

SEGMENTACIÓN DE CENIZA VOLCÁNICA MEDIANTE DEEP LEARNING: VOLCÁN SANGAY, ECUADOR.

VOLCANIC ASH SEGMENTATION USING DEEP LEARNING: SANGAY VOLCANO, ECUADOR

Roberth J. Aldás, Katherine Tuz, Jair Vega, Marco Velasco, Christian Mejía

Universidad Central del Ecuador, Facultad de Ingeniería en Geología, Minas, Petróleos y Ambiental, Escuela de Geología
rjaldas@uce.edu.ec, kvtuz@uce.edu.ec, javegao@uce.edu.ec, msvelasco@uce.edu.ec, cimejia@uce.edu.ec

Resumen— La Inteligencia Artificial ha tenido un gran impacto en los últimos años, ya que cada vez este subcampo de la informática y la computación es más utilizado para la elaboración de diversos programas en distintas materias, que permiten una automatización de procesos y ayuden a cumplir con los objetivos planteados. En el campo de la geología, una de las tantas aplicaciones, corresponde a la detección y segmentación de ceniza en imágenes satelitales. Se diseñó un modelo de Deep Learning, basado en los métodos de segmentación y detección, utilizando una arquitectura de una red neuronal convolucional típica. Para la elaboración del modelo se recopiló imágenes satelitales de distintos satélites, con la condición de que en éstas se encuentre aplicado el filtro “ash”, que proporciona una coloración rosada rojiza a la ceniza, lo que ayuda para realizar el proceso de segmentación de las imágenes satelitales. El modelo tuvo una precisión final del 70%, que se considera como aceptable y un máximo del 99%. Para la utilidad que se le puede dar a nuestro modelo de Deep Learning, se seleccionó el caso de ceniza emitida por el Volcán Sangay, el cual en los últimos años ha tenido periodos de actividad, emitiendo grandes columnas de ceniza volcánica; pudiendo evidenciar mediante nuestro modelo, las zonas que presentan ceniza y las que no presentan ceniza.

Palabras Clave— Deep Learning, Segmentación, Red neuronal convolucional, Imágenes Satelitales, Ceniza, Volcán Sangay.

Abstract— Artificial Intelligence has had a great impact in recent years, since this subfield of computing is increasingly used for the development of various programs in different subjects, which allow process automation and help meet the objectives set in creating these programs. In the field of geology, one of the many applications corresponds to the detection and segmentation of ash in satellite images, choosing this application for our case study. In the project, a Deep Learning model was designed, based on segmentation and detection methods, using a typical convolutional neural network architecture. For the elaboration of the model,

satellite images of different satellites were compiled, with the condition that the “ash” filter is applied to them, which provides a reddish-pink coloration to the ash, which helps to carry out the segmentation process satellite images. The model had a final precision of 70%, which is considered acceptable and a maximum of 99%. For the utility that can be given to our Deep Learning model, the case of ash emitted by the Sangay Volcano was selected, which in recent years has had periods of activity, emitting large columns of volcanic ash; being able to show through our model, the areas that present ash and those that do not present ash.

Keywords— Deep Learning, Segmentation, Convolutional neuronal network, Satellite Images, Ash, Sangay Volcano.

I. INTRODUCCIÓN

La rama de la Inteligencia Artificial denominada Machine Learning proporciona algoritmos que posibilitan el aprendizaje automático por parte de las computadoras (Porcelli, 2020). Entre estos algoritmos, la técnica de Deep Learning ha tenido una atención especial debido a su utilidad en aplicaciones del mundo real, como detección y localización de objetos, segmentación, entre otras. Con base en el funcionamiento del cerebro humano, un sistema de Red Neuronal se compone por múltiples capas que realizan abstracciones de las características de los objetos de interés, útiles en nuevas situaciones (Silva, 2020).

Específicamente, la segmentación de imágenes es un problema que puede ser tratado adecuadamente mediante Deep Learning, de forma supervisada o no supervisada, y cumple con dividir la imagen en segmentos mediante la discriminación y clasificación de píxeles, designándole a cada uno de ellos una categoría, basándose en parámetros previamente asignados (Reyes, et al., 2018).

El propósito del presente trabajo es diseñar e implementar una solución basada en Deep Learning que permita segmentar emisiones de ceniza en imágenes

satelitales, mediante Redes Neuronales Convolucionales, además de elaborar un extenso y confiable dataset de imágenes, que sirva de entrenamiento para la predicción de la delimitación del esparcimiento de ceniza volcánica de nuevas imágenes satelitales. La relevancia del tema es debido al poder abrasivo que tiene el material volcánico, pudiendo ser perjudicial para los seres humanos, tanto para la ganadería, la agricultura, la salud y el transporte terrestre, marino y aéreo (OPS, 2021).

Se ha considerado como caso de estudio para la predicción, el actualmente activo Volcán Sangay, ubicado en el corredor Subandino del Ecuador (Figura 1) con coordenadas límites Noroeste (786003 E, 9784619S); Sureste (805184 E, 9769198 S) proyectadas en UTM, WGS84 17 S. La emisión paulatina de ceniza desde el 2019; ha afectado a zonas cercanas al sur y oeste de la zona, e incluso llegando a Guayaquil (IG-EPN, 2021). El Grupo de Investigación sobre la Ceniza Volcánica en Ecuador (GICVE) desde el 2017 monitorea la dispersión de este material que resulta peligroso para los seres vivos, pero no han presentado aún su modelo automático (GICVE, 2020).

Nuestras contribuciones principales son: 1) la implementación de un modelo basado en el Deep Learning para la segmentación de ceniza en imágenes satelitales y 2) la elaboración de un repositorio de imágenes satelitales de varios volcanes activos del mundo. Para tal fin hemos recolectado imágenes proporcionadas por los satélites GOES 16 y 17, Meteosat-8, Meteosat-11 y Himawari-8. Estas imágenes aplican un filtro Ash, en donde la ceniza tiene una coloración rosada rojiza y se encuentran publicadas en páginas web del Instituto Cooperativo de Estudios de Satélites Meteorológicos (CIMSS) (<https://cimss.ssec.wisc.edu/>) perteneciente al Centro de Ingeniería y Ciencias Espaciales (SEEC), del Instituto Cooperativo de Investigación en la Atmósfera (CIRA) (<https://www.cira.colostate.edu/>) y de la Administración Nacional Oceánica y Atmosférica (NOOA) (<https://www.noaa.gov/>).

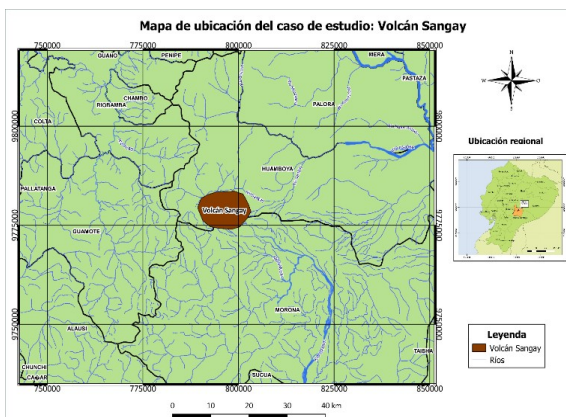


Figura 1. Mapa de Ubicación del Caso de Estudio.

II. METODOLOGÍA

En la Figura 2 se muestra un flujograma de la metodología realizada en este trabajo para la delimitación de ceniza volcánica mediante Deep Learning. Seguidamente, cada una de las etapas son descritas de manera detallada.

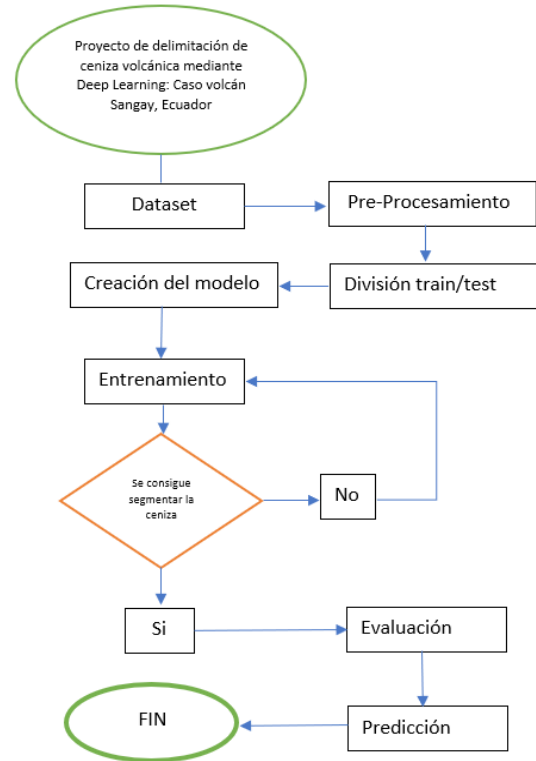


Figura 2. Metodología de trabajo

Dataset

Las imágenes fueron obtenidas de forma gratuita de la página web CIMSS (Space Science and Engineering Center of University of Wisconsin-Madison) Satellite Blog, descargando archivos .gif de emisiones de ceniza de volcanes activos que tengan el filtro Ash proporcionados por el satélite GOES (Figura 3); los cuales tienen como propiedades un ancho de 100 píxeles, alto de 821 píxeles y profundidad de 8 bits.

Los archivos tipo .gif se cargaron al programa en línea Online-convert, para posteriormente elegir la opción convertir GIF a JPG; en la que se puede subir el archivo .gif y mantener su calidad o modificarla; además de proporcionar opciones de coloración, monocromático, negativo, recortar píxeles, umbral de color, etc. Para este caso se utilizó la opción dada por defecto, usando calidad alta de 300x200 píxeles y ajustes de calidad bastante buena, obteniendo un total de 606 imágenes.

Para la denominación de las imágenes, se mantuvo el nombre por defecto de la descarga desde las páginas web del Instituto Cooperativo de Estudios de Satélites Meteorológicos (CIMSS), del Instituto Cooperativo de Investigación en la Atmósfera (CIRA) y de la Administración Nacional Oceánica y Atmosférica (NOOA), y el procesamiento del archivo .gif en Online-convert; denominándolas según el volcán, satélite y numeración (Novarupta_Goes17_091).

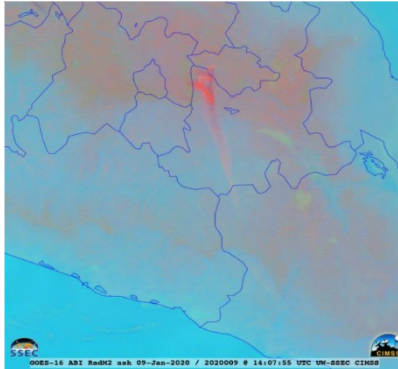


Figura 3. Imagen satelital GOES-16 aplicado el filtro Ash del volcán Popocatepetl.

Pre-Procesamiento

No es suficiente con la adquisición del Dataset, sino que también es necesario procesarlo; se deben de tener dos grupos de imágenes, el primer grupo correspondiente a las imágenes satelitales adquiridas y el segundo con las imágenes segmentadas, mostrando de color negro la zona o el área de la imagen satelital donde se evidencia ceniza, y en color blanco el resto de área de la imagen donde no exista ceniza, como se muestra en la Figura 4.

El proceso de segmentación se lo realizó en el programa Photoshop CS6, el cual cuenta con herramientas para la modificación de parámetros de las imágenes, como la saturación, tono, brillo, intensidad, contraste, etc., para finalmente, transformar nuestra imagen de color, a una blanco y negro mediante la especificación del valor de un umbral, provocando así que el área de la imagen donde exista ceniza tome el color negro y el resto de la imagen sea blanca; además, se modificó el tamaño de las imágenes a 300 por 200 píxeles. Cada una de estas acciones se las realizó en lotes de imágenes.

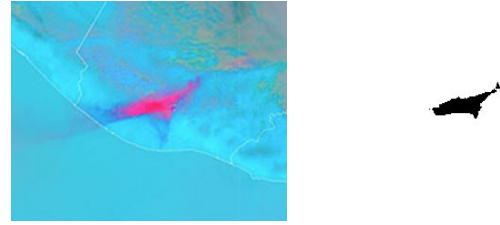


Figura 4. La imagen de la izquierda corresponde al Data Set original, en donde la ceniza se encuentra de color rojo y rosado. La imagen de la derecha corresponde a la imagen segmentada, en donde únicamente se marca la ceniza con tonalidad negra.

Una vez modificadas las imágenes, se crearon dos carpetas, la primera llamada **Ceniza_images** que contiene las imágenes redimensionadas en formato .jpg con sistema de color RGB y la segunda carpeta nombrada **Ceniza_masks**, que contiene las imágenes, con el mismo nombre, en formato .jpg de la carpeta **Ceniza_images**, pero con sistema de color B&N.

Con la importación de las librerías NumPy y cv2 de Python, se permitió la incorporación de arrays y la manipulación de matrices de datos numéricos, normalizando las imágenes para llevar todos los valores del array a un rango definido (SciPy, 2021), y posteriormente, redimensionar las imágenes a 32 píxeles x 32 píxeles, brindando mayor facilidad y rapidez durante el proceso del entrenamiento.

Al ser un caso supervisado, para poder alimentar y entrenar la red se definen entradas (inputs) con salidas (outputs) conocidas. A través de la función `np.empty`, se crea y devuelve una referencia a un array vacío (Sánchez, 2020) con las dimensiones 32, 32, 3; incluyendo además el argumento `dtype='float32'`, el cual indica un bitsize de 32.

A través de la librería matplotlib, se pueden visualizar las imágenes normalizadas y redimensionadas.

División del Dataset

Para la división del dataset se utilizó el comando: `train_split()`, especificando dos subconjuntos; el primero de entrenamiento (train) que contiene el 80% de los datos y el segundo de prueba (test) con el 20% de los datos restantes; la distribución de los datos se puede ver en la Tabla 1. Los datos presentan un input total de (606, 32, 32, 3) y para el `inputs_train.shape` una salida de (485, 32, 32, 3).

El 80% de la base de datos son el conjunto de entrenamiento, con los que se ajusta los parámetros del modelo y el 20% de la base de datos es el conjunto de test, con los que se evalúa el rendimiento del modelo entrenado. (Aguilar, et al., 2019)

Tabla 1. Número de imágenes y división del Dataset.

Variable	Número de imágenes	Porcentaje
Tamaño del conjunto de datos	606	100 %
Datos para entrenamiento o	485	80 %
Datos para prueba	121	20 %

Creación del Modelo

El modelo es una adaptación del presentado por Ronneberger et al. (2015), el cual se diseñó para tareas de segmentación con fines biomédicos. Esta arquitectura se basa en una típica red neuronal convolucional (CNN) compuesta por un codificador (encoder) y un decodificador (decoder) (Livne, et al., 2019), configurando un patrón de contracción (encoder) para reducir la resolución de la imagen y un patrón expansivo simétrico (decoder) para el aumento de su resolución (Li, et al., 2019).

La imagen de entrada (input) tiene una dimensión de $32 \times 32 \times 3$. El encoder (parte izquierda de la Figura 5) está

establecido de acuerdo con el número de filtros (8, 16, 32 y 64) aplicados por capa. A cada capa convolucional le compuesto por 5 capas convolucionales, con filtros de tamaño de 3×3 que crean mapas de características capturando patrones en la imagen, en la que el volumen de salida está

sigue una unidad linear rectificada (ReLU) que anula los valores negativos y deja pasar a los positivos a la siguiente capa tal como entran (Yasrab, 2018). La operación de MaxPooling de 2×2 realiza un muestreo descendente (downsampling), en donde, en cada contracción las dimensiones son reducidas a la mitad.

El decodificador tiene como objetivo recuperar las dimensiones originales de las imágenes de entrada mediante un muestreo ascendente (upsampling) del mapa de características (Livne, et al., 2019). Así en el decoder (parte derecha de la Figura 5) se incluyen 5 capas convolucionales transpuestas de 3×3 , que reducen a la mitad el número de canales de características, continuando con la función de activación ReLU. La capa final es una convolución de 3×3 , con una activación de tipo sigmoide usada para mapear cada vector de características (Ronneberger, et al., 2015). El resultado (output) es una imagen de dimensiones de $32 \times 32 \times 3$.

Se debe considerar que la implementación de las capas del modelo fue importada mediante la instrucción `from tensorflow.keras.models import Model`.

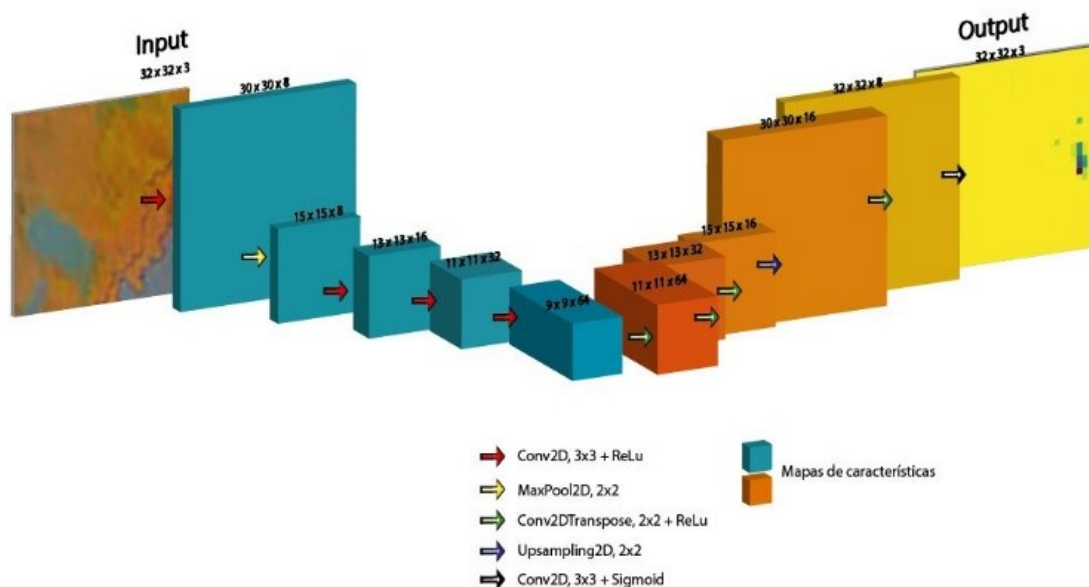


Figura 5. Ilustración de la arquitectura del modelo. A la izquierda el Encoder y a la derecha el Decoder. Cada caja corresponde a un mapa de características multicanal.

III. RESULTADOS

Entrenamiento y Evaluación del Modelo

El entrenamiento se realiza en Google Colaboratory, una plataforma on-line de hardware y software para aprendizaje automático de acceso gratuito que brinda 12 GB de RAM y 50 GB de almacenamiento en disco. Los códigos y comandos que se ejecuten mediante los notebooks de Colab usan GPU y TPU (graphics processing unit y tensor processing unit, respectivamente), que es un circuito integrado de aplicación específica y acelerador de Inteligencia Artificial, que no afecta el rendimiento de la computadora. (De la Fuente, 2019)

Para realizar el entrenamiento se utiliza el comando `model.fit`, estableciendo las iteraciones del conjunto de datos o épocas que se realizan para tener una mejor precisión del modelo, en este caso se establecieron 100 épocas, teniendo un tiempo de ejecución de 15 minutos aproximadamente.

Se estableció un `verbose` de 1, lo que nos permite observar una barra de proceso animada y ayuda a detectar el sobreajuste que ocurre si su precisión de entrenamiento ('`acc`') sigue mejorando mientras su precisión de validación ('`val_acc`') empeora.

Se obtuvo una precisión final de entrenamiento y evaluación de 0.70 y una pérdida final de entrenamiento y evaluación de 0,01. La mejor precisión del entrenamiento se obtuvo en la época 10 con un valor de 0.99 y un valor de pérdida de 0.16. Para visualizar el comportamiento del entrenamiento y validación del modelo se realizó el plot de los valores de pérdida y precisión del entrenamiento y de la validación (Figura 6 y 7) a través de las instrucciones de `plt.figure` y `plt.show`.

Estas gráficas ayudan a entender el rendimiento del modelo, en el caso de la pérdida se observa que tiende a cero, mientras en la precisión se observa una fluctuación entre 0.7 y 1.0. Además, las gráficas demuestran que no existe overfitting en el entrenamiento y validación del modelo.

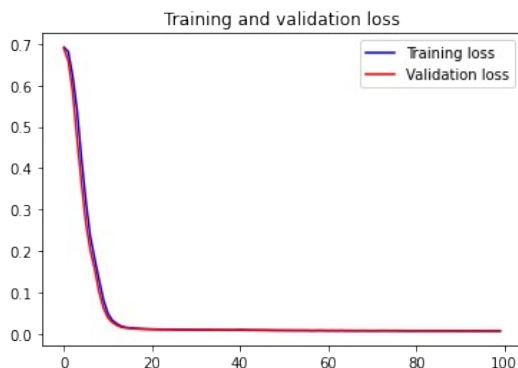


Figura 6. Pérdida del entrenamiento y validación del modelo.

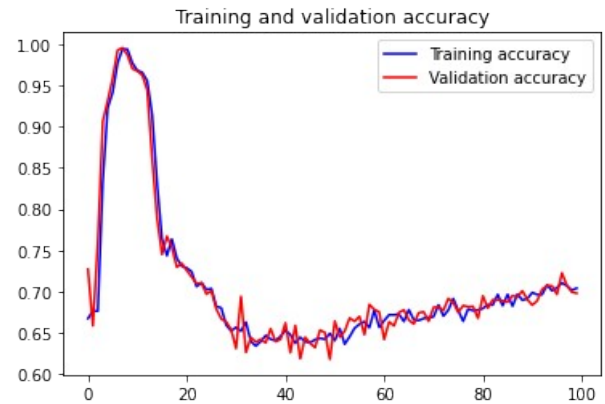


Figura 7. Precisión del entrenamiento y validación del modelo

Para un mejor detalle de los resultados, se implementa la matriz de confusión, que es una tabla resumida utilizada para poder evaluar el rendimiento del modelo de clasificación. El número de predicciones correctas e incorrectas se resumen en la tabla con un conteo y se muestra gráficamente para cada clase. (Shin , 2020)

En la Figura 8 se observa la matriz de confusión de lo que se destaca que el modelo tiene una precisión de 88%, que se obtiene de la suma del cuadro amarillo y verde dividido contra todos los cuadros, es decir, se utiliza los valores de verdadero positivo y verdadero negativo. La fracción de verdaderos positivos (sensibilidad) del modelo es del 85.7% capaz de detectar correctamente la ceniza, esto obtenido mediante el verdadero positivo (cuadro amarillo) y falso negativo (cuadro morado).

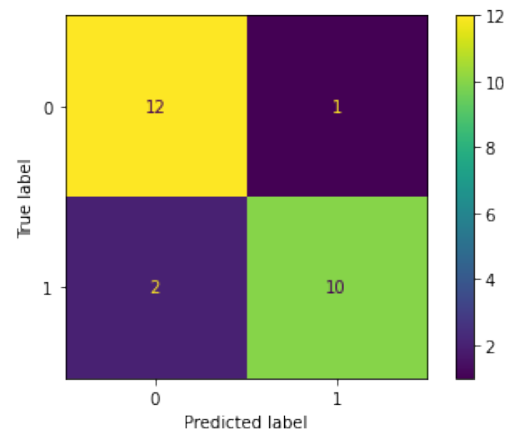


Figura 8. Matriz de confusión del modelo.

Segmentación de Imágenes de Validación

Para la validación del modelo se realizó con el 20% de la base de datos, que corresponde a 121 imágenes cargadas en el `inputs_test`. En la Figura 9, en la imagen de la derecha, se observa la imagen predicha para ceniza y concuerda con la zona establecida de ceniza en el output. Esto tiene relación con la pérdida de validación (`val_loss`) que tiene un valor de 0,16 y con el valor de precisión de validación (`val_acc`) de 0,98, lo que indica un comportamiento adecuado del modelo.

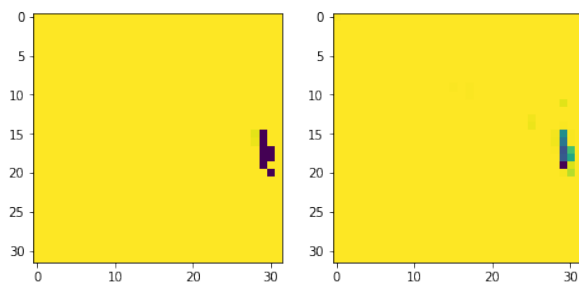


Figura 9. Izquierda: imagen segmentada del output. Derecha: imagen de validación segmentada. Los píxeles de color morado y verde indican la existencia de ceniza, los píxeles en amarillo no presentan ceniza

IV. DISCUSIÓN

Las gráficas de precisión y pérdida (Figura 6 y 7) evidencian que se alcanzó la mejor precisión en la época 10 con un valor de 0.99 y con una pérdida de 0.16. Por su parte, la matriz de confusión (Figura 8) indica que la precisión del modelo es del 88% y su sensibilidad 85,7%, mostrando que los resultados son los esperados y el modelo tiene un buen entrenamiento y validación.

Por ende, se estima que el modelo tendrá un buen reconocimiento en una nueva imagen ya que el aprendizaje fue óptimo con respecto al dataset inicial y tampoco tendrá un desajuste debido a que se ha trabajado con una base de datos extensa, la cual se puede aumentar para seguir mejorando la precisión final, pero teniendo en cuenta las especificaciones que se pide para las imágenes satelitales. El modelo realizado puede ser implementado para la segmentación de ceniza volcánica ya que se considera como un modelo válido que aún se puede mejorar.

Predicción

Para la predicción del modelo mediante el comando `model.predict`, se utilizaron imágenes satelitales con filtro Ash del Volcán Sangay, obtenidas del satélite GOES-16, estas imágenes no han sido utilizadas en el entrenamiento o en la validación. En la Figura 10 se observa la imagen de entrada y la imagen segmentada, la cual

muestra similitud en la segmentación de ceniza que se encuentra en la parte superior central de la imagen. Sin embargo, la delimitación no es exacta, ya que no se segmenta todo el contorno de ceniza, pero es considerada aceptable debido a su precisión.

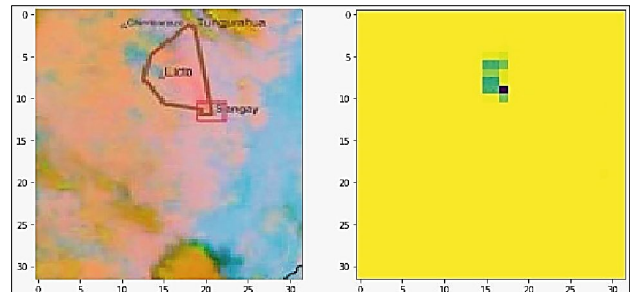


Figura 10. Imagen izquierda es una imagen satelital del Volcán Sangay obtenida del satélite GOES-16 el 9 de marzo de 2021 (IG-EPN, 2021), para la predicción del modelo. Imagen derecha es una imagen segmentada que delimita el contorno de la ceniza del Volcán Sangay.

V. CONCLUSIONES

Hemos implementado un modelo para la segmentación de ceniza de imágenes satelitales, aplicando Deep Learning, a través la plataforma Google Colaboratory en lenguaje Python; obteniendo así, para el caso de estudio, la predicción de la segmentación de ceniza en imágenes satelitales de emanaciones del Volcán Sangay.

La elaboración del dataset, con la cual se entrenó al modelo, tiene un total de 606 imágenes en formato .jpg. El Dataset, obtenido de las páginas web del Instituto Cooperativo de Estudios de Satélites Meteorológicos (CIMSS) perteneciente al Centro de Ingeniería y Ciencias Espaciales (SEEC), del Instituto Cooperativo de Investigación en la Atmósfera (CIRA) y de la Administración Nacional Oceánica y Atmosférica (NOOA), resultando en una base de información confiable y extensa.

El modelo diseñado e implementado fue basado en una red neuronal convolucional típica, compuesta por capas convolucionales que configuran un codificador y un decodificador; de esta manera se produjo un rendimiento aceptable para la predicción en la delimitación de ceniza volcánica en imágenes satelitales, teniendo que el valor de pérdida es igual a 0.01, el valor de precisión más alto que se obtuvo es de 0.99 y la matriz de confusión muestra una evaluación positiva.

Con el modelo se pudo realizar la segmentación de ceniza proveniente del Volcán Sangay en imágenes satelitales con filtro Ash, como se muestra en los resultados, en donde se segmenta la ceniza del volcán, capturada en una imagen satelital del GOES-16 obtenida el 9 de marzo del 2021.

El código del modelo para la segmentación de ceniza volcánica en imágenes satelitales es una alternativa a los métodos tradicionales, encontrándose disponible en el siguiente enlace:

https://colab.research.google.com/drive/12ZNt6lSKcnUbWRIboylozM4Brey_vO1S?usp=sharing

El Dataset se encuentra libre en el siguiente enlace:

<https://drive.google.com/drive/folders/1jFAg5MiF9zZK1ESlhGP7usHTs4zeprGU?usp=sharing>

VI. RECOMENDACIONES

Realizar un preprocesamiento adecuado y riguroso para obtener imágenes que puedan ser procesadas con el modelo, utilizando programas de edición, en donde sea posible manipular la saturación, tono, brillo, intensidad, contraste y umbral de las éstas.

El presente modelo está entrenado con imágenes satelitales con el filtro Ash, por lo que, para futuras predicciones, se deberán considerar imágenes con un filtro o tipo de variación contrastante para obtener un resultado óptimo.

Se debe tener un conocimiento básico de programación para manipular el código, sugiriendo continuar con el tratamiento de imágenes relacionadas a ciencias geológicas o ramas afines, para permitir una mayor facilidad en su análisis.

VII. BIBLIOGRAFÍA

- Aguilar, R., Torres, J. & Martín, C., 2019. Aprendizaje Automático en la Identificación de Sistemas. Un Caso de Estudio en la Predicción de la Generación Eléctrica de un Parque Eólico. s.l.: Revista Iberoamericana de Automática e Informática Industrial 16 (2019) 114-127.
- De la Fuente, O., 2019. Google Colab: Python y Machine Learning en la nube. [En línea] Available at: <https://www.adictosaltrabajo.com/2019/06/04/google-colab-python-y-machine-learning-en-la-nube/> [Último acceso: 15 Marzo 2021].
- GICVE, 2020. Ceniza Volcánica en Ecuador. [En línea] Available at: <https://www.usfq.edu.ec/es/investigacion/grupo-de-investigacion-sobre-la-ceniza-volcanica-en-el-ecuador-gicve>
- IG-EPN, 2021. IGAI Instante Informativo VOLCÁN SANGAY N° 2021-034. Quito: Escuela Politécnica Nacional.
- IG-EPN, 2021. IG-EPN. [En línea] Available at: <https://www.igepn.edu.ec/sangay> [Último acceso: 20 Marzo 2021].
- Li, S., Yang, C., Sun, H. & Zhang, H., 2019. Seismic fault detection using an encoder–decoder convolutional neural network with a small training set. Journal of Geophysics and Engineering, pp. 1-15.
- Livne, M. y otros, 2019. A U-Net Deep Learning Framework for High Performance Vessel Segmentation in Patients With Cerebrovascular Disease. Frontiers in Neuroscience.
- OPS, 2021. Los impactos a la salud asociados con las cenizas de los volcanes. [En línea] Available at: https://www.paho.org/hq/index.php?option=com_content&view=article&id=8194:2013-los-impactos-salud-asociados-cenizas-volcanes&Itemid=39797
- Porcelli, A., 2020. Artificial Intelligence and Robotics: Its social, ethical and legal dilemmas. Scielo.
- Reyes, O., Mejía, M. & Useche, S., 2018. Técnicas De Inteligencia Artificial Utilizadas En El Procesamiento De Imágenes Y Su Aplicación En El Análisis De Pavimentos. Revista EIA- Colombia., pp. 1-5.

- Ronneberger, O., Fischer, P. & Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. Springer International Publishing Switzerland, pp. 234-241.
- Sánchez, A., 2020. Aprende con Alf. [En línea] Available at: <https://aprendeconalf.es/docencia/python/manual/numpy/> [Último acceso: 22 Marzo 2021].
- SciPy, 2021. Numpy. [En línea] Available at: <https://numpy.org/doc/stable/user/index.html> [Último acceso: 22 Marzo 2021].
- Shin, T., 2020. Comprensión de la Matriz de Confusión y Cómo Implementarla en Python. [En línea] [Último acceso: 21 Marzo 2021].
- Silva, A., 2020. Estudio comparativo de modelos de clasificación automática de señales de tráfico. Universidad Pública de Navarra-España., pp. 7-14.
- Yasrab, R., 2018. ECRU: An Encoder-Decoder Based Convolution Neural Network (CNN) for Road-Scene Understanding. Journal of Imaging.