

Detección de objetos bajo el agua para Unmanned Underwater Vehicles

Instituto Tecnológico y de Estudios Superiores de Monterrey
Del Valle, Diego; Torres, Claudia & Ramírez, Hortencia.



Abstract

En este documento se abordará la posibilidad de poder otorgar una posible propuesta de solución para el reconocimiento de objetos bajo el agua en vehículos no tripulados a través del entrenamiento de una red neuronal capacitada contra diversas tonalidades de agua, corrientes, clases de objetos para lograr identificar al estar flotando libremente. Por otro lado, se implementa el uso de ROS en conjunto con la librería YoloV5 que ofrece Python para realizar la detección de objetos en el sistema operativo Linux para ser implementado en la cámara, todo esto para lograr la detección de objetos debajo del agua.

Introducción

Los vehículos submarinos no tripulados (UUV) son el futuro de la exploración submarina. Aunque actualmente están en uso, aún no son tan avanzados y necesitan un mayor desarrollo para una investigación más intensiva de los entornos no descubiertos en el planeta Tierra. Es un tema de creciente interés que presenta múltiples desafíos debido a que la presencia de cuerpos acuáticos dificultan en gran medida a los UUV el reconocimiento exacto de sus alrededores.

El progreso tecnológico es vertiginoso, tenemos la oportunidad de ver importantes desarrollos e inversiones casi todas las semanas en todo el mundo. Las nuevas investigaciones son ejecutadas e implementadas rápidamente por grandes empresas y expandiendo el avance global en las tecnologías más nuevas. ¿La automatización de estos vehículos será capaz de revolucionar la exploración submarina?

El problema establecido es la exploración acuática, más específicamente el comprender cómo a base de herramientas tecnológicas es posible llegar a lugares difíciles de alcanzar por los humanos de manera que no haya vidas en peligro. Con los hallazgos y creaciones de la actualidad es posible crear herramientas más complejas y rápidas de la mano de algoritmos tales como YOLO v5 y del uso de softwares como lo es ROS.

Metodología

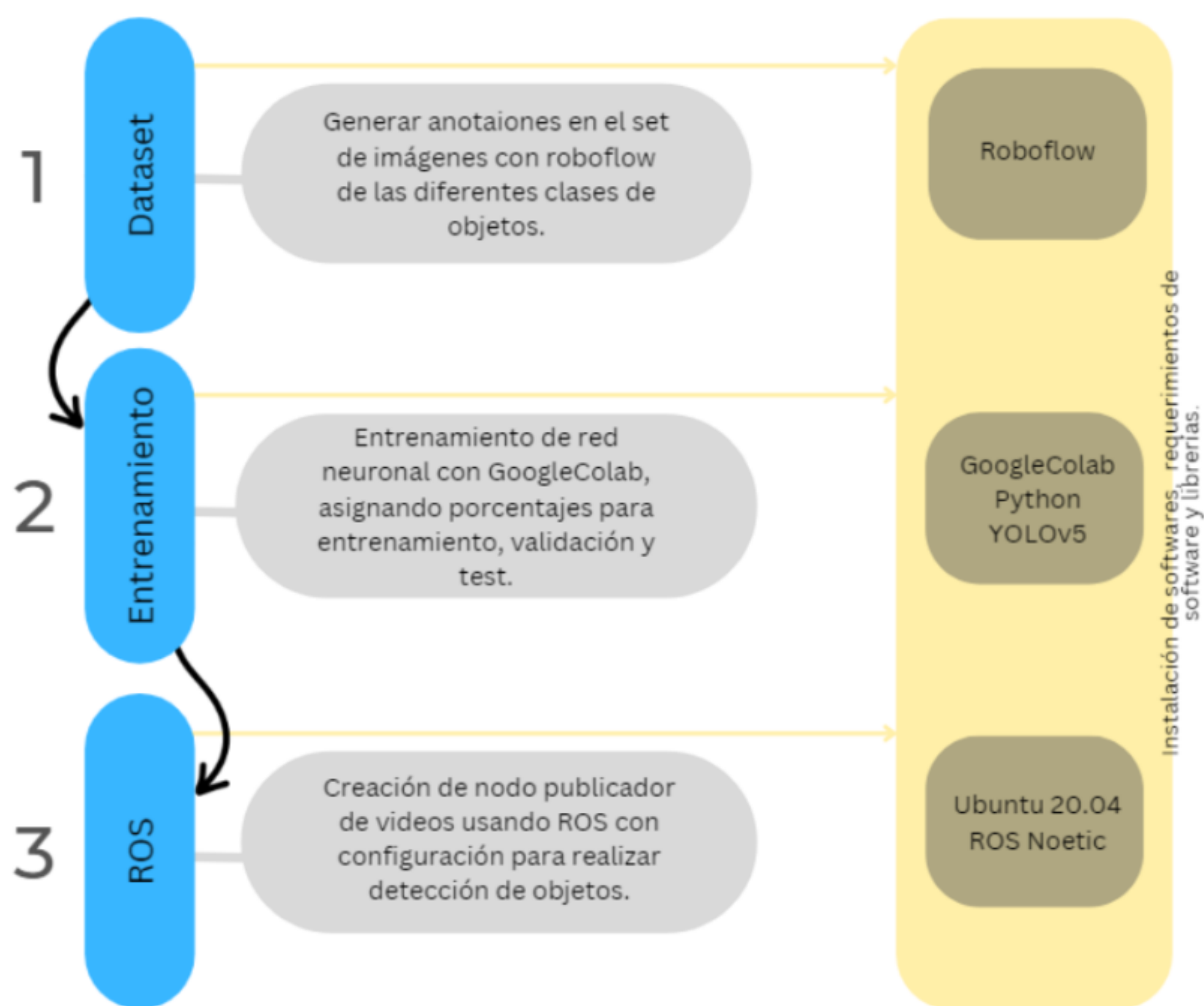


Figure 1: Diagrama de metodología

Herramientas de Ingeniería:

Artículos científicos, sitios oficiales de softwares de desarrollo (ROS, Python, YoloV5, Ubuntu, NVIDIA) y repositorios de GitHub oficiales.

Resultados

Los resultados obtenidos tras dar de alta el servidor de ROS, ejecutar el nodo y hacer launch con el archivo de configuraciones que contiene la red de YOLOv5 entrenada da como respuesta un conjunto de frames, dichos resultados pueden ser observados en las siguientes figuras. Se realizó una prueba del funcionamiento, con un video grabado fuera del agua, asignando un filtro azul encima de él para simular el contexto bajo el cual se encontrará el submarino, con ello se obtuvo el resultado mostrado en la siguiente figura.

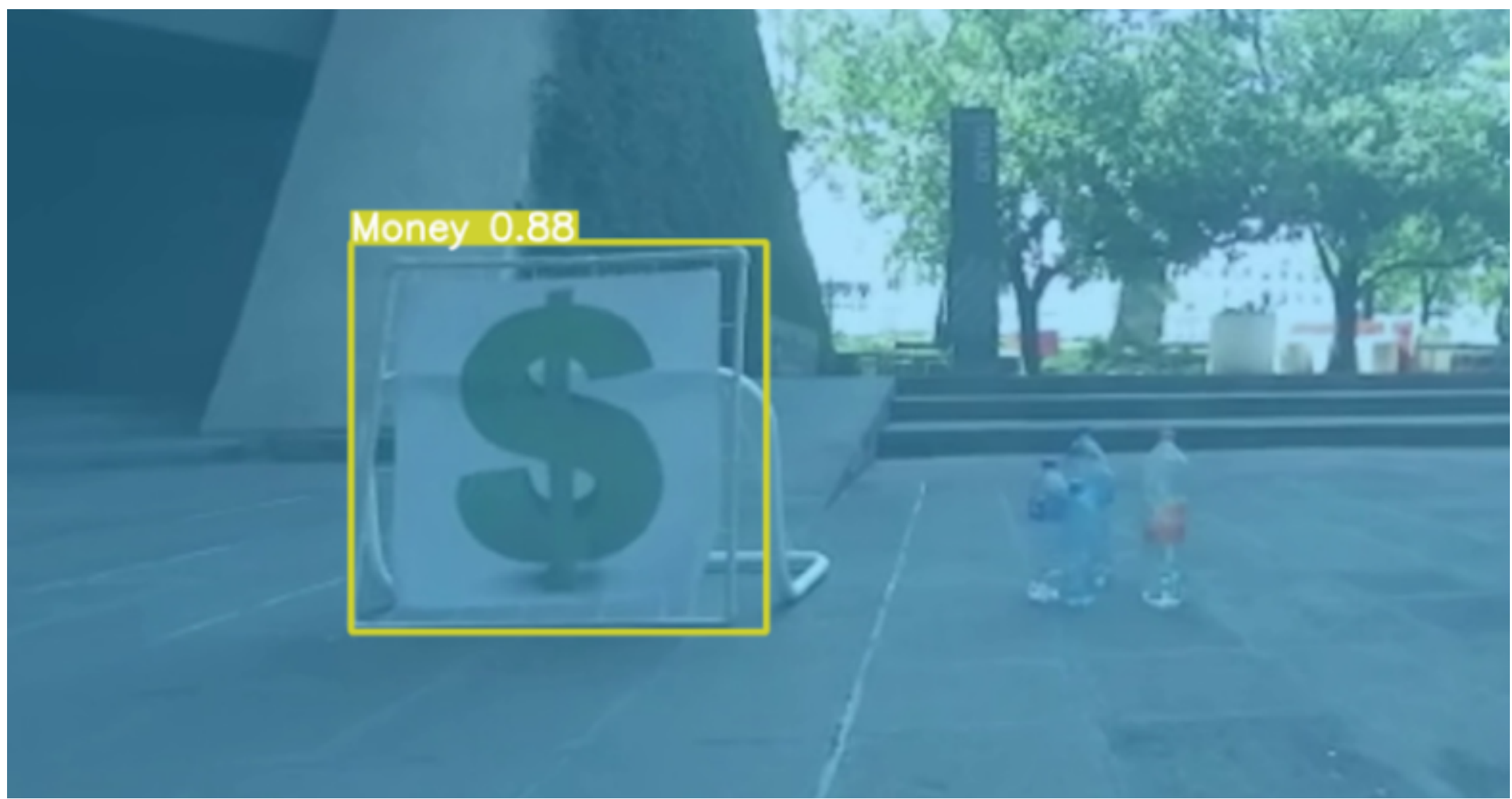


Figure 2: Detección de la clase objeto Money fuera del agua usando un filtro azul

Tras observar que las condiciones bajo las cuales se estaban realizando las pruebas del funcionamiento del algoritmo eran poco parecidas a las condiciones reales bajo las cuales estaría el subamrino, se realizaron nuevas pruebas. Con el objetivo de probar el resultado que se obtendría en un contexto bajo condiciones más similares a las de la competencia y tomando como referencia

vídeos de la competencia en ediciones anteriores, obtuvimos resultados mucho más precisos en la identificación de las diferentes clases de objetos, como se muestra en las siguientes figuras.

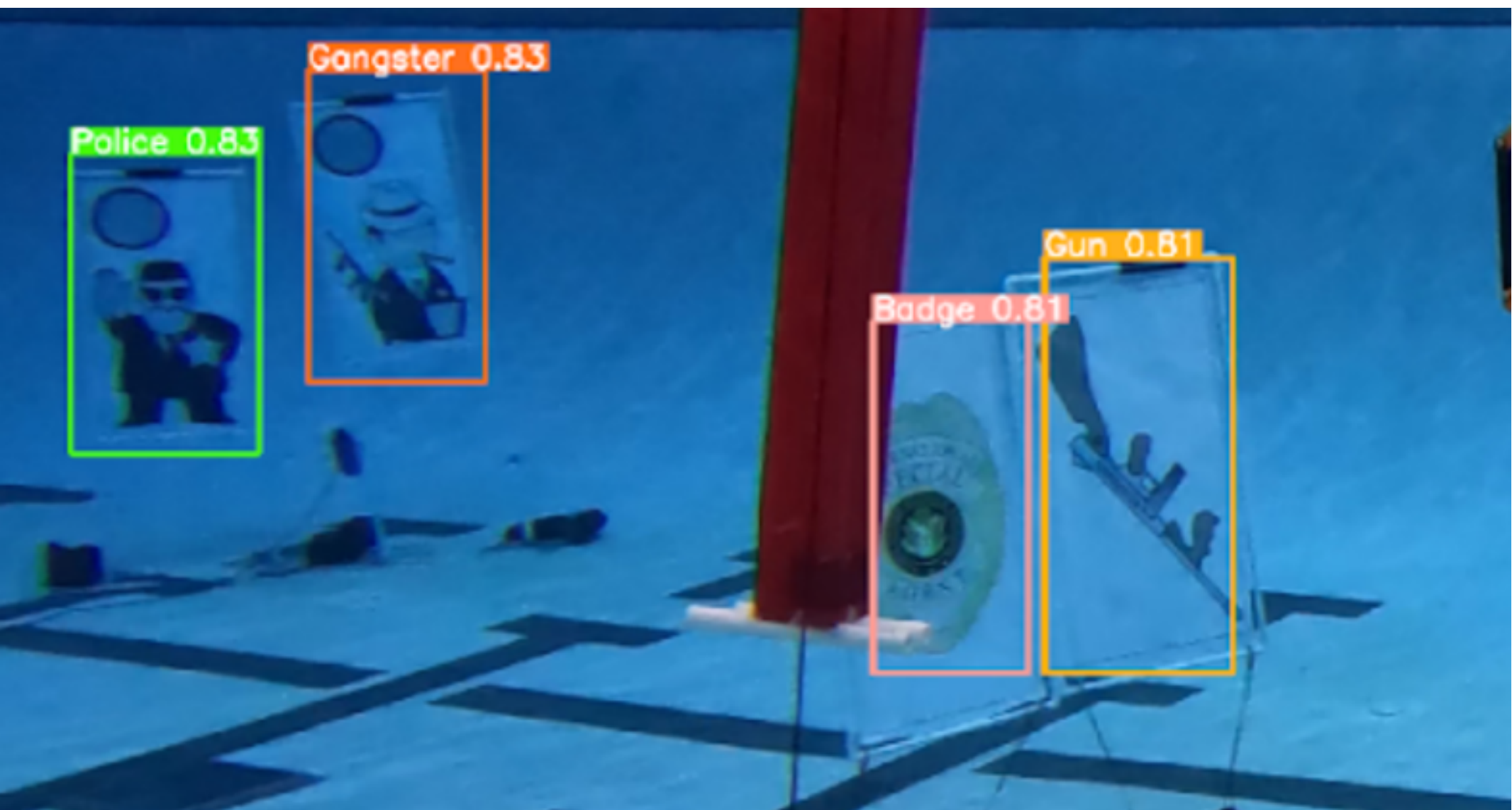


Figure 3: Detección de la clase objeto Police, Gangster, Badge y Gun.

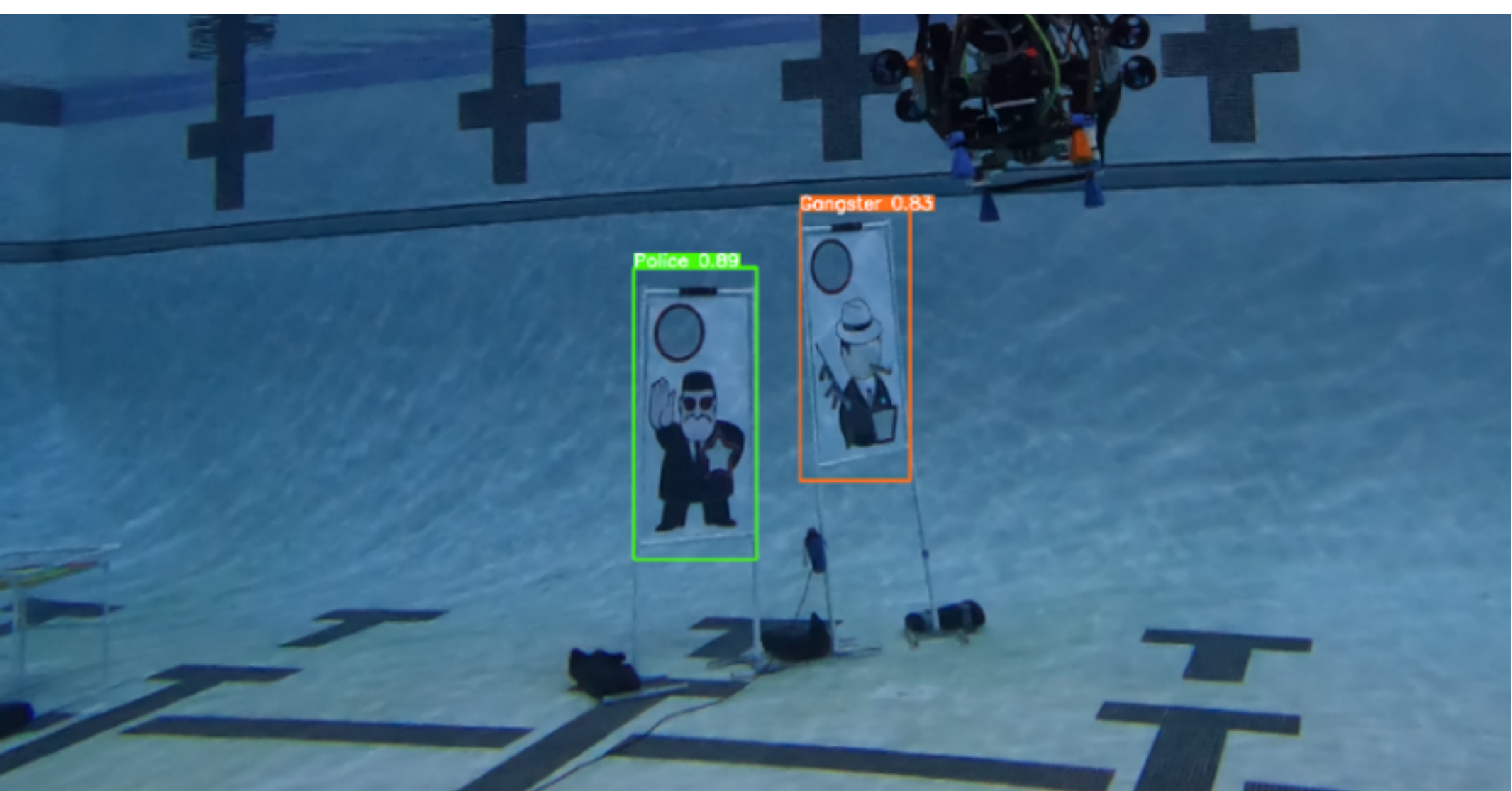


Figure 4: Detección de la clase objeto Police y Gangster.

Conclusiones

Tras realizar las pruebas con los diferentes videos, tanto los que se encuentran dentro del RosBag pudimos comprobar un buen funcionamiento del algoritmo, sin embargo, cuando el algoritmo esta bajo condiciones fuera de las de la competencia puede llegar a perder la eficiencia. Al día de hoy, el algoritmo puede detectar varios objetos bajo el agua con diferentes características tales como formas, tamaños y colores que se encuentran en nuestro Dataset. Todo esto se comunica a través de nodos con arquitectura ROS.

Referencias

Calvo, D. (2017, July 13). Clasificación de Redes neuronales artificiales. Diego Calvo. Retrieved October 1, 2022, from <https://www.diegocalvo.es/clasificacion-de-redes-neuronales-artificiales/>
Matplotlib v3.4.3 Documentation. matplotlib.org. 2021.Retrieved from <https://pypi.org/project/matplotlib/>
OpenCV Open Source Computer Vision v4.5.2. 2020. Documentation.Retrieved from https://docs.opencv.org/4.5.2/d6/d00/tutorial_py_yolo.html
ROS. (2022). Wiki: Documentation. ROS.org. <http://wiki.ros.org>
Ultralytics "YOLOv5" 2021.Retrieved from <https://github.com/ultralytics/yolov5>

Agradecimientos

Este trabajo fue patrocinado por: Techmake, SBG Systems, Google, IFM efector, RoboNation, Velodyne LiDAR, NVIDIA, Akky, ZF Group, Güntner y Siemens. Y por último a todos los miembros y colaboradores dentro del equipo Vanttec, al Dr. Herman Castañeda y el apoyo del Tecnológico de Monterrey.

Contacto

Claudia Torres (a01284940@tec.mx)
Hortencia Ramírez (a01750150@tec.mx)
Diego del Valle (a01284940@tec.mx)
Vanttec (vanttec@servicios.tec.mx)

Vanttec

