviación típica.
- VPP: valor predictivo positivo.
- VPN: valor predictivo negativo.
- VP: verdaderos positivos.
- VN: verdaderos negativos.
- FP: falsos positivos.
- FN: falsos negativos.

1.2. Consideraciones previas

Para la tarea de detección de barcos en fotos es fundamental primero el aprendizaje del concepto de "barco". Cuando el aprendizaje lo realiza una máquina estamos hablando de Aprendizaje Automático. Según Mahesh, 2020 el Aprendizaje Automático (AA) se define como el campo de estudio que dota a los computadores de la habilidad de aprender sin ser explícitamente programados.

Más concretamente nosotros usaremos modelos de AA supervisados para clasificar una foto en dos clases distintas: barco, no barco. Según Singh y col., 2016 el aprendizaje automático supervisado es la construcción de algoritmos que pueden producir patrones e hipótesis generales mediante el uso de instancias suministradas externamente para predecir el destino de futuras instancias. En este caso, a partir del entrenamiento con imágenes, ya clasificadas, en las que pueden aparecer o no barcos queremos conseguir algoritmos que nos clasifiquen futuras imágenes a partir del conocimiento extraído de las anteriores. Para ello,

vamos a usar cuatro técnicas de las más habituales que se han propuesto para resolver este tipo de problemas (explicadas asimismo en el artículo de Singh y col., 2016):

- **Redes de neuronas artificiales** consisten en modelos computacionales basados en la estructura neuronal, el método de procesamiento y la capacidad de aprendizaje del cerebro humano, pero a menor escala.
- **Máquinas de soporte vectorial** (o SVM por sus siglas en inglés), su idea es representar el conjunto de instancias de entrenamiento en el espacio como puntos, para luego trazar un hiperplano que separe los puntos en dos clases distintas. De este modo, si llegan nuevas instancias se clasificarán en una u otra clase dependiendo del lado del hiperplano donde caigan. Está pensado para resolver problemas que son linealmente separables.
- **Árboles de decisión:** consisten una representación de los procesos de decisión involucrados en las tareas de clasificación. Consta de hojas, que describen la etiqueta asociada a una clasificación, y nodos, que son preguntas sobre el valor de un atributo. Cada rama representa un valor distinto del atributo por el que se pregunta en el nodo. Son fácilmente explicables a diferencia de las dos técnicas anteriores.
- **KNN** o los k vecinos más próximas, es otra de las técnicas que vamos a utilizar. Asigna a una instancia sin clasificar una etiqueta en función de las etiquetas de las k instancias, también llamadas prototipos, más próximas. Depende mucho de tener un buen valor de K.

Una diferencia importante a tener en cuenta dentro de estas técnicas es el determinismo en el entrenamiento de las mismas. Las SVM, DT, y KNN son técnicas determinísticas en el sentido de que si entrenamos el modelo con los mismos datos siempre nos va a dar los mismos resultados al aplicarlos a datos nuevos. Esto no sucede con las Redes de neuronas artificiales ya que tienen componente importante debida al azar en su inicialización, por lo que pudiéramos entrenar dos modelos con los mismos datos y que nos dieran resultados completamente distintos. Esto supone un inconveniente para comparar una red con otra u otro modelo distinto. En este caso lo paliamos creando y entrenando, con los mismos datos y parámetros, varias redes las cuales tendrán cada una inicializaciones distintas, para finalmente hacer el promedio de los resultados obtenidos, para cada una de las redes de la métrica escogida para evaluarlas. El promedio resultante será utilizado como la métrica misma, que se ha promediado, para evaluar una red con cierta configuración de hiperparámetros. De este modo, podemos estimar la bondad de un modelo dependiente del azar.

Los modelos anteriormente explicados son modelos clásicos de AA en los que primero se extraen manualmente las características de los datos, en este caso las imágenes y luego se realiza un entrenamiento supervisado a partir de esas características. Sin embargo, en los últimos años han ganado mucha importancia las Redes Neuronales Convolucionales, que

son la base del Deep Learning. Las RNC como se explica en LeCun y col., 2015 están diseñadas para procesar datos que vienen en forma de arrays de distintas dimensiones, como puede ser una imagen o señal de audio en 2D, videos o images volumétricas en 3D, etc. Es decir, trabajan directamente con las representaciones de los datos, encargándose de la extracción automática de características. Esta extracción se realiza en una etapa previa al entrenamiento de la red para la tarea de clasificación. Cuando esta extracción se divide en varias etapas, en las que en cada una se extrae característica de más alto nivel, decimos que se trata de una RNC profunda (de ahí viene el término de Deep Learning).

Por tanto, otro de los modelos que aplicaremos en nuestro problema es el de **RNC profunda**. Debido a la naturaleza de este modelo solo haremos una aproximación con el, ya que no tenemos que probar diferentes características.

1.3. Estructura de la memoria

La memoria consta de 4 apartados principales:

- **Introducción:** donde se ha contextualizado el problema, además de explicar la estructura de la memoria y consideraciones previas acerca de las técnicas utilizados.
- **Descripción del problema:** donde se explica en detalle el problema a resolver, indicando sus restricciones y características, además de describir la base de datos utilizada así como las propiedades de los mismos.
- **Análisis bibliográfico:** aquí se presenta el estado del arte del problema de detección de barcos con los avances que hay en el campo hasta ahora.
- **Desarrollo:** este es el desarrollo propio del problema a resolver con las distintas aproximaciones de problemas parciales o acotados resueltos que se siguen hasta la resolución del problema completo. Para cada aproximación se da una descripción de la misma, se comentan los resultados que obtuvimos al resolverla y se discute el éxito de la misma así como las aproximaciones futuras a desarrollar.
- **Conclusiones:** en este apartado comentamos como ha ido el trabajo, los modelos que nos han dado los mejores resultados y donde hemos tenido las mayores dificultades en el mismo.
- **Trabajo futuro:** este es el apartado en el que abrimos nuevas líneas de trabajo, relacionadas con el trabajo actual, a desarrollar en el futuro. Estas líneas tienen como objetivo el de profundizar aún más en el campo de detección de barcos, basándose en el trabajo realizado hasta ahora.

2. Descripción del problema.

Nuestro problema consiste en detectar la presencia barcos a partir de imágenes satelitales con vistas de la parte superior del barco, todos estos barcos están rodeados de agua, que varía de color entre imágenes, además, todas estas imágenes son a color, es decir, están en formato RGB y tienen distintos formatos y resoluciones. Nuestra base de datos cuenta con varias características y restricciones, entre las cuales debemos destacar:

2.1. Características:

- Todas las imágenes serán a color.
- Todas las imágenes contarán con al menos un barco, pudiendo haber más.
- En todas las imágenes los barcos estarán rodeados por agua, aunque pueden estar anclados en un puerto o existir otros objetos obstruyendo la imagen (islas, puentes, ...).
- Todos los barcos tienen la forma característica de éstos: alargado y estrecho.
- Todos los barcos se ven desde arriba.
- Todas las imágenes se han sacado de día y tienen buena iluminación.
- En todas las imágenes se puede apreciar la totalidad del barco (no hay imágenes con barcos cortados).

2.2. Restricciones:

- La resolución y calidad varían con cada imagen.
- Los barcos pueden estar estacionados o en movimiento, lo que implica que puede o no aparecer una estela de espuma detrás del barco.
- No hay un arquetipo de barco concreto, su color y forma varía, aunque sí existe una tendencia hacia los barcos de color blanco.
- Las imágenes pueden tener distintos formatos (16:9, 4:3, cuadrado, ...).

2.3. Descripción de la base de datos

La base de datos escogida cuenta con 621 patrones. Cada instancia cuenta con una imagen a color de tamaño variable. En secciones posteriores de la memoria indicaremos

características extraídas de estas imágenes. En la Figura 1 podemos ver algunas imágenes de ejemplo de la base de datos.



(a) Barco grande (con costa)



(b) Barco grande (solo mar)



(c) Varios barcos pequeños (con puerto)



(d) Muchos barcos pequeños (Solo mar)

Figura 1: Distintas imágenes de la BBDD

2.4. Origen de la base de datos

En nuestro caso, no hemos usado una base de datos de desarrollo propio, sino que hemos escogido una base de datos de dominio público disponible en Kaggle, subida por Yam Peleg, “Boats Boats Boats”, en el 2021-12-09. Peleg, 2021

2.5. Propiedades de los datos

Dado que la base de datos que usamos no es de cuenta propia, disponemos tan solo de los datos facilitados por su creador, que en nuestro caso son inexistentes. Sí que podemos destacar que existe una cota mínima en cuanto al número de barcos en cada imagen, existiendo al menos un barco en cada imagen.

3. Análisis bibliográfico.

Para conocer el avance de sector en la detección de barcos desde satélite analizamos varios documentos, de los cuales hablamos a continuación.

Un trabajo que soluciona el mismo problema que el nuestro es el de Yang y col., 2014, usando un nuevo método llamado (SDSSA). En este trabajo, primero analizan la imagen para saber si la superficie marina es homogénea o no y después utilizan un función lineal para seleccionar candidatos de barco. Los autores concluyen que este método es mucho más eficiente detectando barcos que los métodos existentes. Otro trabajo con la misma idea que el nuestro es el de Geronimo y col., 2018, donde utilizan imágenes nocturnas captadas mediante la tecnología VIIRS (*Visible Infrared Imaging Radiometer Suite*) para distinguir zonas pesqueras a través de detectar barcos pesqueros a través de las luces que emiten éstos a la hora de atraer a los peces cuando pescan por la noche. En este trabajo se consiguió distinguir satisfactoriamente zonas pesqueras juntando todas las detecciones de barcos.

Algunos trabajos publicados en estos últimos años resuelven el mismo problema que nuestro sistema utilizando Deep Learning e imágenes tomadas por satélites ópticos y satélites *Synthetic Aperture Radar* (SAR), como en el artículo de Kartal y Duman, 2019 en el que usan la API de *Tensorflow Object Detection* para la detección de barcos en imágenes de satélite con la que obtienen resultados prometedores o el de Motomura y Nagao, 2020 donde usan algoritmos como M-UNet o FishNet para la predicción de la actividad pesquera a través de imágenes satelitales que abarcan un mayor espacio. Los autores concluyen que el algoritmo que mejor funciona es FishNet, con un porcentaje de cobertura del 70 %. Otros autores, como Chang y col., 2019, utilizan un framework de Deep Learning, llamado *You Only Look Once* (YOLOv2), que según concluyen los autores supera al resto de tecnologías para la detección en tiempo real.

También hay trabajos más específicos, como el de Proia y Pagé, 2010, en el que utilizan un modelo bayesiano para detectar barcos de tamaño muy pequeño captados por satélite. Otro proyecto que mezcla la detección de barcos en imágenes de satélites con métodos estadísticos y técnicas de procesamiento de señales es el Corbane y col., 2010, que tiene el objetivo de detectar pesca ilegal y tráfico marino. En ambos trabajos, los autores reconocen que en los experimentos realizados la tasa de falsos negativos fue muy baja pero, sin embargo, la tasa de falsos positivos fue demasiado elevada (debido a nubes, estelas de barco, etc.).

Además de estos, cabe mencionar el trabajo de Krüger y Orlov, 2010 donde, además de detectar barcos a través de imágenes térmicas tomadas desde cámaras situadas en plataformas autónomas, como boyas o buques patrulla, también mantienen un seguimiento del barco detectado. Los resultados de este trabajo fueron prometedores y los autores tienen pensado seguir probando y mejorando el software basándose en los resultados obtenidos.

4. Desarrollo.

Para el preprocesado de datos del problema, decidimos tener en cuenta principalmente la diferencia entre los colores de los barcos y el del fondo de la imagen, que es casi siempre todo mar. Así tratamos de identificar los barcos dentro de la imagen y en aproximaciones posteriores, también los diferenciaremos de puertos y costas presentes en las fotos. Para ello empleamos la convolución, recortando la foto en partes más pequeñas para calcular la media de los colores en cada uno y según el resultado concluir en qué parte o partes de la imagen se encuentran los barcos buscando los recuadros con colores con una media menor.

En las aproximaciones empezamos dividiendo el conjunto de datos en 10 subconjuntos de los cuales 9 son de entrenamiento y el último es el de test, el que tendremos en cuenta a la hora de anotar los resultados de cada fold del cross-validation.

La métrica principal que se usa para los resultados de cada aproximación será la precisión, indicando el porcentaje.

4.1. Aproximación 1

Para la primera aproximación simplificamos la variedad de imágenes con las que trabajamos, acotando el dominio del problema, de forma que tan solo usamos imágenes de barcos sobre el mar (sin otros obstáculos como puertos, puentes...) y imágenes en las que tan solo haya mar. Para poder trabajar con estas imágenes como inputs debemos extraer características de ellas. En nuestro caso, convertimos las imágenes en arrays de 3 dimensiones en las que guardamos sus colores RGB. Además, calculamos la desviación estándar de los campos red, green y blue para cada una de las imágenes. Este último paso con el objetivo de ayudar a la red a distinguir el mar de los barcos en cada imagen. Sobre este problema de clasificación realizaremos la validación cruzada usando 4 modelos: RNA, SVM, decision tree y KNN. Sobre estos modelos aplicaremos distintas combinaciones de hiperparámetros y más adelante, en la sección resultados analizaremos y compararemos sus resultados.

4.1.1. Descripción

Para poder llevar a cabo esta aproximación, llevamos a cabo un preprocesado de la base de datos mencionada anteriormente, en el que recortamos las imágenes, de forma que el resultado del recorte entre en las dos categorías que ya mencionamos (barco o mar), posteriormente, separamos los casos negativos de los positivos en dos carpetas. Es importante mencionar que en las imágenes recortadas de los barcos, sus bordes se ajustan a los extremos del barco en cada eje, siempre manteniendo el barco al completo, como se puede ver en la Figura 2.

Como mencionamos anteriormente, para esta primera aproximación, decidimos extraer únicamente la desviación estándar de los colores RGB para cada imagen, esto, con el objetivo de poder diferenciar los barcos del agua. La desviación estándar proporcionará un valor mas alto cuanto menos similares sean los colores de los píxeles en la imagen. Es esta característica la que nos ayudará en nuestro problema de clasificación, ya que las fotos en las que solo haya mar tendrán una menor variación de colores, mientras que siempre que haya un objeto en la imagen con un color que resalte sobre el color del mar, existirá una mayor variación. Debido a esto y a que en esta primera aproximación acotamos las imágenes de entrada, solo permitiendo barcos sobre el mar o solo mar, podemos asumir que será suficiente para conseguir un buen resultado de clasificación.

Una vez obtenidas las matrices con los colores RGB, decidimos aplicar la normalización min-max, ya que por defecto, cada uno de los campos de una imagen RGB oscila entre 0 y 255. Utilizamos este tipo de normalización para reducir la redundancia de los datos, ya que, realiza una transformación lineal de los datos, ajustando los valores a un nuevo máximo y mínimo, que en nuestro caso serán 1 y 0, respectivamente. Decidimos acotar los datos entre estos valores con el objetivo de facilitar la interpretación y la extracción de características de los datos y evitar crear un bias en la red neuronal, ya que las unidades de medida pueden acabar afectando al análisis de datos y estandarizar la medida ayuda a que todos los datos sean relevantes.



(a) Barco



(b) No barco (mar)

Figura 2: Diferencia entre barco y no barco



(a) Barco (filtrado por STD)



(b) No barco (mar) (filtrado por STD)

Figura 3: imágenes de Figura 2 filtradas por STD

En las tablas 1 y 2 incluimos las características extraídas a partir de varias imágenes, en la tabla 1 serían de imágenes de barcos (patrones positivos) y en la 2 imágenes de mar solo (patrones negativos).

Características por valor RGB (imágenes de barcos)		
Red	Green	Blue
0.353772	0.331426	0.320017
0.284936	0.283531	0.251523
0.290564	0.292253	0.262237
0.218705	0.210569	0.217075
0.174222	0.153507	0.144507
0.350224	0.345159	0.348407
0.123104	0.11299	0.111436
0.130535	0.126107	0.125757
0.18674	0.184062	0.186881
0.228607	0.210673	0.211544
0.311305	0.290826	0.293184
0.16101	0.148539	0.145561
0.117822	0.114368	0.102878
0.3239	0.324675	0.312317
0.306515	0.295663	0.284685
0.341207	0.304094	0.291967
0.396768	0.386701	0.36069

Tabla 1: Desviación típica de los colores rojo, verde y azul para varias imágenes con barcos.

Características por valor RGB (imágenes de mar)		
Red	Green	Blue
0.0166632	0.030038	0.024195
0.167941	0.166051	0.159896
0.0226296	0.0247728	0.024626
0.013122	0.012959	0.0137109
0.123417	0.116666	0.11587
0.0210793	0.0194112	0.0220298
0.0658837	0.0578977	0.0569175
0.05896	0.0591794	0.0598308
0.0221889	0.0217769	0.0222725
0.0122218	0.0117652	0.00983029
0.010964	0.0111101	0.0113132
0.0246447	0.0247412	0.0240994
0.0217827	0.0213781	0.0220595
0.00686337	0.00719302	0.00750644
0.0238253	0.0225857	0.0226297
0.0110798	0.00971412	0.0109148
0.0198276	0.0185908	0.0177399
0.0320336	0.029889	0.0308864

Tabla 2: Desviación típica de los colores rojo, verde y azul para varias imágenes de mar.

4.1.2. Resultados

Para obtener los resultados de la aproximación hemos empleado los 4 modelos con diferentes valores en los parámetros de cada uno para después elegir los que ofrezcan mejores resultados. Con cada algoritmo hay 2 tablas, 1 con la media y desviación típica de la validación cruzada del algoritmo correspondiente con distintos valores de los parámetros y otra con las métricas resultantes de entrenar con todos los patrones y después testear el modelo usando los parámetros que dieron mejores resultados en la validación cruzada.

Primero se muestran en las tablas 3, 4 y 5 los resultados, las métricas del mejor y su matriz de confusión para el modelo KNN.

Modelo KNN										
Métrica / Valor K	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
Media	0.917	0.908	0.949	0.930	0.958	0.954	0.967	0.953	0.972	0.958
Desviación Típica	0.047	0.052	0.046	0.063	0.039	0.037	0.022	0.053	0.038	0.041

Tabla 3: Resultados para el modelo KNN

Mejor resultado							
k	Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
7	0.968	0.031	0.942	0.991	0.99	0.949	0.965

Tabla 4: Mejor valor de K para el modelo KNN con sus resultados

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	99	6
Negativos	1	113

Tabla 5: Matriz de confusión del modelo KNN

A continuación, se pueden ver los resultados (tabla 6), el mejor resultado junto con sus métricas (tabla 7) y la matriz de confusión (tabla 8) para el modelo Decision Tree.

Modelo Decision Tree										
Métrica / Depth	D=1	D=2	D=3	D=4	D=5	D=6	D=7	D=8	D=9	D=10
Media	0.932	0.935	0.912	0.953	0.935	0.940	0.917	0.931	0.940	0.945
Desviación Típica	0.048	0.044	0.066	0.044	0.049	0.048	0.066	0.045	0.056	0.041

Tabla 6: Resultados para el modelo Decision Tree

Mejor resultado							
D	Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
1	0.940	0.059	0.884	1.0	1.0	0.890	0.938

Tabla 7: Mejor valor de D para el modelo Decision Tree con sus resultados

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	100	13
Negativos	0	106

Tabla 8: Matriz de confusión del modelo Decision Tree

Para los modelos de RNA y SVM, dado el mayor número de combinaciones de valores para los parámetros, optamos por mostrar los valores que dieron mejores medias y desviación típica. Para RNA, se muestra el mejor resultado de cada capa de nivel 1 y para SVM, varios de cada tipo de kernel. Así, para la RNA, la 9 contiene los resultados del modelo, la 10 el mejor resultado de este y la 11 su matriz de confusión.

Modelo RNA										
Metrica/Topology	[1,6]	[2,2]	[3,6]	[4,10]	[5,5]	[6,7]	[7,2]	[8,3]	[9,5]	[10,7]
Media	0.919	0.934	0.931	0.920	0.926	0.930	0.940	0.940	0.927	0.927
Desviación Típica	0.039	0.050	0.039	0.070	0.079	0.064	0.038	0.057	0.070	0.059

Tabla 9: Resultados para el modelo RNA

Mejor resultado							
Topology	Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
[7, 2]	0.941	0.059	0.884	1.0	1.0	0.891	0.940

Tabla 10: Mejor topología para el modelo RNA con sus resultados

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	100	13
Negativos	0	106

Tabla 11: Matriz de confusión del modelo RNA

Para finalizar, se muestran las tablas 12, 13 y 14 para el modelo SVM.

Modelo SVM								
Métrica/(Kernel, C, gamma, degree)	(rbf, 2, 9)	(rbf, 8, 7)	(rbf, 9, 4)	(linear, 3)	(linear, 7)	(linear, 9)	(poly, 2, 6, 1)	(poly, 4, 6, 1)
Media	0.9407	0.9405	0.9407	0.9314	0.9271	0.9271	0.9314	0.9316
Desviación Típica	0.0430	0.0485	0.0431	0.0493	0.0487	0.0683	0.0444	0.0489

Tabla 12: Resultados para el modelo SVM

Mejor resultado								
(Kernel, gamma, degree)	C, de-	Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
(rbf, 2, 10)		0.9406	0.0593	0.8849	1.0	1.0	0.890	0.938

Tabla 13: Mejor configuración de parámetros para el modelo SVM con sus resultados

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	100	13
Negativos	0	106

Tabla 14: Matriz de confusión del modelo SVM

4.1.3. Discusión

La media en el mejor de los casos nunca baja del 90 % para ninguno de los modelos y, para KNN, en ninguno de sus valores (como se puede ver en las tablas 3, 6, 9 y 12). Luego los otros 3 modelos: DT, RNA y SVM, quedan empatados obteniendo peores mejores medias que KNN, todas con los mismos valores. Aún así, es en RNA donde se encuentran las mayores Desviaciones típicas, por lo que es el modelo que da los resultados menos estables.

KNN ha sido el que dio los mejores resultados, concretamente con el uso de $k=7$, con una F1 y precisión superior al 96 % (véase tabla 4). Así, queda como la mejor metodología a ser utilizada. Sin embargo, los otros 3 modelos (como detallan las tablas 7, 10 y 13) lo superan muy ligeramente en su Especificidad y VPP (aunque logrando un éxito del 100 % para ambos valores), pero esa ligera mejoría no es suficiente para compensar la bajada en Sensibilidad y VPN (en la mayoría de los casos) pues dan un resultado inferior al 90 % (mientras que KNN consigue mantenerse casi al 95 %), implicando excesiva dejadez de lograr encontrar verdaderos positivos a cambio de certeza de determinar verdaderos negativos.

Ésta es la tendencia general (apreciable en las tablas 5, 8, 11 y 14): que trate como barco cuando en verdad sea un barco, dando pocos FP (a cambio de un mayor número de FN). Esto es bueno cuando premia más garantizar que cuando sea crea que hay un barco verdaderamente haya un barco y sean pocas las veces que diga que hay barco cuando solo hay agua.

La precisión es, generalmente, ligeramente mayor que F1, lo cual implica una mayor inclinación a favorar VP y VN sobre FP y FN.

Aún con todo, las métricas generales son superiores al 90 %, rondando el 95 % en los mejores casos y una desviación típica inferior al 5 %. Esto implica que se ha conseguido que funcione la extracción de características correctamente y tampoco parece que usar otras características distintas vayan a dar resultados sustancialmente superiores, por lo menos para esta aproximación.

4.2. Aproximación 2

En esta nueva aproximación de nuestro problema de clasificación nos centraremos únicamente en expandir el dominio de imágenes sobre el que se clasifica, permitiendo una mayor variedad de elementos en estas, con el objetivo de no crear una red con "bias" que produzca falsos positivos ante la presencia de objetos en las imágenes que no encajen con la descripción de barco o mar. En estas nuevas imágenes podemos apreciar la presencia de muelles de barcos, puentes y estelas de barcos, como se puede apreciar en los ejemplos de la figura 4. En cuanto a la forma en la trabajamos con las imágenes, conservaremos la aplicada en la primera aproximación y lo mismo sucederá en cuanto a la extracción de características, ya que en esta aproximación nos centramos en observar como se ve afectada la red ante la presencia de una mayor diversidad de patrones de entrada y discutiremos como puede ser mejorada la capacidad de clasificación para aproximaciones posteriores.

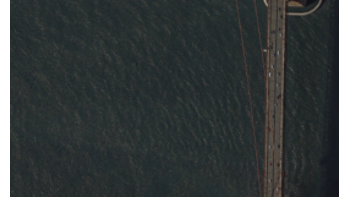
4.2.1. Descripción

La parte de preprocesado es la que más cambios sufre con respecto a la primera aproximación, ya que recortamos más imágenes de la base de datos de la que partíamos. Todas las nuevas imágenes son negativos, es decir, deben ser clasificadas como negativos, ya que no hay barcos en ellas, pero si existe una diferencia con respecto a los negativos que usábamos en la primera aproximación. Esta diferencia se encuentra en los elementos que se encuentran en estas, mientras que los anteriores negativos eran únicamente agua, lo que facilitaba distinguirlas de los barcos mediante la desviación estándar como comentamos anteriormente, ahora existirán imágenes con otros elementos, que también afectarán a la desviación estándar. Es este aspecto el que fomentará la variedad de patrones de entrada aunque la precisión de la clasificación también se verá afectada.

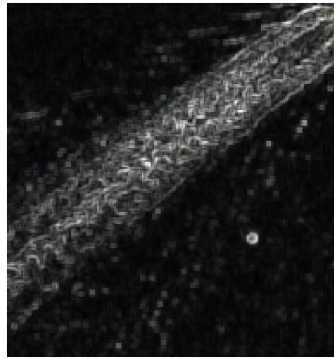
Para la extracción de características y para la normalización seguimos utilizando los mismos procedimientos que los comentados en la primera aproximación.



(a) Estela (sin barco)



(b) Muelle



(c) Imagen a filtrada por desviación típica

Figura 4: Ejemplos de imágenes con obstáculos

En esta aproximación volvemos a usar las entradas RGB que se usaron en la primera (Tablas 1 y 2).

4.2.2. Resultados

Para los resultados de esta aproximación se sigue el mismo formato que en la anterior. Primero se muestran en las tablas 15, 16 y 17 los resultados, las métricas del mejor y su matriz de confusión para el modelo KNN. Cabe mencionar que en general son más bajos debido al aumento de complejidad del problema.

Modelo KNN										
Métrica / Valor K	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
Media	0.819	0.822	0.867	0.852	0.859	0.852	0.863	0.841	0.867	0.841
Desviación Típica	0.061	0.051	0.068	0.078	0.042	0.054	0.045	0.039	0.060	0.080

Tabla 15: Resultados para el modelo KNN

Mejor resultado							
k	Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
9	0.892	0.107	0.844	0.922	0.87	0.906	0.857

Tabla 16: Mejor parámetro para el modelo KNN con sus resultados

Matriz de confusión		
Predicción / Real	Positivos	Negativos
Positivos	87	16
Negativos	13	155

Tabla 17: Matriz de confusión del modelo KNN

A continuación, se pueden ver los resultados (tabla 18), el mejor resultado junto con sus métricas (tabla 19) y la matriz de confusión (tabla 20) para el modelo Decision Tree.

Modelo Decision Tree										
Métrica / Depth	D=1	D=2	D=3	D=4	D=5	D=6	D=7	D=8	D=9	D=10
Media	0.852	0.866	0.844	0.837	0.841	0.819	0.822	0.834	0.822	0.796
Desviación Típica	0.055	0.080	0.038	0.078	0.065	0.105	0.069	0.065	0.049	0.068

Tabla 18: Resultados para el modelo Decision Tree

Mejor resultado							
d	Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
2	0.874	0.125	0.843	0.891	0.81	0.912	0.826

Tabla 19: Mejor valor de D para el modelo Decision Tree con sus resultados

Matriz de confusión		
Predicción / Real	Positivos	Negativos
Positivos	81	15
Negativos	19	156

Tabla 20: Matriz de confusión del modelo Decision Tree

El tercer modelo del cual se muestran los resultados es la RNA. Estos se pueden ver en las tablas 21, 22 y 23.

Modelo RNA										
Métrica / Topology	[1,8]	[2,9]	[3,8]	[4,10]	[5,8]	[6,10]	[7,10]	[8,10]	[9,10]	[10,9]
Media	0.834	0.848	0.839	0.848	0.840	0.850	0.848	0.853	0.850	0.853
Desviación Típica	0.032	0.086	0.065	0.051	0.091	0.061	0.065	0.053	0.065	0.066

Tabla 21: Resultados para el modelo RNA

Mejor resultado							
Topology	Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
[7, 10]	0.841	0.157	0.788	0.872	0.78	0.877	0.784

Tabla 22: Mejor topología para el modelo RNA con sus resultados

Matriz de confusión		
Predicción / Real	Positivos	Negativos
Positivos	78	21
Negativos	22	150

Tabla 23: Matriz de confusión del modelo RNA

Para finalizar, se muestran las tablas 24, 25 y 26 para el modelo SVM, con el mismo formato utilizado para el anterior modelo.

Modelo SVM								
Métrica / (Kernel, C, gamma, degree)	(rbf, 2, 9)	(rbf, 8, 7)	(rbf, 9, 4)	(linear, 3)	(linear, 7)	(linear, 9)	(poly, 2, 6, 1)	(poly, 4, 6, 1)
Media	0.8519	0.8564	0.8412	0.8486	0.8559	0.8484	0.8486	0.849
Desviación Típica	0.0632	0.0638	0.0655	0.0444	0.0791	0.09005	0.0789	0.0572

Tabla 24: Resultados para el modelo SVM

Mejor resultado							
(Kernel, C, gamma, degree)	Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
(rbf, 3, 9)	0.859	0.140	0.878	0.851	0.72	0.941	0.791

Tabla 25: Mejor combinación de parámetros para el modelo SVM con sus resultados

Matriz de confusión		
Predicción / Real	Positivos	Negativos
Positivos	72	10
Negativos	28	161

Tabla 26: Matriz de confusión del modelo SVM

4.2.3. Discusión

KNN, el que daba los mejores resultados en la primera iteración, pasa a no obtener ninguna media superior o igual al 90 % (como muestra la tabla 15). Esto mismo es lo que pasa con todos los demás modelos y en esas se acentúa más (véase tablas 18, 21 y 24). Así, la peor de las mejores medias la proporciona RNA, que produce 5 de los peores 7 resultados.

Cabe destacar que aunque RNA obtuviese la mayor cantidad de peores resultados, en comparación con DT y SVM (con los que estaba empatado en la aproximación anterior) es el que obtiene mayor Especificidad y VPP (como muestra la tabla 22), aunque son peores que los de KNN.

KNN obtiene la mejor precisión, error y F1, mientras que DT mejor Especificidad y VPP y SVM mejor sensibilidad y VPN, mostrando cuál sería más preferible en caso de preferir resultados menos equilibrados por la elección de priorizar VP o VN.

Los resultados generales siguen favoreciendo los mismos resultados, a excepción del VPN, que se vuelve el valor superior por encima de Sensibilidad, Especificidad y VPP en todos los casos (como detallan las tablas 16, 19, 22 y 25).

Finalmente, KNN se mantiene como el mejor (como también se puede observar en las tablas 17, 20, 23 y 26) aunque con un cambio de k , ya que ahora pasa a valer 9, con una desviación típica del 6 %. Los valores son por lo general peores, pero se mantiene su orden y es normal debido a la ampliación del problema, aunque tanto F1 como la precisión son de las bajadas más significativas a la hora de determinar la validez de los resultados.

4.3. Aproximación 3

Con esta aproximación nos centramos en mejorar la capacidad de clasificación de la RNA extrayendo nuevas características de diferentes secciones de las imágenes usadas en el entrenamiento.

4.3.1. Descripción

Para esta aproximación no realizaremos ninguna modificación en la parte de preprocesado con respecto a la anterior aproximación, es decir, conservaremos los patrones de la segunda aproximación.

En cuanto a la extracción de características, decidimos extraer dos características a mayores, con el objetivo de mejorar la capacidad de clasificación de nuestra RNA. En concreto, extrajimos la media de los valores red, green y blue para la porción central de la imagen y la diferencia entre esta última y la media de los valores RGB de todas las esquinas. Ejemplos de patrones con la extracción de estas nuevas características junto con las de la aproximación anterior se pueden ver en la tabla 27 (patrones de barcos) y tabla 28 (patrones de no barcos). Para seleccionar la parte central y las esquinas de la imagen, dividimos tanto el largo como el ancho de la imagen en 5 porciones iguales. Decidimos usar esta nueva característica para mejorar la RNA, ya que podría ayudar a detectar barcos en la parte central de la imagen, cuya media RGB sería distinta de la de las esquinas, en las que se encontraría el mar, puertos...

Seguiremos usando el mismo tipo de normalización (mín-máx) que en la segunda aproximación, pero es importante apuntar que también normalizamos las dos nuevas características con el mismo tipo de normalización.

Para esta aproximación, aparte de los valores RGB de la imagen total, incluimos la media de los valores rojo verde y azul en la porción central de la imagen dividiéndola en 5x5 (2,2 a 3,3) y esta misma media menos la de las cuatro esquinas de la imagen, dividiéndola de nuevo en 5x5.

Características extraídas a partir de valores RGB de imágenes de barcos								
STD imagen entera			Media porción central			Porción central - esquinas		
Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
0.353772	0.331426	0.320017	0.915432	0.887089	0.838987	0.674861	0.609668	0.588166
0.284936	0.283531	0.251523	0.802919	0.819935	0.811438	0.649405	0.650371	0.564
0.290564	0.292253	0.262237	0.422745	0.438824	0.493333	0.214587	0.207635	0.18663
0.218705	0.210569	0.217075	0.428431	0.465359	0.482353	0.202219	0.200599	0.240632

Tabla 27: Distintas características extraídas de las imágenes de barcos

Características extraídas a partir de valores RGB de imágenes de mar, puertos,...								
STD imagen entera			Media porción central			Porción central - esquinas		
Red	Green	Blue	Red	Green	Blue	Red	Green	Blue
0.072607	0.070583	0.062867	0.24678	0.25051	0.2391	0.043252	0.040444	0.025385
0.077898	0.071771	0.06564	0.28193	0.27693	0.22577	0.019034	0.012246	0.01796
0.21264	0.18995	0.1731	0.21493	0.29253	0.2728	0.13082	0.11851	0.1155
0.12791	0.12137	0.12081	0.29045	0.28523	0.25434	0.012733	0.012544	0.02444

Tabla 28: Distintas características extraídas de las imágenes de mar, puertos, costa...

4.3.2. Resultados

En los resultados de esta aproximación se sigue el mismo formato que el las anteriores. En esta aproximación se mejoran los resultados comparados con la segunda aproximación. Primero se muestran en las tablas 29, 30 y 31 los resultados, las métricas del mejor y su matriz de confusión para el modelo KNN.

Modelo KNN										
Métrica / Valor K	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
Media	0.900	0.870	0.911	0.882	0.922	0.901	0.915	0.896	0.897	0.889
Desviación Típica	0.052	0.058	0.053	0.073	0.061	0.048	0.065	0.067	0.062	0.050

Tabla 29: Resultados para el modelo KNN

Mejor resultado							
k	Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
5	0.934	0.066	0.918	0.942	0.9	0.953	0.909

Tabla 30: Mejor parámetro para el modelo KNN con sus resultados

Matriz de confusión		
Predicción / Real	Positivos	Negativos
Positivos	90	8
Negativos	10	163

Tabla 31: Matriz de confusión del modelo KNN

A continuación, se pueden ver los resultados (tabla 32), el mejor resultado junto con sus métricas (tabla 33) y la matriz de confusión (tabla 34) para el modelo Decision Tree.

Modelo Decision Tree										
Métrica / Depth	D=1	D=2	D=3	D=4	D=5	D=6	D=7	D=8	D=9	D=10
Media	0.871	0.885	0.900	0.885	0.889	0.874	0.878	0.886	0.881	0.878
Desviación Típica	0.044	0.079	0.058	0.045	0.054	0.050	0.041	0.079	0.038	0.053

Tabla 32: Resultados para el modelo Decision Tree

Mejor resultado							
D	Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
6	1.0	0.0	1.0	1.0	1.0	1.0	1.0

Tabla 33: Mejor valor de D para el modelo Decision Tree con sus resultados

Matriz de confusión		
Predicción / Real	Positivos	Negativos
Positivos	100	0
Negativos	0	171

Tabla 34: Matriz de confusión del modelo Decision Tree

El tercer modelo del cual se muestran los resultados es la RNA. Estos se pueden ver en las tablas 35, 36 y 37.

Modelo RNA										
Métrica / Topology	[3]	[5]	[2]	[3,8]	[3,4]	[6,1]	[4]	[4,1]	[3,2]	[5,1]
Media	0.932	0.924	0.936	0.911	0.917	0.915	0.933	0.920	0.925	0.909
Desviación Típica	0.055	0.075	0.040	0.043	0.052	0.061	0.043	0.033	0.035	0.053

Tabla 35: Resultados para el modelo RNA

Mejor resultado							
Topology	Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
[3]	0.900	0.100	0.797	0.986	0.980	0.854	0.879

Tabla 36: Mejor topología para el modelo RNA con sus resultados

Matriz de confusión		
Predicción / Real	Positivos	Negativos
Positivos	94	6
Negativos	6	165

Tabla 37: Matriz de confusión del modelo RNA

Para finalizar, se muestran las tablas 38, 39 y 40 para el modelo SVM, con el mismo formato utilizado para los anteriores modelos.

Modelo SVM								
Métrica / (Kernel, C, gamma, degree)	(poly, 1, 4, 6)	(poly, 10, 6, 2)	(poly, 10, 4, 3)	(rbf, 5, 6)	(rbf, 9, 2)	(rbf, 8, 1)	(linear, 10)	(linear, 3)
Media	0.952	0.944	0.952	0.926	0.930	0.945	0.893	0.886
Desviación Típica	0.046	0.040	0.035	0.060	0.037	0.044	0.044	0.037

Tabla 38: Resultados para el modelo SVM

Mejor resultado							
(Kernel, C, gamma, degree)	Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
(poly, 1, 4, 6)	0.993	0.007	1.0	0.988	0.98	1.0	0.990

Tabla 39: Mejores parámetros para el modelo SVM con sus resultados

Matriz de confusión		
Predicción / Real	Positivos	Negativos
Positivos	98	0
Negativos	2	171

Tabla 40: Matriz de confusión del modelo SVM

4.3.3. Discusión

En esta aproximación, los resultados han sido considerablemente mejores. En el caso de KNN, su peor media resulta superior a la mejor de la anterior aproximación (como se puede ver comparando las tablas 29, 30 y 31 con sus contrapartes de la anterior aproximación) y sorprende RNA que consigue mejores resultados sin bajar del 0.909 de media, como se muestra en la tabla 35. Aún así, como muestra la tabla 36, presenta un error de 0.1, y en general sigue presentando en su mejor resultado, el peor de los mejores resultados para cada modelo, aunque la tabla 37, muestre un equilibrio entre los VP y VN.

En las tablas 33 y 34 se puede apreciar que el mejor resultado es el Decision Tree (aunque la tabla 32 muestra que este último cuenta con una media de 0.9 en su mejor caso, y que no fue con el valor con el que sacó los mejores resultados), logrando resultados perfectos, los mejores resultados hasta el momento.

Como muestra la tabla 39, SVM consigue valores casi perfectos, que podrían fácilmente haber sido los mejores de no ser por el gran resultado de DT. La matriz de confusión en la tabla 40 muestra que falla solo 2 de sus predicciones, pues con mejor Sensibilidad y VPN, obtiene 2 falsos negativos.

En conclusión, DT consigue los mejores resultados hasta el momento obteniendo resultados perfectos, seguido de cerca por SVM. RNA consiguió mejorar sobre la anterior aproximación y KNN dejó de dar los mejores resultados, pero con una mejora sobre la anterior aproximación. De este modo, se puede considerar que la extracción de características ha sido un éxito rotundo, saliendo resultados perfectos y especialmente porque la mejor metodología es la que explica más claramente su razonamiento.

4.4. Aproximación 4

En las anteriores aproximaciones partíamos siempre de imágenes en formato RGB para extraer características, lo que limitaba el número de características significantes que podíamos obtener. Sin embargo, para esta cuarta aproximación hemos convertido las imágenes a formato HSV, que cuenta con tres nuevos valores: saturation, hue y value. Usaremos estas variables para extraer nuevas características en nuestros patrones.

4.4.1. Descripción

Para esta aproximación no realizaremos ninguna modificación en la parte de preprocesado con respecto a la anterior aproximación, es decir, conservaremos los patrones de la segunda y tercera aproximación.

Como se comentó en la introducción, en esta aproximación disponemos de 3 nuevos valores para cada patrón. Con lo cual las características serán las mismas que en la aproximación 3, pero extraídas de los 3 valores HSV antes mencionados en lugar de los valores RGB, de las anteriores aproximaciones. Decidimos añadir estas características, ya que proporcionan una nueva perspectiva desde la que analizar cada patrón y esto puede ayudar a afianzar la precisión de la clasificación de nuestra RNA. En la tabla 41 podemos ver algunos ejemplos de características extraídas de imágenes de barcos (patrones positivos), mientras que las de la tabla 42 están extraídas de imágenes de mar (patrones negativos).

En cuanto a la normalización de los datos, seguiremos usando la normalización mín-máx como en todas nuestras anteriores aproximaciones, pero también se la aplicaremos a las nuevas características.

Características extraídas a partir de valores de valores HSV de imágenes de barcos								
STD imagen entera			Media porción central			Porción central - esquinas		
Hue	Satura.	Value	Hue	Satura.	Value	Hue	Satura.	Value
78.957	0.26085	0.33682	38.322	0.09208	0.91543	91.1385	0.23937	0.63445
65.505	0.24262	0.25634	138.199	0.03701	0.82251	87.8222	0.4452	0.57421
28.951	0.28152	0.26594	219.66	0.32819	0.49392	0.79356	0.09030	0.18683
26.896	0.12257	0.21244	168.435	0.16389	0.48824	24.6447	0.02217	0.22316

Tabla 41: Distintas características extraídas de las imágenes de barcos

Características extraídas a partir de valores de valores HSV de imágenes de barcos								
STD imagen entera			Media porción central			Porción central - esquinas		
Hue	Satura.	Value	Hue	Satura.	Value	Hue	Satura.	Value
57.5823	0.063753	0.071737	87.3505	0.055125	0.252259	8.74757	0.041185	0.041968
32.7636	0.046645	0.073198	55.4456	0.205974	0.283972	39.6969	5.986e-5	0.006818
44.4868	0.096982	0.19197	164.276	0.266149	0.292669	13.2487	0.037904	0.118607
46.0932	0.072895	0.127336	60.168	0.12876	0.292742	21.7489	0.012925	0.016665

Tabla 42: Distintas características extraídas de las imágenes de barcos

4.4.2. Resultados

Se usa el mismo formato utilizado para las aproximaciones anteriores al mostrar los resultados. Primero se muestran en las tablas 43, 44 y 45 los resultados, las métricas del mejor y su matriz de confusión para el modelo KNN. Observamos otra ligera mejoría en estos con respecto a los de las aprox. anteriores.

Modelo KNN										
Métrica / Valor K	k=1	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10
Media	0.952	0.933	0.922	0.937	0.933	0.937	0.944	0.933	0.926	0.93
Desviación Típica	0.046	0.041	0.044	0.042	0.054	0.049	0.050	0.051	0.046	0.047

Tabla 43: Resultados para el modelo KNN

Mejor resultado								
k	Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1	
1	1.0	0.0	1.0	1.0	1.0	1.0	1.0	

Tabla 44: Mejor valor de K para el modelo KNN con sus resultados

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	100	0
Negativos	0	171

Tabla 45: Matriz de confusión del modelo KNN

A continuación, se pueden ver los resultados (tabla 46), el mejor resultado junto con sus métricas (tabla 47) y la matriz de confusión (tabla 48) para el modelo Decision Tree.

Modelo Decision Tree										
Métrica / Depth	D=1	D=2	D=3	D=4	D=5	D=6	D=7	D=8	D=9	D=10
Media	0.885	0.900	0.892	0.896	0.874	0.918	0.892	0.897	0.863	0.878
Desviación Típica	0.036	0.074	0.064	0.059	0.087	0.038	0.053	0.060	0.072	0.090

Tabla 46: Resultados para el modelo Decision Tree

Mejor resultado							
D	Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
6	0.996	0.003	0.990	1.0	1.0	0.994	0.995

Tabla 47: Mejor valor de D para el modelo Decision Tree con sus resultados

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	100	1
Negativos	0	170

Tabla 48: Matriz de confusión del modelo Decision Tree

El tercer modelo del cual se muestran los resultados es la RNA. Estos se pueden ver en las tablas 49, 50 y 51.

Modelo RNA								
Métrica/Topology	[1]	[1, 1]	[2]	[3]	[4]	[5]	[6]	[7]
Media	0.929	0.933	0.935	0.927	0.938	0.931	0.935	0.935
Desviación Típica	0.051	0.040	0.038	0.063	0.033	0.049	0.045	0.037

Tabla 49: Resultados para el modelo RNA

Mejor resultado							
Topology	Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
[2]	0.956	0.044	0.949	0.960	0.93	0.971	0.939

Tabla 50: Mejor topología para el modelo RNA con sus resultados

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	93	5
Negativos	7	166

Tabla 51: Matriz de confusión del modelo RNA

Para finalizar, se muestran las tablas 52, 53 y 54 para el modelo SVM, con el mismo formato utilizado para los anteriores modelos.

Modelo SVM								
Métrica/(Kernel, C, gamma, degree)	(rbf, 2, 9)	(rbf, 4, 3)	(rbf, 9, 4)	(linear, 3)	(linear, 7)	(linear, 9)	(poly, 2, 6, 1)	(poly, 4, 6, 1)
Media	0.945	0.945	0.956	0.927	0.937	0.937	0.930	0.923
Desviación Típica	0.040	0.036	0.034	0.045	0.046	0.058	0.048	0.061

Tabla 52: Resultados para el modelo SVM

Mejor resultado							
(Kernel, C, gamma, degree)	Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
(rbf,4,3)	0.9852	0.0147	0.9897	0.9826	0.97	0.9941	0.9797

Tabla 53: Mejor configuración de parámetros para el modelo SVM con sus resultados

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	97	1
Negativos	3	170

Tabla 54: Matriz de confusión del modelo SVM

4.4.3. Discusión

En esta aproximación vuelve a ser KNN el modelo que da los mejores valores. En su mejor media presenta resultados perfectos como se puede ver en las tablas 44 y 45, que es cuando utiliza $k=1$.

Las medias de DT son las más pobres, ya que es el único que presenta mayores desviaciones típicas generales (aún inferior al 0.1) y medias inferiores al 0.9, como se puede apreciar en la tabla 46, comparada con las tablas de los otros 3: 43, 49 y 52. Sin embargo, como aparece en la tabla 47, obtiene la segunda posición cuando consideramos su mejor resultado, con Especificidad y VPP perfecta y el resto de valores rondando también ese valor siendo superiores al 0.99 (una muy ligera bajada respecto a la anterior aproximación). Esto se puede comprobar con su matriz observable en la tabla 48, con solo un error.

RNA queda en la última posición (como se puede ver comparando la tabla 51 de RNA, con 12 fallos en total, con la 54 de SVM, la penúltima que solo tiene 4 fallos en total), pero aún así con valores bastante altos, siendo el más bajo para sus mejores métricas el VPP con un valor de 0.93 (como muestran las tablas 50), aunque SVM consigue como peor también VPP con un valor de 0.97.

En conclusión, los resultados han sido muy buenos. Aunque tanto DT y SVM hayan sido rápidamente superados en sus mejores resultados comparados con la aproximación anterior, la bajada en valores ha sido mínima y han sido superados por un modelo que obtuvo resultados absolutamente perfectos, lo cual es un gran avance en general e indicativo de la buena extracción de características.

4.5. Aproximación 5

En esta última aproximación nos dedicaremos a investigar como se altera la capacidad de detección de nuestro modelo al usar deep learning. Como ya mencionamos en la introducción, a diferencia de los modelos clásicos, el deep learning permite procesar los datos con un enfoque no lineal, lo que ayuda a consolidar un aprendizaje profundo. En concreto haremos uso de redes convolucionales (RNC) ya que son muy útiles en la detección y categorización de objetos y en la clasificación y segmentación de imágenes, características que trataremos de aprovechar para aumentar la capacidad de clasificación de nuestro modelo.

4.5.1. Descripción

En esta aproximación, pese a que no añadiremos nuevas imágenes al conjunto de entradas con respecto a la anterior aproximación, sí que modificaremos los patrones de entrada, reescalando las imágenes de entrada para que ocupen 28x28 píxeles. Esto se hace ya que

en las redes convolucionales se asocia una neurona de entrada a cada píxel de la imagen y canal (en nuestro caso el canal es RGB).

Con respecto a la extracción de características, dado que usamos una RNC, no será necesario extraerlas manualmente, porque este tipo de redes integran una fase de extracción de características compuesta por neuronas convolucionales y reducción de muestreo.

Y por último, con respecto a la normalización, utilizaremos la mín-máx y tan solo será necesario llevarla a cabo en los valores RGB de cada píxel, que más tarde se pasarán a la red convolucional como entrada. No será necesario llevarla a cabo en la extracción de características, ya que esta no se lleva a cabo.

4.5.2. Resultados

Esta vez, obtenemos los resultados de la aproximación de redes de neuronas convolucionales con 4 capas, las 3 primeras de convolución y otra dense. Entre cada una hay otras de MeanPool o MaxPool y además, antes de la dense, hay una capa de reshape. En las tablas de resultados (tablas 55, 58, 61, 64, 67, 70, 73, 76, 79 y 82), indicamos tanto el número de entradas como de salidas de cada capa y si entre cada capa hay una de MaxPool o de MeanPool, además de la media y desviación típica de cada topología. También incluimos los valores de las métricas para cada topología (tablas 56, 59, 62, 65, 68, 71, 74, 77, 80 y 83) como en aproximaciones anteriores, centrándonos en concreto en la precisión (la mejor topología en este aspecto es la de la tabla 82), y la matriz de confusión (tablas 57, 60, 63, 66, 69, 72, 75, 78, 81 y 84).

Topología 1							
Tipo de capa	Conv.	Conv.	Conv.	Dense	Pool	Media	STD
(entradas, salidas)	(3,16)	(16,32)	(32,32)	(288, 1)	MaxPool	0.874	0.082

Tabla 55: Resultados para topología 1

Métricas						
Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
0.907	0.092	0.921	0.901	0.82	0.959	0.867

Tabla 56: Métricas de la topología 1

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	82	18
Negativos	7	164

Tabla 57: Matriz de confusión de la topología 1

Topología 2							
Tipo de capa	Conv.	Conv.	Conv.	Dense	Pool	Media	STD
(entradas, salidas)	(3,4)	(4,8)	(8,8)	(72, 1)	MaxPool	0.919	0.066

Tabla 58: Resultados para topología 2

Métricas						
Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
0.966	0.033	0.950	0.976	0.96	0.97	0.955

Tabla 59: Métricas de la topología 2

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	96	4
Negativos	5	166

Tabla 60: Matriz de confusión de la topología 2

Topología 3							
Tipo de capa	Conv.	Conv.	Conv.	Dense	Pool	Media	STD
(entradas, salidas)	(3,4)	(4,8)	(8,16)	(144, 1)	MaxPool	0.904	0.114

Tabla 61: Resultados para topología 3

Métricas						
Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
0.981	0.018	0.970	0.988	0.98	0.982	0.975

Tabla 62: Métricas de la topología 3

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	98	2
Negativos	3	168

Tabla 63: Matriz de confusión de la topología 3

Topología 4							
Tipo de capa	Conv.	Conv.	Conv.	Dense	Pool	Media	STD
(entradas, salidas)	(3,8)	(8,16)	(16,32)	(288, 1)	MaxPool	0.866	0.048

Tabla 64: Resultados para topología 4

Métricas						
Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
0.885	0.114	0.794	0.954	0.93	0.859	0.857

Tabla 65: Métricas de la topología 4

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	93	7
Negativos	24	147

Tabla 66: Matriz de confusión de la topología 4

Topología 5							
Tipo de capa	Conv.	Conv.	Conv.	Dense	Pool	Media	STD
(entradas, salidas)	(3,8)	(8,16)	(16,16)	(144, 1)	MaxPool	0.923	0.037

Tabla 67: Resultados para topología 5

Métricas						
Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
0.985	0.014	0.98	0.988	0.98	0.988	0.98

Tabla 68: Métricas de la topología 5

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	98	2
Negativos	2	169

Tabla 69: Matriz de confusión de la topología 5

Topología 6							
Tipo de capa	Conv.	Conv.	Conv.	Dense	Pool	Media	STD
(entradas, salidas)	(3,16)	(16,32)	(32,32)	(288, 1)	MeanPool	0.872	0.092

Tabla 70: Resultados para topología 6

Métricas						
Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
0.915	0.084	0.859	0.951	0.92	0.912	0.888

Tabla 71: Métricas de la topología 6

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	92	8
Negativos	15	156

Tabla 72: Matriz de confusión de la topología 6

Topología 7							
Tipo de capa	Conv.	Conv.	Conv.	Dense	Pool	Media	STD
(entradas, salidas)	(3,4)	(4,8)	(8,16)	(144, 1)	MeanPool	0.908	0.038

Tabla 73: Resultados para topología 7

Métricas						
Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
0.937	0.062	0.927	0.942	0.9	0.959	0.913

Tabla 74: Métricas de la topología 7

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	90	10
Negativos	7	164

Tabla 75: Matriz de confusión de la topología 7

Topología 8							
Tipo de capa	Conv.	Conv.	Conv.	Dense	Pool	Media	STD
(entradas, salidas)	(3,8)	(8,16)	(16,32)	(288, 1)	MeanPool	0.935	0.041

Tabla 76: Resultados para topología 8

Métricas						
Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
0.944	0.055	0.947	0.943	0.9	0.970	0.923

Tabla 77: Métricas de la topología 8

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	90	10
Negativos	5	160

Tabla 78: Matriz de confusión de la topología 8

Topología 9							
Tipo de capa	Conv.	Conv.	Conv.	Dense	Pool	Media	STD
(entradas, salidas)	(3,8)	(8,16)	(16,16)	(144, 1)	MeanPool	0.932	0.029

Tabla 79: Resultados para topología 9

Métricas						
Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
0.977	0.022	0.979	0.976	0.96	0.988	0.969

Tabla 80: Métricas de la topología 9

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	96	4
Negativos	2	169

Tabla 81: Matriz de confusión de la topología 9

Topología 10							
Tipo de capa	Conv.	Conv.	Conv.	Dense	Pool	Media	STD
(entradas, salidas)	(3,4)	(4,8)	(8,8)	(72, 1)	MeanPool	0.961	0.024

Tabla 82: Resultados para topología 10

Métricas						
Precisión	Error	Sensibilidad	Especificidad	VPP	VPN	F1
0.974	0.025	0.960	0.982	0.97	0.976	0.965

Tabla 83: Métricas de la topología 10

Matriz de confusión		
Predicción/Real	Positivos	Negativos
Positivos	97	3
Negativos	4	167

Tabla 84: Matriz de confusión de la topología 10

4.5.3. Discusión

Los mejores resultados (media (0.961) y STD (0.024)) se obtuvieron usando MeanPool (como muestra la tabla 82), mientras que los peores fueron con MaxPool (media en topología 4 (tabla 64) y STD superior a 0.1 en topología 3 (tabla 61)). Sin embargo, no hay diferencias notables entre el uso de MaxPool o MeanPool, como se puede ver comparando el mejor resultado de MeanPool (tabla 67) con el peor de MaxPool (tabla 70).

En cuanto a las mejores métricas, destaca la topología 5 (tabla 68), que obtiene los mejores valores globales en todas las métricas, nunca inferiores al 98 % y con un error inferior al 0.015 (con un empate en Especificidad y VPP con otra que también usa MeanPool, la topología 3 (tabla 62), y en VPP con una que usa MaxPool, la 9 (tabla 80)).

El segundo lugar lo ocuparía la topología 3, superada en Sensibilidad y VPN por la 9. Aún así, la 9 no obtiene los mejores resultados en todo con respecto a las que usan MaxPool, pues pierde tanto en Especificidad como en VPP contra la topología 10 (tabla 83).

A pesar de esto, es MeanPool la que obtiene los peores valores. La topología 4 (tabla 65) obtiene los peores valores en Precisión, Error, Sensibilidad, VPN y F1, mientras que los peores valores en Especificidad y VPP los proporciona la topología 1 (tabla 56). Cabe destacar error de 0.114, y Sensibilidad por debajo de 0.8, junto a los 24 FN (y 32 fallos en total) que muestra la tabla 66, valores muy pobres en general y sobre todo al compararlos con los buenos resultados generales, en especial con los mejores de todos que son casi perfectos.

En conclusión, los resultados han sido bastante buenos. Quizás les hacen sombra los mejores modelos de las últimas 2 aproximaciones, pero eso fue porque estás 2 obtuvieron modelos que devolvieron resultados perfectos. Los resultados en general al compararlos con los de la primera aproximación sí que sobresalen y son muy superiores a los de la segunda aproximación, por lo que los peores valores, aún al destacar, son más una excepción, por lo que podemos concluir que el objetivo de esta aproximación ha sido cumplido con éxito.

5. Conclusiones

Después de haber realizado estas 5 aproximaciones incrementales y tras haber visto los resultados que nos ofrecen las distintas arquitecturas al haber incluido distintas características, podemos concluir que los resultados son ciertamente satisfactorios, ya que podemos alcanzar precisiones a la hora de clasificar mucho más altas que el 50 % (decisión aleatoria), lo que nos garantiza que nuestro modelo tiene un cierto grado de certeza en la clasificación.

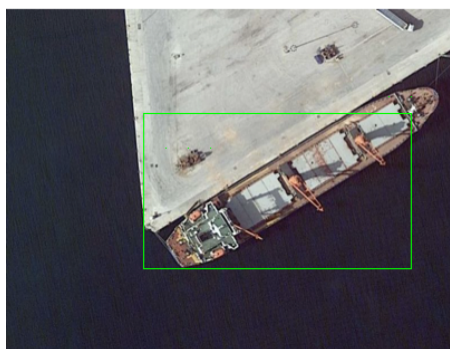
A la hora de aplicar nuestro sistema a situaciones reales, no podemos tener la certeza de que nos ofrezca unos resultados tan buenos como en el entorno en el que la hemos probado, ya que la base de datos que hemos usado es relativamente pequeña y muy posiblemente tenga bias en diferentes aspectos que se quedan fuera de nuestro alcance, por ejemplo, como mencionamos al principio, todas las imágenes fueron tomadas en momentos del día en los que hay buenas situaciones de luz. Además estas fotos nunca están obstruidas por fenómenos meteorológicos etc. En definitiva, este bias podría limitar la capacidad de clasificación de nuestro modelo en situaciones reales. Debido a esto, en caso de que quisiésemos usarla en entornos reales, deberíamos someter nuestro sistema a pruebas intensivas en la que se prueben diferentes aspectos en situaciones reales para poder afirmar que nuestro modelo es lo suficientemente buena para ser usada comercialmente.

Con respecto a la variedad de arquitecturas que hemos probado en las aproximaciones, la mejor, sin duda ha sido KNN, que ofreció los mejores resultados en casi todas las aproximaciones, alcanzando unos resultados perfectos en la cuarta aproximación para el valor de $k = 1$. Es necesario mencionar la tercera aproximación, en la que la mejor arquitectura fue DT, logrando también unos resultados perfectos. Con respecto a la última aproximación, algunas de las arquitecturas que probamos para las redes convolucionales consiguen resultados satisfactorios, alcanzando precisiones del 97.4 %, con una media de 96.1 % y una desviación típica de 0.024. Por último mencionar SVM, que oscila el tercer puesto en cuanto a rendimiento y RNA, que es consistentemente la que peor rinde, exceptuando la primera aproximación.

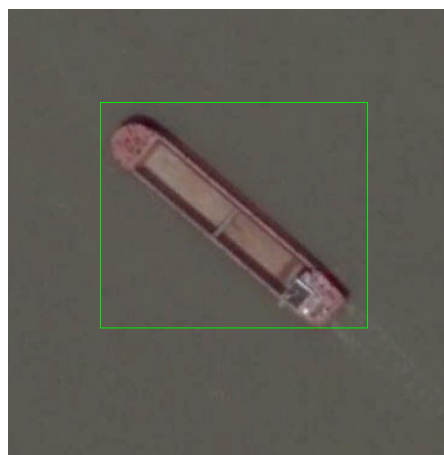
Con respecto a la realización de este proyecto, los aspectos más difíciles han sido el hecho de familiarizarnos con el entorno de desarrollo (Julia) y con el funcionamiento de las distintas arquitecturas a la hora de implementarlas y depurar los errores, así como el hecho de razonar las distintas extracciones de características a medida que avanzaban las aproximaciones, ya que las primeras fueron más sencillas e intuitivas, pero en las últimas era más complicado pensar en una característica que pudiese ayudar a mejorar nuestra capacidad de clasificación. Debatimos diversos sistemas, como el uso de centroides o el análisis de componentes principales.

A continuación en la figura 5 podemos ver algunos ejemplos de fotos en las que hemos puesto a prueba al mejor modelo de la aproximación 4, KNN con parámetro $k=1$. Para hacer esta detección de barcos en fotos hemos creado un script¹ en Julia que a partir de un modelo de los que hemos usado en las aproximaciones (RNA, SVM, DT o KNN). Este script consiste en ir haciendo convoluciones de la imagen objetivo, es decir, dividiendo toda la imagen en porciones más pequeñas, de tamaño x por y , y clasificando esas imágenes más pequeñas, en barco o no barco, con el modelo especificado; para luego recuadrar el bounding box de las regiones del espacio donde ha habido más positivos. De esta forma nos quedan las imágenes recuadradas en verde de la figura 5.

En la figura, se puede ver que el modelo da buenos resultados como en la imagen b o en la a que detecta los barcos sin problema aún habiendo puerto. Sin embargo, las imágenes c y d nos muestran que a pesar de los altos porcentajes de clasificación en los resultados de la aproximación 4, uno de los mejores modelos que conseguimos (KNN con $k=1$) no es totalmente capaz de distinguir los barcos de puertos o costas donde hay gran variedad de colores por haber diferentes objetos. Esto es una señal de que aún habiendo avanzado bastante en el campo de detección de barcos, aún queda trabajo para el futuro.

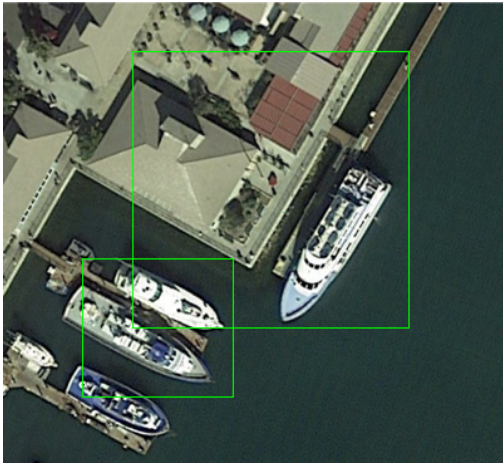


(a) Barco grande atracado en puerto



(b) Barco grande (solo mar)

¹El script que se adjunta con el código de las aproximaciones junto con un README.txt en el que se explica su uso



(c) Varios barcos atracados en puerto



(d) Barco cerca de la playa

Figura 5: Barcos detectados mediante el modelo KNN con características de la aproximación 4

6. Trabajo futuro

Este sistema puede tener varios usos en el presente y futuro. Uno de ellos puede ser en el ámbito de la defensa, para detectar y poder seguir el rumbo de navíos enemigos y así poder planear estrategias con la nueva información en situaciones de guerra o conflicto marítimo.

Otro uso más pacífico podría ser controlar el aforo en puertos o embarcaderos. Para esto se podrían utilizar imágenes sacadas desde drones o dispositivos similares para observar desde el cielo las embarcaciones que entran y salen de estas instalaciones. Un último uso en el que este sistema puede ser útil es en el ámbito de la pesca ilegal o la observación de zonas de pesca protegida para evitar que entren barcos no autorizados a ellas.

Es importante comentar que, para sacar el sistema al mundo real, habría que probarlo antes de forma intensiva con problemas reales, ya que este solo ha sido probado en un ambiente controlado y con una base de datos relativamente pequeña. Es posible que, al probar el sistema con un conjunto de imágenes mayor y con interferencias, el sistema no funcione como se esperaba. En el caso de que esto sucediese, habría que intentar solucionar los problemas contemplando nuevas características o nuevos modelos.

En cuanto a las mejoras de este sistema, se podría mejorar para detectar barcos en imágenes en los que aparezcan incompletos o en imágenes que no tengan tan buena visibilidad como las utilizadas en este proyecto. Otro camino que se podría tomar sería diferenciar los barcos detectados según su tipo (cruceros, veleros, portacontenedores, etc.), ya que esto también podría tener muchos usos en el mundo real.

Referencias

- Chang, Y.-L., Anagaw, A., Chang, L., Wang, Y. C., Hsiao, C.-Y. & Lee, W.-H. (2019). Ship Detection Based on YOLOv2 for SAR Imagery. *Remote Sensing*, 11(7). <https://doi.org/10.3390/rs11070786>
- Corbane, C., Najman, L., Pecoul, E., Demagistri, L. & Petit, M. (2010). A complete processing chain for ship detection using optical satellite imagery. *International Journal of Remote Sensing*, 31(22), 5837-5854. <https://doi.org/10.1080/01431161.2010.512310>
- Geronimo, R. C., Franklin, E. C., Brainard, R. E., Elvidge, C. D., Santos, M. D., Venegas, R. & Mora, C. (2018). Mapping Fishing Activities and Suitable Fishing Grounds Using Nighttime Satellite Images and Maximum Entropy Modelling. *Remote Sensing*, 10(10). <https://doi.org/10.3390/rs10101604>
- Kartal, M. & Duman, O. (2019). Ship Detection from Optical Satellite Images with Deep Learning. *2019 9th International Conference on Recent Advances in Space Technologies (RAST)*, 479-484. <https://doi.org/10.1109/RAST.2019.8767844>
- Krüger, W. & Orlov, Z. (2010). Robust layer-based boat detection and multi-target-tracking in maritime environments. *2010 International WaterSide Security Conference*, 1-7. <https://doi.org/10.1109/WSSC.2010.5730254>
- LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444. <https://doi.org/10.1038/nature14539>
- Mahesh, B. (2020). Machine Learning Algorithms - A Review. *International Journal of Science and Research (IJSR)*, 9, 381-386. https://www.ijsr.net/get_abstract.php?paper_id=ART20203995
- Motomura, K. & Nagao, T. (2020). Fishing Activity Prediction from Satellite Boat Detection Data. *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2870-2875. <https://doi.org/10.1109/SMC42975.2020.9283451>
- Peleg, Y. (2021). Boats Boats Boats (1.^a ed.). <https://doi.org/10.34740/KAGGLE/DSV/2906695>.
- Proia, N. & Pagé, V. (2010). Characterization of a Bayesian Ship Detection Method in Optical Satellite Images. *IEEE Geoscience and Remote Sensing Letters*, 7(2), 226-230. <https://doi.org/10.1109/LGRS.2009.2031826>
- Singh, A., Thakur, N. & Sharma, A. (2016). A review of supervised machine learning algorithms. *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 1310-1315.
- Yang, G., Li, B., Ji, S., Gao, F. & Xu, Q. (2014). Ship Detection From Optical Satellite Images Based on Sea Surface Analysis. *IEEE Geoscience and Remote Sensing Letters*, 11(3), 641-645. <https://doi.org/10.1109/LGRS.2013.2273552>