

Detección de arrecifes de coral con Machine Learning.

Brian Armando Quiñón Padilla
Universidad Veracruzana
Facultad de Electrónica y Comunicaciones
Email: brianqp9889@gmail.com

Abstract—Con la necesidad de preservar y cuidar los arrecifes de coral, este proyecto planea sentar las bases de investigaciones futuras para la detección de arrecifes de coral utilizando Machine Learning y Opencv con Python.

Keywords: Arrecifes de Coral, Machine Learning.

I. INTRODUCCIÓN

Los arrecifes de coral son complejos e importantes para la fauna, siendo una fuente primaria de alimentos, ingresos, y vida marina para varias ciudades. Por ello es importante mantenerlos sanos identificarlos de manera correcta. En ellos abundan muchísima vida, por lo tanto, hay muchos diferenciadores entre cada metro cuadrado recorrido en ellos, por lo tanto, intentar identificar uno por uno puede llegar a ser tedioso. En este proyecto se plantearán las bases para clasificar algunos tipos de corales marinos y demostrar la eficacia si se toman en cuenta otros factores.

II. ESTADO DEL ARTE

Antes que nada, vamos a desglosar conceptos importantes necesarios para entender lo siguiente:

A. Arrecife de coral

El arrecife de coral es una montaña bajo el agua formada por esqueletos de coral. Los arrecifes también están compuestos de otros seres vivos, como algas o moluscos. El arrecife de coral tiene colores brillantes y puede crecer cientos de años sin ser destruido por el océano. Por lo tanto, en ellos abunda la vida. En este proyecto nos centraremos únicamente en 2 tipos, el primero de ellos es el Brain Coral, pertenecientes a la familia Mussidae y Merulinidae. El segundo es el Staghorn, que se puede identificar fácilmente como una ramificación perteneciente a la familia Acropora cervicornis.

B. Machine Learning

La herramienta a utilizar es un modelo secuencial de una red neuronal que simula el sistema nervioso. Para poder identificar las imágenes en tiempo real, se utilizan redes convolucionales para identificar características y clasificar una imagen. Esta red posee capas ocultas y varias salidas que indican qué elemento esta identificando la computadora, este modelo tiene que pasar por un proceso de entrenamiento y validación para ajustar sus pesos, y una vez finalizado, puede utilizarse.

III. DISEÑO E IMPLEMENTACIÓN

Se siguió la siguiente metodología:

```
for path in direcciones:
    contador = 0
    print("Imprimiendo: ", path)
    for file in os.listdir(path):
        file = os.path.join(path, file)
        temp = cv2.imread(file)
        temp = cv2.resize(temp, (128, 128))
        cv2.imwrite(file, temp)
        cv2.imshow('a', temp)
        print(temp.shape)
    # contador += 1
```

Fig. 1: Utilizando el resize.

A. Recolección de imágenes para entrenamiento y validación

La primera parte de la información fue obtenida de un dataset especializado en algunos tipos de corales. El conjunto de datos fue tomado el 7 de diciembre de 2017 como herramienta para la utilización de redes neuronales como material de entrenamiento para la extracción de características. El conjunto de datos no fue suficiente debido a que se necesitan datos de entrenamiento y de validación. Por lo tanto, tuve que reducir las categorías a 4 para simplemente clasificar los labels como:

- Brain Coral
- Staghorn Coral
- Sand
- Water and fishes

El truco en todo esto es que el modelo pueda distinguir un coral en todo el ambiente que puede rodearlo; siendo estos agua, peces o superficies rocosas. Los datos de validación fueron sacados de Google Imágenes para tener material.

B. Utilización de funciones de opencv para escalar las imágenes.

Una vez que obtenidas las imágenes del dataset y de Google, con el archivo resize por medio de la librería os y opencv, accede a las carpetas y cambia el tamaño de cada una a 128x128 pixeles.

Una problemática de intentar clasificar el agua y un arrecife de coral, es que no todas las imágenes se pueden aislar para eliminar el agua; pero lo que se puede hacer es crear

```
def remove_color(low_color, high_color, img):
    hsv_blue = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
    # blue range
    lower_blue = np.array(low_color)
    upper_blue = np.array(high_color)
    # Define a mask ranging from lower to upper
    mask = cv2.inRange(hsv_blue, lower_blue, upper_blue)
    # Do masking
    res = cv2.bitwise_and(img, img, mask=mask)
    return res
```

Fig. 2: En function remove.

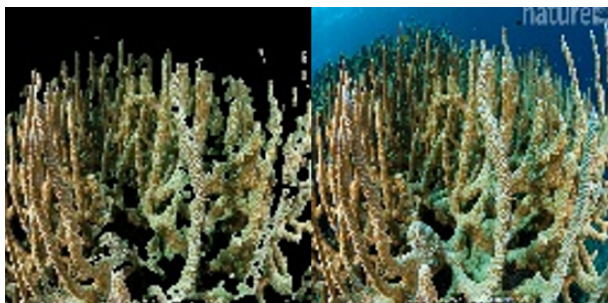


Fig. 3: En function remove.

una mascara que trate de eliminar el color azul para que el resultado sea mas 'eficiente', pero esto puede ser una problemática debido a la lógica de una neurona compleja que trata de buscar o asimilar figuras. De todas formas intente aplicarlo para mejorar los resultados

Teniendo como resultado algo similar:

C. Creación del modelo secuencial y entrenamiento con imágenes de prueba y de validación.

El modelo que se uso fue una red neuronal con:

- 1 capa de entrada en donde el input es una imagen de 128x128.
- 3 capas densas de 1000, 500 y 100 neuronas con activación 'relu'. (La cantidad se escogió tomando en cuenta que son pocas imágenes a procesar).
- 4 neuronas de salida que indican la clase que está identificando.

Compilacion

Utilice valores predeterminados, tales como el optimizar Adam, la perdida Categorical Crossentropy.

IV. RESULTADOS OBTENIDOS

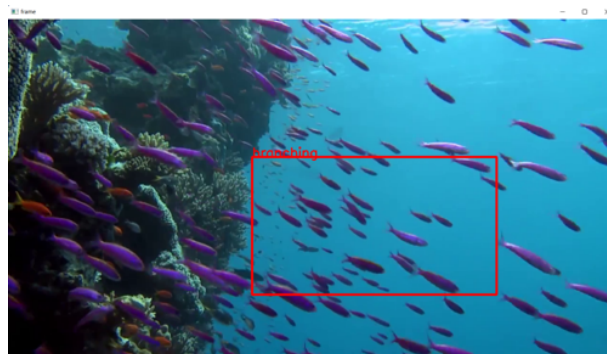


Fig. 4: Error causado a la cantidad de peces.

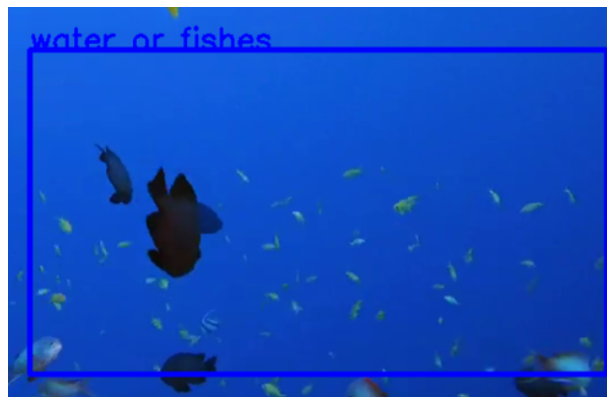


Fig. 5: Water or Fishes.



Fig. 6: Branching detectado.



Fig. 7: Mayoria de agua y peces.



Fig. 8: Brain Coral detectado.

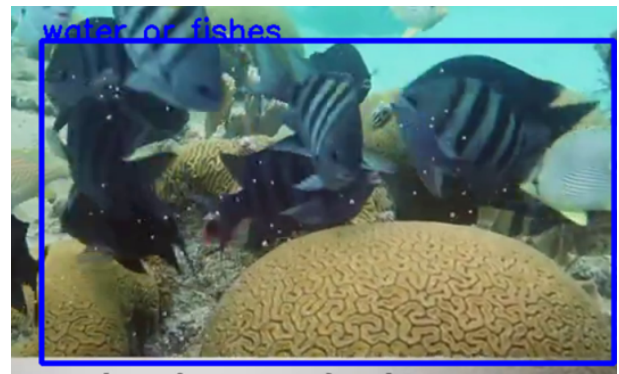


Fig. 10: Mayoria de peces



Fig. 9: Brain Coral detectado.

V. COMENTARIOS FINALES.

El algoritmo aún tiene muchas fallas, es un modelo entrenado con 70 imágenes de entrenamiento y 40 imágenes de validación por clase. Sería muchísimas veces mas preciso si pudiera entrenarse con mas datos y con aun más amplitud; pero debo de admitir que da mejores resultados de los esperados al jugar bien con las épocas, pasos e imágenes seleccionadas. De manera personal considero que no solo se deben de entrenar los arrecifes de coral, sino también todos los elementos que incluye el océano, tales como agua, seres vivos, navíos naufragados, etc; aunado con un tratamiento de imágenes para hacerle un crop y quedarnos exclusivamente con lo necesario.