DESCRIPCIÓN Y MEDICIÓN AUTOMÁTICA DE RIPIOS DE PERFORACIÓN MEDIANTE IA

Jhoel Ortiz, Paola Vargas, y Christian Mejia-Escobar

Facultad de Ingeniería en Geología, Minas, Petróleos y Ambiental (FIGEMPA)
Universidad Central del Ecuador
Quito, Ecuador

djortizm@uce.edu.ec, pvvargas@uce.edu.ec, cimejia@uce.edu.ec

ABSTRACT

Los ripios de perforación son importantes porque otorgan información casi instantánea de las formaciones durante la perforación de un pozo. Sin embargo, la subjetividad del técnico y el poco tiempo para analizarlos pueden generar falencias durante su descripción. Este trabajo busca compilar un robusto conjunto de datos y entrenar un modelo de IA para generar descripciones textuales y verbales de ripios de perforación de manera automática. Un beneficio adicional incluye la medición de clastos. Para ello, el usuario deberá ingresar la imagen a describir y la escala de la misma para que el programa dé el porcentaje de los componentes de la roca y la descripción de cada litología, seguido de la descripción oral del texto. Por otro lado, la medición le entrega al usuario la opción de realizarlo de forma automática, en la que el programa escoge el mejor fragmento para ser medido, o el componente semiautomático, en el que la persona puede señalar manualmente el fragmento de su interés. Todo esto se verá plasmado en un Notebook de Google Colab y una página web interactiva con el usuario.

Keywords Drill cuttings · IA · CNN · Transformer · Description · Measurement · Google Colab · MobileNet

1. Introducción

El mundo moderno funciona a base de energía. Una de las principales fuentes de energía es el petróleo, el cual abastece gran parte de la demanda energética a nivel global. Incluso, este recurso desempeña un papel determinante para el desarrollo económico de muchos países. En un entorno de mercado cada vez más exigente, la industria petrolera requiere garantizar una producción y suministro continuos para las necesidades diarias, por lo que busca permanentemente mejorar la eficiencia y optimización de sus procesos y operaciones.

La perforación de pozos petroleros es un procedimiento de extracción que permite el acceso a los depósitos de petróleo localizados debajo de la superficie. Durante este proceso, una broca giratoria atraviesa formaciones geológicas que rompe capas de roca y otros materiales. Estos fragmentos son conocidos como *ripios de perforación* (*cuttings*), que son transportados a la superficie por los lodos de perforación que se inyectan en la broca para enfriarla y retirar los fragmentos que se acumulan al fondo del hoyo. Una vez en superficie, los ripios llegan a un agitador que los separa del fluido y se acumulan para su almacenamiento. Posteriormente, estos se limpian y recolectan para su análisis.

El análisis de los ripios de perforación en tiempo real produce datos esenciales para evaluar las características geológicas y geofísicas del subsuelo, comprender la naturaleza de las rocas subterráneas y sus propiedades como la porosidad y permeabilidad, construir modelos geológicos, determinar la presencia de hidrocarburos así como la distancia y el tiempo para alcanzar la zona de interés, caracterizar el yacimiento, establecer la correlación entre pozos y prevenir problemas de estabilidad y colapso estructural del pozo.

Las muestras de ripios consituyen una de las fuentes de información más directas e instantáneas, la cual abarca un rango estratigráfico más amplio en comparación con otros tipos de datos que se concentran únicamente en intervalos

específicos. A medida que se extraen, los ripios de perforación se vuelven un insumo indispensable que ayuda a geólogos, ingenieros de petróleo, técnicos de perforación, operadores y otros profesionales especializados a tomar decisiones sobre la presencia y ubicación del hidrocarburo.

La metodología comúnmente utilizada para obtener información a partir de muestras de ripios consiste en la inspección visual de las muestras recuperadas por parte del geólogo de campo, a través de microscopios de alta resolución y otros instrumentos de análisis, lo que conlleva a generar una descripción explicativa según la experiencia y conocimiento del profesional. Como resultado, se obtiene un reporte que reúne las descripciones textuales de cada muestra(Tolstaya, Shakirov, Mezghani, y Safonov, 2023).

Problema. Algunas problemáticas surgen durante el análisis convencional de muestras de ripios, tales como: (1) descripciones sesgadas debido a la naturaleza subjetiva de los resultados, (2) un proceso que exige alta demanda de tiempo pues requiere un análisis meticuloso de cada muestra, (3) errores humanos debido a la contaminación de los cortes con lodo de perforación, (4) una descripción errónea debido a la similitud entre los tipos de litología, y (5) errores humanos debido a que al acercarse a la zona de interés se obtienen más muestras de ripios que deben ser analizados a mayor detalle en intervalos de tiempo cada vez más cortos. Por ende, mejorar la extracción de información útil de dichas muestras es una prioridad en el desarrollo del campo investigativo de esta área (Wang, Yang, Zhao, y Wang, 2018).

Propuesta. El uso de la Inteligencia Artificial (IA) ha demostrado ser una alternativa exitosa para el tratamiento de problemas relacionados con análisis visual y explicación textual. Una de las técnicas mayormente implementadas en esta temática es el aprendizaje profundo o Deep Learning (DL). En este estudio se busca implementar un sistema que combine herramientas de DL de visión por computadora, procesamiento del lenguaje natural y conversión de texto a voz para producir descripciones de ripios a partir de imágenes. El uso de modelos basados en redes neuronales convolucionales (CNNs) y redes transformers será el pilar para la solución propuesta. Para que el resultado se considere satisfactorio, este debe poseer una descripción textual de la litología, así como porcentajes de sus componentes, la escala a la que fue tomada la imagen y al menos la medida de algunos de sus clastos para que puedan ser utilizados como referencia.

Método. El desarrollo del proyecto comprende varias etapas: la creación del conjuto de datos, la extracción de carcaterísticas de la imagen, la descripción textual a partir de estas características, la conversión del texto a un formato de audio y la medición de imágenes de ripios de perforación. La creación del conjunto de datos demanda recolectar un número significativo de imágenes de 1600 x 1200 pixeles, mismas que son divididas en imágenes más pequeñas de 500 x 500 pixeles y nombradas de acuerdo con las litologías presentes en la imagen. Seguidamente, se crean las respectivas descripciones textuales, cada una comienza con la lista de litologías y el porcentaje en el que se encuentran presentes en la foto. Adicionalmente, se incluye cada uno de los tipos de rocas según las características enumeradas en la Tabla 1. Una vez preparado el conjunto de datos, una red convolucional se encarga de extraer las características más relevantes de cada fotografía. Estas características son empleadas para entrenar una red transformer que aprende a relacionarlas con las descripciones, obteniendo un modelo de descripción automático. Adicionalmente, se ha implementado la conversión del resultado textual en voz por la computadora. Finalmente, se agrega un componente de medida de carácter automático y semiautomático. Ambos funcionan con la misma imagen de entrada y la escala, sin embargo, en el caso automático, el usuario no debe realizar ningún paso adicional porque el programa se encarga de escoger y medir el clasto, mientras que, en el caso semiautomático, el ususario puede escoger y dibujar manualmente el clasto que desea medir. Como resultado en ambos casos, el programa mide los ejes mayor y menor, además de generar una tabla con otros parámetros de medida como radio, circunferencia, área, perímetro, entre otros.

Contribuciones. Los aportes concretos de nuestro trabajo son:

- Un conjunto de datos mixto que combina 1100 fotografías clasificadas en 11 categorías con 100 fotos cada una, y 1100 descripciones correspondientes a cada una de las imágenes compiladas en un documento de texto. Este recurso es un aporte a los escasos conjuntos de datos que permitan entrenar modelos de IA para clasificación de litología de los ripios de perforación Becerra1, de Lima, Galvis-Portilla, y Clarkson (2022).
- Un sistema de IA para la descripción automática de imágenes digitales de ripios de perforación. A esto se añade la medición del eje mayor y menor de un clasto de la muestra, lo cual se incluye en una tabla junto con parámetros como área, perímetro, entre otros.
- La aplicación es presentada en dos versiones: 1) un notebook de programación interactivo en la plataforma Google Colab, y 2) una página web mucho más amigable. En ambos casos, el usuario puede subir sus propias imágenes para ser descritas de manera textural y oral.

Estos productos son de carácter público para usuarios en general y ofrecen una alternativa automatizada que pueda servir de apoyo no solo para los profesionales inmersos en el área de la perforación, sino también a estudiantes o personas relacionadas a la industria petrolera que quieran adquirir conocimientos sobre el tema.

2. Trabajos relacionados

El interés en utilizar la Inteligencia Artificial (IA) para abordar problemas en el campo de la geología está en constante crecimiento. Diversas investigaciones han evidenciado las valiosas contribuciones de la IA en la optimización de procesos y la reducción de errores en el análisis de datos litológicos a través de ripios de perforación. En esta sección, vamos a explorar los estudios previos más relevantes llevados a cabo por otros autores, centrándonos en la problemática específica abordada, los datos utilizados, la metodología empleada y los resultados obtenidos más destacados.

Kathrada y Adillah (2019) se enfrentaron al problema de la clasificación incorrecta de las imágenes de ripios de perforación. Esta inexactitud se debe a que las descripciones de las imágenes varían según el experto que las analiza, o por rapidez, porque no se describen en el momento de la toma, lo que conduce a una clasificación errónea al realizar las descripciones posteriormente. Se plantea el uso de redes neuronales convolucionales entrenadas con imágenes de ripios dentro de 4 clases. Inicialmente, se empleó un modelo de transfer learning basado en AlexNet, sin embargo, al ser un problema específico, las pruebas demostraron la conveniencia de una red personalizada, a la que denominaron *Bayesian optimised network*. Nuevas fotos nunca vistas por el modelo se consideraron para la fase de validación. Los resultados indican una precisión satisfactoria y que exhibe potencial para mejora futura.

Tamaazousti, François, y François (2020) implementan el modelo convolucional ResNet para la identificación y cuantificación automatizadas de tipos de roca a partir de ripios de perforación. Se utilizó un conjunto de 300 imágenes compuesto de muestras secas y húmedas clasificadas en 3 tipos de roca. La precisión del modelo alcanzó 98.2 % en muestras simples y mixtas. También se obtienen buenos resultados con muestras húmedas, lo que abre la oportunidad de trabajar con imágenes RGB de bajo costo, así como también eliminar o al menos reducir la duración del proceso de secado durante la interpretación. Las predicciones de cuantificación resultan de 95 %, siendo este el primer estudio enfocado en esta temática. Los autores consideran como siguiente paso la aplicación en datos de campo y adaptar la inferencia a las condiciones operativas.

Becerra1 y cols. (2022) reconocen que, a pesar de los avances en IA, existen pocos conjuntos de datos de acceso libre que permitan entrenar a un modelo para clasificar la litología de los ripios de perforación. No obstante, empleando una base de 16700 imágenes divididas en 5 clases, correspondientes a un reservorio de baja permeabilidad al oeste de Canadá, entrenaron un modelo ResNet-18. Los resultados fueron satisfactorios, con una precisión del 88 al 91 %, demostrando que las redes convolucionales son útiles para una clasificación rápida de litologías que después deberá ser revisada por geólogos. A diferencia de otros proyectos, se resalta que el conjunto de datos y el código se encuentran disponibles en la red y son de libre acceso.

Ismailova, Dochnika, al Ibrahim, y Mezghani (s.f.) abordan la estimación de los tamaños e interpretación de partículas en ripios de perforación realizadas manualmente y en tiempo real, lo que conduce a resultados imprecisos e inconsistentes. Proponen un método automatizado para identificar tamaños de grano usando 402 imágenes digitales. Estas imágenes corresponden a 4 pozos y poseen litologías con variados tamaños de grano. En cada grano se establecieron ejes fijos y se realizaron las respectivas mediciones. Al comparar el error entre los resultados obtenidos por el modelo y las mediciones manuales, apenas existe un 10 % de error, por lo que el modelo se considera satisfactorio.

Tolstaya y cols. (2023) emplean MobileNet_2 en la predicción litológica de rocas a partir de imágenes de ripios de perforación. Se propone un flujo de trabajo para la selección de archivos de imagen, denotando un aumento en los valores de métrica conforme las imágenes son mejor procesadas. La precisión del modelo alcanza entre 77 % a 95 % a lo largo de 3 etapas de entrenamiento, cada una con un conjunto de datos más elaborado. Adicionalmente, los resultados fueron evaluados con las muestras de un pozo ajeno al proyecto, dando en el mejor de los casos un rendimiento del 66 %, lo cual es considerado como un valor aceptable de incertidumbre en la predicción. La siguiente etapa es la optimización de la solución propuesta para muestras con litología mixta en base a métodos de regresión.

M. Sitar y Leary (2023) crearon un paquete de Python ejecutable en Google Colab que mide el tamaño de los granos de zircones. Este proyecto se debe a la importancia de los zircones para determinar el transporte de clastos a partir de su morfología y también como indicador de provenanza; sin embargo, el código puede ser empleado para medir el tamaño de grano de cualquier clasto o mineral. Aunque el objetivo fue segmentar los clastos en granos individuales, se presentaba la división de cada grano en subgranos más pequeños y la identificación de bordes inexistentes. Esto se resolvió con modelos de deep learning que dejaron de confundir las fracturas intraclasto con los bordes de estos. Se creó el paquete denominado *colab-zirc-dims* escrito en Python 3.8 y que compila paquetes tanto estándar como no estándar. La librería Pillow se usa para cargar las imágenes y Matplotlib para crear y guardar las segmentaciones de los granos en la imagen. La interactividad del código es posible usando IPython, mientras que la librería Detectron2, desarrollada por Facebook, fue empleada para la construcción y entrenamiento del modelo. Se llevó a cabo el entrenamiento con varios modelos, entre ellos ResNet y VovNetV2-99, utilizando 1558 imágenes, de las cuales se identificaron y analizaron 16464 granos, obteniendo sus respectivas características. El código se encuentra en el repositorio GitHub, pero se

instala e interactúa con el usuario a través de Google Colab. Como resultado, se tiene un código interactivo automático y semiautomático donde se ingresan imágenes en formato PNG y a partir de ellas se extraen mediciones de cada grano.

Paucar, Mejía-Escobar, y Collaguazo (2024) reconocen la importancia de las secciones delgadas de roca dentro de varias áreas de las ciencias de la tierra, sin embargo, también recalcan el alto grado de subjetividad que poseen sus descripciones. Los autores elaboraron un dataset de 5600 imágenes de secciones delgadas bajo luz natural y polarizada distribuidas en 14 categorías con sus respectivas descripciones textuales para el entrenamiento de un modelo que combina la red EfficientNetB7 y la red Transformer para extracción de características y la descripción textual, respectivamente. Los resultados experimentales indican un valor de precisión de 0.892 y un valor BLEU de 0.71, considerado como aceptable.

Los trabajos previos se limitan mayormente a una tarea de clasificación de rocas buscando identificar 3 o 4 tipos de litología. Sin embargo, también hay trabajos que se enfocan en la medición de clastos y minerales. En cuanto al conjunto de datos utilizados, este va desde unos pocos cientos de imágenes hasta algunos miles. Se puede notar que el uso de deep learning es una alternativa prometedora, obteniendo resultados satisfactorios a través de modelos convolucionales de vanguardia como ResNet y MobileNet. Las experiencias dejadas por estas investigaciones previas, nos permiten identificar los nuevos aportes que proporciona el desarrollo del presente proyecto. Si bien el conjunto de datos que formamos para este trabajo es más reducido que el de algunos casos anteriores, la eficiencia de la red MobilNetV2 nos ha permitido identificar 7 tipos de litologías distribuidos en 11 clases, o combinaciones entre ellas. Hemos complementado la tarea de clasificación incluyendo la descripción textual y verbal, lo cual no es considerado hasta el momento en el caso de ripios de perforación. Cabe señalar que esta descripción define el porcentaje de roca que compone la imagen. Por otro lado, tomando como base el trabajo de (M. Sitar y Leary, 2023), ofrecemos al usuario la oportunidad de medir los fragmentos de ripios de forma automática y semiautomática.

3. Metodología

Comúnmente, la descripción y la medición de ripios de perforación son actividades efectuadas por expertos humanos a partir de la observación. Esto puede generar resultados subjetivos, dependientes del grado de conocimiento de la persona y que consumen tiempo. En los últimos años, la inteligencia artificial ha demostrado avances significativos en su objetivo de imitar y, en algunos casos, superar el desempeño humano en la realización de diversas tareas. Nuestra propuesta combina técnicas y herramientas de vanguardia en los campos de visión por computadora y procesamiento del lenguaje natural para el desarrollo de un sistema de descripción y medición automática de ripios de perforación. Este producto es útil para el ámbito petrolero y afines, tanto a nivel profesional como académico. Para este propósito, seguimos la metodología clásica de un proyecto de aprendizaje automático, la cual puede ser dividida en dos grandes etapas: los datos y el modelo.

3.1. Dataset

Primeramente, es necesario disponer del insumo principal del aprendizaje automático, es decir, el conjunto de datos. El dataset se encuentra compuesto de una combinación de imágenes y texto, en la que cada imagen cuenta con su respectiva descripción textual. Para que el programa pueda generar una correlación entre ambos, tanto la fotografía como la descripción cuentan con la misma codificación.

3.1.1. Imágenes

Los registros fotográficos y todo lo relacionado con los fragmentos de rocas (ripios) obtenidos durante las perforaciones de pozos petroleros, se consideran propiedad de cada empresa y es información sensible que no se libera al público por motivos de seguridad y estrategia. A través de un requerimiento especial a una empresa del sector petrolero, cuyo nombre omitimos por razones obvias de confidencialidad, hemos logrado el permiso para acceder a una parte de su base de datos conformada por imágenes y descripciones textuales asociadas. Se recolectaron 280 imágenes de ripios de perforación de 4 pozos de la Cuenca Oriente ecuatoriana correspondientes a distintas formaciones. Se recolectaron un total de 140 imágenes correspondientes a fotografías de ripios, cada una con dimensiones de 1600 x 1200 píxeles. Aumentamos el tamaño del dataset mediante la subdivisión de cada una de estas imágenes aprovechando su gran resolución. Cada foto fue recortada en 8 fragmentos más pequeños para incrementar el número de imágenes A partir de cada imagen, se extrajeron 8 partes o secciones de 500 x 500 píxeles (Figura 1). Debido a la naturaleza aleatoria de las litologías que se pueden encontrar en un ripio, el recorte de secciones en la imagen original permite tener variciones en la predominancia del tipo de roca dentro de una misma fotografía, por lo que se vuelve más robusta la base de imágenes. Por tanto, se generaron 2200 imágenes, mismas que se guardaron en formato JPG, debido a que tiene un menor peso que aquellas con formato PNG, lo cual vuelve más versátil y veloz el entrenamiento del modelo.

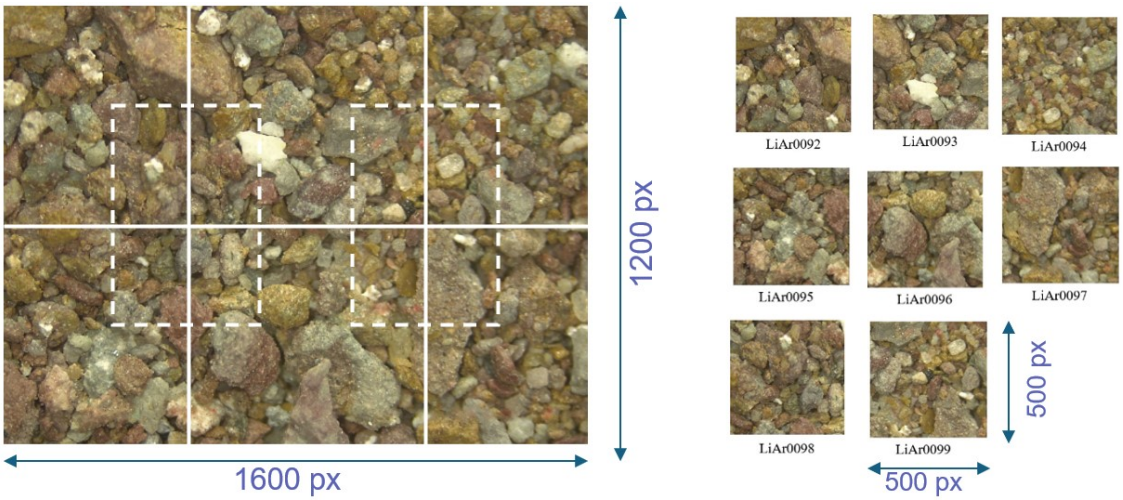

Figura 1: Ejemplo de la división de una de las imágenes originales de 1600 x 1200 píxeles. De cada una, se obtuvieron ocho imágenes más pequeñas de 500 x 500 píxeles.

Las imágenes se agruparon por clases en una estructura de carpetas, donde cada carpeta tiene el nombre de su respectiva clase. Se identificaron 7 clases de litologías: areniscas (A), arcillolita (Ar), conglomerado (Co), Caliza (Ca), caolín (K), limolita (Li) y lutita (Lu). Sin embargo, los ripios de perforación son muestras mixtas de fragmentos de roca, por lo que estas litologías generalmente no se presentan solas, sino que aparecen en grupos de dos o tres en las láminas de ripios de perforación. Por tal razón, se procedió a establecer clases por sus asociaciones litológicas más comunes, obteniendo así 11 clases con 200 imágenes cada una. Al contar con la misma cantidad de datos, las clases se encuentran balanceadas y se minimizan las posibilidades de un sesgo. Estas clases se indican seguidamente con el nombre de la carpeta entre paréntesis: lutita-caliza (LuCa), arenisca-limolita-arcillolita (ALiAr), limolita-arcillolita (LiAr), Arcillolita (Ar), conglomerado-arcillolita (CoAr), arenisca-lutita (ALu), arenisca-lutita-caolín (ALuK), lutita-caolín (LuK), lutita (Lu) y arcillolita-conglomerado-arenisca (ArCoA). La Figura 2 muestra un ejemplo de cada una de estas clases.

Cabe señalar el cambio de la codificación para que tuviera coherencia con la nueva clasificación litológica. Todo este conjunto de datos fue cuidadosamente etiquetado y clasificado. Las carpetas fueron nombradas según la asociación litológica que estas representan, mientras que el nombre de los archivos de imagen incluyen su clase y el número de archivo al que corresponde en su respectivo grupo.

La codificación se conforma por las inciales de las litologías presentes en la foto y número de imagen. También es imperativo que en el bloc de notas al final de cada codificación se agregue #0, debido a que en el código original existían 5 variaciones de la misma imagen y el autor, Nain (2021), las diferenciaba colocando #1-5 al final del código en las descripciones. En nuestro caso de estudio no existen variaciones de la misma foto y con afán de no realizar cambios innecesarios en el código se agrega este último elemento en las descripciones para que el programa pueda leerlo. También es importante mencionar que dentro del bloc de notas la codificación y la descripción se encuentran separados por una tabulación. Ejemplo: Si la muestra se encuentra conformada por arenisca, limo y arcilla el código sería ALiAr0001.jpg#0 |tabulador| Descripción.

Finalmente, todas las carpetas fueron almacenadas en la plataforma de *Google Drive*.

3.1.2. Descripciones textuales

Luego, realizamos las descripciones textuales de cada una de las imágenes considerando los parámetros más relevantes. Estas descripciones son almacenadas en un archivo de texto y están asociadas con las imágenes a través de un identificador alfanumérico. Así, conformamos un dataset mixto de imágenes y descripciones textuales.

Además de las imágenes provistas, también tuvimos acceso a sus respectivas descripciones de texto. Aunque las imágenes originales contaban con una descripción en la que se remarcaba la roca predominante y sus características, el proceso de recorte en fragmentos más pequeños origina que los ripios que se encuentran dispersos en la fotografía hacen que en ciertas áreas se acumule más de una litología que en otras. Es por ello que las descripciones tuvieron que ser modificadas por medio de un editor de texto convencional para mencionar la litología cada nueva imagen. Las nuevas definiciones se componen de dos partes principales . La primera parte enuncia las litologías y el porcentaje que

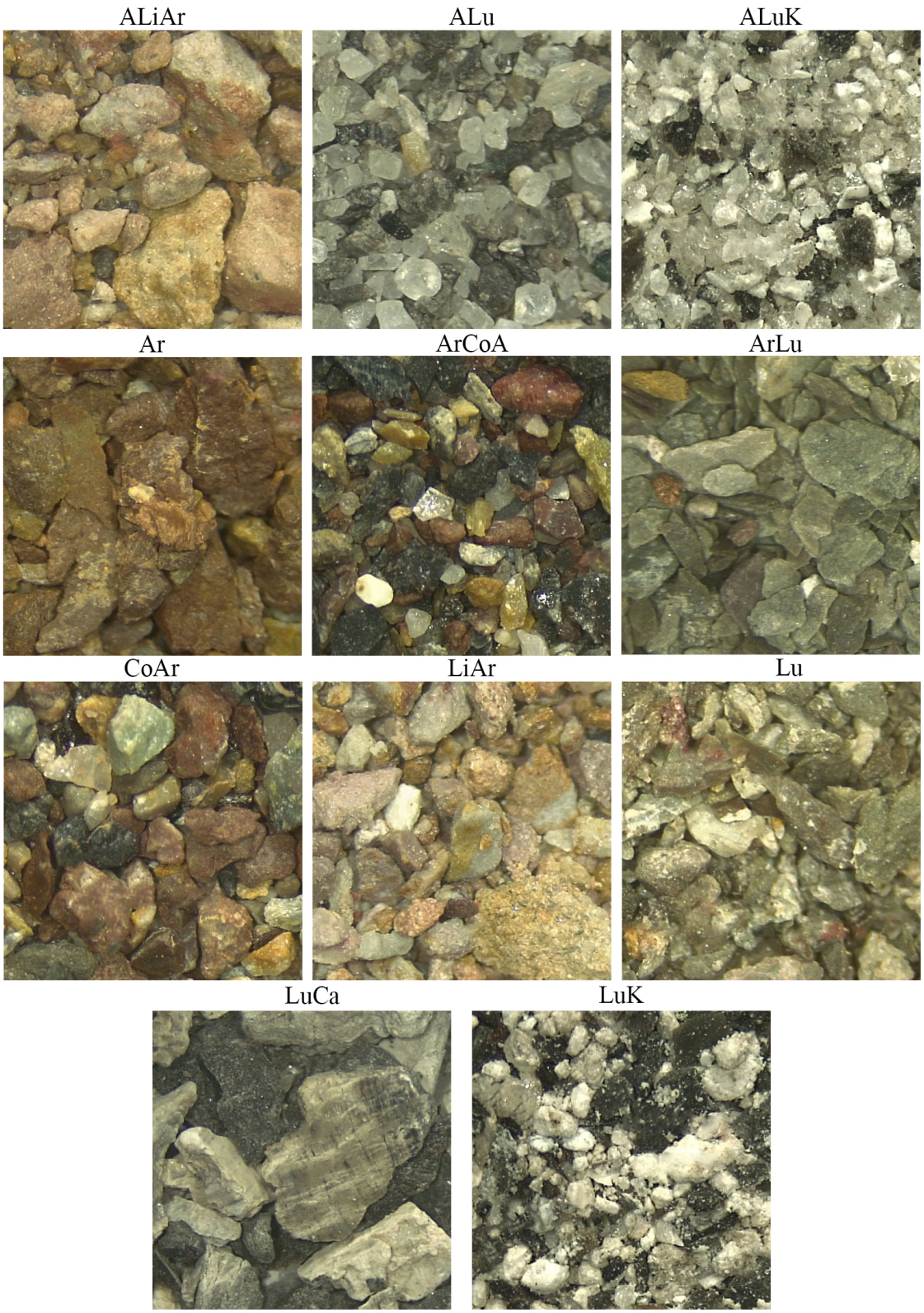

Figura 2: Clases de imágenes de ripios de perforación según su asociación litológica.

se aprecia de cada una en la fotografía. Estos porcentajes fueron interpretados por los autores de este trabajo y están sujetos al criterio y conocimiento previo en el tema. La segunda parte explica cada tipo de roca presente en la foto utilizando las características presentadas en la Tabla 1. Ejemplo: ALiAr006.jpg#0 Se presenta una muestra conformada por 100 % de arcillolitas, limolitas y areniscas en traza. La arcillolita es de color café, con forma blocosa y brillo grasoso. La limolita es de color gris, con forma blocosa y brillo terroso. La arenisca es cuarzosa, de color gris, grano fino, con clasto redondeados, suelta, con moderado sorteo y sin matriz.

Cuadro 1: Parámetros de descripción de las imágenes de láminas delgadas para los ripios de perforación.

Litología	Color	Forma	Brillo	Minerales	Tamaño	Redondez	Sorteo
Lutita	✓	✓	✓	-	-	-	-
Limolita	✓	✓	✓	-	-	-	-
Arcillolita	✓	✓	✓	-	-	-	-
Caliza	✓	✓	✓	-	-	-	-
Arenisca	✓	✓	✓	✓	✓	✓	✓
Caolín	✓	✓	✓	-	-	-	-
Conglomerado	✓	-	-	✓	✓	✓	✓

Las areniscas constituyen una de las litologías más importantes de la Cuenca Oriente porque son reservorios. Es así que en ellas se emplean las 8 características de la tabla. Con excepción de la arenisca, no todas las características aplican a una determinada litología.

A diferencia de las muestras de mano, en los ripios de perforación no se observa una sola litología, sino una mezcla de varios fragmentos de colores similares que pueden ser fácilmente confundidos entre sí. El color se describe en las 7 litologías, no porque sea característico sino porque ayuda a identificarlas en esa fotografía en específico.

En cuanto al brillo y la forma (cómo se rompen), estas características están relacionadas con el tamaño de grano, específicamente para el caso de la lutita, caliza, caolín, limolita y arcillolita. Por ejemplo, la lutita tiene un tamaño de grano <0.004 mm, a diferencia de la limolita que es entre 0.004 y 0.032 mm, por lo que al tener la limolita un mayor tamaño de grano posee un brillo terroso, mientras que en la lutita los granos son tan pequeños que se vuelven indistinguibles y le otorgan un brillo grasoso, similar al de una superficie lisa.

La forma y el tamaño nos permiten saber qué tan lejos viajó desde su fuente o roca originaria hasta su lugar de depositación. La redondez también nos permite saber el lugar de origen, pero se enfoca más en la calidad del reservorio. Si los granos son redondos generan una buena porosidad y permeabilidad para almacenar hidrocarburo. A esto también se suma el sorteo de la misma. Esta característica analiza si todos los granos presentan el mismo tamaño, a mayor homogeneidad en la forma de los granos, estos calzan entre sí uniformemente y dejan más espacios libres, lo que se traduce en un mejor reservorio.

Tanto la arenisca como el conglomerado se forman a partir de rocas o minerales preexistentes con un tamañano visible. Es por eso que en ambos casos se incluye esta característica. La arenisca usualmente está compuesta por cuarzo, pero en el caso del conglomerado varia enormemente.

Finalmente, el conglomerado además de tener una litología variada, también cuenta con diversos tamaños de grano. Su tamaño es mayor por mucho a las otras rocas mencionadas previamente por lo que se utiliza ese parámetro para diferenciarlo, además se caracterizan por ser sub-redondeado a redondeados y estar compuestos de varias rocas o minerales de ser el caso. Una vez más, se analiza el sorteo para determinar la calidad de un posible reservorio.

La codificación de las imágenes generadas a partir de la misma imagen mantiene una nomenclatura similar, pero sigue un orden secuencial, es decir, en cada clase se numeran del 1 al 200.

Por último, es importante la codificación y la extensión del archivo de descripciones. Este archivo debe estar en formato '.txt' y en la codificación usada por Nain (2021), la cual es ASCII. Si se utiliza otras codificaciones, por ejemplo UTF-8, se generarán errores de lectura al momento de entrenar el modelo, indicando que no existen los archivos que se están presentando.

3.2. Modelo

El conjunto de datos elaborado constituye la materia prima que nos permite obtener modelos que automaticen dos tareas fundamentales relacionadas con los ripios de perforación, tanto su descripción textual como su medición. En el primer caso, el especialista humano realiza una inspección visual de los ripios de perforación con el fin de identificar sus características principales, las cuales serán conectadas de manera sintáctica y semántica en una explicación de tipo textual y verbal. Nuestro objetivo es trasladar dicha tarea al ámbito computacional, específicamente bajo el enfoque de la visión artificial y el procesamiento del lenguaje natural. Para tal fin, la solución propuesta combina una red neuronal convolucional (CNN) y una red Transformer.

Por otra parte en la medición manual de granos se usa un programa que permite dibujar manualmente la longitud a ser medida del fragmento, haciendo dos clicks para marcar el principio y fin de la línea, luego el programa mide la longitud de la línea que se dibujó. Para ello el técnico debe invertir tiempo en identificar la longitud más representativa del fragmento. El código empleado en nuestro trabajo suplir la medición se encuentra conformado tanto por el paquete colab-zirc-dims y la librería Detectron2.

En esta sección, presentamos una ilustración gráfica que representa la arquitectura del sistema, además de describir detalladamente sus componentes y la interacción entre ellos. También explicamos el funcionamiento de ambas soluciones propuestas.

3.2.1. Redes CNN y Transformer

El modelo de descripción automática combina una red neuronal convolucional para el procesamiento de las imágenes y una de tipo transformer para la generación de la descripción textual (Figura 3). Tomamos como base el trabajo de (Nain, 2021), el cual forma parte de los ejemplos disponibles en la página de Keras.

El diseño de Una CNN trata de imitar el funcionamiento del sistema visual humano y se encarga de extraer las características más relevantes de una imagen. Luego, estas características suelen emplearse para tareas de clasificación; sin embargo, en este estudio se utilizan para describir de manera textual el contenido de la imagen.

El modelo de transfer learning utilizado en la CNN, fue MobileNet, que se comprobó en el código de clasificación de Visión por Computadora en la sección anterior del documento. El mejor modelo de la etapa de Extracción de características (CNN) será empleado en la red transfomer para extraer las features de las fotografías.

En el presente estudio, utilizamos el código de Nain (2021) para elegir el modelo de CNN con mejor adaptación al procesamiento de las imágenes de ripios de perforación. Este código trabaja con un modelo conformado por la CNN MobilNet_V2 y una red neuronal regular que clasifica imágenes de uso de suelo en 14 clases; para su entrenamiento se utilizó el conjunto de más de 1000 imágenes *Imagenet*. Es así que, para la aplicación de este modelo en base a los requerimientos de nuestro estudio, con la ayuda de *transfer learning* se probaron tres estructuras de CNNs.

El *transfer learning* o Aprendizaje por Transferencia es un área dentro del Aprendizaje Automático, que se focaliza en la incorporación de conocimiento durante la resolución de un determinado problema y su posterior aplicación sobre otro diferente pero relacionado De Luca, Irigoitia, Pérez, y Pons (2021). Es decir, se aprovecha la estructura y los parámetros de una red neuronal preexistente en la resolución de una problemática similar. En este proyecto, se entrenó al modelo de Nain (2021) con 2200 imágenes de ripios de perforación agrupadas en 11 clases y con tres CNN diferentes: MobilNet_V2, EfficientNet_V2S y RestNet50_V2. La elección de los modelos mencionados se realizó en función de sus características. Los tres modelos son actuales, modernos y muy populares, sin embargo, cada uno destaca por si mismo. MobilNet por un lado, se caracteriza por ser ligero y con excelentes resultados en aplicaciones de Google. EfficientNet destaca por su rapidez y eficiencia en el entrenamiento del modelo, y ResNet es la CNN con más capas en su estructura.

Por su parte, la red transformer se encuentra conformada por un codificador y decodificador. Ambos componentes se basan en un mecanismo de atención para que el modelo pueda generar la salida enfocándose en las partes más relevantes de la entrada. El codificador toma las características de la imagen como entrada y las relaciona entre sí para crear representaciones más elaboradas, las cuales se traducen en vectores que capturan el contexto visual. El decodificador examina cómo cada palabra de la descripción se relaciona con las demás utilizando un mecanismo de autoatención con máscara para considerar solo las palabras anteriores y que omite las posteriores a la palabra actual. De esta forma se genera un vector específico que captura el contexto particular de cada palabra. Este vector de contexto se combina con la salida del codificador mediante un proceso de atención cruzada, lo que permite predecir la siguiente palabra de manera secuencial. Este proceso se repite iterativamente para generar progresivamente cada palabra de la secuencia textual completa. Previamente, cada palabra debe pasar por un proceso de *embedding*, el cual convierte las palabras a números. Durante el embedding se reconoce y realiza un recuento del vocabulario total usado en las descripciones.

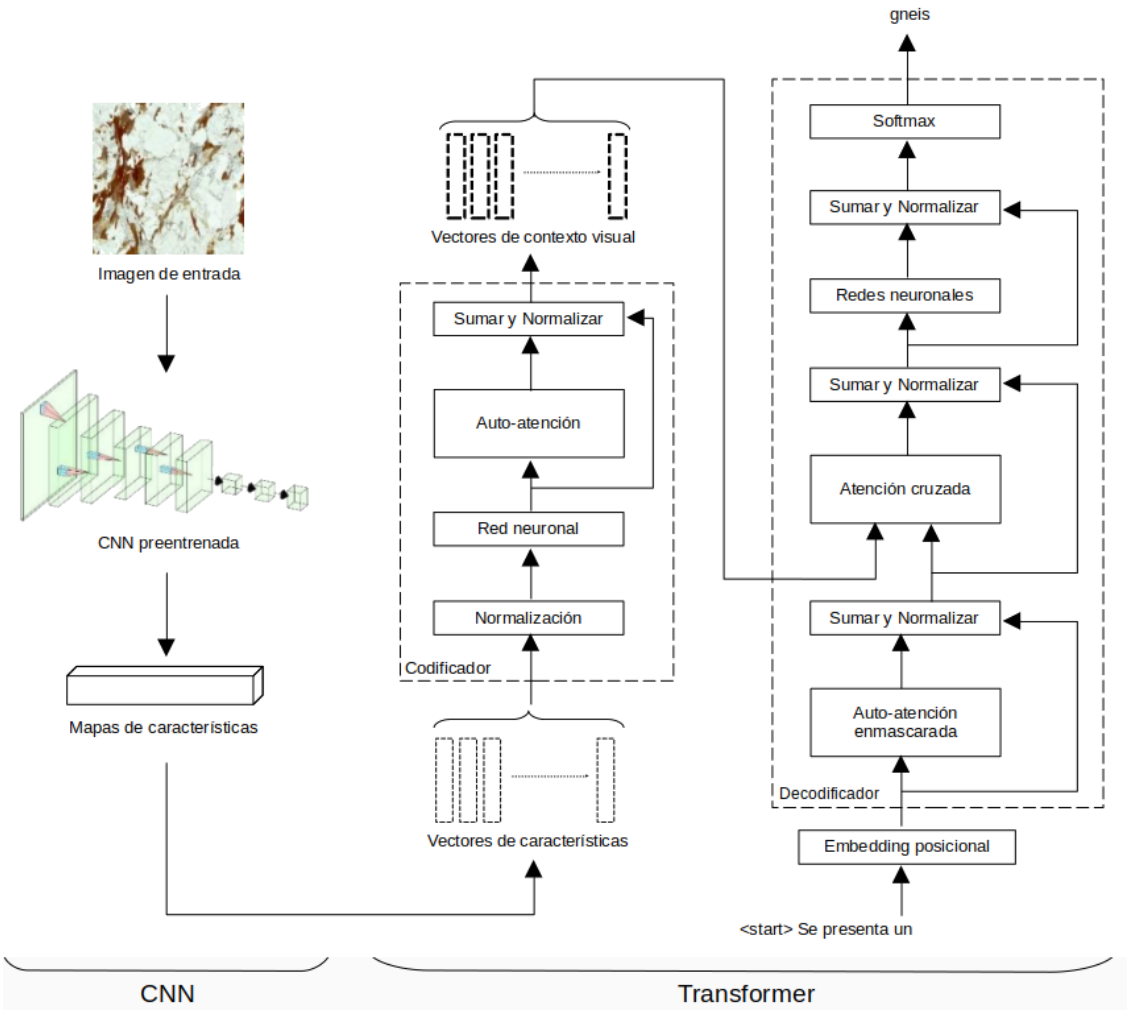

Figura 3: Arquitectura del modelo de descripción textual.

Luego, cada palabra se convierte en una unidad básica de trabajo o *token*, pero debido a que los caracteres no representan información válida para un ordenador se convierten en números. Cada token pasa a ser un vector de tamaño fijo cuyo valor es de 512. Se convierte en esta cifra exacta por ser un parámetro pre-establecido de la arquitectura Transformer. Durante la tokenización, cada palabra se vuelve una unidad básica de trabajo. Esta independencia ocasiona que se pierda la secuencialidad que formaba las frases. Para evitar esta situación, se agrega información posicional a cada uno de los tokens para preservar el orden entre ellos.

3.2.2. Segmentación y medición de clastos

Además, se utiliza un modelo de deep learning que dibuja los bordes de los granos, proceso conocido como segmentación, el cual escoge el mejor y mide los ejes mayor y menor de uno de los fragmentos identificados para que el usuario tenga una referencia del tamaño de estos. Como dato extra, en la medición también se agrega una tabla con parámetros adicionales de ese fragmento como área, perímetro, entre otros.

Para la medición de fragmentos de ripios de perforación, aprovechamos la implementación de (Sitar y Leary, 2023), misma que se adaptó a los requerimientos del presente estudio. Este código originalmente era utilizado para la medición de imágenes con fragmentos de circones. Estos minerales son de interés porque se pueden realizar dataciones cronológicas, pero también porque con su tamaño se puede comprender el transporte de granos y datos para estudios de provenanza.

Sin embargo, su implementación con imágenes de ripios de perforación ha dejado muy buenos resultados. Para lograr una adecuada medición de ripios de perforación, los autores originales crearon su propio paquete colab_zirc_dims

(https://github.com/MCSitar/colab_zirc_dims/tree/main/colab_zirc_dims) escrito en Python 3.8, es decir, un conjunto de librerías necesarias para soportar el código y que el programa pueda realizar las mediciones correspondientes. En este paquete se emplean las librerías Pillow para cargar imágenes y Matplotlib para crear, visualizar y guardar la segmentación de los granos de la imagen. Aquí una de las librerías más importantes es Detectron2, que fue implementada para la construcción y entrenamiento de los modelos de segmentación.

La arquitectura con la que esta propuesta trabaja es una Mask-RCNN,que es una red neuronal convolucional que trabaja en conjunto con otra red de soporte o *Backbone network* ((He, Zhang, Ren, y Sun, 2015)). El *Backbone network* es un módulo o parte de un modelo más grande que se encarga de identificar las características o features de la imagen, para generar un mapa de características que será entregado a otros módulos para generar el producto final. En este caso ResNet es el *Backbone* que genera el mapa de características con el que trabajará Mask R-CNN, para que con base en ellas, pueda proponer regiones que contengan objetos. Luego, módulos independientes generarán una máscara para cada objeto identificado ((He, Gkiozaru, Dollár, y Girshick, 2018)). Es importante mencionar que Mask-RCNN no es parte del Detectron2.

Mediante funciones de OpenCV (Bradski, 2000) y el módulo de medida Scikit (van der Walt y cols., 2014), el modelo puede dimensionar y calcular los siguientes parámetros para los fragmentos seleccionados: área, área convexa, excentricidad, diámetro equivalente, perímetro, longitud del eje mayor, longitud del eje menor, circularidad, diámetro rectangular del eje largo, longitud del eje corto, diámetro rectangular del eje, mejor longitud de eje largo y mejor longitud de eje corto.

Los parámetros calculados por el modelo son varios datos cuantitativos de cada fragmento, lo que resulta una ventaja en comparación con la metodología tradicional que se emplea para la descripción de ripios de perforación, donde los únicos datos cuantitativos que se obtienen corresponden a mediciones del eje más largo del fragmento. Con la presente metodología se obtienen datos cuantitativos en función del contorno del ripio, lo que permite adquirir mediciones del eje largo, eje corto, la circularidad, entre otros. Toda esta información puede ser utilizada para la caracterización de reservorios, tomando en cuenta que las únicas mediciones reales en ripios de perforación corresponden a las arenas; sin embargo, la circularidad de lutitas también puede ser utilizada para la identificación de zonas de derrumbe en los pozos petroleros.

Para la adaptación de la propuesta de (Sitar y Leary, 2023) en la medición de ripios de perforación se realizaron las siguientes actividades: (1) adaptación del código de medición al flujo establecido; en esta actividad se modificaron las líneas de código, de tal modo que el modelo de medición trabaje con la misma imagen de entrada que la etapa de descripción, además se agregó una lista de escalas para su selección y configuración del modelo. (2) ilustración de resultados en el flujo de trabajo de la sección automática y semi-automática; los resultados originalmente eran únicamente almacenados en la carpeta de Google Drive, motivo por el cual, se agregaron líneas de código para ilustrar estos directamente en el código.

Para lograr estos cambios, el flujo de trabajo fue el siguiente (Figura 4): Al iniciar, es necesario cargar las librerías con las que va a trabajar el modelo, seguido del paquete Detectron2 que se encargará de segmentar las imágenes dibujando el borde de los clastos. Seguidamente, de los dos parámetros de entrada que se ingresan, la fotografía no debe cargarse nuevamente porque el programa fue diseñado para trabajar con la misma que se usó para la predicción en el código transformer de la sección de Descripción textual.

Luego se carga el modelo de Deep Learning creado por los autores originales para identificar, segmentar y medir los clastos. Si bien una vez cargado está listo para usarse, se divide en dos componentes: automático y semi-automático. El componente automático dibuja los bordes de todos los clastos (segmenta) que identifica en la imagen y escoge aquel cuyos bordes hayan sido delineados con mayor precisión. En aquel fragmento se miden los ejes mayor y menor; a partir del cual se genera una tabla con los parámetros previamente mencionados. El componente semi-automático no escoge el fragmento a medir, por el contrario, requiere que el usuario dibuje manualmente los bordes del grano de interés, del mismo modo al objeto seleccionado se le medirán sus ejes mayor y menor y a partir de ellos dimensiona y calcula los mismos parámetros que en la tabla del componente automático.

4. Experimentación

4.1. Plataforma computacional

Optamos por utilizar la página web de Google Colab, que es un producto de (Research, 2024), que permite a todos los usuarios escribir y ejecutar código arbitrario de Python en el navegador. No tiene ningún costo y presenta la ventaja de que no es necesario descargar un programa específico ni presenta una interfaz complicada, al contrario, es simple.

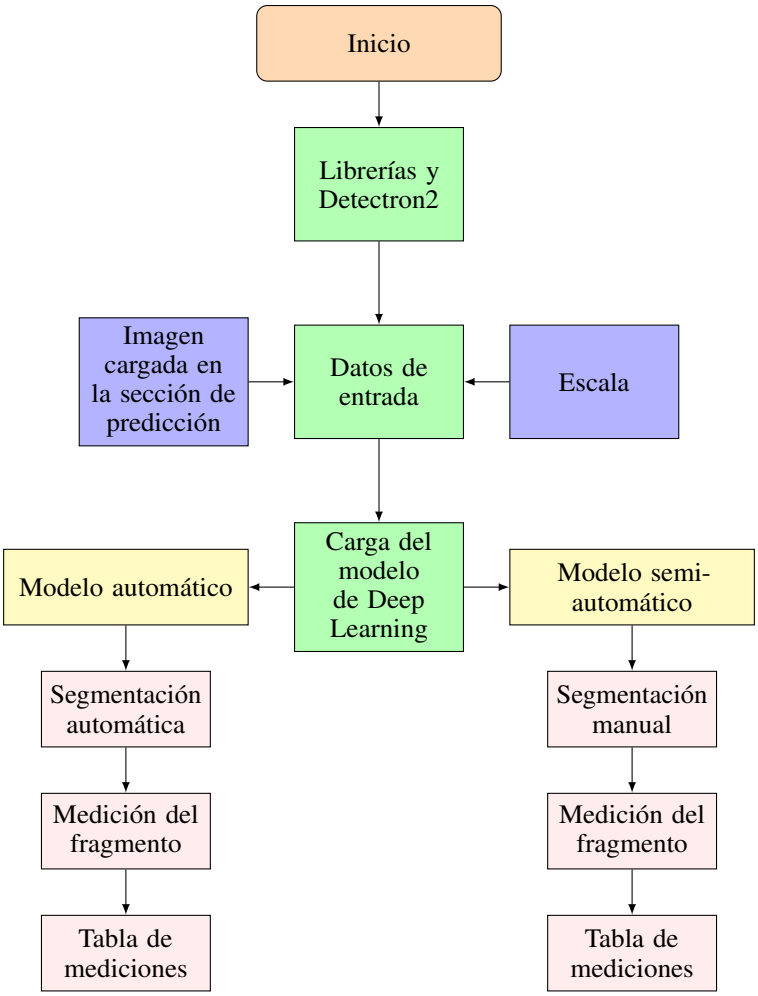

Figura 4: Flujograma para la medición y segmentación de ripios de perforación

Además, es ideal para aplicarlo en proyectos de aprendizaje automático, análisis de datos y educación. El primer caso es pertinente al objetivo de este trabajo. Para utilizar Colab sólo es necesario tener una cuenta de *gmail*. En esta cuenta se cargan la carpeta con el conjunto de imágenes y el archivo txt con las descripciones. Para ver cómo colocar las direcciones de estos archivos en el código, remitirse a la sección 4.5.3. del Notebook. De la misma forma, en este trabajo los outputs o resultados de las mediciones se guardarán en el drive de los usuarios.

Sin embargo, durante el desarrollo del proyecto se hizo evidente un hecho particular y es que mientras más se emplee una cuenta para el desarrollo de un código, más lento e inestable se vuelve la conexión del entorno en Colab. Tal y como se aprecia en el cuadro 2, las características del ordenador son buenas y permitirían un rápido flujo de trabajo, del mismo modo en el cuadro 3 el entorno T4 posee un RAM y disco suficientes como para realizar el entrenamiento en cortos periodos de tiempo. A pesar de ello, conforme se aumentaba el tamaño del conjunto de datos y corría el código más frecuentemente, el entorno era más lento y se desconectaba con mayor frecuencia.

Si bien es es una gran oportunidad gratuita que apoya a aquellos cuyos ordenadores no se encuentran a la par de realizar una programación compleja, es necesario aceptar que también cuenta con fallos.

Características técnicas de la computadora

4.1.1. Elección de la red neuronal convolucional (CNN)

Como se mencionó anteriormente, el modelo implementado consta de una red neuronal convolucional y una red neuronal regular de clasificación (Figura 1). Las CNNs poseen capas de convolución y reducción que extraen las características de las imágenes, estas pasan a las redes neuronales regulares que las clasifican en función de clases predeterminadas. De

Cuadro 2: Características técnicas del hardware.

Características principales	
Procesador	Intel Core i7
RAM	8 GB
Disco duro	1845 GB
Tarjeta gráfica	NVIDIA GeForce MX1030
Velocidad	1,99 GHz
Sistema operativo	64 bits

Cuadro 3: Características del entorno T4 GPU de Google Colab.

Características principales	
RAM	1.30 GB/ 12.67 GB
Disco	26.31 GB/ 78.19 GB

la misma forma funcionan los modelos MobilNet, EfficienNet y ResNet, donde las capas de convolución y reducción reciben el nombre de cabecera y las redes regulares ápice. Lo que realizamos en este estudio, con la implementación de *transfer learning*, es aprovechar la cabecera de estos tres modelos, y utilizarlas para la clasificación de imágenes de ripios con nuestra propia red regular o ápice. De esta forma, en la etapa de descripción automática, aprovecharemos la cabecera del modelo con mejores resultados de entrenamiento, pero en lugar de llevar las características a redes regulares densas, se las llevará a redes neuronales transformer (TNN).

En esta etapa se utilizó las CNNs de los modelos MobileNet, Efficient y ResNet, sumado a una red neuronal regular de clasificación, que predice la clase a la que pertenece una imagen de ripios de perforación. La red neuronal regular se compone de: (1) una capa de entrada con 256 neuronas y una función de activación relu, (2) una capa oculta con 512 neuronas con función de activación relu, y (3) una capa de salida con 18 nueronas y una función de activación softmax (Figura 5). Adicionalmente y exceptuando la capa de salida, tanto la capa de entrada como la capa oculta están configuradas con dropout para la optimización de su rendimiento.

El flujo de trabajo del código de Nain (2021) se ilustra en el flujograma 1, donde la ejecución inicia con el ingreso de las 11 clases de imágenes, adaptadas a los requerimientos del código a través de su procesamiento. Posterior a esto, el conjunto de datos se divide en subconjuntos de entrenamiento y validación. Además, se configura los parámetros del modelo conformado por una CNN y una red neuronal regular de clasificación; en este paso se eligió la CNN con mejores resultados en nuestros archivos de imagen. Para después, proceder al entranemiento del modelo, cuyo desempeño es evaluado según valores de precisión y perdida, así como también con una matriz de confusión, que de dar resultados positivos se procede a la predicción. De esta manera, obtenemos un clasificador de imágenes de ripios de perforación en función de sus litologías. Las subetapas de esta sección se especifican con más detalle en los siguientes apartados.

4.1.2. Entrenamiento y matriz de confusión

La etapa de entrenamiento se llevó a cabo mediante un compilador de descenso estocástico Adam, configurado con 15 épocas y una taza de entrenamiento de 0.001. En esta etapa, los valores de pérdida y precisión serán fundamentales para la elección del tipo de red neuronal que mejor se adapte al objetivo del proyecto. Es recomendable tener relaciones cercanas tanto de entrenamiento y validación, hacia valores altos y bajos de precisión y pérdida respectivamente. En las siguientes gráficas se presenta las métricas de los modelos implementados en este proyecto. El modelo de CNN con mejor métrica fue EfficientNet, obteniendo valores del 90 % en precisión y del 36 % en pérdida. De igual manera, los valores de evaluación arrojan buenos resultados en ambos casos.

La precisión del modelo, como la mayoría de los problemas de clasificación, se evalúa con una matriz de confusión, donde se esquematiza los valores reales vs los valores de predicción, denotando los errores y aciertos de las predicciones realizadas por el modelo de red convolucional (CNN). En esta etapa, lo óptimo es tener una matriz de confusión con el mayor número de verdaderos positivos y verdaderos negativos; no obstante, en la mayoría de los casos, valores de falsos positivos y falsos negativos concentran un porcentaje importante dentro de los resultados de una matriz de confusión.

4.2. Descripcion textual (Transformer)

La sección de Descripción textual se centra en desarrollar y correlacionar las descripciones textuales con la imagen correspondiente, con ello se logra entrenar al modelo para generar descripciones de imágenes externas. Por ello se siguen los pasos representados en el flujograma de la Figura 5 Para ello será necesario preparar una base de datos que mantenga un orden y formato establecido, al igual que usar la misma codificación que para las imágenes. Una vez que se hayan realizado las correcciones correspondientes al dataset y se encuentre en su forma final, se pasa a la siguiente fase que es emplear un modelo de reconocimiento adecuado. El modelo utiliza la CNN con mejor rendimiento que se definió en la sección de Extracción de características. Luego emplea las redes transformer para correlacionar entre sí los datos de entrada (features de imágenes y descripciones), gracias a lo cual se entrenará el modelo y se podrán generar descripciones de imágenes externas. La métrica de control que mide la calidad de las descripciones es BLEU.

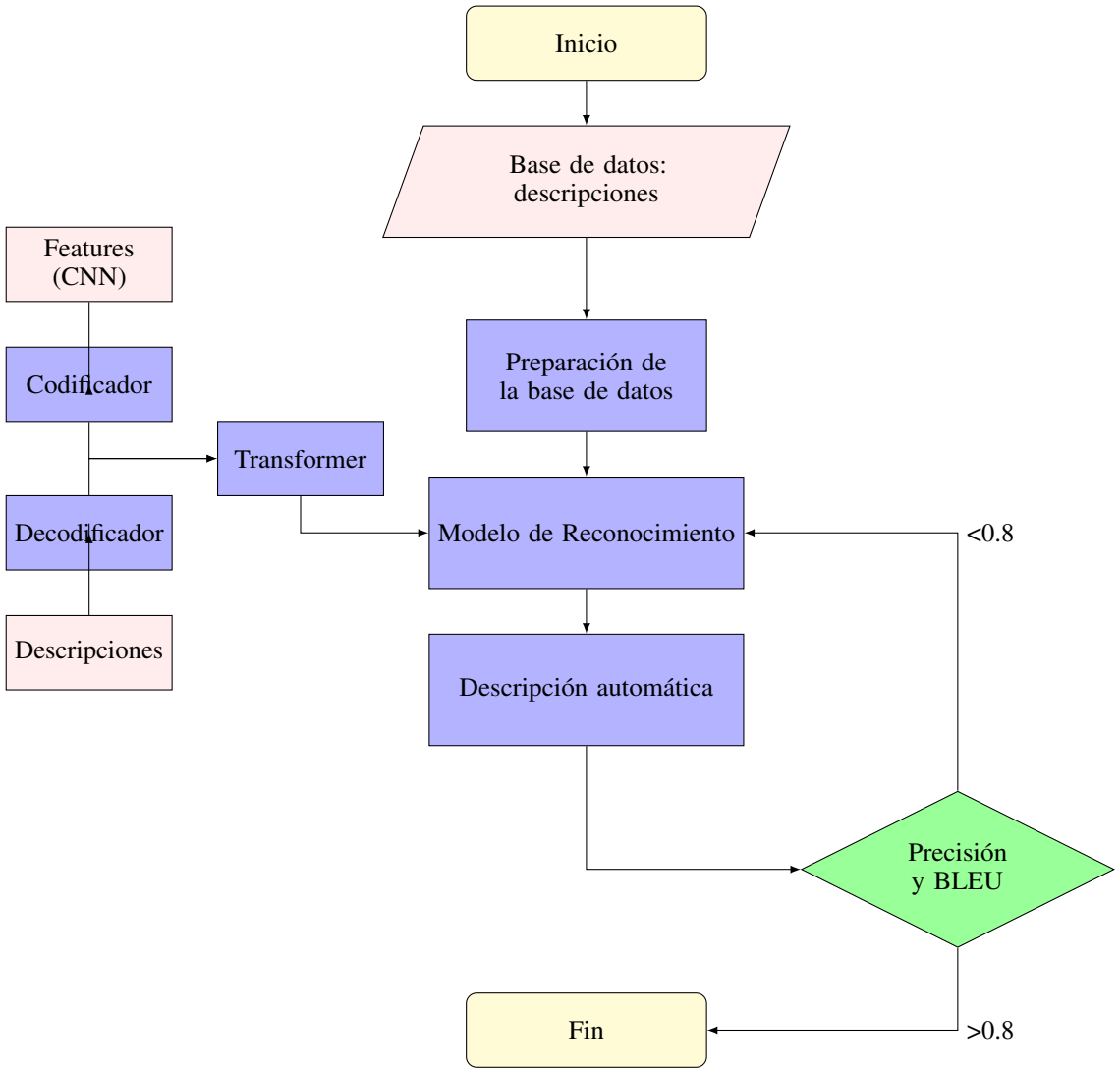

Figura 5: Flujograma para el reconocimiento y descripción automática de imágenes de ripios de perforación

4.2.1. Preparación del conjunto de datos

Para mantener la coherencia dentro del código de Nain (2021), que es aquel utilizado para el desarrollo de este proyecto, es necesario verificar que el código de la descripción textual y de la imagen sean los mismos. Las descripciones se encuentran en un archivo de bloc de notas con formato txt.

El código pre-existente fue ligeramente modificado para que el programa reconoza los signos de puntuación: coma (,), punto (.) y adicionalmente el simbolo de porcentaje (%), de esta forma se asegura que posteriormente el paso de texto a voz sea suave y suene natural.

Durante el entrenamiento, el modelo presentó un problema recurrente en las descripciones: espacios adicionales al final de cada descripción. En el código, el modelo no está programado para identificar estos espacios adicionales que surgen a raíz de errores humanos. La solución fue corregirlos manualmente eliminando cualquier espacio inecesario después del punto final.

Una vez que la base de datos se encuentra en óptimas condiciones, puede ser usado por el decodificador de la red Transformer.

el texto atraviesa un proceso de tokenización y vectorización. La tokenización identifica a cada palabra como un token o unidad básica de trabajo para el modelo, luego esta unidad pasará por la vectorización donde cada token pasa a ser un vector y la palabra ya no se representa a través de caracteres, sino números. Esta información numérica es la que ingresa en el modelo de reconocimiento.

4.2.2. BLEU

Para comprobar la precisión de las descripciones con respecto a la validación, se empleó BLEU, que son las siglas de Bilingual Evaluation Understudy Score. Originalmente BLEU es un parámetro diseñado para evaluar la calidad de las traducciones generadas con Machine Learning Brownlee (s.f.), comparando palabra por palabra (componentes) el grado de similitud entre un parámetro de referencia y un candidato. Este grado de similitud se mide con valores que abarcan de 0 a 1. Una similitud perfecta entre ambas da un resultado 1, mientras que si son completamente diferentes el programa arroja un resultado de 0.

En nuestro código antes de que el programa se considere apto para generar predicciones de imágenes externas, se realiza una evaluación en la que el programa escoge una imagen al azar del dataset para ser clasificada y descrita (Figura 6). Para ello se debe ingresar manualmente el codigo de la imagen escogida en la sección de referencia (Figura 7). El programa busca el código de la foto dentro del dataset y extrae su descripción, ese será el parámetro de referencia. Para que busque este parámetro se uso un bucle con with en el que la función a cumplir es que en la descripción se encuentra presente el código. Si se cumple, se aplican lower(), para que todas las letras sean minúsculas y split(), que permite que cada palabra se vuelva un componente individual para ser calificado por BLEU.

La funcion lower() es importante porque BLEU considera incluso la falta de mayúscula en una palabra como un término completamente diferente y disminuiría el valor de la evaluación. Ejemplo: para el programa la palabra 'Caolín' no es igual a 'caolín'.

El candidato es la descripción que generó el programa. Se copia manualmente y se pega en la sección de candidato (Figura 8). Del mismo modo, para que la evaluación no disminuya por la presencia o ausencia de mayúsculas se emplea lower() y para que califique cada palabra como un componente individual se emplea split(). Una vez que se obtienen los dos parámetros, BLEU compara ambos palabra por palabra; cada término que difiera de la referencia disminuirá el valor de precisión sobre 1. Esto nos permitirá saber qué tan confiable son las descripciones que produce nuestro modelo.

Si el valor de BLEU es igual o mayor a 0.8 el modelo es apto para describir imágenes externas, caso contrario se considera que el trabajo del modelo no es confiable.

4.3. Conversión de texto a voz

Una vez obtenida la descripción textual, accedemos al servicio *gTTS* (Google Text-To-Speech), el cual convierte texto escrito en palabras habladas, por lo que la computadora también será capáz de realizar la descripción de la muestra verbalmente.

La conversión de archivos de texto a voz trabaja con una red y un vocodificador neuronal. En primera instancia, la red neuronal transforma una disposición de fonemas en espectrogramas, los cuales son representaciones visuales de frecuencias de sonido en diferentes niveles de energía. Posteriormente, el vocodificador recibe los espectrogramas de la capa de salida y los transforma a formas de onda de voz, las cuales se representan con gráficos bidimensionales continuos en tiempo e intensidad del sonido recibido, lo que le permite al vocoder seleccionar detalles minuciosos en el sonido procesado para reconocer los aspectos de la voz humana relacionados con textos escritos. Despues de todo este esntrenamiento, el resultado final es una voz realista que apenas se distingue de una voz humana real Goodwin (2022).

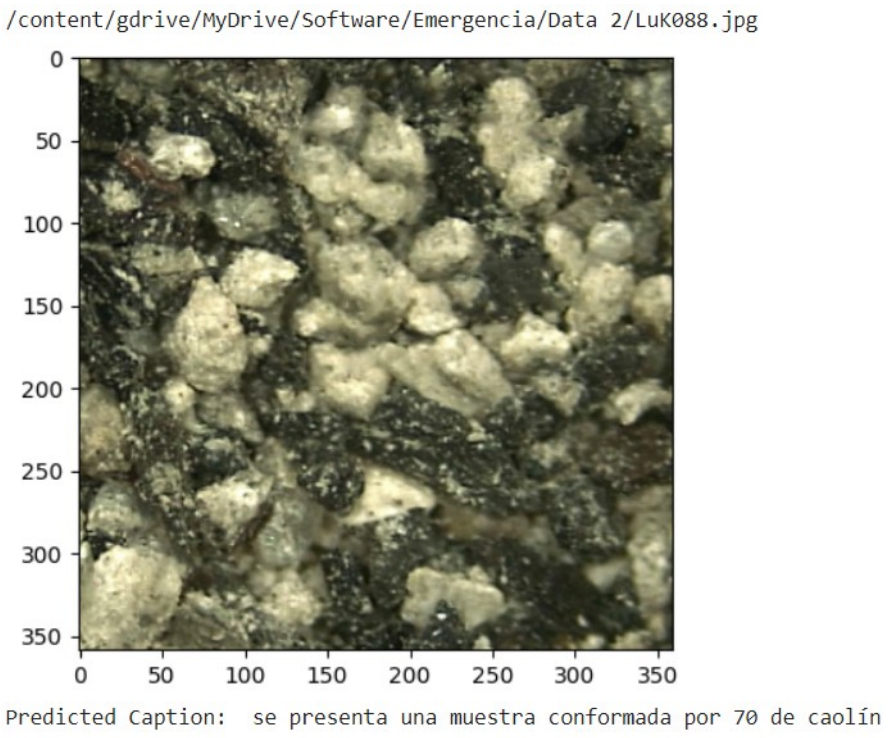

Figura 6: Imagen del conjunto de datos escogida por el código para verificar la calidad de las predicciones.

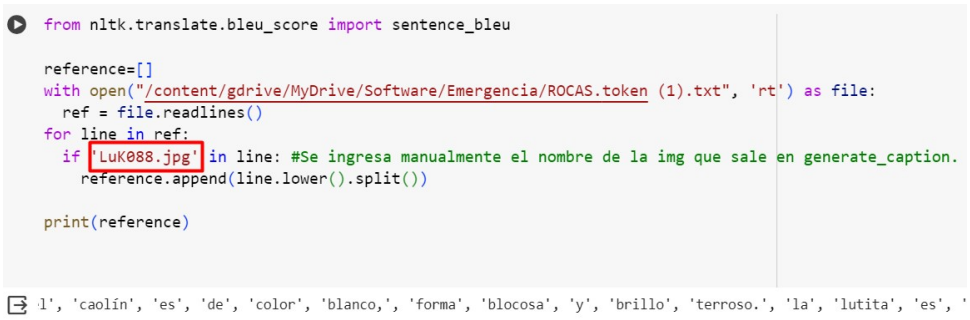

Figura 7: En el cuadrado rojo se ingresa el código de la imagen escogida para la verificación de predicciones.

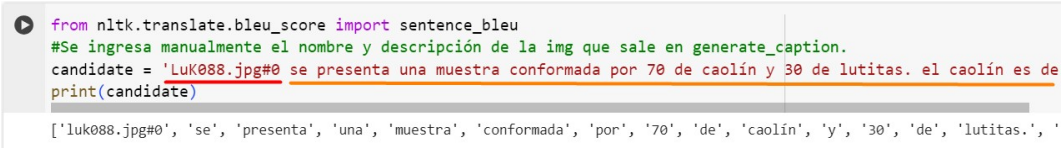

Figura 8: Descripción generada por el modelo. Su codificadión (rojo) y descripción (naranja) fueron copiados y pegados manualmente.

4.4. Notebook

Los frutos de este trabajo se ven plasmados en un notebook interactivo en la plataforma de Google Colab. Esta plataforma fue creada por la empresa Google y otorga al usuario la ventaja de crear códigos con el lenguaje python sin la necesidad de descargar un programa especializado. Estos códigos se encontrarán en un cuaderno o notebook. Para poder ingresar y usar el código que se generó en el presente trabajo es necesario tener un usuario de google.

Nuestro noteebook consta de tres partes:

4.4.1. Descripción y medición semi-automática de ripios de perforación mediante IA

En esta primera subdivisión se menciona a los autores de este trabajo y los códigos originales que fueron modificados para el desarrollo de este proyecto. Del mismo modo, también se enlista brevemente la estructura de las siguientes dos secciones.

El funcionamiento de las redes convolucionales y transformes, así como de los componentes automático y semiautomático ya fueron explicados en detalle en la sección de metodología, por lo que aquí únicamente se mencionarán aspectos del notebook que solo se encuentran en el codigo. También se detallan cuáles son los pocos datos que debe ingresar el usuario en cada sección.

4.4.2. Descripción textual y oral de imágenes de ripios de perforación

Al iniciar esta subdivision el primer paso es cargar las librerías. Continuando con el flujo de trabajo, para cargar las imágenes y las descripciones, el usuario debe colocar la ruta de la carpeta drive en la que se encuentran estos dos elementos. Para saber la dirección de la carpeta, es necesario digir el mouse al margen izquierdo de la pantalla y hacer click en el símbolo de una carpeta, se escoge la opción de 'gdrive' donde se desplegarán todos los componentes que el usuario tenga en su carpeta drive. Una vez que llegue hasta donde se encuentra el archivo, lo selecciona y en los tres puntos que activan se escoge la opción 'copiar ruta'. La dirección copiada deberá ser remplazada en IMAGES_PATH, para cargar la carperta con las fotografías y en text_data.

Seguidamente se corre el código hasta llegar a la sección de BLEU, donde se debe modificar manualmente la referencia y el candidato. EN la referencia se copia el código de la imagen que se generó en el paso anterior de 'verificación de las predicciones' y se remplaza en el if de la sección de referencia. Se corre la celda y el código extrae la descripción del conjunto de datos textuales.

En el candidato también se remplaza el código de la imagen, pero sin borrar la terminación '.jpg#0'. Seguidamente se copia y pega el ´predicted caption' de la sección anterior y se corre la celda. Seguidamente, se corre la celda que contiene el score y observa si el resultado obtenido es satisfactorio de acuerdo a lo explicado en la sección de BLEU. De ser así, se procede al siguiente y úlimo punto de esta subdivisión, cargar la fotografía que se desea describir en la sección de 'predicción con imágenes externas'.

Por último, el audio mp3 de las descripciones se genera automáticamente al correr las celdas que le siguen a la carga de imágenes externas.

4.4.3. Medición de imágenes de ripios de perforación

En esta última subdivisión se debe comenzar por cargar dos celdas: la primera corresponde al modelo Detectron2 creado por los autores originales, (Sitar y Leary, 2023), mientras que la siguiente son las librerías que contienen los comandos para el desarrollo de esta sección.

En la siguiente sección de 'Inspección de la imagen' el usuario ya no debe vovler a cargar la imagen, debido a que en el programa está diseñado para funcionar con la misma fotografía que fue cargada previamente en 'predicción con imágenes externas'. Lo que sí es necesario es que escoja la escala (µm/ pixel) a la que fue tomada la imagen. Para que exista certeza de que es la misma foto,la siguiente línea de código proyecta la fotografía con la que se va a trabajar.

A continuación y después de correr las líneas que cargan el modelo, en la 'evaluación de prueba' se visualiza cómo el modelo dibuja los bordes de los fragmentos para diferenciarlos entre sí (segmentación). De todos ellos solo escogerá a aquel cuyos bordes hayan sido dibujados con mayor exactitud y en la imagen continua muestra cuáles serán los ejes mayor y menor a ser medidos, con sus respectivos valores.

Es después de este punto que el usuario tiene la potestad de decidir emplear el componente automático, semiautomático o ambos.

En el componente automático el usuario no debe realiza ninún paso adicional además de correr las celdas de 'Procesamiento automatizado' e 'Ilustración de resultados automáticos' para visualizarlos. En 'Procesamiento automatizado' se observa un espacio para colocar la ruta de la carpeta en Google Drive, esto no debe ser modificado porque el código ha colocado una ruta general de drive aplicable para todos los usuarios, por lo que adicionalmente a los resultados que se plotean en el notebook, también se crea una carpeta en el drive con el nombre 'auto_zirc_processing_runfechadeejecución.

En la sección semiautomática el usuario debe encargarse de dibujar manualmente el fragmento de roca que desea que sea medido. Seguidamente 'Ilustración de resultados semiautomáticos' puede apreciar los resultados. Al igual que en la sección automática, no es necesario cambiar la dirección del drive porque automáticamente se creará la carpeta en la que se guarden los polígonos o dibujos de los fragmentos seleccionados y la tabla con las mediciones correspondientes.

4.5. Aplicación web

Finalmente, lo anterior se desplegará en una página Web amigable con el usuario en la que cualquier persona que desee emplearla no deberá hacer más que ingresar la imagen para que el programa arroje los resultados correspondientes. Todas estas etapas son explicadas en detalle a continuación.

La implementación de una App Web amigable con el usuario es de suma importancia para los objetivos del presente proyecto. Si bien un Nootebook de Google Colab permite aprovechar los modelos de Deep learning para los objetivos deseados, una App Web de interfaz simple facilita en sobremanera el rendimiento y ejecución de los mismos, sin necesidad de correr largas líneas de código, que en muchos de los casos necesita de unidades de procesamiento de vanguardia y tiempos considerables de espera. Por tales motivos, se ha tomado como prioridad el desarrollo de una App Web con el modelo ya entrenado de descripción y medición de ripios de perforación, de tal forma que cualquier usuario pueda utilizarla con imágenes propias de este, tanto en el ámbito académico como profesional.

La aplicación fue desarrollada con herramientas de libre acceso y de código abierto. Su diseño fue realizado con la ayuda de Google Colab y Hugging Face.

4.5.1. Desarrollo de la aplicación web (App Web)

Un entorno adecuado para aplicaciones web debe ser eficiente en cuanto al procesamiento y privacidad de los datos de ingreso del usuario. Actualmente, existen varios entornos en el mercado, tales como: Streamlit, Ambil, Gradio, Docker, Static, entre otros. Para efectos de este estudio, se utilizó Gradio en el diseño de la App Web, esto debido a la sencilla forma que este entorno presenta para generar interfaces amigables y de buena apariencia con modelos de aprendizaje profundo. Además, el hecho de que Gradio funcione como una librería de Python permite realizar todo el procedimiento en Google Colab, donde gracias a pocas líneas de código, se puede generar una App Web con tiempo de duración de 72 horas. Sin embargo, el objetivo de este estudio es generar una aplicación sin restricciones de tiempo, razón por la cual se utilizó hugginface para este cometido, obteniendo así una App Web que pueda ser utilizada en cualquier momento y por cualquier usuario.

El diseño de la App Web se realizó en dos partes, tal y como se subdivide el presente modelo. Una sección donde se describe automáticamente imágenes de ripios de perforación (Figura X) y otra donde se realiza la medición automática de fragmentos (Figura X). El usuario unicamente tendrá que ingresar la imagen con su escala en um_seg y en cuestión de pocos instantes tendra los resultados esperados por la aplicación. La descripción oral, así como las imágenes medidas y las tablas con los parámetros de medición podrán ser descargados por el usuario.

4.5.2. Descripción automática de imágenes

La interfaz de de la descripción automática de imágenes se ilustra en la Figura X. Tal interfaz se compone de recuadros y widgets donde se ingresaran datos y se ilustran resultados, por lo general los elementos de la página ubicados hacia la izquierda son los inputs y los elementos de la derecha son los outputs. En el presente caso, tal interfaz se compone de un recuadro como input; donde se debe ingresar la imagen a ser procesada, y dos recuadros como outputs; los cuales mostrarán la descripción textual y oral como resultados. Adicionalmente, el imput de la imagen ofrece las opciones de Cam Web y Paste from Clipboard, los cuales pueden ser utilizados para adjuntar las imágenes de ripios de perforación. A su vez, la descripción oral se genera en un archivo mp3, el cual puede ser descargado con la opción en la esquina del recuadro.

4.5.3. Medición automática de fragmentos

Para la medición automática de fragmentos de ripios de perforación se tiene la interfaz de la Figura X. Aquí se presentan dos imputs y dos outputs. Los inputs corresponden a la imagen a ser medida y a su factor de escala en um_seg. El factor de escala es primordial para la medición de fragmentos, ya que en función de este se realizarán las mediciones y cálculos correspondientes, de esta forma se presentan las escalas más recurrentes en imágenes de ripios de perforación según el aumento del microscopio con el que se adquirió la imagen. Por otro lado, en la parte izquierda se ubican dos recuadros donde se ilustrarán los outputs del procedimiento, los cuales corresponden a una imagen con el fragmento ya medido y a una tabla con los valores numéricos medidos y calculados. De la misma forma que en la sección anterior, tanto la imagen con el fragmento ya medido como la tabla con los parámetros correspondientes podrán ser descargados.

5. Resultados

5.1. Extracción de características (CNN)

La ejecución del modelo con tres CNNs dió como resultado las métricas presentadas en el cuadro 4. El mejor desempeño se lo obtiene mediante la implementación de Mobilnet_V2 con un valor de precisión del 99.8 %, superando tan solo en un 0.3 % a RestNet50_V2 y dejando a EfficientNet_V2S como la CNN con menor adaptación al conjunto de imágenes. Las gráficas de presición y pérdida resaltan un equilibrio en el ejuste de los modelos que utilizan Mobilnet_V2 y RestNet50_V2, que al llegar a la época 19 adquieren valores constantes hasta el final del entrenamiento. Por otro lado, las gráficas de la ejecución con EfficientNet_V2S denotan un pobre desempeño y un mal ajuste del modelo durante su entrenamiento (Figura 9). Lo mencionado anteriormente también se refleja en la evaluación de cada modelo mediente una matriz de confusión, donde Mobilnet_V2 presenta el mayor número de verdaderos positivos, seguido por RestNet50_V2 y muy por debajo EfficientNet_V2S. De esta forma, la CNN seleccionada para la siguiente etapa del proyecto fue Mobilnet_V2 (Figura 10).

Cuadro 4: Precisión de las CNNs implementadas en el modelo.

CNNs	Precisión
MobilNet_V2	99.8 %
EfficientNet_V2S	44.3 %
RestNet50_V2	99.5 %

5.2. Descripción de imágenes

5.2.1. Precisión

El entrenamiento de la red transformer se realizó con 2200 imágenes descritas y una iteración de 30 épocas, dando como resultado 0.948 /1 de precisión. Esto puede verificarse gracias a las siguientes gráficas en donde la Figura 11 muestra que la distancia entre las curvas de entrenamiento (azul) y validación (rojo) es mínima, esto significa que el desempeño del modelo fue tan bueno en el entrenamiento como en la prueba y al estar cerca (0.948) del máximo valor de precisión (1) se otorga un alto grado de confiabilidad.

Del mismo modo, en la Figura 12 se observa que en ambas instancias las curvas se estabilizan en bajos valores de pérdida, por lo que la computadora cometía pocos errores al identificar litologías y relacionarlas con su respectiva descripción.

5.2.2. BLEU

En la 'Verificación de predicciones' la fotografía que escogió la computadora al azar fue LuK088.jpg, tal y como puede observarse en la Figura 13 y generó la siguiente predicción, que será colocada a continuación para que se aprecie adecuadamente. Textualmente la descripción del candidato dice: "Predicted Caption: se presenta una muestra conformada por 70 de caolín y 30 de lutitas. El caolín es de color blanco, forma blocosa y brillo terroso. La lutita es de color negro, forma planar y brillo graso."

Mientras que la descripción original era: "Luk088 jpg#0 se presenta una muestra conformada por 60 % de caolín y 40 % de lutitas el caolín es de color blanco forma blocosa y brillo terroso la lutita es de color negro forma planar y brillo graso."

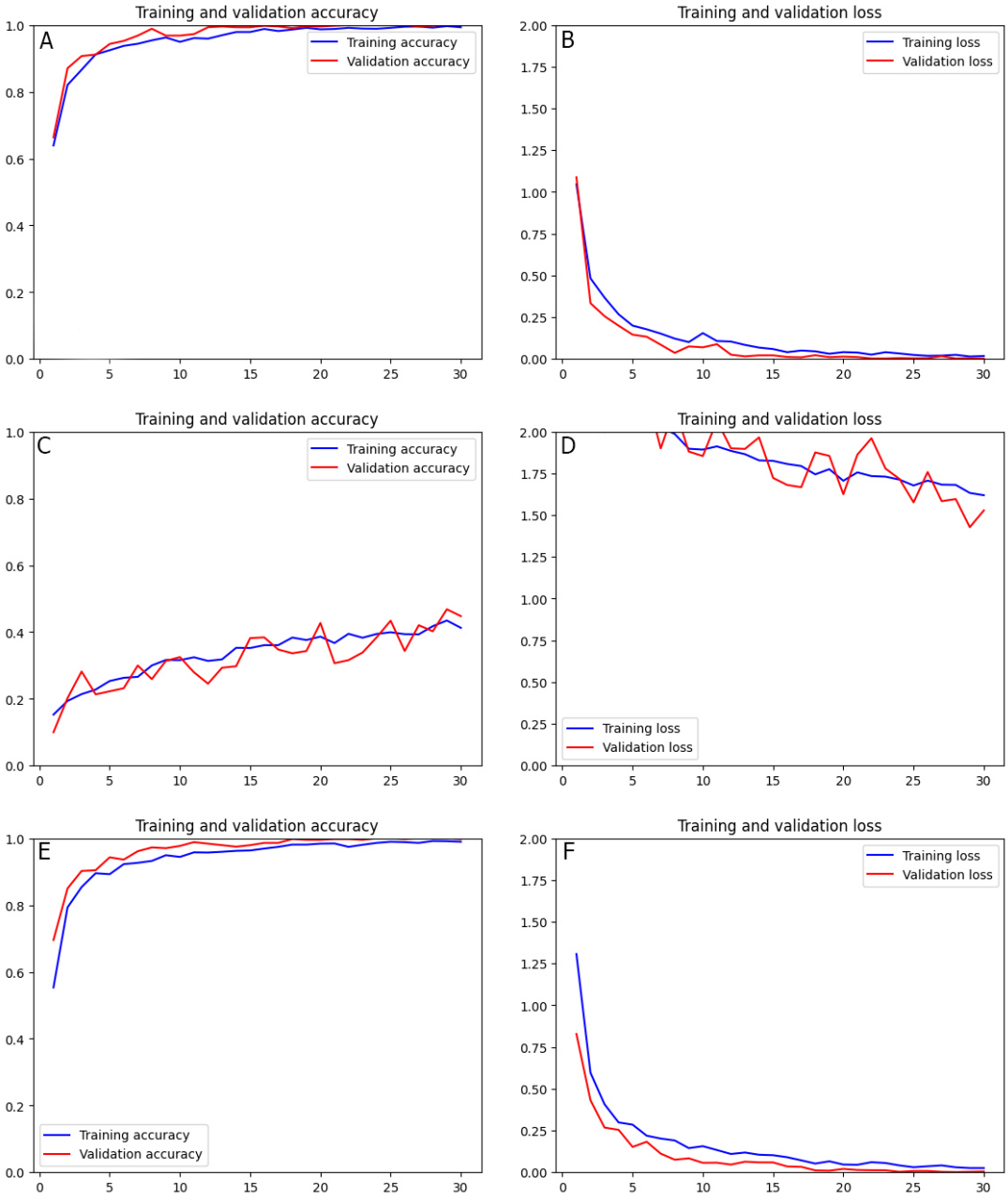

Figura 9: Precisión y pérdida de los modelos con Mobilnet_V2 (A y B), EfficientNet_V2S (C y D) y RestNet50_V2 (E y F).

Al comparar cada palabra se obtuvo una puntuación de 0.849 (Figura 14). Al pasar del umbral de 0.8 podemos asegurar con certeza que nuestro modelo de Red Neuronal Transformer es confiable.

5.2.3. Descripción textual

La imagen escogida para su predicción fue una lámina compuesta por limolita, arcillolita y arenisca. El programa pudo identificar adecuadamente las tres litologías y generó porcentajes coherentes; sin embargo, estos porcentajes cuentan con un pero y es que solo imprimen valores que existan en la base de datos. Este trabajo no llegó a generar un código capaz de calcular matemáticamente la porción de cada roca con respecto a un total. A pesar de ello, las predicciones que genera con un limitado número de valores son bastante rescatables.

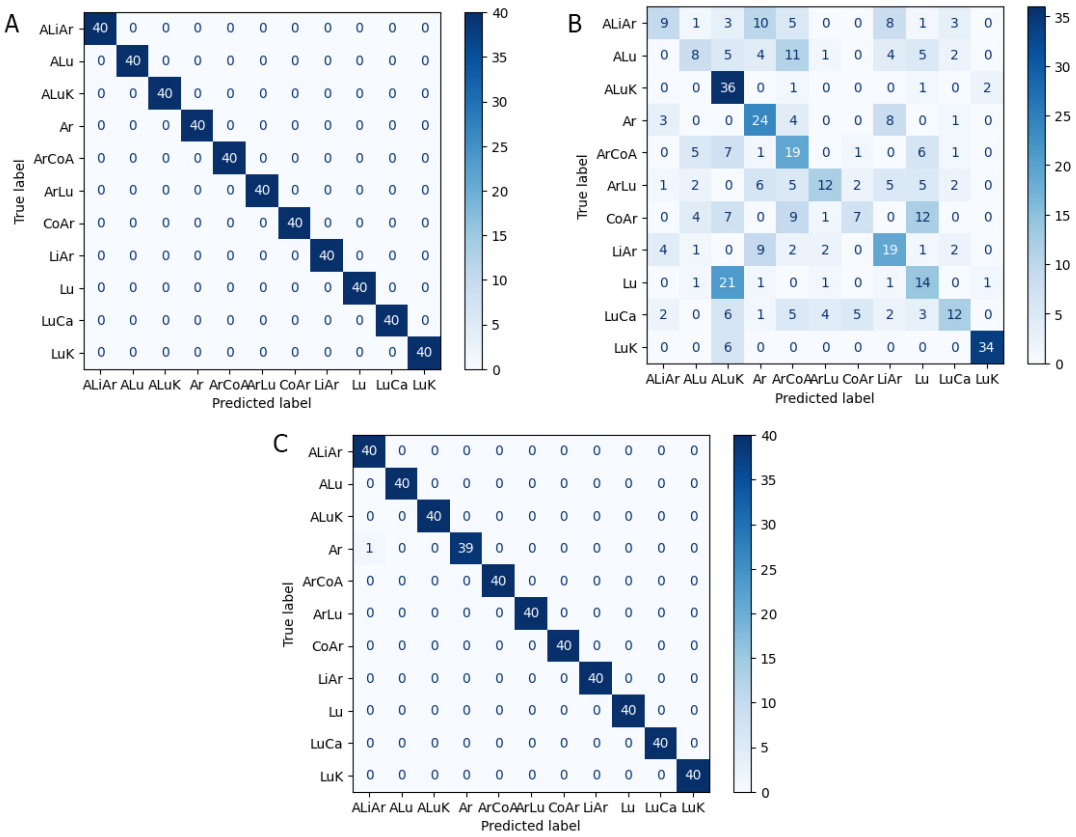

Figura 10: Matrices de confusión de los modelos con Mobilnet_V2 (A), EfficientNet_V2S (B) y RestNet50_V2 (C).

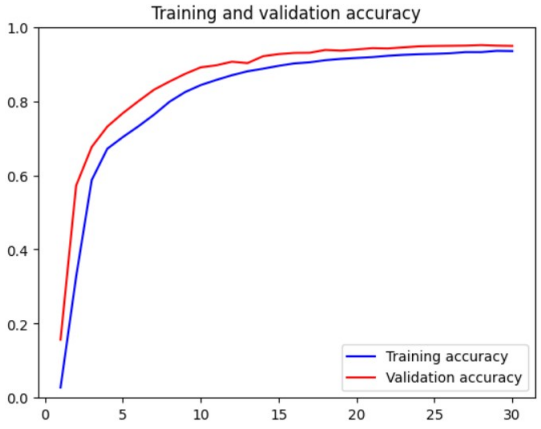

Figura 11: Precisión del entrenamiento vs precisión de la prueba de validación de la red neuronal Transformer.

5.2.4. Descripción oral

El archivo mp3 generado puede leer el texto de forma pausada y correcta gracias a los signos de puntuación y símbolo de porcentajes que reconoce el código. Si bien la voz femenina que se escucha no suena completamente natural, debido a que es una opción gratuita y no la pagada, posee una dicción clara y buena cadencia.

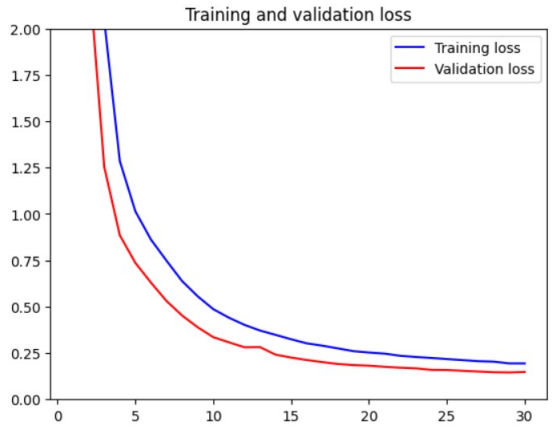

Figura 12: Pérdida del entrenamiento vs pérdida de la prueba de validación de la red neuronal Transformer.

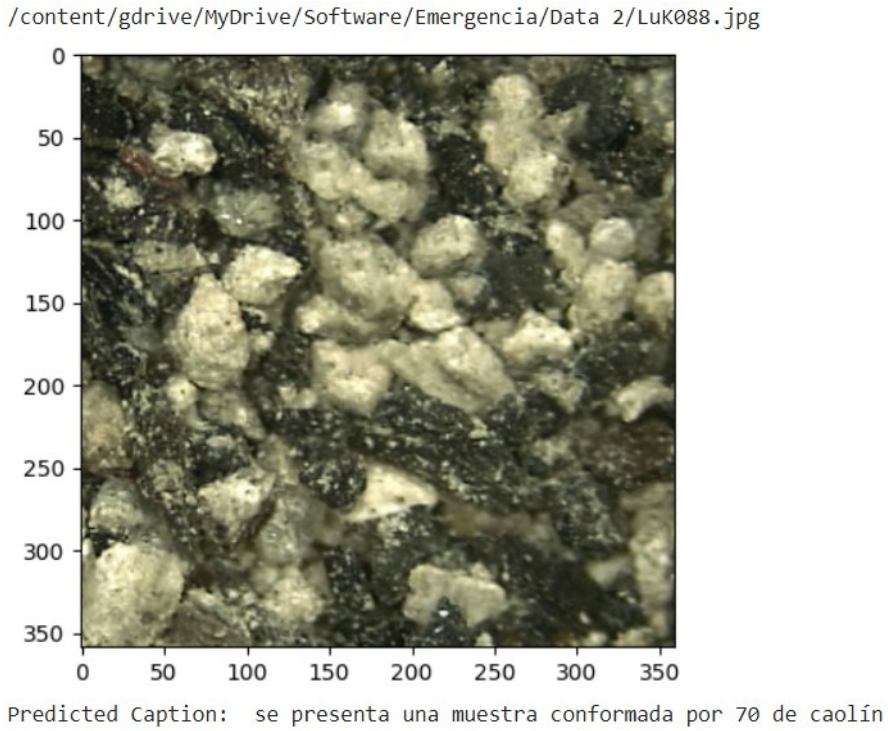

Figura 13: Imagen del conjunto de datos escogida por el código para verificar la calidad de las predicciones.

```
[ ] score=sentence_bleu(reference, candidate)
    print(score)

0.8499508493439808
```

Figura 14: Puntaje de la métrica BLEU para la evaluación de LuK088.jpg.

5.3. Medición de fragmentos

En la medición de fragmentos, los resultados serán ilustrados en la aplicación Web y en el Notebook de Google Colab. En ambos casos, basta con ejecutar la aplicación y el Notebook para poder visualizar la imagen escalada, con el fragmento medido y sus ejes mayor y menor ilustrados (Figura 15), así como también, una tabla con los parámetros calculados.

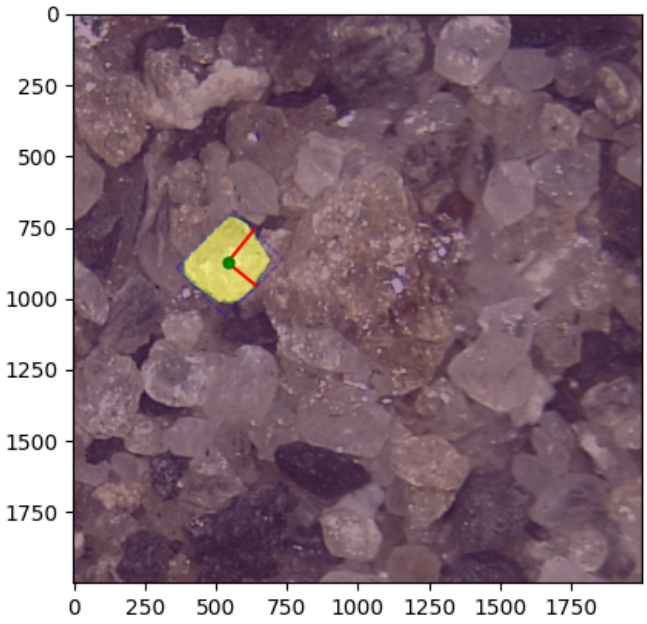

Figura 15: Medición de un fragmento de ripios de perforación.

A su vez, los resultados también serán almacenados en el Google Drive del usuario si así lo desea, donde automaticamente se creará una carpeta según la metodología de medición que se haya realizado, es decir, si se utilizó la metodología automática o semiautomática. La Figura X ilustra la estructura de almacenamiento tanto de la imagen medida en formato .png como de la tabla de parámetros en formato .csv.

6. Discusión

La obtención de los datos empleados en este trabajo no fue sencillo debido a que las empresas guardan ese tipo de información por cuestiones estratégicas, además de que una de las actividades a las más fuertes y demorosas es el recortar, codificar y describir las imágenes que conforman el conjunto de datos.

A diferencia del trabajo realizado por (Kathrada y Adillah, 2019), que buscaron identificar cuatro tipos de litologías en ripios, para lo que crearon su propia red convoluciona Bayesian optimised network con resultados satisfactorios y (Tamaazousti y cols., 2020), que trabajaron con la red CNN ResNet 300 imágenes de muestras tanto secas como húmedas para identificar 3 litologías con 95 % de precisión; nuestro trabajo consigue identificar 7 litologías y las subsecuentes 11 clases que conforman entre sí al mezclarse. Sin embargo, también es rescatable la investigación realizada por (Becerra1 y cols., 2022) en la que crearon una robusta base de datos de 16700 imágenes que trabajaron con ResNet-18 para clasificar el tipo de roca consiguiendo una precisión de 88-91 %. Si bien nuestro trabajo puede describir más litologías, no logramos desarrollar nuestro propio modelo para la identificación y descripción de ripios, como sí lo hicieron (Kathrada y Adillah, 2019). Por otro lado, (Paucar y cols., 2024) es el único autor mencionado en trabajos previos que toma la posta de identificar y describir 14 litologías en láminas delgadas, que es bastante similar a lo que se realizó en este trabajo; sin embargo, nuestra investigación cuenta con la diferencia de que adicionalmente utilizamos un código que permite la medición de los fragmentos de roca. Fuera de ello, también nos alimentamos de la

idea de (Becerra1 y cols., 2022) de que el código sea de libre acceso para motivar el aprendizaje y la perfectibilidad de un proyecto, por lo cual también tomaremos la misma postura con este trabajo.

Es así que las 3 redes convolucionales que fueron probadas para escoger el mejor modelo demuestran que no todos los códigos entrenados para trabajar con imágenes son aptos para abordar problemas específicos, como en el caso de la geología. Las litologías de los ripios de perforación poseen colores similares y la forma de los fragmentos no es característico de un solo tipo de roca, por lo que las redes no tienen la misma facilidad de diferenciar los objetos tal y como lo haría en otros casos. MobilNet_V2 prevalece sobre las otras dos redes porque es ligero y sobre todo, muy efectivo para la detección y segmentación de objetos ((Howard, 2018)). Es así que Mobilenet posee las precisiones más altas (99.8 %), los errores más bajos y una impecable matriz de confusión. Al implementar MobilNet_V2 en la red transformer se aseguró que el código contara con materia prima de calidad para realizar predicciones certeras.

Del mismo modo, la métrica BLEU destacó por ser un método eficiente para la verificación de predicciones, sin embargo tiene la ligera desventaja de que es muy estricta con los tokens o palabras y no admite pequeñas variaciones, como que una palabra de referencia cuente con una letra mayúscula (Roca) y la otra no (roca), tomándolas por dos términos diferentes. Esto baja el nivel de precisión y puede generar interpretaciones erróneas donde las predicciones del modelo se den por deficientes. Por otro lado, es necesario comprender que si bien la programación ha avanzado a pasos agigantados en estos últimos años, aún quedan aspectos que mejorar. Si se colocan líneas de código adicionales con medidas previas, tal y como se hizo en este trabajo al pasar todas las letras a minúscula, BLEU es una herramienta confiable para medir la precisión y descripciones textuales.

Con respecto a la medición de fragmentos, el programa creado por (M. Sitar y Leary, 2023) resultó útil a nuestro propósito en la medición de los mismos, pero cuenta con la desventaja de escoger únicamente el clasto que se encuentre mejor segmentado para su medición. Impidiendo que el programa mida automáticamente varios fragmentos a la vez. Pero la posibilidad del componente automático y semiautomático suple este tipo de inconvenientes, dándole al programa el atractivo de una aplicación interactiva. En el cuaderno original de Google Colab los resultados no se imprimían en la interfaz, sino que se podía visualizar únicamente en la dirección de drive que se ingresó en el sistema. Nosotros logramos hacer que el programa imprima en la interfaz tanto la imagen con los granos medidos y segmentados, como la tabla con los parámetros de medida.

Los resultados obtenidos pueden ser replicados en bases de datos similares, sin embargo es necesario considerar que se trabajó únicamente con las siete litologías enlistadas en las secciones anteriores, por lo que es muy probable que si se ingresa una foto con rocas que el programa no haya reconocido previamente puede generar predicciones erróneas. A pesar de ello, al volver este código en un archivo de acceso libre invitamos a otros investigadores a crear conjuntos de datos más robustos que les permitan describir una gran variedad de imágenes y emplearlo para el aprendizaje de esta rama.

7. Conclusiones y posibles trabajos a futuro

- Hemos expuesto el desarrollo de un sistema basado en IA que identifica litologías a través de la implementación del modelo de CNN MobileNet_V2, que es práctico para la detección y segmentación de objetos lo que le otorgó gran versatilidad en el trabajo con ripios de perforación generando altos valores de precisión (0.99). La experiencia nos ha demostrado que no todos los modelos entrenados con imágenes son aptos para un problema tan específico. Posteriormente la implementación de Redes Transformer creó descripciones textuales que pueden considerarse como válidas al comprobarlas con el parámetro de métrica Bleu donde alcanzan 0.84, lo que resulta un valor aceptable para cada predicción.

- La elaboración de un conjunto de datos robusto y de calidad es fundamental para el entrenamiento de un modelo, por lo que es importante cerciorarse de que la codificación sea la misma tanto para imágenes como para descripciones. La base de datos de este proyecto no es de libre acceso por su origen, pero se puede descargar imágenes similares de páginas web o de proyectos referentes a la temática.

- Se elaboró un Notebook interactivo que servirá como apoyo para cualquier usuario interesado en la temática. El Notebook de Google Colab es de libre acceso y fácil ejecución, de tal manera que puede ser utilizado tanto por cualquier usuario relacionado a la industria petrolera, como por estudiantes y personas en general que

deseen aprender sobre el tema.

Si bien los resultados obtenidos en este trabajo fueron satisfactorios, hubo ámbitos que escaparon del alcance del proyecto y se recomienda a los futuros investigadores ahondar en estas áreas para mejorar la IA en beneficio de la geología. Ámbitos como:

- La implementación de un filtro o 'guardián' que no permita al usuario ingresar otro tipo de imágenes que no sean ripios de perforación.
- Entrenar al modelo para diferenciar los fragmentos de roca de los aditivos que se agregan durante la perforación.
- Calcular matemáticamente los porcentajes de cada roca presentes en la imagen.

Del mismo modo, como autores estamos conscientes de que nuestro trabajo puede evolucionar y dar paso a mayores mejoras dentro del campo de las ciencias de la tierra, por lo que el código desarrollado en esta investigación será de acceso libre al público. Sin embargo, no liberaremos las imágenes debido a acuerdos de confidencialidad con la empresa que nos entregó muy amablemente las fotografías para desarrollar este tema.

Referencias

Becerra1, D., de Lima, R. P., Galvis-Portilla, H., y Clarkson, C. R. (2022). Generating a labeled data set to train machine learning algorithms for lithologic classification of drill cuttings. *SEG*, 1-11.

Bradski, G. (2000). The opencv library. *Dr. Dobb's Journal of Software Tools*, 120-125.

Brownlee, J. (s.f.). *A gentle introduction to calculating the bleu score for text in python.* Descargado de `https://machinelearningmastery.com/calculate-bleu-score-for-text-python/`

De Luca, A. M., Irigoitia, M. E., Pérez, G. A., y Pons, C. F. (2021). so de la técnica de transfer learning en machine learning para la clasificación de productos en el banco alimentario de la plata. *In IX Congreso Nacional de Ingeniería Informática/Sistemas de Información*.

Goodwin, O. (2022). *What is neural text to speech?* Descargado de `https://synthesys.io/blog/what-is-neural-text-to-speech/`

He, K., Gkiozaru, G., Dollár, P., y Girshick, R. (2018). Mask r-cnn. *arXiv*. doi: https://doi.org/10.48550/arXiv.1703.06870

He, K., Zhang, X., Ren, S., y Sun, J. (2015). Deep residual learning for image recognition. *arXiv*. doi: https://doi.org/10.48550/arXiv.1512.03385

Howard, M. S. A. (2018). *Mobilenetv2: The next generation of on-device computer vision networks.* Descargado de `https://blog.research.google/2018/04/mobilenetv2-next-generation-of-on.html#:~:text=MobileNetV2%20is%20a%20very%20effective,Object%20Detection%20API%20%5B4%5D.`

Ismailova, L., Dochnika, V., al Ibrahim, M., y Mezghani, M. (s.f.). *Automated drill cuttings size estimation.* Descargado de `https://www.sciencedirect.com/science/article/abs/pii/S0920410521014911`

Kathrada, M., y Adillah, B. (2019). Visual recognition of drill cuttings lithologies using convolutional neuralnetworks to aid reservoir characterisation. *SPE Reservoir Characterisation and Simulation Conference.*, 1-11.

Nain, A. (2021). *Image captioning.* (Code examples-Computer Vision. [Source code]. Availability: https://keras.io/examples/vision/image_captioning/)

Paucar, S., Mejía-Escobar, C., y Collaguazo, V. (2024). *Descripción automática de secciones delgadas de rocas: Una aplicación web.* Descargado de `https://www.researchgate.net/publication/378491699_Descripcion_Automatica_de_Secciones_Delgadas_de_Rocas_Una_Aplicacion_Web`

Research, G. (2024). *¿qué es colaboratory?* Descargado de `https://research.google.com/colaboratory/faq.html?authuser=3&hl=es`

Sitar, y Leary. (2023). Technical note: colab_zirc_dims: a google colab-compatible toolset for automated and semi-automated measurement of mineral grains in laser ablation–inductively coupled plasma–mass spectrometry images using deep learning models. *European Geosciences Union*, *4*, 109-126.

Sitar, M., y Leary, R. (2023). Technical note: colab_zirc_dims: a google colab-compatible toolset for automated and semi-automated measurement of mineral grains in laser ablation–inductively coupled plasma–mass spectrometry images using deep learning models . *Geochronology*, 109-126.

Tamaazousti, Y., François, M., y François, J. (2020). Automated identification and quantification of rock types from drill cuttings. *Journal of Petroleum Science and Engineering*.

Tolstaya, Shakirov, Mezghani, y Safonov. (2023). Fast reservoir characterization with ai-based lithology prediction using drill cuttings images and noisy labels. *Journal of Imaging*, *9*, 126.

van der Walt, Schönberger, Nunez-Iglesias, Boulogne, Warner, Yager, ... Yu (2014). scikit-image: Image processing in python. *PeerJ*. doi: https://doi.org/10.7717/peerj.453

Wang, Yang, Zhao, y Wang. (2018). Lithology identification using an optimized knn clustering method based on entropy-weighed cosine distance in mesozoic strata of gaoqing field, jiyang depression. *Journal of Petroleum Science and Engineering*, 157 - 174.