LOS LIBERTADORES
FUNDACIÓN UNIVERSITARIA

Uso de Deep Learning como herramienta para encontrar las zonas que están siendo explotadas por hidrocarburos en el sistema nacional de áreas protegidas.

Use of Deep Learning as a tool to find the areas that are being exploited for hydrocarbons in the national system of protected áreas.

Director: Manuel Francisco Romero Ospina - John González Veloza
Autor: Andrea Patricia Acosta Ovalle
apacostao@libertadores.edu.co
Estudiante de la Especialización en Estadística Aplicada (Virtual)
Facultad de Ingeniería y Ciencias Básicas

RESUMEN

La identificación geográfica de los sectores con distintas características permite visualizar la distribución espacial del territorio, por esta razón este proyecto busca reconocer los sectores explotados por la industria de hidrocarburos contenidos en el sistema nacional de áreas protegidas, para facilitar y reducir las actividades ejecutadas en campo con datos que permitan acotar el tiempo de elaboración de un diagnóstico jurídico catastral que se debe repetir en cada ocasión que requiera ingresar al predio de interés.

Palabras clave: Deep Learning, imágenes, hidrocarburos, áreas protegidas, modelos.

ABSTRACT

The geographical identification of the sectors with different characteristics allows visualizing the spatial distribution of the territory, for this reason this project seeks to recognize the sectors exploited by the hydrocarbon industry contained in the national system of protected areas, in order to facilitate and reduce the activities carried out in the field with data that allow shortening the time of elaboration of a cadastre legal diagnosis that must be repeated every time it is required to enter the property of interest.

Keywords: Deep Learning, images, hydrocarbons, protected areas, models.

JUSTIFICACIÓN

En gestión inmobiliaria para la adquisición y administración de derechos superficiarios, es necesario realizar un diagnóstico jurídico catastral con el fin de determinar las características del predio de interés, particularmente si ha sido afectado por las limitaciones que impone la declaratoria de un área protegida y determinar las actividades permitidas de conservación, investigación, educación, etc. Puesto que prima el interés general de conservación sobre los derechos particulares de propiedad, teniendo clara la facultad del Estado de poder declarar áreas de especial importancia ecológica con el fin de proteger y preservar el medio ambiente. (SINAP, 2020)

En Colombia la industria petrolera en los últimos años se ha posicionado como una de las actividades económicas más importantes del país, por lo tanto, para la celebración de nuevos contratos de hidrocarburos se debe realizar permanentemente la identificación de nuevas zonas con reservas de crudo, el continuo seguimiento a los contratos de explotación vigentes y la reactivación de bloques con áreas devueltas parcial o totalmente que pueden ser objeto de nuevas asignaciones (ANH, 2019); son labores necesarias para mantener activa esta industria.

En los departamentos de Antioquia, Santander, Norte de Santander, Arauca, Casanare, Meta, Huila, Vichada y Putumayo se concentra buena parte de la industria petrolera del país con la mayor producción de hidrocarburos; dichas entidades territoriales cuentan con áreas de protección ambiental que están siendo explotadas por varias industrias nacionales y procesos catastrales que cambian constantemente el inventario predial en sus aspectos jurídico, económico y físico; a partir de este conjunto de elementos nace el interés de realizar análisis que revelen el estado actual en estas zonas.

Existen gran variedad de alternativas como Deep Learning que utiliza redes neuronales artificiales para aprender de grandes cantidades de datos, en este caso encontrar zonas delimitadas para la extracción de hidrocarburos en imágenes que faciliten su exploración detallada y libres de nubes.

Es así como nace la idea de automatizar dichos procesos con nuevas tecnologías para facilitar la toma de decisiones, descartando zonas en conflictos de interés además que la dinámica de la información descrita obliga a revisar continuamente las posibles áreas de trabajo.

OBJETIVOS

Objetivo General

Identificar aquellas zonas contenidas en el sistema nacional de áreas protegidas que están siendo explotadas por hidrocarburos; de acuerdo con el marco jurídico no deberían ser objeto de explotación, pero es un fenómeno que se presenta en el país.

Objetivos Específicos

- Crear un dataset de imágenes con zonas destinadas para perforación de pozos petroleros y libres de nubes.

- Usar técnicas de data augmentation para tener una mayor diversidad de imágenes.

- Implementar modelos transfer Deep Learning ya definidos que permitan reconocer en imágenes aquellas zonas delimitadas para la extracción de hidrocarburos contenidas en áreas de protección ambiental.

- Analizar y evaluar los modelos implementados para detectar las zonas de interés.

REFERENTES TEORICOS

La Inteligencia Artificial nace por la necesidad de automatizar procesos para ganar tiempo con resultados óptimos a corto y largo plazo; fue por esta razón que en el año 1950 Alan Turing creó el "Test de Turing" en el cual una máquina tenía que ser capaz de engañar a un humano haciéndole creer que era humana en lugar de un computador. (Nadal, 2019). En el año 1952 Arthur Samuel escribe el primer algoritmo capaz de aprender, este programa jugaba a las damas y mejoraba tras cada partida su juego. Posteriormente, en el seno de una conferencia de Darthmouth nace el término "Inteligencia Artificial" para nombrar el nuevo campo que se estudiaba en el verano de 1956. (González, 2019). El resurgimiento interés en el aprendizaje basado en máquina se debe a la gran variedad, capacidad computacional actual y cantidad de datos disponibles que incentiva la creación de nuevas técnicas como Machine Learning para el procesamiento de datos de todo tipo.

Machine Learning es una rama de la Inteligencia Artificial capaz de crear sistemas de aprendizaje automático que a través de algoritmos permite a las maquinas aprender sin necesidad de ser programadas, identificando patrones en datos masivos y elaborando predicciones, capaces de auto programarse aprendiendo de su propia experiencia combinando datos de entradas y situaciones del mundo real. (Nadal, 2019). Dividido en tres categorías, aprendizaje supervisado, no supervisado y aprendizaje por refuerzo. El aprendizaje supervisado se basa en modelos predictivos que hacen uso de datos de entrenamiento, a partir de un conjunto de datos conocidos, se pretende que el sistema sea capaz de lograr una determinada salida logrando los resultados esperados; en el aprendizaje no supervisado el algoritmo estudia los datos para identificar patrones que le permitan organizarlos de alguna manera, es decir no cuenta con un conocimiento previo y en el aprendizaje por refuerzo se aprende a partir de la propia experiencia, deberá tomar la mejor en un proceso de prueba y error en el que se recompensan las decisiones correctas. (Baquero & Valdés, 2019).

Deep Learning, también conocido como aprendizaje profundo es un tipo de Machine Learning que tuvo sus orígenes en 2006 cuando Geoffrey Hinton acuña el término para explicar nuevas arquitecturas de Redes Neuronales profundas capaces de aprender mucho mejor que modelos más planos. (González, 2019). Este algoritmo usa redes neuronales artificiales compuestas por niveles jerárquicos; el primer nivel permite a la red aprender algo simple y enviar la información al siguiente nivel; en el siguiente nivel toma dicha información y la convierte en algo un poco más complejo y lo remite al siguiente nivel y así sucesivamente hasta obtener el resultado esperado mediante aprendizaje progresivo. (Baquero & Valdés, 2019).

Es importante en Deep Learning optimizar los hiperparametros de la red para mejorar la representación de los datos, estos parámetros ajustables se eligen para preparar el modelo que rige el proceso de entrenamiento, el rendimiento del modelo depende en gran medida de los valores de hiperparametros seleccionados porque se deben buscar varias configuraciones hasta obtener un resultado óptimo, lo que puede ser un proceso dispendioso. (Gómez & Ortega, 2020).

Las redes neuronales más usadas son las redes neuronales recurrentes (RNN), redes generativas antagónicas (GAN) y redes neuronales convolucionales (CNN). Las RNN fueron concebidas en la década de 1980 que usan datos de series de tiempo, dado solución a problemas ordinales o temporales tomando información de inputs anteriores para influenciar los inputs y outputs actuales. Las GAN presentadas en el 2014, usa dos redes neuronales artificiales que se oponen entre ellas para generar datos sintéticos que pueden pasar como reales; una red genera contenido artificial y la otra discrimina porque es entrenada para reconocer contenido real; y las CNN creadas en 1980, para procesar matrices estructuradas, como imágenes que se pueden clasificar basándose en patrones y objetos que aparecen en ellas (Baquero & Valdés, 2019); trabajan modelando de forma consecutiva pequeñas piezas de información, que luego combinan en las capas más profundas de la red; compuestas por una capa de entrada, una de salida y capas ocultas, suponiendo explícitamente que las entradas son imágenes, codificando algunas propiedades en la arquitectura para reconocer

objetos concretos en ellas, cada capa va aprendiendo de diferentes niveles de abstracción, por lo que un número significativo de capas puede lograr identificar estructuras complejas en los datos de entrada, permitiendo ganar eficiencia y reducir la cantidad de parámetros en la red.

Para que la red aprenda a reconocer una diversidad de objetos dentro de una imagen es necesario que se realice el entrenamiento para ir obteniendo los parámetros que más se ajusten al problema se hace una aproximación inicial hacia adelante de la red (forward propagation) para obtener una salida aproximada. Con esa salida nos devolvemos hacia atrás para refinar los parámetros iniciales. Dentro de tensorflow esto es un epoch. Se realiza varias veces hasta que los parámetros de evaluación se ven aceptables accuracy, loss y learning curves; es decir que se determina la importancia que tendrá esa relación en la neurona al multiplicarse por el valor de entrada; sin embargo, es casi imposible obtener un cien por ciento de predicciones correctas (Baquero & Valdés, 2019).

El objetivo del entrenamiento es medir la pérdida para luego ir iterando hasta que el algoritmo descubra los parámetros del modelo con la pérdida más baja posible. Existen varias técnicas para evitar un mal entrenamiento, sin embargo, con una correcta elección de parámetros la red podrá obtener buenas soluciones con los siguientes resultados no convergencia y convergencia, en el primero la red es incapaz de encontrar una solución y continúa iterando sin fin y convergencia ocurre cuando se encuentra una aproximación al problema ya que n o existen modelos perfectos en Machine Learning, con tres posibilidades, underfitting o subajuste que ocurre cuando los datos de entrenamiento son muy pocos y la red no será capaz de generalizar, overfitting o sobreajuste se da cuando la red solo aprende de los casos particulares y será incapaz de reconocer nuevos datos y un buen entrenamiento cuando la red genera predicciones con mucha precisión pero a su vez ha desarrollado una capacidad de generalización que le permite predecir correctamente. (Viñuela, 2004)

La principal aplicación de los algoritmos de Deep Learning son las tareas de clasificación, en especial, el reconocimiento de imágenes, traducción automática, asistencia sanitaria,

automóviles autónomos, entre otros. Por lo que muchas empresas están transformando sus negocios procesando datos con el fin de obtener ventajas competitivas sobre la competencia.

En Colombia a través de la Ley 165 de 1994, suscribió el convenio sobre Diversidad Biológica en la cual se formuló la Política Nacional de Biodiversidad y adquiere el compromiso de crear y reglamentar el Sistema Nacional de Áreas Protegidas (SINAP). Las áreas protegidas se definen como el conjunto de zonas preservadas, actores sociales e instrumentos de gestión que se articulan para ayudar con el cumplimiento de los objetivos de conservación del país, conformadas principalmente por: el sistema de parques nacionales, reservas naturales, santuario de flora y fauna, áreas de manejo especial y zonas de reserva campesina. (SINAP, 2020).

El Instituto Geográfico Agustín Codazzi (IGAC) es la entidad encargada de producir la cartografía básica de Colombia, elaborar el catastro nacional de la propiedad inmueble, el mapa oficial, entre otros. La Subdirección de Catastro es la encargada de la producción, custodia, preservación y documentación de los procesos de formación, actualización, conservación del catastro y avalúos, para administrar el Sistema de Información Catastral en Colombia. (IGAC, 2021)

METODOLOGÍA

El presente proyecto utiliza un método descriptivo con el propósito de elaborar un algoritmo que permita identificar en imágenes aquellas zonas delimitadas para la extracción de hidrocarburos contenidas en áreas de protección ambiental con un enfoque mixto porque se realiza un análisis cuantitativo de las imágenes para ser implementadas en el modelo ya que deben contar con una resolución espacial acorde para delimitar la zona de interés, libres de nubes o en su de efecto que no interfieran con el área de estudio y cuantitativo con el fin de realizar los análisis descriptivos del modelo.

Descripción Zona de Estudio

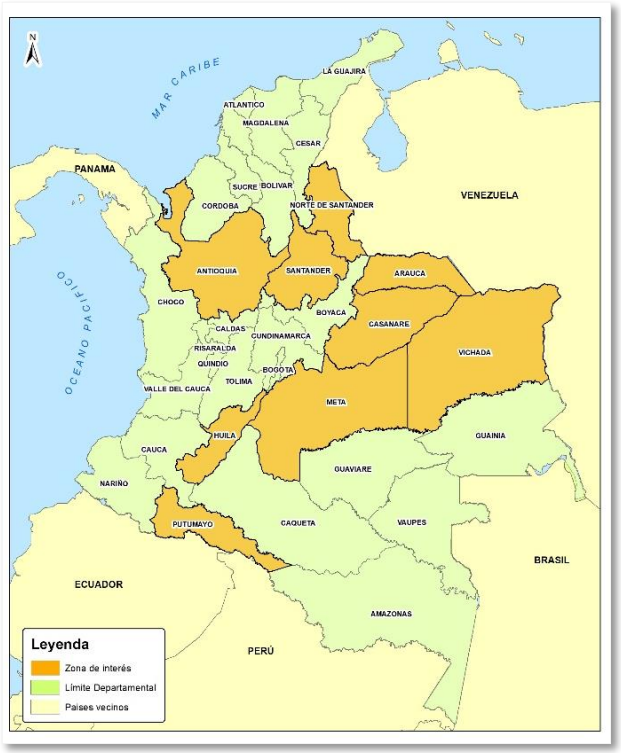

Figura 1 - Localización general zona de estudio

Fuente: Elaboración Propia a partir de la Información base del IGAC 2021.

Para el desarrollo de este proyecto se seleccionaron los departamentos de Antioquia, Santander, Norte de Santander, Arauca, Casanare, Meta, Huila, Vichada y Putumayo porque en ellos se concentra principalmente la problemática analizada. *Ver Figura 1.*

Diseño

Se implementaron cuatro fases de diseño que se presentan a continuación:

1. Selección de la muestra: Se seleccionaron 300 imágenes con zonas de explotación de hidrocarburos, divididas en dos grupos; 150 imágenes encontradas en áreas de protección ambiental y 150 en áreas sin restricciones ambientales con el fin de comparar las características en cada grupo. Dichas imágenes se obtuvieron con SAS Planet que es un programa gratuito con licencia GNU[1] usado para descargar mosaicos de los principales proveedores de servicios de mapas e imágenes con sistema de referencia WGS84 (EPSG:4326). Luego se realiza un Data Augmentation para introducir artificialmente transformaciones aleatorias, pero realistas a las imágenes de entrenamiento y así reducir el sobreajuste; luego se repiten todos los pasos descritos en la implementación del modelo comparando los resultados obtenidos antes y después de su ejecución.

2. Procesamiento de imágenes: Para el desarrollo del proyecto se utilizó Google Colab que es un servicio cloud, el cual permite usar y compartir cuadernos de Jupyter con otros usuarios sin tener que descargar e instalar nada, bajo el lenguaje de programación Python.

   En colab se realiza la importación de las 300 imágenes almacenadas en Google Drive, divididas en dos grupos: grupo N°1 con 150 imágenes encontradas en zonas con áreas protegidas (AP) y grupo N°2 con 150 imágenes localizadas en zonas sin áreas protegidas (ASP). Al cargar la base se debe visitar un enlace "Google Files Stream" con el fin de conceder los permisos necesarios para acceder a la unidad que muestra un código de autenticación alfanumérico que se ingresa en el cuaderno del Colab. Luego se elabora el análisis descriptivo y se crea un Dataset para almacenar, administrar y consultar los datos de cada imagen.

[1] Es una licencia de derecho de autor ampliamente usada en el mundo del software libre que garantiza a los usuarios finales la libertad de usar y modificar el software.

3. Implementación de modelos: Se inicia la modelación con Keras que es una API de aprendizaje profundo escrita en Python, ejecutada sobre la plataforma de aprendizaje automático TensorFlow, en los últimos años la comunidad de desarrolladores lo ha posicionado como la herramienta líder en el sector del Deep Learning.
   Se implementaron 5 arquitecturas de redes neuronales profundas preentrenadas para la clasificación de imágenes que fueron adaptas al problema de interés y se presentan a continuación:

   - MobilenetV2: *Inverted Residuals and Linear Bottlenecks*, diseñada para dispositivos móviles, se basa en una estructura residual inversa, en donde la entrada y la salida del bloque residual es un bottleneck más corto. Las redes móviles vienen en varios tamaños controlados por un multiplicador de la profundidad en las capas convolucionales; también se pueden entrenar para varios tamaños de imágenes de entrada con el fin de controlar la velocidad de inferencia. (TensorFlow, 2021)

   - InceptionV3: *Rethinking the Inception Architecture for Computer Vision*, permitir redes más profundas y evita que el número de parámetros crezca demasiado; contiene una instancia entrenada de la red, empaquetada para realizar la clasificación de imágenes en la que se entrenó la red, transforma imágenes en vectores de características. (TensorFlow, 2021)

   - ResNet50V2: *Identity Mappings in Deep Residual Networks*, se basa en la arquitectura ResNet con un número variable de capas, recurre al uso de la normalización por lotes antes de cada capa de peso; el modelo contiene una instancia entrenada de la red, empaquetada para realizar la clasificación de imágenes en la que se entrenó la red. (TensorFlow, 2021)

- NasNetMobile: *Learning Transferable Architectures for Scalable Image Recognition,* utiliza principalmente una capa normal y capa de reducción; aplica sus operaciones en un conjunto de datos pequeño y luego lo transfiere al conjunto de datos para una regularización efectiva mejorando el rendimiento de Nasnet. (TensorFlow, 2021)

- EfficientNetB3V2: *Rethinking Model Scaling for Convolutional Neural Networks,* logra una mejor eficiencia de parámetros y una velocidad de entrenamiento más rápida que técnicas anteriores, utiliza la búsqueda de arquitectura neuronal (NAS) para optimizar de manera conjunta el tamaño del modelo y la velocidad de entrenamiento; el modelo contiene una instancia entrenada de la red, empaquetada para realizar la clasificación de imágenes en la que se entrenó la red. (TensorFlow, 2021)

4. Análisis de resultados: una vez seleccionados los modelos con los mejores resultados se elaboran *Learning curves* para ver gráficamente la precisión y pérdida de entrenamiento y validación, se realizan *Check the predictions* y se ejecuta un *Data Augmentation* para introducir artificialmente transformaciones aleatorias, pero realistas a las imágenes de entrenamiento y así reducir el sobreajuste.

RESULTADOS

Los resultados se presentan conforme se desarrollan las fases de diseño previamente definidas para elaborar un algoritmo para identificar en imágenes las zonas delimitadas para la extracción de hidrocarburos contenidas en áreas de protección ambiental.

1. Selección de la muestra

   Se eligieron 300 imágenes con SAS Planet que es un programa gratuito usado para descargar mosaicos de los principales proveedores de servicios de mapas e imágenes con sistema de referencia WGS84 (EPSG:4326).

Figura 2 - Muestra de las imágenes seleccionadas.
Fuente: SAS Planet.

En la *Figura 2* se presenta una muestra de las imágenes utilizadas para el desarrollo de este proyecto divididas en dos grupos; 150 imágenes encontradas en áreas de protección ambiental y 150 en áreas sin restricciones ambientales con el fin de comparar las características en cada grupo.

2. Análisis descriptivo

Para elaborar los análisis descriptivos en cada grupo se seleccionó una muestra con el fin mostrar las generalidades y así comprender el comportamiento espacial de los datos con el propósito de facilitar su aplicación en el trabajo propuesto.

Zonas con áreas protegidas:

- Imagen seleccionada N°1: Del repositorio creado en Google drive se presenta la imagen 96 en composición de color natural *ver Figura 3*, esta combinación es la que más se aproxima a los colores reales de la superficie terrestre y solo utiliza las bandas que hacen parte de espectro electromagnético visible (bandas rojas, verdes y azules visibles) con los correspondientes canales rojos, verdes y azules.

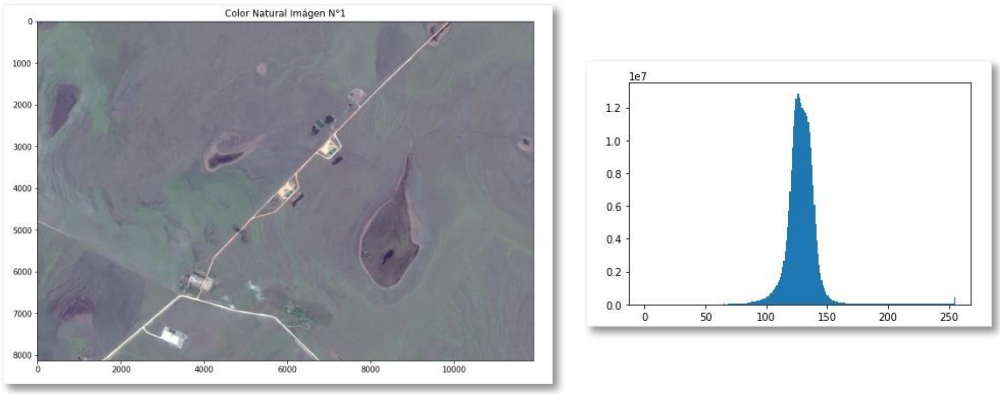

Figura 3 - Imagen en color natural *(Izquierda)* / Histograma *(derecha)*
Fuente: Elaboración propia - Google colab.

En la *Figura 3* se muestra el histograma es simétrico porque la información no está concentrada en los primeros niveles (razón por la cual la imagen tiende verse oscura) o en niveles finales (la imagen se vería clara), gráficamente se determina que la curtosis tiene una distribución leptocúrtica, con presencia de datos atípicos.

En la *Tabla 1* se presenta un resumen estadístico general para la imagen seleccionada N°1.

Tabla 1 - Datos obtenidos

Valor mínimo	0
Promedio	128,463
Valor máximo	255
N° de filas	8131
N° de columnas	11905
Canales	3
Tamaño	96799555 pixels

Fuente: Elaboración propia.

En la *Figura 4* se presenta la imagen en escala de grises que permite detectar fácilmente cambios de cobertura; el sensor de imagen es sensible a una gran cantidad de longitudes de onda de luz como evidencia se realzan las vías y locaciones petroleras que son las zonas delimitadas para la extracción de hidrocarburos. Las composiciones en falso color permiten visualizar las longitudes de onda que el ojo humano no ve; incrementa la separación espectral y puede mejorar la interpretación de datos, en color amarillo se detectan las zonas de interés.

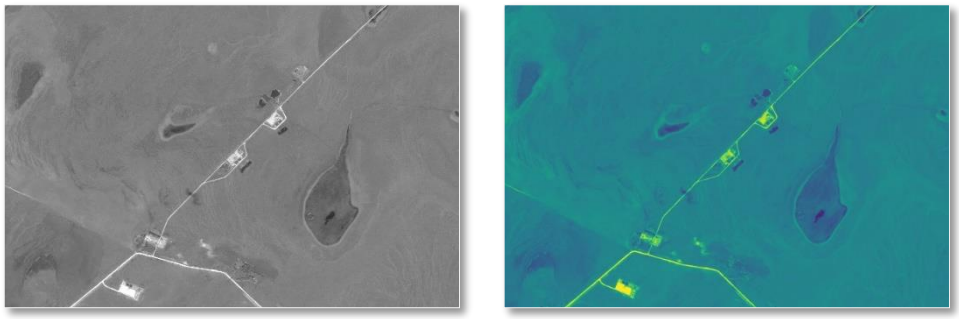

Figura 4 - Imagen pancromática (Izquierda) / Imagen en falso color (derecha)
Fuente: Elaboración propia - Google colab.

Una imagen RGB tiene tres canales (rojo, verde y azul), cada banda se puede filtrar para componer imágenes realzando elementos gracias a su comportamiento en el espectro electromagnético. *Ver Figura 5*

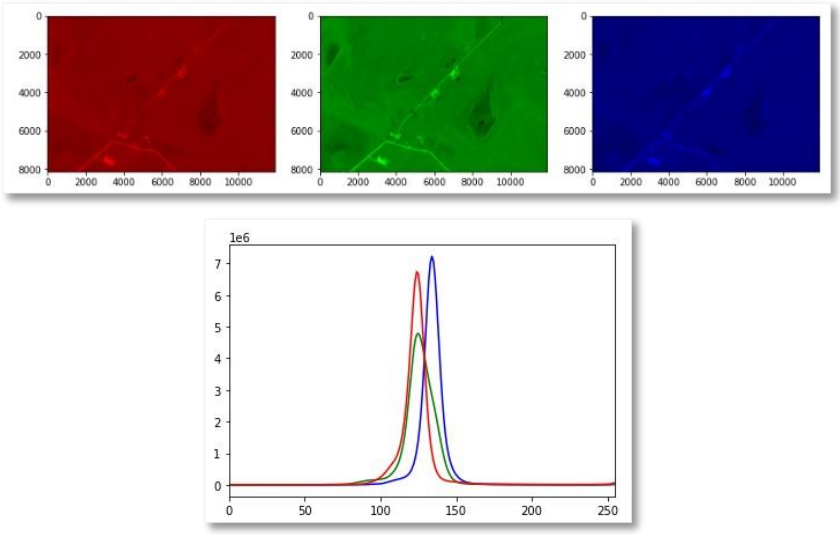

Figura 5 - Bandas visibles (arriba) – Histograma RGB (abajo)
Fuente: Elaboración propia - Google colab.

Los histogramas de la *Figura 5* son simétricos, cada banda en términos generales tiene un buen contraste visual, gráficamente se determina que la curtosis tiene una distribución leptocúrtica, pero en la banda verde la curtosis es pequeña comparada con las bandas azul y roja.

- Imagen seleccionada N°2: Del repositorio creado en Google drive se presenta la imagen 95 en composición de color natural *ver Figura 6*, también se muestra el histograma que es simétrico, gráficamente se determina que la curtosis tiene una distribución leptocúrtica, con presencia de datos atípicos.

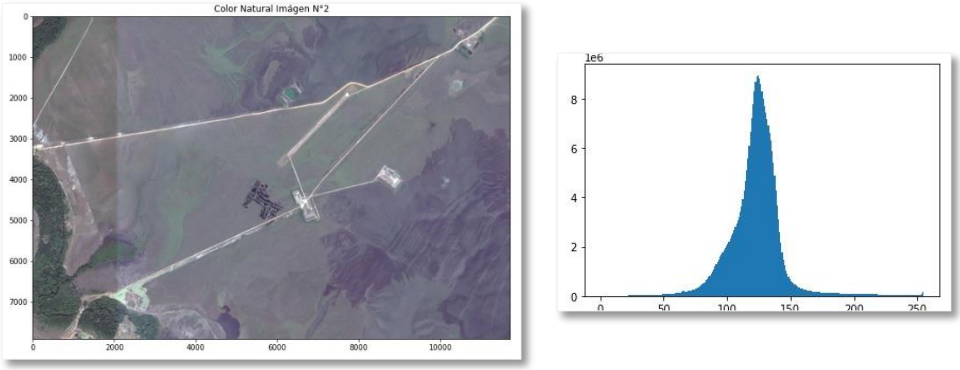

Figura 6 - Imagen en color natural (Izquierda) / Histograma (derecha)
Fuente: Elaboración propia - Google colab.

En la *Tabla 2* se presenta un resumen estadístico general para la imagen seleccionada N°2.

Tabla 2 - Datos obtenidos

Valor mínimo	0
Promedio	122,304
Valor máximo	255
N° de filas	7908
N° de columnas	11713
Canales	3
Tamaño	92626404 pixels

Fuente: Elaboración propia.

En la *Figura 7* se presenta la imagen en escala de grises y una composición falso color, en ambas figuras se mejora la interpretación las vías y locaciones petroleras que son las zonas delimitadas para la extracción de hidrocarburos.

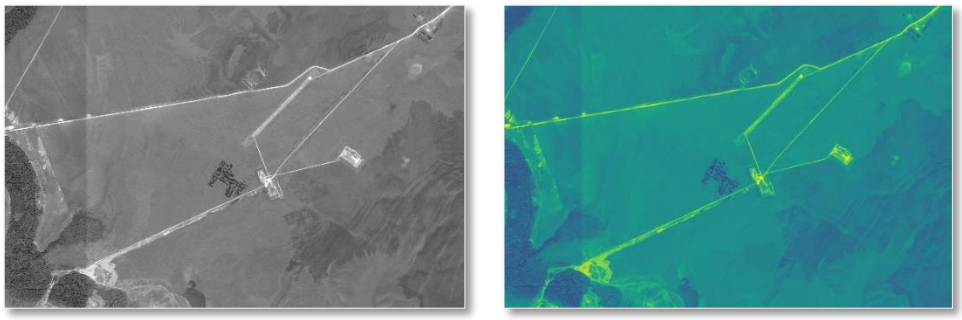

Figura 7 - Imagen pancromática (Izquierda) / Imagen en falso color (derecha)
Fuente: Elaboración propia - Google colab.

Los histogramas de la *Figura 8* son simétricos, cada banda en términos generales tiene un buen contraste visual, gráficamente se determina que la curtosis tiene una distribución leptocúrtica, pero en la banda azul la curtosis es grande comparada con las bandas verde y roja.

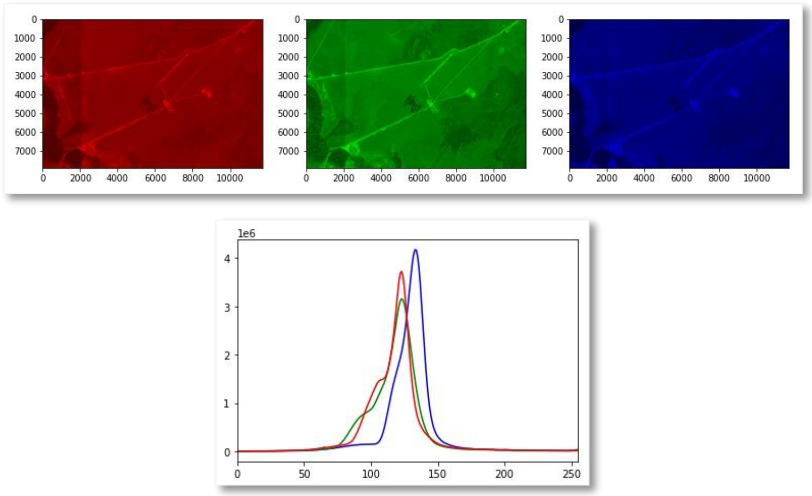

Figura 8 - Bandas visibles (arriba) – Histograma RGB (abajo)
Fuente: Elaboración propia - Google colab.

Zonas sin áreas protegidas:

- Imagen seleccionada N°3: Del repositorio creado en Google drive se presenta la imagen 121 en composición de color natural *ver Figura 9*, esta combinación es la que más se aproxima a los colores reales de la superficie terrestre y solo utiliza las bandas que hacen parte de espectro electromagnético visible (bandas rojas, verdes y azules visibles) con los correspondientes canales rojos, verdes y azules.

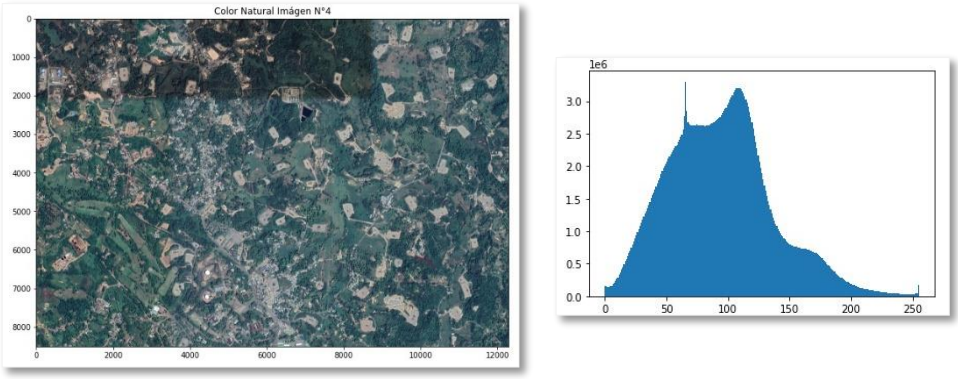

Figura 9 - Imagen en color natural (Izquierda) / Histograma (derecha)
Fuente: Elaboración propia - Google colab.

En la *Figura 9* el histograma del cual se puede inferir es asimétrico hacia la derecha, la información de los primeros niveles hace que la imagen se vea oscura en algunas zonas, esto se debe a la cantidad de elementos presentes en la escena, gráficamente se determina que la curtosis tiene una distribución leptocúrtica con presencia de datos atípicos.

En la *Tabla 3* se presenta un resumen estadístico general para la imagen seleccionada N°3.

Tabla 3 - Datos obtenidos

Valor mínimo	0
Promedio	93,519
Valor máximo	255
N° de filas	8513
N° de columnas	12289
Canales	3
Tamaño	104616257 pixels

Fuente: Elaboración propia.

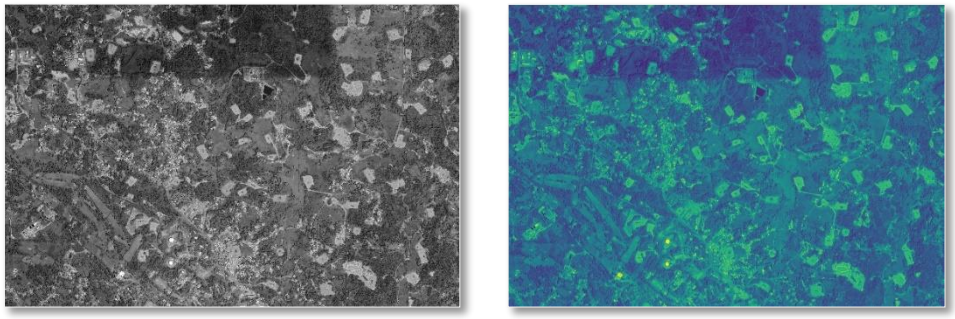

Figura 10 - Imagen pancromática (Izquierda) / Imagen en falso color (derecha)
Fuente: Elaboración propia - Google colab.

En la *Figura 10* se presenta la imagen en escala de grises que permite detectar fácilmente cambios de cobertura; el sensor de imagen es sensible a una gran cantidad de longitudes de onda de luz como evidencia se realzan las vías y locaciones petroleras que son las zonas delimitadas para la extracción de hidrocarburos. Las composiciones en falso color permiten visualizar las longitudes de onda que el ojo humano no ve; incrementa la separación

espectral y puede mejorar la interpretación de datos, en color amarillo se detectan las zonas de interés.

Una imagen RGB tiene tres canales (rojo, verde y azul), cada banda se puede filtrar para componer imágenes realzando elementos gracias a su comportamiento en el espectro electromagnético. *Ver Figura 11*

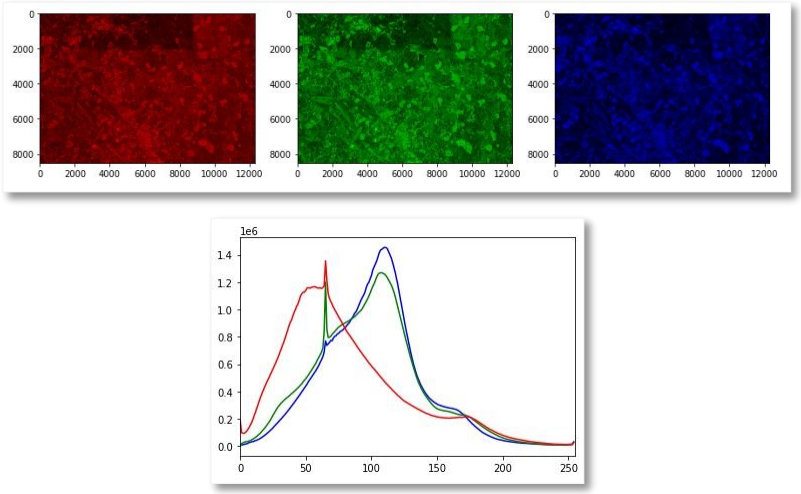

Figura 11 - Bandas visibles (arriba) – Histograma RGB (abajo)
Fuente: Elaboración propia - Google colab.

Los histogramas de la *Figura 11* son asimétricos hacia la derecha con bastante información espacial, gráficamente se determina que la curtosis tiene una distribución leptocúrtica.

- Imagen seleccionada N°4: Del repositorio creado en Google drive se presenta la imagen 51 en composición de color natural y se presenta el histograma a partir del cual se puede inferir que es asimétrico hacia la derecha, la información está concentrada en los primeros niveles, gráficamente se determina que la curtosis tiene una distribución leptocúrtica, con presencia de datos atípicos. *Ver Figura 12*.

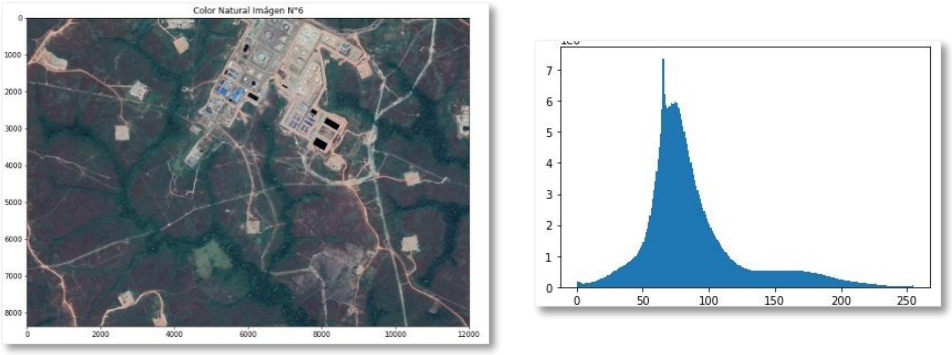

Figura 12 - Imagen en color natural (Izquierda) / Histograma (derecha)
Fuente: Elaboración propia - Google colab.

En la *Tabla 4* se presenta un resumen estadístico general para la imagen seleccionada N°4.

Tabla 4 - Datos obtenidos

Valor mínimo	0
Promedio	87,096
Valor máximo	255
N° de filas	8367
N° de columnas	12001
Canales	3
Tamaño	100412367 pixels

Fuente: Elaboración propia.

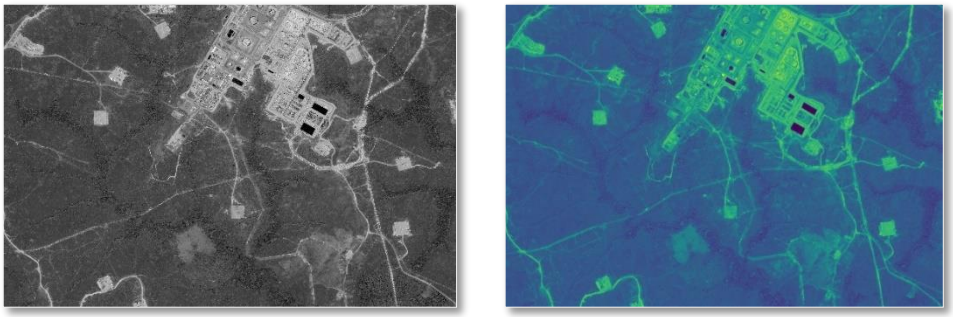

Figura 13 - Imagen pancromática (Izquierda) / Imagen en falso color (derecha)
Fuente: Elaboración propia - Google colab.

En la *Figura 13* se presenta la imagen en escala de grises y una composición falso color, con estas combinaciones se mejora la interpretación las vías y locaciones petroleras que son las zonas delimitadas para la extracción de hidrocarburos.

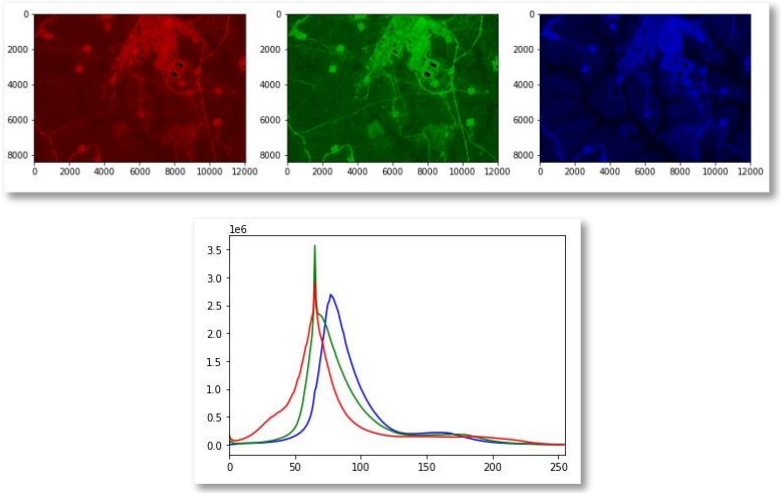

Figura 14 - Bandas visibles (arriba) – Histograma RGB (abajo)
Fuente: Elaboración propia - Google colab.

Los histogramas de la *Figura 14* son asimétricos hacia la derecha con un ajuste deficiente, en términos generales cada banda no tiene el mejor contraste visual, gráficamente se determina que la curtosis tiene una distribución leptocúrtica y en la banda verde la curtosis es grande comparada con las bandas roja y azul.

En términos generales con la información obtenida en los histogramas, se puede inferir que los niveles digitales de las imágenes seleccionadas no están concentrados en los primeros niveles (razón por la cual la imagen tiende verse oscura) o en niveles finales (la imagen se vería clara). Los análisis descriptivos evidencian que las 300 imágenes seleccionadas para el desarrollo de este proyecto cuentan con las especificaciones técnicas esperadas.

Nubes:

Eliminar la presencia de nubes en las imágenes no es el objetivo de este proyecto, por lo tanto, se seleccionaron imágenes libres de nubes o en su de efecto que no interfieran con las zonas delimitadas para la extracción de hidrocarburos. A continuación, se muestra como la presencia de nubes podrían afectar procesos automatizados porque se pueden confundir con otro tipo de coberturas. *Ver Figura 15.*

Figura 15 – Imagen en color natural (Izquierda) / Imagen pancromática (Mitad) / Imagen en falso color (derecha)
Fuente: Elaboración propia - Google colab.

3. Implementación del modelo

   Una vez realizada la importación de las 300 imágenes almacenadas en Google Drive, divididas en dos clases, la primera con 150 imágenes encontradas en zonas con áreas protegidas **(AP)** y la segunda con 150 imágenes localizadas en zonas sin áreas protegidas (**SAP)**, se creó un *Dataset* para almacenar, administrar y consultar los datos de cada imagen.

   Luego el conjunto de datos se divide en 80 - 20 para entrenamiento y para validación respectivamente; se define el tamaño *batch* de imágenes a procesar en este caso de 32, un *seed* o *semilla* de 123, una altura y ancho que varía de acuerdo con las condiciones de cada modelo. Se preprocesa el valor de los pixeles y se normalizan para que tengan un valor entre 0 y 1, por eso se divide en 255. También se realiza una optimización para asegurar que el número de elementos para captar previamente debe ser igual o mayor al número de lotes consumidos por un solo paso de entrenamiento, por lo tanto se implementó *tf.data.AUTOTUNE* para ajustar el valor dinámicamente en tiempo de ejecución.

   Con el fin de determinar a partir de los resultados obtenidos cuál es el modelo que más se ajusta al presente trabajo, se seleccionaron 5 arquitecturas previamente entrenadas de clasificación de imágenes de TensorFlow Hub *(MobilenetV2, InceptionV3, ResNet50V2, NasNetMobile y EfficientNetB3V2)*.

Para implementar cada modelo se usó *feature_extractor_model* que permite extraer las características preentrenadas de aprendizaje profundo con una URL única para asignada cada modelo en el módulo *TF2- SavedModel* de TensorFlow Hub, permitiendo que la imagen de entrada se propague hacia adelante y tome las salidas de esa capa como características propias.

Se usa el modelo **Sequential** determina el orden de las capas dentro del modelo, apropiado para Layers o capas simples en donde cada capa tiene exactamente un tensor de entrada y un tensor de salida. Luego se usó **model.compile** para definir el oprtimizador, la pérdida y la métrica de precisión:

- Se emplea como optimizador **Adam** que es un método de descenso de gradiente estocástico, basado en la estimación adaptativa de momentos de primer y segundo orden, se adaptó bastante bien en el modelo propuesto.
- Loss function se usó **sparse categorical crossentropy** porque se utiliza con mucha frecuencia en cuando hay dos o más clases de etiquetas.
- La métrica de precisión seleccionada fue **Accuracy** porque es efectiva en determinar la frecuencia con la que las predicciones son iguales a las etiquetas.

Log_dir define el nombre de la ruta del directorio al que se enviarán los registros de ejecución del sistema.

Para entrenar el modelo se emplea **model.fit** con un número determinado de *Epochs*, *train_ds* con imágenes para entrenamiento, *validation_data* usada para evaluar la perdida y cualquier métrica del modelo al final de cada iteración y *callbacks* que es la lista de devoluciones de llamada para aplicar durante la predicción. Se uso **Model.summary()** que proporciona información sobre las capas y su orden en el

modelo, la forma de salida de cada capa, el número de parámetros en cada capa y total de parámetros en el modelo y se llevó a cabo una visualización sencilla del modelo resultante.

Se elaboraron **Learning curves** para ver gráficamente de la precisión y pérdida de entrenamiento y validación; luego se realizaron los **Check the predictions;** a los modelos con los mejores resultados se les realizó un *Data Augmentation* para introducir artificialmente transformaciones aleatorias, pero realistas a las imágenes de entrenamiento y así reducir el sobreajuste; luego se repiten todos los pasos descritos en la implementación del modelo comparando los resultados obtenidos antes y después de su ejecución.

4. Análisis de resultados:

   En la *Tabla 5 se* presenta un resumen con los principales los resultados obtenidos.

Tabla 5 - Datos obtenidos

	Modelo 1 MobilenetV2	Modelo 2 InceptionV3	Modelo 3 ResNet50V2	Modelo 4 NasNetMobile	Modelo 5 EfficientNetB3V2
Img_height	224	800	800	224	800
Img_width	224	800	800	224	800
Epochs	10	10	7	10	7
KerasLayer	1280	1001	1001	1001	1000
Trainable params	2562	2004	2004	2004	2002
Non-trainable params	2257984	23853833	25615849	5327773	14467622
Total params	2260546	23855837	25617853	5329777	14469624
Loss	0,2109	0,3003	0,303	0,3817	0,286
Acc	0,9625	0,8917	0,8583	0,8583	0,8875
Val_Loss	0,3398	0,3137	0,2905	0,5163	0,2685
Val_Acc	0,8833	0,8833	0,9	0,7167	0,9167

Fuente: Elaboración propia.

Las imágenes del Dataset en los modelos *MobilenetV2* y *NasNetMobile* quedaron con un arreglo de *(224, 224, 3)*, esto hace referencia al tamaño del píxel y las bandas de colores, en los demás modelos implementados fue posible establecer un mayor tamaño de píxel de *800x800* con la misma cantidad de bandas. En todos los modelos se definió la misma división de validación, utilizando *80%* de las imágenes para entrenamiento y el *20%* para validación. Se definieron 2 clases, *AP* (zonas con áreas protegidas) y *ASP* (zonas sin áreas protegidas); luego se estandarizaron los datos porque los valores de los canales RGB están en un rango de 0 a 255 y esto no es ideal en una red neuronal; por lo tanto, se pasan a valores de entrada entre 0 y 1. Se configuro el conjunto de datos para el rendimiento del disco y reducir los tiempos de ejecución en los pasos de entrenamiento de manera eficiente.

En todos los modelos seleccionados se utilizó *Sequential* para crear instancias de las clases y *model.compile* con el oprtimizador *Adam*, la pérdida *sparse categorical crossentropy* y la métrica de precisión *Accuracy*. Para entrenar los modelos se usó el método *fit()*, teniendo en cuenta las características de entrada *train_ds*; la cantidad de épocas para entrenar que fueron definidas luego de varias iteraciones de prueba, para los modelos *ResNet50V2* y *EfficientNetB3V2* quedaron 7 y en los demás modelos 10; se dejó un conjunto de validación para medir la pérdida y las métricas adicionales al final de cada época para evaluar el desempeño del modelo y *callbacks* para obtener la lista de devoluciones aplicadas durante la predicción.

Los resultados obtenidos después de las iteraciones realizadas por cada modelo alcanzaron una exactitud sobre el set de datos de entrenamiento así *MobilenetV2* con 96%, *InceptionV3* con 89%, *ResNet50V2* 86%, *NasNetMobile* 86%, *EfficientNetB3V2* 88% y sobre el set de validación de *MobilenetV2* con 88%, *InceptionV3* con 88%, *ResNet50V2* 90%, *NasNetMobile* 71%, *EfficientNetB3V2* 91%. En las *Figura 16* a la *18* se muestran las curvas de aprendizaje para cada modelo con el fin de comprender gráficamente el rendimiento del aprendizaje a lo largo de su ejecución; en términos generales se puede inferir que el ajuste en los modelos no obedece a líneas planas ni tampoco disminuyen, por lo tanto, es capaz de aprender del set de datos, es decir no comete *underfitting;* tampoco se comente *overfitting*

porque no se observan fluctuaciones aleatorias. Por tal razón, son curvas de aprendizaje con buen ajuste porque se identifican pérdidas de entrenamiento y validación que disminuyen hasta un punto de estabilidad con una brecha mínima entre los dos valores de pérdida final.

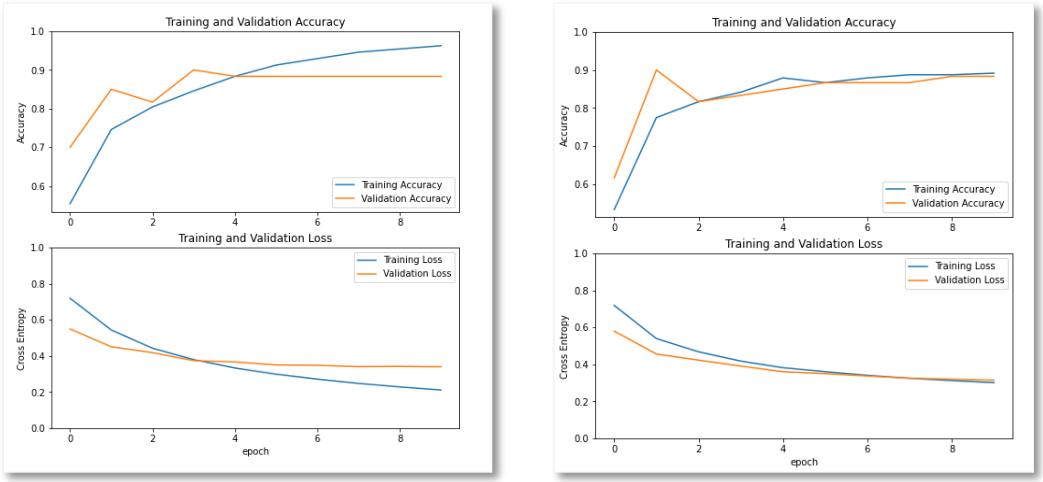

Figura 16 - Learning curves para MobilenetV2 (Izquierda) / Learning curves para InceptionV3 (derecha)
Fuente: Elaboración propia - Google colab.

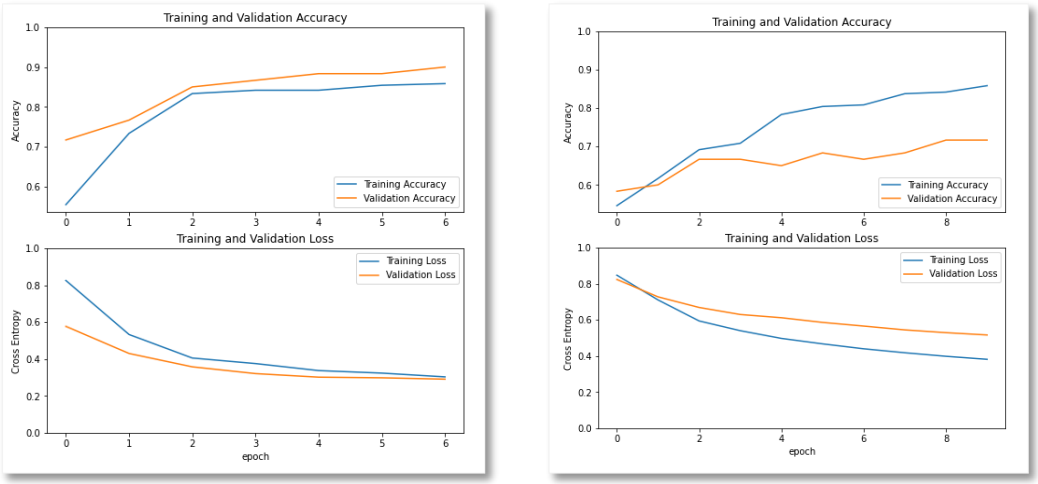

Figura 17 - Learning curves para ResNet50V2 (Izquierda) / Learning curves para NasNetMobile (derecha)
Fuente: Elaboración propia - Google colab.

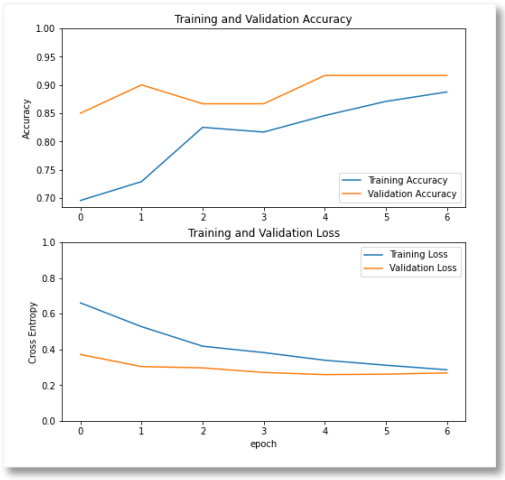

Figura 18 - Learning curves para EfficientNetB3V2
Fuente: Elaboración propia - Google colab.

En la *Figura 19* combinaron las curvas de aprendizaje de las predicciones obtenidas en en los modelos implementados.

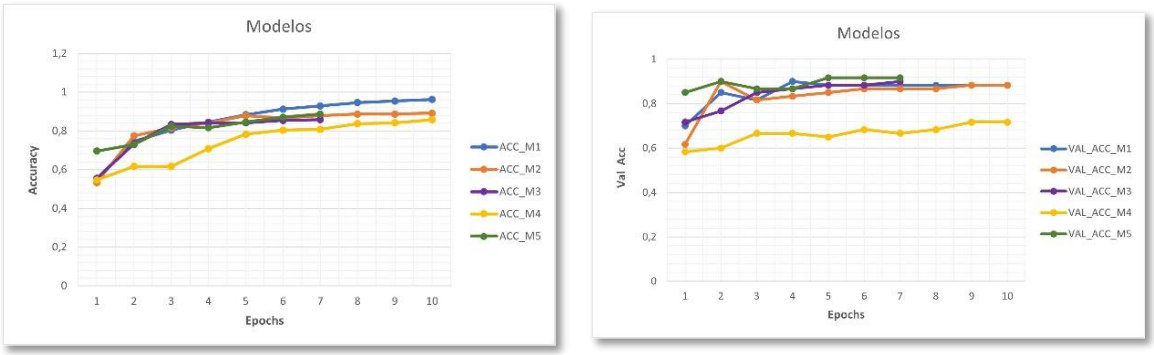

Figura 19 - Learning curves para el set de entrenamiento (Izquierda) / Learning curves para el set de validación (derecha)
Fuente: Elaboración propia - Google colab.

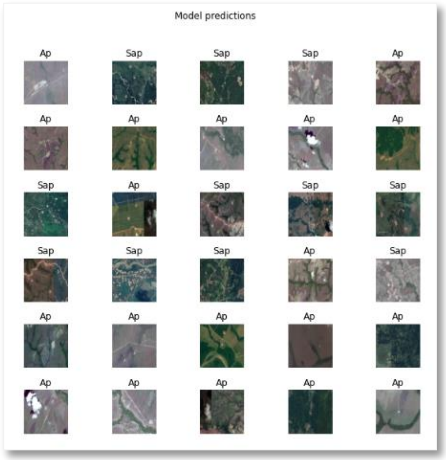

Figura 20 - Predicciones
Fuente: Elaboración propia - Google colab.

En la *Figura 20* se presentan las predicciones obtenidas en los modelos implementados, para todos se obtuvieron resultados satisfactorios.

Por todo lo anterior, los modelos que mejor se adaptaron a las necesidades del proyecto son *ResNet50V2* y *EfficientNetB3V2*. A los modelos con los mejores resultados se les realizó un *Data Augmentation* para introducir artificialmente transformaciones aleatorias, pero realistas a las imágenes de entrenamiento y se repitieron los ítems descritos en la implementación.

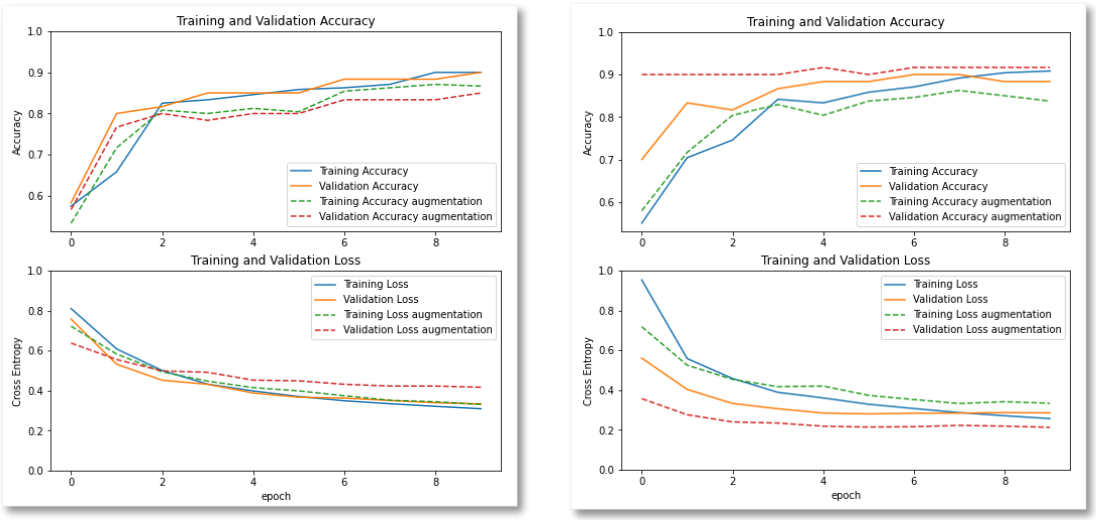

Figura 21 - Learning curves para InceptionV3 (Izquierda) / Learning curves para EfficientNetB3V2 (derecha)
Fuente: Elaboración propia - Google colab.

Las iteraciones realizadas por cada modelo alcanzaron una exactitud sobre el set de datos de entrenamiento así *InceptionV3* con 89% - 86%, *EfficientNetB3V2* 88% - 83% y sobre el set de validación de *InceptionV3* con 88%- 85%, *EfficientNetB3V2* 91% - 90% sin data y con data respectivamente, se redujo la posibilidad de un sobreajuste.

CONCLUSIONES

1. Se creó un Dataset de imágenes libres de nubes con zonas destinadas para la perforación de pozos petroleros divididas en dos grupos; 150 imágenes encontradas en áreas de protección ambiental y 150 en áreas sin restricciones ambientales comparando las principales características en cada grupo. Los análisis descriptivos evidenciaron que las 300 imágenes seleccionadas cumplieron con los requerimientos necesarios porque las coberturas de interés son visibles lo que indica que los rasgos presentes no requieren algún tipo de realce temporal o permanente, se puede inferir que los niveles digitales de las imágenes están equilibrados porque no se concentran en los primeros niveles *(razón por la cual la imagen tiende verse oscura)* o en niveles finales *(la imagen se vería clara)*.

2. Se implementaron 5 arquitecturas de redes neuronales profundas preentrenadas para la clasificación de imágenes de *TensorFlow Hub* que fueron adaptas al problema de interés *MobilenetV2, InceptionV3, ResNet50V2, NasNetMobile* y *EfficientNetB3V2*.

3. Los resultados obtenidos después de las iteraciones realizadas por cada modelo alcanzaron una exactitud sobre el set de datos de entrenamiento así *MobilenetV2* con 96%, *InceptionV3* con 89%, *ResNet50V2* 86%, *NasNetMobile* 86%, *EfficientNetB3V2* 88% y sobre el set de validación de *MobilenetV2* con 88%, *InceptionV3* con 88%, *ResNet50V2* 90%, *NasNetMobile* 71%, *EfficientNetB3V2* 91%. En términos generales las curvas de aprendizaje permiten inferir que el ajuste en los modelos no comete *overfitting* o *underfitting*, son curvas con buen ajuste porque se identifican pérdidas de entrenamiento y validación que disminuyen hasta un punto de estabilidad con una brecha mínima entre los dos valores de pérdida final.

4. Los modelos que mejor se adaptaron a las necesidades del proyecto son *ResNet50V2* y *EfficientNetB3V2*. Las iteraciones realizadas por cada modelo alcanzaron una exactitud sobre el set de datos de entrenamiento así *InceptionV3* con 89% - 86%,

EfficientNetB3V2 88% - 83% y sobre el set de validación de *InceptionV3* con 88%-85%, *EfficientNetB3V2* 91% - 90% sin data y con data respectivamente, se redujo la posibilidad de un sobreajuste.

5. Se realizó un *Data Augmentation* a los modelos con los mejores resultados para introducir artificialmente transformaciones aleatorias, pero realistas a las imágenes de entrenamiento y así reducir el sobreajuste.

6. En gestión inmobiliaria será bastante útil implementar este tipo de análisis para reducir el tiempo en la adquisición y administración de derechos superficiarios, por lo tanto al seguir alimentando los modelos implementados con las demás características que se deben calificar y que se puedan detectar en imágenes, facilitará y reducirá las actividades ejecutadas en campo con datos que permiten acotar el tiempo de elaboración de un diagnóstico jurídico catastral que se debe repetir en cada ocasión que requiera ingresar al predio objeto de estudio.

REFERENCIAS BIBLIOGRÁFICAS

ANH. (2019). *Agencia Nacional de Hidrocarburos*. Obtenido de https://www.anh.gov.co/

Castelo Cabay, M. J., Buñay Gualoto, G. I., & Pillajo Landa, B. G. (2021). Uso de Redes Neuronales Artificiales y Computación en la Nube para clasificar la cobertura del suelo. *Polo del conocimiento, 6*(5).

Gómez Sarasa, C. C., & Ortega Pabón, J. D. (2020). Técnicas de aumento de datos para imágenes aéreas y evaluación de rendimiento en modelos de. *Revista Universidad Católica de Oriente, 31*(45).

González Diez, S. (2019). *Evaluación del uso de redes neuronales convolucionales para clasificación de cultivos mediante.* Tesis de grado, Universidad Distrital Francisco José de Caldas, Facultad de ingeniería, Bogotá, Colombia.

IGAC. (2021). *Instituto Geográfico Agustín Codazzi*. Obtenido de https://www.igac.gov.co/

Nadal, V. (2019). *FutureSpace*. Obtenido de https://www.futurespace.es/machine-learning-los-origenes-y-la-evolucion/

Rodríguez Rodríguez, J. E. (2008). Redes Neuronales Artificiales para la Clasificación de Imágenes Satelitales. *Avances en investigación en ingeniería, 1*(9).

SINAP. (2020). *Sistema Nacional de Áreas Protegidas*. Obtenido de https://www.parquesnacionales.gov.co/portal/es/sistema-nacional-de-areas-protegidas-sinap/

TensorFlow. (2021). Obtenido de https://www.tensorflow.org/

Valdés Ávila, L. S., & Baquero Vanegas, J. M. (2019). *Deep Learning aplicado a imágenes satelitales como herramienta de detección de viviendas sin servicio de energía en el caserío Media Luna-Uribia-Guajira.* Tesis de grado, Universidad Distrital Francisco José de Caldas, Facultad de ingeniería, Bogotá, Colombia.

Viñuela, P., & Galván León, I. (2004). *Redes de Neuronas Artificiales.* Madrid, España: Pearson Prentice Hall.