

DETECCIÓN AUTOMÁTICA DE CUERPOS DE AGUA DEL BAJÍO UTILIZANDO PARÁMETROS MORFOMÉTRICOS OBTENIDOS DE IMÁGENES SATELITALES Y PROCESADOS CON REDES NEURONALES

AUTOMATIC DETECTION OF WATER BODIES OF EL BAJÍO USING MORPHOMETRIC PARAMETERS OBTAINED FROM SATELLITE IMAGES AND PROCESSED WITH NEURONAL NETWORKS

José Daniel Fernández Peña

Tecnológico Nacional de México / IT de Celaya, México
M2003019@itcelaya.edu.mx

Claudia A. Gallegos Sánchez

Tecnológico Nacional de México / IT de Celaya, México
cgallegos.ia@gmail.com

José A. Padilla Medina

Tecnológico Nacional de México / IT de Celaya, México
alfredo.padilla@itcelaya.edu.mx

Alejandro I. Barranco Gutiérrez

Tecnológico Nacional de México / IT de Celaya, México
israel.barranco@itcelaya.edu.mx

José A. Vázquez López

Tecnológico Nacional de México / IT de Celaya, México
antonio.vazquez@itcelaya.edu.mx

Pedro J. Correa Caicedo

Tecnológico Nacional de México en Celaya
D1803025@itcelaya.edu.mx

Recepción: 28/septiembre/2021

Aceptación: //2021

Resumen

La conservación de las masas de agua superficiales es un tema crucial para el desarrollo socioeconómico de los asentamientos humanos. Por tanto, el seguimiento constante de la distribución territorial del agua es cada vez más una prioridad.

Las imágenes procesadas fueron obtenidas de la misión satelital SENTINEL 2 que provee su información en base a 13 bandas multispectrales, de las cuales se usaron las bandas B3 (verde) y B8 (infrarrojo cercano) en el cálculo del parámetro de identificación de agua NDWI. En este trabajo se propone un sistema de identificación de masas de agua mediante redes neuronales a partir de una base de datos de clasificadores que permite a la red discriminar píxel a píxel si existe o no presencia de agua. La red se implementa en dos ambientes diferentes: Matlab y Google Colaboratory, dos plataformas que logran obtener buenos resultados en el diseño de modelos de redes neuronales. En las pruebas realizadas queda demostrada la capacidad predictiva del sistema implementado, logrando un rendimiento por encima del 90% de precisión, cumpliendo con los objetivos de identificación.

Palabras Clave: cuerpos de agua, visión artificial, Google Colaboratory, Matlab, red neuronal.

Abstract

The conservation of surface water bodies is a crucial issue for the socioeconomic development of human settlements. Therefore, the constant monitoring of territorial water distribution is increasingly a priority. The processed images were obtained from the SENTINEL 2 satellite mission that provides its information based on 13 multispectral bands, of which bands B3 (green) and B8 (near infrared) were used in the calculation of the water identification parameter NDWI. In this work, a system for identifying bodies of water using neural networks is proposed from a database of classifiers that allows the network to discriminate pixel by pixel whether it is in the presence of water or not. The network is implemented in two different environments: Matlab and Google Colaboratory, two platforms that achieve good results in the design of neural network models. In the tests carried out, the predictive capacity of the implemented system is demonstrated, achieving a performance above 90% accuracy, meeting the identification objectives.

Keywords: *water bodies, artificial vision, Google Colaboratory, Matlab, neural network.*

1. Introducción

El agua es un recurso natural esencial para la vida, pues provee soporte a toda la biodiversidad del planeta. Su importancia abarca incluso la sostenibilidad de ciudades y comunidades, en las cuales se debe lograr una integración económico-biológica para su desarrollo oportuno [Kafy, 2018]. La conservación de los cuerpos de agua superficiales es tema crucial para el desarrollo económico-social de asentamientos humanos [Utton, 2019].

Estos cuerpos de agua están influenciados por el clima, cambios de cobertura y uso de la tierra, y el medio ambiente en general. Las afectaciones en estos recursos pueden provocar contaminación, sequías e inundaciones [Zhang et al., 2018]. Por lo que el monitoreo constante de la distribución de agua territorial constituye una tarea de impacto.

La percepción remota ha constituido, en estos últimos años, una herramienta muy útil para aplicaciones en monitoreo ambiental debido a su rentabilidad y capacidad para superar las limitaciones de los métodos convencionales [Ko et al., 2015], [Chawla et al., 2020]. Plataformas como la serie LANSAT, Sentinel, entre otros, brindan la posibilidad de obtener imágenes en múltiples longitudes de onda del espectro electromagnético [Pahlevan et al., 2019]. Mediante el procesamiento de estas imágenes es posible clasificar e identificar cambios en un área de interés determinada.

En conjunto con la teledetección satelital, se han unido otras tecnologías como las redes neuronales que han llevado a un nuevo nivel la clasificación e identificación de patrones en imágenes multiespectrales [Yuan et al., 2020], [Cheng et al., 2020], [Hong et al., 2020]. Este nuevo concepto se encuentra en la base de muchos algoritmos de aprendizaje profundo: modelos (redes) compuestos por un conjunto de capas que transforman los datos de entrada en resultados y al mismo tiempo adquieren conocimiento que les permitirán describir características de alto nivel [Litjens et al., 2017].

En este trabajo se propone un sistema de identificación de cuerpos de agua mediante redes neuronales a partir de una base de datos de clasificadores que permite a la red discriminar píxel a píxel si está en presencia o no de agua.

2. Métodos

La implementación se desarrolla a partir de una serie temporal de ocho imágenes satelitales capturadas en el área del Bajío en México, específicamente en una zona del estado de Guanajuato comprendida entre Celaya, Apaseos, Acámbaro y Querétaro, como se muestra en la figura 1. Estas imágenes se obtuvieron de la misión Sentinel-2 y datan del 2018.

Sentinel-2 es una misión que consiste en un par de satélites, lanzados en 2015 y 2017 respectivamente. Cada satélite trae consigo un generador de imágenes multiespectral o MSI (*MultiSpectral Imager*) con una franja de 290km y provee información en 13 bandas espectrales, como se muestra en la tabla 1, que se distribuye desde la región del infrarrojo cercano o visible hasta la región del infrarrojo de onda corta, incluidas cuatro bandas a 10 m, seis bandas a 20 m y tres bandas a 60 m de resolución espacial [Xie et al., 2019].

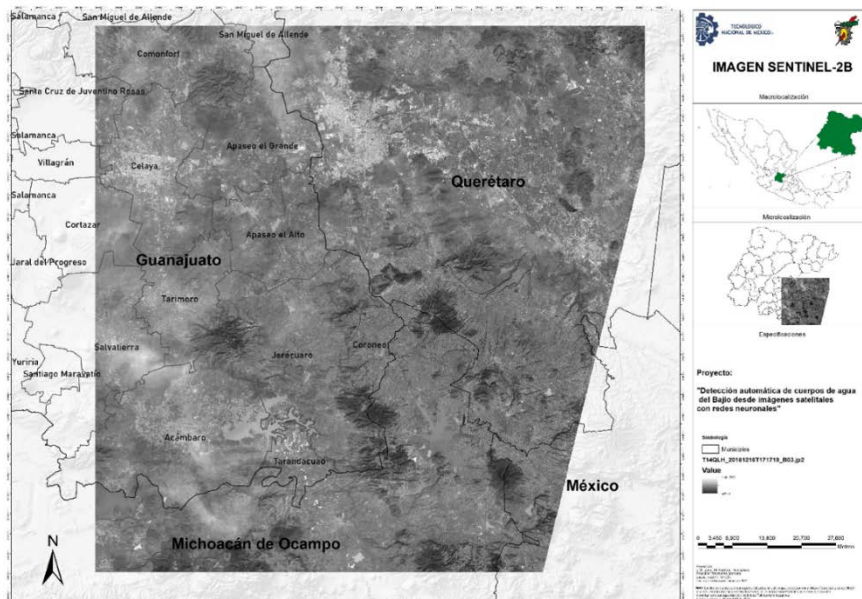


Figura 1 Área de estudio.

Herramientas de software utilizadas

La meta que se persigue en este trabajo consiste en implementar una red neuronal que logre discriminar píxel a píxel en una imagen determinada si existe o no presencia de agua. Pero lo es, además, lograr su implementación en el ambiente

de Matlab y Colaboratory, dos softwares que en la actualidad se hacen muy presente en el desarrollo de investigaciones y aplicaciones basadas en Inteligencia Artificial.

Tabla 1 Configuración de bandas – SENTINEL 2.

Bandas	Centro de Banda (nm)	Ancho de Banda (nm)	Resolución espacial (m)
B1	443	20	60
B2	490	65	10
B3	560	35	10
B4	665	30	10
B5	705	15	20
B6	740	15	20
B7	783	20	20
B8	842	15	10
B8A	865	20	20
B9	945	20	60
B10	1375	30	60
B11	1610	90	20
B12	2190	180	20

Matlab

MATLAB es un entorno de computación y desarrollo de aplicaciones totalmente integrado. Enfocado en el análisis numérico, cálculo matricial, procesamiento de señales y visualización gráfica. Dispone de un amplio set de programas de apoyo especializados, denominados *Toolboxes*, que amplían su funcionalidad.

Para el caso que ocupa, se hizo uso del *Deep Learning Toolbox*. Este *toolbox* es utilizado en el diseño e implementación de redes neuronales con algoritmos, modelos pre-entrenados y aplicaciones. Dentro del mismo, se pone a disposición la herramienta *nprtool (Neural Net Pattern Recognition Tool)*, que permite generar el código de una red de dos capas prealimentada a partir de parámetros configurables como son: entradas, salidas y cantidad de neuronas en las capas ocultas; además utiliza como función de activación neuronal la sigmoide logarítmica [MATLAB, 2019].

Google Colaboratory

De igual manera, se implementó una red neuronal en el ambiente de Google Colaboratory. Esta es una tecnología creada con base en Jupiter Notebooks, una

herramienta de código abierto basada en navegadores web que es posible ejecutar localmente o en la nube.

Una Jupiter Notebook se compone de varias celdas que contienen scripts o código y su resultado aparece embebido en el documento. El resultado de la ejecución de código en la Jupiter Notebook puede presentarse como texto, tablas, gráficas o imágenes. De esta manera se facilita compartir o replicar trabajos científicos en un ambiente controlado y embebido [Randles et al., 2017].

Por su parte, Colab (Google Colaboratory) es un proyecto que tiene como objetivo principal el uso de Machine Learning para la educación y la investigación. Provee un ambiente para la ejecución de Python v2 y v3 con librerías precargadas para Machine Learning e Inteligencia Artificial tales como Tensorflow, Matplotlib y Keras. Se trabaja a través de una Máquina Virtual, puesto que la potencia computacional lo sirve Google a través de su plataforma Google Cloud. Este ambiente es temporal, por lo que al finalizar su trabajo cualquier configuración o dato generado o copiado a su almacenamiento virtual se pierde. Pero, el código de la Jupiter Notebook es preservado en el almacenamiento de la cuenta de Google con la que se inició sesión en primer lugar [Pessoa et al., 2018], [Bisong, 2019].

Obtención de la muestra

Los estudios desarrollados sobre la caracterización de tipos de cobertura en general basan sus consideraciones en parámetros calculados a partir de los valores de reflectancia a diferentes longitudes de onda. Estos índices son utilizados para mejorar la discriminación, reduciendo el efecto del relieve en la caracterización espectral de las diferentes cubiertas.

Como el Índice de Vegetación de Diferencia Normalizada (NDVI, *Normalized Difference Vegetation Index*) que es ampliamente utilizado para mapear vegetación, el Índice de Agua de Diferencia Normalizada (NDWI, *Normalized Difference Water Index*), desarrollado por Stuart McFeeters en 1996, es el parámetro usado para mapear cuerpos de agua. El rango dinámico de este indicador se encuentra entre -1 y 1, donde los valores mayores a 0 son asociados a presencia de agua. Cuanto más cercano es a 1, mayor es el contenido de agua presente. Este índice es posible

calcularlo a partir de las bandas espectrales B3 y B8, cuyas especificaciones se muestran en la tabla 1, correspondientes al verde y al infrarrojo cercano respectivamente, como se presenta en la ecuación 1.

$$NDWI = \frac{B3 - B8}{B3 + B8} \quad (1)$$

Donde:

- **NDWI**: Índice de Agua de Diferencia Normalizada.
- **B3**: Valores de píxel en la banda del verde.
- **B8**: Valores de píxel en la banda del infrarrojo cercano.

A partir de este índice se obtuvo una máscara binaria que permitió obtener la clasificación adecuada de cada píxel en el área de estudio, para obtener los conjuntos de datos necesarios de cada clase.

Posteriormente, se extrajeron datos aleatorios con su respectiva clasificación, píxel a píxel, basado en dos estados: agua presente o sin agua. Mediante este proceso se obtuvieron 40 conjuntos de datos por imagen donde se encontraban la misma cantidad de muestras por clase. Cada conjunto de datos estaba conformado por los valores de píxel de cada componente de color de la imagen, o sea Rojo (R), Verde (G) y Azul (B), como se muestra en la figura 2.

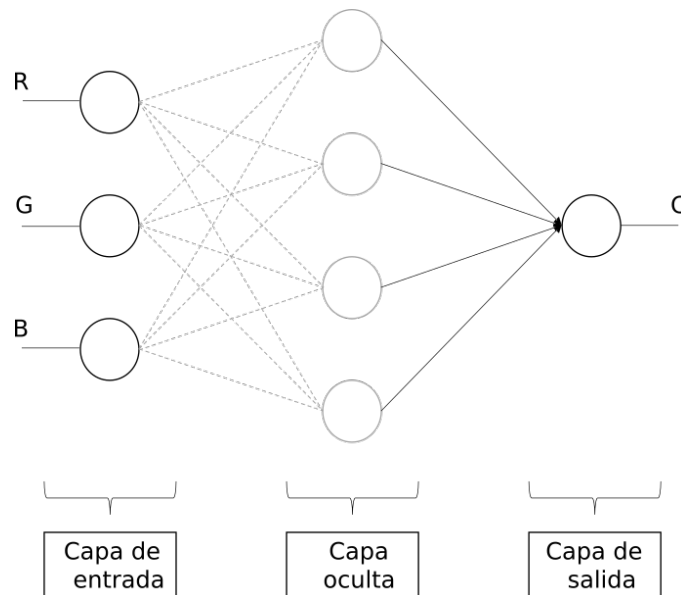


Figura 2 Esquema de la red neuronal.

Entrenamiento y validación de la red neuronal

La etapa de identificación se puede desglosar en dos pasos. Un primer paso, donde se entrena a la red para brindarle la capacidad de encontrar similitudes existentes entre los datos que procesa. En segundo lugar, mediante otro conjunto de datos se valida el algoritmo definido en el entrenamiento.

En la figura 2 se puede observar la arquitectura de una red neuronal:

- Una primera capa de entrada, con tres neuronas que representan el vector de datos de entrada, es la encargada de recibir información del exterior.
- Una serie de capas ocultas, cuyo número no puede ser determinado sin ser probado empíricamente para averiguar qué configuración es la adecuada para un problema concreto. Estas capas ocultas son las encargadas de realizar la transformación de los datos de entrada a través de los pesos que las neuronas tienen en sus conexiones.
- Una capa de salida, que representa el resultado final de la red.

La red neuronal queda representada como se muestra en la ecuación 2.

$$Output = \text{logsig}[W2 * \text{tansig}(W1 * I + b1) + b2] \quad (2)$$

Donde:

- ***I***: entrada de la red
- ***W***: pesos de las neuronas en las capas 1 y 2
- ***b***: bias en las capas 1 y 2
- ***Output***: salida de la red

Usando la herramienta **nprtool** de Matlab puede generarse una red neuronal en pocos pasos:

- Seleccionar los datos de entrada y salida.
- Escoger los porcentajes de muestra para entrenamiento, validación y prueba.
- Definir la cantidad de neuronas en la capa oculta de la red.
- Realizar el entrenamiento hasta que se alcancen los parámetros adecuados.
- Ejecutar la evaluación de la red obtenida.
- Guardar el resultado para su posterior uso.

Por otra parte, en el ambiente de Colab, se desarrolla la red neuronal usando la librería Keras que corre encima del framework Tensorflow. La metodología que se sigue para implementar dicha red es la siguiente [Manaswi et al., 2018]:

- Cargar los datos.
- Preprocesar la información.
- Definir un modelo, creando un modelo secuencial y añadiendo las capas que estructurarán la red.
- Compilar el modelo aplicando la función de pérdida y el optimizador a usar.
- Ejecutar el entrenamiento usando los datos de prueba realizando la llamada a la función fit().
- Evaluar el modelo y ejecutar las predicciones generadas a partir de nuevos datos haciendo llamadas a las funciones evaluate() y predict().
- Guardar el modelo

3. Resultados

El sistema descrito se prueba a partir dos conjuntos de datos:

- El primero es un conjunto de 40 muestras tomadas de 7 imágenes multiespectrales. Este conjunto es utilizado como referencia en la etapa de entrenamiento y validación del sistema. Cada muestra contiene el valor de un píxel, seleccionado aleatoriamente, de cada canal de la imagen RGB y el valor binario de la clasificación dada según la clase asignada.
- El segundo conjunto está formado por 4 imágenes de dimensión 100x100 tomadas de una imagen multiespectral, figura 3. Es el conjunto de datos utilizado para evaluar el correcto desempeño de la red.



Figura 3 Imágenes de prueba con zonas de agua presente.

El juego de datos de entrenamiento y validación fue obtenido mediante un algoritmo, el cual toma de cada imagen posiciones aleatorias no repetidas y evalúa cada posición considerando el valor de una máscara creada a partir de los valores NDWI obtenidos. Luego, se rellena un arreglo compuesto por el valor del píxel en cada componente RGB de la imagen, y el valor binario del píxel tomado de la máscara hasta obtener un total de 20 muestras de cada clase (agua presente o sin agua) por imagen. Esta información fue utilizada para entrenar la red neuronal diseñada, de la cual se obtiene un valor de salida que actúa como clasificador de cada muestra que se presente como entrada.

Mediante la herramienta **nprtool** de Matlab, la red es configurada con 20 neuronas en las capas ocultas, destinando el 90% de la información de la matriz para entrenamiento y 10% para validación y comprobación de resultados. En la figura 4 se muestra el resultado de la aplicación de esta red en las imágenes de prueba, de la cual se logra obtener una máscara que delimita el cuerpo de agua con respecto al terreno. El resultado se expone aplicando la máscara obtenida a la imagen de prueba, en la cual queda identificada de color azul la región especificada como agua. En Colab, se estructura un modelo de red prealimentada de dos capas ocultas tal y como se define en Matlab. Se seleccionó como función de pérdida la configuración **binary_crossentropy** y **adam** como optimizador, ejecutando un entrenamiento de 37 épocas. Al finalizar, se hizo uso de la biblioteca Matplotlib para mostrar la máscara obtenida en una escala de colores, figura 5.



Figura 4 Resultado en Matlab.

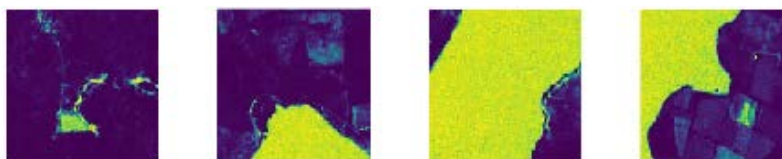


Figura 5 Resultado en Colab.

4. Discusión

La implementación o selección de modelos neuronales ha sido una de las arduas tareas en el desarrollo de la Inteligencia Artificial. La concepción adecuada de la arquitectura de la red en torno a la selección número de neuronas, capas ocultas y las conexiones neuronales debe lograr un equilibrio entre la precisión de la red y la generalización de la muestra para obtener resultados fiables. En tabla 2 se muestran los resultados del proceso de entrenamiento y validación de las redes implementadas en Matlab y Colab. Se presenta el valor porcentual de entropía cruzada obtenido (% CE), donde valores cercanos a cero corresponden a una buena clasificación.

Tabla 2 Resultados del proceso de entrenamiento y validación de la red.

Plataforma	Entrenamiento 252 muestras		Validación 28 muestras	
	% CE	% E	% CE	% E
Matlab	3.27	6.75	1.95	0
Colab	2.66	6.81	2.35	0.14

Es común usar un régimen de minimización de entropía cruzada en el entrenamiento de redes neuronales con el objetivo de generar mejores muestras en cada iteración del proceso. También se denota el porciento de error generado, en cada una de las etapas, que indica la fracción de muestras que fueron clasificadas incorrectamente. Al igual que el parámetro anterior, valores cercanos a cero indican una adecuada categorización de los resultados.

La figura 6 muestra la curva ROC (Receiver operating characteristic) correspondientes a la red neuronal generada en Matlab. Para el caso que ocupa, se logró una capacidad predictiva aceptable, por encima del 90% de precisión, que tributó a los resultados presentados cumpliendo el objetivo de identificación. En Colab, como se puede observar en la figura 7, se presenta la curva ROC obtenida de los procesos de entrenamiento y validación de la red implementada. Donde se logró también una capacidad predictiva fiable por encima del 90% de precisión. Si el conjunto de datos de entrenamiento y validación fuera mayor y variado, se alcanzarían mejores indicadores. Pero, a pesar del nivel de error, se lograron los

objetivos de discriminación pues se identificó con éxito cuerpos de agua en todas las imágenes de prueba. Los valores obtenidos demuestran una similitud en la capacidad predictiva de ambas redes ya que evidencian una diferencia de aproximadamente media unidad porcentual en promedio entre sus indicadores.

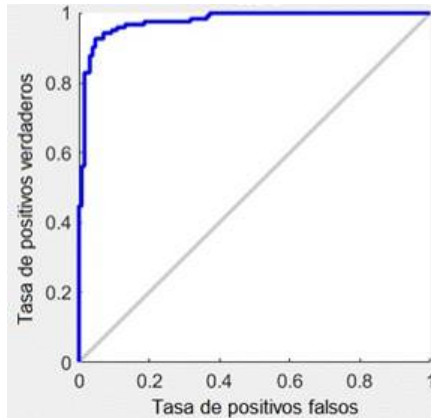


Figura 6 Curva ROC referente a la red neuronal generada en Matlab.

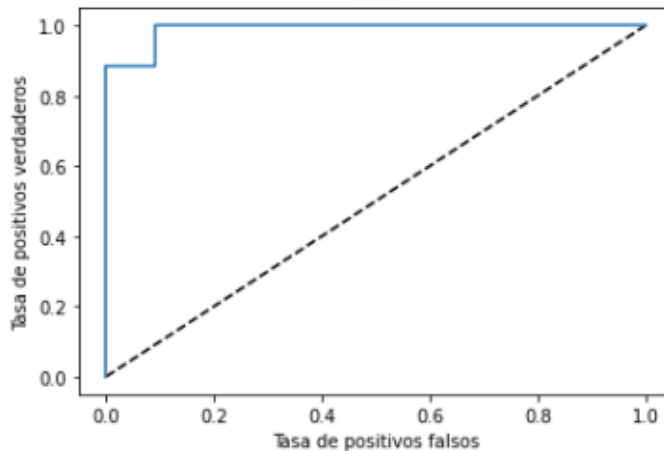


Figura 7 Curva ROC referente a la red neuronal generada en Colab.

En investigaciones presentadas sobre este tema se han logrado mayores niveles de precisión, de alrededor del 99%, usando enfoques tales como el uso de superpíxeles [Chen et al, 2018], redes neuronales convolucionales [Isikdogan et al, 2017] e incluso algoritmos SVM (Support Vector Machine) [Maragoudakis et al, 2018]. Estos métodos usados aseguran una mejor clasificación y capacidad predictiva.

La diferencia en el tiempo de ejecución del procesamiento de las imágenes de prueba en ambas plataformas fue notable, pero no es posible establecer una comparación adecuada en este aspecto. Mientras Matlab se ejecuta y es limitado sólo por las especificaciones de hardware local, Colab es una herramienta que se ejecuta en la nube y es limitado por el hardware que brinda la sesión virtual.

5. Conclusiones

La aplicación del procesamiento de imágenes en conjunto con redes neuronales, en las actuales actividades de monitoreo, reconocimiento e identificación de objetos de interés, ha impactado a grandes niveles en el análisis y toma de decisiones ante casos determinados. Además, se pueden encontrar variadas herramientas para el desarrollo de proyectos de este tipo. Matlab y Colaboratory son ambientes diferentes que logran obtener buenos resultados en el diseño de modelos de redes neuronales, apoyados en herramientas como nprtool y el API Keras de Tensorflow. Tal y como aparece representado en las figuras 4 y 5, se puede observar cómo el sistema logra la identificación del cuerpo de agua aun considerando que no se logra el mayor nivel de exactitud. El rendimiento del clasificador, presentado en las curvas ROC, es concluyente respecto a la aceptable capacidad predictiva de la red. Ampliando la cantidad y variedad de la muestra se puede lograr una mejor discriminación de masas acuíferas.

6. Bibliografía y Referencias

- [1] Kafy, Abdulla - Al. (2018). Importance of Surface Water Bodies for Sustainable Cities: A Case Study of Rajshahi City Corporation.
- [2] Zhang, Fangfang & Li, Junsheng & Shen, Qian & Ye, Huping & Wang, Shenglei & lu, Zoe. (2018). A simple automated dynamic threshold extraction method for the classification of large water bodies from landsat-8 OLI water index images. *International Journal of Remote Sensing*. 39. 3429-3451. <https://doi.org/10.1080/01431161.2018.1444292>.
- [3] Ko, Byoungchul & Kim, Hyeong & Nam, Jae. (2015). Classification of Potential Water Bodies Using Landsat 8 OLI and a Combination of Two Boosted

- Random Forest Classifiers. Sensors. 15. 13763-13777. <https://doi.org/10.3390/s150613763>.
- [4] Litjens, G. & Kooi, T. & Bejnordi, B.E. & Setio, A. & Ciompi, F. & Ghafoorian, M. & Laak, J.V. & Ginneken, B. & Sánchez, C. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42, 60-88.
- [5] Xie, Qiaoyun & Dash, Jadu & Huete, Alfredo & Jiang, Aihui & Yin, Gaofei & Ding, Yanling & Peng, Dailiang & Hall, Christopher & Brown, Luke & Shi, Yue & Ye, Huichun & Dong, Yingying & Huang, Wenjiang. (2019). Retrieval of crop biophysical parameters from Sentinel-2 remote sensing imagery. *International Journal of Applied Earth Observation and Geoinformation*. 80. 187-195. <https://doi.org/10.1016/j.jag.2019.04.019>.
- [6] Utton, Albert. (2019). *Water in a Developing World: The Management of a Critical Resource*. Routledge.
- [7] Chawla, Ila & Karthikeyan, L. & Mishra, Ashok. (2020). A Review of Remote Sensing Applications for Water Security: Quantity, Quality, and Extremes. *Journal of Hydrology*. 585. 124826. <https://doi.org/10.1016/j.jhydrol.2020.124826>.
- [8] Yuan, Qiangqiang & Shen, Huanfeng & Li, Tongwen & Li, Zhiwei & Li, Shuwen & Jiang, Yun & Xu, Hongzhang & Weiwei, Tan & Yang, Qianqian & Wang, Jiwen & Gao, Jianhao & Zhang, Liangpei. (2020). Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sensing of Environment*. 241.111716. <http://doi.org/10.1016/j.rse.2020.111716>
- [9] Cheng, Gong & Xie, Xingxing & Han, Junwei & Li, Kaiming & Xia, Gui-Song. (2020). Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. 13. 3735-3756. <https://doi.org/10.1109/JSTARS.2020.3005403>.
- [10] Nima Pahlevan & Sandeep K. Chittimalli & Sundarabalan V. Balasubramanian & Vincenzo Vellucci. (2019). Sentinel-2/Landsat-8 product consistency and implications for monitoring aquatic systems. *Remote Sensing of Environment*. 20. 19-29. <https://doi.org/10.1016/j.rse.2018.10.027>.

- [11] Hong, Danfeng & Gao, Lianru & Yokoya, Naoto & Yao, Jing & Chanussot, Jocelyn & Du, Qian & Zhang, Bing. (2020). More Diverse Means Better: Multimodal Deep Learning Meets Remote-Sensing Imagery Classification. *IEEE Transactions on Geoscience and Remote Sensing*. PP. 1-15. <https://doi.org/10.1109/TGRS.2020.3016820>.
- [12] Randles, Bernadette & Pasquetto, Irene & Golshan, Milena & Borgman, Christine. (2017). Using the Jupyter Notebook as a Tool for Open Science: An Empirical Study. 1-2. <https://doi.org/10.1109/JCDL.2017.7991618>.
- [13] Bisong E. (2019) Google Colaboratory. In: *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-4470-8_7
- [14] Pessoa, Tiago & Medeiros, Raul & Nepomuceno, Thiago & Bian, Gui-Bin & Albuquerque, V.H.C. & Filho, Pedro Pedrosa. (2018). Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications. *IEEE Access*. PP. 1-1. <https://doi.org/10.1109/ACCESS.2018.2874767>.
- [15] MATLAB. (2019). version 9.6.0.1062519 (R2019a). The MathWorks Inc. Natick, Massachusetts
- [16] Manaswi, Navin Kumar & John, Suresh. (2018). *Deep learning with applications using python*. Springer.
- [17] Maragoudakis, Manolis & Kontos, Konstantinos. (2018). Machine learning for water bodies identification from satellite images. *International Journal of Data Mining, Modelling and Management*. 10. 209. 10.1504/IJDM.2018.10015048.
- [18] Isikdogan, Furkan & Bovik, Alan & Passalacqua, Paola. (2017). Surface Water Mapping by Deep Learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. PP. 1-10. 10.1109/JSTARS.2017.2735443.
- [19] Chen, Yang & Fan, Rongshuang & Yang, Xiucheng & Wang, Jingxue & Latif, Aamir. (2018). Extraction of Urban Water Bodies from High-Resolution Remote-Sensing Imagery Using Deep Learning. *Water*. 10. 585. 10.3390/w10050585.