

CONAHCYT

CONSEJO NACIONAL DE HUMANIDADES
CIENCIAS Y TECNOLOGÍAS



CentroGeo

Centro de Investigación en
Ciencias de Información Geoespacial, A.C.

**CENTRO DE INVESTIGACIÓN EN
CIENCIAS DE INFORMACIÓN GEOESPACIAL, A.C.**

CentroGeo

Centro Público de Investigación CONAHCYT

Modelo de segmentación semántica de columnas de humo
derivadas de incendios forestales en México

TESIS

Que para obtener el grado de

Maestro en Ciencias de Información Geoespacial

Presenta

Colvert Gomez Rubio

Directora de Tesis

Dra. Daniela Alejandra Moctezuma Ochoa

Ciudad de México

2024

CENTRO DE INVESTIGACIÓN EN
CIENCIAS DE INFORMACIÓN GEOESPACIAL, A.C.

CentroGeo

Centro Público de Investigación CONAHCYT

Modelo de segmentación semántica de columnas de humo
derivadas de incendios forestales en México

TESIS

Que para obtener el grado de
Maestro en Ciencias de Información Geoespacial

Presenta

Colvert Gomez Rubio

Director de Tesis

Dra. Daniela Alejandra Moctezuma Ochoa

Sinodales

Dra. Daniela Alejandra Moctezuma Ochoa

Dr. José Luis Silván Cárdenas

Dr. Rodrigo López Farías

Examinador Externo

Dra. Lilia de Lourdes Manzo Delgado

Ciudad de México, enero, 2024

Resumen

Los incendios forestales son un problema global que representan un riesgo para la salud humana, el medio ambiente y la economía. El humo producido por los incendios forestales genera emisiones de gases como CO, CO₂, NO_x y SO_x, que pueden afectar la calidad del aire, causar enfermedades respiratorias y contribuir significativamente al cambio climático.

En este estudio, se desarrolló un modelo de segmentación semántica de columnas de humo causadas por incendios forestales en México, utilizando imágenes satelitales GOES-16. Para esto, se hizo una revisión exhaustiva de las imágenes del sensor ABI (*Advanced Baseline Imager*), se elaboró un conjunto de datos de imágenes segmentadas a mano para entrenar un modelo y se exploraron soluciones basadas en visión por computadora para segmentar automáticamente una nueva imagen.

Los resultados del estudio mostraron que las bandas 1 (0.47 μ m), 2 (0.64 μ m) y el compuesto de color verdadero son los más relevantes para la detección de columnas de humo, ya que presentaron la mayor capacidad para distinguir las columnas de humo del resto de la imagen de manera efectiva. Así mismo, se generó un conjunto de datos de 1061 imágenes de columnas de humo de segmentadas a mano, este conjunto de datos se usó para el entrenamiento y la validación del modelo.

El modelo final se basa en la arquitectura U-Net de aprendizaje profundo y alcanzó métricas de 0.8252 de IoU, coeficiente Dice de 0.9042 y precisión de 0.9898 en el conjunto de prueba. Los resultados de este estudio demuestran que es factible desarrollar y hacer operativo un modelo de segmentación semántica de columnas de humo utilizando imágenes satelitales y aprendizaje profundo. Este modelo puede ser usado para monitorizar y detectar incendios forestales desde el espacio, lo que puede contribuir a mejorar la gestión de estos desastres.

A mi familia, por su apoyo incondicional y amor infinito.

Agradecimientos

Agradezco sinceramente a:

El Centro de Investigación en Ciencias de Información Geoespacial, A.C. Por brindarme el espacio y las oportunidades para desarrollar este proyecto de investigación. Mi gratitud también se extiende a CONAHCYT por otorgarme la beca que hizo posible este camino académico.

A mis respetados profesores, cuya dedicación y orientación han sido fundamentales en mi formación. Su conocimiento y apoyo han enriquecido mi aprendizaje de manera significativa.

A la Dra. Daniela Alejandra Moctezuma Ochoa, mi directora de tesis, por su liderazgo, guía y paciencia a lo largo de este proceso. Su experiencia y compromiso fueron esenciales para la culminación exitosa de este trabajo.

Mis distinguidos sinodales, la Dra. Lilia de Lourdes Manzo Delgado, el Dr. Rodrigo López Farías, y el Dr. José Luis Silván Cárdenas, por sus valiosas contribuciones y evaluación detallada de mi trabajo. Sus aportaciones han enriquecido notablemente la calidad de esta tesis.

A Nadxieli Mateu por su valiosa colaboración en una parte fundamental durante la creación del conjunto de datos.

Índice general

Acrónimos y abreviaturas.....	vii
Capítulo 1. Introducción.....	1
1.1. Contexto y justificación.....	1
1.2. Objetivo general.....	6
1.2.1. Objetivos Particulares.....	6
1.3. Hipótesis.....	6
1.4. Antecedentes y trabajos relacionados.....	6
1.5. Que se hace actualmente.....	11
Capítulo 2. Marco teórico.....	14
2.1. GOES-16.....	14
2.1.1. Resolución.....	15
2.2. Visión por computadora.....	18
2.2.1. Segmentación semántica.....	19
2.2.2. Aprendizaje automático.....	20
2.2.3. Red neuronal.....	21
2.2.3.1. Función de pérdida.....	22
2.2.3.2. Algoritmo de optimización.....	24
2.2.3.2.1. Descenso de gradiente.....	24
2.2.3.3. Algoritmo de propagación hacia atrás.....	25
2.2.4. Redes neuronales convolucionales (CNN).....	26
2.2.4.1. Convolución.....	26
2.2.4.2. Capas de agrupación.....	28
2.2.4.3. Convolución transpuesta.....	28
2.2.4.4. U-Net.....	29
2.2.5. Hiperparámetros.....	30
2.2.5.1. Número de capas o convoluciones iniciales.....	31
2.2.5.2. Función de activación.....	31
2.2.5.3. Algoritmos de optimización basados en descenso de gradiente.....	34
2.2.5.4. Tasa de aprendizaje.....	36
2.2.5.5. Tamaño de lote.....	37
2.2.6. Métricas de evaluación.....	38
Capítulo 3. Metodología.....	41
3.1. Diseño del conjunto de datos de columnas de humo.....	42
3.1.1. Determinación de la temporalidad de los datos.....	42
3.1.2. Selección de incendios que conforman el conjunto de datos.....	43

3.1.3. Análisis de las bandas y compuestos de GOES-16.....	44
3.1.4. Descarga de las imágenes GOES-16 y procesamiento.....	49
3.1.5. Segmentación manual.....	50
3.2. Modelo basado en U-Net.....	51
3.2.1. Aumento de datos.....	53
3.2.2. Dataset y DataLoader.....	55
3.2.3. Optimización de hiperparámetros.....	56
3.2.3.1. Número de capas o convoluciones iniciales.....	56
3.2.3.2. Función de activación.....	57
3.2.3.3. Optimizador.....	59
3.2.3.4. Tasa de aprendizaje.....	60
3.2.3.5. Tamaño de lote.....	62
Capítulo 4. Resultados y discusión.....	65
4.1. Descripción General de los Resultados.....	65
4.2. Modelo final.....	66
4.3. Aumento de datos.....	67
4.4. Métricas resultantes del modelo final.....	71
4.5. Ejemplo de segmentación.....	74
Capítulo 5. Conclusiones y trabajo futuro.....	77
5.1. Resumen de los objetivos.....	77
5.2. Validación de la hipótesis.....	78
5.3. Contribuciones y limitaciones.....	78
5.4. Trabajo futuro.....	80
5.5. Conclusión Final.....	81
Referencias bibliográficas.....	82

Índice de figuras

Figura 2.1: Flota operativa de GOES a abril de 2023, GOES-East (16) y GOES-West (17). Modificada de <i>Blaylock, (2023)</i>	14
Figura 2.2: Escenas de escaneo de ABI, Tomada de Mahonchak, (2019).....	18
Figura 2.3: Ejemplo de segmentación semántica para las clases persona (rosa), bicicleta (verde) y fondo (negro). Tomada de Lamba, (2019).....	20
Figura 2.4: Representación gráfica de modelo de perceptrón. Elaboración propia.....	21
Figura 2.5: Representación gráfica de una red neuronal muy sencilla. Elaboración propia.....	22
Figura 2.6: Ejemplo del mapa de activación resultado de una convolución con un kernel de , en este caso los pesos del kernel hacen que se resalten los bordes verticales de la imagen. Elaboración propia.....	27
Figura 2.7: Ejemplo de convolución transpuesta. Tomada de A. Zhang et al., (2023).....	29
Figura 2.8: Arquitectura original de la U-net. Tomada de Ronneberger et al., (2015).....	30
Figura 2.9: Gráfica de la función sigmoide. Elaboración propia.....	32
Figura 2.10: Gráfica de la función ReLU. Elaboración propia.....	32
Figura 2.11: Gráfica de la función Leaky ReLU con un α de 0.02. Elaboración propia.....	33
Figura 2.12: Gráfica de la función tanh. Elaboración propia.....	33
Figura 2.13: Gráfica de la función ELU con un α de 1. Elaboración propia.....	34
Figura 2.14: Ajuste de tasa e aprendizaje de un valor mínimo a un máximo por 3 ciclos. El eje x representa las iteraciones del modelo. Tomada de Smith, (2017).....	37
Figura 3.1: Diagrama de flujo de la metodología. Elaboración propia.....	41
Figura 3.2: Puntos de calor detectados por el sistema FIRMS usando MODIS de 2018 a 2022, agrupado por mes. Elaboración propia con datos de FIRMS (https://firms.modaps.eosdis.nasa.gov).....	43
Figura 3.3: Acercamiento a una columna de humo de un incendio forestal, coordenadas geográficas: (-104,-102, 19, 21) del día 1 de mayo de 2022 a las 18:47 UTC de las 16 bandas de GOES-16. Elaboración propia.....	45
Figura 3.4: Respuesta espectral de las bandas visibles ABI (banda 1 y 2) y diagrama de transmisión atmosférica (ventanas atmosféricas). Tomada de CIMSS and ASTER spectral library and Mat Gunshor.....	46
Figura 3.5: Acercamiento a una columna de humo de un incendio forestal, coordenadas geográficas: (-104,-102, 19, 21) del día 1 de mayo de 2022 a las 18:47 UTC en diversos compuestos generados con las bandas de GOES-16 . En un recuadro rojo se resaltan los compuestos donde se observan mas claramente la columna de humo. Elaboración propia.....	48
Figura 3.6: Ejemplo aleatorio de algunas de las imágenes pertenecientes al conjunto de datos generado. Estas imágenes muestran columnas de humo generadas a raíz de incendios forestales. Elaboración propia.....	50

Figura 3.7: Ejemplo de imágenes y su respectiva mascaró segmentada a mano. Elaboración propia.....	51
Figura 3.8: Representación gráfica de ajuste insuficiente (underfitting) y sobreajuste (overfitting). El eje x representa la complejidad del modelo y el eje y el error. Tomada de Smith, (2018).....	54
Figura 3.9: Ejemplo de transformaciones aleatorias aplicadas a imágenes del conjunto de entrenamiento, se aplica la misma transformación a la imagen y su máscara correspondiente. Elaboración propia.....	55
Figura 3.10: Comportamiento de la función de pérdida en el conjunto de validación durante el entrenamiento del modelo con diferentes convoluciones iniciales (8, 16, 32, 64). Elaboración propia.....	57
Figura 3.11: Comportamiento de la métrica de intersección sobre la unión en el conjunto de validación durante el entrenamiento del modelo con diferentes convoluciones iniciales (8, 16, 32, 64). Elaboración propia.....	57
Figura 3.12: Comportamiento del costo en el conjunto de validación durante el entrenamiento del modelo con diferentes funciones de activación. Elaboración propia.....	58
Figura 3.13: Comportamiento del IoU en el conjunto de validación durante el entrenamiento del modelo con diferentes funciones de activación. Elaboración propia.....	58
Figura 3.14: Comportamiento del costo en el conjunto de validación durante el entrenamiento del modelo con diferentes optimizadores. Elaboración propia.....	59
Figura 3.15: Comportamiento del IoU en el conjunto de validación durante el entrenamiento del modelo con diferentes optimizadores. Elaboración propia.....	59
Figura 3.16: Función de pérdida durante el entrenamiento de una época variando la tasa de aprendizaje en cada iteración. Se utiliza escala logarítmica en el eje X para facilitar su visualización. Elaboración propia.....	60
Figura 3.17: Ejemplo real de como varía la tasa de aprendizaje a lo largo de todo el entrenamiento usando el método de "One Cycle Learning Rate". Elaboración propia.....	61
Figura 3.18: Comportamiento del costo en el conjunto de validación durante el entrenamiento del modelo. La línea gris representa el costo obtenido al utilizar una tasa de aprendizaje estático de 0.01, mientras que la línea azul corresponde al costo obtenido al aplicar el método de "One Cycle Learning Rate" (One Cycle LR). Elaboración propia.....	62
Figura 3.19: Comportamiento del IoU en conjunto de validación durante el entrenamiento del modelo. La línea gris representa el costo obtenido al utilizar un learning rate estático de 0.01, mientras que la línea azul corresponde al IoU obtenido al aplicar el método de "One Cycle Learning Rate" (One Cycle LR). Elaboración propia.....	62
Figura 4.1: Diseño de la arquitectura U-Net final. Modificada de Ronneberger et al., (2015).....	66
Figura 4.2: IoU y coeficiente Dice en los conjuntos de entrenamiento y prueba para 0, 1 y 2 aumentos de datos. Elaboración propia.....	67
Figura 4.3: Pérdida final en el conjunto de entrenamiento para 0, 1 y 2 aumentos de datos. Elaboración propia.....	67
Figura 4.4: Ejemplos de segmentación utilizando imágenes del conjunto de prueba, tanto sin aumentos de datos como con uno y dos aumentos de datos. La segmentación verdadera se	

representa en verde, mientras que la segmentación predicha por el modelo se muestra en azul. Elaboración propia.....	68
Figura 4.5: Ejemplos de segmentación utilizando imágenes que no pertenecen al conjunto de datos, tanto sin aumentos de datos como con uno y dos aumentos de datos. En verde se representa la segmentación predicha por el modelo. Elaboración propia.....	69
Figura 4.6: Comportamiento de la Intersección sobre la Union (IoU) del modelo final en el conjunto de entrenamiento y validación para cada época. Elaboración propia.....	70
Figura 4.7: Comportamiento de coeficiente Dice del modelo final en el conjunto de entrenamiento y validación para cada época. Elaboración propia.....	71
Figura 4.8: Comportamiento de la precisión del modelo final en el conjunto de entrenamiento y validación para cada época. Elaboración propia.....	71
Figura 4.9: Matriz de confusión de los resultados de la segmentación a nivel pixel en el conjunto de datos de prueba. Elaboración propia.....	73
Figura 4.10: Ejemplo de segmentaciones predichas por el modelo (azul), y segmentación manual (verde) de 20 imágenes aleatorias del conjunto de prueba. Elaboración propia.....	75

Índice de tablas

Tabla 1.1: Tabla comparativa del estado del arte de la segmentación de columnas de humo.....	11
Tabla 2.1: Bandas del sensor ABI, resolución espectral, espacial y radiométrica. Fuente: NASA y NOAA.....	17
Tabla 3.1: Incendios seleccionados para la creación del conjunto de datos, donde se observa la fecha, el rango de horario donde se observó la columna de humo en horario UTC (una imagen cada 5 minutos), las coordenadas geoestacionarias (X, Y) del vértice inferior izquierdo de cada incendio, así como el nombre del lugar, ciudad o población más cercana al incendio.....	40
Tabla 4.1: Resultados del entrenamiento con cero, con uno y con dos aumentos de datos.....	62
Tabla 4.2: Métricas del modelo final usando un aumento de datos, se muestra el IoU, coeficiente Dice y la precisión en los conjuntos de datos de entrenamiento, validación y prueba.....	67

Acrónimos y abreviaturas

ABI- Advanced Baseline Imager	MODIS- Moderate Resolution Imaging Spectroradiometer
CONAFOR- Comisión Nacional Forestal	NASA- National Aeronautics and Space Administration
CMIP- Cloud and Moisture Imagery Product	NetCDF- Network Common Data Form
CNN- Convolutional Neural Network	NOAA- National Oceanic and Atmospheric Administration
ELU- Exponential Linear Unit	NRT- Near Real Time
EPOC- Enfermedad pulmonar obstructiva crónica	RAMMB- Regional and Mesoscale Meteorology Branch
FIRMS- Fire Information for Resource Management System	ReLU- Rectified Lineal Unit
FRP- Fire Radiative Power	SPPIF- Sistema de Predicción de Peligro de Incendios Forestales
FCN- Fully Convolutional Network	SUVI- Solar Ultraviolet Imager
GLM- Geostationary Lightning Mapper	SEISS- Space Environment In-Situ Suite
GPU- Graphics Processing Unit	SGD- Stochastic Gradient Descent
GOES- Geostationary Operational Environmental Satellite	UNEP- United Nations Environment Programme
HMS- Hazard Mapping System	UTC- Coordinated Universal Time
IA – Inteligencia Artificial	VIIRS- Visible Infrared Imaging Radiometer Suite
IoU- Intersection over Union	
MAG – Magnetometer	

Capítulo 1. Introducción

1.1. Contexto y justificación

Se entiende por un incendio al fuego que se propaga y que se sale del control del ser humano. Cuando este fuego ocurre en la naturaleza y quema vegetación natural como bosques, matorrales y pastizales, se le llama incendio forestal (Pausas, 2020).

Los incendios forestales han ocurrido de manera histórica y continúan incrementándose en diversas regiones del mundo (Jones et al., 2022; Reilly et al., 2022), en gran medida, por consecuencia de los patrones climáticos, especialmente durante períodos de sequía. La aparición del fuego a menudo se debe a causas naturales como tormentas eléctricas y erupciones volcánicas, no obstante, la estabilidad del ecosistema se ha visto afectada progresivamente por la actividad humana, la cual ha intensificado su intervención sobre los recursos naturales renovables (Castillo et al., 2003).

Esto mismo menciona el último informe del Programa de las Naciones Unidas para el Medio Ambiente, (UNEP, por sus siglas en inglés) que lleva por título “Propagándose como un reguero de pólvora: La creciente amenaza de incendios forestales extraordinarios, 2022”, en el cual, se destaca el aumento significativo de incendios forestales a gran escala en todo el mundo. El informe señala que la actividad humana, el cambio climático y otros factores, han contribuido al aumento de la frecuencia y la intensidad de los incendios forestales que amenazan la biodiversidad y la estabilidad del ecosistema global, inclusive en lugares donde antes no eran frecuentes o comunes los incendios, como en el Ártico. El informe destaca la importancia de la cooperación internacional y la implementación de medidas efectivas de manejo y prevención de incendios forestales para proteger la salud de los ecosistemas y garantizar la sostenibilidad a largo plazo (Sullivan et al., 2022).

Los incendios forestales presentan dos elementos de riesgo principales: el fuego y el humo. Las columnas de humo, generadas como subproducto de la combustión de la vegetación y residuos en ecosistemas forestales, selvas y zonas de vegetación xerófila, son un indicador inequívoco de la presencia de incendios forestales. Durante períodos de alta frecuencia de incendios, como el ocurrido en Australia de 2019- 2020, los efectos negativos del humo sobre el medio ambiente y la salud humana fueron más significativos (Borchers Arriagada et al., 2020; Van der Velde et al., 2021; Ward et al., 2020). En situaciones de extrema sequía, la carga atmosférica de humo puede favorecer la generación de rayos, lo que puede dar lugar a nuevos incendios (Contreras-Moctezuma et al., 2003).

Este humo resultante de los incendios forestales puede afectar significativamente la calidad del aire en áreas urbanas y rurales cercanas a las zonas siniestradas, especialmente en poblaciones vulnerables como niños y ancianos, aumentando el riesgo de enfermedades respiratorias y generando costos económicos significativos para el tratamiento médico y el combate los de incendios (Contreras-Moctezuma et al., 2003). La probabilidad de sufrir enfermedades respiratorias y los costos económicos asociados con la exposición al humo de los incendios forestales aumentan significativamente si no se identifican y combaten de manera oportuna y eficaz. La respuesta temprana y coordinada puede contribuir a reducir los impactos negativos de los incendios forestales en la salud pública, así como en la economía y el medio ambiente.

Según diversos estudios (Chen et al., 2021; Lara et al., 2022; Sanderfoot et al., 2022; Selimovic et al., 2019), se indica que los principales gases emitidos por el humo de los incendios forestales son monóxido de carbono (CO), dióxido de carbono (CO₂), óxidos de nitrógeno (NO_x) y óxidos de azufre (SO_x).

El dióxido de carbono (CO₂) y el monóxido de carbono (CO) son gases incoloros e inodoros que se producen durante la combustión de materiales orgánicos, como la madera y los combustibles fósiles. Sin embargo, tienen diferentes efectos en la salud humana. El monóxido de carbono es tóxico para los seres humanos en concentraciones relativamente bajas¹ y puede provocar envenenamiento y la muerte en casos graves, ya que se une a la hemoglobina en la sangre y reduce la cantidad de oxígeno que puede transportar el cuerpo. Por otro lado, el dióxido de carbono no es tóxico en concentraciones bajas y se produce naturalmente en el cuerpo humano como un producto residual del metabolismo celular (US EPA, 2015).

En términos del impacto ambiental, el dióxido de carbono es uno de los principales gases de efecto invernadero que contribuyen al calentamiento global y el cambio climático, mientras que el monóxido de carbono es un contaminante atmosférico que afecta la calidad del aire y la salud respiratoria. Según estudios recientes, los niveles de CO₂ en la atmósfera terrestre han aumentado a niveles sin precedentes en los últimos 800,000 años, principalmente debido a la actividad humana, como la quema de combustibles fósiles y la deforestación. Este aumento en la concentración de CO₂ en la atmósfera es considerado uno de los principales factores que contribuyen al cambio climático y a los impactos negativos en la salud humana (U.S. Environmental Protection Agency, 2016).

Los óxidos de nitrógeno (NO_x) son gases tóxicos que se emiten como subproducto de la combustión de combustibles fósiles en vehículos, plantas de energía y procesos industriales, así

1 Según la Norma Oficial Mexicana NOM-021-SSA1-1993: "La concentración de monóxido de carbono, como contaminante atmosférico, no debe rebasar el valor permisible de 11.00 ppm o lo que es equivalente a 12,595 µg/m³ en promedio móvil de ocho horas una vez al año, como protección a la salud de la población susceptible".

como en incendios forestales, ya que se liberan a partir de la combustión incompleta de material vegetal y otros combustibles. Los NO_x se componen de óxido nítrico (NO) y dióxido de nitrógeno (NO_2). La exposición a los óxidos de nitrógeno puede tener efectos adversos para la salud, especialmente en personas con enfermedades respiratorias preexistentes. Los NO_x pueden irritar las vías respiratorias, causar tos, dificultad para respirar, y empeorar enfermedades como el asma y la enfermedad pulmonar obstructiva crónica (EPOC) (U.S. Environmental Protection Agency, 2023). Además, los NO_x también contribuyen al calentamiento global al actuar como gases de efecto invernadero, lo que aumenta la temperatura de la Tierra y causa cambios climáticos. Por lo tanto, reducir las emisiones de NO_x puede tener beneficios tanto para la salud como para el medio ambiente.

Los óxidos de azufre (SO_x) son gases tóxicos compuestos de dióxido de azufre (SO_2) y trióxido de azufre (SO_3) y también se emiten a la atmósfera como resultado de la combustión de combustibles fósiles y procesos industriales. Estos gases pueden reaccionar con otros compuestos químicos en la atmósfera y formar partículas finas, conocidas como material particulado (World Health Organization, 2023), que pueden ser inhaladas por las personas y provocar una serie de efectos adversos para la salud, incluyendo irritación de las vías respiratorias, exacerbación del asma, bronquitis crónica, enfermedades cardiovasculares y muerte prematura. El tamaño y la composición química de las partículas son factores clave en la gravedad de los efectos en la salud. Partículas más pequeñas, como las $\text{PM}_{2.5}$, son capaces de penetrar en los pulmones y el torrente sanguíneo y son consideradas más peligrosas que las partículas más grandes (Kim et al., 2015). Los SO_x también contribuyen a la formación de lluvia ácida, la cual puede tener efectos negativos en la calidad del suelo, el agua y los ecosistemas naturales (Y. Zhang et al., 2019).

En síntesis, resulta innegable la magnitud de la gravedad del humo producido por los incendios forestales, ya que no solo representa un riesgo para la salud humana, sino también para el medio ambiente. Por lo tanto, es crucial obtener información precisa y en el tiempo oportuno sobre la ubicación de las columnas de humo para poder combatir los incendios y así mismo medir su impacto. En consecuencia, se debe buscar una mejora en la tecnología de monitoreo y alerta temprana para detectar y responder rápidamente a los incendios forestales y minimizar su impacto en la salud y el medio ambiente.

Según los hallazgos del estudio titulado “Estimated global mortality attributable to smoke from landscape fires”, proporciona una estimación del número de muertes prematuras en todo el mundo atribuibles al humo producido por incendios forestales en el año 2012. Johnston y colaboradores utilizaron un modelo estadístico para estimar la exposición al humo y su relación con la mortalidad en diferentes regiones del mundo. Los resultados indican que el humo de

incendios forestales fue responsable de aproximadamente 339,000 muertes prematuras en 2012, la mayoría de ellas en Asia y África (Johnston et al., 2012).

Los avances tecnológicos en la observación de la tierra a través de satélites, junto con la creciente cantidad de datos y los progresos en la percepción remota y la visión por computadora a través de la inteligencia artificial y el aprendizaje automático, están mejorando significativamente la capacidad para obtener información precisa y en el tiempo oportuno. Estas mejoras son cruciales para la toma de decisiones eficaces en la gestión de incendios forestales y otros desastres naturales.

El proceso de segmentación semántica¹ de una imagen puede ser realizado manualmente por un experto en la materia, pero esta tarea es tediosa, costosa y puede llevar mucho tiempo. Por lo tanto, cada vez se utiliza más el aprendizaje profundo, también conocido como *deep learning* en inglés, para efectuar la segmentación semántica de imágenes de forma automática.

El uso de estos modelos tiene varias ventajas sobre el enfoque manual. En primer lugar, son mucho más rápidos que la segmentación manual, tienen la capacidad de procesar grandes cantidades de información en muy poco tiempo. Esto se debe en gran medida a la gran capacidad de cómputo que existe actualmente, en segundo lugar, la segmentación de imágenes con modelos de aprendizaje profundo es menos costosa que la segmentación manual. La segmentación manual requiere la contratación de expertos en la materia, y el proceso puede llevar días o incluso semanas para completarse. Por el contrario, los modelos de aprendizaje profundo solo requieren tiempo y recursos para entrenarse y optimizarse. En tercero, los modelos de aprendizaje profundo tienen el potencial de ser más precisos que la segmentación manual, ya que pueden ayudar a mitigar ciertos aspectos del error humano; Cuando nos referimos a "error humano" en la segmentación de imágenes, estamos haciendo referencia a la posibilidad de que los expertos en la materia puedan cometer errores debido a la subjetividad y la fatiga, entre otros factores. Por ejemplo, el experto puede tener sesgos o prejuicios que afecten su juicio al realizar la segmentación, o puede experimentar fatiga después de muchas horas de trabajo, lo que puede disminuir su capacidad de atención y afectar su precisión en el proceso de segmentación. No obstante, es importante destacar que el proceso de selección de datos y etiquetado, en el caso del aprendizaje supervisado, también involucra a personas expertas y, por lo tanto, puede ser susceptible a errores en la recopilación y etiquetado de datos.

Por último, el uso de modelos de aprendizaje profundo para la segmentación de imágenes puede ser más cercano a tiempo real que la segmentación manual. El concepto de "tiempo real"

1 Tarea de cortar o dividir una imagen a nivel pixel y asignar una etiqueta de clase a cada píxel de dicha imagen (ver 2.2.1. Segmentación semántica).

o “casi real” es inherentemente subjetivo y, en gran medida, está condicionado por la aplicación específica en cuestión.

En el contexto de este estudio, cuando hacemos referencia a tiempo casi real, nos referimos a la obtención de la imagen segmentada de la columna de humo lo más cercano posible a la hora de la captura de la imagen. Por ejemplo, si consideramos el uso de imágenes obtenidas por el satélite GOES-16 en la escena de escaneo de disco completo, el tiempo que el sensor requiere para escanear la escena completa es aproximadamente de 9 minutos, y estas imágenes se ponen a disposición del público en general en el transcurso del siguiente minuto en nivel de procesamiento L2. A esto, debemos agregar el tiempo necesario para descargar y procesar la imagen, el cual varía en función de la velocidad de la conexión a internet y las capacidades del servidor utilizado. Idealmente, un modelo de segmentación previamente entrenado puede llevar a cabo la segmentación de una nueva imagen en cuestión de segundos. Por lo tanto, en el mejor de los escenarios y con un modelo óptimo, podríamos estar hablando, de manera aproximada, de entre 10 a 15 minutos después del inicio de la toma de la imagen o dentro de los próximos 5 minutos después de la publicación de la misma, para obtener la segmentación con las columnas de humo identificadas. Es importante destacar que la etapa de segmentación en sí misma constituye la menor parte del tiempo requerido, ya que la mayor parte de este intervalo temporal se destina a la toma, el procesamiento y descarga de la imagen.

En resumen, el uso de modelos de aprendizaje profundo para la segmentación de imágenes es una opción más rápida, económica y precisa que la segmentación manual. Además, los modelos de aprendizaje profundo realizan la segmentación en tiempo casi-real, lo que lo convierte en una opción muy adecuada para muchas aplicaciones como lo es el caso de segmentar columnas de humo provenientes de incendios forestales.

1.2. Objetivo general

Crear un modelo de segmentación semántica de columnas de humo provocadas por incendios forestales para México usando imágenes satelitales GOES-16.

1.2.1. Objetivos Particulares

1. Analizar cuáles bandas, productos y compuestos generados por las imágenes del satélite GOES-16 nos aportan información útil acerca de las columnas de humo.
2. Generar un conjunto de datos de columnas de humo segmentadas a mano para el entrenamiento y validación del modelo.
3. Explorar soluciones basadas en visión por computadora y adaptar una estructura de Aprendizaje Profundo para abordar el problema de la segmentación semántica.

1.3. Hipótesis

- Utilizando un modelo de visión por computadora y aprendizaje profundo, es factible segmentar y detectar de manera precisa las columnas de humo de un incendio forestal a partir de imágenes obtenidas por satélites meteorológicos en órbita geoestacionaria, con una alta resolución temporal, a pesar de la limitada resolución espacial de estas imágenes.

1.4. Antecedentes y trabajos relacionados

El objetivo de la sección de antecedentes y trabajos relacionados es proporcionar un contexto más amplio para el trabajo propuesto y revisar las investigaciones previas relacionadas con la clasificación y segmentación semántica de imágenes de columnas de humo. A continuación, se presentan algunos trabajos previos relevantes sobre segmentación semántica de imágenes de humo y detección de incendios forestales que ejemplifican como se ha venido trabajando este tema a lo largo del tiempo. Un resumen de los artículos mencionados en este capítulo se puede consultar en la Tabla 1.1.

Uno de los primeros acercamientos al análisis de las columnas de humo con imágenes satelitales fue por el trabajo presentado por Xie et al., (2007), donde se describe un método para detectar y monitorear columnas de humo utilizando datos del espectrorradiómetro de imágenes de resolución moderada (MODIS) a bordo de los satélites Terra y Aqua de la NASA, así como múltiples umbrales para identificar anomalías térmicas y espectrales de diferentes tipos de cobertura en las imágenes satelitales y filtrar los píxeles hasta que queden solamente los píxeles que estén asociados al humo. Además, el estudio también destaca la importancia de usar múltiples canales de las imágenes de satélite y datos de imágenes en tiempo real para mejorar la precisión de la detección de columnas de humo.

Sin embargo, el estudio también menciona algunas deficiencias de este tipo de algoritmos, como la dificultad para distinguir entre columnas de humo y nubes, que puede verse afectado por la presencia de otros factores que causan anomalías térmicas y espectrales, como la contaminación y la actividad industrial. A pesar de que los umbrales fijos para bandas múltiples pueden ser válidos en regiones locales y para un determinado periodo de tiempo, es difícil determinar umbrales óptimos globales debido a las variaciones en la reflectancia y la temperatura de brillo de distintas latitudes y estaciones del año.

Varios años más tarde, y con el auge del aprendizaje profundo que se produjo aproximadamente a partir de 2012, cuando el equipo de Geoffrey Hinton, ganó el desafío de reconocimiento de objetos en imágenes utilizando la base de datos de ImageNet y una Red Neuronal Convolucional (CNN, por sus siglas en inglés) profunda llamada "AlexNet" (Krizhevsky et al., 2012), se han publicado trabajos como el de Luo et al., (2018), que propone un algoritmo de detección de humo de incendios basado en características de movimiento y redes neuronales convolucionales. El enfoque se basa en la observación de que el humo de los incendios suele tener una dirección de movimiento diferente a la del fondo de la imagen, lo que permite distinguirlo de otros objetos en movimiento.

El algoritmo emplea características de movimiento para detectar las áreas de la imagen donde se encuentra el humo y luego usa una CNN para clasificar si estas áreas contienen humo o no. El enfoque ha sido evaluado en un conjunto de datos de imágenes de video de incendios, logrando una precisión¹ de 0.99 en el conjunto de entrenamiento, sin embargo, una de las razones por las que este enfoque no se puede aplicar directamente a imágenes satelitales es debido a la resolución espacial y temporal de los datos. A pesar de que algunos satélites proporcionan resoluciones temporales aceptables, la perspectiva del humo en imágenes satelitales es diferente, y puede generar muchos errores de comisión² debido al movimiento del humo en la misma dirección que las nubes.

La siguiente aproximación al problema fue desde el enfoque de la clasificación de imágenes que contenían humo de las que no, un ejemplo de esto es el presentado en el artículo de Ba et al., (2019), donde los autores crearon un conjunto de datos, el cual consta de 6225 imágenes satelitales de seis clases diferentes, incluyendo nube, polvo, neblina, tierra, mar y humo, dichas imágenes abarcan varias áreas y regiones del mundo. Proponen un nuevo modelo de red neuronal convolucional llamado SmokeNet, que logra una precisión de 0.9275.

Es importante destacar que el modelo presentado se enfoca en el problema de clasificación de imágenes de humo en imágenes satelitales y no aborda la tarea de segmentación de columnas de humo. En otras palabras, SmokeNet no identifica la ubicación exacta de las columnas de humo en la imagen, sino que solo clasifica la imagen completa en nube, polvo, neblina, tierra, mar o humo.

Con esto damos paso al problema de la segmentación semántica, Mommert et al., (2020) se enfocan en la detección y cuantificación de columnas de humo industriales a partir de datos de imágenes multibanda utilizando una combinación de ResNet-50 y U-net. Emplean imágenes de Sentinel-2 de varias regiones de la Unión Europea. Las métricas de precisión de 0.943 y una intersección sobre la unión (IoU³, por sus siglas en inglés) de 0.608 resaltan la eficacia de su metodología en la identificación precisa de humo industrial.

Frizzi et al., (2021) abordan la detección y localización de humo y fuego en tiempo casi real empleando imágenes de cámaras a bordo de drones o helicópteros. Su arquitectura, basada en el modelo de red neuronal VGG16, logra altas métricas de precisión (0.928) e IoU (0.858) en la

1 La precisión representa la proporción de predicciones correctas entre todas las predicciones totales (ver 2.2.6. Métricas de evaluación).

2 Los errores de comisión es cuando se predice incorrectamente una clase positiva.

3 La intersección sobre la unión (IoU) es una métrica que se emplea comúnmente en problemas de segmentación y clasificación, especialmente cuando se trabaja con conjuntos de datos binarios (ver 2.2.6. Métricas de evaluación).

segmentación semántica. Su enfoque destaca la importancia de la integración de sistemas de vigilancia en tiempo real para la detección temprana de humo y fuego.

Larsen et al., (2021) usan una red neuronal totalmente convolucional (FCN, *Fully Convolutional Network*) para detectar y segmentar el humo de incendios en imágenes satelitales en el norte de Australia, que se vio afectado por una gran cantidad de humo de incendios forestales entre el 1 y el 21 de septiembre de 2015. Se utilizaron seis bandas espectrales de Himawari-8, incluyendo bandas de espectros visibles e infrarrojos, y el FRP¹ de los puntos de calor asociados a incendios se utilizó como un séptimo canal en la matriz de entrada. Para generar los datos de entrenamiento se empleó un algoritmo analítico de enmascaramiento de nubes (Qin et al., 2015). La precisión de segmentación promedio fue de 0.995 y una IoU de 0.576, con una tasa de verdaderos positivos de 22.8% y una tasa de verdaderos negativos de 99.5%. Sin embargo, la mayoría de los errores de clasificación se producen a lo largo de los bordes de la columna, lo que puede ser mejorado en futuras investigaciones agregando dependencias temporales y variabilidad temporal en las columnas de humo.

Otro artículo relacionado con la segmentación semántica de humo de incendios forestales es el de Wen & Burke, (2021), en este trabajo se emplea una red neuronal totalmente convolucional llamada U-Net para segmentar columnas de humo en imágenes de satélite geoestacionarias. Para entrenar el modelo, se utilizan observaciones satelitales como entrada y columnas de humo anotadas a mano por el HMS² como etiquetas objetivo. Las imágenes usadas en este estudio son de 2018 y provienen de California y Nevada durante la temporada de incendios forestales entre mayo y diciembre. Además, se incluyen múltiples bandas espectrales en experimentos posteriores para mejorar los resultados. Con un coeficiente Dice³ de 0.265, sugieren que los métodos de segmentación automatizados son comparables cualitativamente a las columnas de humo anotadas a mano. Así mismo, se menciona que en futuros trabajos, se podría mejorar la robustez del modelo frente a etiquetas ruidosas y extender el método para identificar el humo en todo el mundo.

En 2022, Wang et al., (2022) abordaron la segmentación semántica y el análisis de parámetros sensibles al humo de incendios forestales usando imágenes de Landsat-8. Su enfoque, basado en la arquitectura Smoke-unet derivada de U-Net, busca segmentar y analizar el humo en áreas forestales. Los resultados obtenidos destacan la eficacia de su enfoque en la identificación

1 FRP (*Fire Radiative Power*): Esta es una medida que se utiliza para cuantificar la energía radiada por los incendios forestales y otros tipos de incendios.

2 HMS (*Hazard Mapping System*): Es un sistema de monitoreo de incendios y humo con procesamiento directo casi en tiempo real (ver 1.5. Que se hace actualmente)

3 El coeficiente Dice, también conocido como Coeficiente de Sørensen-Dice, es una medida empleada comúnmente para comparar la similitud entre dos conjuntos (ver 2.2.6. Métricas de evaluación).

precisa de humo en imágenes de satélite, como se refleja en las métricas de precisión (0.725) e IoU (0.644). Estas cifras validan la capacidad de su modelo para capturar la complejidad de las áreas afectadas por incendios y resaltan su contribución al avance de la detección y análisis de humo en imágenes de satélite en el contexto de incendios forestales.

También existen diversos estudios que trabajan con columnas de humo de otro origen como el de centrales eléctricas, en Dumont Le Brazidec et al., (2023) proponen una técnica de segmentación de imágenes de CO₂ usando una U-Net modificada, reemplazando la parte de codificador con una EfficientNetB0. Se centra en segmentar plumas de CO₂ de centrales eléctricas a partir de datos del programa Copernicus.

En términos generales, la arquitectura U-Net ha sido ampliamente utilizada en aplicaciones de procesamiento de imágenes biomédicas, como la segmentación de tumores en imágenes de resonancia magnética (Ghosh et al., 2021), la segmentación de células en imágenes de microscopía (Long, 2020) y la segmentación de núcleos en imágenes de histología (Mahbod et al., 2022), entre otras. Así mismo, se ha demostrado que es muy eficiente para la segmentación de otro tipo de imágenes, como lo son las imágenes satelitales de columnas de humo de incendios forestales. Los resultados han demostrado que la arquitectura U-Net supera a los enfoques existentes en términos de precisión y eficiencia computacional en muchos casos.

1.4. Antecedentes y trabajos relacionados

Autor	Título	Año	Objetivo resumido	Metodología	Datos	Área de estudio	Métricas
Dumont Le Brazidec et al.	Segmentation of XCO2 images with deep learning: application to synthetic plumes from cities and power plants.	2023	Segmentar plumas de CO2 de centrales eléctricas. (segmentación semántica)	U-net modificada, con EfficientNetB0 para el codificador.	Copernicus program.	París y Berlín.	Precisión 0.755
Wang et al.	Semantic Segmentation and Analysis on Sensitive Parameters of Forest Fire Smoke Using Smoke-Unet and Landsat-8 Imagery.	2022	Analizar parámetros y segmentar humo de incendios forestales. (Segmentación semántica)	Smoke-unet, basado en U-net.	Landsat-8.	Varias regiones del mundo.	Precisión 0.725 IoU 0.644
Wen & Burke	Wildfire smoke plume segmentation using geostationary satellite imagery.	2021	Segmentar columnas de humo a partir de imágenes de satélites geoestacionarios. (Segmentación semántica)	U-net	GOES-16 y anotaciones del HMS.	California y Nevada.	Dice 0.265
Larsen et al.	A deep learning approach to identify smoke plumes in satellite imagery in near-real time for health risk communication.	2021	Predecir el humo de incendios en imágenes satelitales en tiempo casi real (NRT). (Detección de objetos y segmentación semántica)	Red Neuronal Totalmente Convolutiva (FCN).	Himawari-8 y potencia radiativa del fuego (FRP).	Australia.	Precisión 0.995 IoU = 0.576
Frizzi et al.	Convolutional neural network for smoke and fire semantic segmentation.	2021	Detectar y localizar humo y fuego en tiempo casi real con una cámara integrada en un vehículo (Segmentación semántica)	Nueva arquitectura basada en VGG16.	Imágenes de cámaras a bordo de un dron o helicóptero.	--	Precisión 0.925 IoU 0.858
Mommert et al.	Characterization of Industrial Smoke Plumes from Remote Sensing Data	2020	Detectar y cuantificar columnas de humo industriales a partir de datos de imágenes multibanda. (Detección de objetos y segmentación semántica)	ResNet-50 para detección + U-net para segmentación.	Sentinel-2	Unión Europea.	Precisión 0.943 IoU 0.608
Ba et al.	SmokeNet: Satellite Smoke Scene Detection Using Convolutional Neural Network with Spatial and Channel-Wise Attention.	2019	Clasificar de imágenes de humo de otras clases. (Clasificación)	Red Neuronal Convolutiva (CNN).	MODIS.	Varias regiones del mundo.	Precisión 0.9275
Luo et al.	Fire smoke detection algorithm based on motion characteristic and convolutional neural networks.	2018	Detectar humo de incendios basado en video. (Detección de objetos)	Red Neuronal Totalmente Convolutiva (FCN).	Imágenes de video-vigilancia.	--	Precisión 0.99 en el entrenamiento
Xie et al.	Smoke plume detection in the eastern United States using MODIS.	2007	Detectar columnas de humo basado en el análisis de las características espectrales. (Segmentación)	Algoritmo multiumbral	MODIS.	Este de los Estados Unidos	Cualitativa, pocas omisiones.

Tabla 1.1: Tabla comparativa del estado del arte de la segmentación de columnas de humo.

1.5. Que se hace actualmente

Actualmente, en México, se utiliza la percepción remota para monitorear, detectar y predecir incendios forestales. Uno de los ejemplos más relevantes es el Sistema de Predicción de Peligro de Incendios Forestales (SPPIF) de La Comisión Nacional Forestal (CONAFOR). Este sistema permite evaluar los conglomerados de puntos de calor e incendios forestales activos y mapear en tiempo casi real la superficie aproximada de los mismos a partir de los perímetros de

conglomerados de puntos de calor. El sistema mapea diariamente las condiciones de sequedad del combustible, el peligro de ignición y el peligro de incendio forestal, así como el número de incendios forestales esperados por estado. Los mapas se actualizan diariamente con base en información meteorológica, incendios y conglomerados de puntos de calor activos (Vega Nieva, 2020).

Aunque el Sistema de Monitoreo y Alerta Temprana de Incendios Forestales en México (SPPIF) ha sido una herramienta efectiva en la prevención y combate de incendios forestales, aún no incluye información precisa acerca de la ubicación y comportamiento de las columnas de humo generadas por dichos incendios. La falta de esta información se debe, en gran parte a la complejidad de la tarea de segmentación de las columnas de humo, como se ha evidenciado en la literatura. Sin embargo, la inclusión de esta información en el SPPIF podría resultar de gran utilidad y fortalecer significativamente el sistema.

De manera global, uno de los sistemas operativos más importantes en la segmentación semántica de humo de incendios es el *Hazard Mapping System Fire and Smoke Product* (HMS), es un sistema desarrollado por la Agencia Nacional Oceánica y Atmosférica (NOAA) de los Estados Unidos para proporcionar información sobre la ubicación, la intensidad y la dirección del humo y los incendios forestales en tiempo casi real. El sistema también proporciona información sobre la calidad del aire y el impacto del humo en la salud humana y la visibilidad (McNamara et al., 2004).

El análisis de humo del HMS se realiza mediante la clasificación manual y visual de las columnas de humo usando imágenes del sensor ABI¹ de GOES²-16 y GOES-17 en un compuesto de color verdadero, disponibles durante la parte del día iluminada por el sol. El primer análisis de humo para el día actual generalmente se produce alrededor del mediodía local. Los atributos de humo incluyen las horas de inicio y finalización de la secuencia de imágenes que se empleó para segmentar, el satélite correspondiente y la densidad de la pluma, que se etiqueta cualitativamente como ligera, media y pesada según el grosor aparente del humo en las imágenes de satélite. De acuerdo a lo que se menciona, la clasificación visual es necesaria, ya que la segmentación automática de las columnas de humo en las imágenes satelitales puede ser difícil debido a la variabilidad en la apariencia y el movimiento del humo (*NOAA HMS Smoke Detection*, 2023). Estos datos han sido utilizados como datos de entrenamiento en algunos artículos, como en el caso de Wen & Burke, (2021). La cuestión radica en que las anotaciones o

-
- 1 ABI (*Advanced Baseline Imager*): Es el instrumento principal de la serie GOES-R para obtener imágenes multiespectrales de la tierra.
 - 2 GOES (*Geostationary Operational Environmental Satellite*): Es un programa satelital colaborativo de la NOAA y la NASA que proporciona imágenes y datos continuos sobre las condiciones atmosféricas de la tierra y la actividad solar (ver mas en 2.1. GOES-16).

segmentaciones manuales que se llevan a cabo en este sistema son imágenes tomadas en diferentes momentos del día. Por lo tanto, la segmentación de una columna de humo en particular no se refiere a un momento específico, sino a una síntesis temporal. Este enfoque plantea un problema significativo al intentar vincular la máscara de humo a una imagen de un momento específico, lo que podría conllevar un entrenamiento deficiente del modelo.

Capítulo 2. Marco teórico

2.1. GOES-16

GOES-16 o GOES-West (*Geostationary Operational Environmental Satellite*) es un satélite meteorológico geoestacionario operado por la Administración Nacional Oceánica y Atmosférica (NOAA) de los Estados Unidos. Fue lanzado al espacio en noviembre de 2016 y es el primer satélite de la serie GOES-R¹, cubre principalmente la región de América del Norte, Centroamérica y Sudamérica, así como el Océano Atlántico hasta la costa occidental de África. La región de cobertura de GOES-16 se extiende desde aproximadamente 0 grados de longitud hasta 150 grados de longitud oeste y desde el Círculo Ártico hasta el extremo sur de América del Sur, lo que incluye gran parte de los Estados Unidos, México, Centroamérica, Sudamérica y el Océano Atlántico (ver Figura 2.1).

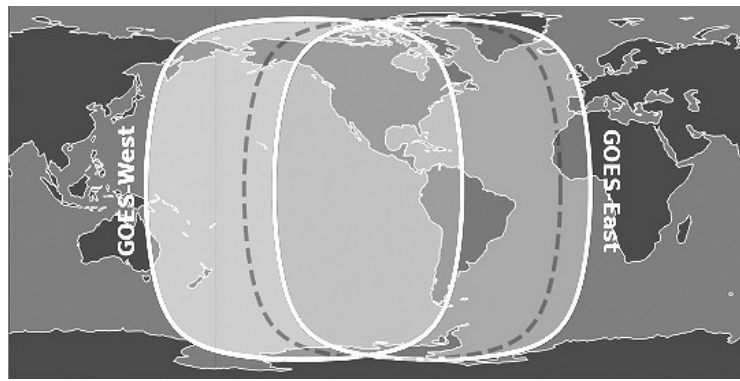


Figura 2.1: Flota operativa de GOES a abril de 2023, GOES-East (16) y GOES-West (17). Modificada de Blaylock, (2023).

GOES-16 se encuentra en una órbita geoestacionaria, lo que significa que se mueve a la misma velocidad que la rotación de la Tierra y permanece en la misma posición sobre el ecuador. Esta órbita se encuentra aproximadamente a 35,786 kilómetros sobre la superficie terrestre. Esta altitud es crucial, ya que permite que el satélite GOES-16 cubra todo el disco terrestre visible desde su posición, abarcando un hemisferio completo en cada imagen. Esto le permite monitorear continuamente el clima en su área de cobertura y tener una alta resolución temporal. También lleva una variedad de instrumentos que están diseñados para proporcionar

1 La serie GOES-R es un programa que consiste de dos naves espaciales, GOES-R (16) y GOES-S (17) y dos naves espaciales adicionales que serán lanzadas próximamente (GOES-T y GOES-U) y que pasaran a llamarse GOES-18 y GOES-19 una vez que alcancen su órbita final.

observaciones precisas y detalladas del clima y el medio ambiente en su área de cobertura (Dirk Krebs, 2020). Los principales instrumentos que porta son:

1. *Advanced Baseline Imager (ABI)*: Es un sensor multiespectral de alta resolución que puede tomar imágenes detalladas de la Tierra en 16 bandas espectrales diferentes, desde el visible hasta el infrarrojo térmico. El ABI proporciona imágenes meteorológicas de alta resolución temporal y baja resolución espacial.
2. *Geostationary Lightning Mapper (GLM)*: Es un sensor que puede detectar la ubicación y la intensidad de los rayos en tiempo real. El GLM proporciona información valiosa para la predicción de tormentas y la seguridad de la aviación.
3. *Solar Ultraviolet Imager (SUVI)*: Es un instrumento que mide la radiación ultravioleta emitida por el sol y puede proporcionar alertas tempranas de tormentas solares que pueden afectar la tecnología y las comunicaciones en la Tierra.
4. *Space Environment In-Situ Suite (SEISS)*: Es un conjunto de instrumentos que miden las partículas cargadas y el campo magnético en el espacio cercano a la Tierra. Estos datos pueden ayudar a predecir los efectos de las tormentas solares en la tecnología en la Tierra.
5. *Magnetometer (MAG)*: Es un sensor que mide el campo magnético en el espacio cercano a la Tierra, lo que puede ayudar a predecir las perturbaciones geomagnéticas y sus efectos en la tecnología en la Tierra.

De estos instrumentos el que se usa principalmente en percepción remota para la extracción de variables del terreno y atmósfera es el sensor ABI.

Los valores o unidades de pixel del sensor ABI pasan por una corrección radiométrica y geométrica, estas correcciones las realiza las instituciones correspondientes (NOAA y NASA), después de eso, los datos son puestos a disposición del público en un nivel de procesamiento “*Level 1b (L1b)*” donde la información de los píxeles está en unidades de radiancia y de estos datos obtienen una serie de productos “*Level 2 (L2)*”, entre los cuales se encuentra el producto “*Cloud and Moisture Imagery Product (CMIP)*”, que contiene las 16 bandas; en unidades de reflectancia para las bandas (1-6) y temperatura de brillo para las bandas (7-16) (Schmit et al., 2012).

2.1.1. Resolución

En percepción remota, las imágenes satelitales cuentan con 4 tipos de resolución:

1. Resolución espectral: Número de bandas y su longitud de onda en el espectro electromagnético. ABI cuenta con 16 bandas, 2 en el visible, 4 en el infrarrojo cercano y 10 en infrarrojo térmico, sus longitudes de onda centrales se puede observar en la Tabla 2.1.
2. Resolución espacial: Se refiere al área que es representada por cada píxel en la imagen. En ABI es de 0.5 km para la banda 2, de 1 km para las bandas 1, 3 y 5, y de 2 km para las bandas 4 y de la 6 a la 16.
3. Resolución radiométrica: Cantidad de niveles discretos de intensidad que el sensor puede representar. Para todas las bandas de ABI es de 12 bits, lo que implica que el sensor es capaz de representar 2^{12} , es decir, 4096 niveles discretos de intensidad.
4. Resolución temporal: Cada cuanto tiempo es capaz el sensor de tomar una imagen del mismo lugar. Como ABI está montado en un satélite geoestacionario, la escena nunca cambia, por lo tanto, la resolución temporal es alta y esta depende de la escena de escaneo. ABI cuenta con 3 distintas escenas de escaneo: FullDisk, CONUS (Contiguous U.S.) y MESO (Mesoescala) (ver Figura 2.2) y dependiendo de la escena de escaneo es su resolución temporal, siendo de cada 10 minutos para el caso de FullDisk, de cada 5 minutos para CONUS y de hasta cada 30 segundos para MESO.

Número de banda	Longitud de onda central (μm)	Resolución espacial al nadir [km]	Tipo	Nombre	Resolución radiométrica [bits]
1	0.47	1	Visible	Azul	12
2	0.64	0.5	Visible	Roja	12
3	0.87	1	Infrarrojo cercano	Veggie	12
4	1.38	2	Infrarrojo cercano	Cirrus	12
5	1.61	1	Infrarrojo cercano	Nieve/Hielo	12
6	2.25	2	Infrarrojo cercano	Tamaño de las partículas de la nube	12
7	3.89	2	Infrarrojo térmico	Ventana de onda corta	12
8	6.17	2	Infrarrojo térmico	Vapor de agua de nivel superior	12
9	6.93	2	Infrarrojo térmico	Vapor de agua de nivel medio	12
10	7.34	2	Infrarrojo térmico	Vapor de agua de nivel inferior	12
11	8.44	2	Infrarrojo térmico	Fase superior de la nube	12
12	9.61	2	Infrarrojo térmico	Ozono	12
13	10.33	2	Infrarrojo térmico	Ventana de onda larga "limpia"	12
14	11.19	2	Infrarrojo térmico	Ventana de onda larga	12
15	12.27	2	Infrarrojo térmico	Ventana de onda larga "sucia"	12
16	13.27	2	Infrarrojo térmico	Onda larga de CO ₂	12

Tabla 2.1: Bandas del sensor ABI, resolución espectral, espacial y radiométrica. Fuente: NASA y NOAA.

Para generar imágenes RGB en color verdadero se requiere de las bandas roja (0.64 μm), verde (0.55 μm) y azul (0.47 μm), pero en ABI no cuenta con la banda verde (0.55 μm), sin embargo, posee la banda de veggie (0.86 μm) que, cuando se combina proporcionalmente con las bandas roja (0.64 μm) y azul (0.47 μm), puede generar una banda similar al verde como una aproximación, teniendo en cuenta que este enfoque es una aproximación de primer orden; no reemplaza el contenido de información de la banda verde que falta (Bah et al., 2018). A través de la experimentación, la proporción que produjo resultados razonables fue:

$$\text{Falso Verde} = 0.45 \cdot \text{Roja} + 0.10 \cdot \text{Veggie} + 0.45 \cdot \text{Azul} \quad (\text{Ec. 2.1})$$

De este modo, el compuesto de color verdadero de ABI estaría dado por:

$$\text{Color Verdadero} = (\text{Banda 2}, \text{Falso Verde}, \text{Banda 1})$$

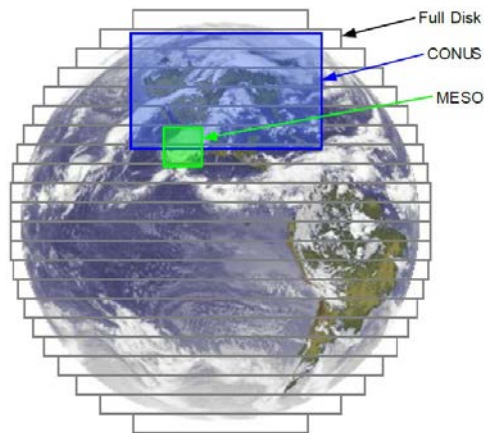


Figura 2.2: Escenas de escaneo de ABI, Tomada de Mahonchak, (2019).

2.2. Visión por computadora

Como humanos, percibimos la estructura tridimensional del mundo que nos rodea con aparente facilidad, si miramos una imagen de grupo de personas, podemos contar fácilmente (y nombrar) a todas las personas en la imagen e incluso adivinar sus emociones a partir de su apariencia facial. En la visión por computadora o también llamada visión artificial, se intenta replicar este tipo de inteligencia humana que tiene que ver con interpretar visualmente el mundo que nos rodea, es decir, describir el mundo que vemos en una o más imágenes y reconstruir sus propiedades, como la forma, la iluminación y las distribuciones de color. Es sorprendente que los humanos y los animales hagan esto sin esfuerzo, mientras que los algoritmos de visión por computadora son tan propensos a errores (Szeliski, 2011).

La visión por computadora implica la utilización de técnicas y herramientas de procesamiento de imágenes, como filtros, detección de bordes, reconocimiento de patrones, modelos de segmentación, entre otras, para extraer información útil de imágenes o videos. Algunas de las aplicaciones más comunes de la visión por computadora incluyen la detección de objetos, el reconocimiento facial, el seguimiento de movimiento, la realidad aumentada y la robótica.

En este capítulo, es importante señalar que algunos términos técnicos se utilizan de manera convencional en la literatura científica en su forma original en inglés, mientras que otros carecen de una traducción literal u oficial al español. Por lo tanto, en algunas ocasiones, haremos referencia a estos términos empleando su denominación en inglés, con el propósito de mantener

la coherencia terminológica y facilitar la comprensión en el contexto del ámbito académico y científico.

2.2.1. Segmentación semántica

La segmentación semántica es una de las herramientas o técnicas que ocupa la visión por computadora para describir una imagen. Segmentar quiere decir “cortar o partir algo en segmentos” (*Real Academia Española*, 2023). Y semántica es una rama de la lingüística dedicada al estudio del significado. Por lo tanto, podemos decir que la segmentación semántica es la tarea de cortar o dividir una imagen a nivel pixel y asignar una etiqueta de clase a cada píxel en dicha imagen. Esto involucra comprender el contenido y el contexto de una imagen a un nivel detallado.

La tarea de segmentar una imagen o lo que vemos en tiempo real es una tarea muy fácil para el ser humano; sin embargo, es bastante compleja para una computadora. Actualmente, no está claro cómo el cerebro humano encuentra la segmentación correcta. Segmentar una imagen implica una profunda comprensión semántica del mundo y qué cosas son partes de un todo (Guo et al., 2018).

La concepción de un modelo que permita la segmentación de imágenes en clases semánticas específicas, presenta una diversidad de posibles aplicaciones. Entre ellas, destaca su utilidad en el ámbito de la percepción remota, donde el volumen de imágenes existentes y las que se generan diariamente dificultaría y encarecería enormemente la segmentación manual. Sin embargo, mediante el uso de modelos de visión por computadora, resulta posible segmentar grandes cantidades de datos o a medida que se agrega una nueva imagen a la base de datos de manera eficiente.

En la Figura 2.3 se puede observar un ejemplo una segmentación semántica, en ese caso fue para las clases persona, bicicleta y fondo de la imagen.

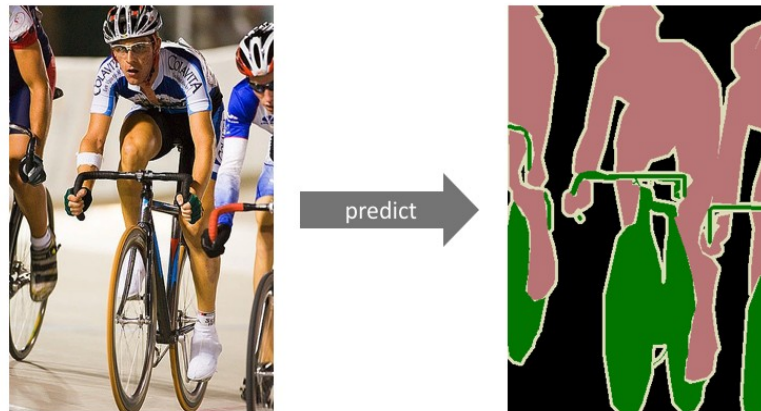


Figura 2.3: Ejemplo de segmentación semántica para las clases persona (rosa), bicicleta (verde) y fondo (negro). Tomada de Lamba, (2019).

2.2.2. Aprendizaje automático

La inteligencia artificial (IA) es un campo de la informática que busca crear sistemas capaces de simular o replicar los comportamientos que se consideran inteligentes en la naturaleza, ya sean de los seres humanos, de otros animales o de fenómenos naturales como la organización social de algunas especies (aves, abejas) o la evolución natural. Hay muchos comportamientos inteligentes que desarrolla el ser humano de manera habitual como jugar, conducir, analizar patrones, pintar, cantar y moverse. La IA se divide en varios subcampos que abordan distintos tipos de comportamiento inteligente, como por ejemplo, la robótica, el procesamiento del lenguaje natural, aprendizaje automático, etc.

El aprendizaje automático es una rama de la IA que se enfoca en dotar a las máquinas de la capacidad de aprender a partir de experiencias, en lugar de programar cada comportamiento individualmente. El objetivo del aprendizaje automático es generalizar el conocimiento a partir de ejemplos, permitiendo que la máquina realice tareas que no se le habían enseñado específicamente.

Existen diversas técnicas de aprendizaje automático, entre ellas, los árboles de decisión, los modelos de regresión, máquina de soporte de vectores y en los últimos años, el uso de redes neuronales se han vuelto cada vez más populares, en particular el llamado aprendizaje profundo, dado el avance en optimización, y descubrimientos fundamentales relacionados con la posibilidad de entrenar redes neuronales con múltiples capas internas. Las redes neuronales profundas aprenden de forma jerarquizada, con las primeras capas aprendiendo conceptos más concretos o simples y las capas siguientes aprendiendo conceptos más complejos y abstractos. Por lo que son muy eficientes para extraer o resolver problemas más complejos.

En resumen, parte de la IA y el aprendizaje automático buscan imitar el comportamiento inteligente de los seres humanos, permitiendo a las máquinas aprender a partir de ejemplos y generalizar el conocimiento. El uso de redes neuronales profundas ha llevado a avances significativos en la capacidad de las máquinas para realizar tareas complejas, como la segmentación semántica.

2.2.3. Red neuronal

La forma más básica de una red neuronal es una neurona y es el modelo del perceptrón, desarrollado por Frank Rosenblatt en 1957. En su forma más simple, un perceptrón toma varias entradas X , aplica pesos W a esas entradas, realiza una suma ponderada, y luego pasa el resultado a través de una función de activación no lineal para producir una salida binaria (Figura 2.4).

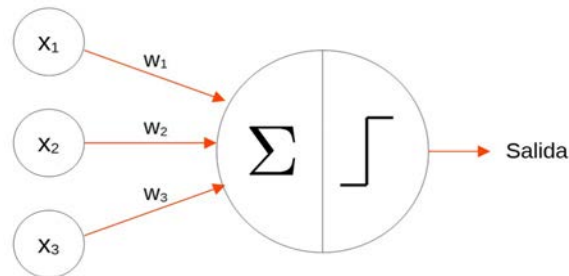


Figura 2.4: Representación gráfica de modelo de perceptrón.
Elaboración propia.

Los pesos W son parámetros ajustables que determinan la contribución relativa de cada entrada a la salida. Cada conexión entre una entrada y el perceptrón tiene un peso asociado, y estos pesos son los que la red ajusta durante el proceso de entrenamiento para aprender a realizar tareas específicas.

Un perceptrón puede tomar decisiones sencillas y una combinación de ellos pueden tomar decisiones más complejas. Cuando se apilan múltiples perceptrones en capas y se les permite interactuar entre sí mediante conexiones ponderadas y funciones de activación no lineales, se crea una red neuronal.

La red neuronal de propagación hacia adelante, es un modelo fundamental en el ámbito del aprendizaje profundo, esta arquitectura de red consta de capas de nodos, cada uno de los cuales realiza una operación de ponderación y activación (Figura 2.5). La estructura básica se compone de tres tipos de capas:

- Capa de entrada (*input layer*): Esta capa consiste en nodos que representan las variables de entrada.
- Capas ocultas (*hidden layers*): Estas capas contienen nodos intermedios que realizan operaciones de ponderación y activación.
- Capa de salida (*output layer*): La capa de salida produce los resultados finales de la red.

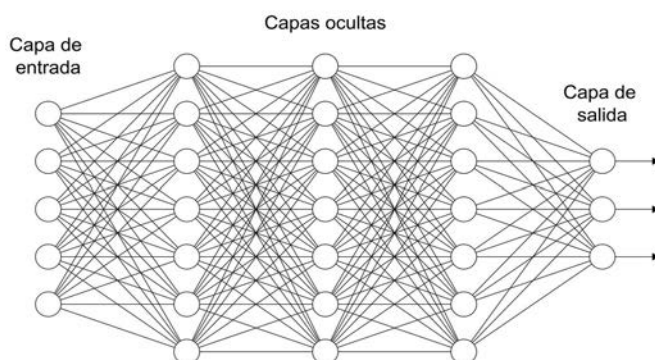


Figura 2.5: Representación gráfica de una red neuronal muy sencilla.
Elaboración propia.

En problemas de clasificación, a la salida de la última capa se le aplica una función de activación softmax, que nos permite convertir las salidas provenientes a probabilidades que nos indican la posibilidad de que una determinada entrada pertenezca a una clase. En otras palabras, la softmax realiza una operación exponencial y normaliza los elementos del vector de entrada, convirtiéndolos en probabilidades. La salida de la softmax puede interpretarse como una distribución de probabilidad sobre las diferentes clases o categorías.

Supongamos que la función softmax toma como entrada un vector $z = (z_1, z_2, \dots, z_k)$ y produce un nuevo vector $p = (p_1, p_1, \dots, p_k)$, donde cada elemento p_i se calcula mediante la fórmula:

$$p_i = \frac{e^{x_i}}{\sum_{j=1}^k e^{z_j}} \quad (\text{Ec. 2.2})$$

2.2.3.1. Función de pérdida

La función de pérdida, también conocida como función objetivo o función de costo, es una medida que nos permite evaluar que tan buenos son los parámetros actuales que nuestra red neuronal tiene, eso lo hace cuantificando la diferencia entre las predicciones de un modelo y los valores reales, con base en ese indicador podemos realizar ajustes a los parámetros tratando de minimizar la función de pérdida mediante un algoritmo de optimización.

Existen diferentes tipos de funciones de pérdida, la elección depende del tipo de tarea que se esté abordando (clasificación, regresión, etc.) y de las propiedades específicas del modelo. Algunas de las funciones de pérdida más comunes son:

- Error cuadrático medio: Se utiliza en problemas de regresión, donde la tarea es predecir un valor numérico. Se define como la media de los cuadrados de las diferencias entre los valores reales y los predichos. La pérdida cuadrática penaliza los errores grandes, y es sensible a los valores atípicos. Su fórmula es:

$$L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (\text{Ec. 2.3})$$

Donde:

- y es el valor real.
 - \hat{y} es el valor predicho.
 - n es el número de observaciones.
- Hinge Loss: Se emplea para entrenar clasificadores de margen máximo, especialmente para las máquinas de vectores de soporte (SVM de *Support Vector Machine*). Para una salida deseada $t = \pm 1$ y una puntuación del clasificador. Su fórmula es:

$$L(y, \bar{y}) = \max(0, 1 - t \cdot y) \quad (\text{Ec. 2.4})$$

Donde:

- y es el valor real.
 - \hat{y} es el valor predicho.
 - t es la salida deseada.
- Entropía cruzada (*cross entropy*): Se define como la suma de los productos de los valores reales y el logaritmo de los valores predichos y se utiliza para problemas de clasificación multiclase, donde se quiere predecir una etiqueta categórica de más de dos clases. La pérdida de entropía cruzada categórica penaliza las predicciones erróneas, y es insensible a los valores atípicos. Su fórmula es:

$$L(y, \hat{y}) = - \sum_{i=1}^n \sum_{j=1}^m y_{ij} \log(\hat{y}_{ij}) \quad (\text{Ec. 2.5})$$

Donde:

- y es el valor real.
- \hat{y} es el valor predicho.
- n es el número de observaciones.

- m es el número de clases

2.2.3.2. Algoritmo de optimización

La optimización se define como la labor de encontrar el valor mínimo o máximo de una función de pérdida $f(x)$ mediante la exploración de diversos valores de x . Cabe destacar que en el ámbito de cualquier modelo de aprendizaje automático, la implementación de un algoritmo de optimización es obligatoria, dado que su propósito esencial es reducir al mínimo la incidencia de errores y maximizar el número de éxitos.

Comúnmente, los problemas se formulan orientados a la minimización, y para abordar objetivos de maximización se recurre a la siguiente equivalencia:

$$\max_x f(\mathbf{x}) = \min_{\mathbf{x}} -f(\mathbf{x}) \quad (\text{Ec. 2.6}).$$

En esta expresión, la maximización de x en una función $f(x)$ es análoga a la minimización de la misma función multiplicada por el signo contrario. Por ende, resolver un problema de minimización proporciona la solución equivalente para maximización.

La elección entre minimizar o maximizar dependerá de la naturaleza específica del problema y como se esté abordando. Por ejemplo, se puede optar por minimizar el error de clasificación o, en cambio, maximizar el número de éxitos.

Con el fin de determinar el argumento que minimiza o maximiza dicha función de costo, se utiliza la expresión:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} f(\mathbf{x}) \quad (\text{Ec. 2.7})$$

Donde, \mathbf{x}^* representa el valor que minimiza la función $f(x)$ y la notación \arg indica el interés en identificar los argumentos que conducen a la minimización de la función.

2.2.3.2.1. Descenso de gradiente

El gradiente desempeña un papel fundamental en el algoritmo de optimización, ya que proporciona información crucial sobre la dirección y magnitud en la cual se debe desplazar la solución actual para llegar a la solución óptima.

La expresión general para la actualización de x en el caso unidimensional se formula como:

$$x_{t+1} = x_t + \alpha \cdot f'(x) \quad (\text{Ec. 2.8})$$

Aquí, se parte de un vector x en la iteración t y el objetivo es dirigir ese vector hacia la minimización de la función $f(x)$, siendo la dirección de minimización indicada por el gradiente $f'(x)$, que puede ser positivo o negativo. α denota la tasa de aprendizaje y es un valor menor que 1, esta tasa indica en qué medida se debe considerar la magnitud de la derivada en la

actualización, determinando la velocidad de convergencia del algoritmo de descenso de gradiente.

La búsqueda se centra en identificar puntos críticos en la función de pérdida, aquellos en los que la derivada es igual a cero. En términos generales, el algoritmo de descenso de gradiente consiste en los siguientes pasos:

1. Se define un valor inicial x_t
2. Se especifica un valor para la tasa de aprendizaje (α).
3. Se inicia un bucle for con h iteraciones.
 - 3.1. Se calcula un nuevo valor $x_2 = x_t - \alpha \cdot (f'(x)$ ó jacobiano).
 - 3.2. Se actualiza a x_t con el valor de x_2 , $x_t = x_2$.
 - 3.3. Se repite el bucle h veces.

2.2.3.3. Algoritmo de propagación hacia atrás

El entrenamiento de la red neuronal se hace mediante el algoritmo de propagación hacia atrás o retropropagación (*backpropagation*) junto con un método de optimización. La retropropagación es un algoritmo de entrenamiento para redes neuronales que se lleva a cabo en dos pasos principales.

Paso adelante: En este, la red realiza una predicción utilizando sus pesos actuales, calcula la pérdida comparando la predicción con los valores reales y almacena la información relevante para calcular las derivadas parciales. La predicción se obtiene aplicando una función de activación a la suma ponderada de las entradas de cada neurona, y propagando el resultado a través de las capas de la red. La pérdida se mide mediante una función de costo, como la mencionada entropía cruzada, que cuantifica la diferencia entre la salida de la red y la salida deseada. La información que se almacena incluye los valores de entrada, salida y error de cada neurona, así como los pesos y sesgos de cada conexión.

Paso atrás: Estas derivadas parciales se usan para actualizar los pesos de la red en la dirección que minimiza la pérdida, utilizando un algoritmo de optimización. Las derivadas parciales se calculan mediante la regla de la cadena, que permite obtener el gradiente de la pérdida con respecto a cada peso y sesgo de la red. El algoritmo de optimización, como el descenso de gradiente, determina el tamaño y la dirección del cambio que se aplica a cada peso y sesgo, teniendo en cuenta factores como la tasa de aprendizaje.

Este proceso iterativo de ajuste de pesos se repite hasta que la red alcanza un nivel deseado de precisión en sus predicciones. La retropropagación es un algoritmo eficiente y generalizable que permite entrenar redes neuronales de diferentes arquitecturas y aplicaciones. Sin embargo,

también presenta algunos desafíos, como el sobreajuste o la elección de los hiperparámetros adecuados

2.2.4. Redes neuronales convolucionales (CNN)

En el campo del aprendizaje profundo, las CNN han demostrado ser una herramienta muy útil cuando se trabaja con imágenes. Este tipo de redes están inspiradas en las células del cerebro humano encargadas de la visión, más no buscan replicar su comportamiento, las cuales trabajan de forma jerárquica en las que las primeras capas se encargan de detectar formas simples y las siguientes capas formas cada vez más complejas compuestas por las formas simples detectadas en capas anteriores.

2.2.4.1. Convolución

La principal característica de las CNN es la aplicación de capas de convolución. La convolución es una operación matemática (Ec. 2.9) que se emplea para extraer características de una imagen I de tamaño $(m \times n)$, esta operación consiste recorrer la imagen e ir aplicando un filtro o *kernel* K y calcular el producto punto de los valores del filtro y la porción de la imagen.

$$(I * K)(x, y) = \sum_{a=-\infty}^{\infty} \sum_{b=-\infty}^{\infty} I(x - a, y - b) \cdot K(a, b) \quad (\text{Ec. 2.9})$$

Donde:

- $(I * K)(x, y)$ es el valor resultante en la posición (x, y) de la imagen filtrada después de aplicar la convolución con el kernel K , $*$ representa la operación de convolución.
- $I(x - a, y - b)$ es el valor del píxel en la imagen original I desplazado en a posiciones horizontalmente y b posiciones verticalmente desde el píxel central (x, y) al que se le está aplicando la convolución.
- $K(a, b)$ es el valor en el kernel en la posición (a, b) .

Dependiendo de los valores o pesos del kernel, será el resultado o la característica de la imagen que se resaltará. El resultado se le llama mapa de activación y una representación de las características extraídas de la imagen con un kernel en particular. En la Figura 2.6 se muestra un ejemplo del mapa de activación obtenido tras aplicar una convolución usando el kernel vertical del filtro de Prewitt, en este ejemplo, los valores de los pesos del kernel están diseñados para destacar los bordes verticales presentes en la imagen.

Los pesos de los kernels son la variable clave que una CNN ajusta durante el proceso de entrenamiento, con el propósito de minimizar el error al segmentar o clasificar una imagen específica, adaptándose a los objetivos establecidos.

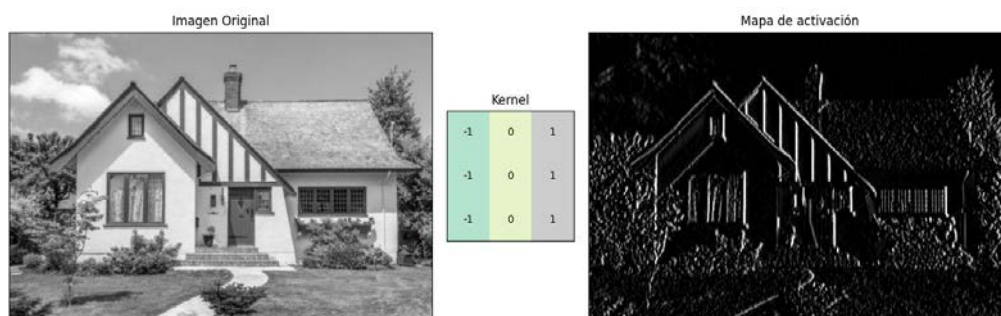


Figura 2.6: Ejemplo del mapa de activación resultado de una convolución con un kernel de 3×3 , en este caso los pesos del kernel hacen que se resalten los bordes verticales de la imagen. Elaboración propia.

Cuando se aplica una convolución a una imagen se reduce el tamaño de la imagen, ya que los píxeles de los bordes no pueden ser calculados, pues no tienen vecinos, si se quiere conservar el tamaño de la imagen se pueden agregar líneas de ceros en los extremos de la imagen, a esto se le llama valor de relleno o *padding* (p). De no emplear un relleno, una convolución de 3×3 generaría una pérdida de una fila superior e inferior y una columna izquierda y derecha. Por lo tanto, un relleno de 1 agregaría una fila o columna de ceros en cada lado de la imagen.

El parámetro de valor de paso o *stride* (s) define cada cuantos píxeles se mueve el kernel al hacer la convolución, el valor por defecto es de 1, pero puede haber valores más altos como 2, en este caso el kernel se mueve dos posiciones a la vez y esto puede resultar en una reducción en la resolución espacial de la matriz resultante, ya que se está saltando un cierto número de posiciones en cada paso.

Para calcular el tamaño de la imagen resultante, dado un kernel, un valor de relleno y un paso específico se puede calcular con la siguiente fórmula:

$$T_{salida} = \left\lfloor \frac{T_{entrada} + 2p - k}{s} \right\rfloor + 1 \quad (\text{Ec. 2.10})$$

Donde:

- T_{salida} es el tamaño de salida de la imagen después de la convolución.
- $T_{entrada}$ es el tamaño de entrada de la imagen.
- p es el valor de relleno.
- k es el tamaño del kernel.
- s es el valor de paso.

En las CNN se aplican múltiples capas de convolución, los pesos de los filtros con los que se inicia se asignan aleatoriamente y el objetivo del modelo es aprender o encontrar los mejores pesos para que pueda caracterizar la imagen que queremos.

2.2.4.2. Capas de agrupación

Cuando en una CNN se aplican múltiples capas de convolución va creciendo la profundidad o dimensión d del tensor (m, n, d) que representa los mapas de activación, si se aplican l capas de convolución, nuestro tensor pasará a tener las mismas l dimensiones y esto es algo que computacionalmente podría ser muy costoso, por lo tanto, cada cierto número de capas de convolución se van aplicando capas de agrupación (*pooling layers* en inglés), que lo único que hacen es ir reduciendo el tamaño del tensor (m, n) pero dejan intacta la dimensión (d) . Entre las distintas capas de agrupación que se pueden aplicar se encuentran *max pooling*, *average pooling*, *min pooling*, etc.

Por ejemplo, una capa de agrupación máxima o *max pooling* de 2×2 , recorrería cada mapa de activación del tensor con un kernel de 2×2 que iría tomando los valores del tensor y se quedaría con el valor máximo, así, el tensor resultante sería de tamaño $(\frac{m}{2}, \frac{n}{2}, d)$.

2.2.4.3. Convolución transpuesta

Una vez que hemos aplicado múltiples capas de convolución y agrupación, tendremos un tensor de tamaño (i, j) reducido, pero con muchas dimensiones (l) , en la segmentación semántica esto no nos serviría para clasificar cada pixel de la imagen, por lo que la convolución transpuesta nos permite regresar el tamaño del tensor al original, es importante destacar que el objetivo de la convolución transpuesta no es recuperar los valores exactos antes de la convolución, sino únicamente restaurar sus dimensiones (m, n) . Para esto agrega filas y columnas de ceros entre cada fila y columna de la imagen y después se aplica un kernel como en el caso de la convolución normal.

Los parámetros de relleno p y tamaño de paso s que ocupa la convolución transpuesta para regresar una imagen al mismo tamaño (m, n) son los mismos que se usaron para la convolución inicial, pero se usan de otra forma, para calcular los parámetros z y p' . El parámetro z representa el número de filas y columnas que se agregarán de manera intercalada en el interior de la imagen y es igual a $s - 1$, p' son las filas y columnas de ceros que se agregan en los extremos de la imagen ($p' = k - p - 1$) y s' es el nuevo tamaño de paso y es igual a 1 en todos los casos. En la Figura 2.7 se observa un diagrama de este proceso, ****** representa la operación de la convolución transpuesta, *Input* es la imagen resultante después de aplicar una convolución a una imagen de 6×6 con un valor de relleno p de 1 y un tamaño de paso s de 2. Y *Output* es la salida de la convolución transpuesta, donde se puede observar que recupera su tamaño.

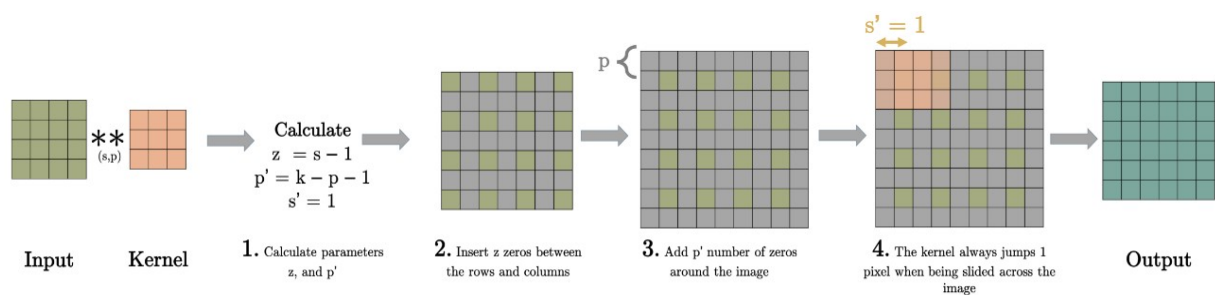


Figura 2.7: Ejemplo de convolución transpuesta. Tomada de A. Zhang et al., (2023)

2.2.4.4. U-Net

La U-Net es una de las arquitecturas de red neuronal convolucional más usada para segmentación semántica. Fue propuesta por Ronneberger et al., (2015) para la segmentación de imágenes biomédicas, donde la disponibilidad imágenes es escasa, debido a su alto costo de adquisición, por lo que destaca el uso efectivo de aumento de datos. Es totalmente convolucional porque está compuesta solamente por capas convolucionales y de agrupación, eliminando las capas completamente conectadas que suelen encontrarse al final de las redes neuronales convencionales.

La principal característica y la razón de su nombre es su forma de U, ya que está compuesta por dos partes, una parte de codificador o contracción que extrae el contexto de la imagen y otra de decodificación o expansión que nos precisa localizaciones (Ronneberger et al., 2015).

En la parte de contracción se aplican bloques de convolución de 3×3 con función de activación ReLU, tamaño de paso de 1, no usa relleno y cada dos bloques de convolución aplica una agrupación máxima (*max pooling*, en inglés) de 2×2 , el resultado es que va disminuyendo el tamaño (m, n) en dos píxeles en cada bloque de convolución y a la mitad cuando aplica una agrupación máxima (ver Figura 2.8).

En la parte de la expansión, en lugar de una agrupación máxima, aplica convoluciones transpuestas de 2×2 para ir duplicando el tamaño de la imagen en alto y ancho, seguido de dos bloques de convoluciones comunes hasta completar los mismos bloques que en la contracción. El tamaño final se reduce debido a que no ocupa un valor de relleno, por lo que en cada bloque de convolución va perdiendo algunos píxeles (ver Figura 2.8). También incorpora un mecanismo de salto de conexión que permite que la información de la parte contracción se transmita directamente a la parte expansión, lo que ayuda a preservar la información de detalle de la imagen original.

El último paso es aplicar i convoluciones de 1×1 solamente para reducir la dimensionalidad de los datos al número de clases que estemos segmentando, i representa el número de clases y para el caso de segmentación semántica binaria, tomaría el valor de 2, por lo que tendríamos como un tensor de $388 \times 388 \times 2$. Y esas dos dimensiones son los dos mapas de activación finales, que representarían el puntaje de que cada píxel pertenezca a la clase 0 ó 1. Por lo que podemos usar la función $\text{argmax}_c(x_c)$, donde x_c representa la activación de la clase c para ese píxel, para encontrar el índice de la clase con la probabilidad más alta para cada píxel.

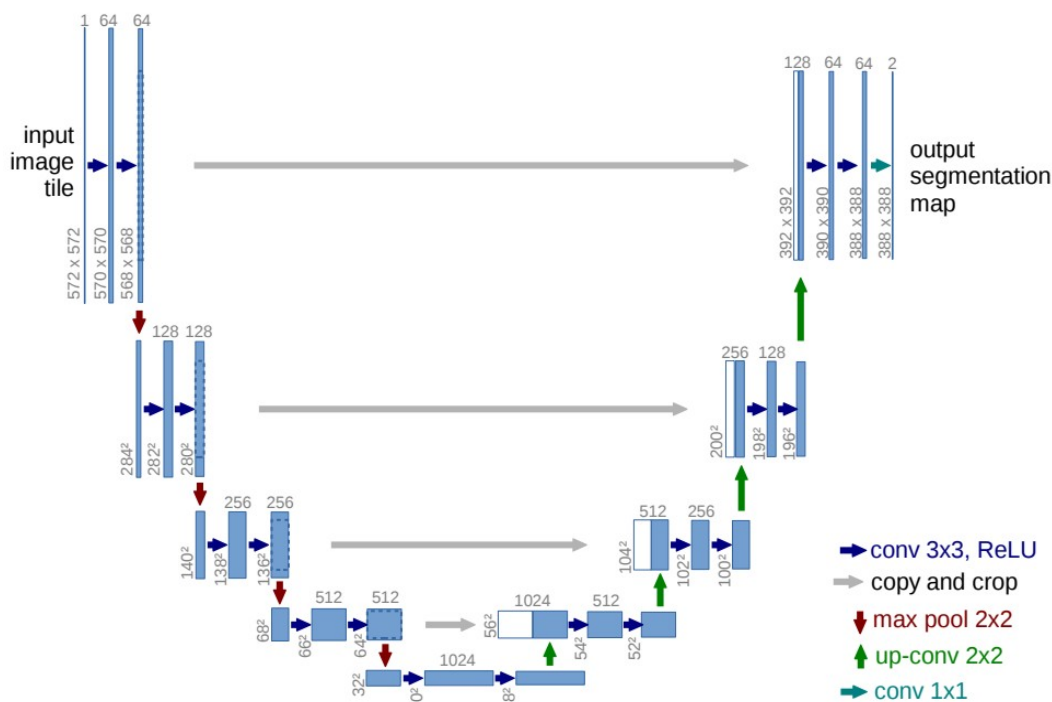


Figura 2.8: Arquitectura original de la U-net. Tomada de Ronneberger et al., (2015)

2.2.5. Hiperparámetros

Los hiperparámetros de una red neuronal son variables que no se aprenden directamente del conjunto de datos durante el entrenamiento, a diferencia de los pesos de los kernels, en cambio, los hiperparámetros se configuran antes de iniciar el proceso de entrenamiento y afectan cómo se entrena y se ajusta la red neuronal. Estos hiperparámetros son esenciales para diseñar y ajustar una red neuronal que funcione adecuadamente para una tarea específica. Los siguientes hiperparámetros son los utilizados para una U-Net:

2.2.5.1. Número de capas o convoluciones iniciales

Las convoluciones iniciales en una arquitectura U-Net son las capas convolucionales que se encuentran en la parte superior de la red, justo después de la capa de entrada, y se utilizan para realizar las operaciones iniciales de procesamiento de la imagen de entrada. Estas convoluciones iniciales desempeñan un papel crucial en la U-Net y están diseñadas para extraer características básicas y representaciones iniciales de la imagen que luego se utilizarán en las capas descendentes y ascendentes de la red.

La lógica que sigue la U-Net es ir duplicando progresivamente el número de convoluciones en el camino de contracción, seguido de una reducción a la mitad en cada paso de expansión. Por lo tanto, el número inicial de convoluciones tiene una repercusión en toda la arquitectura. Variar el número de convoluciones iniciales puede ser una forma de controlar la complejidad del modelo.

2.2.5.2. Función de activación

El objetivo principal de la función de activación es introducir no linealidad en el modelo, permitiendo que la red neuronal pueda aprender relaciones y patrones más complejos en los datos. Sin funciones de activación, las redes neuronales se reducirían a combinaciones lineales de sus entradas, lo que limitaría su capacidad de representar funciones más complejas y realizar tareas de aprendizaje no lineal.

Algunas de las funciones de activación que existen y que se evaluaron en este trabajo fueron las siguientes:

- Sigmoides: La función sigmoide es una función de activación que toma un valor de entrada y lo transforma en un valor en el rango de 0 a 1. Esta función tiene forma de S y tiene una curva suave. La curva se acerca asintóticamente a 0 a medida que x vuelve negativo y se acerca asintóticamente a 1 a medida que x se vuelve positivo.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (\text{Ec. 2.11})$$

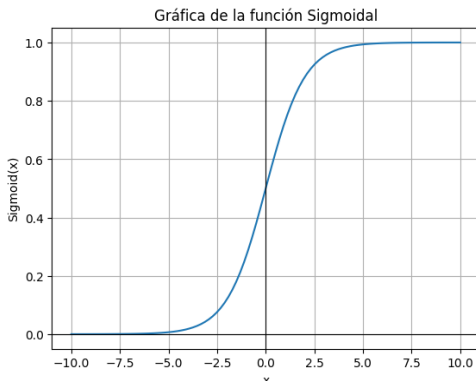


Figura 2.9: Gráfica de la función sigmoide.
Elaboración propia.

- ReLU: (*Rectified Linear Unit*), es una función de activación ampliamente utilizada en redes neuronales y otros modelos de aprendizaje profundo como autocodificadores y transformadores. Se define de la siguiente manera:

$$f(x) = \max(0, x) \quad (\text{Ec. 2.12})$$

La función ReLU tiene un comportamiento simple, pero efectivo: para valores positivos de x , devuelve el mismo valor y para valores negativos de x , devuelve 0. En otras palabras, activa la neurona solo cuando la entrada es positiva y apaga la neurona cuando la entrada es negativa.

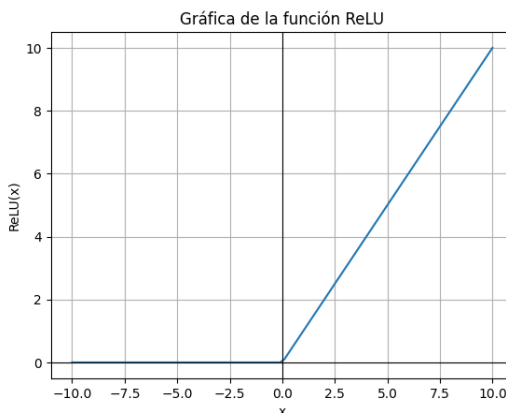


Figura 2.10: Gráfica de la función ReLU. Elaboración propia.

- Leaky ReLU: (*Leaky Rectified Linear Unit*), es una variante de la función ReLU (*Rectified Linear Unit*) que aborda el problema de "neuronas muertas" que puede ocurrir con los valores negativos en la función ReLU original.

$$f(x) = \begin{cases} x & \text{si } x > 0 \\ \alpha x & \text{si } x \leq 0 \end{cases} \quad (\text{Ec. 2.13})$$

α : Es un hiperparámetro que generalmente se establece en un valor pequeño y positivo, como 0.01. Controla la pendiente de la parte negativa de la función, permitiendo una inclinación pequeña en la parte negativa de la función.

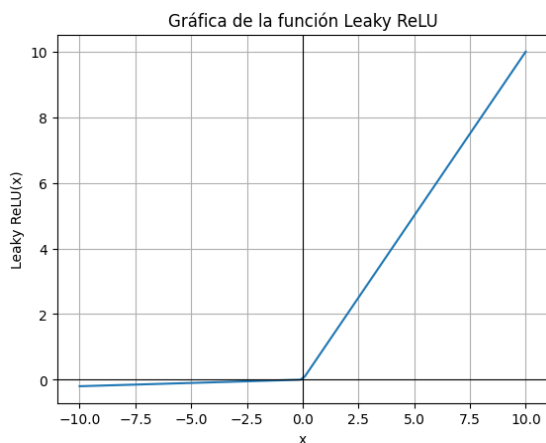


Figura 2.11: Gráfica de la función Leaky ReLU con un α de 0.02. Elaboración propia.

- Tanh: (*Hyperbolic Tangent*), la función tangente hiperbólica es una versión modificada de la función sigmoide y tiene la siguiente definición:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (\text{Ec. 2.14})$$

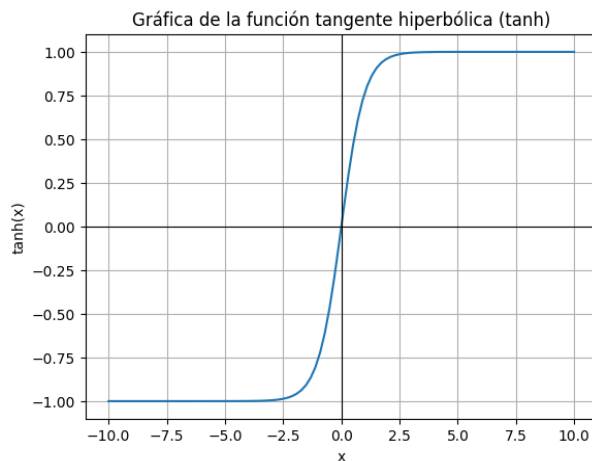


Figura 2.12: Gráfica de la función tanh. Elaboración propia.

- ELU (*Exponential Linear Unit*) es una función de activación cuyo objetivo principal es abordar algunas limitaciones de otras funciones de activación, como ReLU y sus variantes.

$$\text{ELU}(x) = \begin{cases} x & \text{si } x > 0 \\ \alpha \cdot (e^x - 1) & \text{si } x \leq 0 \end{cases} \quad (\text{Ec. 2.15})$$

La función ELU es similar a la función Leaky ReLU en la medida en que introduce una inclinación pequeña en la parte negativa de la función para prevenir el problema de "neuronas muertas". Sin embargo, lo que distingue a la función ELU es que, en lugar de simplemente multiplicar la parte negativa por un valor constante como en Leaky ReLU, la parte negativa se calcula usando una función exponencial, lo que permite que tome valores continuos.

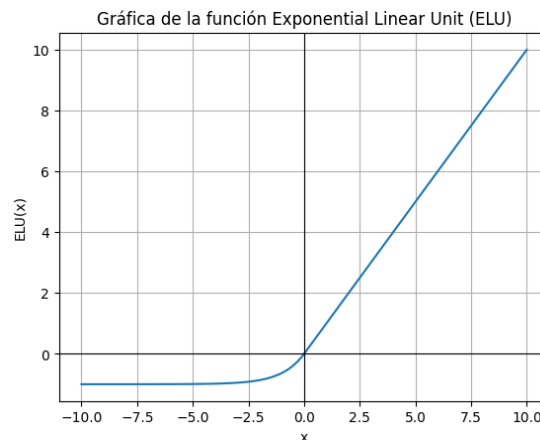


Figura 2.13: Gráfica de la función ELU con un α de 1.
Elaboración propia.

2.2.5.3. Algoritmos de optimización basados en descenso de gradiente

Como ya se mencionó anteriormente, en la U-Net también se usa un algoritmo de optimización u optimizador para ajustar los pesos y los sesgos de un modelo de manera iterativa con el objetivo de minimizar una función de pérdida. Los algoritmos de optimización evaluados en esta tesis son los siguientes:

El algoritmo de descenso de gradiente estocástico (SGD) con momento, conocido también como "*Stochastic Gradient Descent with Momentum*"; Es una variante del algoritmo de descenso de gradiente que incorpora el concepto de momento para mejorar la velocidad de convergencia y ayudar a evitar oscilaciones en la dirección de la optimización. El momento se refiere a la acumulación gradual de una fracción del gradiente pasado en las iteraciones anteriores. Esto ayuda a mantener una dirección de actualización más consistente y suaviza las fluctuaciones

causadas por el ruido en los gradientes. La idea es que, en lugar de realizar actualizaciones bruscas basadas en el gradiente instantáneo, el algoritmo considera la dirección promedio de las iteraciones anteriores para guiar las actualizaciones.

Las ecuaciones generales para la actualización del momento y los parámetros son las siguientes:

$$v_t = \beta \cdot v_{t-1} + (1 - \beta) \cdot \nabla J(\theta_t) \quad (\text{Ec. 2.16})$$

$$\theta_{t+1} = \theta_t - \alpha \cdot v_t \quad (\text{Ec. 2.17})$$

Donde:

- v_t es el vector de momento en la iteración.
- β es el factor del momento, generalmente es un valor entre 0 y 1. Controla cuánto del momento acumulado de las iteraciones anteriores se debe tener en cuenta en la actualización actual. Un valor más alto de β da más peso al momento acumulado y puede ayudar a suavizar las actualizaciones, mientras que un valor más bajo dará más importancia al gradiente instantáneo.
- $\nabla J(\theta_t)$ es el gradiente de la función de pérdida con respecto a los parámetros θ_t en la iteración actual.
- α es la tasa de aprendizaje.

Adam (*Adaptive Moment Estimation*) es uno de los algoritmos de optimización más empleados en los últimos años. Fue diseñado para combinar las ventajas de dos enfoques populares de optimización: el RMSProp (que ajusta las tasas de aprendizaje de manera adaptativa) y el momento (que acumula el gradiente pasado para suavizar las actualizaciones).

El algoritmo Adam calcula tasas de aprendizaje adaptativas para cada parámetro y también acumula momentos del gradiente para tener una dirección más consistente en las actualizaciones. Esto permite que el algoritmo se adapte de manera eficiente a diferentes características del espacio de parámetros y en muchos casos acelere la convergencia en diferentes direcciones.

Las fórmulas generales para la actualización de los parámetros utilizando Adam son las siguientes:

$$m_t = \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot \nabla J(\theta_t) \quad (\text{Ec. 2.18})$$

$$v_t = \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot (\nabla J(\theta_t))^2 \quad (\text{Ec. 2.19})$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (\text{Ec. 2.20})$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (\text{Ec. 2.21})$$

$$\theta_{t+1} = \theta_t - \frac{\alpha}{\sqrt{\hat{v}_t + \epsilon}} \cdot \hat{m}_t \quad (\text{Ec. 2.22})$$

Donde:

- m_t es el momento en la iteración t .
- v_t es el momento del cuadrado del gradiente en la iteración t .
- \hat{m}_t y \hat{v}_t son las estimaciones corregidas de momento y momento del cuadrado, respectivamente.
- β_1 y β_2 son los factores de momento y momento del cuadrado, respectivamente (generalmente cerca de 1, como 0.9 y 0.999).
- $\nabla J(\theta_t)$ es el gradiente de la función de pérdida con respecto a los parámetros θ_t en la iteración actual.
- α es la tasa de aprendizaje.
- ϵ es una pequeña constante para evitar la división por cero.

2.2.5.4. Tasa de aprendizaje

La tasa de aprendizaje o *learning rate* (α) es uno de los hiperparámetros más importantes en una red neuronal. Representa la magnitud o el tamaño de los pasos que un algoritmo de optimización toma para ajustar los pesos de un modelo durante el proceso de entrenamiento, una tasa de aprendizaje muy pequeña puede conllevar a una convergencia lenta o requerir un gran número de iteraciones para alcanzar un mínimo y una tasa de aprendizaje muy alta podría causar saltos muy grandes y nunca llegar a un mínimo deseado.

Para resolver el problema de encontrar un α óptimo existen métodos como el de “super convergencia”, el cual consta de ir variando el α entre un valor mínimo y un valor máximo, conforme avanzan las iteraciones. El primer acercamiento fue usar un α cíclico a lo largo del entrenamiento (Smith, 2017), y podía haber varios ciclos, en los que el α va variando entre un valor mínimo y un máximo (Figura 2.14)

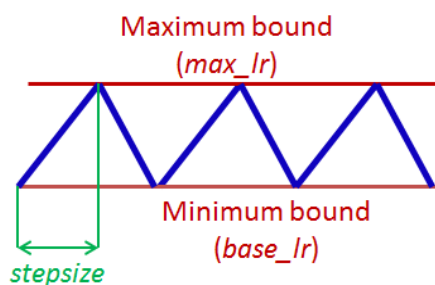


Figura 2.14: Ajuste de tasa de aprendizaje de un valor mínimo a un máximo por 3 ciclos. El eje x representa las iteraciones del modelo. Tomada de Smith, (2017).

Para determinar los valores mínimo y máximo se entrena el modelo por una época, y para cada mini-lote se ajusta el α de un valor muy pequeño a uno muy grande. Luego, se procede a trazar un gráfico que represente los valores de la tasa de aprendizaje frente a la precisión alcanzada o la función de pérdida obtenida en dicho mini-lote. El propósito de esta visualización es identificar un rango de tasas de aprendizaje donde la función de pérdida o el valor de costo disminuyan de manera constante o donde la precisión mejore significativamente. Al ubicar este punto de mejora constante, pero antes de que se produzcan oscilaciones o divergencias, se logra determinar el α máximo y el mínimo es igual a un décimo del máximo.

El siguiente enfoque fue el de “One Cycle LR” (Smith & Topin, 2018), que parte de la base anterior, pero como su nombre lo dice, en lugar de entrenar por varios ciclos, solo varía por un ciclo. El α va incrementando hasta aproximadamente el 40% de las iteraciones y comienza a bajar, y antes de llegar al final de las iteraciones comienza a bajar de una forma más suave y hasta llegar a un valor significativamente pequeño (Figura 3.17). Cabe destacar que este enfoque fue capaz de demostrar la posibilidad de reducir el número de iteraciones requeridas en el proceso de entrenamiento en un factor de hasta ocho veces, manteniendo o incluso mejorando la precisión del modelo.

2.2.5.5. Tamaño de lote

El tamaño de lote, también conocido como *batch size*, es un hiperparámetro en el entrenamiento de modelos de redes neuronales, incluida la arquitectura U-Net. Este parámetro define la cantidad de ejemplos de entrenamiento que se utilizan en cada iteración o época para actualizar los pesos de la red neuronal. En el contexto de la U-Net, el tamaño de lote juega un papel crucial en la eficiencia del entrenamiento y la convergencia del modelo. Al ajustar este hiperparámetro, se pueden experimentar diferentes dinámicas durante el proceso de optimización.

Un tamaño de lote más grande puede aprovechar la eficiencia computacional al procesar múltiples ejemplos simultáneamente, permitiendo una paralelización más efectiva en arquitecturas de hardware modernas, como las GPU. Además, los tamaños de lote mayores pueden proporcionar estimaciones de gradiente más estables, lo que a menudo resulta en una convergencia más rápida del modelo.

Por otro lado, tamaños de lote más pequeños pueden tener beneficios en términos de memoria, especialmente en situaciones donde los recursos computacionales son limitados. Además, tamaños de lote más pequeños pueden ayudar a evitar mínimos locales y permitir una exploración más amplia del espacio de parámetros durante el entrenamiento.

El tamaño de lote puede ser cualquier número, pero se suele elegir el más grande que no cause problemas de memoria y se reduce si se quiere ajustar el gradiente. Las GPU están diseñadas para manejar operaciones en paralelo, y los tamaños de lote que son potencias de 2 como 4, 8, 16, 32, 64, etc; pueden aprovechar mejor esta capacidad.

2.2.6. Métricas de evaluación

La métrica de evaluación es una herramienta utilizada para cuantificar o evaluar de manera cuantitativa el rendimiento de modelos o sistemas en diferentes tareas y contextos. Una de las más usadas en visión por computadora es la precisión (*accuracy* en inglés). La precisión representa la proporción de predicciones correctas entre todas las predicciones totales (Ec. 2.18), el rango de valores va de 0 a 1, un valor de 0 significa que todas las predicciones son incorrectas, mientras que un valor de 1 indica que todas las predicciones son correctas.

$$\text{Precisión} = \frac{TP + TN}{TP + TN + FP + FN} \quad (\text{Ec. 2.23})$$

Donde:

- TP (*True Positives*, verdaderos positivos): Representa el número de casos en los que el modelo predijo correctamente la clase positiva (verdadero) cuando la clase verdadera también es positiva.
- TN (*True Negatives*, verdaderos negativos): Representa el número de casos en los que el modelo predijo correctamente la clase negativa (falso) cuando la clase verdadera también es negativa.
- FP (*False Positives*, falsos positivos): Representa el número de casos en los que el modelo predijo incorrectamente la clase positiva (verdadero) cuando la clase verdadera es negativa (falso).

- FN (*False Negatives*, falsos negativos): Representa el número de casos en los que el modelo predijo incorrectamente la clase negativa (falso) cuando la clase verdadera es positiva (verdadero).

Sin embargo, en segmentación semántica la métrica de precisión puede ser engañosa debido a que el número de píxeles pertenecientes a cada clase puede estar muy sesgado, por ejemplo, supongamos que estamos tratando con la segmentación de un objeto simple en una imagen binaria (solo dos clases: fondo y objeto) de $100 \times 100 = 10000$ píxeles y tenemos la siguiente situación:

- ✗ Verdaderos positivos (TP): 250 píxeles.
- ✗ Falsos positivos (FP): 750 píxeles.
- ✗ Verdaderos negativos (TN): 8250 píxeles.
- ✗ Falsos negativos (FN): 750 píxeles.

El cálculo de la precisión sería el siguiente:

$$\text{Precisión} = \frac{250 + 8250}{250 + 8250 + 750 + 750} = 0.85$$

Lo cual no representa de manera muy clara el rendimiento del modelo. Por lo tanto, utilizamos otras métricas para evaluar el rendimiento, como lo es el índice de Jaccard o intersección sobre la unión (IoU por sus siglas en inglés).

El IoU es una métrica que se emplea comúnmente en problemas de segmentación y clasificación, especialmente cuando se trabaja con conjuntos de datos binarios. El coeficiente de Jaccard mide la similitud entre dos conjuntos dividiendo el tamaño de su intersección sobre el tamaño de su unión. En el contexto de la segmentación semántica, se usa para evaluar la superposición entre las máscaras predichas y las máscaras objetivo.

$$\text{IoU} = \frac{|A \cap B|}{|A \cup B|} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \quad (\text{Ec. 2.24})$$

Donde:

- A es el conjunto de instancias que el modelo predice como positivas.
- B es el conjunto real de instancias que son positivas.
- $|A \cap B|$ es la intersección entre los conjuntos A y B . Esta intersección representa a los verdaderos positivos (TP).
- $|A \cup B|$ es unión entre los conjuntos A y B , también se puede expresar en términos de la suma de $\text{TP} + \text{FP} + \text{FN}$.

Para el ejemplo planteado anteriormente, el IoU sería el siguiente:

$$\text{IoU} = \frac{250}{250 + 750 + 750} = 0.142$$

El coeficiente Dice, también conocido como Coeficiente de Sørensen-Dice, es otra medida empleada comúnmente para comparar la similitud entre dos conjuntos. Se utiliza en campos como la estadística, la teoría de la información, la visión por computadora y la segmentación de imágenes. La fórmula del coeficiente Dice se expresa como:

$$\text{Dice} = \frac{2 \times |A \cap B|}{|A| + |B|} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \quad (\text{Ec. 2.25})$$

Donde:

- $|A|$ es el tamaño del conjunto A.
- $|B|$ es el tamaño del conjunto B.
- $|A \cap B|$ es la intersección entre los conjuntos A y B.

Para el ejemplo anterior, el coeficiente Dice sería el siguiente:

$$\text{Dice} = \frac{2 \times 250}{2 \times 250 + 750 + 750} = 0.25$$

Es evidente que las métricas de IoU y Dice proporcionan una evaluación más adecuada del ejercicio planteado. Con la métrica de precisión, podría parecer que se está logrando una segmentación muy buena cuando, en realidad, no es el caso. Las métricas de IoU y Dice tienen en cuenta la superposición y la similitud entre las predicciones y las etiquetas verdaderas, lo que las hace más adecuadas para medir la calidad de la segmentación semántica.

Capítulo 3. Metodología

El objetivo de este capítulo es describir el camino que se siguió para abordar los objetivos particulares y central de esta investigación. Comenzando con un análisis de la distribución temporal de los incendios forestales en México (etapa 1), seguido de un análisis cualitativo de las diversas bandas y compuestos generables a partir de las imágenes de ABI, en aras de obtener las bandas que mayor información aportan sobre las columnas de humo (etapa 2). A continuación, se procede con la construcción y diseño del conjunto de datos con el que entrenaremos el modelo (etapa 3) y por último, se detalla el diseño del modelo definitivo, haciendo hincapié en la justificación de cada elección de hiperparámetros realizada en este proceso (etapa 4).

Se ha subido todo el código y los scripts generados al repositorio de GitHub: <https://github.com/colvertG/unet-goes16-segmentacion-columnas-humo.git>

El diagrama de flujo contenido en la Figura 3.1, proporciona una síntesis visual de la metodología empleada en este capítulo.

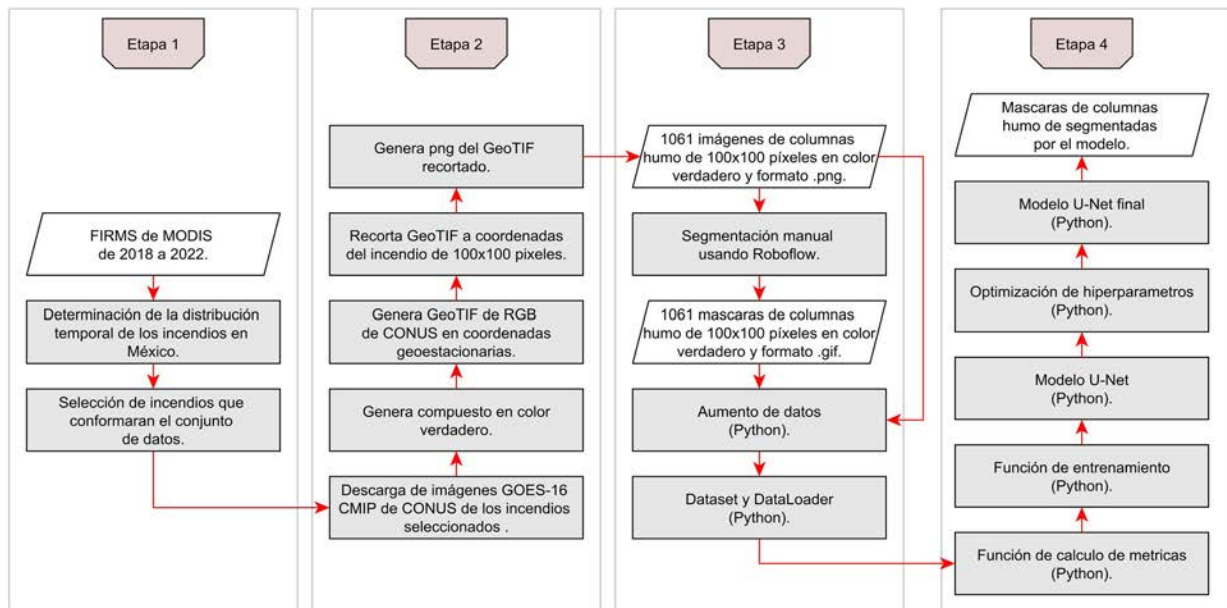


Figura 3.1: Diagrama de flujo de la metodología. Elaboración propia.

3.1. Diseño del conjunto de datos de columnas de humo

En el presente subcapítulo, se detalla exhaustivamente el proceso llevado a cabo para la creación del conjunto de datos de columnas de humo segmentadas manualmente, abordando cada fase esencial de su elaboración. Se comienza explorando la determinación de la temporalidad de los datos y la selección precisa de sus ubicaciones geográficas. Luego, se describe en profundidad el procedimiento de descarga de las imágenes y su posterior procesamiento. Después, se aborda el aspecto crucial de la segmentación manual, donde se delinea el método empleado y se proporciona una descripción detallada del proceso. Finalmente, se brinda una visión completa de los resultados obtenidos.

El conjunto de datos final se puede descargar del sitio web: <https://www.kaggle.com/datasets/colvertgomez/goes16-wildfires-smoke-plumes-dataset>

3.1.1. Determinación de la temporalidad de los datos

El primer paso fue determinar el periodo de tiempo de las imágenes con las que se construyó el conjunto de datos, para ello, conocer la distribución temporal de los incendios en México fue de vital importancia, ya que la cantidad de imágenes que existen de GOES-16 es muy grande, debido a su alta resolución temporal, por lo que desplegar todas las imágenes sería una tarea muy tardada y costosa, así que se realizó una gráfica de los puntos de calor en México asociados a incendios forestales de 2018 a 2022, agrupando los incendios por mes.

Los datos utilizados para hacer este análisis fueron descargados de FIRMS (*Fire Information for Resource Management System*), este es un sistema desarrollado por la NASA que proporciona información en tiempo casi real sobre puntos de calor asociados a incendios forestales en su mayoría y algunas otras anomalías térmicas como actividad volcánica e industria. El sistema usa datos MODIS¹ y VIIRS² y los resultados son publicados en la página <https://firms.modaps.eosdis.nasa.gov/> aproximadamente 3 horas después del paso del satélite.

Los datos seleccionados fueron los generados por MODIS, ya que su resolución espacial es de 1 km, lo que podría resultar en la omisión de incendios de menor tamaño y centrarse en los de mayor magnitud, en comparación con VIIRS (375 m).

1 MODIS (*Moderate Resolution Imaging Spectroradiometer*) es el sensor a bordo de los satélites Aqua y Terra de la NASA. Tiene una resolución espacial que varía entre 250 metros y 1 kilómetro dependiendo la banda.

2 VIIRS (*Visible Infrared Imaging Radiometer Suite*) es el sensor a bordo de los satélites Suomi-NPP y NOAA-20 de la NASA/NOAA. Tiene una resolución espacial de 375 metros.

Como resultado del análisis de la distribución temporal de los incendios en México entre 2018 y 2022 encontramos que existe una constante en relación con los períodos de actividad de incendios en todos los años. Este ciclo comienza en febrero con un aumento exponencial de los incendios, que alcanza su punto máximo en mayo, seguido de un marcado descenso en el mes de junio y seguir bajando en los meses posteriores. También observamos que los meses de mayo y abril, consecutivamente, son los de mayor número de incendios por mucha diferencia, aproximadamente 9 veces más incidentes que los meses de menor actividad (agosto a noviembre), como se observa en la Figura 3.2.

Adicionalmente, pudimos constatar que, al igual que en otras zonas del globo, el año 2019 se posicionó como uno de los años más significativos en términos de número de incendios forestales (Gomez, 2019).

Debido a lo observado, nos centramos en trabajar con los datos de abril y mayo de 2022.

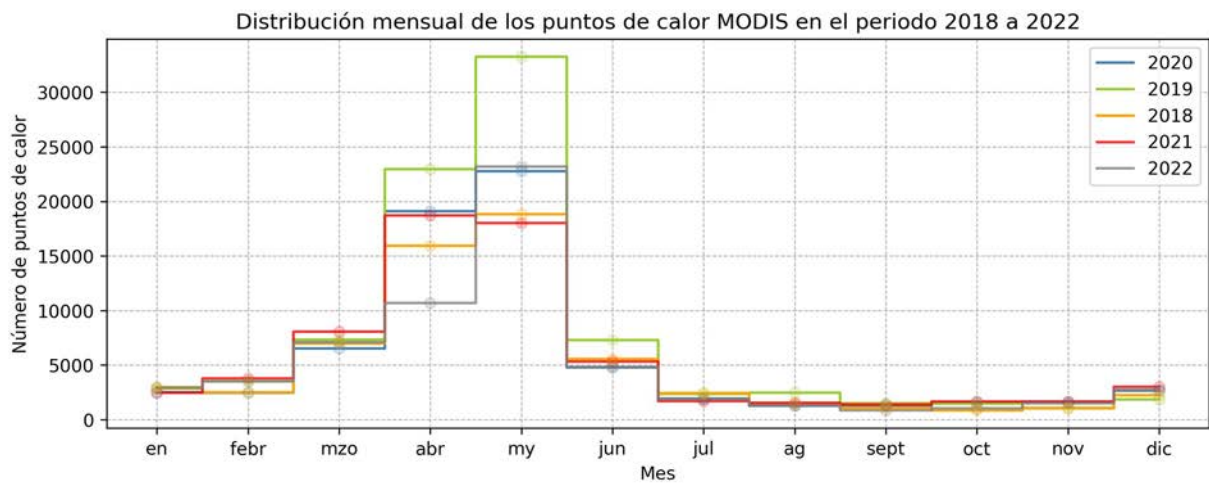


Figura 3.2: Puntos de calor detectados por el sistema FIRMS usando MODIS de 2018 a 2022, agrupado por mes. Elaboración propia con datos de FIRMS (<https://firms.modaps.eosdis.nasa.gov>).

3.1.2. Selección de incendios que conforman el conjunto de datos

Tras completar el análisis previo, se optó por usar las imágenes de abril y mayo de 2022 para la elaboración de conjunto de datos, debido a que estos meses presentaron la mayor incidencia de incendios. Posteriormente, se hizo un análisis más detallado de estos meses con la ayuda de Google Earth Engine para una visualización rápida en un compuesto de color verdadero y se seleccionaron los 20 incendios que exhibieron columnas de humo claramente definidas. Como se utilizó CONUS, se aceptaron algunos incendios en la frontera con México, sobre el estado de

Nuevo México, Estados Unidos, y uno en Guatemala. Los incendios seleccionados se muestran en la Tabla 3.1.

Fecha	Horario (UTC)	X	Y	Lugar aproximado
2022-04-05	18:00 – 21:00	-1690000	1890000	Campeche, Escárcega
2022-04-05	17:00 – 21:00	-1590000	1740000	Guatemala, El remate
2022-04-10	15:20 – 21:00	-2430000	2400000	Tamaulipas, C. Victoria
2022-04-13	15:00 – 20:00	-2420000	2500000	N. León, Monterrey
2022-04-19	17:30 – 20:00	-1710000	1700000	Chiapas, Zamora
2022-04-20	15:00 – 18:00	-2720000	1940000	Michoacán, Uruapan
2022-04-30	16:00 – 21:00	-3020000	2830000	Sonora, Valle de Tacupeto
2022-04-30	14:30 – 20:30	-2790000	1980000	Michoacán, Colima
2022-05-01	15:00 – 21:00	-2770000	1990000	Jalisco, Pihuamo
2022-05-02	14:30 – 20:30	-2770000	2000000	Jalisco, Pihuamo
2022-05-05	15:00 – 19:30	-2850000	1910000	Michoacán, Tepalcatepec
2022-05-10	15:00 – 19:30	-2560000	1760000	Guerrero, Chilpancingo
2022-05-15	14:30 – 20:30	-2810000	3240000	Nuevo México, Silver City
2022-05-16	14:30 – 21:30	-2760000	3260000	Nuevo México, Silver City
2022-05-17	14:30 – 21:30	-2760000	3240000	Nuevo México, Silver City
2022-05-17	18:00 – 21:00	-2550000	3440000	Nuevo México, Santa Fe
2022-05-17	19:00 – 21:30	-2960000	2880000	Chihuahua, Madera
2022-05-18	15:00 – 19:30	-2450000	1700000	Guerrero, Ayutla
2022-05-19	15:30 – 17:30	-2450000	1700000	Guerrero, Ayutla
2022-05-21	17:30 – 19:30	-2900000	1960000	Jalisco, Autlán de Navarro

Tabla 3.1: Incendios seleccionados para la creación del conjunto de datos, donde se observa la fecha, el rango de horario donde se observó la columna de humo en horario UTC (una imagen cada 5 minutos), las coordenadas geoestacionarias (X, Y) del vértice inferior izquierdo de cada incendio, así como el nombre del lugar, ciudad o población más cercana al incendio.

3.1.3. Análisis de las bandas y compuestos de GOES-16

Partiendo de la premisa de que, si un ser humano es capaz de segmentar una imagen visualmente, es posible desarrollar algoritmos y técnicas de procesamiento de imágenes que también puedan realizar segmentación de manera automatizada. Procedimos a efectuar un análisis visual de cada una de las bandas en una imagen donde se evidenciara una columna de humo distintiva.

Para esto ubicamos la columna de humo de un incendio forestal del 1º de mayo de 2022 cerca de Ciudad Guzmán, Jalisco, en las coordenadas geográficas:

- Longitud: de -104 grados a -102 grados.
- Latitud: de 19 grados a 21 grados.

3.1. Diseño del conjunto de datos de columnas de humo

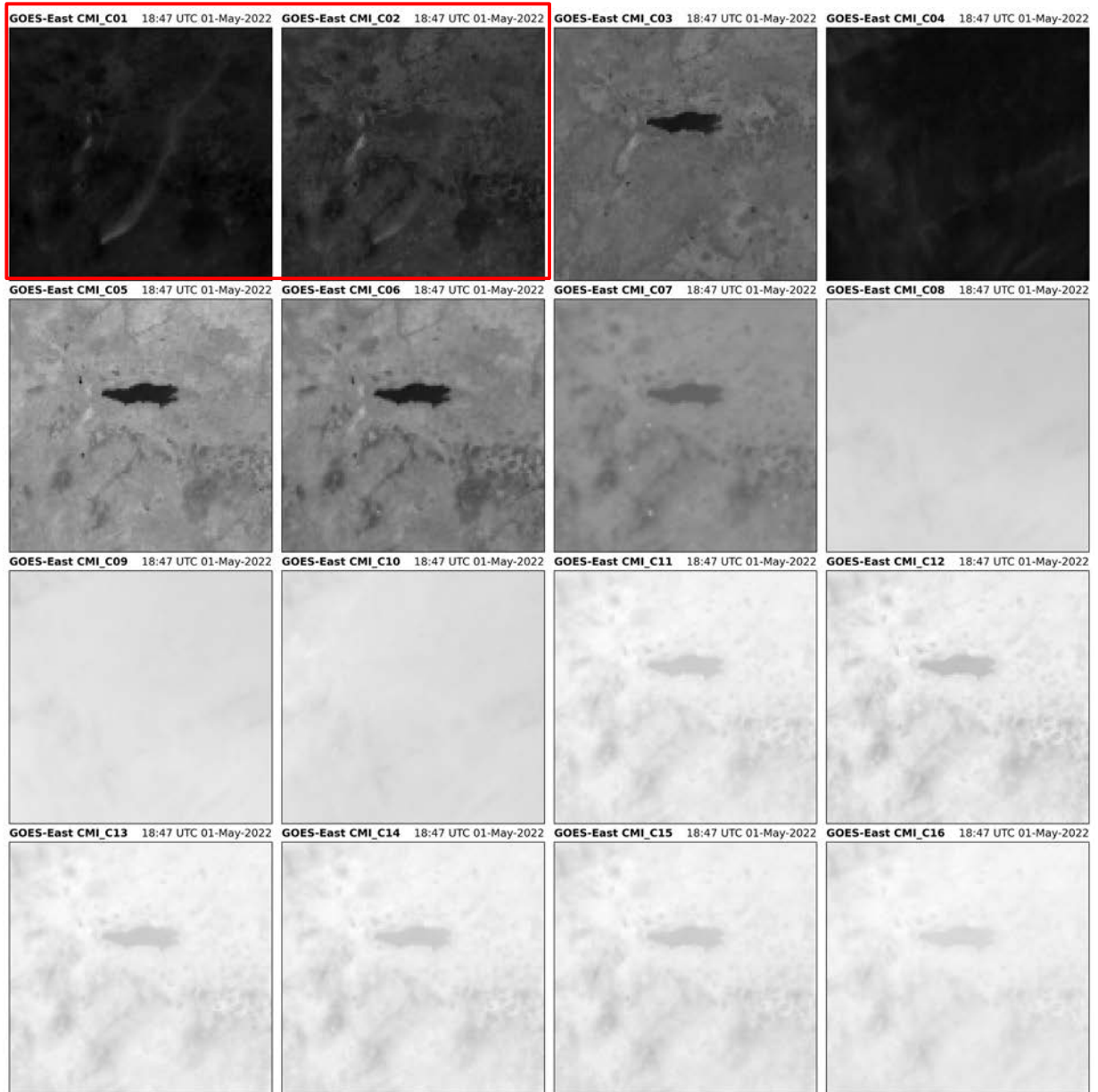


Figura 3.3: Acercamiento a una columna de humo de un incendio forestal, coordenadas geográficas: (-104, -102, 19, 21) del día 1 de mayo de 2022 a las 18:47 UTC de las 16 bandas de GOES-16. Elaboración propia.

La conclusión después de analizar visualmente cada una de las 16 bandas (Figura 3.3), fue que solamente las bandas 1 y 2 aportaban información donde se pudiera hacer una clara diferenciación del “humo” con los píxeles de “no humo”

Esto tiene mucho sentido, ya que la banda 1 ($0.47\mu\text{m}$), o “Azul” sirve particularmente para monitorear aerosoles, es más sensible a los aerosoles porque, esa longitud de onda está en una

parte del espectro electromagnético donde la dispersión atmosférica de Rayleigh¹ en cielo despejado es más frecuente. Las señales de humo y polvo en esta banda son más evidentes cuando el sol está bajo en el cielo, como el amanecer y el atardecer (Yung, 2003). Adicionalmente, hay una transmitancia disminuida (dispersión incrementada) a longitudes de onda visibles más cortas (Figura 3.4). Esto es importante porque los filamentos delgados de humo, particularmente los aerosoles pequeños podrían no ser detectables a partir de bandas de longitud de onda más largas (Schmit et al., 2018). Por lo tanto, las columnas de humo de incendios forestales se observan claramente en esta banda si el cielo está despejado.

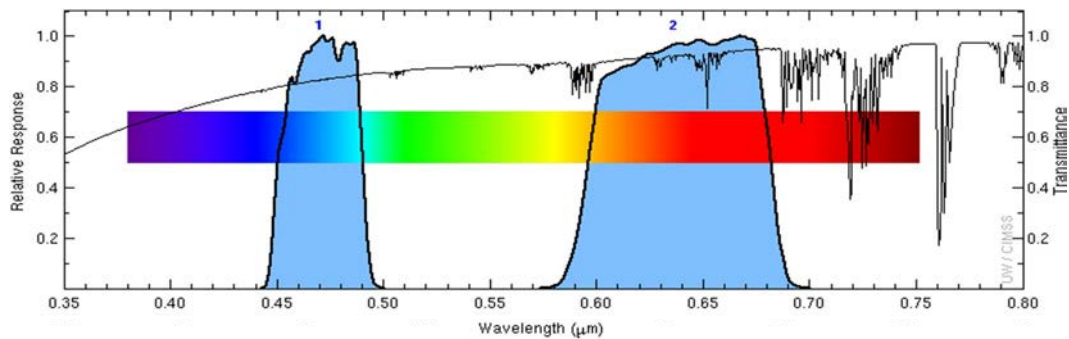


Figura 3.4: Respuesta espectral de las bandas visibles ABI (banda 1 y 2) y diagrama de transmisión atmosférica (ventanas atmosféricas). Tomada de CIMSS and ASTER spectral library and Mat Gunshor

En la banda 2 (0.64μm) o “Roja”, las nubes y el polvo espeso se ven similares a la banda 1. Más allá de los aerosoles, las bandas visibles tienen multitud de aplicaciones. Son ideales para identificar características a pequeña escala, como la niebla del río, los bordes de la niebla, o las nubes de tipo cúmulos durante el día. La banda de 0.64μm también se ha utilizado durante el día para monitorear la capa de nieve y hielo, ayudar a detectar cenizas volcánicas y analizar huracanes y tormentas de invierno (Schmit et al., 2018).

El siguiente análisis fue sobre algunos compuestos que se pueden generar a partir de las bandas de GOES-16, en particular, usamos algunos de los propuestos por la *Regional and Mesoscale Meteorology Branch* (RAMMB, 2023). Los 20 compuestos analizados fueron los siguientes: *air mass, ash, day cloud convection, day cloud phase, day convection, day land cloud, day land cloudfire, day snow fog, differential water vapor, dust, fire temperature, natural color, night fog difference, nighttime microphysics, normalized burn ratio, rocket plume, split window difference, sulfur dioxide, true color* y *water vapor*. Una descripción más detallada de cada uno de estos compuestos se puede consultar en https://rammb2.cira.colostate.edu/training/visit/quick_reference/#tab17.

1 La dispersión atmosférica de Rayleigh ocurre cuando la radiación interactúa con moléculas y partículas en la atmósfera que tienen un diámetro menor que la longitud de onda de la radiación entrante, es la responsable del color azul del cielo.

Observamos que la columna de humo era más claramente identificable en los compuestos *Day cloud convection*, *day cloud phase*, *day land cloud*, *day land cloud fire*, *natural color*, y *true color* (ver Figura 3.5).

El producto *day cloud convection*, es un RGB simple que usa las bandas 2 (0.64 μ m) del visible y 13 (10.3 μ m) de infrarrojo en un compuesto (13, 2, 13). La contribución del visible se relaciona con la iluminación y reflectancia de las características de la superficie y las nubes, mientras que el infrarrojo se relaciona con la temperatura. La combinación ayuda a distinguir entre nubes altas y bajas (RAMMB, 2023).

Day cloud phase se emplea para evaluar la fase del agua de los topes de nubes que se enfrían, con el fin de vigilar el inicio de la convección, el crecimiento de la tormenta y el decaimiento. Este RGB aprovecha las diferencias en la reflectancia de la nube entre el canal visible (banda 2) e infrarrojo cercano (banda 5) y las varianzas de temperatura entre la superficie y las nubes en el infrarrojo (banda 13) en un compuesto (13, 2, 5), con el fin de proporcionar un mayor contraste entre las superficies del entorno y las fases del agua en la nube (es decir, agua vs hielo) (RAMMB, 2023).

Los compuestos *day land cloud* y *day land cloud fire* son dos RGB muy similares, usan las bandas 2 del visible y la banda 3, 5 y 6 del infrarrojo cercano en un compuesto (5,3,2) y (6,3,2), respectivamente. Son útiles para discriminar nubes de agua/hielo y para identificar nubes bajas de altas. Nubes altas de hielo, nieve y hielo marino aparecen en tono cian, mientras que las nubes bajas de agua aparecen en tonos grises apagados o blancos que podrían ser columnas de humo. Las superficies de tierra/océano tienen los colores esperados (aunque no sean colores reales). En el compuesto *day land cloud fire* resaltan los puntos calientes de incendios con un color de píxel rojo (RAMMB, 2023).

En el compuesto *true color*, se muestra una imagen que es aproximadamente como se vería desde el espacio exterior. Con ABI, el desafío de crear un color real surge por la falta de una banda Verde. Por lo tanto, se aproxima el verde combinando las bandas Azul (0.47 μ m), Roja (0.64 μ m) y 'Veggie' (0.86 μ m) (ver Ec. 2.1) en un compuesto (2, G, 1). El uso de la banda Veggie es importante porque imita la reflectividad mejorada presente en la banda Verde. En este compuesto se ve más claramente la columna de humo en colores grises, a diferencia de las nubes que se ven en color blanco.

3.1. Diseño del conjunto de datos de columnas de humo

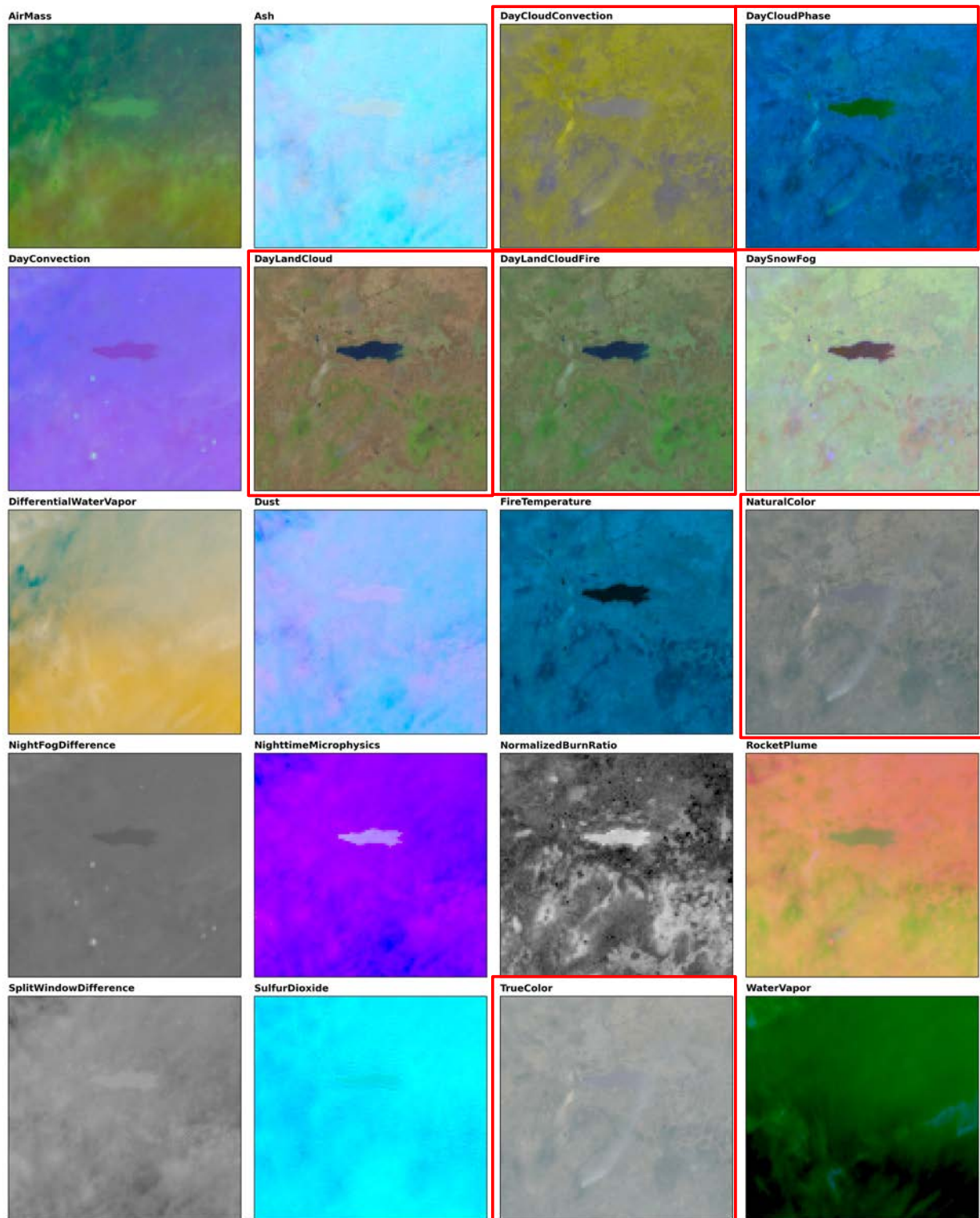


Figura 3.5: Acercamiento a una columna de humo de un incendio forestal, coordenadas geográficas: (-104, -102, 19, 21) del día 1 de mayo de 2022 a las 18:47 UTC en diversos compuestos generados con las bandas de GOES-16 . En un recuadro rojo se resaltan los compuestos donde se observan mas claramente la columna de humo. Elaboración propia

Las bandas utilizadas en los compuestos donde se observó la presencia de humo fueron las bandas 1, 2, 3, 5 y 13. Sin embargo, el compuesto de color verdadero fue el que mejor evidenció la columna de humo, y es interesante notar que este compuesto usa las bandas 1, 2 y 3 (La 3 en una menor proporción) (Ec. 2.1). Esto concuerda con los resultados obtenidos al analizar las bandas de manera individual. Por consiguiente, en esta primera aproximación, se optó por emplear imágenes RGB en forma de compuesto de color verdadero.

3.1.4. Descarga de las imágenes GOES-16 y procesamiento

Las imágenes utilizadas en esta investigación fueron las del producto CMIP de GOES-16/ABI, en la región de escaneo de CONUS, con una resolución temporal de 5 minutos, estas fueron descargadas del sitio web de “GOES dataset on Amazon Web Services (AWS)” (https://home.chpc.utah.edu/~u0553130/Brian_Blaylock/cgi-bin/goes16_download.cgi). El cual, ofrece acceso libre y casi en tiempo real al público en general. Para descargar y procesar las imágenes de ABI se utilizó el lenguaje de programación Python y las principales librerías y paquetes fueron los siguientes:

- GOES-2-Go: Para descargar los archivos NetCDF¹ de la serie GOES-16 del archivo de Amazon Web Services.
- NetCDF: Es un módulo que ayudó a implementar las funciones de leer y escribir archivos netCDF.
- Numpy: Se usó para realizar operaciones matemáticas y manipulación de matrices de datos. Estas operaciones y funciones de NumPy se emplean para procesar y manipular los datos de las bandas del *dataset* GOES-16, ejecutando transformaciones y ajustes necesarios para generar las imágenes de color verdadero.
- Cv2: es una biblioteca de código abierto utilizada para el procesamiento de imágenes y visión por computadora. Se empleó para efectuar operaciones como ajuste de contraste, potenciación de la imagen y manipulación de los canales de color.
- Gdal: (*Geospatial Data Abstraction Library*) es una biblioteca empleada para leer, escribir y manipular datos geoespaciales en diferentes formatos. Sirvió para crear y recortar archivos GeoTIFF, definir su proyección y guardar los datos en las bandas correspondientes.

El código python elaborado para la descarga y procesamiento de las imágenes GOES-16 se puede consultar en: https://github.com/colvertG/unet-goes16-segmentacion-columnas-humo/blob/main/01-creación-dataset/01_goes16_netcdf_2_png_auto.ipynb

1 NetCDF (*Network Common Data Form*) es un formato de archivo destinado a almacenar datos científicos multidimensionales.

Con los incendios seleccionados anteriormente (Tabla 3.1), su temporalidad, las bandas y compuesto RGB con el que íbamos a trabajar, procedimos a realizar la descarga y procesamiento de las imágenes de GOES-16. El resultado fueron 1061 imágenes de 100 x 100 píxeles en color verdadero y formato .png, en las que se puede observar una columna de humo (Figura 3.6).

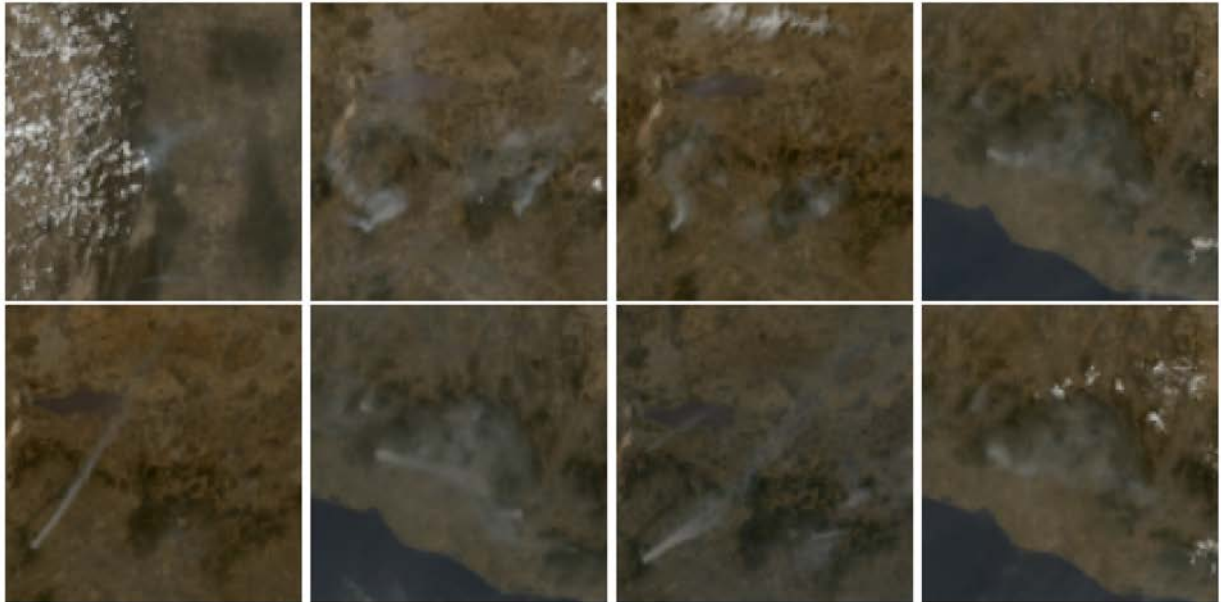


Figura 3.6: Ejemplo aleatorio de algunas de las imágenes pertenecientes al conjunto de datos generado. Estas imágenes muestran columnas de humo generadas a raíz de incendios forestales. Elaboración propia.

3.1.5. Segmentación manual

Para realizar el proceso de la segmentación manual de cada una de las 1061 imágenes del conjunto de datos generado anteriormente, se utilizó Roboflow (<https://roboflow.com/>).

Roboflow es una plataforma web que ofrece herramientas y servicios para la preparación y despliegue de conjuntos de datos para entrenar modelos de aprendizaje automático, especialmente en el ámbito de la visión por computadora. Algunas de las características y funcionalidades comunes de Roboflow son generación de etiquetas y anotaciones, y entrenamiento de modelos.

En este caso solo se empleó para la creación de las máscaras de humo, para esto se importaron las 1061 imágenes generadas, se crearon dos clases: “*smoke*” la cual se le asignó el valor de 1 y “*no smoke*” con valor 0 y se procedió a segmentar cada una de las columnas de humo que aparecían en las imágenes con sus herramientas de creación de polígonos. Una vez segmentadas las 1061 imágenes se hizo una partición de los datos en 70% para entrenamiento

(731 imágenes), 20% para validación (207 imágenes) y del 10% para prueba (123 imágenes). Un ejemplo de las máscaras generadas se puede observar en la Figura 3.7.

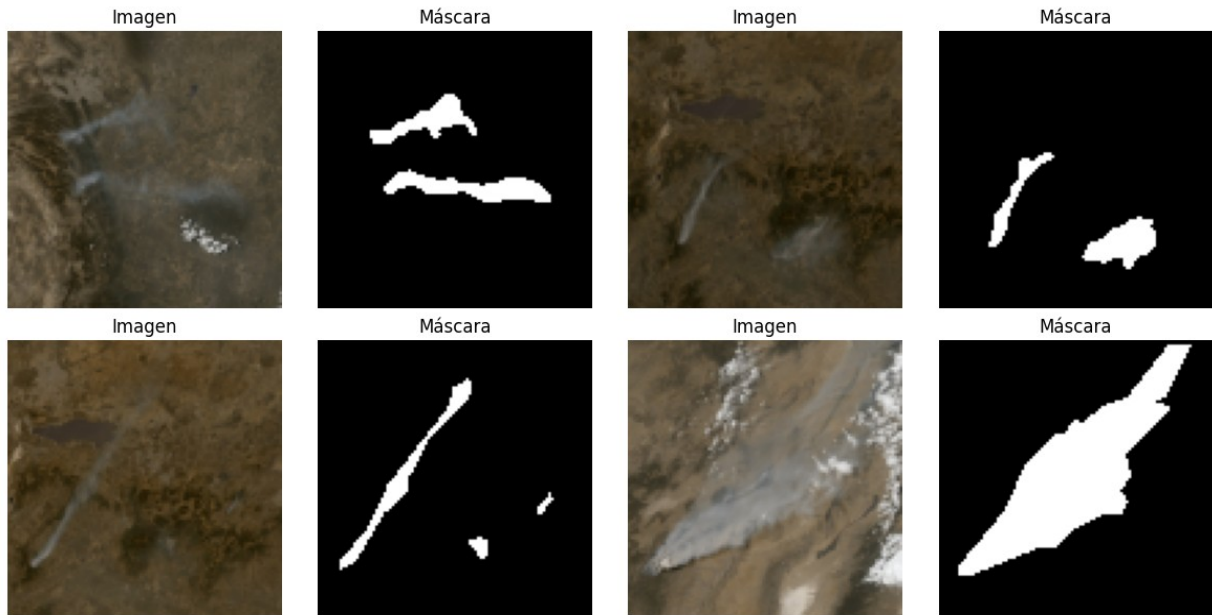


Figura 3.7: Ejemplo de imágenes y su respectiva máscara segmentada a mano. Elaboración propia.

3.2. Modelo basado en U-Net

En este subcapítulo, se aborda la construcción y la arquitectura del modelo basado en la U-Net original, utilizado para abordar el problema de la detección de columnas de humo de incendios forestales en imágenes satelitales de GOES-16. Se describe en detalle cómo se diseña y se configura el modelo, incluyendo las capas, los parámetros y las técnicas específicas empleadas. Además, se discute la justificación detrás de la elección de la arquitectura, sus hiperparámetros y cómo esta se adapta a las características del problema.

El uso de la arquitectura U-Net en lugar de otras arquitecturas de segmentación semántica para detectar columnas de humo de incendios forestales puede estar justificado por varias razones:

- Arquitectura especializada: U-Net es una arquitectura de red neuronal convolucional profundamente especializada para tareas de segmentación semántica y la generación de máscaras de píxeles. Su estructura de codificador-decodificador es eficaz para capturar detalles y características en diferentes escalas espaciales, lo que es beneficioso para la segmentación de objetos como columnas de humo.

- Recuperación de detalles: Las columnas de humo en imágenes de incendios forestales pueden tener detalles sutiles y variaciones en su forma y textura. U-Net, con su arquitectura de "salto de conexión", permite la recuperación de detalles finos, lo que puede ser esencial para una segmentación precisa.
- Capacidad de generalización: U-Net es conocida por su capacidad de generalización en tareas de segmentación semántica. Puede aprender a reconocer patrones y características relevantes en diferentes contextos y condiciones de iluminación, lo que es importante cuando se trabaja con imágenes de incendios forestales que pueden variar ampliamente.
- Eficiencia computacional: La U-Net, en comparación con algunas arquitecturas más recientes y complejas, tiende a ser más eficiente en términos computacionales. Esto nos permite que pueda ejecutarse en equipos de cómputo con requisitos moderados sin necesidad de una infraestructura de hardware extremadamente potente, lo que puede ser beneficioso en entornos donde los recursos computacionales son limitados o se busca un equilibrio entre el rendimiento y la eficiencia como es el caso.
- Historial de éxito: Como se mencionó en los capítulos 1.4 y 2.2.4.4, la U-Net ha demostrado su eficacia en numerosas aplicaciones de segmentación semántica, como la segmentación de órganos médicos en imágenes de resonancia magnética y la de columnas de humo de distintos tipos de imágenes, lo que respalda su idoneidad para tareas similares.

Sin embargo, es relevante señalar que la elección de la arquitectura de red neuronal óptima para una tarea tan específica como la de segmentar columnas de humo de imágenes satelitales no es una tarea sencilla. Otras arquitecturas también pueden ser adecuadas y la elección debe basarse en experimentos y pruebas de rendimiento en el conjunto de datos y la tarea específicos. No obstante, debido a la influencia de investigaciones previas en los antecedentes, el éxito documentado de la arquitectura U-Net, además de las limitaciones de tiempo que restringen la capacidad de adaptar o evaluar exhaustivamente otras alternativas, hemos optado por centrarnos en su implementación.

Se partió del modelo original de la U-Net, con el mismo número de filtros de convolución y de capas de agrupación máxima, pero se agregó un valor de relleno de 1 para no reducir la dimensión de los datos de salida y se remplazaron las convoluciones transpuestas por un proceso de "up-sample"¹ con una interpolación bicúbica seguido de una convolución de 1×1 (Cantoral Ceballos, 2023). Esto porque al agregar ceros en las convoluciones transpuestas tradicionales se genera ruido y se pierde mucha información en la imagen, lo que se reduce al aplicar una

1 *Up-sample*: Se refiere a la acción de incrementar el tamaño de una imagen o conjunto de datos, generalmente utilizando técnicas de interpolación.

interpolación para incrementar el tamaño de la imagen. Es importante recordar que la convolución transpuesta original usa un kernel y dicho kernel tiene parámetros que la red neuronal aprende, y al sustituirla por una interpolación bicúbica ya no tendríamos dichos parámetros por aprender en la parte de la expansión, por eso, seguido del *up-sample* se aplica una convolución de 1×1 para que haya parámetros que nos permitan abstraer o localizar el detalle de lo que estamos segmentando.

Las librerías que se utilizan fueron las ya mencionadas más las siguientes:

- Pytorch: Es una biblioteca de aprendizaje profundo desarrollado por *Facebook's AI Research lab* (FAIR) que ofrece una manera flexible y eficiente de construir, entrenar y desplegar modelos de aprendizaje automático y redes neuronales. Es ampliamente utilizado tanto en la academia como en la industria para una variedad de aplicaciones de inteligencia artificial.
- PIL: (y su bifurcación Pillow) es una biblioteca en Python que permite trabajar con imágenes, realizar operaciones de manipulación, transformación y guardar imágenes en diferentes formatos.
- Pandas: Es una biblioteca para el análisis de datos en Python. Facilita la manipulación y el procesamiento de datos tabulares.

3.2.1. Aumento de datos

El aumento de datos (*data augmentation*, en inglés) es una técnica utilizada en visión por computadora para incrementar la cantidad y diversidad de los datos de entrenamiento mediante la aplicación de transformaciones a las imágenes y sus respectivas máscaras de segmentación. Esta técnica es especialmente útil cuando se trabaja con conjuntos de datos limitados, ya que ayuda a mejorar la capacidad del modelo para generalizar y realizar predicciones precisas en nuevas imágenes. Algunas de las transformaciones comunes que se pueden aplicar en el aumento de datos incluyen:

- Rotación: Girar la imagen y la máscara alrededor de su centro.
- Reflejo horizontal/vertical: Voltar la imagen y la máscara horizontal o verticalmente.
- Cambio de tamaño: Aumentar o reducir el tamaño de la imagen y la máscara.
- Corte y recorte: Tomar una parte de la imagen y la máscara, o recortar áreas específicas.
- Cambios de color y brillo: Alterar los valores de color y brillo de la imagen, mientras se mantiene la coherencia con la máscara.

Es importante resaltar que las transformaciones deben ser aplicadas de manera coherente en las imágenes y sus máscaras para que la correspondencia entre píxeles y etiquetas se mantenga válida.

El aumento de datos es ampliamente conocido como una técnica simple pero sorprendentemente efectiva de regularización (Balestriero et al., 2022; Y. Wang et al., 2022). La regularización es una técnica que busca prevenir el sobreajuste (*overfitting* en inglés) de un modelo. El sobreajuste ocurre cuando la complejidad del modelo es demasiado alta y memoriza o aprende los datos de entrenamiento, lo que lleva a un bajo error en los datos de entrenamiento, pero un alto error en los datos de prueba, a medida que las iteraciones aumentan (Figura 3.8).

La regularización ayuda a ajustar modelos complejos para que generalicen mejor a datos nuevos y no vistos.

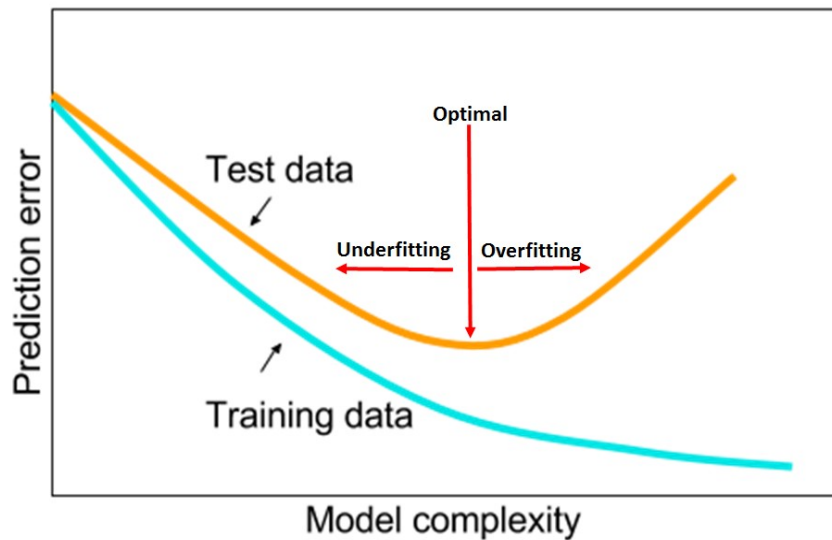


Figura 3.8: Representación gráfica de ajuste insuficiente (*underfitting*) y sobreajuste (*overfitting*). El eje x representa la complejidad del modelo y el eje y el error. Tomada de Smith, (2018).

Se procedió a evaluar el modelo definitivo mediante tres configuraciones distintas: en primer lugar, sin aplicar aumento de datos; en segundo lugar, empleando aumento de datos una vez; y en tercer lugar, aplicando aumento de datos dos veces.

Las transformaciones solo fueron aplicadas al conjunto de datos de entrenamiento y fueron las siguientes: Una rotación aleatoria de entre -15° a 15° , un reflejo horizontal y vertical con el 50% de probabilidad de ocurrencia y un ajuste del brillo de la imagen aleatorio de entre 0.7 y 1.3. Los resultados en el conjunto de datos fue que duplicamos su tamaño del conjunto de entrenamiento, pasando de 731 a 1462, en el caso de un aumento, y se triplicó, llegando a un total de 2193 imágenes, en el caso de dos aumentos.

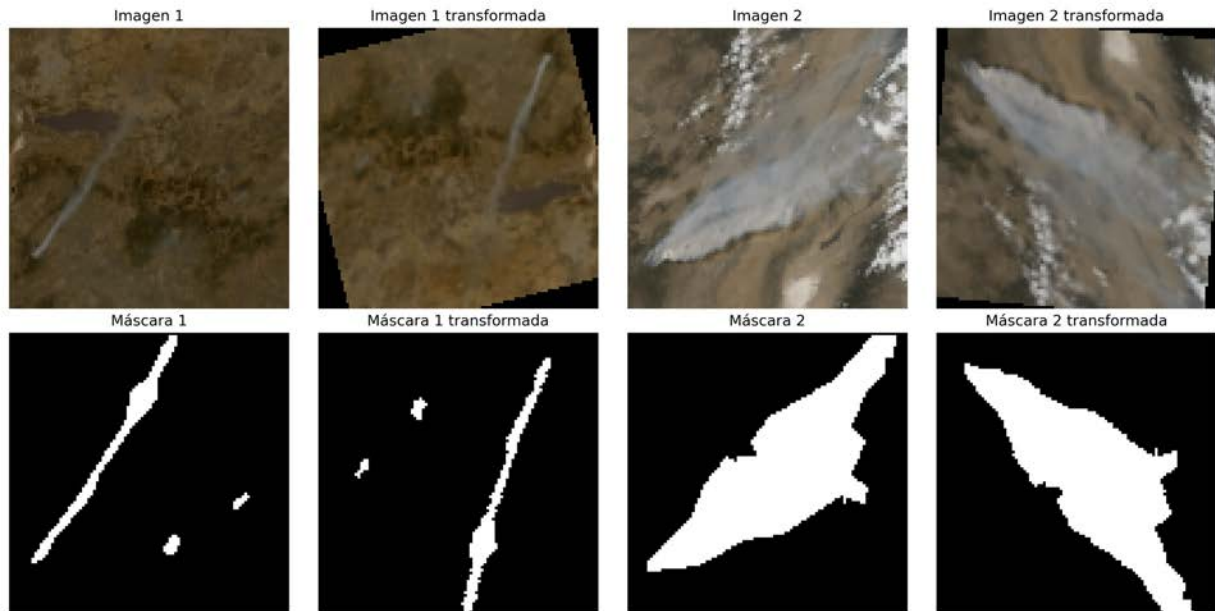


Figura 3.9: Ejemplo de transformaciones aleatorias aplicadas a imágenes del conjunto de entrenamiento, se aplica la misma transformación a la imagen y su máscara correspondiente. Elaboración propia.

El código de python elaborado para el aumento de datos se puede consultar en: https://github.com/colvertG/unet-goes16-segmentacion-columnas-humo/blob/main/01-creación-dataset/02_aumento_datos.ipynb

3.2.2. Dataset y DataLoader

Aun en la actualidad, donde la accesibilidad a unidades de procesamiento gráfico (GPU, *Graphics Processing Unit*) se ha simplificado considerablemente, ya sea a través de opciones económicas o incluso gratuitas, como en el caso de Google Colab (<https://colab.research.google.com/>), la cantidad de datos necesarios para entrenar modelos de aprendizaje automático de manera eficiente puede llegar a ser muy grande y al intentar abrir todos los datos en memoria conllevaría a un fallo por agotamiento de recursos. Por lo tanto, se usaron las estructuras de *dataset* y *dataloader* de PyTorch para afrontar este problema.

Un *dataset* es una estructura que almacena y proporciona acceso a los datos utilizados para entrenar y evaluar modelos. Puede contener ejemplos de entrada y sus correspondientes etiquetas o salidas. Está diseñado para interactuar con el modelo de manera eficiente y proporcionar los datos necesarios durante la medida que avanza el proceso de entrenamiento, validación o prueba (*Datasets & DataLoaders — PyTorch Tutorials 2.0.1+cu117 documentation, 2023*).

Un *dataloader* es una herramienta de PyTorch que facilita la carga y la administración de datos desde un *dataset*. Un *dataloader* se encarga de dividir los datos en lotes, realizar la carga de los datos en la memoria o en la GPU de tener disponible, y ofrecer iteradores que permiten al modelo acceder a los lotes de datos de manera eficiente durante el proceso de entrenamiento.

3.2.3. Optimización de hiperparámetros

Seleccionar los hiperparámetros óptimos para un modelo de U-Net, o cualquier otro modelo de aprendizaje profundo, es de suma importancia para lograr un rendimiento adecuado en tareas específicas. En este punto es importante el conjunto de validación, ya que se utiliza para comparar el rendimiento de diferentes configuraciones de hiperparámetros sin involucrar los datos de prueba y con base en ese análisis se determinaran los hiperparámetros de la U-Net

Una vez que seleccionó el conjunto de hiperparámetros óptimos, se entrena el modelo con ellos y se evalúa con esos hiperparámetros en el conjunto de prueba. Esto nos brinda una idea más precisa de cómo se comportará en datos no conocidos. Una descripción más a fondo de los hiperparámetros optimizados en este capítulo se puede consultar en el capítulo 2.2.5. Hiperparámetros.

El código con el que se generaron los resultados mostrados en este subcapítulo se puede consultar en: <https://github.com/colvertG/unet-goes16-segmentacion-columnas-humo/tree/main/02-optimización-hiperparámetros>

3.2.3.1. Número de capas o convoluciones iniciales

Tal como se mencionó en el capítulo 2.2.4.4 dedicado a la arquitectura U-Net de este documento, el número original de convoluciones iniciales es de 64 y va incrementando al doble conforme avanza en el camino de contracción. Como parte de la optimización, se llevó a cabo un ejercicio en el cual se entrenó el modelo con distintos valores iniciales de convolución, abarcando valores de 8, 16, 32 y 64, a lo largo de un ciclo de entrenamiento compuesto por 20 épocas y tamaño de lote de 16. Es relevante destacar que las métricas de evaluación se calcularon con base en el conjunto de datos de validación, el cual proporcionó una evaluación imparcial al modelo, dado que no había sido previamente expuesto a dichos datos.

Los resultados obtenidos a partir de este análisis, que se presentan gráficamente en la Figura 3.10 y la Figura 3.11, ofrecen una clara representación de las tendencias observadas en cuanto a la función de pérdida e IoU. Estos resultados indican que el rendimiento más óptimo se alcanza cuando se emplean 64 capas iniciales de convolución.

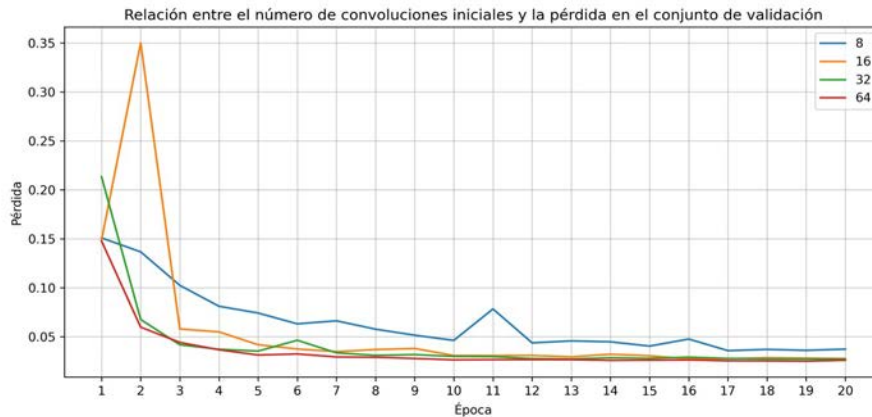


Figura 3.10: Comportamiento de la pérdida en el conjunto de validación durante el entrenamiento del modelo con diferentes convoluciones iniciales (8, 16, 32, 64). Elaboración propia.

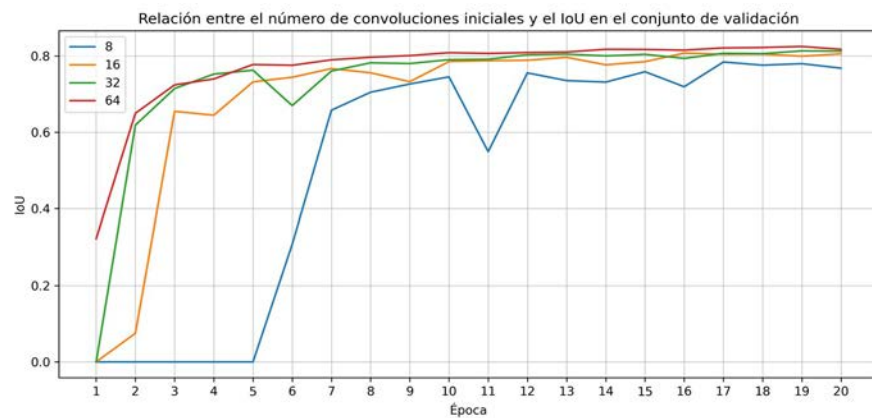


Figura 3.11: Comportamiento del IoU en el conjunto de validación durante el entrenamiento del modelo con diferentes convoluciones iniciales (8, 16, 32, 64). Elaboración propia.

3.2.3.2. Función de activación

Con el propósito de determinar la función de activación más adecuada para nuestro modelo U-Net, se llevó a cabo un proceso de entrenamiento con la siguiente configuración: convoluciones iniciales de 32, tamaño de lote de 16 y ejecutando el entrenamiento a lo largo de 20 épocas. La variable que se manipuló en este experimento fue exclusivamente la función de activación.

Los resultados derivados de este proceso de experimentación se han ilustrado en las Figuras 3.12 y 3.13, estas gráficas ofrecen una visión clara de cómo se comportaron diferentes funciones de activación durante todas las épocas del entrenamiento.

Se observa que las funciones de activación ReLU, LeakyReLU y ELU exhiben patrones de comportamiento bastante similares en las fases finales del entrenamiento, siendo LeakyReLU ligeramente superior. Por lo tanto, se ha determinado que la función de activación LeakyReLU será implementada en el modelo definitivo como la elección óptima para este componente clave de la arquitectura de la U-Net.

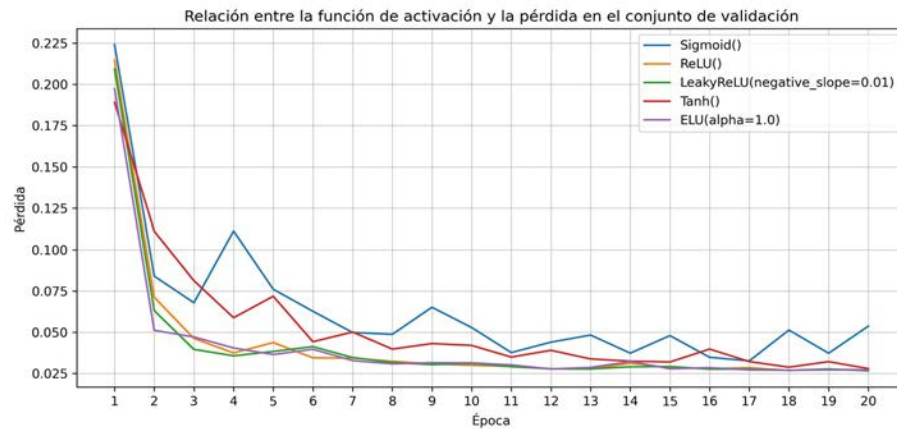


Figura 3.12: Comportamiento de la pérdida en el conjunto de validación durante el entrenamiento del modelo con diferentes funciones de activación. Elaboración propia.

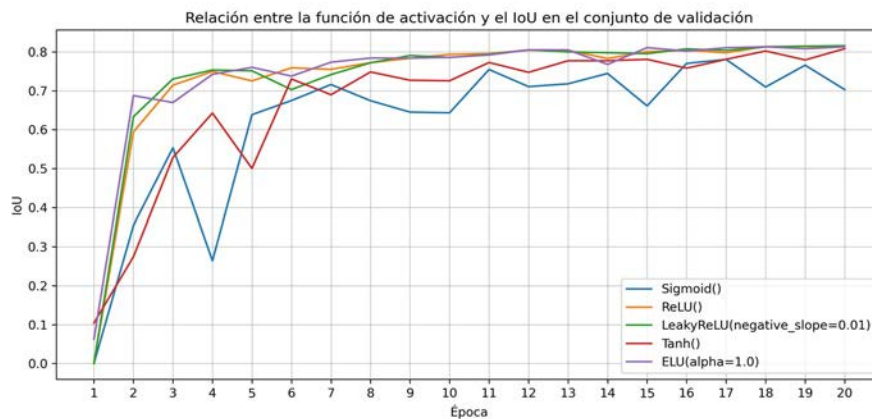


Figura 3.13: Comportamiento del IoU en el conjunto de validación durante el entrenamiento del modelo con diferentes funciones de activación. Elaboración propia.

3.2.3.3. Optimizador

En congruencia con la metodología aplicada anteriormente, se procedió a llevar a cabo un entrenamiento del modelo a lo largo de 20 épocas, con tamaño de lote de 16, una configuración inicial de convoluciones de 32, y función de activación LeakyReLU. Los resultados subsiguientes a la etapa de entrenamiento del modelo se calcularon en el conjunto de validación. Se emplearon

dos optimizadores distintos: SGD y Adam. Los resultados de dicha evaluación se presentan gráficamente en la Figura 3.14 y en la Figura 3.15.

De manera destacable, los hallazgos arrojados por el análisis comparativo de estos resultados indican que el optimizador SGD sobresale notoriamente en relación con Adam, en el contexto de la presente investigación. Estos resultados sugieren una superioridad significativa de SGD en términos de convergencia y rendimiento en la tarea de segmentación semántica bajo las condiciones experimentales establecidas.

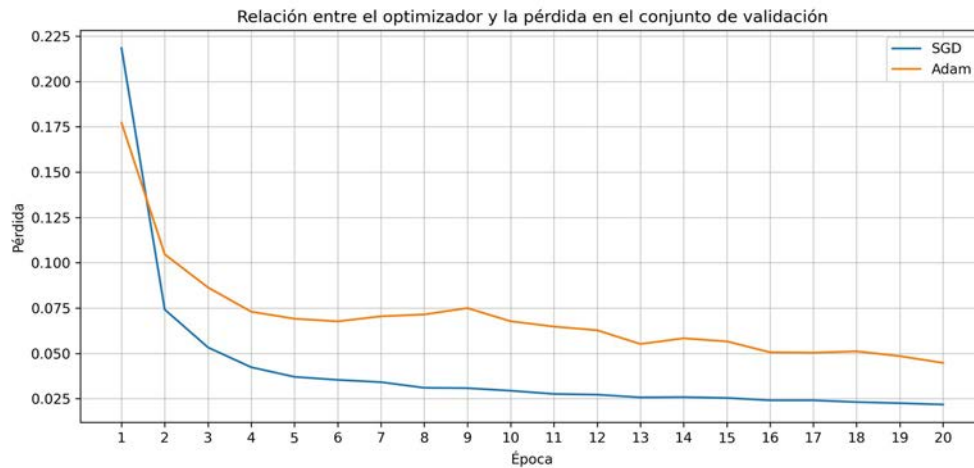


Figura 3.14: Comportamiento de la pérdida en el conjunto de validación durante el entrenamiento del modelo con diferentes optimizadores. Elaboración propia.

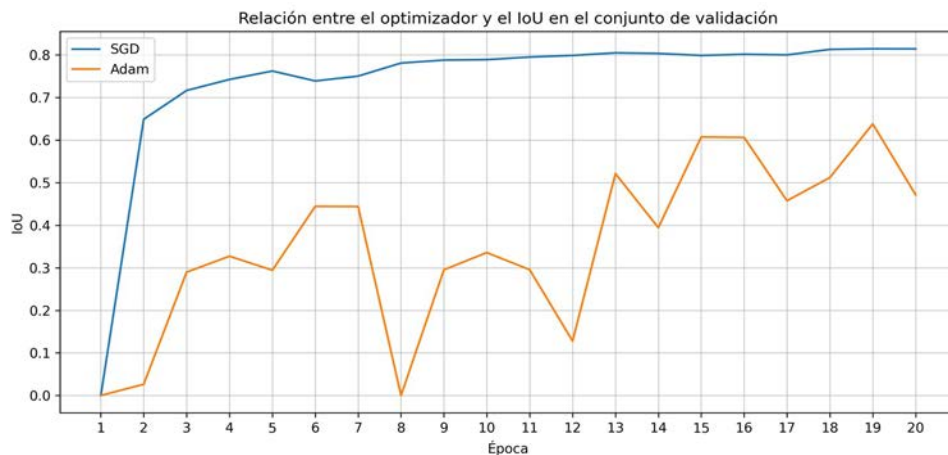


Figura 3.15: Comportamiento del IoU en el conjunto de validación durante el entrenamiento del modelo con diferentes optimizadores. Elaboración propia.

3.2.3.4. Tasa de aprendizaje

En lo que se refiere a la selección de la tasa de aprendizaje, se implementó el enfoque de super convergencia utilizando el método de One Cycle Learning Rate, como se expone en el capítulo 2.2.5.4 de este trabajo. El proceso se inició con la búsqueda de los valores máximo y mínimo de la tasa de aprendizaje. Para llevar a cabo esta tarea, se procedió a entrenar el modelo durante una sola época, variando la tasa de aprendizaje en cada iteración, oscilando en valores desde 1×10^{-6} hasta 1, los resultados se pueden apreciar en la Figura 3.16

Siguiendo la fundamentación teórica de este método, se optó por establecer el valor máximo de la tasa de aprendizaje en 1×10^{-1} , justo antes de alcanzar el mínimo global en la función de pérdida. Asimismo, el valor mínimo de la tasa de aprendizaje se fijó en una décima parte del valor máximo, es decir, en 1×10^{-2} . Esta selección se basa en la premisa de lograr un equilibrio entre una tasa de aprendizaje lo suficientemente alta para impulsar el proceso de entrenamiento y una tasa lo suficientemente baja para permitir una convergencia estable y eficiente en el modelo.

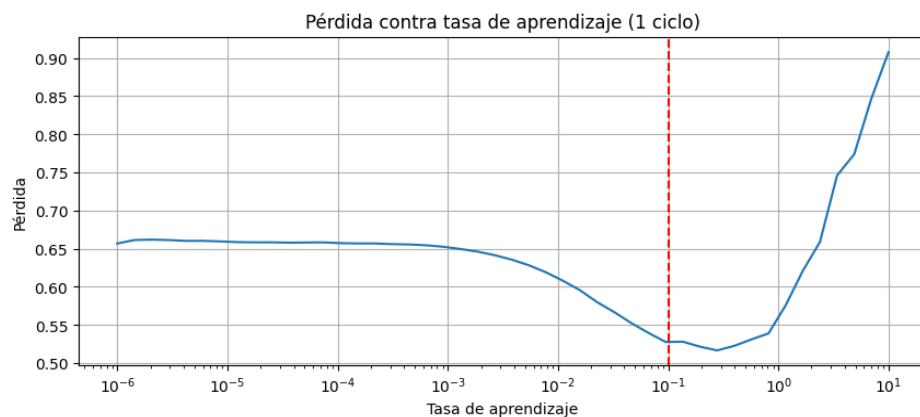


Figura 3.16: Función de pérdida durante el entrenamiento de una época variando la tasa de aprendizaje en cada iteración. Se utiliza escala logarítmica en el eje X para facilitar su visualización. Elaboración propia.

Con los valores máximos y mínimos de la tasa de aprendizaje determinados, en la Figura 3.17 se observa una representación gráfica que ilustra la evolución de esta tasa en cada iteración durante el proceso de entrenamiento del modelo final. Se aprecia claramente un aumento progresivo hasta alcanzar aproximadamente el 40% de las iteraciones, a partir de lo cual comienza a descender hasta llegar al valor mínimo. Posteriormente, en las últimas iteraciones, se observa una disminución más gradual hasta alcanzar un valor significativamente bajo.

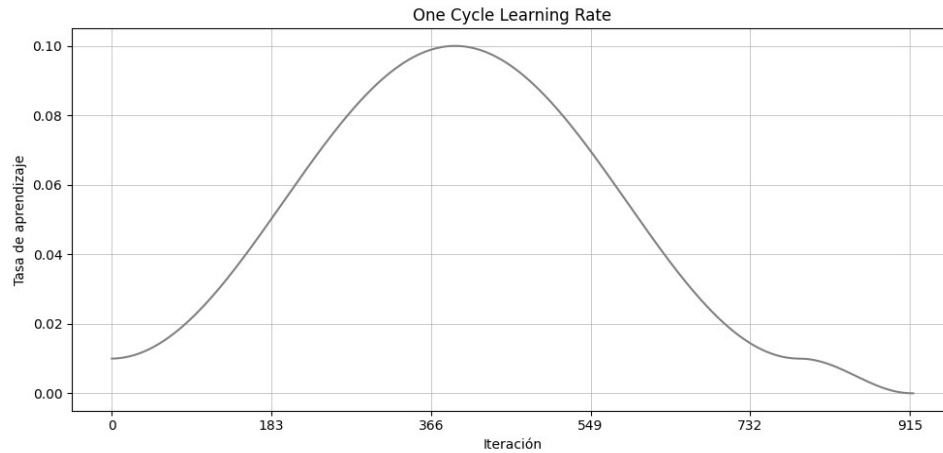


Figura 3.17: Ejemplo real de como varía la tasa de aprendizaje a lo largo de todo el entrenamiento usando el método de "One Cycle Learning Rate". Elaboración propia.

Con el propósito de validar la eficacia del enfoque implementado, se llevó a cabo un experimento comparativo mediante el entrenamiento del modelo durante 20 épocas. Se realizaron dos pruebas diferentes: una utilizando una tasa de aprendizaje estática fija en 0.01 y otra empleando el enfoque de One Cycle Learning Rate. Los resultados obtenidos de estas pruebas se representan gráficamente en la Figura 3.18 y la Figura 3.19

Los datos revelan de manera concluyente que la estrategia basada en One Cycle Learning Rate supera significativamente en términos de rendimiento a la técnica de tasa de aprendizaje estática. Este hallazgo sugiere que la adaptación dinámica de la tasa de aprendizaje contribuye de manera efectiva a la aceleración de la convergencia del modelo y a la mejora general de la optimización durante el proceso de entrenamiento.

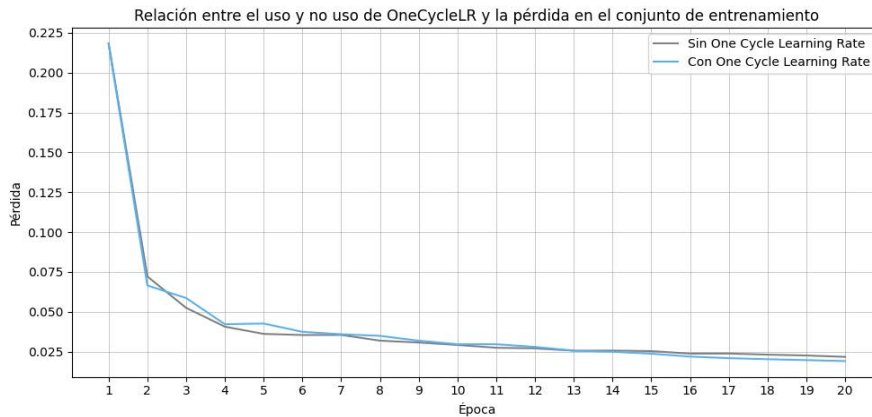


Figura 3.18: Comportamiento de la pérdida en el conjunto de validación durante el entrenamiento del modelo. La línea gris representa el costo obtenido al utilizar una tasa de aprendizaje estático de 0.01, mientras que la línea azul corresponde al costo obtenido al aplicar el método de "One Cycle Learning Rate" (One Cycle LR). Elaboración propia.

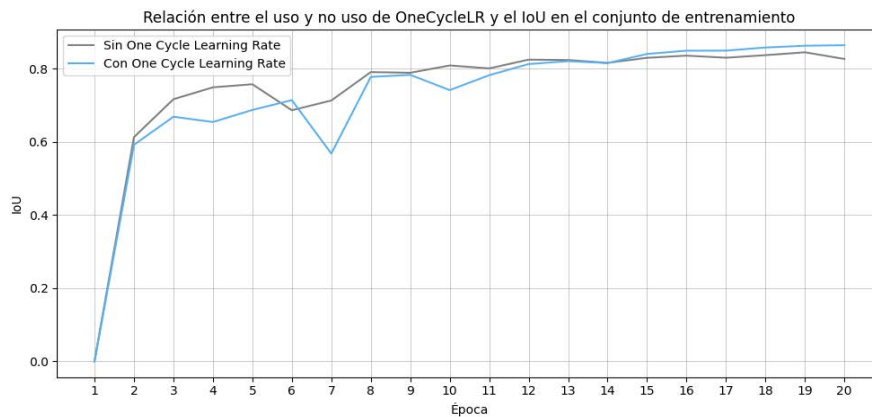


Figura 3.19: Comportamiento del IoU en conjunto de validación durante el entrenamiento del modelo. La línea gris representa el costo obtenido al utilizar un learning rate estático de 0.01, mientras que la línea azul corresponde al IoU obtenido al aplicar el método de "One Cycle Learning Rate" (One Cycle LR). Elaboración propia.

3.2.3.5. Tamaño de lote

Con el propósito de determinar el tamaño de lote óptimo, se llevaron a cabo múltiples entrenamientos variando el tamaño de lote, los valores probados son 4, 8, 16 y 32. Esta selección se fundamentó en las limitaciones de la memoria del GPU disponible, así como en la necesidad de armonizar la eficiencia computacional con la convergencia del modelo.

En el transcurso de la experimentación, se ejecutaron iteraciones de entrenamiento consistentes en 20 épocas para cada tamaño de lote, evaluando el desempeño del modelo a través de métricas específicas, tales como la función de pérdida y el Índice de Jaccard (IoU), sobre el conjunto de datos de validación.

Las figuras resultantes (Figura 3.20 y Figura 3.21), presentan de manera diáfana las tendencias observadas en relación con la función de pérdida e IoU en función del tamaño de lote. Estos gráficos proveen una visión integral de cómo la arquitectura U-Net responde a diversas configuraciones de tamaño de lote durante el proceso de entrenamiento.

Los resultados destacan que, en virtud de las restricciones de memoria del GPU, el tamaño de lote de 32 se erige como la elección más eficaz en términos de rendimiento del modelo. Aunque se contemplaron tamaños de lote de magnitud superior, la limitación de la memoria obstaculizó su exploración exhaustiva. Este descubrimiento respalda la elección del tamaño de lote de 32 para el entrenamiento definitivo del modelo U-Net.

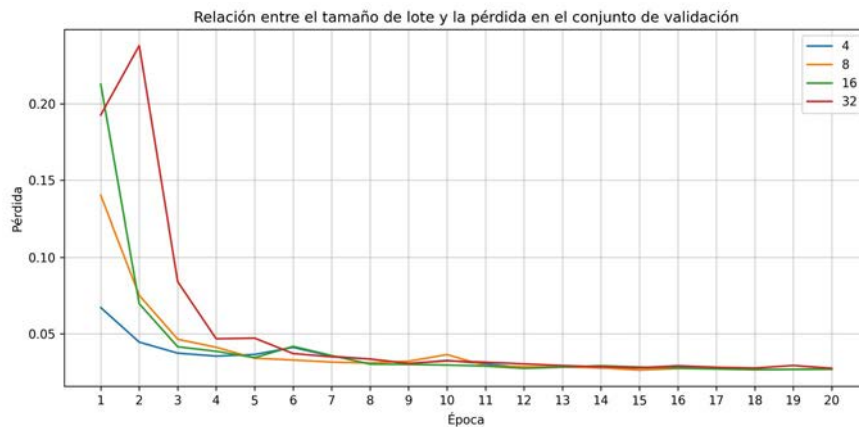


Figura 3.20: Comportamiento de la pérdida en el conjunto de validación durante el entrenamiento del modelo con diferentes tamaños de lote. Elaboración propia.

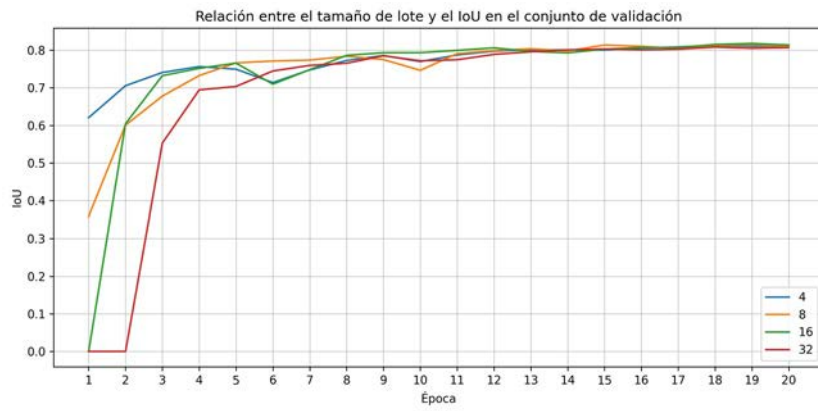


Figura 3.21: Comportamiento del IoU en el conjunto de validación durante el entrenamiento del modelo con diferentes tamaños de lote. Elaboración propia.

Capítulo 4. Resultados y discusión

En este capítulo, se presentan y analizan los resultados obtenidos a través de la investigación realizada. Los experimentos y análisis se llevaron a cabo con el fin de abordar los objetivos particulares en el capítulo de introducción. Se presentan las métricas de evaluación, las comparaciones entre diferentes enfoques y se analizan los hallazgos más relevantes.

El código con el que se entrenó el modelo, se validó y se generaron los resultados mostrados en este capítulo se puede consultar en: <https://github.com/colvertG/unet-goes16-segmentacion-columnas-humo.git>

4.1. Descripción General de los Resultados

En esta sección, se proporciona una visión panorámica de los resultados alcanzados a través del proceso de investigación. Los resultados presentados reflejan la efectividad de las técnicas aplicadas en la detección y segmentación de incendios forestales utilizando imágenes del satélite GOES-16. Estos resultados representan un paso significativo hacia la mejora de la capacidad de monitoreo y predicción de incendios forestales, lo que tiene un impacto directo en la gestión y mitigación de estos eventos devastadores.

Para construir la base de datos, se recopilaron y segmentaron a mano un total de 1061 imágenes de incendios forestales capturadas por el satélite GOES-16. La segmentación manual permitió obtener máscaras precisas de las regiones de humo en cada imagen, lo que sirvió como conjunto de datos de referencia para el entrenamiento, evaluación y prueba de los modelos.

Para abordar el desafío de la limitada cantidad de datos, se implementó la técnica de aumento de datos. Esta estrategia permitió aumentar la diversidad del conjunto de datos al aplicar transformaciones como rotaciones, recortes y cambios de brillo a las imágenes de entrenamiento originales y sus correspondientes máscaras de segmentación. La validación de esta técnica demostró su capacidad para mejorar la generalización y la capacidad de detección del modelo, al reducir la tendencia al sobreajuste. El conjunto de datos final se puede descargar en: <https://www.kaggle.com/datasets/colvertgomez/goes16-wildfires-smoke-plumes-dataset>

La arquitectura U-Net se modificó y adaptó para mejorar su rendimiento en la tarea de segmentación de columnas de humo de incendios forestales. Se introdujo un valor de relleno de 1 en las capas convolucionales para preservar las dimensiones espaciales de las imágenes en el proceso de convolución. Además, se reemplazó la función de activación ReLU por la función

LeakyReLU, lo que permitió la propagación de gradientes incluso para valores de entrada negativos, lo que mejoró la capacidad del modelo para aprender características relevantes en regiones de baja intensidad.

Para optimizar el modelo, se seleccionó el optimizador SGD con momento, que se utilizó para ajustar los pesos del modelo durante el proceso de entrenamiento. Se implementó un método de super convergencia para determinar la tasa de aprendizaje óptimo.

En términos de las métricas, los resultados obtenidos son alentadores y prometedores. El modelo final logró un índice de Jaccard (IoU) de 0.8252 y un coeficiente Dice de 0.9042 en el conjunto de prueba, lo que indica una precisión sustancial en la superposición entre las regiones de humo predichas y las máscaras de referencia. Además, la métrica de precisión alcanzó un valor de 0.9898.

En términos de rendimiento, el modelo final tardó en entrenar las 20 épocas (920 iteraciones) 28 minutos y 21 segundos, usando una computadora con un procesador i5-12500H, tarjeta de video NVIDIA GeForce RTX 3050 Ti y 16 gb de RAM. Una vez entrenado el modelo, segmentar una nueva imagen de 128x128 píxeles le toma en promedio 2.3 segundos.

En resumen, los resultados presentados en este capítulo demuestran el éxito de la metodología aplicada para la detección y segmentación de columnas de humo de incendios forestales en imágenes del satélite GOES-16. Estos resultados abren el camino hacia un monitoreo más efectivo de los incendios forestales y la mitigación temprana de sus impactos.

4.2. Modelo final

El modelo final fue una U-Net con el mismo número de pasos o niveles, el mismo número inicial de convoluciones (64) y tamaño de lote de 32. Sin embargo, se hizo una modificación importante: se reemplazó la función de activación por LeakyReLU. Además, se agregó un valor de relleno de 1, se empleó el optimizador SGD con momento y se aplicó el método de One Cycle Learning Rate para ajustar la tasa de aprendizaje (Figura 4.1).

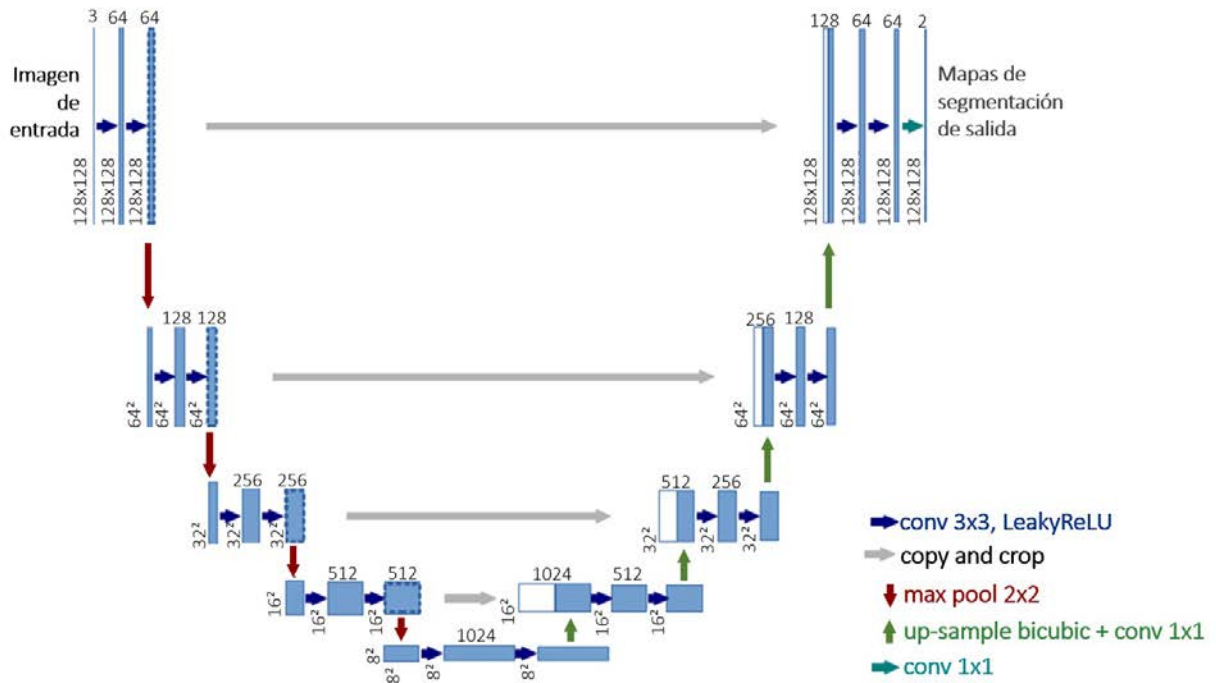


Figura 4.1: Diseño de la arquitectura U-Net final. Modificada de Ronneberger et al., (2015).

4.3. Aumento de datos

El modelo se entrenó en su totalidad usando solamente los datos del conjunto de entrenamiento. Los resultados sin aumento de datos y aplicando uno y dos aumentos se resumen en la Tabla 4.1 y las figuras 4.2 y 4.3, que muestra los valores de IoU, tanto en el conjunto de entrenamiento como en el conjunto de prueba, así como el coeficiente Dice y la función de pérdida al final del entrenamiento. Cuantitativamente, se observa una mejora en las métricas a medida que aumenta la cantidad de datos de entrenamiento, sin embargo, las mejoras son relativamente pequeñas, lo que indica que el modelo ya ha alcanzado un rendimiento significativo incluso sin el uso de aumentos de datos adicionales.

Aumento de datos	IoU entrenamiento	IoU prueba	Dice entrenamiento	Dice prueba	pérdida
0	0.8758	0.8213	0.9338	0.9019	0.0169
1	0.8873	0.8252	0.9403	0.9042	0.0155
2	0.8934	0.8245	0.9437	0.9038	0.0144

Tabla 4.1: Resultados del entrenamiento con cero, con uno y con dos aumentos de datos.

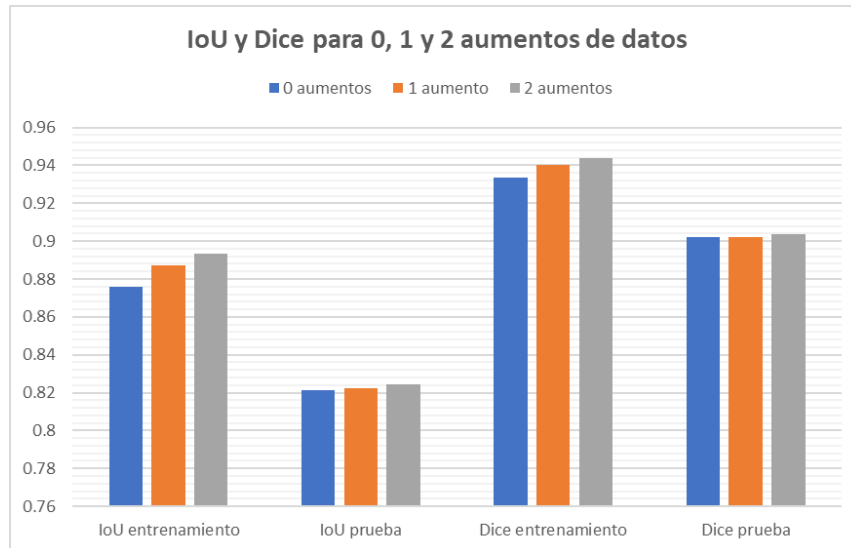


Figura 4.2: IoU y coeficiente Dice en los conjuntos de entrenamiento y prueba para 0, 1 y 2 aumentos de datos. Elaboración propia.

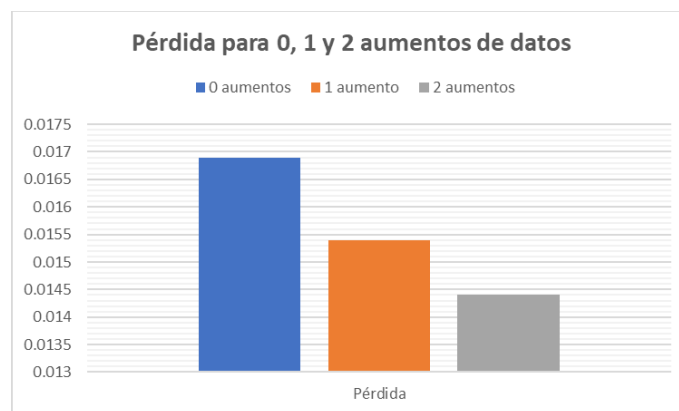


Figura 4.3: Pérdida final en el conjunto de entrenamiento para 0, 1 y 2 aumentos de datos. Elaboración propia

Desde una perspectiva cualitativa, se llevó a cabo la evaluación de la segmentación de los tres modelos utilizando imágenes del conjunto de prueba. Sin embargo, los resultados mostraron que

los tres modelos lograron una segmentación precisa y muy similar. Esto puede atribuirse en parte a la limitada variabilidad de los datos (Figura 4.5).

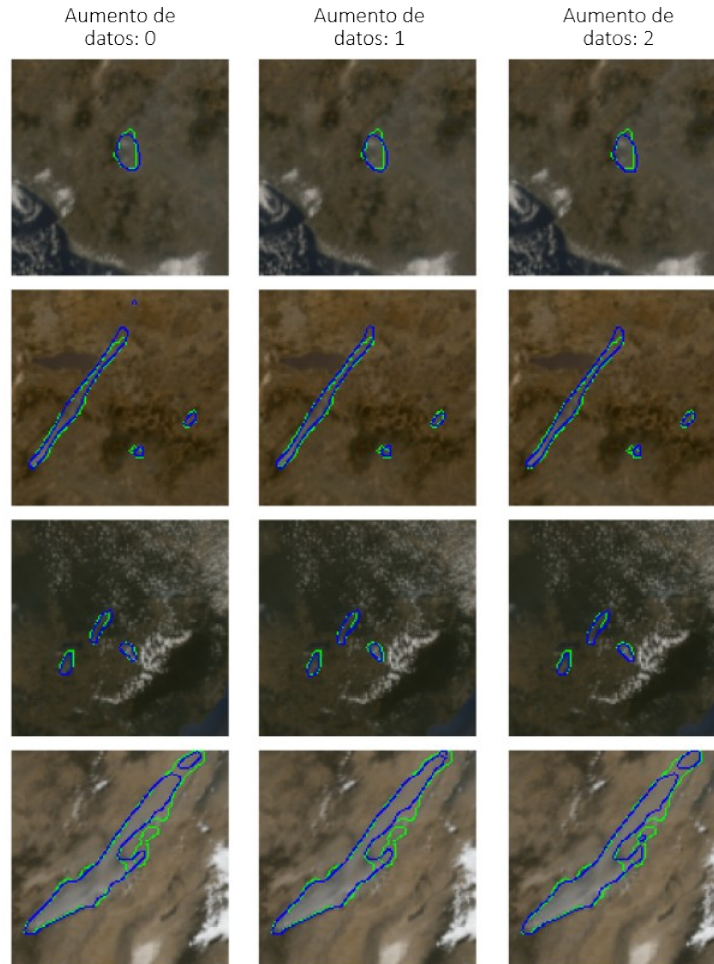


Figura 4.4: Ejemplos de segmentación utilizando imágenes del conjunto de prueba, tanto sin aumentos de datos como con uno y dos aumentos de datos. La segmentación verdadera se representa en verde, mientras que la segmentación predicha por el modelo se muestra en azul. Elaboración propia.

A la luz de los resultados previamente obtenidos, decidimos llevar a cabo pruebas adicionales un poco al límite de lo razonable, desafiando aún más los límites de la evaluación. Procedimos a someter los tres modelos a imágenes de columnas de humo que no formaban parte del conjunto de datos original, tomadas en horarios donde apenas va amaneciendo y en horarios de la tarde/noche, momentos en los cuales las nubes pueden ser confundidas fácilmente con las columnas de humo, incluso para la vista humana. La elección de estos horarios se basó en el

hecho de que los tres modelos segmentaban de manera efectiva en condiciones de iluminación constante durante el día.

Es importante señalar que las imágenes utilizadas en este ejercicio no estaban incluidas en el conjunto de datos original, por lo que no había garantía de una segmentación precisa en estos escenarios difíciles. No obstante, los resultados fueron alentadores, ya que revelaron que aun en las condiciones planteadas, la segmentación se mantuvo altamente efectiva, y solo se observaron discrepancias significativas en los horarios desafiantes, siendo un aumento de datos donde se observaron menos falsos positivos (Figura 4.5).

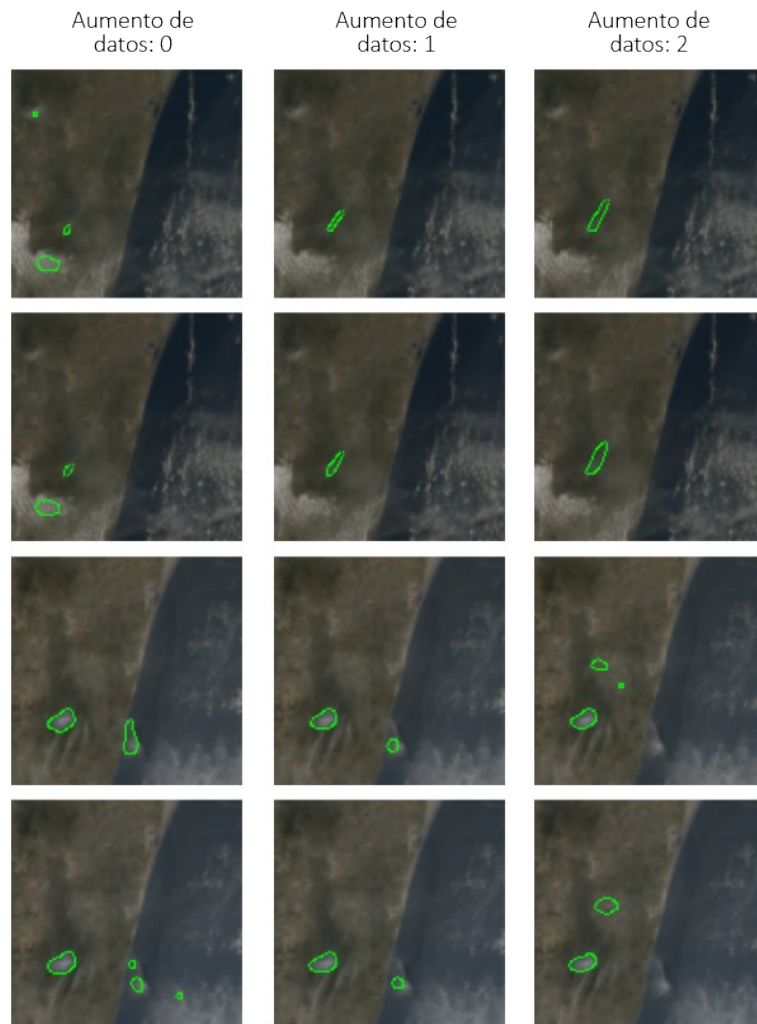


Figura 4.5: Ejemplos de segmentación utilizando imágenes que no pertenecen al conjunto de datos, tanto sin aumentos de datos como con uno y dos aumentos de datos. En verde se representa la segmentación predicha por el modelo. Elaboración propia.

4.4. Métricas resultantes del modelo final

Las métricas de Intersección sobre la Unión (IoU), el coeficiente Dice y la precisión del modelo final fueron calculadas por época, promediando los resultados de las 46 iteraciones realizadas en cada una de ellas. Estos resultados se representaron gráficamente, con el eje x mostrando la época y en el eje y indicando el valor de la métrica correspondiente (Figuras 4.6, 4.7 y 4.8).

Es notable que tanto el IoU como el coeficiente Dice exhiben un comportamiento muy similar en el transcurso de las épocas. Ambos comienzan con valores muy bajos en la primera época, experimentan un rápido incremento en la segunda época, y a medida que se acercan a la décima época, el incremento en la métrica se vuelve más pronunciado. Y por último, en las épocas finales, se observa una mayor separación entre las curvas de IoU entre entrenamiento y validación, a diferencia de Dice.

En contraste, la precisión muestra un comportamiento diferente desde la primera época, comenzando con una métrica notablemente alta y manteniéndose así a lo largo del proceso de entrenamiento.

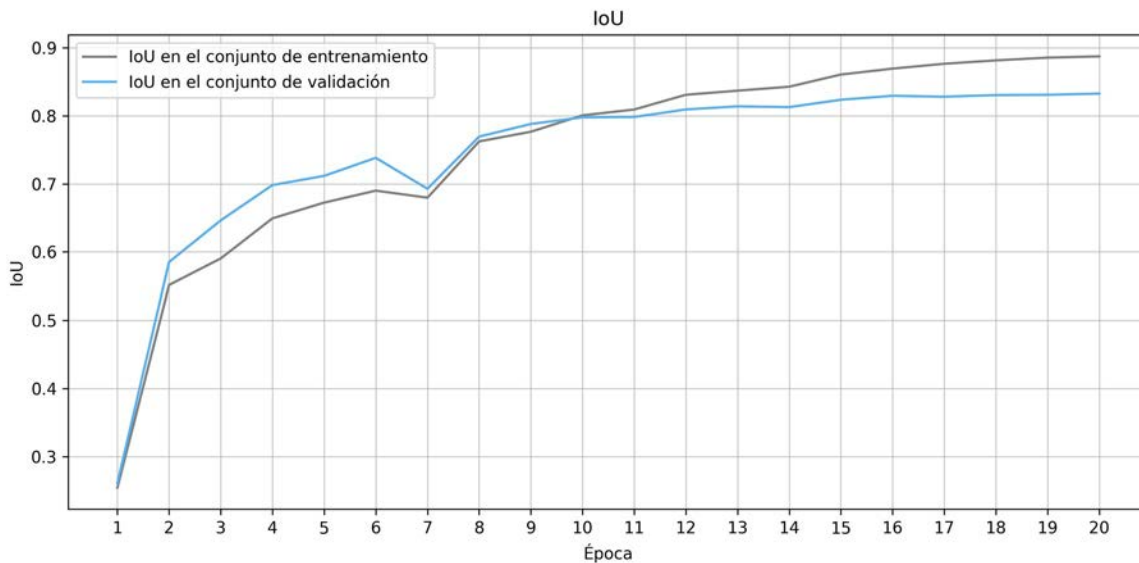


Figura 4.6: Comportamiento de la Intersección sobre la Unión (IoU) del modelo final en el conjunto de entrenamiento y validación para cada época. Elaboración propia.

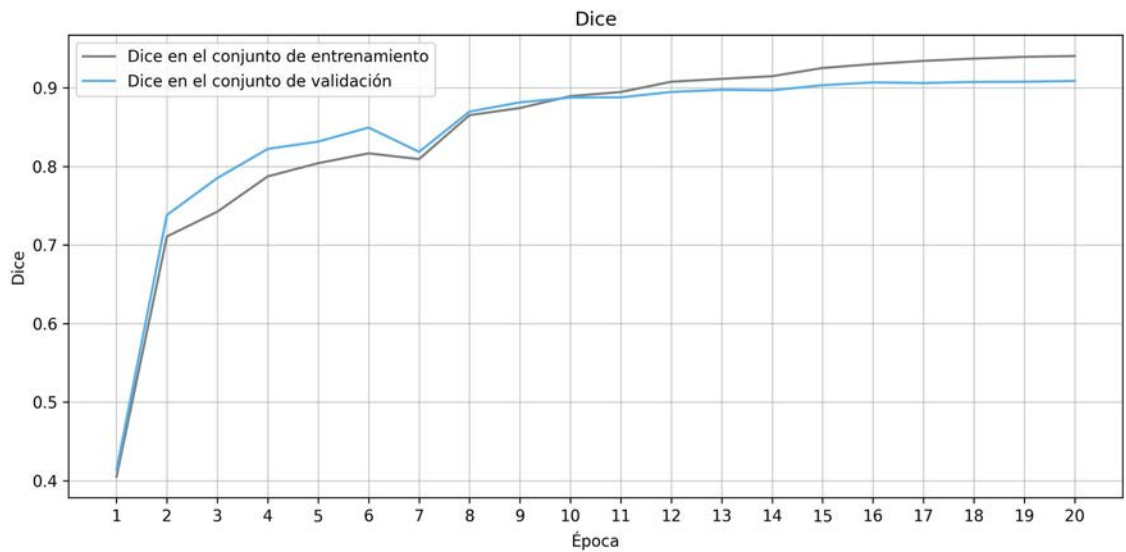


Figura 4.7: Comportamiento de coeficiente Dice del modelo final en el conjunto de entrenamiento y validación para cada época. Elaboración propia.

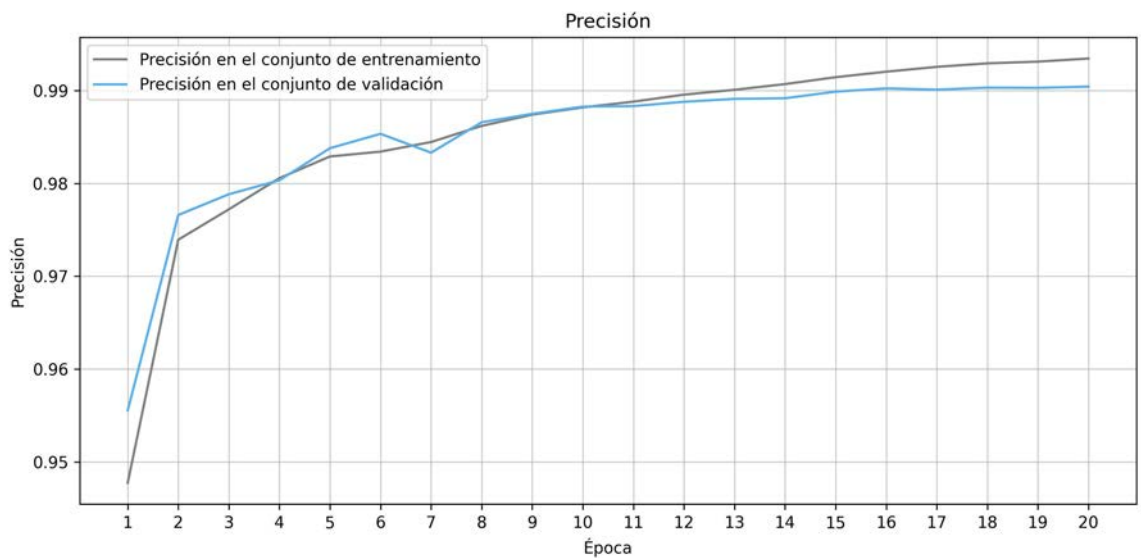


Figura 4.8: Comportamiento de la precisión del modelo final en el conjunto de entrenamiento y validación para cada época. Elaboración propia.

En la Tabla 4.2 se pueden apreciar los resultados al término del proceso de entrenamiento. Se observa que el índice IoU disminuye de 0.8873 durante el entrenamiento a 0.8326 en la fase de validación, lo que equivale a una disminución de 0.0547 unidades. En el conjunto de prueba, el IoU alcanza un valor de 0.8252. En cuanto al coeficiente Dice, se aprecia una diferencia de

0.0317 entre el entrenamiento y la validación. Estos resultados indican que no se evidencia un caso de sobre entrenamiento en el modelo.

Con relación a la precisión, apenas se observan variaciones notables. Esto concuerda con las observaciones realizadas en las gráficas anteriores, donde se constató que el modelo alcanzó una métrica muy elevada desde la primera época de entrenamiento.

En las tres métricas, se observa una tendencia consistente en la que la métrica es menor en el conjunto de validación en comparación con el conjunto de entrenamiento, y a su vez, es inferior en el conjunto de prueba en relación con el conjunto de validación. Este patrón es coherente y tiene una explicación lógica. El conjunto de validación influyó en el modelo al ser utilizado para la selección de hiperparámetros óptimos, mientras que el conjunto de prueba no tuvo ningún tipo de influencia, ni cuantitativa ni cualitativamente, en el proceso de entrenamiento del modelo.

Conjunto	IoU	Dice	Precisión
Entrenamiento	0.8873	0.9403	0.9934
Validación	0.8326	0.9086	0.9904
Prueba	0.8252	0.9042	0.9898

Tabla 4.2: Métricas del modelo final usando un aumento de datos, se muestra el IoU, coeficiente Dice y la precisión en los conjuntos de datos de entrenamiento, validación y prueba.

La matriz de confusión de la Figura 4.9 se construyó a nivel píxel, considerando el total de píxeles del conjunto de prueba, esto es, 123 imágenes de 128 x 128, que da un total de 2015232 píxeles. Dicha matriz nos revela que el modelo tiene un rendimiento sólido en la identificación de casos verdaderos positivos (97161) y verdaderos negativos (1897494), principalmente verdaderos negativos, debido a la distribución sesgada de las clases. No obstante, la presencia de algunos falsos positivos (9357) y falsos negativos (11220) indica áreas para mejorar. Los falsos positivos sugieren píxeles identificados erróneamente como columnas de humo, mientras que los falsos negativos indican casos no detectados.

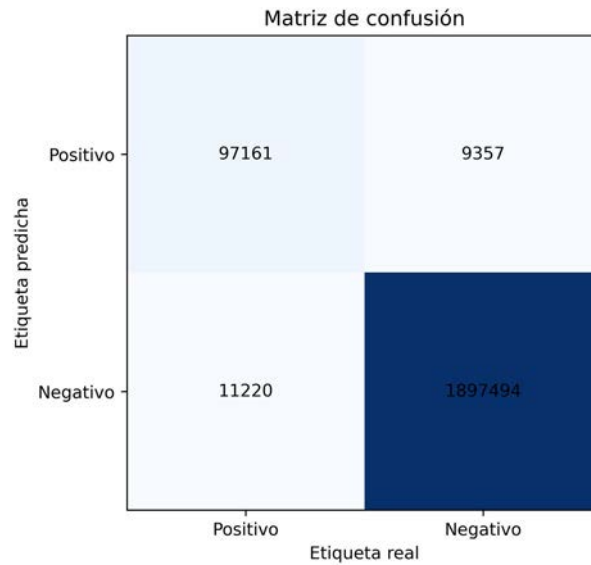


Figura 4.9: Matriz de confusión de los resultados de la segmentación a nivel pixel en el conjunto de datos de prueba. Elaboración propia.

4.5. Ejemplo de segmentación

Cualitativamente, el rendimiento del modelo final también resultó sobresaliente. Es evidente la notoria coincidencia entre las máscaras segmentadas manualmente y las que el modelo predijo (ver Figura 4.10). La intersección entre estas dos, es particularmente discernible, lo que resalta la capacidad del modelo para capturar con precisión los contornos y las áreas relevantes en la detección de columnas de humo de incendios forestales. Es destacable que los errores de omisión; donde el modelo no logra reconocer adecuadamente casos que pertenecen a la categoría positiva, se mantuvieron en niveles muy bajos. De manera análoga, se observaron pocas instancias de errores de comisión, en las que el modelo etiquetó incorrectamente ejemplos como positivos cuando en realidad son negativos; Estos errores de comisión o falsos positivos, en particular, son críticos, ya que pueden generar alarmas innecesarias o resultados incorrectos. La capacidad del modelo para minimizar tanto los errores de omisión como los de comisión subraya su confiabilidad y su habilidad para tomar decisiones precisas en la detección y segmentación de columnas de humo de incendios forestales.

También hay que destacar que debido a la baja resolución del sensor (2km), la tarea de segmentar, incluso a mano puede llegar a ser confusa, aunque se intentó homologar la segmentación manual, intentando mantener un mismo acercamiento o escala de la imagen en

todo momento, así como un mismo nivel de brillo en la pantalla, hay muchas variables que no se contemplaron que están implícitas en el error humano, como lo es el agotamiento visual y la subjetividad. En muchos casos había duda en cuanto a si los píxeles observados pertenecían o no a una columna de humo, pues la densidad de esta no era tanta.

Por lo tanto, los datos segmentados a mano podrían estar sesgados por el criterio del experto que segmentó las imágenes manualmente. Y habría que realizar un análisis más exhaustivo sobre los errores de omisión y comisión observados, en aras de definir si efectivamente se trata de errores.

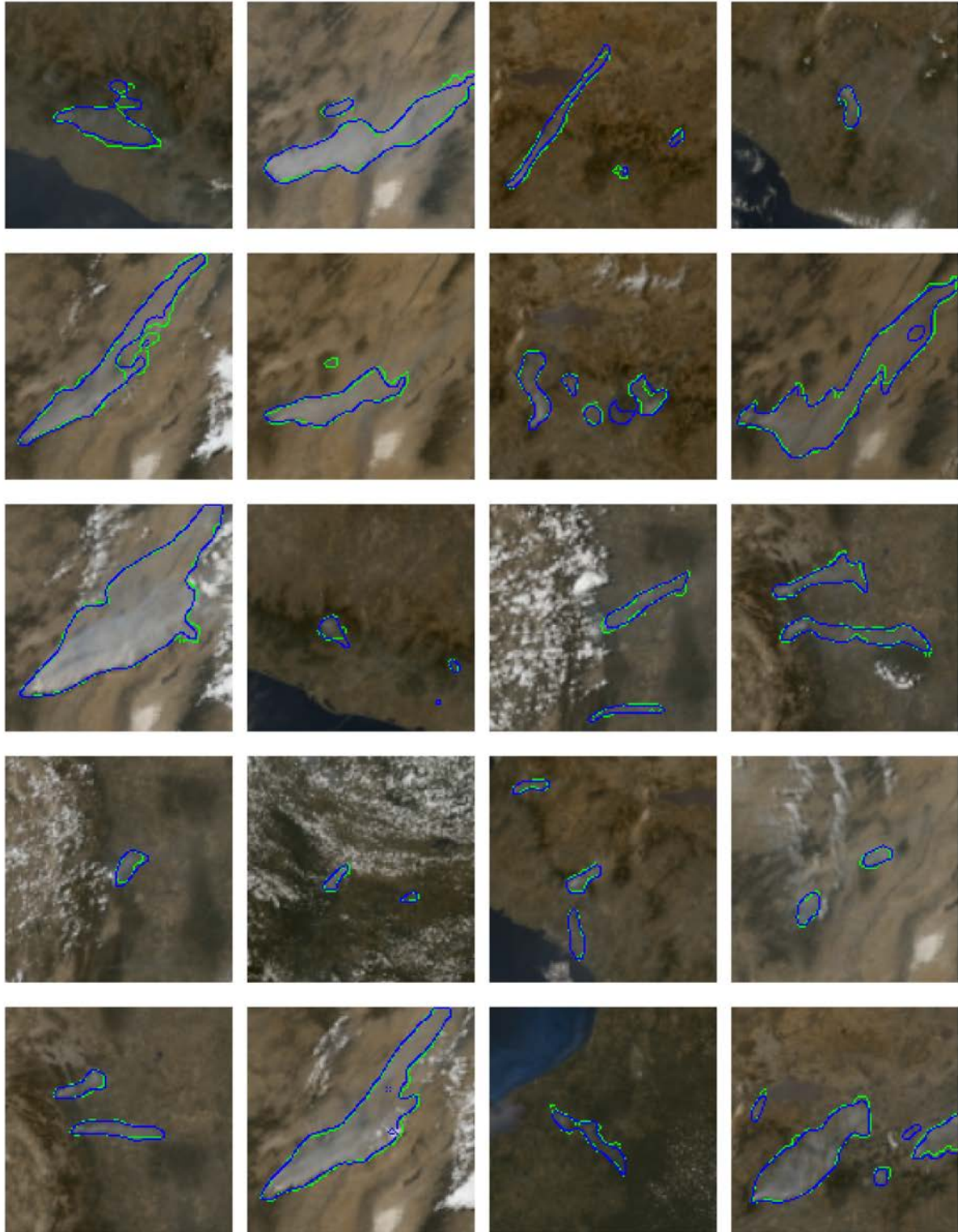


Figura 4.10: Ejemplo de segmentaciones predichas por el modelo (azul), y segmentación manual (verde) de 20 imágenes aleatorias del conjunto de prueba. Elaboración propia.

Capítulo 5. Conclusiones y trabajo futuro

futuro

En este capítulo, se exponen las conclusiones derivadas del presente trabajo de investigación. Se realiza un análisis exhaustivo del cumplimiento de los objetivos y la hipótesis establecida, se exploran las contribuciones realizadas y se identifican las limitaciones encontradas. Además, se delimitan posibles direcciones futuras de investigación en el ámbito de la segmentación semántica de columnas de humo originadas por incendios forestales en México a través de imágenes provenientes del satélite GOES-16.

5.1. Resumen de los objetivos

En este estudio, se abordó el desafío de desarrollar un modelo de segmentación semántica de columnas de humo causadas por incendios forestales utilizando imágenes satelitales GOES-16. Para lograrlo, se establecieron objetivos específicos que guiaron nuestro trabajo de investigación. A continuación, resumimos el cumplimiento de estos objetivos:

- **Análisis de bandas y productos de GOES-16:** En primer lugar, se llevó a cabo un análisis exhaustivo de las bandas, productos y compuestos generados por las imágenes del satélite GOES-16. Concretamente, se llegó a la conclusión de que las bandas 1 ($0.47\mu\text{m}$) y 2 ($0.64\mu\text{m}$) resultaron ser las más relevantes para la detección de columnas de humo. Asimismo, se concluyó que el compuesto de color verdadero presentó la mayor capacidad para distinguir las columnas de humo de manera efectiva.
- **Generación de un conjunto de datos segmentados a mano:** Un componente esencial de nuestro trabajo fue la creación de un conjunto de datos de 1061 imágenes del satélite GOES-16 de columnas de humo segmentadas a mano. Este conjunto de datos se usó para el entrenamiento y la validación de nuestro modelo, y su creación fue un proceso laborioso pero fundamental para asegurar la calidad de los resultados.
- **Adaptación de modelos de aprendizaje profundo:** Se exploraron soluciones basadas en visión por computadora y se adaptó exitosamente la arquitectura U-Net de aprendizaje profundo para abordar el problema de la segmentación semántica de columnas de humo. Alcanzando métricas de 0.8252 de IoU, coeficiente Dice de 0.9042 y precisión de 0.9898 en el conjunto de prueba.

5.2. Validación de la hipótesis

Uno de los aspectos clave de nuestra investigación fue la validación de la hipótesis planteada al inicio de este trabajo. La hipótesis afirmaba que, a pesar de la limitada resolución espacial de las imágenes de satélite GOES-16, era factible segmentar y detectar de manera precisa las columnas de humo de incendios forestales utilizando un modelo de visión por computadora y aprendizaje profundo.

Los resultados obtenidos en este estudio confirman la validez de esta hipótesis. Nuestro modelo demostró ser capaz de segmentar y detectar con precisión las columnas de humo en las imágenes de satélite GOES-16, incluso en situaciones donde las nubes y vapor de agua eran muy similares. Esto respalda la idea de que la combinación de técnicas de visión por computadora y aprendizaje profundo puede ser una herramienta efectiva para monitorear incendios forestales desde el espacio.

5.3. Contribuciones y limitaciones

Las contribuciones del presente trabajo en el campo de la detección de incendios forestales y la monitorización de columnas de humo desde satélites meteorológicos en órbita geoestacionaria pueden ser las siguientes:

- **Enfoque geográfico específico:** Es importante destacar que las condiciones climáticas y las características del suelo pueden variar significativamente en diferentes regiones del mundo debido a diversos factores. A diferencia de varios de los trabajos revisados, esta investigación se enfoca en un área geográfica específica, México, lo que implica la consideración de condiciones ambientales y características particulares de esta región.
- **Utilización de imágenes de un satélite en órbita geoestacionaria:** Este trabajo se basa en el uso de imágenes de satélite GOES-16, que a pesar de su baja resolución espacial, cuenta con una muy alta resolución temporal (una imagen cada 5 minutos), y esto proporciona la posibilidad de obtener datos valiosos para la detección y segmentación de columnas de humo en tiempo oportuno.
- **Aplicación potencial en gestión de desastres:** Al proporcionar un método efectivo para la monitorización de incendios forestales desde el espacio, esta investigación tiene

implicaciones relevantes para la gestión de desastres naturales y la protección del medio ambiente en México.

- Conjunto de datos: La creación de un conjunto de datos de columnas de humo segmentadas a mano es una contribución importante. Aunque existen bases de datos de este tipo de imágenes, es relevante señalar que estas pertenecen a otras regiones del mundo y no están disponibles de manera libre y pública. Este conjunto de datos, generado manualmente, representa un recurso crucial para el entrenamiento y la validación del modelo presentado. Su creación implica un esfuerzo significativo pero esencial para garantizar la calidad de los resultados y permitir la replicabilidad de la investigación en un contexto específico, como el de los incendios forestales en México. Este recurso no solo beneficia a la presente investigación, sino que también puede ser valioso para la comunidad científica y los profesionales involucrados en la monitorización y gestión de incendios forestales en la región.

Aun con las contribuciones mencionadas, es importante reconocer algunas limitaciones y áreas de mejora:

- La calidad de la segmentación podría mejorarse aún más, especialmente en horarios cercanos al amanecer y el atardecer, así como en situaciones de condiciones atmosféricas anormales o cuando las columnas de humo son tenues o fácilmente confundibles con algunos tipos de nubes.
- La cantidad y variabilidad de los datos aún es limitada, aunque el número de imágenes es considerable, estas se derivan solamente de 20 incendios y puede existir una similitud considerable entre una imagen y su consiguiente de una misma columna de humo, debido a que solo las separan 5 minutos de diferencia. Esto podría llevar a una falsa percepción de la variabilidad de los datos de entrenamiento, validación y prueba, aun cuando fueron mezclados de manera aleatoria, estos incluyen información muy similar y, por lo tanto, se podría no alcanzar a entender o visualizar en profundidad si existen algún indicio de sobre-entrenamiento.
- La segmentación manual de las columnas de humo solo fue elaborada por una sola persona. Esto introduce una limitación inherente debido a la subjetividad humana en la interpretación de las imágenes.
- La adaptación de modelos de aprendizaje profundo requiere una atención continua y cuidadosa, dado que los datos y las condiciones ambientales pueden evolucionar con el tiempo, especialmente en este contexto, donde solo se emplearon imágenes de los meses de abril y mayo de 2022. El monitoreo constante se vuelve esencial para garantizar

que el modelo mantenga su eficacia a medida que se enfrenta a nuevas variaciones estacionales, cambios climáticos, y posibles transformaciones en los patrones de incendios forestales en México. Esta adaptabilidad continua es crucial para mantener la precisión y relevancia de las predicciones en el futuro.

5.4. Trabajo futuro

A continuación, se presentan algunas recomendaciones y observaciones derivadas de la experiencia obtenida y del análisis de las limitaciones de esta investigación:

- Integrar datos de múltiples satélites de los mismos incendios o los que coincidan, para mejorar la precisión de las detecciones, siempre que sea posible: que la columna de humo este dentro de la escena de escaneo y en la línea temporal con los horarios de paso de los satélites. Además, la incorporación de múltiples escalas espaciales y temporales en el análisis podría proporcionar una base sólida para realizar evaluaciones más exhaustivas y precisas del comportamiento de las columnas de humo.
- Mejorar la calidad de la segmentación del conjunto de datos creado, por medio de una segmentación manual de más de una persona; si varios expertos segmentaran las mismas imágenes, posteriormente se podrían comparar los resultados entre sí y tomar una decisión de la columna de humo final. Además, la disponibilidad limitada de un único experto podría imponer restricciones en la expansión y actualización del conjunto de datos en el futuro, lo que podría ser necesario para mejorar y mantener el modelo a lo largo del tiempo.
- Ampliar el tamaño y variabilidad del conjunto de datos, incorporando incendios, en medida de lo posible, de todos los meses, así como de distintos años. Esto no podría permitir entrenar modelos con mejor capacidad de generalización.
- Desarrollo de algoritmos para la estimación de parámetros adicionales que puedan caracterizar y brindar más información del contenido de las columnas de humo segmentadas, así como la predicción del comportamiento de una columna de humo, agregando variables como la de velocidad de propagación y dirección del viento combinado con modelos de aprendizaje automático. Para relacionar estas variables es importante la componente espacial.
- Realizar pruebas y adaptaciones de diferentes arquitecturas de aprendizaje profundo en el contexto de la segmentación semántica utilizando el conjunto de datos generado. Esto se lleva a cabo con el propósito de comparar el desempeño de diversos modelos y, a

través de un enfoque experimental, determinar cuál de ellos es el más óptimo. La elección de la arquitectura adecuada es una tarea que requiere de una exploración detallada y empírica para evaluar y seleccionar el modelo que mejor se adapte a las necesidades y requisitos específicos del proyecto.

- Hacer operativo el modelo final en tiempo casi real. Para esto es necesaria infraestructura de procesamiento como servidores o servicios de cómputo web, además, se requiere automatizar tanto la adquisición de imágenes provenientes del satélite GOES-16 como el proceso de segmentación realizado por el modelo final. Una vez operativo, se debe monitorizar con el objetivo de supervisar el rendimiento y su exactitud del modelo y estar alerta ante posibles problemas. La operatividad del modelo de segmentación semántica de columnas de humo generadas por incendios forestales permitirá una respuesta más eficaz ante eventos de esta naturaleza, lo que facilitará la monitorización y la toma de decisiones informadas en situaciones críticas.

5.5. Conclusión Final

En resumen, este estudio ha logrado su objetivo de desarrollar un modelo de segmentación semántica de columnas de humo causadas por incendios forestales utilizando imágenes satelitales GOES-16 y una arquitectura de aprendizaje profundo. Los resultados respaldan la hipótesis inicial y ofrecen una herramienta efectiva para la monitorización y detección de incendios forestales desde el espacio. Los resultados de este trabajo contribuyen a las nuevas perspectivas en la gestión de desastres naturales y la protección del medio ambiente mediante el uso percepción remota y aprendizaje automático.

Referencias bibliográficas

- Ba, R., Chen, C., Yuan, J., Song, W., & Lo, S. (2019). SmokeNet: Satellite Smoke Scene Detection Using Convolutional Neural Network with Spatial and Channel-Wise Attention. *Remote Sensing*, 11(14), Article 14. <https://doi.org/10.3390/rs11141702>
- Bah, M. K., Gunshor, M. M., & Schmit, T. J. (2018). Generation of GOES-16 True Color Imagery without a Green Band. *Earth and Space Science*, 5(9), 549–558. <https://doi.org/10.1029/2018EA000379>
- Balestriero, R., Bottou, L., & LeCun, Y. (2022). *The Effects of Regularization and Data Augmentation are Class Dependent*.
- Blaylock, B. K. (2023). *GOES-2-go: Download and display GOES-East and GOES-West data [Python]*. <https://github.com/blaylockbk/goes2go>
- Borchers Arriagada, N., Palmer, A. J., Bowman, D. M., Morgan, G. G., Jalaludin, B. B., & Johnston, F. H. (2020). Unprecedented smoke-related health burden associated with the 2019–20 bushfires in eastern Australia. *Medical Journal of Australia*, 213(6), 282–283. <https://doi.org/10.5694/mja2.50545>
- Cantoral Ceballos, J. A. (2023). *DL_fundamentals/Fundamentals_DL_UNET_4_video_v2.ipynb*. GitHub. https://github.com/JACantoral/DL_fundamentals/blob/main/Fundamentals_DL_UNET_4_video_v2.ipynb
- Castillo, M., Pedernera, P., & Peña, E. (2003). Incendios forestales y medio ambiente: Una síntesis global. *Revista Ambiente y Desarrollo*, 9(3).
- Chen, H., Samet, J. M., Bromberg, P. A., & Tong, H. (2021). Cardiovascular health impacts of wildfire smoke exposure. *Particle and Fibre Toxicology*, 18(1), 2. <https://doi.org/10.1186/s12989-020-00394-8>
- Contreras-Moctezuma, J., Rodríguez-Trejo, D. A., Retama-Hernández, A., & Sánchez-Rodríguez, J. J. (2003). Gases del humo de incendios en bosques de pinus hartwegii. *Agrociencia*, 37(3), 309–316.
- Datasets & DataLoaders—PyTorch Tutorials 2.0.1+cu117 documentation*. (2023). https://pytorch.org/tutorials/beginner/basics/data_tutorial.html
- Dirk Krebs, G. (2020). *GOES R, S, T, U*. Gunter's Space Page. https://space.skyrocket.de/doc_sdat/goes-r.htm
- Dumont Le Brazidec, J., Vanderbecken, P., Farchi, A., Bocquet, M., Lian, J., Broquet, G., Kuhlmann, G., Danjou, A., & Lauvaux, T. (2023). Segmentation of XCO₂ images with deep learning: Application to synthetic plumes from cities and power plants. *Geoscientific Model Development*, 16(13), 3997–4016. <https://doi.org/10.5194/gmd-16-3997-2023>

- Frizzi, S., Bouchouicha, M., Ginoux, J.-M., Moreau, E., & Sayadi, M. (2021). Convolutional neural network for smoke and fire semantic segmentation. *IET Image Processing*, *15*(3), 634–647. <https://doi.org/10.1049/ipr2.12046>
- Ghosh, S., Chaki, A., & Santosh, K. (2021). Improved U-Net architecture with VGG-16 for brain tumor segmentation. *Physical and Engineering Sciences in Medicine*, *44*(3), 703–712. <https://doi.org/10.1007/s13246-021-01019-w>
- Gomez, T. (2019, diciembre 16). *2019: El año del fuego en Latinoamérica*. Noticias ambientales. <https://es.mongabay.com/2019/12/2019-el-ano-del-fuego/>
- Guo, Y., Liu, Y., Georgiou, T., & Lew, M. S. (2018). A review of semantic segmentation using deep neural networks. *International Journal of Multimedia Information Retrieval*, *7*(2), 87–93. <https://doi.org/10.1007/s13735-017-0141-z>
- Johnston, F. H., Henderson, S. B., Chen, Y., Randerson, J. T., Marlier, M., Defries, R. S., Kinney, P., Bowman, D. M. J. S., & Brauer, M. (2012). Estimated global mortality attributable to smoke from landscape fires. *Environmental Health Perspectives*, *120*(5), 695–701. <https://doi.org/10.1289/ehp.1104422>
- Jones, M. W., Abatzoglou, J. T., Veraverbeke, S., Andela, N., Lasslop, G., Forkel, M., Smith, A. J. P., Burton, C., Betts, R. A., van der Werf, G. R., Sitch, S., Canadell, J. G., Santín, C., Kolden, C., Doerr, S. H., & Le Quéré, C. (2022). Global and Regional Trends and Drivers of Fire Under Climate Change. *Reviews of Geophysics*, *60*(3), e2020RG000726. <https://doi.org/10.1029/2020RG000726>
- Kim, K.-H., Kabir, E., & Kabir, S. (2015). A review on the human health impact of airborne particulate matter. *Environment International*, *74*, 136–143. <https://doi.org/10.1016/j.envint.2014.10.005>
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, *25*. https://proceedings.neurips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html
- Lamba, H. (2019, febrero 17). *Understanding Semantic Segmentation with UNET*. Medium. <https://towardsdatascience.com/understanding-semantic-segmentation-with-unet-6be4f42d4b47>
- Lara, P., Fitzgerald, R. M., Karle, N. N., Talamantes, J., Miranda, M., Baumgardner, D., & Stockwell, W. R. (2022). Winter and Wildfire Season Optical Characterization of Black and Brown Carbon in the El Paso-Ciudad Juárez Airshed. *Atmosphere*, *13*(8), Article 8. <https://doi.org/10.3390/atmos13081201>
- Larsen, A., Hanigan, I., Reich, B. J., Qin, Y., Cope, M., Morgan, G., & Rappold, A. G. (2021). A deep learning approach to identify smoke plumes in satellite imagery in near-real time for health risk communication. *Journal of Exposure Science & Environmental Epidemiology*, *31*(1), Article 1. <https://doi.org/10.1038/s41370-020-0246-y>

- Long, F. (2020). Microscopy cell nuclei segmentation with enhanced U-Net. *BMC Bioinformatics*, 21(1), 8. <https://doi.org/10.1186/s12859-019-3332-1>
- Luo, Y., Zhao, L., Liu, P., & Huang, D. (2018). Fire smoke detection algorithm based on motion characteristic and convolutional neural networks. *Multimedia Tools and Applications*, 77(12), 15075–15092. <https://doi.org/10.1007/s11042-017-5090-2>
- Mahbod, A., Schaefer, G., Dorffner, G., Hatamikia, S., Ecker, R., & Ellinger, I. (2022). A dual decoder U-Net-based model for nuclei instance segmentation in hematoxylin and eosin-stained histological images. *Frontiers in Medicine*, 9. <https://www.frontiersin.org/articles/10.3389/fmed.2022.978146>
- Mahonchak, G. D. (2019). *GOES-R Series Data Book*.
- McNamara, D., Stephens, G., Ruminski, M., & Kasheta, T. (2004). The Hazard Mapping System (HMS)—NOAA'S multi-sensor fire and smoke detection program using environmental satellites. *Conference on Satellite Meteorology and Oceanography*.
- Mommert, M., Sigel, M., Neuhausler, M., Scheibenreif, L., & Borth, D. (2020). *Characterization of Industrial Smoke Plumes from Remote Sensing Data* (arXiv:2011.11344). arXiv. <http://arxiv.org/abs/2011.11344>
- NOAA HMS Smoke Detection. (2023). <https://noaa.maps.arcgis.com/home/item.html?id=ab7a5fbd76e3499296350eabf599fc63>
- Pausas, J. G. (2020). *Incendios forestales*. Los Libros De La Catarata. [https://books.google.com.mx/books?hl=es&lr=&id=i1jZDwAAQBAJ&oi=fnd&pg=PT2&dq=Pausas,+J.+G.+\(2020\).+Incendios+forestales.+Los+Libros+De+La+Catarata.&ots=OLKxI0xUFG&sig=JilCmiQsKppbkxaQPhsBeQW9Ax4&redir_esc=y#v=onepage&q=Pausas%2C%20J.%20G.%20\(2020\).%20Incendios%20forestales.%20Los%20Libros%20De%20La%20Catarata.&f=false](https://books.google.com.mx/books?hl=es&lr=&id=i1jZDwAAQBAJ&oi=fnd&pg=PT2&dq=Pausas,+J.+G.+(2020).+Incendios+forestales.+Los+Libros+De+La+Catarata.&ots=OLKxI0xUFG&sig=JilCmiQsKppbkxaQPhsBeQW9Ax4&redir_esc=y#v=onepage&q=Pausas%2C%20J.%20G.%20(2020).%20Incendios%20forestales.%20Los%20Libros%20De%20La%20Catarata.&f=false)
- Qin, Y., Mitchell, R., & Forgan, B. (2015). Characterizing the Aerosol and Surface Reflectance Over Australia Using AATSR. *IEEE Transactions on Geoscience and Remote Sensing*, 53, 1–20. <https://doi.org/10.1109/TGRS.2015.2433911>
- RAMMB. (2023). *Regional and Mesoscale Meteorology Branch VISIT – Quick Reference Material*. Regional and Mesoscale Meteorology Branch. https://rammb2.cira.colostate.edu/training/visit/quick_reference/#tab17
- Real Academia Española. (2023). Real Academia Española. <https://www.rae.es/inicio>
- Reilly, M. J., Zuspan, A., Halofsky, J. S., Raymond, C., McEvoy, A., Dye, A. W., Donato, D. C., Kim, J. B., Potter, B. E., Walker, N., Davis, R. J., Dunn, C. J., Bell, D. M., Gregory, M. J., Johnston, J. D., Harvey, B. J., Halofsky, J. E., & Kerns, B. K. (2022). Cascadia Burning: The historic, but not historically unprecedented, 2020 wildfires in the Pacific Northwest, USA. *Ecosphere*, 13(6), e4070. <https://doi.org/10.1002/ecs2.4070>
- Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation* (arXiv:1505.04597). arXiv. <http://arxiv.org/abs/1505.04597>

- Sanderfoot, O. V., Bassing, S. B., Brusa, J. L., Emmet, R. L., Gillman, S. J., Swift, K., & Gardner, B. (2022). A review of the effects of wildfire smoke on the health and behavior of wildlife. *Environmental Research Letters*, 16(12), 123003. <https://doi.org/10.1088/1748-9326/ac30f6>
- Schmit, T., Gunshor, M., Fu, G., Rink, T., Bah, K., Zhang, W., & Wolf, W. (2012). *GOES-R Advanced Baseline Imager (ABI) Algorithm Theoretical Basis Document For Cloud and Moisture Imagery Product (CMIP)*. 63. <https://www.star.nesdis.noaa.gov/goesr/docs/ATBD/Imagery.pdf>.
- Schmit, T., Lindstrom, S., Gerth, J., & Gunshor, M. (2018). Applications of the 16 spectral bands on the Advanced Baseline Imager (ABI). *Journal of Operational Meteorology*, 06, 33–46. <https://doi.org/10.15191/nwajom.2018.0604>
- Selimovic, V., Yokelson, R. J., McMeeking, G. R., & Coefield, S. (2019). In situ measurements of trace gases, PM, and aerosol optical properties during the 2017 NW US wildfire smoke event. *Atmospheric Chemistry and Physics*, 19(6), 3905–3926. <https://doi.org/10.5194/acp-19-3905-2019>
- Smith, L. N. (2017). *Cyclical Learning Rates for Training Neural Networks* (arXiv:1506.01186). arXiv. <http://arxiv.org/abs/1506.01186>
- Smith, L. N. (2018). *A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay* (arXiv:1803.09820). arXiv. <http://arxiv.org/abs/1803.09820>
- Smith, L. N., & Topin, N. (2018). *Super-Convergence: Very Fast Training of Neural Networks Using Large Learning Rates* (arXiv:1708.07120). arXiv. <https://doi.org/10.48550/arXiv.1708.07120>
- Sullivan, A., Baker, E., & Kurvits, T. (2022). *Spreading Like Wildfire: The Rising Threat of Extraordinary Landscape Fires*. https://policycommons.net/artifacts/2259313/wildfire_rra/3017991/
- Szeliski, R. (2011). *Computer Vision: Algorithms and Applications*. Springer London. <https://doi.org/10.1007/978-1-84882-935-0>
- U.S. Environmental Protection Agency. (2015, diciembre 29). *Sources of Greenhouse Gas Emissions* [Overviews and Factsheets]. <https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions>
- U.S. Environmental Protection Agency. (2016, junio 27). *Climate Change Indicators: Atmospheric Concentrations of Greenhouse Gases* [Reports and Assessments]. <https://www.epa.gov/climate-indicators/climate-change-indicators-atmospheric-concentrations-greenhouse-gases>
- U.S. Environmental Protection Agency. (2019, febrero 19). *What is carbon monoxide?* [Overviews and Factsheets]. <https://www.epa.gov/indoor-air-quality-iaq/what-carbon-monoxide>
- U.S. Environmental Protection Agency. (2023). *Nitrogen Oxides Control Regulations | Ground-level Ozone | New England | US EPA* [Overviews & Factsheets,]. <https://www3.epa.gov/region1/airquality/nox.html#main-content>

- Van der Velde, I. R., Van der Werf, G. R., Houweling, S., Maasakkers, J. D., Borsdorff, T., Landgraf, J., Tol, P., van Kempen, T. A., van Hees, R., Hoogeveen, R., Veefkind, J. P., & Aben, I. (2021). Vast CO₂ release from Australian fires in 2019–2020 constrained by satellite. *Nature*, 597(7876), Article 7876. <https://doi.org/10.1038/s41586-021-03712-y>
- Vega Nieva, D. J. (2020). *Manual del Sistema de Predicción de Peligro de Incendios Forestales de México*. http://forestales.ujed.mx/incendios2/php/publicaciones_documentos/1_1-%20MANUAL_DE_USUARIO_SPPIF_v15_DV290820.pdf
- Wang, Y., Huang, G., Song, S., Pan, X., Xia, Y., & Wu, C. (2022). Regularizing Deep Networks With Semantic Data Augmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7), 3733–3748. <https://doi.org/10.1109/TPAMI.2021.3052951>
- Wang, Z., Yang, P., Liang, H., Zheng, C., Yin, J., Tian, Y., & Cui, W. (2022). Semantic Segmentation and Analysis on Sensitive Parameters of Forest Fire Smoke Using Smoke-Unet and Landsat-8 Imagery. *Remote Sensing*, 14(1), Article 1. <https://doi.org/10.3390/rs14010045>
- Ward, M., Tulloch, A. I. T., Radford, J. Q., Williams, B. A., Reside, A. E., Macdonald, S. L., Mayfield, H. J., Maron, M., Possingham, H. P., Vine, S. J., O'Connor, J. L., Massingham, E. J., Greenville, A. C., Woinarski, J. C. Z., Garnett, S. T., Lintermans, M., Scheele, B. C., Carwardine, J., Nimmo, D. G., ... Watson, J. E. M. (2020). Impact of 2019–2020 mega-fires on Australian fauna habitat. *Nature Ecology & Evolution*, 4(10), Article 10. <https://doi.org/10.1038/s41559-020-1251-1>
- Wen, J., & Burke, M. (2021). *Wildfire smoke plume segmentation using geostationary satellite imagery* (arXiv:2109.01637). arXiv. <https://doi.org/10.48550/arXiv.2109.01637>
- World Health Organization. (2023). *Air pollution*. <https://www.who.int/health-topics/air-pollution>
- Xie, Y., Qu, J. J., Xiong, X., Hao, X., Che, N., & Sommers, W. (2007). Smoke plume detection in the eastern United States using MODIS. *International Journal of Remote Sensing*, 28(10), 2367–2374. <https://doi.org/10.1080/01431160701236795>
- Yung, Y. (2003). An introduction to atmospheric radiation. By K. N. Liou. Academic Press. Second edition, 2002. Pp. Xiv+583. ISBN 0 12 451451 0. *Quarterly Journal of The Royal Meteorological Society - QUART J ROY METEOROL SOC*, 129, 1741–1741. <https://doi.org/10.1256/003590003102695746>
- Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). *Dive into Deep Learning*. Cambridge University Press.
- Zhang, Y., Dong, T., Hu, W., Wang, X., Xu, B., Lin, Z., Hofer, T., Stefanoff, P., Chen, Y., Wang, X., & Xia, Y. (2019). Association between exposure to a mixture of phenols, pesticides, and phthalates and obesity: Comparison of three statistical models. *Environment International*, 123, 325–336. <https://doi.org/10.1016/j.envint.2018.11.076>