Clasificación Automática de Fenómenos de Remoción en Masa mediante Deep Learning, Caso de Estudio: Vía Guasuntos-Achupallas.

Automatic Classification of Mass Removal Phenomenon through Deep Learning, Case Study: Via Guasuntos-Achupallas.

Universidad Central del Ecuador

Facultad de Ingeniería en Geología, Minas, Petróleos y Ambiental, Carrera de Geología

[1]embeltran@uce.edu.ec

[2]fafuentes@uce.edu.ec

[3]djnoguera@uce.edu.ec

Resumen

El desarrollo tecnológico permite acceder a un sinnúmero de herramientas computacionales que facilitan el análisis e interpretación de cualquier variable, por lo que, el presente estudio se basa en el análisis de Fenómenos de Remoción en Masa (FRM), en la vía que conecta los poblados de Guasuntos y Achupallas, mediante el uso de imágenes satelitales y algoritmos de aprendizaje profundo (Deep Learning), específicamente la clasificación multiclase que permite generar y entrenar modelos de Redes neuronales que extraen características específicas observadas en cada imagen satelital analizada, permitiendo su identificación y posterior clasificación. Después de un proceso de prueba y error en la creación del modelo, se optó por trabajar con una red neuronal, 64 neuronas, 32 filtros y 150 épocas, obteniendo una precisión de entrenamiento del 97% y en validación un máximo de 76, mientras que en el entrenamiento tuvo un valor de 0.23 y de validación un mínimo de 0.68. En la matriz de confusión se obtuvo valores máximos de precisión y reconocimiento para casa tipo de FRM, se tiene para Flujos un valor de 0.81 (Clase 0), Reptación con 0.76 (Clase 1), Rotacionales con 0.79 (Clase 2) y Traslacionales con 60% (Clase 3). Considerando que el valor para aceptar un modelo debe ser superior a 0.8 se tiene que los FRM tipo Flujo y Rotacional presentan mayor precisión en su predicción.

Palabras clave: *Fenómenos de Remoción en Masa (FRM), clasificación, imágenes satelitales, Red neuronal.*

Abstract

Technological development allows access to countless computational tools that facilitate the analysis and interpretation of any variable, so this study is based on the analysis of Mass Removal Phenomena (MRF) in the road connecting the towns of Guasuntos and Achupallas, through the use of satellite images and deep learning algorithms (Deep learning), specifically multiclass classification that allows the generation and training of neural network models that extract specific characteristics observed in each satellite image analyzed, allowing their identification and subsequent classification. After a process of trial and error in the creation of the model, it was decided to work with a neural network, 64 neurons, 32 filters and 150 epochs, obtaining a training accuracy of 97% and in validation a maximum of 76, while in training it had a value of 0.23 and in validation a minimum of 0.68. In the confusion matrix, maximum values of precision and recognition were obtained for each type of FRM, with a value of 0.81 (Class 0) for Flows, 0.76 (Class 1) for Reptation, 0.79 (Class 2) for Rotational and 60% (Class 3) for Translational. Considering that the value to accept a model must be higher than 0.8, it is found that the Flow and Rotational WRFs present greater accuracy in their prediction.

Keywords: *Mass Removal Phenomena (FRM), classification, satellite images, neural network.*

1 Introducción

Los Fenómenos de Remoción en Masa (FRM) constituyen un peligro latente para los habitantes de las zonas de influencia; por esta razón se han desarrollado varios estudios, con el objetivo de mitigar el impacto generado (SGR, 2014). En Ecuador, es de especial interés los tipos de FRM que se suscitan a lo largo de la vía Guasuntos-Achupallas, cantón Alausí, provincia de Chimborazo (Figura 1). Según Valdiviezo (2014), son deslizamientos de tipo Traslacionales, Rotacionales, Reptacional y Flujos, siendo deslizamientos de tipo flujo los más recurrentes.

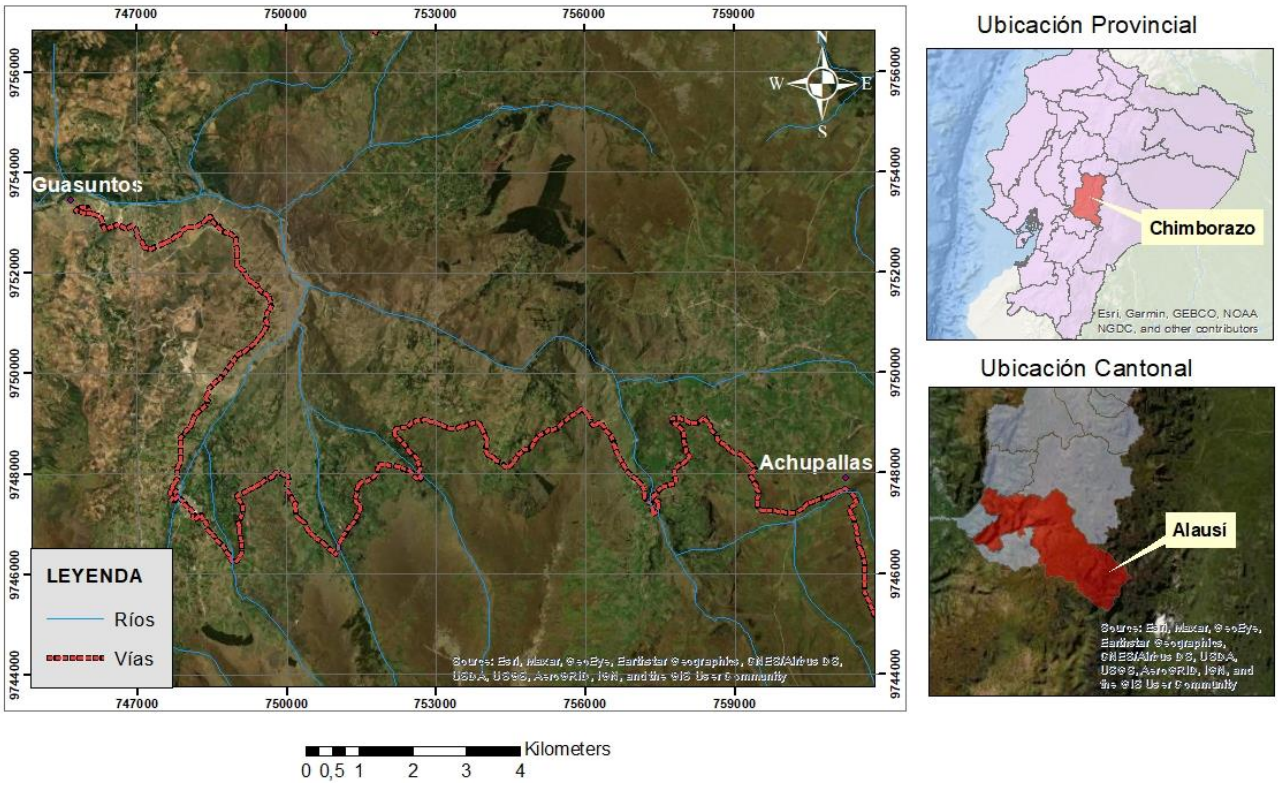

Fig. 1 Mapa de ubicación de la zona de estudio.

El conocimiento geológico es la base para identificar y catalogar distintos tipos de FRM. Si lo combinamos con el aprendizaje automatizado, es posible desarrollar modelos de clasificación que reduzcan el tiempo y esfuerzo en el análisis, a diferencia de un método tradicional (Flores, 2019).

El aprendizaje automático (Machine Learning) es una alternativa computacional para el tratamiento de problemas de mayor avance tecnológico en los últimos años. Según Molinero (2019), el uso de algoritmos de aprendizaje profundo (Deep Learning, una de las ramas del Machine Learning) y el fácil acceso a imágenes satelitales, permite definir la tipología de un deslizamiento en base a sus características morfológicas. De esta manera, las zonas afectadas contarán con información complementaria para implementar obras de mitigación y campañas de capacitación para los pobladores, reduciendo el impacto económico y social, en caso de suscitarse un nuevo evento (Naranjo, 2012). Además, considerando la exactitud en predicción de deslizamientos del modelo planteado, podría en un futuro aplicarse a nivel mundial como un complemento a estudios de campo o fotogeología. Las contribuciones concretas del presente trabajo son: a) el código fuente del modelo de reconocimiento y clasificación automática que se aplicará para la predicción del tipo de FRM en la vía Guasuntos-Achupallas; y b) Un Dataset de imágenes satelitales con presencia de FRM a nivel mundial que sirve como conjunto de entrenamiento para dicho modelo. Ambos productos están disponibles públicamente (descargable) en las Fanpage del Capitulo Estudiantil AAPG-UCE (Facebook: @Aapg-Uce.figempa, Instagram: @aapg_uce_figempa).

2 Metodología

La metodología utilizada para la predicción de FRM, mediante el análisis de imágenes satelitales con un modelo de clasificación de Deep learning, se describe a continuación:

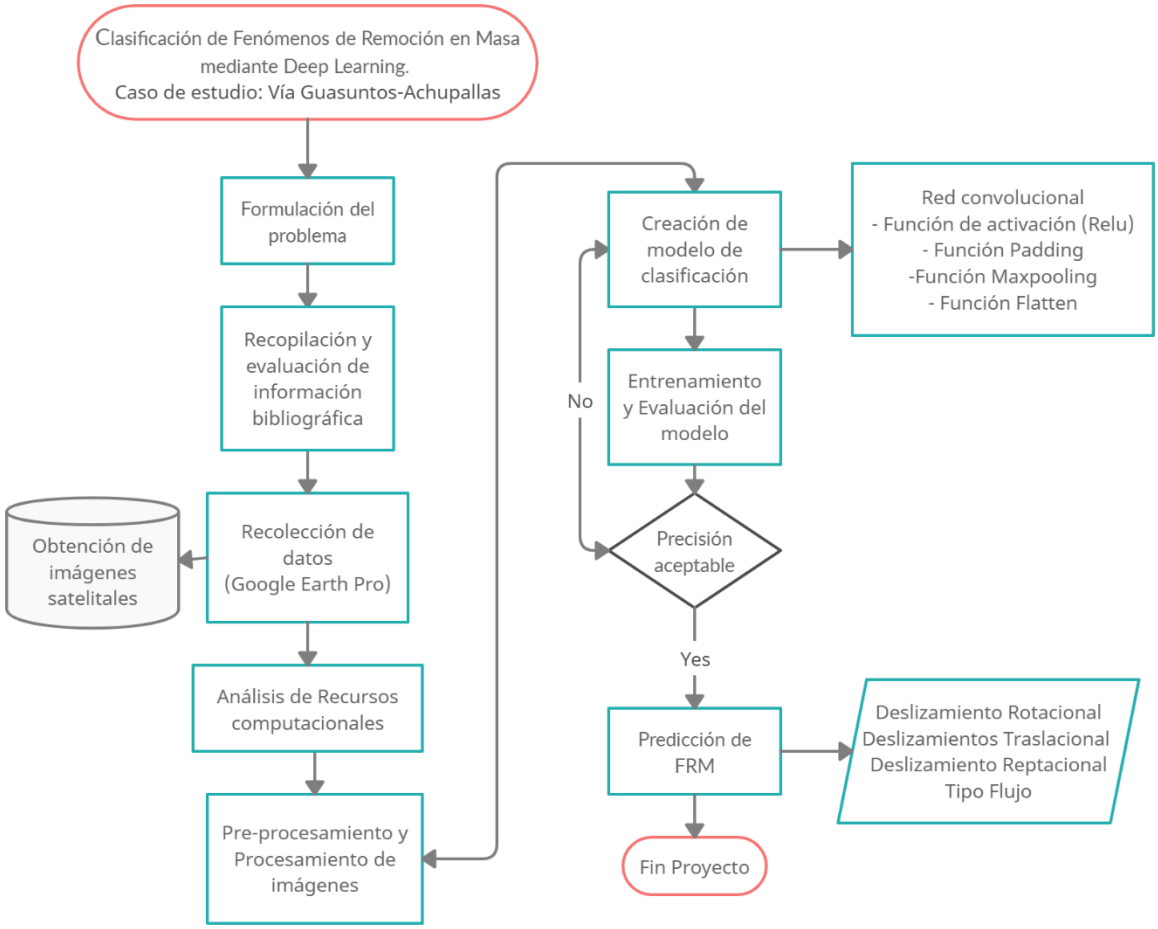

Fig. 2 Diagrama de flujo de la metodología utilizada en clasificación para FRM.

2.1 Formulación del problema

Fenómeno de Remoción en masa o FRM, es un proceso donde cierto volumen de material (suelo, roca, escombros, etc.) se desplaza ladera abajo por varios factores como fuertes precipitaciones, al pendiente del terreno, presencia de fallas, material débil o poco compactado (Eras, 2014). Varios estudios realizados a lo largo de la vía Guasuntos-Achupallas revelan la existencia de un sinnúmero de deslizamientos que han generado pérdidas materiales y humanas, además el cierre de esta vía impide el comercio-transporte de productos agrícolas. Las principales obras que se han realizado en la zona son cortes de laderas y limpieza de carreteras, pero los deslizamientos continúan, es así como un estudio

exhaustivo de la zona considerando medios tecnológicos y tradicionales, puede ayudar a desarrollar planes de mitigación, mejorando el estilo de vida de las poblaciones afectadas.

2.2 Recopilación y evaluación de información bibliográfica

(Flores, 2019) "Técnicas para la predicción espacial de zonas susceptibles a deslizamientos". Esta investigación realiza la predicción de zonas susceptibles a deslizamiento por medio de aprendizaje automático, utilizando como datos de entrada factores condicionantes de un FRM, por medio de clasificación. El presente estudio se diferencia por utilizar clasificación multicapa, e imágenes satelitales provenientes de Google Earth.

(González Vilas, 2013) "Teledetección y técnicas de aprendizaje automático supervisado aplicados a la discriminación de vertidos de hidrocarburos". Esta investigación detecta vertidos de hidrocarburos a partir de imágenes ASAR, por medio de clasificación binaria. Las principales diferencias con el presente estudio es la fuente de los datos de entrada como la aplicación de un algoritmo de clasificación multicapa.

2.3 Recolección de datos

La fuente única de obtención de imágenes satelitales fue Google Earth Pro con un tamaño de 200 x 300, después de una exhaustiva búsqueda alrededor de todo el planeta, se logró recopilar un total de 250 imágenes para cada tipo de deslizamiento. Cada una de las imágenes fue procesada en Adobe Photoshop, para disminuir las dimensiones de estas, ya que en un inicio su tamaño era de 1920 x 1080 (Figura 3).

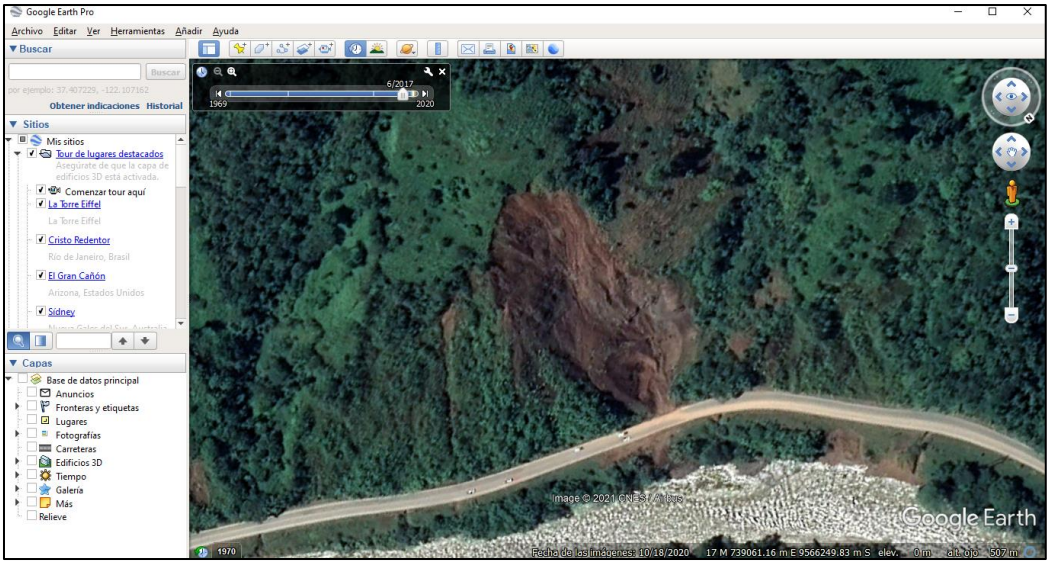

Fig. 3 Búsqueda de Imágenes en Google Earth.

El grupo de 1000 de imágenes obtenidas en la búsqueda fueron direccionadas al disco local D, dentro de la carpeta DATABASE (Figura 4), en la cual se encuentran las 4 categorías a clasificar: Flujos, Reptación, Rotación y Traslación.

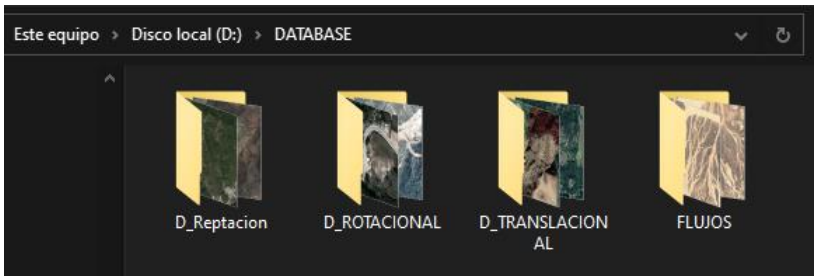

Fig. 4 Distribución de carpetas en el Disco local D.

2.4 Recursos computacionales

2.4.1. Hardware

Se necesita un computador potente con el cual se pueda realizar varias pruebas hasta tener un resultado aceptable, las características del computador, donde se puede destacar el Procesador (Intel Core i7, 7ma generación) y la tarjeta gráfica integrada (Gráficos AMD Radeon de 2 GB dedicados) las cuales son de gran ayuda en el momento de reducción del tiempo de ejecución de las diferentes pruebas.

2.4.2. Software.

Se utilizó el sistema operativo Windows 10 de 64 bits y se trabajó con el lenguaje de programación Python en aplicación Jupyter Notebook, el cual es un entorno de trabajo interactivo que permite desarrollar el código de manera dinámica, a la vez que integrar en un mismo documento tanto bloques de código como texto, gráficas o imágenes. Jupyter se puede utilizar a través de suites, la más conocida es Anaconda, dado su avanzado crecimiento en los últimos años, sus actualizaciones se han vuelto demasiado pesadas para computadoras que no tienen las características necesarias, esta suite va a ser un problema, es por esta razón que se utilizó una distribución ligera de la Suite Anaconda, llamada Miniconda, esta distribución contiene lo mínimo necesario, una vez instalado Jupyter notebook se empieza a crear el código. Se usarán varias librerías, estas son gratuitas y contienen un sin número de funciones que nos van a facilitar la creación del código, las librerías que utilizamos se resumen en la (Tabla 1):

Tabla 1. Descripción de librerías utilizadas en el proyecto

Nombre	Propósito	Funciones
Matplotlib	Visualización; Librería encargada de la generación de gráficos.	Pyplot, subplot, show, imshow, title
NumPy	Estructuras de datos, vectores y matrices; proporciona funciones matemáticas de alto nivel que operan en estas estructuras de datos.	Dtype, list, nd.array, np.array, np.unique
Scikitlearn	Es una librería de Python para Machine Learning y Análisis de Datos.	train_test_split, rezise classification_report
Tensor Flow	Es una librería de Python, desarrollada por Google, para Deep learning y otras aplicaciones de cálculo científico.	to_categorical, Sequential, Input, Model, Dense, Dropout, Flatten, Conv2D, Relu, MaxPooling2D, BatchNormalization,

Keras	Es un interfaz de alto nivel para trabajar con redes neuronales. Keras utiliza otras librerías de Deep learning (TensorFlow, CNTK o Theano) de forma transparente.	Model, Dense, Dropout, Flatten, Conv2D, Relu, MaxPooling2D

2.5 Pre – procesamiento y procesamiento de imágenes

2.5.1. División del Dataset

Mediante la instrucción '*train_test_split()',* se dividió el Dataset en dos subconjuntos, training data (70%), aquellos datos o imágenes utilizadas para que el modelo identifique características y de esta manera aprenda a reconocerlas , la calidad del modelo de aprendizaje automático será directamente proporcional a la calidad de los datos, y testing data (30%), aquellos datos que son reservados para comprobar si el modelo que se ha generado a partir de los datos de entrenamiento es adecuado, es decir, si las respuestas predichas por el modelo para un caso totalmente nuevo son acertadas o no. Se utilizó esta relación 70/30 de los datos, debido a que presentó los mejores resultados de precisión. El número de imágenes utilizadas en cada subconjunto se detallan en la (Tabla 2).

Tabla 2. División del Dataset (Entrenamiento y prueba)

	Porcentaje	Numero de imágenes
Training data	70%	700
Testing data	30%	300
Total	100%	1000

2.5.2. Normalización de imágenes

Cada imagen posee 3 canales de colores: RGB (Red, Green, Blue) con valores de 0 a 255, el proceso de normalización se obtiene dividiendo cada pixel para 255 obteniendo valores entre 0 y 1. Posteriormente en esta sección se vuelve a dividir el conjunto de datos,

utilizando ‘*train_test_split()’,* en una relación 90/10, cuyo objetivo es validar las imágenes en pleno proceso de entrenamiento, la división y número de imágenes utilizadas se detalla en la (Tabla 3).

Tabla 3. División del Dataset (Entrenamiento y validación)

	Porcentaje	Numero de imágenes
Entrenamiento	90%	630
Validación	10%	70
Total	100%	700

2.6 Creación de Modelo

Se diseñó un modelo Deep Learning a partir de uno preexistente, realizado para clasificación de imágenes deportivas (Bagnato, J. 2018) y que ha sido adaptado a nuestro proyecto con características propias en base a los objetivos planteados del proyecto. El modelo final se puede resumir en dos fases: Base de convolución y fase del Clasificador. Los parámetros del modelo fueron obtenidos tras un proceso de prueba y error de varios ensayos (Figura 5).

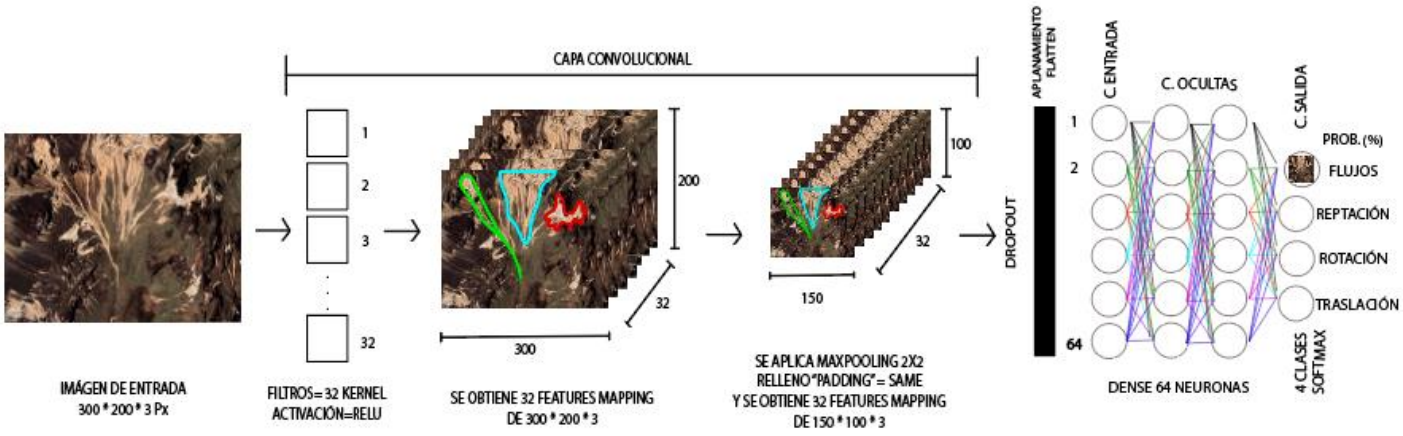

Fig. 5 Representación gráfica del modelo de red convolucional.

La capa de convolución es una operación matemática, que permite que en una función de alguna manera se apliquen otras funciones (mezcla de funciones) de tal manera que se simplifica el modelo y se detecta de manera eficiente detalles pequeños, la base de

convolución está conformada por: 32 filtros, cuyo tamaño de kernel es de (3, 3), este valor es porque las imágenes son de rango 3 (tridimensionales: largo, ancho y canal). Se utilizó*: 'Función de activación (Relu)'* encargada de transformar los valores introducidos, anulando los valores negativos y dejando los positivos tal y como entran (Solsona, 2018), seguido de un '*Padding'* (Anh-Duc, Seonghwa, & Woojae, 2019), el cual permite preservar la información que se encuentra en los bordes de las imágenes de entrada, estableciendo un espacio de relleno requerido por la imagen, la función '*Maxpooling'* encargada de disminuir la altura y longitud de la imagen, pero manteniendo su profundidad, es decir las imágenes de 200*300 Px se redujeron a 100*150 Px. (Artola, 2019), este proceso se resume en la reducción de las dimensiones de la imagen, con el objetivo de reducir el coste computacional. Posteriormente se utilizó la función *'Dropout'*, esta es una técnica de regularización para reducir el sobreajuste en redes neuronales desactivando un número determinado de neuronas en una red neuronal de forma aleatoria obligando a las neuronas cercanas a no depender tanto de las neuronas desactivadas, dicha función toma valores entre 0 y 1, en el modelo se utilizó 0.5 de tal manera que la probabilidad de que las neuronas se desactiven sea del 50%. (SITIOBIGDATA, 2018).

Para la fase del clasificador las funciones utilizadas son: Función '*Flatten*' conocida como aplanar se utiliza para convertir los datos multidimensionales en un vector de características 1D, una vez aplanada la matriz la capa resultante se la conoce como capa de entrada. (Geeksforgeeks, 2020)). La capa dense en el modelo uso 64 neuronas. El '*dense'* son las capas de cálculo que conectan cada neurona en una capa con todas las salidas de la capa anterior y están compuesta por varias capas ocultas. (ICHI.PRO, 2020). Estas funciones de activación se muestran en un sumario (Figura 6), donde constan las dimensiones iniciales de las imágenes, filtros, nuevas dimensiones de las imágenes (Maxpooling), número de entradas (Flatten), neuronas (Dense) y el total de parámetros analizados.

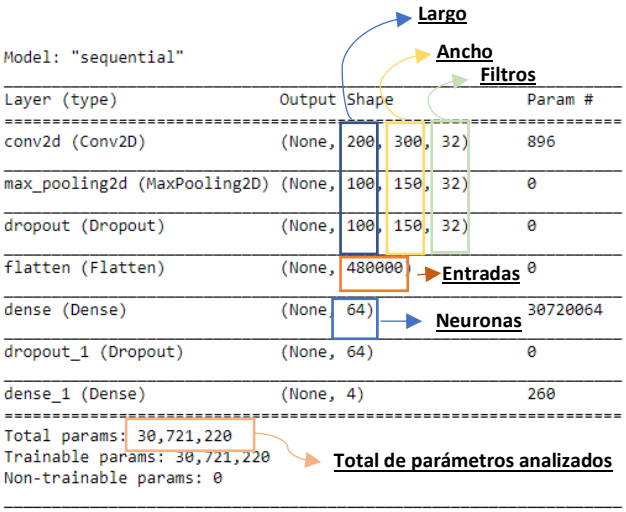

Fig. 6 Sumario del modelo.

El resultado final es una capa de salida con la respuesta codificada, donde tendremos tantas neuronas como clases. Cada salida la interpretaremos como la probabilidad de que el dato de entrada pertenezca a dicha clase. Finalmente, usaremos *'Softmax'* como función de activación, esta función transforma y asigna valores de probabilidad decimales a cada clase (Figura 6). Esas probabilidades decimales deben sumar 1.0 y el valor más alto corresponde a la clase predicha (Rivera, M. 2018). El resultado final (predicción), se trata de una palabra correspondiente a una de las clases indicadas anteriormente, en nuestro caso pueden ser (Flujos, Reptación, Rotación o Traslación).

2.7 Entrenamiento y evaluación del modelo

2.7.1. Entrenamiento del modelo

El modelo secuencial de la red detectó un total de 30721220 parámetros y mediante la instrucción: '*landslides_train ()'* se entrena el modelo para 150 épocas. Una época o '*epochs'* es un hiperparámetro que define el número de veces que el algoritmo de aprendizaje funcionará en todo el conjunto de datos de entrenamiento. (Browniee J., 2018)

Una época significa que cada muestra del conjunto de datos de entrenamiento ha tenido la oportunidad de actualizar los parámetros internos del modelo y se compone varios lotes, para nuestro código se trabajó en lotes de 64 imágenes con ayuda de la instrucción '*batch:size = 64'*, en la etapa de entrenamiento el modelo de red neuronal aprende a discriminar cada tipo de deslizamiento analizado en base a las características obtenidas en la etapa de creación de modelo. La etapa de entrenamiento de red neuronal arroja como resultado 76% de precisión. El proceso de entrenamiento duró aproximadamente una hora con 5 minutos, dicho tiempo varía dependiendo de las características del equipo en el que se corra el modelo. En el código se establece los parámetros finales a utilizar estos fueron: 150 épocas, Batch size=64. '*El Verbose = 1'*, no es más que el texto indicado, el valor de 1 incluye tanto la barra de progreso como una línea por época. Finalmente con *'landslides_model.save()*', guardamos el entrenamiento de la red, para reutilizarla en el futuro y evitarnos el proceso de entrenamiento.

2.7.2. Evaluación del modelo

Mediante la instrucción *'test_eval = ()*', se evalúa los valores de perdida y precisión del modelo, posteriormente se realiza las gráficas de precisión y error (Figura 7) del modelo con ayuda de la librería 'Matplotlib', donde los puntos indican los valores del entrenamiento y las líneas son los valores de validación del modelo.

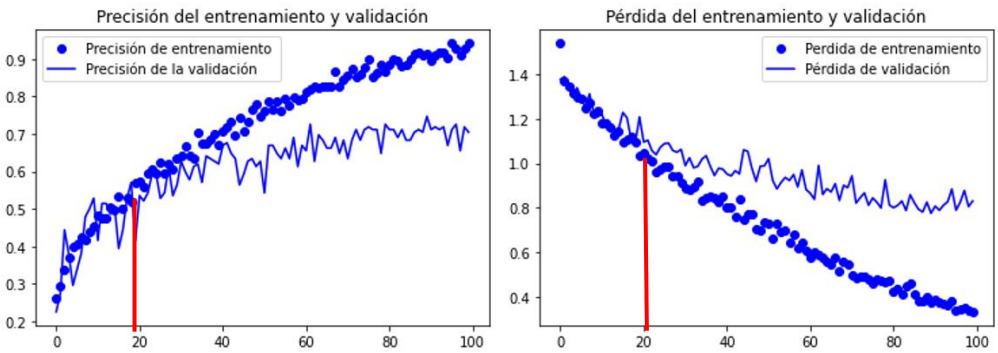

Fig. 7. Precisión y error del Modelo con datos de entrenamiento y validación.

Se puede observar en la Figura 7, que el modelo empieza con una precisión del 20%, conforme sigue su entrenamiento con 150 épocas, este incrementa sus valores de precisión obteniendo valores máximos de entrenamiento del 96% y validación de 76%, la línea de validación posee algunos picos, debido a que aún falta un 24% de precisión para llegar al 100% y esto se puede conseguir aumentando y depurando las imágenes. Posterior a la validación se realiza la evaluación de imágenes correctas se obtuvo un máximo de 226; mientras que las incorrectas se tuvo un mínimo de 74 imágenes, la comparación de estos valores revela que la precisión del modelo es adecuada.

Además, la matriz de confusión evalúa el rendimiento de un modelo de clasificación, comparando los valores reales con los predichos por el modelo, las letras VP representa un valor verdadero positivo (El valor real fue positivo y el modelo predijo un valor positivo), FP (Falso positivo), el valor real fue negativo pero el modelo predijo un valor positivo), FN (Falso negativo), el valor real fue positivo, pero el modelo predijo un valor negativo y finalmente VN (Verdadero negativo), el valor real fue negativo y el modelo predijo un valor negativo. Como se observa en la siguiente ilustración las métricas de la matriz de confusión evalúan la matriz y minimiza los falsos negativos y los falsos positivos (Figura 8-a); finalmente estos valores son usados para determinar las métricas como: Precisión, Recall y Accuracy (*Figura 8-b*),

a

		Valores reales	
		Positivo	Negativo
Valores predichos	Positivo	VP	FP
	Negativo	FN	VN

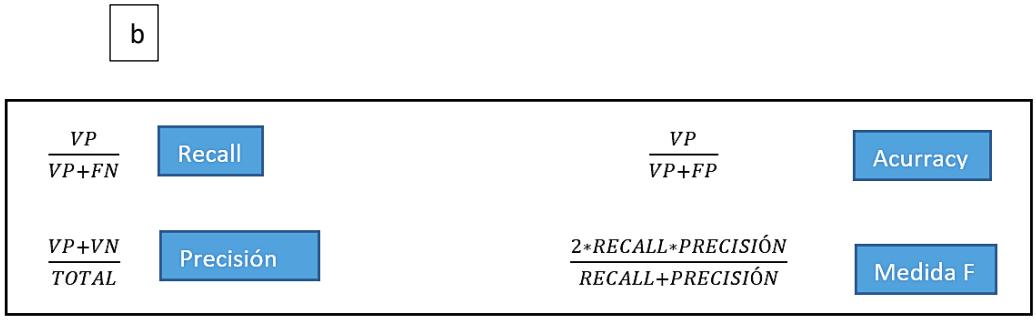

Fig. 8 Métricas de la Matriz de confusión.

En la figura 9, se observa la matriz de confusión definida por el modelo, con un total de 300 datos de los cuales 231 son verdaderos (132 positivos y 99 negativos) y 69 son falsos (44 negativos Y 25 positivos). La precisión se obtiene de la suma de los valores verdaderos (negativos y positivos) dividido para el valor total de datos, teniendo como resultado una precisión de 77%, muy cercano al obtenido en el modelo del entrenamiento (76%).

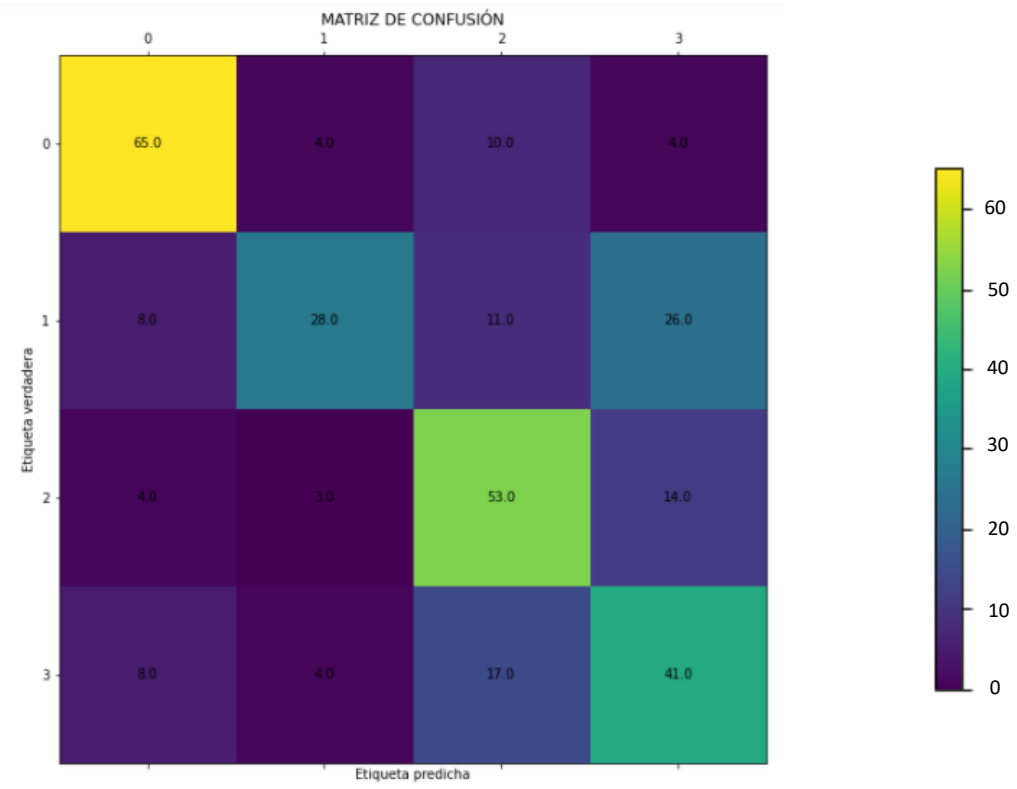

Fig. 9 Valores de la Matriz de Confusión

Los resultados obtenidos (Figura 10), indican que las métricas de Accuracy es de 71%, es decir, que de las predicciones que se realizaron el 71% de interacciones arrojaron una clase correctamente; la métrica de Precisión refleja mayor porcentaje en la clase 0 (81%), es así que la clase de deslizamientos de tipo flujo presenta mayor cantidad de interacciones correctas, mientras que la clase 1 y 2 presentan precisiones relativamente bajas con el 65% y 67%, respetivamente; Recall muestra un valor de 84% en la clase 0, por lo tanto, la mayoría de las imágenes de la clase 0 fueron identificadas y asignadas correctamente a su tipología (predicción correcta de valores verdaderos).

	precision	recall	f1-score	support
Class 0	0.81	0.88	0.84	74
Class 1	0.65	0.70	0.67	84
Class 2	0.67	0.76	0.71	74
Class 3	0.75	0.51	0.61	71
accuracy			0.71	303
macro avg	0.72	0.71	0.71	303
weighted avg	0.72	0.71	0.71	303

Fig. 10 Resultado de Métricas de la Matriz de Confusión.

2.8 Predicción de imágenes

Para la predicción se utilizó imágenes nuevas, que no corresponden a las utilizadas para la generación del dataset. Estas imágenes fueron redimensionadas al mismo tamaño de las entrenadas en el modelo, con el objetivo de mejorar la precisión de la predicción.

Estas imágenes corresponden a la zona de estudio analizada “Vía Achupallas-Guasuntos”, analizado un total de 41 FRM de estos 41 FRM, 11 son Rotacionales, 6 Traslacionales, 6 Reptacionales y 18 tipo Flujo, distribuidos a lo largo de la Vía Achupallas-Guasuntos (Figura 11).

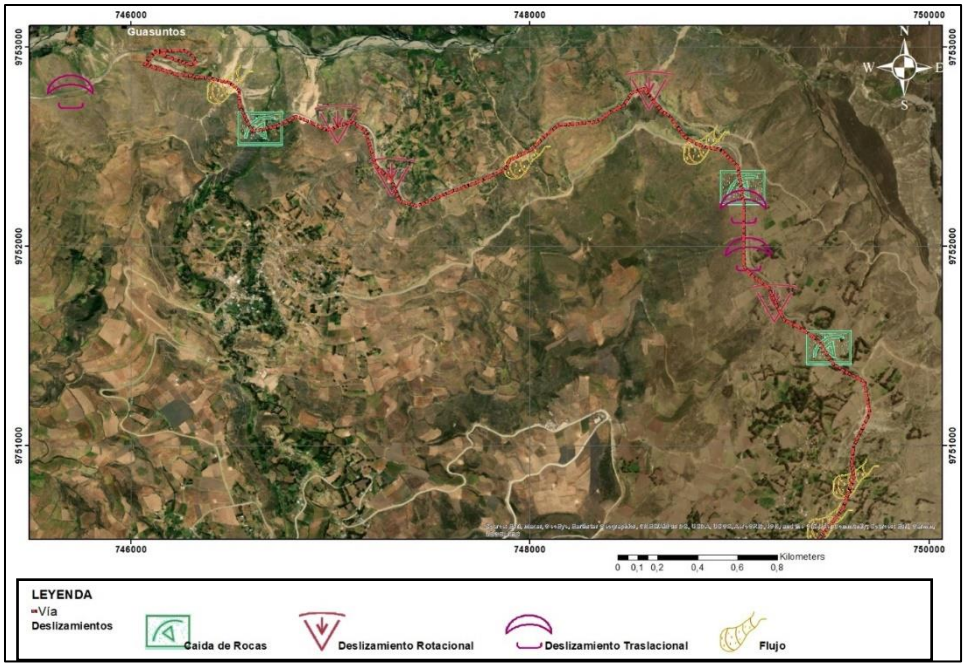

Fig. 11 Mapa de deslizamientos de la vía Achupallas-Guasuntos. Fuente: (Qgis 2021).

Las imágenes utilizadas para la predicción se encuentra dentro del disco local D, en la carpeta Imágenes_Predicción, al ingresar cada imagen al modelo esta será redimensionada a un tamaño de 200 x 300 con ayuda de skimage.transform y la instrucción *'image_resized = resize(image, (200, 300))'* , para posteriormente ser normalizada, esto se consigue gracias a las instrucciones: *'X = np.array(images, dtype=np.uint8)*' que convierte de lista a numpy, y con '*test_X = test_X / 255'* divide cada pixel para 255 obteniendo valores entre 0 y 1. Finalmente la predicción del tipo de FRM, se realiza con la instrucción '*predicted_classes = landslides_model.predict (test_X)',* el cual mediante el modelo entrenado asigna porcentajes de probabilidades a cada categoría definida anteriormente (Flujos, Rotacionales, Reptacionales y Traslacionales), el resultado predicho es la clase que obtuvo mayor porcentaje de probabilidad y el resultado se visualiza de manera textual en la interfaz de Jupiter Notebook. Al tener una precisión del 76%, se obtendrá en la predicción imágenes correctas e incorrectas, las mismas que se han ejemplificado a continuación.

2.8.1. Predicción de una Imagen Correcta (Figura 12)

Predicción de Nuevos deslizamientos

```
from skimage.transform import resize

images=[]
# Elejir las imagenes de la zona de estudio
filenames = ['D:\Imagenes_prediccion/Frm05.jpg']

for filepath in filenames:
    image = plt.imread(filepath,0)
    image_resized = resize(image, (200, 300),anti_aliasing=True,clip=False,preserve_range=True)
    images.append(image_resized)

X = np.array(images, dtype=np.uint8) #convierto de lista a numpy
test_X = X.astype('float32')
test_X = test_X / 255.

predicted_classes = landslides_model.predict(test_X)

for i, img_tagged in enumerate(predicted_classes):
    print(filenames[i], deslizamientos[img_tagged.tolist().index(max(img_tagged))])
```

```
D:\Imagenes_prediccion/Frm05.jpg ROTACIONALES
```

Predice Rotación correctamente

Fig. 12 Predicción correcta de una imagen nueva de la vía Achupallas-Guasuntos.

2.8.2. Predicción de una Imagen Incorrecta (Figura 13)

```
from skimage.transform import resize

images=[]
# Elejir las imagenes de la zona de estudio
filenames = ['D:\Imagenes_prediccion/Frm34.jpg']

for filepath in filenames:
    image = plt.imread(filepath,0)
    image_resized = resize(image, (200, 300),anti_aliasing=True,clip=False,preserve_range=True)
    images.append(image_resized)

X = np.array(images, dtype=np.uint8) #convierto de lista a numpy
test_X = X.astype('float32')
test_X = test_X / 255.

predicted_classes = landslides_model.predict(test_X)

for i, img_tagged in enumerate(predicted_classes):
    print(filenames[i], deslizamientos[img_tagged.tolist().index(max(img_tagged))])
```

```
D:\Imagenes_prediccion/Frm34.jpg REPTACION
```

Predice como Reptación a un Flujo

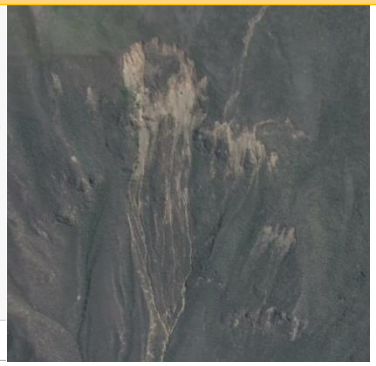

Fig. 13 Predicción incorrecta de una imagen nueva de la vía Achupallas-Guasuntos.

3 Discusión de Resultados

La creación de la red neuronal fue un proceso de prueba y error, donde se modificó distintos parámetros como el número de capas convolucionales, el número de épocas, incluso las dimensiones del Dataset, para mejorar la precisión del modelo, este proceso de modificación se realizó en 5 pruebas, donde se puede destacar: que las divisiones del Dataset (80, 20% y 60, 40%) no dieron buenos resultados, por lo cual se eligió la división

70, 30%. En el modelo de clasificación con 2 o más capas convolucionales, la precisión del modelo se reducía considerablemente y no superaba el 40%, se optó por trabajar con 1 sola capa, 64 neuronas obtuvieron mejores resultados que 32 neuronas, En los filtros con 32 se obtuvo mejor precisión, que con 16 u 8 filtros. Finalmente, en las épocas se trabajó desde 10 hasta 300 épocas probando de 50 en 50 épocas, los resultados indican que pasadas las 125 épocas la tendencia se mantenía, por lo cual se optó por trabajar con 150 épocas, otros factores influyentes en la precisión del modelo fueron: el tipo de imágenes y sus características, al eliminar imágenes que no provenían de una fuente satelital (afloramientos), el error disminuía considerablemente, la precisión máxima alcanzada es de 96% en el entrenamiento y 76% en la validación de estos datos, es decir son superiores al 70% y es un modelo aceptable, esto se corrobora con los datos de la matriz de confusión, ya que 2 de las clases obtuvieron en el f1-score (la relación entre la identificación y precisión de cada clase) valores superiores al 0.8, mientras que el valor más bajo fue de 0.67, un modelo se considera válido siempre y cuando, las clases del modelo tengan valores similares o superiores al 0.7 en el f1-score.

4 Conclusiones

• La aplicación de Deep Learning en el análisis de FRM permite generar un modelo de clasificación multiclase, el cual extrae características de un dataset para predecir el tipo de FRM en determinadas zonas, disminuyendo tiempo y reduciendo costos en el análisis de este tipo de fenómenos.

• Se logró elaborar un dataset confiable pero no muy extenso, debido a la complejidad de la obtención de datos, Google Earth Pro representa una excelente fuente de información, ya sea por su facilidad de adquisición como por la calidad de imágenes que ofrece. Cada una de las imágenes que conformen un buen dataset deben poseer la misma extensión, resolución vertical y horizontal, cada una de la clase debe contener el mismo número de

imágenes, con el fin de mejorar el proceso de ejecución del modelo, y evitar imágenes duplicadas que puedan sobreentrenar el modelo.

• Se diseñó un modelo Deep Learning a partir de uno preexistente, realizado para clasificación de imágenes deportivas y que ha sido adaptado a nuestro proyecto con características propias en base a los objetivos planteados del proyecto, de tal manera que los parámetros del modelo final fueron: 1 red convolucional, 32 filtros, 64 neuronas, 150 épocas y división del dataset en (70 - 30%). Estos parámetros fueron obtenidos tras un proceso de prueba y error de varios ensayos.

• Al aplicar el modelo de red neuronal para 41 FRM en la vía Achupallas – Guasuntos, se obtuvo un resultado de 10 imágenes cuya predicción fue errónea, lo cual corresponde a un 24% de error y un 76 % de precisión. De estos 41 FRM, 11 son Rotacionales, 6 Traslacionales, 6 Reptacionales y 18 tipo Flujo, esta información, complementada con estudios de campo en la zona, permitiría gestionar la correcta aplicación de obras de mitigación en la vía analizada.

5 Referencias bibliográficas

Anh-Duc, N., Seonghwa, C., & Woojae, K. (2019). Distribution Padding in Convolutional Neural Network. *IEEXplore*.

Artola, A. (2019). *Clasificación de imágenes usando redes neuronales convolucionales en Python.* Sevilla: Universidad de Sevilla.

Bagnato, J. (2018). *Clasificación de Imágenes en Python.* Obtenido de https://www.aprendemachinelearning.com/clasificacion-de-imagenes-en-python/

Browniee, J. (2018) Difference Between a Batch and an Epoch in a Neural Network. Obtenido de: https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/

Eras, M. (2014). *Determinación de zonas susceptibles a movimientos en masa en el Ecuador, a escala 1:1000000, utilizando en método de ponderación de parámetros.* Quito: Universidad politécnica Nacional.

Flores, A. (2019). Técnicas para la predicción espacial de zonas susceptibles a deslizamientos. *Creative Commons*, 1-29.

González Vilas, L. (2013). *Teledetección y técnicas de aprendizaje automático supervisado aplicados a la discriminación de vertidos de hidrocarburos.* Vigo: Universidad de Vigo.

Geeksforgeeks, (2020). Flatten a Matrix in Python using NumPy. Obtenido de: https://www.geeksforgeeks.org/flatten-a-matrix-in-python-using-numpy/

IArtificial.net. (2013). *Librerías de Python para Machine Learning.* Obtenido de https://www.iartificial.net/librerias-de-python-para-machine-learning/#Librerias_de_Python_para_Visualizacion.

ICHI.PRO, (2020). Aprendizaje profundo con clasificación de imágenes CIFAR-10. Obtenido de: https://ichi.pro/es/aprendizaje-profundo-con-clasificacion-de-imagenes-cifar-10-110688052776313

Molinero, J. (2019). *Reconocimiento de objetos en Imágenes mediante técnicas de aprendizaje profundo: Rede neuronales Convvolucionales.* Madrid: Universidad Complutense de Madrid.

Naranjo, E. (2012). *Estudio de factores de susceptibilidad (Topográfico, Geológicom Hidrológico, Geomorfológico, Desarrollo social y Económico) en la provincia de Chimborazo.*

Rivera, M., (2018). Red Neuronal Multicapa en Keras. Obtenido de: http://personal.cimat.mx:8181/~mrivera/cursos/aprendizaje_profundo/nn_multicapa/nn_multicapa.html

SITIOBIGDATA, (2018). Redes Neuronales: vista rápida de implementación. Obtenido de: https://sitiobigdata.com/2018/08/28/redes-neuronales-convolucionales-vistarapidaimplementacion/

SGR. (2014). *Agenda de reducción de riesgos - Provincia de Chimborazo.* Obtenido de Aecretaria de Gestión de Riesgos: https://biblio.flacsoandes.edu.ec/libros/digital/54568.

Solsona, A. (2018). Facial Expression Detection using Convolutional Neural Networks. *Universitat Politencia de Catalunya*.

Valdiviezo, A. (2014). *Propuesta metodológica para la aplicación del slope mass rating continuo (SMR-C) mediante un sistema de información geográfica en los taludes de la vía La Moya – Achupallas.* . Guayaquil: Escuela Superior Politécnica del Litoral.