Segmentación de cuerpos de agua con imágenes satelitales usando U-NET

Leonardo Torres Damián, Renzo Bedriñana Orozco y Mauricio Ochoa Samaniego

I. Introducción

En los últimos años, el recurso del agua se ha visto afectado por el cambio climático y la necesidad a nivel mundial ocasionando tanto problemas socioeconómicos como ecológicos. Por ello, se necesitan herramientas que extraigan información sobre nuevas fuentes de agua para así solventar dichos problemas.

Es importante contar con información actualizada y precisa acerca de los cuerpos de agua en superficies para identificar y evaluar el rol que toman en los ecosistemas dentro del contexto de la supervivencia humana y el cambio climático [1].

Sin embargo, la correcta identificación de cuerpos de agua no es un problema trivial debido a los complejos terrenos, métodos usados para clasificación y los procesos de teledetección actualmente utilizados [1]. Para abordar este problema, existen diversas propuestas de técnicas de aprendizaje supervisado con el fin de detectar y segmentar adecuadamente cuerpos de agua.

Actualmente, existe una gran cantidad de métodos que permiten la detección de cuerpos de agua, pero muchos de ellos no son precisos. El *Deep Learning* nos permite entrenar y adaptar un modelo según el problema que se tenga usando una gran cantidad de muestras. Uno de los modelos más usados son las redes convolucionales propuestas por Yann LeCun. Hoy en día, existen diversas arquitecturas de redes convolucionales, como SegNet y DenseNet las cuales buscan solventar el problema propuesto.

Es por este motivo que en este estudio proponemos introducir el uso de la arquitectura U-NET para la segmentación de cuerpos de agua con el fin de medir su desempeño dentro de un campo distinto a la de medicina, donde ha sido tradicionalmente utilizado y así poder compararlo con otros modelos actualmente empleados.

La organización de esta primera presentación se divide en la sección de Estado del Arte, donde se explicará brevemente los estudios relacionados y aportes científicos realizados respecto al objeto de estudio y la Metodología que proponemos en este estudio. En esta última sección, detallaremos el proceso de preprocesamiento de datos, utilización de técnicas de ecualización y refinamiento de imágenes, explicación de la arquitectura del modelo propuesto y la estrategia que se empleará para realizar el aprendizaje supervisado. Finalmente, presentamos un diagrama que ilustra las actividades del proceso propuesto.

II. Estado del arte

Tradicionalmente, se emplearon métodos basados en la entropía para la extracción de cuerpos de agua [2] como lo son, el umbral en varias etapas (multi-staged threshold), el árbol de decisiones basado en conocimiento (knowledge-based decision-tree classification), las mediciones geo-morfológicas (geo-morphological measurement), PCA (Principal Component Analysis) que es una solución de machine learning que también fue utilizada para resolver esta problemática junto con algunas variaciones de esta.

En cuanto a la implementación de Deep Learning, se han empleado soluciones con modelos basados en ResNet, VGG, SegNet, NDWI, entre otras. Donde la que mejores resultados obtuvo es la DenseNet propuesta en el estudio de referencia [1]. Actualmente, el valor F1(F1 score) más alto es de 0.872 ± 0.020, alcanzado por la DenseNet [1].

III. Metodología

A. Dataset

En el conjunto de imágenes, encontramos fotografías de cuerpos de agua de distintos volúmenes, sean ríos, lagos, lagunas, entre otros. Este dataset cuenta con 2516 imágenes tomadas por el satélite *Sentinel-2*. Las imágenes están en formato jpg y presentan tres canales de colores. La resolución de las imágenes del corpus varía de entre 80 y 2200 pixeles de ancho y largo. Cada imagen cuenta con una máscara en blanco y negro en donde el color blanco representa agua mientras que el color negro representa algo que no sea agua. Estas máscaras servirán como *labels* para el entrenamiento. Las máscaras fueron generadas calculando el NWDI (Normalized Water Difference Index).

Una dificultad inicial que identificamos en el conjunto de datos es la gran diversidad de cuerpos de agua que presentan. Se hipotetiza que el modelo a entrenar tendrá mayor facilidad de identificar cuerpos de mayor volumen (como lagos y lagunas), mientras que será más impreciso al identificar a los de menor volumen como ríos y canales. Otra dificultad encontrada en el dataset es la gran variabilidad de las resoluciones de las fotografías. Hipotetizamos que esta gran diferencia en tamaños tendrá un impacto negativo en el aprendizaje del modelo de red neuronal.

Figura 1. Imagen tomada por el satélite Sentinel-2 (izquierda) y máscara correspondiente generada con el índice NWDI (derecha).

B. Procesamiento de Imágenes

Utilizaremos la librería OpenCV para el preprocesamiento de imágenes. En el proceso de uniformización, se transformará el dataset para que todas las imágenes tengan la resolución de 128x128 píxeles. Además, las imágenes presentan una escala RGB de tres canales.

Por otro lado, se empleará la técnica de *Histogram Equalization* para refinar la distribución de intensidad de la imagen. Además, se evaluará la utilización de data augmentation para el entrenamiento del modelo ya que, en principio, se dispone de una buena cantidad de imágenes en el dataset.

C. Medida de Calidad

Para este estudio, se empleará la métrica del valor F1(F1 score) y la precisión (Accuracy) en el conjunto de validación de datos. Se optó por estas métricas debido a su simpleza de análisis y además porque son las utilizadas en el estudio que tomamos como referencia [1].

D. Modelo Base

La arquitectura U-NET tiene como base a las *fully convolutional networks*. Una de sus modificaciones más importantes es la utilización de un un *Encoder*, que extrae características y reduce las dimensiones de las imágenes y de un *Decoder* que permite restaurar el tamaño original de las imágenes y propagar información a capas con resoluciones superiores. El *Encoder* está conformado por ocho capas convolucionales con matrices kernel de 3x3, cada una acompañada de una función de activación ReLU. Además, cada dos convoluciones se realiza un max pooling de 2x2. Por otro lado, el *Decoder* está conformado por otras ocho capas convolucionales con kernels de 3x3, cada una acompañada de la función de activación ReLU. Asimismo, cada dos convoluciones se realiza un upSampling de 2x2.

Esta arquitectura fue propuesta inicialmente para la segmentación de imágenes biomédicas. Sin embargo, en el presente estudio evaluaremos su uso en el contexto de la segmentación de cuerpos de agua.

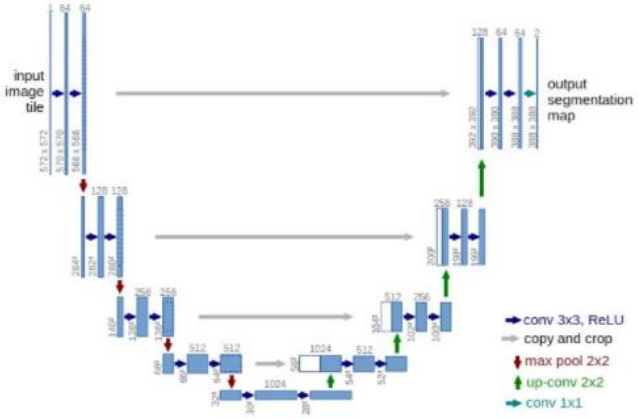

Figura 2. Imagen de la arquitectura del modelo base (U-NET)[4]

E. Estrategia de Validación

El dataset se divide en tres grupos fundamentales:

- El conjunto de datos de prueba representa el 10% del total del corpus de datos. Estas imágenes no se verán involucradas en el proceso de entrenamiento y, al finalizar el mismo, realizaremos predicciones con ellas para evaluar el desempeño del modelo.
- El conjunto de datos de entrenamiento representa el 80% del resto del corpus (luego de extraer al grupo de datos de prueba). Estas imágenes son las que se utilizarán para el entrenamiento supervisado de la red. Además, emplearemos técnicas de data augmentation con este conjunto de datos.
- El conjunto de datos de validación representa el restante 20% del corpus (luego de la extracción del grupo de datos de prueba). Estas imágenes se emplearán en el entrenamiento del modelo para evaluar su desempeño durante el mismo.

F. Proceso Propuesto

A continuación, presentamos el diagrama que representa el proceso desde la carga de imágenes del dataset hasta la evaluación de la métrica de accuracy luego de realizar el entrenamiento supervisado.

Figura 3. Imagen del flujo del proceso propuesto

IV. Experimentación y Resultados

A. Descripción del conjunto de datos

A.1 Descripción del conjunto de datos

El conjunto de datos de las fotografías tomadas por el satélite *Sentinel-2* y sus máscaras fueron obtenidas de un repositorio de Kaggle.

A.2 Características del conjunto de datos

El conjunto de datos consta de 2516 imágenes satelitales. Debido a que el tamaño de cada imagen era variable y en algunos casos eran muy grandes, se optó por uniformizar el tamaño de las imágenes a 128x128 píxeles. Después de estandarizar las imágenes, se aplicó la técnica *Histogram Equalization*, la cual permite aumentar el contraste y redistribuir las intensidades de manera uniforme. Esto permite que las áreas de menor contraste local obtengan un contraste mayor.

Todo este proceso se realizó en un script aparte llamado "preprocessing_images", con la finalidad de reducir el tamaño de las imágenes, de agilizar la lectura de datos y finalmente, de ordenar las imágenes y máscaras en sus respectivos directorios.

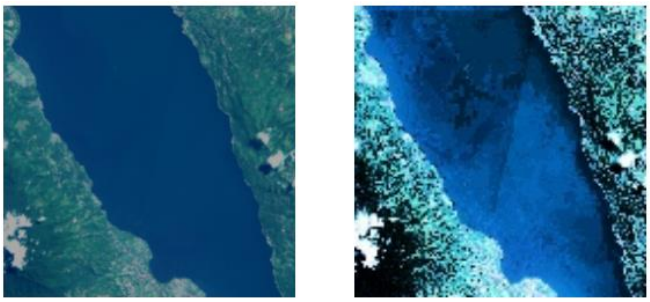

Figura 4. Fotografía original (1291 x 1283 píxeles) tomada por el Satélite-2 (izquierda). Fotografía luego uniformizar el tamaño a 128x128 píxeles y de aplicar la técnica Histogram Equalization (derecha).

Luego de ejecutar el script, se subió el conjunto de datos a un repositorio de GitHub para facilitar la lectura desde el entorno de experimentación empleado (Google Colaboratory).

Una vez el conjunto de datos fue descargado y descomprimido, se le aplicó una normalización. Como ya se mencionó, cada fotografía tiene una máscara correspondiente. Las fotografías y sus máscaras están en formato JPG con valores discretos entre 0-255 en 3 canales. Por ello, al realizar la normalización, se modificaron los valores de cada fotografía para que se encuentren entre 0 y 1. Para este fin, empleamos funciones de la librería OpenVision en Python.

Por otro lado, se encontró que las máscaras, a pesar de solo tener los colores blanco y negro, tenían valores de ruido que eran diferentes de 0 y 1 después de normalizarlas. Esto es muy común al trabajar con imágenes en el formato JPG. Por ello, fue necesario establecer un *threshold* o umbral que permita establecer si un píxel debería tener un valor de 0 o 1. Utilizamos el valor medio 0.5 como umbral y le aplicamos una función a la máscara para convertir todos los píxeles en alguno de estos dos valores. Este procedimiento es fundamental para realizar el entrenamiento.

Finalmente, se decidió emplear la técnica de data augmentation para incrementar en 20% el volumen del corpus de datos del conjunto de entrenamiento. Para este fin, se utilizó el framework de OpenVision en Python para especular (*horizontal flip*) la imagen original y su máscara correspondiente.

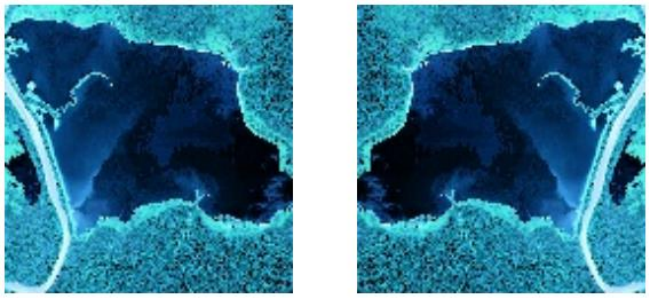

Figura 5. Imagen pre-procesada tomada del corpus de datos (izquierda). Imagen luego de aplicar *horizontal flip*. Se realizó el mismo procedimiento con la máscara correspondiente.

A.3 Distribución del conjunto de datos

El conjunto de datos fue distribuido en tres grupos: entrenamiento, validación y prueba. Luego de aplicar data augmentation, el grupo de entrenamiento consta de un total de 2173 imágenes, el de validación de 544 imágenes y el grupo de prueba contiene 252 imágenes del dataset.

B. Descripción del entorno de experimentación empleado (hardware y software)

El entorno de experimentación empleado para el entrenamiento del modelo fue el proporcionado por Google Colab, el cual se vio conveniente a utilizar debido a que contaba con mejores características computacionales que el equipo disponible por el grupo.

C. Línea Base

El modelo DenseNet [1], el cual es el mejor en el estado del arte, utilizó las imágenes del satélite GF-1 que están enfocadas en los conjuntos de ríos en el Lago Poyang. Dicho modelo obtuvo una precisión de 0.961, un recall de 0.904 y un valor F1 de 0.931 y fue entrenado con un total de 40 épocas con mil iteraciones cada una. Además, para la generación de máscaras para el entrenamiento emplearon la normalización NDWI gracias a los 4 canales que brindaba dicho dataset y, para el umbral, se utilizó el método Otsu el cual permite hallar el umbral óptimo para generar los valores de 0 y 1 de las máscaras.

D. Reporte de entrenamiento del modelo

Al entrenar el modelo, probamos con diferentes configuraciones y valores de hiperparámetros. Se probó variar el número de capas convolucionales, el padding, strides y el tamaño de las matrices kernel en los módulos de encoder y decoder. Cabe resaltar que se empleó una capa convolucional final con función de activación *sigmoid* para predecir un número real entre 0 y 1 si el píxel analizado es o no es parte de un cuerpo de agua.

En el experimento, se mantuvieron constantes los siguientes parámetros:

- Para la función de pérdida utilizamos *binary crossentropy*.
- Como optimizador, empleamos Adam con un *learning rate* de 0.001.
- Como métricas, empleamos el accuracy y el valor F1 en el conjunto de datos de validación.

Para todas las configuraciones anteriores, se realizó un entrenamiento de 50 épocas con *batch size* de 128 imágenes por lote. En cuanto a tiempo de ejecución de entrenamiento, este fue de aproximadamente 30 minutos por cada configuración. Con el fin de aligerar este procedimiento, se empleó el método de *prefetch* en los tensores de entrenamiento y validación.

Los mejores resultados obtenidos fueron de 89.25% para el accuracy y 82% para el F1 score. Esto se logró utilizando la configuración de 16 capas convolucionales en el encoder y 16 capas de deconvoluciones en el decoder (dos capas por cada número de filtros: 64, 128, 256 y 512) con kernels de tamaño 3x3 y función de activación relu en todas las capas. La configuración final cuenta con 18,807,809 parámetros entrenables.

A continuación, presentamos las gráficas para la función de pérdida y las métricas de precisión y valor F1 durante el entrenamiento del modelo.

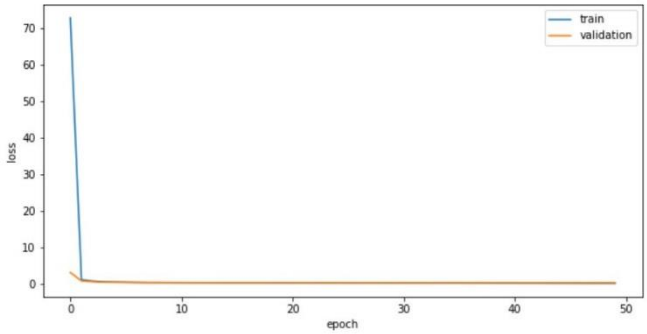

Figura 6. Gráfica de la función de pérdida durante el entrenamiento. La curva azul representa al conjunto de entrenamiento, mientras que la curva naranja representa al conjunto de validación.

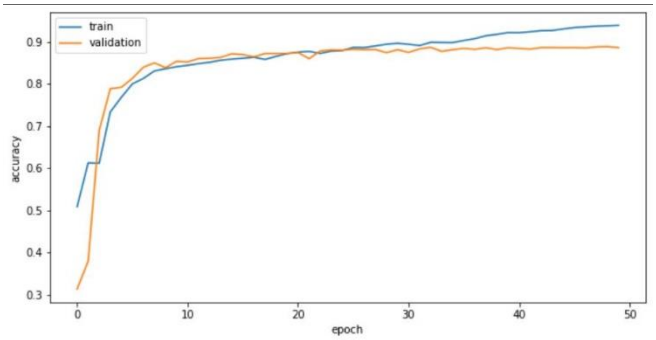

Figura 7. Gráfica de la métrica de precisión durante el entrenamiento. La curva azul representa al conjunto de entrenamiento, mientras que la curva naranja representa al conjunto de validación.

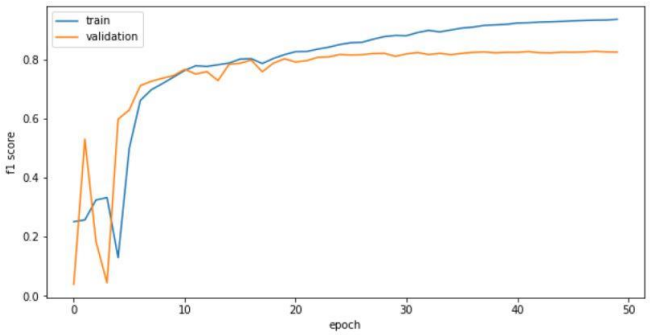

Figura 8. Gráfica de la métrica valor f1 durante el entrenamiento. La curva azul representa al conjunto de entrenamiento, mientras que la curva naranja representa al conjunto de validación.

Al utilizar la función de activación *sigmoid*, obtenemos como predicción del modelo una matriz cuyos píxeles tienen un valor real entre 0 y 1 que representan si dicho píxel pertenece o no a un cuerpo de agua. Dado que para nuestro interés es necesario un valor que sea estrictamente 0 o 1, aplicamos una función de transformación con un *threshold* de 0.35 que permite establecer valores fijos. Esta función convertirá a 1 todos los píxeles que sean mayores o iguales a 0.35 y al valor de 0 a los píxeles menores a 0.35.

A continuación, mostramos un ejemplo de predicción realizada con una imagen del conjunto de prueba. Esta muestra no ha sido utilizada durante el entrenamiento. Se muestra la máscara original, la predicción del modelo empleando la función de activación *sigmoid*, y la predicción del modelo luego de aplicarle el *threshold* explicado anteriormente.

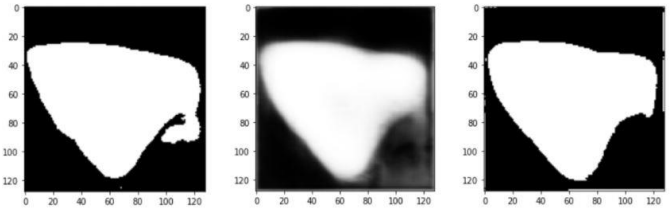

Figura 9. Máscara original tomada del conjunto de datos de prueba (izquierda). Predicción del modelo empleando la función de activación *sigmoid* (centro). Predicción del modelo luego de aplicar una función que uniformiza los valores de los píxeles empleando un *threshold* de 0.35.

V. Discusión

A. Comparación de línea base y resultados propios

Se observa que el modelo logra un máximo de 82% para el F1 score durante el entrenamiento. Si bien no se logró alcanzar el valor de 93.1% del estado del arte, resaltamos que el modelo realizó predicciones por encima de este valor en algunos datos de prueba. Particularmente, encontramos que las predicciones son bastante buenas en las imágenes de cuerpos de agua más voluminosos como lagos y lagunas, mientras que hubo mayores dificultades en las imágenes que presentaban regiones muy pequeñas de volúmenes de agua.

En estos casos, el modelo asume incorrectamente que toda la región donde se encuentran estos pequeños puntos corresponde a un cuerpo de agua.

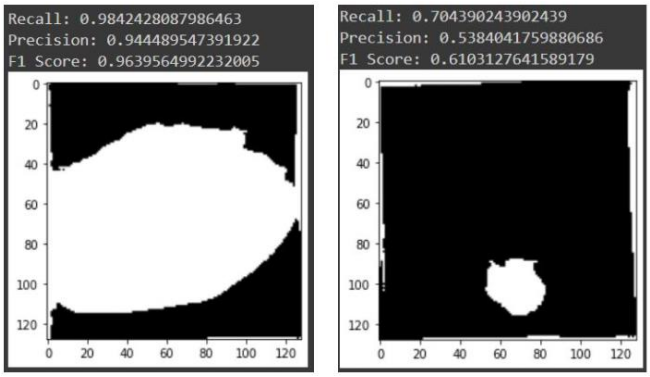

Figura 10. Buena predicción del modelo (izquierda). Predicción no tan acertada (derecha).

B. Interpretación de los datos usados

Los datos que se emplearon fueron un conjunto de 2516 imágenes en formato JPG las cuales solo utilizan 3 canales de color y tenían una gran variedad de dimensiones en comparación con las imágenes del dataset del satélite GF-1 las cuales contenían 4 canales y constaba de resoluciones altas. Este cuarto canal se emplea para guardar valores cerca del infrarrojo, lo cual ayuda al modelo a poder identificar mejor los cuerpos de agua. Además, la gran varianza de dimensiones del dataset ocasiona conflictos al momento de estandarizar dichas dimensiones a 128x128 porque algunas imágenes cuentan con dimensiones muy disparejas como 1130x121, lo que ocasiona pérdida de información y data que ya no puede ser interpretada correctamente.

C. Interpretación de los resultados obtenidos

En líneas generales, el modelo nos da predicciones buenas en su clasificación por píxel de las imágenes. Como se mencionó anteriormente, es particularmente preciso en las imágenes que representan cuerpos de agua de mayor volumen mientras que las principales dificultades surgen en regiones muy angostas (como riachuelos y canales) o en regiones muy pequeñas.

Además, es preciso resaltar que las predicciones realizadas se manifiestan en números reales que representan una probabilidad por píxel. Al introducir una variable de umbral para uniformizar la máscara de la predicción, se altera indirectamente el grado de precisión del modelo. Se piensa que este valor de *threshold* debería ser un parámetro entrenable más del modelo que permita transformar la máscara de la predicción de manera que se maximice el valor F1.

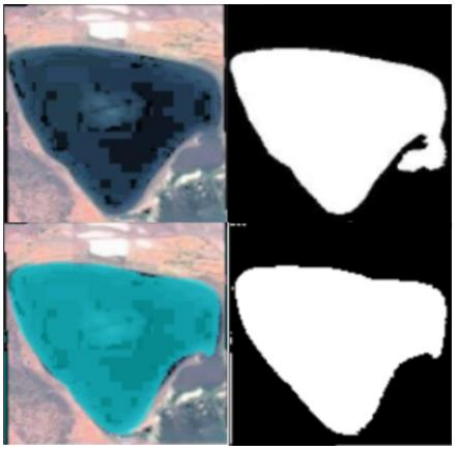

Figura 11. Imagen original del conjunto de prueba a predecir (superior izquierda). Máscara original del conjunto de prueba (superior derecha). Predicción del modelo luego de aplicar la transformación con umbral de 0.35 (inferior derecha). Predicción del modelo sobre la imagen original (inferior izquierda).

Finalmente, utilizamos el modelo para predecir una fotografía satelital de la costa de Lima, Callao. Esta imagen fue obtenida de un satélite de la National Aeronautics and Space Administration (NASA). Se observa que el modelo tiene dificultades para identificar regiones con espuma de mar. Esto es de esperarse ya que las imágenes utilizadas en el entrenamiento no contienen esta característica. Sin embargo, el modelo llega a identificar correctamente la pequeña región de tierra de forma ovalada que se encuentra en medio del mar.

Figura 12. Fotografía de la costa de Lima, Callao luego de aplicar el preprocesamiento (izquierda). Predicción de la máscara del modelo luego de aplicar la transformación con umbral (centro). Predicción del modelo sobre la imagen procesada (derecha).

D. Dificultades encontradas en los modelos ensayados

Dentro de las imágenes utilizadas para la prueba del modelo, identificamos que las regiones que representan una dificultad mayor son aquellas que son estrechas, como ríos, riachuelos y canales de agua. Otros cuerpos de agua que representan un reto para el modelo son los que tienen un volumen muy pequeño, como pozos que se encuentran en medio de una superficie.

Asimismo, se pudo comprobar que las regiones de agua que tenían un bajo contraste respecto a otro tipo de superficies no pudieron ser correctamente identificadas. Esto se debe a que el modelo predice probabilidades pequeñas de que estas zonas representan estructuras de agua, por lo que el umbral

definido de 0.35 realiza una suerte de “truncamiento”, resultando en la pérdida de estas regiones.

Adicionalmente, al probar el modelo con la fotografía de la costa del Callao, encontramos que no se identificaron las zonas correspondientes a espuma marina. Se hipotetiza que este resultado es producto de no haber incluido imágenes de océanos con espuma marina en el dataset. Sin embargo, consideramos que identificar este tipo de características en cuerpos de agua puede representar un reto adicional.

E. Posibles mejoras

Una de las principales limitantes al momento de construir y entrenar el modelo fue la falta de un equipo computacional mucho más potente que el brindado por Google Colaboratory, por este motivo es que tuvimos que reducir la cantidad de capas convolucionales. Así pues, una mejora potencial sería la de construir el modelo con mejores elementos computacionales, para poder aplicar el modelo completo que se tuvo como referencia (U-NET) y obtener resultados más precisos.

Otra posible mejora sería trabajar con un umbral entrenable en el modelo. En el estudio de referencia, se establecía un *threshold* que se iba ajustando en cada época para encontrar un valor óptimo. En el presente trabajo consideramos este valor como 0.35 ya que, empíricamente, fue el que nos daba mejores resultados en la métrica de F1 score. Sin embargo, consideramos que este valor podría no ser el más óptimo y por lo tanto, debería definirse de una manera más rigurosa.

Finalmente, consideramos que el formato de JPG no es el más adecuado para trabajar con cuerpos de agua, puesto que información de otros canales (como el infrarrojo) son bastante útiles para las tareas de segmentación en cuerpos de agua. Una mejora posible sería utilizar un dataset que disponga de imágenes con mayor cantidad de características.

VI. Conclusiones y trabajos futuros

Considerando todos los aspectos, estamos bastante satisfechos con los resultados obtenidos en este trabajo. Tradicionalmente, la segmentación de estructuras como cuerpos de agua se realizan empleando topologías como la *DenseNet* y las Fully *Convolutional Networks*. Sin embargo, postulamos que la arquitectura de U-NET con retroalimentación puede ser bastante efectiva en este contexto.

Nuestro modelo es una versión simplificada de la U-NET originalmente propuesta. A pesar de esta limitación, pensamos que un accuracy de 89% y F1 score de 82% son resultados bastante aceptables considerando las limitaciones tecnológicas de este estudio.

Por otro lado, pensamos que otra limitación importante es el dataset. Desafortunadamente, no pudimos acceder al mismo conjunto de datos empleado en el estudio de referencia. Estas imágenes contaban con información adicional en un canal extra y además presentaban mayores resoluciones. Consideramos que estas características adicionales facilitaron el entrenamiento del modelo que se presentó en el estudio de referencia.

En futuros trabajos, sería pertinente realizar la experimentación con un hardware que permita la implementación del modelo U-NET completo (4 capas convolucionales por cada número de filtros) utilizando el mismo dataset y las mismas dimensiones de imágenes del estudio de referencia (con el fin de realizar un benchmark más justo y adecuado). Además, consideramos que el *threshold* empleado en la transformación de las predicciones del modelo toma un rol fundamental en el estudio ya que afecta directamente a la métrica del valor F1. Por lo tanto, dicho parámetro debería ser configurado para trabajarse como un parámetro entrenable más del modelo.

Referencias

[1] Yang C., Rongshuang F., Xiucheng Y., , Jingxue W., Aamir L. Extraction of Urban Water Bodies from High-Resolution Remote-Sensing Imagery Using Deep Learning. MDPI water. (2018).

[2] Zhaohui Z., Prinet V., Songde M. Water Body Extraction from Multi-Source Satellite Images. IGARSS. IEE International Geoscience and Remote Sensing Symposium. (2003)

[3] Yousefi P., Jalab H., Ibrahim R., Mohd N., Ayub M., Gani A. Water-Body Segmentation in Satellite Imagery Applying Modified Kernel K-Means. Malaysian Journal of Computer Science (2018).

[4] Ronneberger O., Fischer P., Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation. Cornell University (2015).

[5] Wang G., Wu M., Wei X., Song H. Water Identification from High-Resolution Remote Sensing Images Based on Multidimensional Densely Connected Convolutional Neural Networks. Nanjing University of Information Science and Technology (2020)

https://github.com/leonardtd/TAC-Segmentacion-de-cuerpos-de-agua