

## **Comparación de algoritmos de máquinas de aprendizaje para la detección de arrecife de coral**

### ***(Comparison of machine learning algorithms for detecting coral reef)***

Eduardo Tusa<sup>1,2</sup>, Alan Reynolds<sup>1</sup>, Neil Robertson<sup>1</sup>, David Lane<sup>1</sup>,  
Hyxia Villegas<sup>2,3</sup>, Antonio Bosnjak<sup>3</sup>

#### **Resumen:**

El presente trabajo se enfoca en el desarrollo de un detector de coral de desempeño rápido que se utiliza para un vehículo autónomo submarino (AUV, por sus siglas en inglés). Una detección rápida de coral garantiza la estabilización del AUV en las cercanías del arrecife de coral en el menor tiempo posible, evitando que la desorientación del AUV destruya al arrecife. En este trabajo se usó la investigación de Purser, Bergmann, Lundälv, Ontrup, & Nattkemper (2009), por su precisión. Este detector consta de una parte de extracción de vectores característicos, la cual se realiza con filtros Gabor Wavelets; y una parte de clasificación de vectores que usa máquinas de aprendizaje, basado en Redes Neuronales. Debido al extenso tiempo de ejecución de las Redes Neuronales, se reemplazaron por un algoritmo de clasificación basado en Árboles de Decisión. Se utilizó una base de datos de 621 imágenes de corales de Belice (110 imágenes de entrenamiento y 511 imágenes de prueba). Se implementó un banco de filtros Gabor Wavelets utilizando C++ y la librería OpenCV. Se realizó la comparación de la precisión y el tiempo de ejecución de 9 algoritmos de máquinas de aprendizaje, cuyo resultado fue la selección del algoritmo de Árboles de Decisión. Nuestro detector de coral posee un tiempo de ejecución de 70ms en comparación con 22s desarrollados por el algoritmo de Purser et al. (2009).

**Palabras clave:** Coralbot, arrecife de coral, máquinas de aprendizaje, filtros Gabor Wavelets.

#### **Abstract:**

This work focuses on developing a fast coral reef detector, which is used for an autonomous underwater vehicle, AUV. A fast detection secures the AUV stabilization respect to an area of reef as fast as possible, and prevents devastating collisions. We use the algorithm of Purser et al. (2009) because of its precision. This detector has two parts: feature extraction that uses Gabor Wavelet filters, and feature classification that uses machine learning based on Neural Networks. Due to the extensive time of the Neural Networks, we exchange for a classification algorithm based on Decision Trees. We use a database of 621 images of coral reef in Belize (110 images for training and 511 images for testing). We implement the bank of Gabor Wavelets filters using C++ and the OpenCV library. We compare the accuracy and running time of 9 machine learning algorithms, whose result was the selection of the Decision Trees algorithm. Our coral detector performs 70ms of running time in comparison to 22s executed by the algorithm of Purser et al. (2009).

**Keywords:** Coralbot, coral reef, machine learning, Gabor Wavelets filters, OpenCV.

---

<sup>1</sup> Heriot-Watt University, Institute for Sensors Signals and Systems, Edinburgh – United Kingdom (et109@hw.ac.uk)

<sup>2</sup> Universidad Técnica de Machala, Facultad de Ingeniería Civil, Machala – Ecuador (etusa@utmachala.edu.ec)

<sup>3</sup> Universidad de Carabobo, Centro de Procesamiento de Imágenes, Valencia – Venezuela (hyxia@uc.edu.ve)

## **1. Introducción**

La restauración del arrecife de coral de manera convencional envuelve condiciones extremas para los buceadores voluntarios que trasplantan fragmentos perdidos de arrecife hacia un marco de arrecife de coral más grande. *Lophelia pertusa* constituye una de las construcciones de especies de coral más importantes en el mundo. No obstante, su ubicación genera limitaciones a la intervención humana debido a las bajas temperaturas y la profundidad del mar que es alrededor de 200 metros (Coralbot, 2013).

El proyecto Coralbot es una propuesta reciente para la reparación autónoma de los arrecifes de coral, con la finalidad de conservar los ecosistemas oceánicos; los mismos que son vitales para la pesca comercial, el turismo y la supervivencia de otras especies. La idea consiste en combinar AUVs con inteligencia de enjambre. Este tipo de inteligencia reproduce el comportamiento de organismos como abejas, termitas o avispas que actúan en grupo desarrollando tareas complejas utilizando reglas sencillas. De esta forma, el equipo de AUVs es desplegado para reconocer arrecifes de coral y ejecutar tareas de restauración.

La presente investigación propone la implementación de un detector de coral de desempeño rápido para que sea incorporado en el robot Coralbot. En este ambiente se requiere una respuesta de posicionamiento inmediato del AUV frente al arrecife de coral; de otro modo, éste perdería su orientación, lo que causaría daños al coral o a sí mismo.

Nos enfocamos en la obtención de los vectores característicos de textura, como lo realizó Purser et al. (2009), utilizando filtros de Gabor Wavelets que fueron implementados usando C++ y la librería OpenCV. Sin embargo, se hizo un estudio del desempeño del algoritmo de clasificación de estos descriptores de textura debido a que las Redes Neuronales utilizadas por Purser et al. (2009) consumían mucho tiempo de cómputo. Por este motivo, se evaluó la precisión y el tiempo de ejecución de nueve algoritmos de máquinas de aprendizaje. El algoritmo de Árboles de Decisión sustituye a las Redes Neuronales utilizadas por Purser et al. (2009).

A continuación discutiremos el estado del arte donde se resumen las teorías principales y las investigaciones actuales que explican la detección del arrecife de coral. Luego, se ilustra el desarrollo del algoritmo implementado en este proyecto. La siguiente sección presenta los resultados de la evaluación del algoritmo. Finalmente, se presentan las conclusiones.

## **2. Estado del arte del detector de coral**

### **2.1. Descriptores característicos del arrecife de coral**

Una cantidad considerable de literatura ha sido publicada sobre la extracción de descriptores característicos. Generalmente, la imagen es transformada en un conjunto de vectores

característicos, de modo que las diferentes regiones o formas deseadas son descritas cuantitativamente mediante sus propiedades: intensidad de color, información de textura, datos del espacio, señales de borde.

El tipo de características utilizadas en la mayoría de los artículos científicos están basados en información de color y textura. Siguiendo esta tendencia, Stought, Greer & Matt (2012) proponen el uso de características de color denominadas funciones cuantiles y descriptores de textura usando la transformada de características de escala invariante (SIFT, por sus siglas en inglés). Las características de color proporcionan buenos resultados, mientras las características SIFT fueron sensibles al ruido.

Marcos, David, Peñaflor, Ticzon & Soriano (2008) utilizan, o bien los valores de la media, el matiz y la saturación para la porción de la imagen; o los pesos mayores de los canales rojo y verde. Éstos son obtenidos mediante la comparación en el espacio normalizado de color rojo y verde, de los histogramas de las porciones de la imagen con aquellos obtenidos para los cuatro grupos de colores que ocurren en el ambiente del arrecife. Las características de texturas son generadas usando una versión rotacional invariante de los Patrones Binarios Locales (LBP, por sus siglas en inglés) (Ojala, Pietikäinen, & Mäenpää, 2002) para generar un histograma de diez componentes.

Beijbom, Edmunds, Kline, Mitchell, & Kriegman (2012) utiliza un banco del filtro rotacional invariante (el banco del filtro de respuesta máxima (MR)) para datos de textura. La información de color es también incluida mediante la aplicación de filtros a cada canal en el espacio de color Lab y apilando los vectores de respuesta del filtro.

Johnson-Roberson, Kumar & Willams (2006) generan 60 características visuales: 12 son la media y la desviación estándar de los canales RGB y HSV respectivamente. Los 48 restantes son obtenidos mediante la convolución de la región con los filtros Gabor Wavelets a seis escalas y cuatro orientaciones, tomando la media y la desviación estándar de los resultados para cada escala y orientación.

Purser et al. (2009) calcula 15 máscaras diferentemente orientadas y espaciadas con el fin de producir un conjunto de 30 características de texturas. Para comparar un sistema de visión de computadoras con el uso de métodos manuales utiliza un cuadrante de 15 puntos, cuadrante de 100 puntos y un marco de mapeo.

Puesto que la información de color es susceptible a la escasa iluminación del fondo marino, se decidió explotar las características de texturas a través del uso del filtro Gabor Wavelets que utiliza Purser et al. (2009), y Johnson-Roberson et al. (2006); pero utilizando siete escalas y cinco orientaciones.

## 2.2. Algoritmos de discriminación de vectores característicos de textura

El diseño del algoritmo orientado a la discriminación de las clases, coral o no coral, puede ser direccionado de diferentes maneras. En la mayoría de los artículos expuestos, los autores aplican aprendizaje supervisado utilizando máquinas (Mitchell, 1997). Esto consiste en tomar un vector característico por cada píxel de la imagen y asignarle su clase, coral o no coral. La asignación de un vector característico se ajusta a un modelo de predicción derivado de los datos de entrenamiento. Este modelo de algoritmo de predicción se explica a continuación.

En el trabajo de Purser et al. (2009) se realiza la detección de coral utilizando Redes Neuronales. A pesar de que los resultados son satisfactorios, el algoritmo consume mucho tiempo de cómputo para la detección de coral. Este tiempo excesivo de procesamiento podría llevar al robot a enfrentar condiciones catastróficas.

Por esta razón, esta investigación evaluó el tiempo de ejecución del desempeño de nueve algoritmos de máquinas de aprendizaje para seleccionar el más rápido, estos son: 1) Árboles de Decisión (DTR) (Loh, 2011), 2) Árboles Aleatorios (RTR) (Bradski & Kaehler, 2008), 3) Árboles extremadamente aleatorios (ERT) (Geurts, Ernst, & Wehenkel, 2006), 4) Boosting (BOO) (Luber, Spinello, & Arras, 2011), 5) Gradiente de Árboles Boosting (GBT) (Friedman, 2000), 6) Clasificador Normal Bayes (NBA) (Mitchell, 1997), 7) Maximización de la Esperanza (EMA) (Bilmes, 1998), 8) Redes Neuronales (MLP) (Bradski & Kaehler, 2008), 9) Máquinas de Vectores de Soporte (SVM) (Mitchell, 1997). El criterio utilizado para la selección de estos algoritmos radica en que se encuentran implementados en C++, y disponibles en la librería OpenCV.

## 3. Metodología

El detector de coral toma una imagen del conjunto de imágenes de prueba y sigue los siguientes pasos (Figura 1):

**A. Pre-procesamiento:** La imagen se lleva a escala de grises para normalizar el valor del píxel.

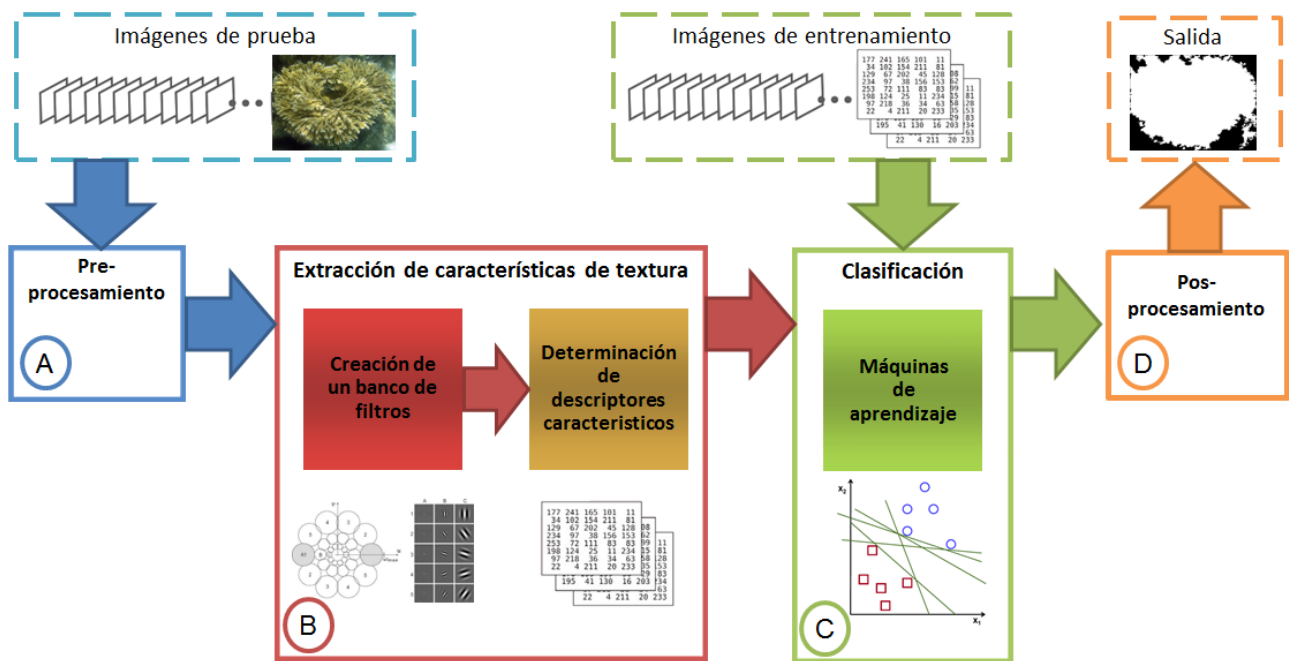
**B. Extracción de características de textura:** Los descriptores característicos de textura se extraen mediante la convolución de la imagen con un banco de filtros Gabor Wavelets. Siguiendo este procedimiento, a cada píxel de la imagen se le asocia un vector característico.

**C. Clasificación:** Se realiza un proceso de discriminación entre la clase coral y no coral mediante un aprendizaje supervisado de máquinas haciendo uso de los nueve algoritmos mencionados anteriormente.

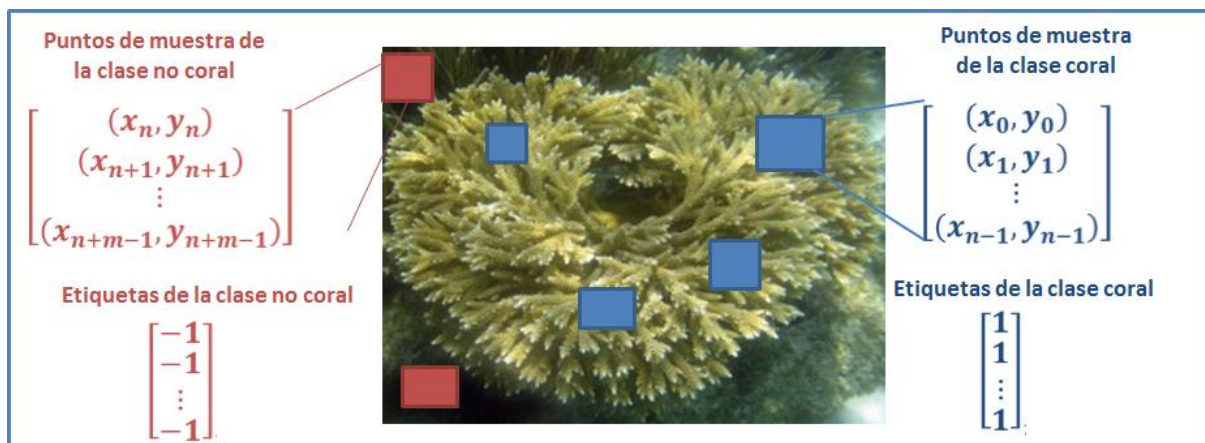
**D. Pos-procesamiento:** Los falsos positivos y los falsos negativos son removidos mediante la selección de los contornos más grandes.

### 3.1. Creación de la base de datos

La base de datos contiene 621 imágenes provenientes del arrecife de coral ubicado en Belice. Este conjunto de imágenes se divide en 110 imágenes para entrenamiento y 511 imágenes para prueba. Sobre cada imagen, el experto seleccionó regiones de interés de la imagen con la forma de un polígono, para separar las regiones que pertenecen al coral, de las regiones que no pertenecen al coral, asignándoles los valores +1 y -1, respectivamente (Figura 2), seleccionando  $n$  puntos para la clase coral, y  $m$  puntos para la clase no coral. Los elementos, en estas regiones seleccionadas, se almacenan en un vector, en donde se registran las coordenadas de los píxeles dentro de estas regiones.



**Figura 1.** Diagrama de bloques del Detector de Coral: A. Pre-procesamiento, B. Extracción de características de textura, C. Clasificación y D. Pos-procesamiento.



**Figura 2.** Las regiones de interés de la imagen está conformada por polígonos de  $n$  puntos para una clase de coral, y  $m$  puntos para una clase que no es coral.

### 3.2. Filtros Gabor Wavelets

En el dominio espacial, un filtro Gabor 2D es el resultado de una función Gaussiana modulada por una onda sinusoidal. La expresión matemática para este filtro se describe en (1)

$$g(x, y) = e^{-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)} e^{-ik(x-x_0)} \quad (1)$$

donde  $(x_0, y_0)$  es el centro de la onda Gaussiana,  $\sigma_x$  y  $\sigma_y$  son las varianzas de la Gaussiana a lo largo de los ejes  $x$  y  $y$ , respectivamente;  $k = 2\pi/\lambda$  es la frecuencia espacial de una onda en el plano complejo con la onda normal a lo largo del eje  $x$ , y  $\lambda$  es la longitud de onda.

Se trabaja con filtros Gabor Wavelets que son funciones similares a sí mismas. Si se considera a la función  $g(x,y)$  como la Gabor Wavelets matriz, entonces un banco de filtros similares a sí mismos puede ser creado mediante escalamientos y rotaciones de la función  $g(x,y)$  a través de las expresiones (2), (3) y (4)

$$g_{mn} = g(x', y') \quad (2)$$

$$x' = a^{-m} (x \cos \Theta_n + y \sin \Theta_n) \quad (3)$$

$$y' = a^{-m} (-x \sin \Theta_n + y \cos \Theta_n) \quad (4)$$

El diseño del diccionario conformado por los filtros Gabor Wavelets utiliza 7 escalas y 5 orientaciones diferentes. Los parámetros del filtro son seleccionados para asegurar que una magnitud de la mitad del pico de las respuestas del filtro, en el espectro de frecuencia, se topen entre ellas (Manjunath & Ma, 1996). Las expresiones para mantener estas condiciones están dadas por (5), (6), (7)

$$a = \left(\frac{U_h}{U_l}\right)^{\frac{1}{M-1}} \quad (5)$$

$$\sigma_x = \frac{(a+1)\sqrt{2\ln(2)}}{2\pi(a-1)U_h} \quad (6)$$

$$\sigma_y = \frac{\sqrt{2\ln(2) - \left(\frac{\ln(2)}{\pi\sigma_x U_h}\right)^2}}{2\pi \tan\left(\frac{\pi}{2N}\right) \left[U_h - 2\ln\left(\frac{1}{2\pi^2\sigma_x^2 U_h}\right)\right]} \quad (7)$$

donde  $a$  es un factor de la escala del filtro,  $U_l$  y  $U_h$  denotan la más baja y la más alta frecuencias de interés, respectivamente.  $M$  es el número de escalas y  $N$  es el número de orientaciones. En la Tabla 1 se muestran los valores de los parámetros utilizados para generar el banco de filtros. Se sigue los procedimientos señalados en Manjunath & Ma (1996), y Chen et. al (2004) para seleccionar los valores de la frecuencia. Chen et al. (2004) resaltan que es muy raro tener imágenes con valores extremos de frecuencias límites inferior y superior entre 0 y 0.5.

**Tabla 1.** Parámetros del banco de filtros Gabor Wavelets

Parámetro	Valor
Número de escalas	7
Número de orientaciones	5
Frecuencia central superior $U_l$	0.40
Frecuencia central inferior $U_h$	0.05

### 3.3. Extracción de las características de textura

Manjunath et al. (1996) propone un método que utiliza propiedades estadísticas de la repuesta de los filtros con la finalidad de obtener una descripción significativa de textura. Esto se logra mediante la determinación de la media y la desviación estándar no normalizada de la imagen.

La idea es dividir la imagen en conjuntos de bloques rectangulares sobrepuestos que están centrados sobre una malla. Por cada bloque, un vector de textura es calculado y asociado con su correspondiente posición en la malla. La resolución de la malla es igual a la resolución de la imagen en píxeles, cuyo tamaño es de 1024 x 768.

El valor de la media sobre un bloque pequeño es equivalente a la convolución de una imagen de coral con la máscara del filtro. Ontrup, Ritter & Wersing, (2004) utilizan máscaras Gaussianas con el fin de mejorar los resultados de alisamiento. Las características de textura están dadas por las siguientes expresiones en (8) y (9)

$$\mu_{mn}(x, y) = c_{mn}(x, y) * gs_{mn}(x, y) \quad (8)$$

$$\sigma_{mn}(x, y) = \sqrt{(c_{mn}(x, y) - \mu_{mn}(x, y))^2 * gs_{mn}(x, y)} \quad (9)$$

donde  $c_{mn}(x, y)$  es la respuesta al canal  $mn$ , correspondiente a la escala  $m$  y la orientación  $n$ , mientras que  $gs_{mn}(x, y)$  está dada por la expresión en (10)

$$g_{sm}(x, y) = \exp\left(-\frac{x^2}{2\rho_x^2} - \frac{y^2}{2\rho_y^2}\right) \quad (10)$$

Ontrup et al. (2004) seleccionaron los anchos del filtro de alisamiento a ser  $\rho_x = 3\sigma_x$  y  $\rho_y = 3\sigma_y$ . Por cada ubicación del píxel en la imagen de entrada, se genera un vector característico de dimensión 70,  $t(x, y)$ , describiendo la textura local en ese punto

$$t(x, y) = [\mu_{1,1} \dots \mu_{5,7}, \sigma_{1,1} \dots \sigma_{5,7}] \quad (11)$$

#### 4. Resultados y Discusión

Los experimentos de este trabajo consistieron en comparar los nueve algoritmos: Árboles de Decisión (DTR), Árboles Aleatorios (RTR), Árboles extremadamente aleatorios (ERT), Boosting (BOO), Gradiente de Árboles Boosting (GBT), Clasificador Normal Bayes (NBA), Maximización de la Esperanza (EMA), Redes Neuronales (MLP), Máquinas de Vectores de Soporte (SVM); disponibles en la librería OpenCV. Se usó una base de datos de 621 imágenes de los arrecifes de coral de Belice, con 110 imágenes para entrenamiento y 511 imágenes para prueba.

En estos experimentos se utiliza una laptop ASUS G55V con procesador Intel Core i7 – 3610QM a 2.30GHz, con memoria RAM de 12 GB, con una tarjeta gráfica NVIDIA GEFORCE GTX 660M 2GB y con sistema operativo UBUNTU. Esto proporciona un ordenador con el triple del número de núcleos que el computador utilizado por Purser et al. (2009), el cual es una laptop Dell M1530 con un procesador Intel Core2 Duo a 2.4 GHz con sistema operativo LINUX.

Los expertos seleccionaron regiones de interés de las 110 imágenes de entrenamiento y generaron 6.800.071 vectores característicos, para el entrenamiento de los algoritmos clasificadores de máquinas de aprendizaje, mencionadas anteriormente. La Figura 3 resume las diferentes métricas utilizadas para comparar los algoritmos clasificadores: Tiempo de ejecución, Precisión, Especificidad y Sensitividad.

El resultado de estos experimentos de comparación dio como algoritmo más rápido, el Árbol de Decisión (DTR) al clasificar una imagen de 1024 x 768 píxeles en un tiempo de ejecución de 70 ms. El tiempo de los demás algoritmos se puede apreciar en la Figura 3a.

El algoritmo de Purser et al. (2009) utiliza redes neuronales basadas en un mapeo auto organizado de características (self-organizing feature map, SOM por sus siglas en inglés). Este algoritmo aprende a mapear descriptores de textura similares, a un vecindario de regiones que están en un disco espacial de dimensión 2. El mapeo divide los datos en clústeres jerárquicamente organizados, de modo que las regiones con apariencia de textura similares son mapeadas a los mismos clústeres, o a los clústeres cercanos sobre el mapa. En términos de tiempo de ejecución, el proceso de mapeo que ordena asimismo la distribución de los vectores de



entrada puede resultar en un tiempo adicional al generado por el algoritmo de Redes Neuronales; lo cual puede ser compensado mediante la reducción de la dimensión de los vectores o del tamaño de la imagen.

Además, el sistema de visión del Coralbot presenta una ventaja importante en los criterios de diseño. El detector necesita discriminar entre dos clases: coral y no coral. Si bien el algoritmo de discriminación de Purser et al. (2009) clasifica en cuatro clases diferentes: coral, no coral y dos tipos de esponja; el tiempo empleado para el proceso de clasificación de vectores característicos se puede reducir, si se disminuye el número de clases que se discriminan. Por esta razón, la disminución del número de las clases a predecir, tiene un importante efecto en el tiempo de ejecución del algoritmo.

#### **4.1. Resultados de la evaluación de la precisión**

La precisión mide la tasa de predicciones correctas del número total de casos evaluados. En la Figura 3b se aprecia que ERT, MLP, GBT y BOO sobrepasan el 70% de precisión. DTR, NBA y EMA poseen precisiones que no sobrepasan el 60%. SVM no produce una clasificación notoria entre los nueve clasificadores.

La Figura 4 resume los resultados obtenidos para la tarea de clasificación del coral Staghorn. En la Figura 4a se presenta la imagen de entrada, mientras que las Figuras restantes, presentan los resultados de los algoritmos de clasificación mencionados anterior. La clase no coral o plano de fondo, se ha coloreado de blanco con el fin de contrastar con lo que el sistema reconoce como coral. Las Figuras 4b y 4c presentan los resultados de los algoritmos DTR y BOO, respectivamente. Se aprecia una ubicación satisfactoria de la región que corresponde al coral dentro de la escena. El algoritmo de EMA (ver Figura 4d) pierde levemente algunas detecciones positivas verdaderas. No obstante, el NBA (ver Figura 4e) incrementa notoriamente el número de falsos negativos, enviando píxeles de coral al plano de fondo. Las Figuras 4f y 4g que corresponden a los algoritmos ETR y MLP, respectivamente; presentan una alta presencia de falsos positivos, es decir; píxeles del plano de fondo considerados como arrecifes de coral. El algoritmo RTR (ver Figura 4h) muestra resultados similares a DTR y a BOO; sin embargo, algunos falsos positivos se filtran en el plano de fondo. La Figura 4i corresponde al algoritmo de GBT, el cual presenta dificultades para reconocer al buceador como plano de fondo. Finalmente, la Figura 4j muestra el resultado del algoritmo SVM. La imagen presenta una pequeña presencia de coral con un abundante plano de fondo detectado como coral. Esto dificulta focalizar un sector específico de arrecife de coral en la escena.

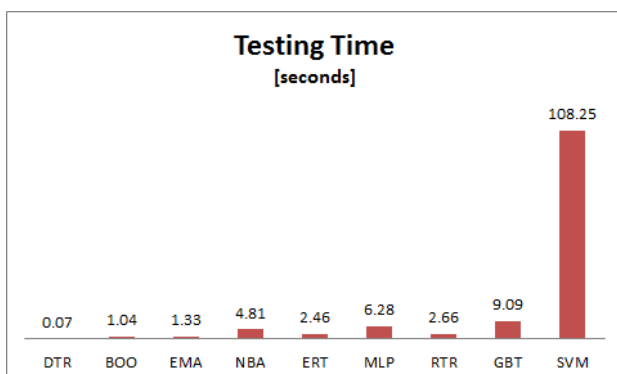
#### **4.2. Resultados de la evaluación de la sensibilidad y especificidad**

La sensibilidad es una medida de probabilidad para una segmentación que recoge la presencia de coral en la ubicación correcta cuando esta clase está originalmente en esa ubicación. La

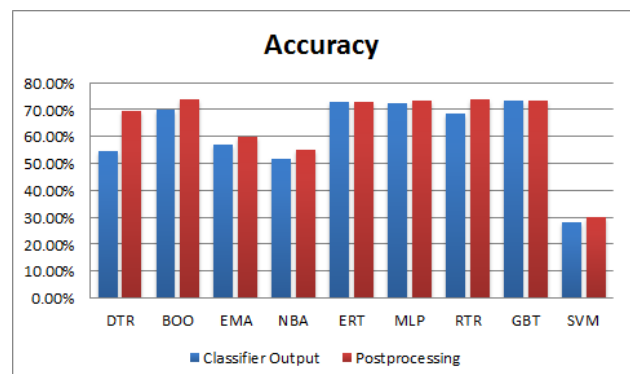
especificidad es definida como la probabilidad en que la imagen segmentada, el arrecife de coral no es detectado en una ubicación cuando de hecho no es arrecife de coral. Un resultado importante es que las tasas de especificidad son más bajas que las tasas de sensibilidad. En general, si la sensibilidad es alta y la especificidad es baja, entonces no hay necesidad de preocuparse de los píxeles del plano de fondo en la imagen segmentada, pero el coral segmentado debe ser reexaminado para eliminar los falsos positivos (plano de fondo seleccionado por error como coral). Por tal motivo, se realiza una etapa de posprocesamiento.

### 4.3. Resultados de la evaluación del pos-procesamiento

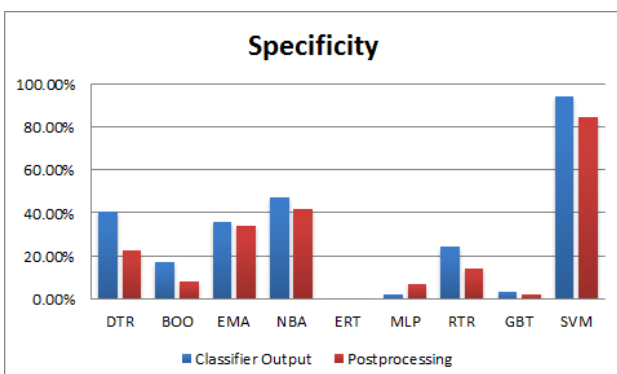
El pos-procesamiento aplica un algoritmo de conexión de componentes. Algunos puntos fueron removidos de las regiones de coral en los resultados de DTR, EMA y NBA. Esto clarifica una mejor ubicación del arrecife de coral en la escena. Además, el algoritmo descarta falsos negativos presentes en el plano de fondo. Los valores de especificidad se disminuyen para casi todos los clasificadores, con excepción de Redes Neuronales que presenta un incremento (Figura 3c). Se muestra incrementos en las tasas de sensibilidad para DTR, BOO y SVM. El resto de clasificadores no exhiben una mejora significativa de sensibilidad (Figura 3d)



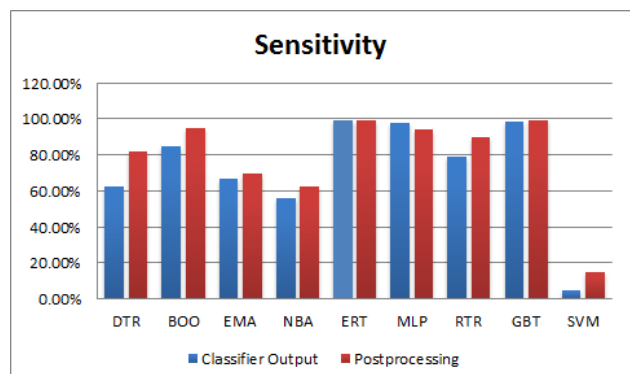
(a) Tiempo de ejecución



(b) Precisión



(c) Especificidad



(d) Sensitividad

**Figura 3.** Métricas utilizadas para la comparación de los nueve clasificadores:

(a) Tiempo de ejecución. (b) Precisión. (c) Especificidad. (d) Sensitividad

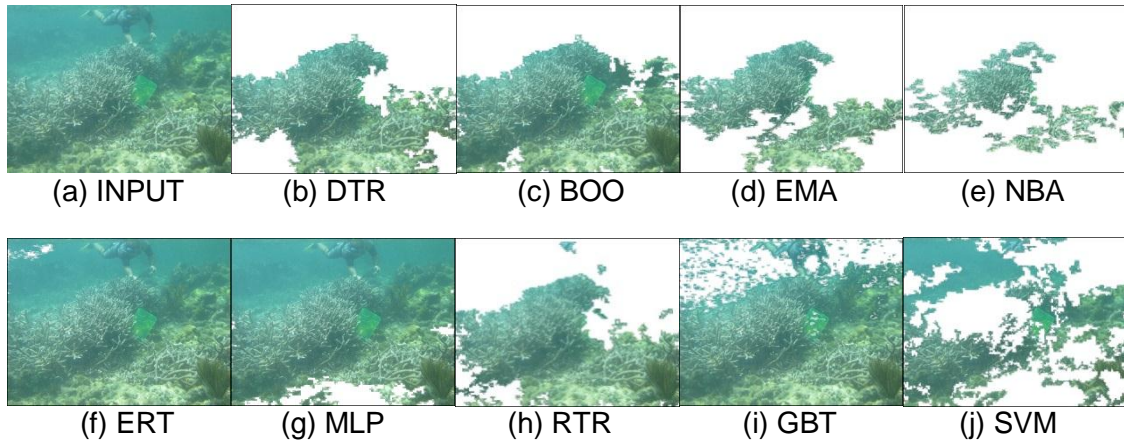


Figura 4. Resultados de la detección del coral Staghorn

## 5. Conclusiones y Recomendaciones

Este artículo presenta un detector de coral que clasifica entre dos clases: la clase que pertenece al coral y el plano de fondo que no pertenece al coral. El algoritmo de aprendizaje supervisado se entrena con vectores característicos de textura generados por un banco de filtros Gabor Wavelets. Nueve clasificadores fueron evaluados mediante 4 métricas. Los clasificadores: Árboles extremadamente aleatorios, Gradiente de Árboles Boosting, Redes Neuronales y Máquinas de Vectores de Soporte; mostraron dificultades para discriminar las clases. En contraste, los clasificadores: Árboles de Decisión, Boosting, Maximización de la Esperanza, Clasificador Normal Bayes y Árboles Aleatorios; presentaron una clasificación visual prometedor. El algoritmo de Árboles de Decisión (DTR) presentó el menor tiempo de ejecución con la más alta precisión antes y después del pos-procesamiento. Por lo tanto, recomendamos la inclusión del algoritmo de Árboles de Decisión (DTR) como un software imprescindible para el robot submarino: Coralbot.

El trabajo futuro consiste en la integración del detector de coral con el AUV Coralbot mediante un simulador de robots marinos, denominado UWSim con la finalidad de probar algoritmos de servocontrol. Éstos permitirán relacionar la velocidad del robot con la ubicación del arrecife en la escena, para ejecutar acciones de control y evitar así su desorientación o la ruptura del arrecife.

## Bibliografía

- Beijbom, O., Edmunds, P., Kline, D., Mitchell, G., & Kriegman, D. (2012). Automated Annotation of Coral Reef Survey Images. *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1170-1177.
- Bilmes, J. (1998). A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. *International Computer Science Institute and Computer Science Division*.

- Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. Cambridge: MA: O'Reilly.
- Chen, L., Lu, G., & Zhang, D. (2004). Effects of different gabor filter parameters on image retrieval by texture. *Multimedia Modelling Conference, 2004. Proceedings. 10th International*, 273 - 278.
- Coralbot, T. (1 de Abril de 2013). *About coral reefs and current approaches for restoration*. Obtenido de Coralbot Web site: <http://www.coralbots.org/>
- Friedman, J. (2000). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 1189-1232.
- Geurts, P., Ernst, D., & Wehenkel, L. (2006). *Extremely randomized trees*. Liege: Springer Science + Business Media.
- Johnson-Roberson, M., Kumar, S., & Williams, S. (2006). Segmentation and classification of coral for oceanographic surveys: A semi-supervised machine learning approach. *Proceedings of OCEANS 2006 - Asia Pacific*.
- Loh, W.-Y. (2011). Classification and regression tree. *WIREs Data Mining and Knowledge Discovery*, 14-23.
- Luber, M., Spinello, L., & Arras, K. O. (2011). People tracking in rgb-d data with on-line boosted target models. *in Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*.
- Manjunath, B. S., & Ma, W. Y. (1996). Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 837-842.
- Marcos, M. S., David, L., Peñaflo, E., Ticzon, V., & Soriano, M. (2008). Automated benthic counting of living and non-living components in Ngedarrak Reef, Palau via subsurface underwater video. *Environmental Monitoring and Assessment*, 177-184.
- Mitchell, T. (1997). *Machine Learning*. New York: McGraw-Hill, Inc.
- Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 971-987.
- Ontrup, J., Ritter, H., & Wersing, H. (2004). A computational feature binding model of human texture perception. *Cognitive Processing*, 34-44.
- Purser, A., Bergmann, M., Lundälv, T., Ontrup, J., & Nattkemper, T. W. (2009). Use of machine-learning algorithms for the automated detection of cold-water coral habitats: a pilot study. 397(3354/meps08154).
- Stough, J., Greer, L., & Matt, B. (2012). Texture and Color Distribution-based Classification for Live Coral Detection. *Proceedings of the 12th International Coral Reef Symposium*.