

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



**MODELO DE RED NEURONAL CONVOLUCIONAL PARA LA
CLASIFICACIÓN DE TIPOS DE NUBES EN IMÁGENES DE WEBCAM**

Tesis para obtener el título profesional de Ingeniero Informático:

AUTOR:

Mario Alejandro Andonaire Tuesta

ASESORES:

Dr. César Armando Beltrán Castañón

Dr. Edwin Rafael Villanueva Talavera


Lima, noviembre de 2024

Informe de Similitud

Yo, César Armando Beltrán Castañón, docente de la Facultad de Ciencias e Ingeniería de la Pontificia Universidad Católica del Perú, asesor(a) de la tesis/el trabajo de investigación titulado MODELO DE RED NEURONAL CONVOLUCIONAL PARA LA CLASIFICACIÓN DE TIPOS DE NUBES EN IMÁGENES DE WEBCAM, del autor Mario Alejandro Andonaire Tuesta, dejo constancia de lo siguiente:

- El mencionado documento tiene un índice de puntuación de similitud de 22%. Así lo consigna el reporte de similitud emitido por el software *Turnitin* el 22/11/2024.
- He revisado con detalle dicho reporte y la Tesis o Trabajo de Suficiencia Profesional, y no se advierte indicios de plagio.
- Las citas a otros autores y sus respectivas referencias cumplen con las pautas académicas.

Lugar y fecha: Lima 24 de Noviembre del 2024

| | |
|--|--|
| Apellidos y nombres del asesor / de la asesora: Beltrán Castañón, César Armando | |
| DNI: 29561260 | Firma  |
| ORCID: 0000-0002-0173-4140 | |

Resumen

La presente investigación aborda la problemática de la clasificación manual de tipos de nubes en estaciones meteorológicas del Servicio Nacional de Meteorología e Hidrología del Perú (SENAMHI), un proceso sujeto a errores y demandante de tiempo. El estudio justifica su relevancia destacando la importancia de la observación precisa de nubes en la meteorología y cómo la automatización mediante una herramienta de clasificación basada en redes neuronales convolucionales podría optimizar este procedimiento. Este trabajo se enmarca dentro del proyecto Meteo-Huascarán, en colaboración con el SENAMHI y el grupo de investigación IA-PUCP.

El método empleado consiste en el entrenamiento de un modelo de red neuronal convolucional, utilizando aprendizaje supervisado para clasificar automáticamente los tipos de nubes a partir de imágenes tomadas desde tierra. La investigación contempla la preparación de una base de datos representativa, la implementación de tres arquitecturas de redes neuronales y la selección de la más adecuada mediante una evaluación comparativa. Además, el desarrollo de una interfaz web para mostrar el funcionamiento del modelo.

Los resultados muestran un avance en la automatización de la clasificación de tipos de nubes, seleccionando la arquitectura Inception v3 como la más adecuada para el proyecto. La implementación de la interfaz web facilita la interacción con el modelo, permitiendo la carga de imágenes de nubes y obteniendo la clasificación correspondiente de forma automática.

Este trabajo contribuye a la automatización del proceso de clasificación de tipos de nubes en el SENAMHI, proponiendo una solución tecnológica que reduce la dependencia de observadores humanos y mejora la eficiencia y precisión en la observación meteorológica. La implementación de este modelo representa un paso adelante hacia la modernización y adaptación a las necesidades futuras en el campo de la meteorología en el Perú.

Tabla de Contenidos

| | |
|--|----|
| Capítulo 1. Generalidades..... | 1 |
| 1.1. Problemática..... | 1 |
| 1.1.1. Árbol de problemas. | 1 |
| 1.1.2. Descripción. | 2 |
| 1.1.3. Problema seleccionado. | 4 |
| 1.2. Presentación de objetivos, resultados esperados y estrategias de verificación | 4 |
| 1.2.1. Objetivo general. | 4 |
| 1.2.2. Objetivos específicos..... | 4 |
| 1.2.4. Mapeo de objetivos, resultados y verificación. | 5 |
| 1.3. Métodos y Procedimientos..... | 7 |
| 1.3.1. Herramientas..... | 8 |
| 1.3.2. Métodos y procedimientos..... | 10 |
| Capítulo 2. Marco Legal/Regulatorio/Conceptual/otros..... | 16 |
| 2.1. Meteorología en el Perú | 16 |
| 2.2. Nube | 16 |
| 2.3. Clasificación de nubes..... | 17 |
| 2.4. Géneros de nubes | 17 |
| Capítulo 3. Estado del Arte..... | 20 |
| 3.1. Objetivos de revisión..... | 20 |
| 3.2. Preguntas de revisión | 20 |
| 3.3. Estrategia de búsqueda..... | 21 |
| 3.3.1. Motores de búsqueda a usar..... | 21 |
| 3.3.2. Cadenas de búsqueda a usar. | 21 |
| 3.3.3. Criterios de inclusión/exclusión. | 24 |

| | |
|---|----|
| 3.3.4. Documentos encontrados..... | 25 |
| 3.4. Formulario de extracción de datos | 27 |
| 3.5. Resultados de la revisión..... | 28 |
| 3.5.1. P1 ¿Qué arquitecturas de redes neuronales convolucionales se han utilizado para la clasificación de nubes con imágenes tomadas desde tierra, cuáles fueron los resultados y con qué métrica se evaluaron?..... | 28 |
| 3.5.2. P2 ¿Qué bases de datos se han utilizado para la clasificación de nubes con imágenes tomadas desde tierra y en qué clases se etiquetaron las imágenes? | 31 |
| 3.5.3. P3 ¿Qué técnicas de preprocesamiento de datos se han utilizado para la clasificación de nubes con imágenes tomadas desde tierra y de qué manera ayudaron a mejorar los resultados? | 33 |
| Capítulo 4. Preparar una base de datos de imágenes de tipos de nubes que sea representativa, suficientemente heterogénea y grande para el entrenamiento adecuado de un modelo de clasificación automática de tipos de nubes | 35 |
| 4.1. Resultado 1: <i>Pipeline</i> para la creación de la base de datos de imágenes de tipos de nubes..... | 35 |
| 4.1.1. Criterios de aceptación de bases de datos..... | 35 |
| 4.1.2. Criterios de aceptación y rechazo de imágenes..... | 36 |
| 4.1.3. Descripción del pipeline para la creación de la base de datos de imágenes de tipos de nubes..... | 36 |
| 4.2. Resultado 2: Base de datos de imágenes de tipos de nubes preparada y separada adecuadamente para el proceso de entrenamiento y prueba | 38 |
| 4.3. Discusión sobre el desarrollo del primer objetivo..... | 42 |
| Capítulo 5. Implementar un modelo basado en una red neuronal convolucional que permita clasificar tipos de nubes automáticamente..... | 44 |
| 5.1. Resultado 3: Revisión sistemática de arquitecturas de redes neuronales convolucionales que han sido usadas para la clasificación automática de tipos de nubes...44 | |
| 5.2. Resultado 4: Implementación de las arquitecturas de redes neuronales convolucionales más efectivas y relevantes | 45 |

| | |
|--|----|
| 5.2.1. Flujo de entrenamiento de redes neuronales convolucionales..... | 46 |
| 5.3. Resultado 5: Evaluación comparativa del desempeño de distintas arquitecturas de redes neuronales convolucionales en la base de datos..... | 49 |
| 5.4. Resultado 6: Desarrollo de una interfaz web que muestre el funcionamiento del modelo de red neuronal convolucional implementado | 56 |
| 5.4.1. Lista de requerimientos funcionales de la interfaz web. | 56 |
| 5.4.2. Lista de requerimientos no funcionales de la interfaz web. | 56 |
| 5.4.3. Arquitectura de despliegue de la interfaz web en AWS..... | 57 |
| 5.4.4. Desarrollo del módulo backend..... | 58 |
| 5.4.5. Desarrollo del módulo frontend..... | 59 |
| 5.4.6. Pruebas funcionales de la interfaz web..... | 61 |
| 5.5. Discusión sobre el desarrollo del segundo objetivo..... | 63 |
| Conclusiones y trabajos futuros..... | 65 |
| Conclusiones | 65 |
| Trabajos futuros..... | 66 |
| Referencias bibliográficas..... | 68 |
| Anexos | 74 |

Índice de Figuras

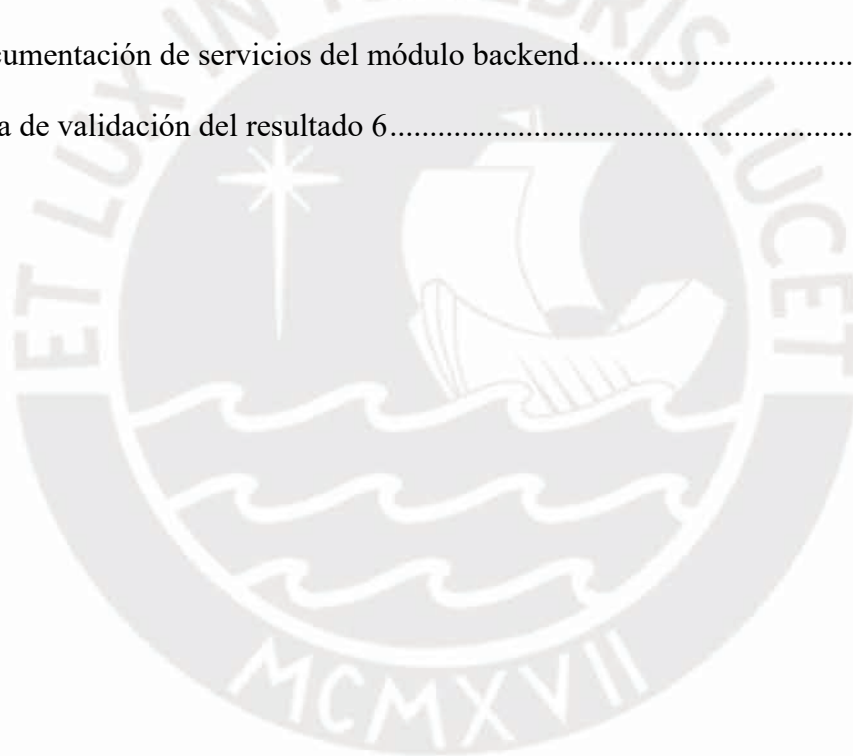
| | |
|--|----|
| Figura 1. Árbol de problemas..... | 1 |
| Figura 2. Diagrama de red neuronal estándar..... | 13 |
| Figura 3. Arquitectura de una red neuronal convolucional..... | 15 |
| Figura 4. Imágenes representativas de cada género de nube..... | 19 |
| Figura 5. Pipeline para la creación de la base de datos de imágenes de tipos de nubes..... | 37 |
| Figura 6. Ejemplares de fotografías de nubes de CCSN..... | 38 |
| Figura 7. Ejemplo de archivo de salida de <i>web scraping</i> | 39 |
| Figura 8. Ejemplos de fotografías de la base de datos final..... | 41 |
| Figura 9. Gráfico de accuracy durante en el entrenamiento..... | 47 |
| Figura 10. Gráfico de la función de pérdida en el conjunto de <i>train</i> | 48 |
| Figura 11. Gráfico de la función de pérdida en el conjunto de <i>validation</i> | 48 |
| Figura 12. Matriz de confusión en el conjunto de <i>test</i> | 49 |
| Figura 13. Gráfico de coordenadas paralelas de los experimentos realizados en CloudNet..... | 51 |
| Figura 14. Gráfico de coordenadas paralelas de los experimentos realizados en ResNet-50..... | 52 |
| Figura 15. Gráfico de coordenadas paralelas de los experimentos realizados en Inception v3..... | 53 |
| Figura 16. F-score macro a lo largo del entrenamiento de las 3 arquitecturas..... | 54 |
| Figura 17. Matriz de confusión en el conjunto de test de la mejor configuración de Inception v3..... | 55 |
| Figura 18. Gráfico de la arquitectura definida para el despliegue de la interfaz web..... | 58 |
| Figura 19. Primer modo de uso de la interfaz web..... | 60 |
| Figura 20. Segundo modo de uso de la interfaz web..... | 61 |
| Figura 21. Ejemplo de archivo csv que contiene los nombres de archivos y tipos de nubes inferidos..... | 61 |

Índice de Tablas

| | |
|--|----|
| Tabla 1. Mapeo del objetivo específico 1, resultados y verificación..... | 5 |
| Tabla 2. Mapeo del objetivo específico 2, resultados y verificación..... | 6 |
| Tabla 3. Mapeo de herramientas, métodos y procedimientos por resultado..... | 7 |
| Tabla 4. Cantidad de artículos encontrados para la pregunta 1 | 22 |
| Tabla 5. Cantidad de artículos encontrados para la pregunta 2 | 23 |
| Tabla 6. Cantidad de artículos encontrados para la pregunta 3 | 24 |
| Tabla 7. Resumen de documentos encontrados | 25 |
| Tabla 8. Detalle de estudios primarios encontrados | 25 |
| Tabla 9. Formulario de extracción de datos..... | 27 |
| Tabla 10. Resultado de los mejores modelos..... | 30 |
| Tabla 11. Resumen de bases de datos relevantes..... | 31 |
| Tabla 12. Cantidad de ejemplares por clase y por conjunto de la base de datos final..... | 41 |
| Tabla 13. Hiperparámetros que se afinaron y los valores que tomaron..... | 50 |
| Tabla 14. Indicadores promedio de 18 experimentos en CloudNet..... | 51 |
| Tabla 15. Indicadores promedio de 18 experimentos en ResNet-50 | 52 |
| Tabla 16. Indicadores promedio de 18 experimentos en Inception v3 | 53 |
| Tabla 17. Resultados de la mejor versión de cada arquitectura..... | 54 |
| Tabla 18. Clase de equivalencia del registro de imagen o imágenes..... | 61 |
| Tabla 19. Escenario 1 de pruebas funcionales | 62 |
| Tabla 20. Escenario 2 de pruebas funcionales | 62 |
| Tabla 21. Escenario 3 de pruebas funcionales | 62 |

Índice de Anexos

| | |
|---|----|
| Anexo A. Páginas web de las herramientas empleadas en el proyecto | 74 |
| Anexo B. Formulario de extracción de datos..... | 75 |
| Anexo C. Plan de Proyecto | 76 |
| Anexo D. Acta de validación del resultado 1 | 85 |
| Anexo E. Bases de datos encontradas en la revisión sistemática de la literatura | 86 |
| Anexo F. Base de datos de Cloud Appreciation Society | 91 |
| Anexo G. Acta de validación del resultado 3 | 92 |
| Anexo H. Acta de validación del resultado 5 | 93 |
| Anexo I. Documentación de servicios del módulo backend..... | 94 |
| Anexo J. Acta de validación del resultado 6..... | 95 |



Capítulo 1. Generalidades

1.1. Problemática

Este capítulo se enfocará en describir la problemática de la clasificación de tipos de nubes en el Perú. Para esto, se construirá un árbol de problemas; posteriormente, se desarrollará la problemática central, sus causas y sus efectos.

1.1.1. Árbol de problemas.

Para la definición de la problemática, se desarrolló un árbol de problemas, que pueden visualizar en la siguiente figura. En el medio se menciona el problema central; debajo, las causas; y encima, los efectos.

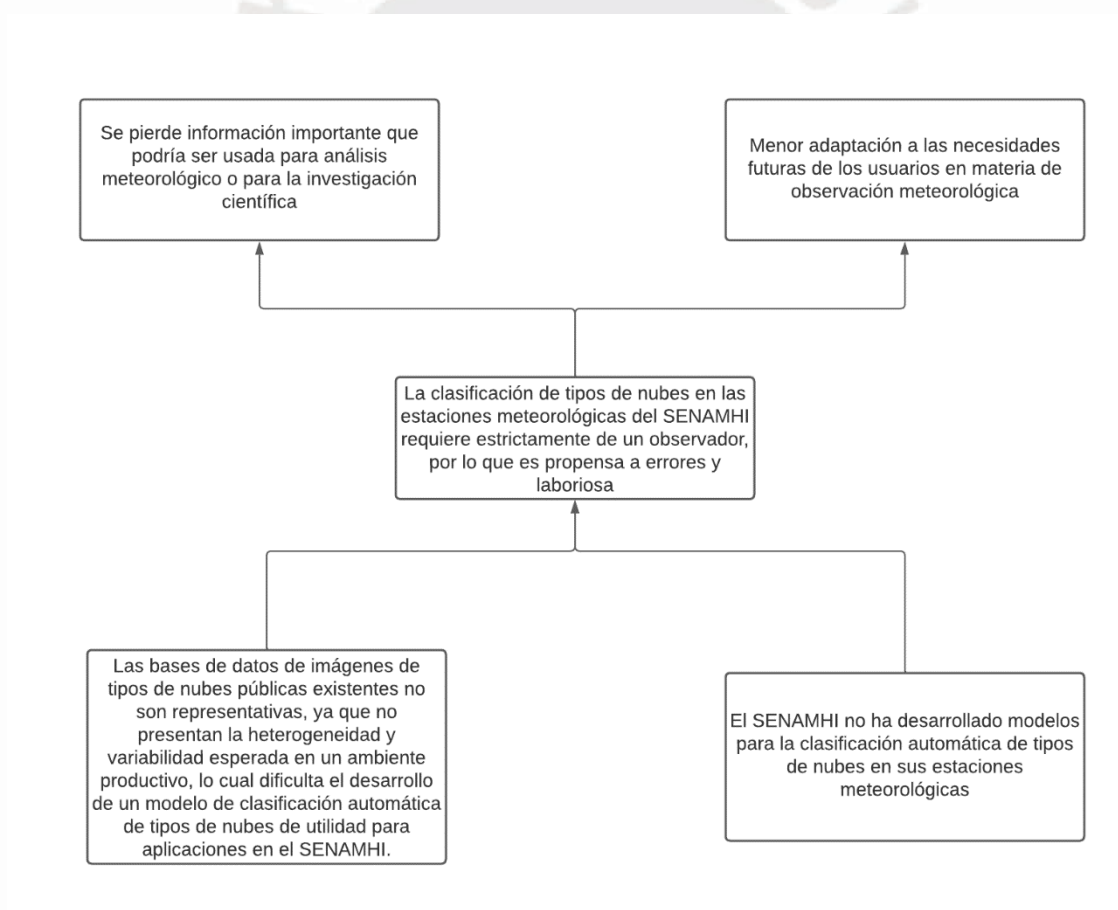


Figura 1. Árbol de problemas.

1.1.2. Descripción.

Las nubes tienen un efecto en el clima y el tiempo terrestre al modular la calefacción solar y alterar el transporte de humedad al formar precipitaciones (Chen, Rossow & Zhang, 2000). Según la Organización Meteorológica Mundial (OMM), se clasifican en 10 géneros principales, donde cada tipo de nube es una manifestación de diferentes condiciones atmosféricas que conduce a diversos fenómenos meteorológicos (World Meteorological Organization, 2017). Por lo tanto, una comprensión sólida y una identificación confiable de los diferentes tipos de nubes, destacando la importancia de su clasificación, puede facilitar la predicción del tiempo,

De acuerdo con la visión del Sistema Mundial Integrado de Sistemas de Observación de la OMM (WIGOS), para 2040, nuevas tendencias sugieren integrar las observaciones obtenidas a través de la colaboración masiva con las observaciones regulares de los Servicios Meteorológicos e Hidrológicos Nacionales (SMHN). Esta esperada afluencia de datos implica una disponibilidad rápidamente creciente de imágenes de nubes tomadas desde tierra que podrían analizarse para producir una rica base de datos que podría ayudar a los pronosticadores y ser de interés para el público y la investigación científica. Por lo tanto, la clasificación automática de tipos de nubes es esencial para manejar y aprovechar estos datos eficientemente (World Meteorological Organization, 2019).

El monitoreo meteorológico en el país está a cargo del Servicio Nacional de Meteorología e Hidrología del Perú (SENAMHI). Este cuenta con una red de estaciones meteorológicas convencionales y automáticas. Las estaciones convencionales cuentan con personal entrenado para realizar mediciones de carácter manual y observaciones; por otro lado, las estaciones automáticas son sistemas electrónicos que cuentan con sensores para medir y registrar las variables (Servicio Nacional de Meteorología e Hidrología del Perú, 2013). Una de las mediciones tomadas en cuenta son los tipos de nubes que se pueden observar en la atmósfera

en un determinado momento. En las estaciones convencionales, la clasificación de los tipos de nubes presentes en la atmósfera es realizada por los observadores mediante el método convencional, un proceso muy laborioso y poco confiable, ya que requiere mucho tiempo y depende de la pericia del observador, lo que genera incertidumbre y sesgos debido a la extracción de la información de manera visual (Zhang, Liu, Zhang & Song, 2018). Por otro lado, las estaciones automáticas carecen de personal, por lo que en ellas la clasificación de tipos de nubes no se lleva a cabo.

La primera causa de esta problemática es que el SENAMHI no ha desarrollado modelos para la clasificación automática de tipos de nubes. La segunda, se relaciona con las bases de datos de imágenes de tipos de nubes públicas existentes, las cuales no son representativas, pues no presentan la heterogeneidad y variabilidad esperadas en un ambiente productivo. Esto dificulta el desarrollo de un modelo de clasificación automática de tipos de nubes de utilidad para aplicaciones en el SENAMHI. El primer efecto del problema mencionado se relaciona con la pérdida de información importante que podría ser usada para análisis meteorológico por esta entidad o por terceros, o para la investigación científica. Además, el segundo efecto se vincula con la menor adaptación a las necesidades futuras de los usuarios en materia de observación meteorológica, ya que se espera mejorar los procesos relacionados con automatización y tecnología, de manera que los diversos sistemas de observación existentes tengan un mayor nivel de interoperabilidad (World Meteorological Organization, 2019).

En consecuencia, en este trabajo se analizará el Estado del arte para el desarrollo de un modelo que pueda clasificar automáticamente los diferentes tipos de nubes. Además, se preparará una base de datos de imágenes para entrenar el modelo desarrollado. El proyecto se realizó en conjunto con investigadores del grupo IA-PUCP y aportará al desarrollo del proyecto Meteo-Huascarán, ejecutado por el SENAMHI en asociación con la PUCP, con el objetivo de disminuir el riesgo asociado al ecoturismo en el Parque Nacional Huascarán mediante el

desarrollo de un sistema operacional automatizado de monitoreo del estado del tiempo y predicción de peligros meteorológicos (Servicio Nacional de Meteorología e Hidrología del Perú, 2021).

Este trabajo fue subvencionado por el CONCYTEC a través del programa PROCIENCIA en el marco del concurso de Investigación Aplicada 2021-02, según contrato N°036-2021.

1.1.3. Problema seleccionado.

El problema identificado para esta tesis es que la clasificación de tipos de nubes en las estaciones meteorológicas del SENAMHI es un proceso que requiere estrictamente de un observador, por lo que es propensa a errores y laboriosa.

1.2. Presentación de objetivos, resultados esperados y estrategias de verificación

1.2.1. Objetivo general.

El objetivo general del presente trabajo es desarrollar un modelo de red neuronal convolucional que contribuya a automatizar la detección del tipo de nube en las estaciones meteorológicas del SENAMHI.

1.2.2. Objetivos específicos.

O 1. Preparar una base de datos de imágenes de tipos de nubes que sea representativa, suficientemente heterogénea y grande para el entrenamiento adecuado de un modelo de clasificación automática de tipos de nubes.

O 2. Implementar un modelo basado en una red neuronal convolucional que permita clasificar tipos de nubes automáticamente.

1.2.3. Resultados esperados por objetivos.

Para el objetivo específico 1:

R 1. *Pipeline* para la creación de la base de datos de imágenes de tipos de nubes.

R 2. Base de datos de imágenes de tipos de nubes preparada y separada adecuadamente para el proceso de entrenamiento y prueba.

Para el objetivo específico 2:

R 3. Revisión sistemática de arquitecturas de redes neuronales convolucionales que han sido usadas para la clasificación automática de tipos de nubes.

R 4. Implementación de las arquitecturas de redes neuronales convolucionales más efectivas y relevantes.

R 5. Evaluación comparativa del desempeño de distintas arquitecturas de redes neuronales convolucionales en la base de datos preparada.

R 6. Desarrollo de una interfaz web que muestre el funcionamiento del modelo de red neuronal convolucional implementado.

1.2.4. Mapeo de objetivos, resultados y verificación.

Para evaluar el logro de cada objetivo se elaboraron las siguientes tablas:

Tabla 1

Mapeo del objetivo específico 1, resultados y verificación

| Objetivo: Preparar una base de datos de imágenes de tipos de nubes que sea representativa, suficientemente heterogénea y grande para el entrenamiento adecuado de un modelo de clasificación automática de tipos de nubes | | |
|--|--|---|
| Resultado | Medio de verificación | Indicador objetivamente verificable |
| R1: Pipeline para la creación de la base de datos de imágenes de tipos de nubes | <ul style="list-style-type: none"> • Diagrama de flujo del pipeline de creación de la base de datos • Código fuente que implementa el pipeline de creación de la base de datos | <ul style="list-style-type: none"> • Pipeline de creación de la base de datos validado por un experto en análisis de datos observacionales en meteorología o climatología • Pruebas unitarias satisfechas al 100% |

| | | |
|---|--|--|
| <p>R2: Base de datos de imágenes de tipos de nubes preparada y separada adecuadamente en para el proceso de entrenamiento y prueba</p> | <ul style="list-style-type: none"> Archivos de imágenes de tipos de nubes organizados por conjunto y por clase Archivos de metadatos adicionales | <ul style="list-style-type: none"> La base de datos creada contiene al menos 300 imágenes por clase |
|---|--|--|

Tabla 2

Mapeo del objetivo específico 2, resultados y verificación

| <p>Objetivo: Implementar un modelo basado en una red neuronal convolucional que permita clasificar tipos de nubes automáticamente.</p> | | |
|--|--|---|
| Resultado | Medio de verificación | Indicador objetivamente verificable |
| <p>R3: Revisión sistemática de arquitecturas de redes neuronales convolucionales que han sido usadas para la clasificación automática de tipos de nubes</p> | <ul style="list-style-type: none"> Reporte de arquitecturas de redes neuronales convolucionales que han sido usadas para la clasificación de tipos de nubes | <ul style="list-style-type: none"> Reporte validado por un experto en aprendizaje de máquina |
| <p>R4: Implementación de las arquitecturas de redes neuronales convolucionales más efectivas y relevantes</p> | <ul style="list-style-type: none"> Código fuente de las arquitecturas de redes neuronales convolucionales implementadas | <ul style="list-style-type: none"> Por lo menos 3 arquitecturas de redes neuronales convolucionales implementadas Pruebas unitarias satisfechas al 100% |
| <p>R5: Evaluación comparativa del desempeño de distintas arquitecturas de redes neuronales convolucionales en la base de datos preparada</p> | <ul style="list-style-type: none"> Reporte de evaluación comparativa del desempeño de las arquitecturas de redes neuronales convolucionales implementadas | <ul style="list-style-type: none"> Reporte de evaluación comparativa validado por un experto en aprendizaje de máquina |
| <p>R6: Desarrollo de una interfaz web que muestre el funcionamiento del modelo de red neuronal</p> | <ul style="list-style-type: none"> Lista de requerimientos que debe satisfacer la interfaz web | <ul style="list-style-type: none"> Lista de requerimientos validada por un experto en aprendizaje de máquina |

| | | |
|----------------------------|---|---|
| convolucional implementado | <ul style="list-style-type: none"> • Interfaz web desplegada en la nube y lista para ser usada • Resultados de pruebas unitarias de integración | <ul style="list-style-type: none"> • 100% de las pruebas funcionales satisfechas |
|----------------------------|---|---|

1.3. Métodos y Procedimientos

En esta parte, se proporcionará una breve descripción de las herramientas, métodos y procedimientos que se utilizarán con el fin de cumplir los objetivos específicos definidos en la sección 1.2.2. A continuación, se muestra un mapeo de herramientas, métodos y procedimientos por resultado.

Tabla 3

Mapeo de herramientas, métodos y procedimientos por resultado

| Resultado | Herramientas | Métodos y procedimientos |
|--|---|--|
| R1. Pipeline para la creación de la base de datos de imágenes de tipos de nubes | <ul style="list-style-type: none"> • Python • Open CV • Jupyter lab • Lucid Chart • Git • Google Drive | <ul style="list-style-type: none"> • Pruebas unitarias • Web scraping |
| R2. Base de datos de imágenes de tipos de nubes preparada y separada adecuadamente para el proceso de entrenamiento y prueba | <ul style="list-style-type: none"> • Mendeley | <ul style="list-style-type: none"> • Estado del arte |
| R3. Revisión sistemática de arquitecturas de redes neuronales convolucionales que han sido usadas para la clasificación automática de tipos de nubes | <ul style="list-style-type: none"> • Python • PyTorch • OpenCV • Jupyter Lab • Visual Studio Code • Albuementations | <ul style="list-style-type: none"> • Aprendizaje supervisado usando redes neuronales convolucionales • Técnicas de aumento de datos • Pruebas unitarias • Early stopping |
| R4. Implementación de las arquitecturas de redes neuronales convolucionales más efectivas y relevantes | <ul style="list-style-type: none"> • Python • PyTorch • Weights & Biases • Open CV • Jupyter lab • Visual Studio Code | <ul style="list-style-type: none"> • Técnicas de optimización de hiperparámetros |
| R5. Evaluación comparativa del desempeño de distintas arquitecturas de redes neuronales convolucionales en la base de datos preparada | | |

| | | |
|--|---|---|
| R6. Desarrollo de una interfaz web que muestre el funcionamiento del modelo de red neuronal convolucional implementado | <ul style="list-style-type: none"> • Python • Vue.js • FastAPI • AMAZON EC2 • Visual Studio Code • Google Drive • AMAZON S3 • Figma | <ul style="list-style-type: none"> • Pruebas funcionales |
|--|---|---|

1.3.1. Herramientas¹.

- Python. Este lenguaje de programación es ampliamente reconocido por su uso en técnicas de aprendizaje automático e inteligencia artificial. Dentro del contexto del proyecto, se empleó Python para implementar el pipeline necesario para crear la base de datos, desarrollar modelos de redes neuronales convolucionales y construir el backend de la interfaz web.
- Open CV. Esta biblioteca, compatible con varios lenguajes de programación, incluyendo Python, es utilizada en aplicaciones de visión computacional. A lo largo del proyecto, Open CV se usó para procesar imágenes y aumentar datos.
- Jupyter Lab. Herramienta de código abierto que permite programar en Python de manera interactiva, es usada particularmente con fines relacionados a la inteligencia artificial. Esta herramienta fue usada para programar los modelos de redes neuronales convolucionales en primera instancia; es decir, para realizar los primeros experimentos y visualizar resultados preliminares, ya que posteriormente los experimentos serán automatizados usando scripts de Python.
- Lucid Chart. Aplicación web para la generación de distintos tipos de diagramas, incluyendo diagramas de flujo. Se utilizó esta herramienta para modelar el diagrama de flujo del pipeline de creación de la base de datos.

¹ Para conocer más sobre estas herramientas, sus páginas web respectivas se encuentran consignadas en el Anexo A.

- Google Drive. Sistema de almacenamiento en línea, se usó para almacenar de manera segura la base de datos generada.
- Git. Programa que sirve para el control de versiones. Esta herramienta fue usada para controlar el versionado de todo el código desarrollado y para almacenarlo en la nube.
- Mendeley. Gestor de referencias de Elsevier Inc., permite la organización de diversas fuentes de información y facilita su citación. Esta herramienta fue usada en la revisión sistemática de tipo Estado del arte.
- PyTorch. Framework de Python de código abierto usado para aplicaciones en aprendizaje profundo, provee diferentes funcionalidades que son útiles para la implementación de redes neuronales. Se utilizó para desarrollar las arquitecturas de redes neuronales convolucionales y para llevar a cabo su evaluación comparativa.
- Visual Studio Code. Entorno de desarrollo integrado de Microsoft que permite programar en varios lenguajes de programación como Python y Javascript. Esta herramienta se empleó para escribir código a lo largo de todo el proyecto.
- Vue.js. Este framework de JavaScript se utiliza para la creación de interfaces web. En este proyecto, Vue.js fue utilizado para implementar la interfaz web que muestra cómo funciona la red neuronal desarrollada.
- FastAPI. Framework para el desarrollo de APIs que se caracteriza por su uso intuitivo, desarrollo rápido y alto rendimiento (Ramírez, s.f.). En este trabajo, se empleó para desarrollar las APIs necesarias para el funcionamiento de la interfaz web a implementar.
- AMAZON EC2. Herramienta que permite el uso de recursos computacionales en la nube (Amazon Web Services Inc., s.f.-a). En este proyecto, se utilizó para el despliegue de las APIs necesarias para el funcionamiento de la interfaz web a implementar.
- Weights & Biases. Esta herramienta se utiliza para visualizar variables mediante diversos gráficos en la implementación de modelos de aprendizaje automático. En el

proyecto, se empleó para la evaluación comparativa de arquitecturas de redes neuronales convolucionales y para visualizar los resultados de los experimentos realizados.

- AMAZON S3. Este servicio de almacenamiento en la nube es capaz de alojar sitios web estáticos (Amazon Web Services Inc., s.f.-b). Esta herramienta se usó para el despliegue de la interfaz web que muestra el funcionamiento de la red neuronal.
- Albuumentations. Librería rápida y flexible para realizar aumentación de imágenes. Es ampliamente usada en la industria, la investigación en aprendizaje profundo y las competencias de aprendizaje de máquina (Buslaev, Iglovikov, Khvedchenya, Parinov, Druzhinin, & Kalinin, 2020). Esta librería se usó en este trabajo para facilitar la implementación de las técnicas de aumentación de datos necesarias para el adecuado entrenamiento de redes neuronales convolucionales.

1.3.2. Métodos y procedimientos.

Para el desarrollo de esta investigación, se han aplicado diversos métodos de recopilación y generación de datos, los cuales se detallan a continuación.

- Estado del arte

El Estado del arte permite determinar, en un momento cronológico determinado, la manera en la que un tema de investigación ha sido abordado, cómo se encuentra el avance al respecto y cuáles son las tendencias (Londoño Palacio, Maldonado Granados, & Calderón Villafañez, 2014). Se usó para conocer cuáles son las arquitecturas de redes neuronales convolucionales que han sido usadas para clasificar automáticamente tipos de nubes.

- *Web scraping*

Web scraping sirve para extraer información de uno o más sitios web y procesarla para obtener estructuras simples como hojas de cálculo o archivos CSV. Un web scraper es un

software que imita la navegación humana en internet para recolectar datos de diferentes páginas web. La ventaja de un scraper radica en su rapidez y su capacidad para ser automatizado y/o programado (Diouf, Sarr, Sall, Birregah, Bousso, & Mbaye, 2019). En este trabajo, este método fue utilizado para obtener imágenes de tipos de nubes adicionales de la web.

- Técnicas de optimización de hiperparámetros

Estas técnicas en el contexto de aprendizaje profundo consisten en la exploración de un amplio espacio de arquitecturas de redes neuronales y valores de hiperparámetros, de manera que se pueda llegar a las configuraciones que funcionan mejor para el problema específico (Ranjit, Ganapathy, Sridhar, & Arumugham, 2019). Algunos de los enfoques en optimización de hiperparámetros son los siguientes: *grid search* y optimización bayesiana. En este proyecto, se usaron estas técnicas para obtener la configuración que maximizaba el desempeño de las redes neuronales implementadas antes de realizar la respectiva evaluación comparativa.

- Técnicas de aumento de datos

Esta técnica se enfoca en la ampliación del conjunto de imágenes etiquetadas para el entrenamiento y está probado que es efectiva para mitigar el sobreajuste (Liu, Lehman, & Alregib, 2020). A continuación, se mencionan algunas técnicas de aumento de datos comunes: recorte aleatorio, volteo horizontal aleatorio, volteo vertical aleatorio, desenfoque gaussiano, etc. Estas técnicas se usarán para aumentar el tamaño del conjunto de entrenamiento antes de entrenar los distintos modelos de redes neuronales convolucionales.

- *Early stopping*

Early stopping es una técnica ampliamente usada para evitar el sobreajuste, debido a su facilidad de entendimiento e implementación. Este método consiste en detener el proceso de entrenamiento cuando el error calculado en el conjunto *validation* supere al error en el conjunto *train*. Este enfoque usa el conjunto *validation* para anticipar el comportamiento del modelo en

el conjunto *test*, asumiendo que el error será similar en ambos (Prechelt, 1998). En el presente trabajo, este método se implementó durante el entrenamiento para mitigar el sobreajuste.

- Pruebas unitarias

Las pruebas unitarias permiten detectar errores en una fase temprana del desarrollo de software (Tillmann, De Halleux, & Xie, 2010). Esta metodología fue usada para garantizar el correcto funcionamiento de los siguientes módulos del proyecto por separado: el pipeline de creación de la base de datos y las redes neuronales.

- Pruebas funcionales

Las pruebas funcionales consisten en evaluar las funciones de un sistema para asegurar que se cumplen los requerimientos y garantizar la calidad. (Kobrosly & Vassiliadis, 1988). En este proyecto, se usó para asegurar el correcto funcionamiento de los siguientes módulos del proyecto en conjunto: las arquitecturas de redes neuronales convolucionales, el servicio que integra el modelo final y la interfaz web que muestra el funcionamiento del mismo.

- Aprendizaje supervisado

La forma más común de aprendizaje de máquina es el aprendizaje supervisado, enfoque que puede ser utilizado para la clasificación automática de imágenes, por ejemplo, se podría construir un sistema capaz de discernir entre imágenes de casas, autos, personas y mascotas. Para esto, sería necesario contar con una base de datos grande y etiquetada que contenga imágenes de todas las clases anteriormente mencionadas, estas imágenes serán usadas para entrenar el modelo de aprendizaje de máquina, de manera que sus parámetros sean ajustados minimizando la función objetivo que mide el error (LeCun, Bengio, & Hinton, 2015). En este trabajo de tesis, se utilizará el aprendizaje supervisado para clasificar tipos de nubes automáticamente.

- Redes Neuronales

Una red neuronal estándar, como la que puede visualizarse en la Figura 2, está compuesta por varios procesadores simples llamados neuronas, donde cada una produce una secuencia de activaciones de valor numérico. Las neuronas de entrada se activan a través de sensores que perciben el entorno, mientras las demás neuronas se activan a través de conexiones ponderadas a partir de neuronas previamente activadas. El aprendizaje consiste en encontrar los pesos que hagan que la red neuronal muestre el comportamiento deseado, como por ejemplo clasificar imágenes (Schmidhuber, 2015).

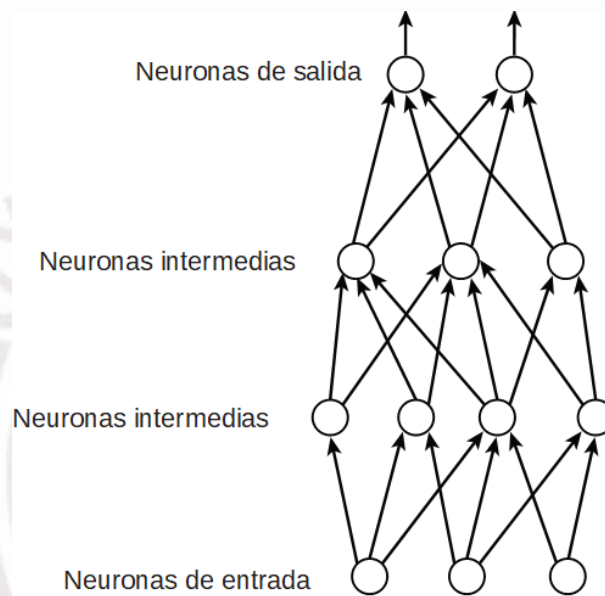


Figura 2. Diagrama de red neuronal estándar.

Adaptado de LeCun et al. (2015).

Durante el entrenamiento de una red neuronal, el algoritmo de aprendizaje ajusta los pesos utilizando un vector de gradiente que indica cómo cambiarían los pesos para minimizar el error. El cálculo de estos gradientes se realiza mediante la técnica conocida como *backpropagation*. La función objetivo puede ser vista como un paisaje montañoso en busca del punto más bajo. Una red neuronal típica puede tener cientos de millones de pesos, así como cientos de millones de ejemplos etiquetados con los cuales entrenar el modelo (LeCun et al., 2015).

- Redes neuronales convolucionales (CNN)

Estas redes neuronales están diseñadas para trabajar con datos estructurados en múltiples matrices. Un ejemplo de esto son las imágenes a color, que se representan mediante 3 matrices 2D, cada una corresponde a las intensidades de los píxeles en los canales rojo, verde y azul (RGB) (LeCun et al., 2015).

La estructura de una red neuronal convolucional, como se ilustra en la Figura 3, se organiza en varias etapas. Las primeras consisten en 2 tipos de capas: convolucionales y de *pooling*. Las unidades dentro de una capa convolucional están organizadas en *feature maps*, donde cada unidad se conecta a regiones locales de los feature maps de la capa anterior mediante un conjunto de pesos denominado banco de filtros. El resultado de esta suma ponderada se pasa a través de una no linealidad como “ReLU” (LeCun et al., 2015).

Este diseño de pesos compartidos tiene 2 objetivos. En primer lugar, en las imágenes, los grupos locales de valores suelen estar altamente correlacionados formando patrones locales distintivos que se detectan fácilmente. En segundo lugar, si un patrón aparece en una parte de la imagen, este podría aparecer en cualquier otra parte, de ahí la idea de que las unidades en diferentes lugares compartan los mismos pesos. Matemáticamente, la operación de filtrado realizada por un feature map es una convolución discreta, de ahí el nombre de este tipo de red neuronal. Mientras el papel de una capa convolucional es detectar patrones, el papel de una capa de pooling es fusionar características similares en una sola, así se reduce la dimensionalidad y se genera invariabilidad ante pequeños desplazamientos y distorsiones. Para construir una red neuronal convolucional, se apilan 2 o 3 capas de convolución, una no linealidad y una capa de pooling, seguidas de más capas convolucionales y capas completamente conectadas. Aplicar la técnica backpropagation en una red neuronal convolucional es tan sencillo como en una red neuronal normal; esto permite entrenar todos los pesos de todos los bancos de filtros (LeCun et al., 2015).

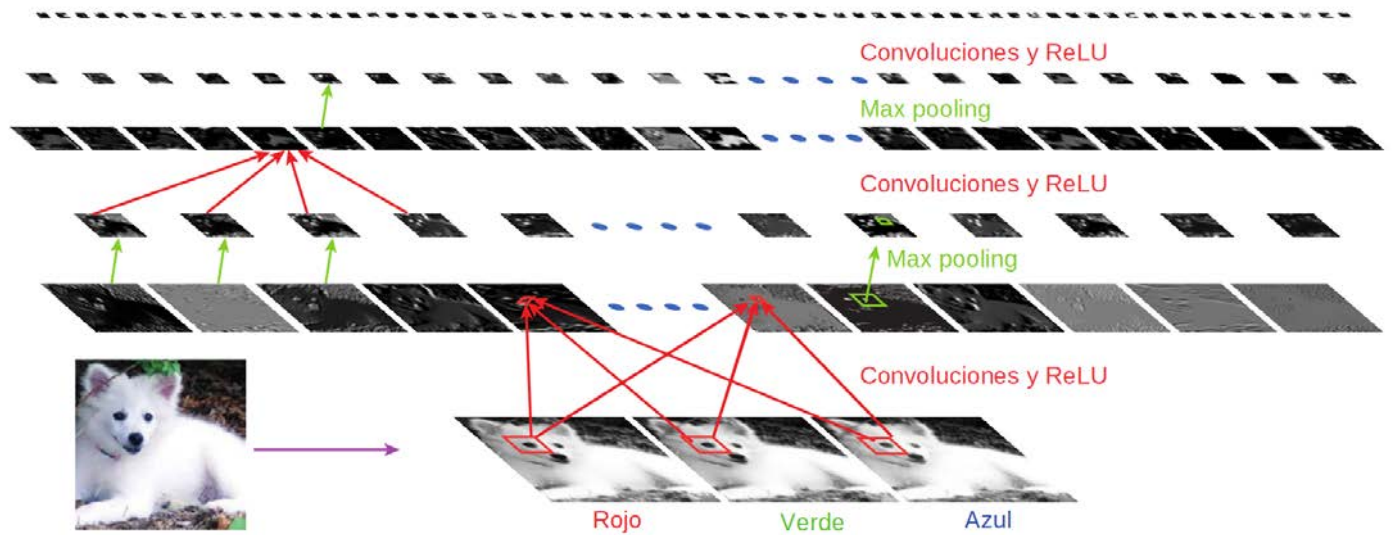
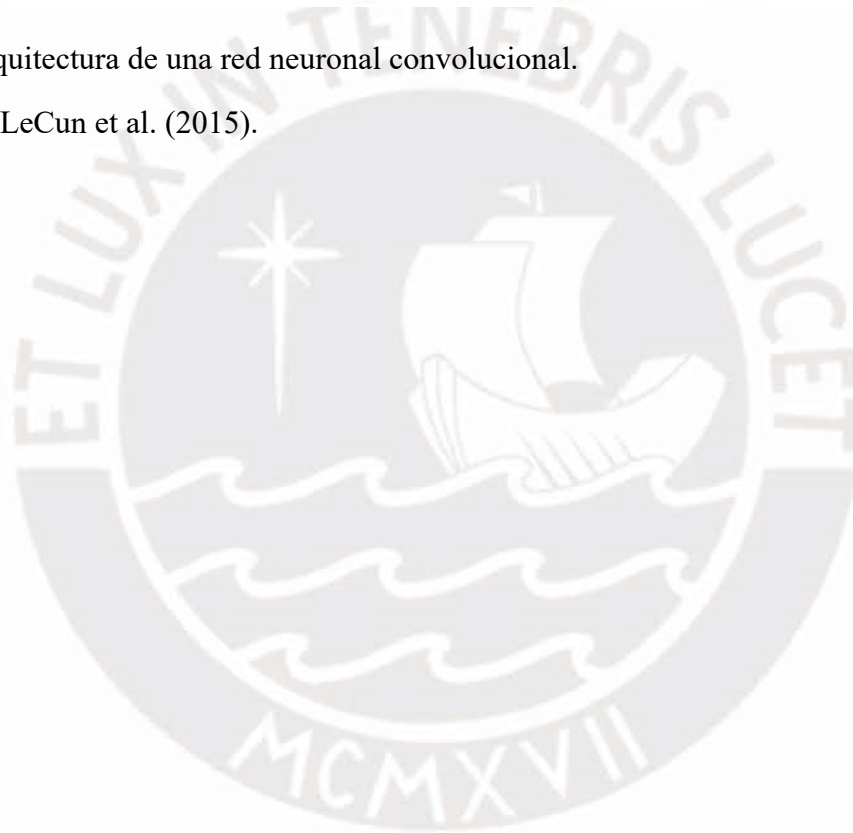


Figura 3. Arquitectura de una red neuronal convolucional.

Adaptado de LeCun et al. (2015).



Capítulo 2. Marco Legal/Regulatorio/Conceptual/otros

En este capítulo se desarrollan las definiciones de los conceptos necesarios para el adecuado entendimiento de la problemática abordada en el capítulo 1: la detección del tipo de nube en las estaciones automáticas del SENAMHI requiere estrictamente de un observador, por lo que es propensa a errores y laboriosa. Por ello, a continuación, se definirán los conceptos relacionados.

2.1. Meteorología en el Perú

La meteorología es la disciplina que estudia el clima y las condiciones del tiempo mediante el estudio de la atmósfera, la extensa capa de aire en la que habitamos (Ackerman & Knox, 2011). En Perú, el Servicio Nacional de Meteorología e Hidrología del Perú (SENAMHI) tiene como objetivo generar y brindar información y conocimientos meteorológicos, hidrológicos y climáticos de manera confiable, oportuna y accesible para el beneficio de la sociedad peruana. Una de las responsabilidades de esta entidad es el análisis y pronóstico de fenómenos a corto plazo, tales como lluvias, vientos fuertes y bajas temperaturas, con el fin de obtener información para la toma de decisiones (Servicio Nacional de Meteorología e Hidrología del Perú, s.f.). Por ello, el SENAMHI dispone de una red de estaciones meteorológicas distribuidas por todo el país (Servicio Nacional de Meteorología e Hidrología del Perú, 2013).

2.2. Nube

Una nube es un hidrometeoro compuesto por diminutas partículas de agua líquida, hielo, o una combinación de ambas, que permanecen suspendidas en la atmósfera y generalmente no alcanzan el suelo (World Meteorological Organization, 2017).

2.3. Clasificación de nubes

Las nubes evolucionan continuamente y aparecen en una infinita variedad de tipos. Sin embargo, hay un número limitado de tipos característicos que se observan con frecuencia en todo el mundo, esto permite agrupar a las nubes a grandes rasgos en un esquema de clasificación. Este esquema contiene géneros, especies y variedades, de manera similar a la clasificación de plantas y animales (World Meteorological Organization, 2017).

El enfoque de este trabajo es la clasificación automática de géneros de nubes, a los cuales se hace referencia, en algunos casos, como tipos de nubes. Estos serán definidos a continuación.

2.4. Géneros de nubes

A continuación, se detallan las características de los 10 géneros de nubes que existen según la Organización Meteorológica Mundial y los fenómenos meteorológicos que causan. Además, podrán encontrar una imagen referencial de cada uno en la Figura 4.

1. Cirros

Nubes con apariencia de filamentos blancos y delicados, o que se presentan en forma de manchas o bandas estrechas de color blanco. Estas nubes tienen una textura fibrosa similar a un cabello (World Meteorological Organization, 2017).

2. Cirrocúmulos

Una capa de nubes delgada y blanca sin sombras, formada por pequeños elementos con forma de ondulaciones o gránulos, dispuestos de manera más o menos regular (World Meteorological Organization, 2017).

3. Cirrostratos

Una nube con forma de velo transparente, blanquecino y de apariencia lisa o fibrosa. Un cirrostrato cubre de manera total o parcial el cielo, y a menudo produce efectos de halo (World Meteorological Organization, 2017).

4. Altocúmulos

Capa de nubes blanca o gris, generalmente con sombreado, compuesta por láminas, masas redondeadas o rollos. Algunas veces, son parcialmente difusas o fibrosas (World Meteorological Organization, 2017).

5. Altoestratos

Una capa de nubes grisácea o azulada con un aspecto uniforme, estriado o fibroso. Tiene partes finas que permiten revelar el sol y no genera fenómenos de halo. Puede presentar precipitaciones (World Meteorological Organization, 2017).

6. Nimbostratos

Una capa de nubes de color gris oscuro que produce lluvia o nieve de manera continua. Es suficientemente gruesa como para tapar el sol (World Meteorological Organization, 2017).

7. Estratocúmulos

Una capa de nubes grises o blanquecinas que casi siempre muestra secciones oscuras. Está formada por teselaciones, masas redondeadas o rollos que pueden estar o no fusionados. Presenta precipitaciones de intensidad débil (World Meteorological Organization, 2017).

8. Estratos

Una capa de nubes grises con una base uniforme que puede provocar llovizna o nieve, el sol puede ser visible o no a través de estas nubes. Los estratos no presentan fenómenos de halo, excepto en condiciones de temperatura extremadamente bajas (World Meteorological Organization, 2017).

9. Cúmulos

Nubes con contornos bien definidos, aisladas y densas que se desarrollan verticalmente en forma de montículos cúpulas o torres. Estas nubes son de color blanco brillante y tienen una base oscura y horizontal. Pueden causar precipitaciones en forma de chubascos (World Meteorological Organization, 2017).

10. Cumulonimbos

Nube densa con gran extensión vertical que tiene forma de torre o montaña, con una parte superior casi siempre aplanada. Su base es generalmente muy oscura. Estas nubes pueden producir precipitaciones intensas, a menudo acompañadas de rayos (World Meteorological Organization, 2017).



Figura 4. Imágenes representativas de cada género de nube.

Tomado de <https://public.wmo.int/en/WorldMetDay2017/classifying-clouds>

Capítulo 3. Estado del Arte

Este capítulo tiene como objetivo investigar la literatura más reciente con respecto a la clasificación de tipos de nubes basada en imágenes tomadas desde tierra. Por ello, se revisará el contexto actual de soluciones planteadas por medio de una revisión sistemática.

3.1. Objetivos de revisión

Las redes neuronales convolucionales han revolucionado el procesamiento de imágenes al automatizar la extracción de características, eliminando la necesidad de hacerlo manualmente (LeCun, Bottou, Bengio & Haffner, 1998). A partir de los años 2000, estas redes se han utilizado con éxito en diversas aplicaciones de procesamiento de imágenes (LeCun et al., 2015). Por ello, el primer objetivo de esta revisión sistemática de tipo estado del arte es examinar el uso de estas redes neuronales en la clasificación de imágenes, centrando la atención en imágenes de tipos de nubes tomadas desde la superficie terrestre. Un segundo objetivo es investigar las bases de datos de imágenes de nubes actualmente disponibles, identificando cuántos tipos de nubes contienen y cuáles son. Finalmente, se estudiará qué técnicas de preprocesamiento de imágenes se aplicaron en la clasificación de tipos de nubes y cómo influyen en el rendimiento del modelo de red neuronal.

3.2. Preguntas de revisión

- P1. ¿Qué arquitecturas de redes neuronales convolucionales se han utilizado para la clasificación de tipos de nubes con imágenes tomadas desde tierra, cuáles fueron los resultados y con qué métrica se evaluaron?
- P2. ¿Qué bases de datos se han utilizado para la clasificación de nubes con imágenes tomadas desde tierra y en qué clases se etiquetaron las imágenes?

- P3. ¿Qué técnicas de preprocesamiento de datos se han utilizado para la clasificación de nubes con imágenes tomadas desde tierra y de qué manera ayudaron a mejorar los resultados?

3.3. Estrategia de búsqueda

Esta sección describe la estrategia de búsqueda, la cual incluye la definición de motores y cadenas de búsqueda para identificar artículos relevantes que contribuyan a responder las preguntas formuladas en la revisión.

3.3.1. Motores de búsqueda a usar.

Debido a su relevancia en temas de ciencias de la computación, se eligieron los siguientes repositorios:

- IEEE/IET Electronic Library
- Scopus
- Web of Science

3.3.2. Cadenas de búsqueda a usar.

Se definió una cadena de búsqueda por cada motor y por cada pregunta de revisión.

- Pregunta 1

Palabras clave:

- Ground-based, ground
- Cloud classification, cloud identification
- Convolution, convolutional
- Neural networks, deep learning

Cadenas de búsqueda:

- IEEE/IET Electronic Library

("All Metadata": "cloud classification" OR "All Metadata": "cloud identification") AND ("All Metadata": "ground" OR "All Metadata": "ground-based") AND ("All Metadata": "convolution*") AND ("All Metadata": "neural network*" OR "All Metadata": "deep learning")

- Scopus

TITLE-ABS-KEY ("cloud classification" OR "cloud identification") AND TITLE-ABS-KEY ("ground-based" OR "ground") AND TITLE-ABS-KEY ("deep learning" OR "neural network*") AND TITLE-ABS-KEY ("convolution*")

- Web of Science

((ALL=("cloud classification" OR "cloud identification"))) AND ALL=("ground" or "ground-based")) AND ALL=("deep learning" or "neural network*")) AND ALL=("convolution*")

Tabla 4

Cantidad de artículos encontrados para la pregunta 1

| Motor de búsqueda | Artículos encontrados |
|-----------------------------|-----------------------|
| IEEE/IET Electronic Library | 39 |
| Scopus | 41 |
| Web of Science | 26 |

- Pregunta 2

Palabras clave:

- Ground-based, ground
- Cloud classification, cloud identification

- Database, dataset

Cadenas de búsqueda:

- IEEE/IET Electronic Library

("All Metadata":"cloud classification" OR "All Metadata":"cloud identification") AND ("All Metadata":"ground" OR "All Metadata":"ground-based") AND ("All Metadata":"database" OR "All Metadata":"dataset")

- Scopus

TITLE-ABS-KEY ("cloud classification" OR "cloud identification") AND TITLE-ABS-KEY ("ground-based" OR "ground") AND TITLE-ABS-KEY("database" OR "dataset")

- Web of Science

((ALL=("cloud classification" OR "cloud identification")) AND ALL=("ground" OR "ground-based")) AND ALL=("dataset" OR "database")

Tabla 5

Cantidad de artículos encontrados para la pregunta 2

| Motor de búsqueda | Artículos encontrados |
|-----------------------------|-----------------------|
| IEEE/IET Electronic Library | 10 |
| Scopus | 90 |
| Web of Science | 39 |

- Pregunta 3

Palabras clave:

- Ground-based, ground
- Cloud classification, cloud identification

- Preprocessing, processing

Cadenas de búsqueda:

- IEEE/IET Electronic Library

("All Metadata": "cloud classification" OR "All Metadata": "cloud identification") AND ("All Metadata": "ground" OR "All Metadata": "ground-based") AND ("Full Text & Metadata": "processing" or "preprocessing")

- Scopus

TITLE-ABS-KEY ("cloud classification" OR "cloud identification") AND TITLE-ABS-KEY ("ground-based" OR "ground") AND TITLE-ABS-KEY("preprocessing" OR "processing")

- Web of Science

((ALL=("cloud classification" OR "cloud identification")) AND ALL=("ground" OR "ground-based")) AND ALL=("preprocessing" OR "processing")

Tabla 6

Cantidad de artículos encontrados para la pregunta 3

| Motor de búsqueda | Artículos encontrados |
|-----------------------------|-----------------------|
| IEEE/IET Electronic Library | 6 |
| Scopus | 80 |
| Web of Science | 32 |

3.3.3. Criterios de inclusión/exclusión.

- Se incluirán los trabajos publicados desde el 2016 en adelante.
- Se excluirán todos los trabajos que no estén escritos en español o inglés.

- Se excluirán aquellos artículos cuya solución no esté basada en una red neuronal convolucional.

- Se comprobará que el estudio utilice una base de datos con imágenes de nubes capturadas desde la superficie terrestre.

- Se excluirán los trabajos que solo contengan resúmenes de investigaciones previas.

- Se excluirán todos los trabajos que hagan referencia a “nube de puntos”.

3.3.4. Documentos encontrados.

Para reunir los datos, el número de documentos encontrados por cada motor de búsqueda, tras aplicar los criterios de inclusión/exclusión y borrar duplicados, se presenta en la siguiente tabla.

Tabla 7

Resumen de documentos encontrados

| Motor de búsqueda | Artículos encontrados | Sin duplicados |
|-----------------------------|-----------------------|----------------|
| IEEE/IET Electronic Library | 9 | 9 |
| Scopus | 18 | 9 |
| Web of Science | 16 | 1 |

Asimismo, se elaboró un índice donde se encuentran las fuentes que se usarán para responder las preguntas anteriormente planteadas. Se definió un identificador para cada fuente y también se incluyó la citación correspondiente.

Tabla 8

Formulario de extracción de datos

| ID | Título | Citación |
|------|--|---|
| P001 | Ground-Based Remote Sensing Cloud Classification via Context Graph Attention Network | (Liu, Duan, Zhang, Cao & Durrani, 2021) |

| | | |
|------|---|--|
| P002 | Cloud classification of ground-based cloud images based on convolutional neural network | (Zhu, Wei, & Guo, 2021) |
| P003 | Transfer Learning Approach - An Efficient Method to Predict Rainfall Based on Ground-Based Cloud Images | (Ambildhuke & Banik, 2021) |
| P004 | A novel robust classification method for ground-based clouds | (Yu, Tang, Li, Hou, Xuan, Zhu & Chen, 2021) |
| P005 | Ground-based Cloud Recognition Based on Lightweight Neural Network | (Zhang, Jia, Liu, & Fang, 2020) |
| P006 | Multimodal Ground-Based Remote Sensing Cloud Classification via Learning Heterogeneous Deep Features | (Liu, Duan, Zhang, Cao & Durrani, 2020) |
| P007 | Clouda: A ground-based cloud classification method with a convolutional neural network | (Wang, Zhou, Yang & Liu, 2020) |
| P008 | Ground-Based Cloud Classification Using Task-Based Graph Convolutional Network | (Liu, Li, Zhang, Cao & Durrani, 2020) |
| P009 | Deep tensor fusion network for multimodal ground-based cloud classification in weather station networks | (Li, Liu & Zhang, 2020) |
| P010 | Hierarchical Multimodal Fusion for Ground-Based Cloud Classification in Weather Station Networks | (Liu, Duan, Zhang & Cao, 2019) |
| P011 | Dual Guided Loss for Ground-Based Cloud Classification in Weather Station Networks | (Li, Liu & Zhang, 2019) |
| P012 | Deep multimodal fusion for ground-based cloud classification in weather station networks | (Liu & Li, 2018) |
| P013 | CloudNet: Ground-Based Cloud Classification With Deep Convolutional Neural Network | (Zhang, Liu, Zhang & Song, 2018) |
| P014 | Multimodal ground-based cloud classification using joint fusion convolutional neural network | (Liu, Li, Zhang, Xiao & Cao, 2018) |
| P015 | Salient dual activations aggregation for ground-based cloud classification in weather station networks | (Zhang, Li & Liu 2018) |
| P016 | DeepCloud: Ground-Based Cloud Image Categorization Using Deep Convolutional Features | (Ye, Cao & Xiao, 2017) |
| P017 | Deep Convolutional Activations-Based Features for Ground-Based Cloud Classification | (Shi, Wang, Wang & Xiao, 2017) |
| P018 | Cloud-type Classification of Ground-Based Images using Deep Learning | (Sinko, Kamencay, Sykora, Benco & Hudec, 2019) |

| | | |
|------|---|---|
| P019 | Ensemble Meteorological Cloud Classification Meets Internet of Dependable and Controllable Things | (Zhang, Liu, Zhang, Iwabuchi, de Moura & de Albuquerque 2021) |
|------|---|---|

3.4. Formulario de extracción de datos

El formulario de extracción de datos empleado para la revisión sistemática se detalla a continuación. La versión completa se puede consultar en el Anexo B.

Tabla 9

Formulario de extracción de datos

| Campo | Descripción | Preguntas |
|----------------------|--|------------------|
| Identificador | Identificador personal para la referencia (Ej. P001) | General |
| DOI | Identificador global de la referencia | General |
| Autores | Autores del trabajo de investigación | General |
| Fecha de publicación | Fecha de publicación del trabajo de investigación | General |
| Título | Título del trabajo de investigación | General |
| Abstract | Resumen del contenido del documento | General |
| Enfoque utilizado | Enfoque que se utilizó para la resolución del problema (Ej. Red neuronal convolucional, máquinas de vectores de soporte, mecanismos de atención) | P1 |
| Descripción técnica | Descripción del enfoque que se utilizó, es decir la arquitectura específica y diseño del modelo | P1 |

| | | |
|---------------------------------------|---|--------|
| Estrategia de entrenamiento | Procedimiento que se siguió para entrenar el modelo (Ej. <i>Transfer learning</i> , entrenamiento desde cero) | P1 |
| Bases de datos | Nombres de las bases de datos que se utilizaron | P2 |
| Características de las bases de datos | Descripción de las bases de datos (cantidad de imágenes, cantidad de clases, nombres de las clases) | P2 |
| Preprocesamiento de datos | Técnicas que se usaron para procesar las imágenes antes de ser procesadas por el modelo | P3 |
| Métrica de evaluación | Nombre de la métrica que se usó para evaluar el modelo | P1, P3 |
| Resultados | Resultado de la métrica | P1, P3 |

3.5. Resultados de la revisión

A continuación, se proporcionan las respuestas a cada una de las preguntas formuladas en la revisión.

3.5.1. P1 ¿Qué arquitecturas de redes neuronales convolucionales se han utilizado para la clasificación de nubes con imágenes tomadas desde tierra, cuáles fueron los resultados y con qué métrica se evaluaron?

Las arquitecturas de redes neuronales convolucionales que fueron usadas son, en su gran mayoría, arquitecturas ya existentes pre-entrenadas en ImageNet (Deng, Dong, Socher, Li, Kai & Li, 2009); es decir, se aplica *Transfer Learning*.

La arquitectura pre-entrenada más usada es ResNet y sus variaciones (He, Zhang, Ren & Sun, 2016). Por ejemplo, se propuso un enfoque que, además de las imágenes de nubes como tales, toma en cuenta la correlación que existe entre ellas mediante la elaboración de redes

neuronales de grafos (CGAT y TGCN) para la agregación de las características extraídas por la red neuronal convolucional principal: ResNet-50 (Liu, Duan, Zhang, Cao & Durrani, 2021; Liu, Li, Zhang, Cao & Durrani, 2020). Además, Li, Liu y Zhang (2019) se basan en el hecho de que hay nubes más difíciles de clasificar que otras y proponen un modelo que presenta a ResNet-50 pre-entrenada como arquitectura y una función de pérdida que prioriza clasificar correctamente las nubes más difíciles de categorizar. También, se propone el modelo RIWSRC de 2 canales que usa ResNet-50 e Inception v3 como estructuras centrales, donde finalmente la clasificación se realiza mediante *sparse coding* (Yu, Tang, Li, Hou, Xuan, Zhu & Chen, 2021). Por otro lado, otras variaciones de ResNet son usadas para experimentos: ResNet-18 (Zhu, Wei, & Guo 2021) y ResNet-152 (Zhang, Jia, Liu, & Fang, 2020). Finalmente, existen enfoques que usan, además de imágenes de nubes, información del tiempo como temperatura, humedad, presión, etc. En estos trabajos, ResNet-50 pre-entrenada es usada como arquitectura principal para el procesamiento de las imágenes (Li, Liu & Zhang, 2020; Liu, Duan, Zhang, Cao & Durrani, 2020; Liu, Li, Zhang, Xiao & Cao, 2018; Liu, Zhou, Wang, Peng, Shen & Zhou, 2019).

Otras arquitecturas pre-entrenadas empleadas para la solución de este problema, aunque con menos frecuencia, se mencionan a continuación: VGG16 (Ambildhuke & Banik, 2021; Zhang et al., 2020; Zhu et al., 2021), VGG19 (Zhang et al., 2020; Zhang, Li & Liu, 2018), Inception v3 (Ambildhuke & Banik, 2021; Yu et al., 2021; Zhang et al., 2020), Xception (Ambildhuke & Banik, 2021), DenseNet-201 (Zhang et al., 2020), AlexNet (Sinko, Kamencay, Sykora, Benco & Hudec, 2019; Zhu et al., 2021), Imagenet-vgg-f (Liu & Li, 2018; Shi, Wang, Wang & Xiao, 2017) e Imagenet-vgg-vd-16 (Shi et al., 2017).

Además del uso de arquitecturas pre-entrenadas, existen trabajos que proponen una arquitectura propia, la cual es entrenada desde cero. En primer lugar, CloudNet es una arquitectura basada en AlexNet que fue adaptada para reconocer mejor las características de

las nubes (Zhang et al., 2018). En segundo lugar, Clouda es una red neuronal que está compuesta por 4 capas convolucionales, 4 capas de pooling y 3 capas *fully-connected*, este modelo usa máquinas de soporte vectorial para realizar la clasificación final (Wang, Zhou, Yang & Liu, 2020). En tercer lugar, DeepCloud presenta una arquitectura que contiene 5 capas convolucionales, 3 capas de pooling y 3 capas fully-connected, la categorización final también se hace mediante máquinas de soporte vectorial (Ye, Cao & Xiao, 2017). En cuarto lugar, Ambildhuke y Banik, (2021) y Zhang, Liu, Zhang, Iwabuchi, de Moura y de Albuquerque (2021) implementan arquitecturas de redes neuronales convolucionales simples, los primeros para comparar su desempeño con arquitecturas pre-entrenadas y los segundos como parte de un modelo de ensamblado que agrega las predicciones de distintos clasificadores como AdaBoost o Super Learner. Finalmente, LCCNet combina características de arquitecturas conocidas como ResNet y DenseNet, además, incluye mecanismos de atención y aplica técnicas para reducir el número de parámetros (Zhang et al., 2020).

Luego de revisar el estado del arte, la siguiente tabla muestra los resultados de los mejores modelos en las bases de datos más importantes y la métrica empleada.

Tabla 10

Resultados de los mejores modelos

| Base de datos | Modelo | Métrica | Valor de la métrica |
|----------------------|---------------|------------------|----------------------------|
| SWIMCAT | CloudNet | Average accuracy | 0.98 |
| CCSN | CloudNet | F-score | 0.88 |
| GCD | CGAT | Average accuracy | 0.84 |
| GRSCD | TGCN | Average accuracy | 0.89 |
| MGCD | RIWSRC | Average accuracy | 0.99 |

| | | | |
|-------|--------|---------|------|
| HBMCD | LCCNet | F-score | 0.96 |
|-------|--------|---------|------|

3.5.2. P2 ¿Qué bases de datos se han utilizado para la clasificación de nubes con imágenes tomadas desde tierra y en qué clases se etiquetaron las imágenes?

Principalmente, 6 bases de datos fueron usadas en los trabajos encontrados; se obtuvo acceso a estas para un análisis más profundo. Las características principales de los conjuntos de datos mencionados se muestran a continuación.

Tabla 11

Resumen de bases de datos relevantes

| Nombre | Número de imágenes | Número de clases | Usado por los artículos |
|---------|----------------------------------|------------------|---|
| SWIMCAT | 784 | 5 | (Liu et al., 2019; Shi et al., 2017; Wang et al., 2020; Zhang et al., 2018; Zhang et al., 2021) |
| CCSN | 2543 | 11 | (Ambildhuke & Banik, 2021; Zhang et al., 2018; Zhang et al., 2021) |
| GCD | 19000 | 7 | (Liu et al., 2021) |
| GRSCD | 8000 | 7 | (Liu et al., 2020) |
| MGCD | 8000 | 7 | (Li et al., 2020; Liu et al., 2020; Liu et al., 2019; Yu et al., 2021) |
| HBMCD | 38209 (después de la ampliación) | 11 | (Zhang et al., 2020) |

Las imágenes recolectadas se presentan en distintos formatos, resoluciones y clasificaciones. En primer lugar, SWIMCAT presenta imágenes de resolución 125 x 125 y las nubes están etiquetadas en 5 clases: despejado, patrón, oscuras densas, blancas densas y velo (Dev, Lee & Winkler, 2015). En segundo lugar, CCSN contiene imágenes de 256 x 256 píxeles etiquetadas en 11 clases: cúmulos, cumulonimbos, estratos, estratocúmulos, nimbostratos, cirros, cirrocúmulos, cirrostratos, altoestratos, altocúmulos y estelas de condensación. Se añade este último tipo de nube artificial, ya que haría una contribución al estudio de su impacto en el calentamiento global (Zhang et al., 2018). En tercer lugar, GCD tiene imágenes 512 x 512

divididas en 7 categorías: cúmulos; altocúmulos y cirrocúmulos; cirros y cirrostratos; cielo despejado; estratocúmulos, estratos y altoestratos; cumulonimbos y nimbostratos; y mixto. GCD es probablemente la colección de datos más grande de este tipo que existe y es casi 4 veces más grande que otros repositorios (Liu et al., 2021).

En cuarto lugar, GRSCD contiene imágenes tomadas con un lente de ojo de pez y una resolución de 1024 x 1024, estas se dividieron en 7 clases de la misma manera que en GCD teniendo en cuenta las definiciones de géneros de nubes establecidas por la Organización Meteorológica Mundial (Liu, Li, Zhang, Cao & Durrani, 2020). En quinto lugar, MGCD presenta 8000 imágenes 1024 x 1024 tomadas por un lente de ojo de pez, esta base de datos está dividida en 7 clases de igual manera que GCD y GRSCD, además contiene información adicional del tiempo: temperatura, humedad, precipitación y velocidad del viento (Liu, Li, Zhang, Xiao & Durrani, 2020). Finalmente, HBMCD es una base de datos formada por imágenes proveídas por la empresa Huayun Sounding (Beijing), originalmente, esta base de datos tenía 10942 ejemplares los cuales fueron etiquetados y ,posteriormente, ampliados hasta conseguir una cantidad total de 38209 imágenes con una resolución de 1358 x 1358 clasificadas en 11 clases: cúmulos, cumulonimbos, estratos, estratocúmulos, nimbostratos, cirros, cirrocúmulos, cirrostratos, altoestratos, altocúmulos y sol (sin nubes) (Zhang et al., 2020).

Como dato adicional, cabe mencionar que casi todas las bases de datos como GCD, GRSC, SWIMCAT y MGCD presentan imágenes exclusivamente del cielo o con una aparición mínima del paisaje alrededor. En cambio, CCSN contiene imágenes tomadas en múltiples direcciones donde en algunos casos se aprecia una gran parte del horizonte.

3.5.3. P3 ¿Qué técnicas de preprocesamiento de datos se han utilizado para la clasificación de nubes con imágenes tomadas desde tierra y de qué manera ayudaron a mejorar los resultados?

La gran mayoría de trabajos usan como principal técnica de preprocesamiento de datos la data *augmentation*. Esta técnica aumenta el conjunto de datos etiquetados para el entrenamiento y se ha probado que es efectiva para mitigar el sobreajuste; además, es un área ampliamente investigada y se cuenta con múltiples métodos para su aplicación (Liu et al., 2020). Por ejemplo, Zhang et al. (2018) aplican métodos como recorte aleatorio y volteo de las imágenes (verticalmente u horizontalmente), se menciona que el modelo entrenado con los datos aumentados presenta una excelente capacidad de generalización lo que evita el sobreajuste durante el entrenamiento y hace que el modelo sea más eficiente y robusto. De la misma manera, Li et al., (2019) recortan aleatoriamente las imágenes de entrada, y las voltean horizontalmente aleatoriamente.

En conclusión, mediante la revisión sistemática de la literatura se encontraron modelos de redes neuronales convolucionales que hacen uso de arquitecturas pre-entrenadas en ImageNet (transfer learning), donde la más común es ResNet-50 (Li et al., 2019, 2020; Liu et al., 2020; Liu et al., 2018, 2021; Liu, Duan, Zhang & Cao, 2019; Liu et al., 2020; Yu et al., 2021). Además, existen arquitecturas diseñadas para la clasificación de nubes con imágenes desde tierra que fueron entrenadas desde cero como CloudNet (Zhang et al., 2018) y LCCNet (Zhang et al., 2020). Asimismo, las redes neuronales convolucionales pueden usarse en conjunto con otros modelos de aprendizaje automático (máquinas de soporte vectorial) con el fin de mejorar su desempeño (Wang et al., 2020; Ye et al., 2017). Mediante el análisis de las características de las bases de datos encontradas, se concluye que GCD (Liu et al., 2021) y HBMCD (Zhang et al., 2020) son las más atractivas por su gran tamaño; sin embargo, estas últimas solo contienen imágenes enteramente del cielo tomadas con una cámara orientada 90

grados hacia arriba. En cambio, CCSN (Zhang et al., 2018) contiene imágenes tomadas desde distintos ángulos y que incluyen distintos paisajes, por lo que presenta una alta variabilidad. Finalmente, *data augmentation* es ampliamente utilizada para el preprocesamiento de datos para la solución de problemas similares por su efectividad en mitigar el sobreajuste (Liu et al., 2020).



Capítulo 4. Preparar una base de datos de imágenes de tipos de nubes que sea representativa, suficientemente heterogénea y grande para el entrenamiento adecuado de un modelo de clasificación automática de tipos de nubes

Contar con una base de datos adecuada es fundamental para el entrenamiento de un modelo de aprendizaje automático. En el enfoque supervisado, es esencial definir los requisitos de la base de datos basada en las características de los datos que el modelo utilizará para hacer inferencias en la etapa de producción. Para este caso, se requiere una base de datos que sea representativa, suficientemente heterogénea y de tamaño adecuado para entrenar un modelo de clasificación automática de tipos de nubes.

Por consiguiente, en este capítulo se explica en detalle el proceso llevado a cabo para preparar la base de datos, asegurando que cumple con los requisitos necesarios. En primer lugar, se describe el pipeline definido para la creación de la base de datos. Luego, se detalla la implementación de este pipeline para obtener una base de datos bien preparada y correctamente segmentada para los procesos de entrenamiento y prueba.

4.1. Resultado 1: *Pipeline* para la creación de la base de datos de imágenes de tipos de nubes

4.1.1. Criterios de aceptación de bases de datos.

Para cumplir con el objetivo de generar una base de datos representativa, es necesario contar con imágenes diversas que representen las condiciones esperadas en un ambiente productivo para una herramienta de clasificación automática de tipos de nubes. Por ello, las bases de datos tomadas en cuenta para este trabajo cumplen con los siguientes criterios:

- Aceptar las bases de datos con imágenes tomadas de distintos ángulos
- Aceptar las bases de datos con imágenes cuyo ángulo de elevación de cámara sea máximo de 60 grados

4.1.2. Criterios de aceptación y rechazo de imágenes.

Debido a que algunas imágenes de tipos de nubes obtenidas para el desarrollo de este trabajo fueron obtenidas de la web, fue necesario verificar que los datos recolectados cumplan con ciertos criterios de aceptación. Se definieron estos criterios con el fin de producir una base de datos de calidad.

- Aceptar las imágenes que estén etiquetadas con un solo tipo de nube
- Rechazar las imágenes que hayan sido tomadas desde satélites

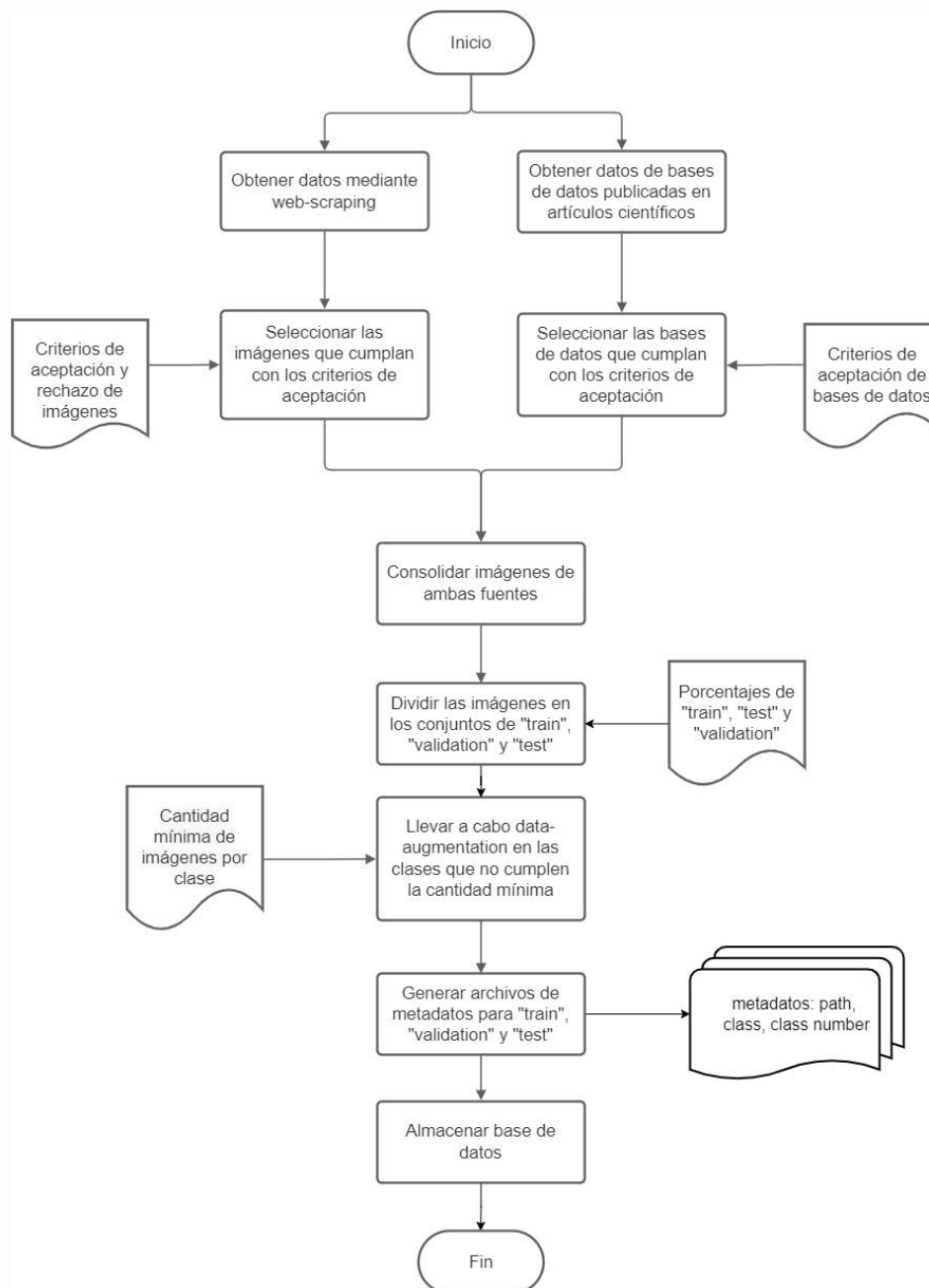
4.1.3. Descripción del pipeline para la creación de la base de datos de imágenes de tipos de nubes.

El diseño de este pipeline para la creación de la base de datos de imágenes de tipos de nubes se representa mediante un flujograma dibujado usando la herramienta Lucidchart (ver Figura 5). Como allí se señala, en primer lugar, es necesario obtener imágenes de tipos de nubes etiquetadas de 2 fuentes: la web mediante web scraping y bases de datos publicadas en artículos científicos. En segundo lugar, se seleccionarán las bases de datos publicadas en artículos científicos útiles para los fines de este trabajo y se filtrarán las imágenes obtenidas mediante web scraping de acuerdo a los criterios de aceptación o rechazo de bases de datos e imágenes respectivamente.

En tercer lugar, se consolidarán las imágenes de ambas fuentes para generar una base de datos representativa final. En cuarto lugar, con fines de reproducibilidad, se dividirá aleatoriamente la base de datos en 3 conjuntos: train, validation y test para el entrenamiento y prueba del modelo de red neuronal convolucional, donde cada conjunto consistirá en 70%, 10% y 20% del total de imágenes respectivamente. En quinto lugar, con el fin de mitigar el desbalance de imágenes entre clases, se llevará a cabo la técnica data augmentation en el conjunto de train, de manera que se obtengan más imágenes en las clases con poca

representación. Después, se generará un archivo de metadatos por cada conjunto obtenido: “train.csv”, “validation.csv” y “test.csv”. En estos se encontrarán 3 columnas de datos donde se detallará la ruta de la imagen, la clase a la cual pertenece y un número que represente la clase mencionada. Finalmente, se almacenarán las imágenes de tipos de nubes etiquetadas junto con los archivos de metadatos generados.

Figura 5. Pipeline para la creación de la base de datos de imágenes de tipos de nubes.



Este pipeline fue validado por un experto en análisis de datos observacionales en climatología, el acta de validación se encuentra en el Anexo D.

4.2. Resultado 2: Base de datos de imágenes de tipos de nubes preparada y separada adecuadamente para el proceso de entrenamiento y prueba

Este resultado consiste en la implementación del pipeline, el cual fue automatizado en su totalidad usando el lenguaje de programación Python.

Como primer paso, se seleccionaron las bases de datos encontradas en la literatura que cumplieran con los criterios de aceptación definidos. Durante la revisión sistemática de la literatura, se identificaron 6 bases de datos, a las cuales se obtuvo acceso para verificar su relevancia para la investigación. En el Anexo E se presenta la cantidad de ejemplares por clase de cada una de las bases de datos y el método utilizado para obtener dicha información.

Entre las bases de datos obtenidas, solo CCSN (Zhang et al., 2018) resultó ser de interés, ya que contiene imágenes de las 10 clases a analizar tomadas en distintos lugares, resoluciones y ángulos de elevación de cámara; además, presenta imágenes en las cuales se puede apreciar parte del horizonte. CCSN se encuentra en el archivo “CCSN_V2.zip”² y en la siguiente figura se pueden observar algunos ejemplares.



Figura 6. Ejemplares de fotografías de nubes de CCSN.

² <https://drive.google.com/file/d/1dzu0S4xNpIejFx3Tc-SWTBNFE94hZB9P/view?usp=sharing>

El segundo paso consistió en obtener imágenes de tipos de nubes de la web mediante web scraping. Para lograr lo mencionado, se programó en Python el script “index.py”³, el cual simula la interacción de un usuario en la galería de imágenes de Cloud Appreciation Society, en adelante CAS (Cloud Appreciation Society, 2022). Mediante el script se logró automatizar el proceso de descarga de imágenes separadas por clase; estas se encuentran en el archivo “Cloudappreciationsociety.zip”⁴. Además, se obtuvo el archivo de metadatos “output_data.json”⁵ que contiene la ruta local de la imagen, el título de la imagen, la clase a la que pertenece, un número identificador de la clase y una lista con las clases adicionales que aparecen en la imagen. En la siguiente figura, se observa un ejemplo del contenido del archivo generado.

```
{
  "path": "./images/3_Stratocumulus/3_Stratocumulus_2639.jpg",
  "image_title": "Clouds at Sunset, from Guidonia (Rome \u2013 Italy)",
  "class": "Stratocumulus",
  "id_class": 3,
  "additional": [
    1,
    4
  ]
},
```

Figura 7. Ejemplo de archivo de salida de *web scraping*.

Posteriormente, para cumplir con los criterios de aceptación y rechazo definidos, se filtraron las imágenes que contenían más de una clase; es decir, se generó un archivo adicional “filtered_data.json”⁶ que contiene solamente los datos de las imágenes que no poseen clases adicionales en el campo *additional*; además, se eliminaron manualmente las imágenes que

³ <https://github.com/TheBlueStuff/cloud-classification/blob/main/web-scraping/CloudAppreciationSociety/index.py>

⁴ <https://drive.google.com/file/d/1Tud26KiN2sN43mfE0KPvXBOe90Fmzs-k/view?usp=sharing>

⁵ https://github.com/TheBlueStuff/cloud-classification/blob/main/web-scraping/CloudAppreciationSociety/output_data.json

⁶ https://github.com/TheBlueStuff/cloud-classification/blob/main/web-scraping/CloudAppreciationSociety/filtered_data.json

fueron tomadas desde satélites. En el Anexo F, se encuentra la cantidad de imágenes por clase obtenidas después de los filtros.

Como tercer paso, se consolidaron las imágenes de CAS y CCSN. Para lograr esto, se tomaron como base las imágenes de CAS, ya que estas poseen más variabilidad y mejor resolución. Así, se consolidaron solo las clases de CAS que poseían menos de 200 ejemplares (Cirrocúmulos, Altoestratos y Nimbostratos). Después, se dividió la base de datos aleatoriamente en los grupos de train, validation y test tomando el 70% 10% y 20% de cada clase para cada conjunto respectivamente. Posteriormente, para mitigar el desbalance, se llevó a cabo data augmentation en las clases con menos representación en el conjunto train (por debajo de 400 ejemplares), estas clases fueron cirrocúmulos, estratos, altoestratos, nimbostratos y cirrostratos. Las técnicas de data augmentation implementadas son las siguientes:

- RandomHorizontalFlip: Invertir aleatoriamente algunas imágenes de manera horizontal.
- RandomCrop: Cortar aleatoriamente un porcentaje de algunas imágenes. Este porcentaje está dentro de un intervalo definido previamente.

De esta manera, se logró incrementar al doble la cantidad de imágenes de cada clase subrepresentada. Luego, se generaron los archivos de metadatos por cada conjunto: “train.csv”⁷, “validation.csv”⁸ y “test.csv”⁹ que contienen la ruta de la imagen, el nombre de la clase a la que pertenece y un número que representa la clase.

⁷ <https://github.com/TheBlueStuff/cloud-classification/blob/main/web-scraping/CloudAppreciationSociety/train.csv>

⁸ <https://github.com/TheBlueStuff/cloud-classification/blob/main/web-scraping/CloudAppreciationSociety/validation.csv>

⁹ <https://github.com/TheBlueStuff/cloud-classification/blob/main/web-scraping/CloudAppreciationSociety/test.csv>

Finalmente, se almacenaron las imágenes separadas por conjunto y por clase. La base de datos final se encuentra en el archivo “data_base.zip”¹⁰. Este proceso, a partir de la división de las imágenes en los 3 conjuntos necesarios, fue automatizado mediante el desarrollo del script “split_and_augment.py”¹¹. A continuación, se pueden apreciar ejemplares de la base de datos final.

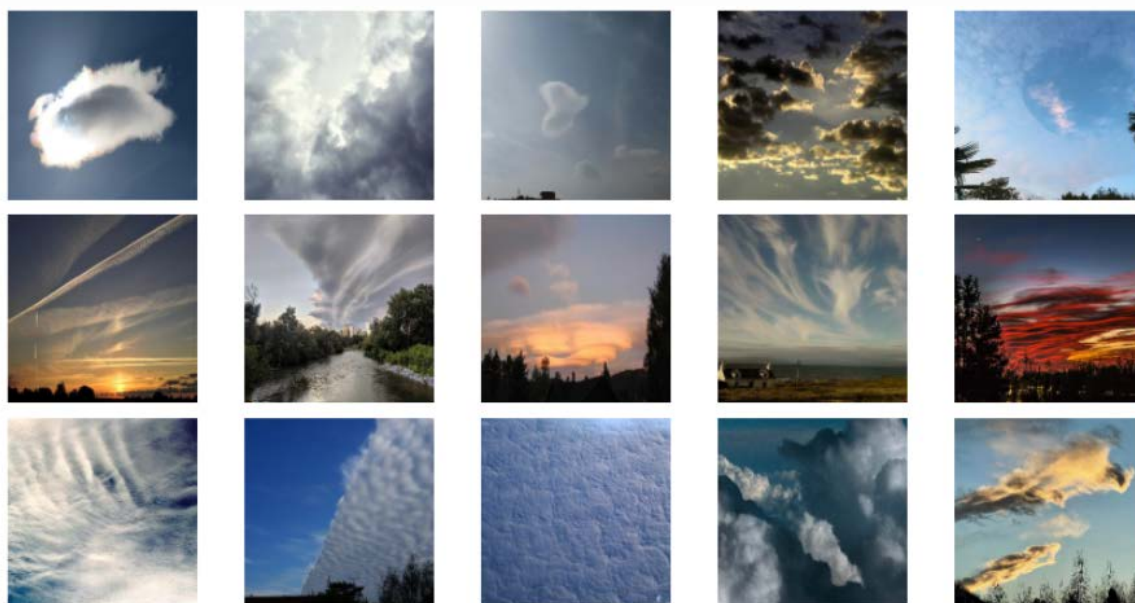


Figura 8. Ejemplos de fotografías de la base de datos final.

Asimismo, en la Tabla 12 se encuentra la cantidad de imágenes por clase y por conjunto.

Tabla 12

Cantidad de ejemplares por clase y por conjunto de la base de datos final

| Clase | Train | Validation | Test | Total |
|--------------|-------|------------|------|-------|
| Altocúmulos | 2797 | 400 | 800 | 3997 |
| Altoestratos | 420 | 30 | 60 | 510 |
| Cumulonimbos | 1043 | 149 | 298 | 1490 |
| Cirrocúmulos | 552 | 40 | 79 | 671 |
| Cirros | 750 | 108 | 215 | 1073 |

¹⁰ https://drive.google.com/file/d/1x8156a3a3QGR4_O67rXB3_C38vXrxxcn/view?usp=sharing

¹¹ https://github.com/TheBlueStuff/cloud-classification/blob/main/web-scraping/CloudAppreciationSociety/split_and_augment.py

| | | | | |
|----------------|------|-----|-----|------|
| Cirrostratos | 328 | 24 | 48 | 400 |
| Cúmulos | 1378 | 197 | 394 | 1969 |
| Nimbostratos | 396 | 29 | 57 | 482 |
| Estratocúmulos | 582 | 84 | 167 | 833 |
| Estratos | 502 | 36 | 72 | 610 |

4.3. Discusión sobre el desarrollo del primer objetivo

En este capítulo se definió un proceso para la elaboración de una base de datos de imágenes de tipos de nubes y este fue automatizado mediante scripts en Python. La base de datos final contiene 10 clases y está dividida en tres conjuntos excluyentes, train, validation y test, para el adecuado proceso de entrenamiento y prueba de un modelo de clasificación automática de tipos de nubes. El primer conjunto se usará para el ajuste de parámetros de la red neuronal, el segundo, para la optimización de hiperparámetros y, finalmente, el tercero, para la prueba final y la obtención de métricas que evalúen el desempeño del modelo.

Es importante mencionar que la técnica de preprocesamiento data augmentation se realizó después de la división de las imágenes en los 3 conjuntos mencionados y solo se llevó a cabo en el grupo de train. Esto fue necesario para evitar el problema conocido como *data leakage* que es originado, entre otras causas, por realizar el preprocesamiento de los datos en los conjuntos de train y test de manera colectiva, lo cual genera errores al momento de evaluar el modelo (Kapoor & Narayanan, 2022). Además, la división de los datos se realizó de manera estratificada debido al desbalance de clases. Esto ayuda a evaluar el desempeño de la red neuronal de forma más objetiva, ya que, en los 3 conjuntos, la distribución de clases es la misma.

La gran mayoría de trabajos científicos que atacan el problema de clasificación automática de tipos de nubes utilizan imágenes tomadas con ángulos de cámara de 90 grados hacia arriba y en las cuales solo se puede observar el cielo; sin embargo, el objetivo de este

trabajo es generar una base de datos representativa, por lo que se usaron fuentes de datos que proveen imágenes variadas; es decir, con diferentes ángulos de orientación de cámara, tomadas en distintos lugares y que incluyen parte del horizonte.



Capítulo 5. Implementar un modelo basado en una red neuronal convolucional que permita clasificar tipos de nubes automáticamente

En este capítulo, se describe detalladamente el proceso de implementación, entrenamiento y prueba de las arquitecturas de redes neuronales convolucionales más relevantes para la clasificación automática de tipos de nubes.

En primer lugar, se eligieron 3 arquitecturas, las cuales fueron implementadas en Python para luego ser entrenadas sobre la base de datos creada en el capítulo 4. Posteriormente se realizó una optimización de hiperparámetros en cada una de las arquitecturas implementadas. De esta manera, se obtuvo la mejor versión de cada una y se realizó una evaluación comparativa para determinar cuál es la mejor entre las 3 desarrolladas.

Finalmente, se desplegó el modelo para ser usado mediante una interfaz web y de esta manera mostrar su funcionamiento.

5.1. Resultado 3: Revisión sistemática de arquitecturas de redes neuronales convolucionales que han sido usadas para la clasificación automática de tipos de nubes

En el Capítulo 3, se realizó una revisión sistemática de la literatura para conocer las arquitecturas de redes neuronales convolucionales más relevantes que se han usado para la clasificación automática de tipos de nubes. Se concluyó que la mayoría de trabajos usan arquitecturas pre-entrenadas y la más usada es ResNet-50. Esta arquitectura reformula las capas como funciones residuales que hacen referencia a las entradas de las capas previas, estas redes han demostrado ser más fáciles de optimizar y pueden ganar efectividad con profundidades mayores (He, Zhang, Ren & Sun, 2016). Por consiguiente, ResNet-50 es la primera arquitectura elegida para implementar.

La segunda arquitectura a desarrollar es CloudNet (Zhang et al., 2018), una arquitectura propuesta específicamente para el problema en cuestión, se eligió esta arquitectura por los

buenos resultados que presenta y, además, fue puesta a prueba sobre CCSN (Zhang et al., 2018), una base de datos con características similares a la preparada en este trabajo.

Finalmente, la tercera arquitectura a implementar es Inception v3 (Szegedy, Vanhoucke, Ioffe, Shlens & Wojna, 2015). Esta arquitectura es una de las pre-entrenadas usadas para atacar este problema, se eligió porque es barata computacionalmente y usa pocos parámetros en comparación con modelos afines. Además, presentó buenos resultados en los artículos científicos en los cuales fue usada. Esta revisión sistemática fue validada por un experto en aprendizaje de máquina, el acta de validación se encuentra en el Anexo G.

5.2. Resultado 4: Implementación de las arquitecturas de redes neuronales convolucionales más efectivas y relevantes

La implementación de las arquitecturas de redes neuronales convolucionales elegidas se realizó en el lenguaje de programación Python usando el framework PyTorch.

En primer lugar, se implementó la arquitectura CloudNet, esta, al no ser usada extensivamente en el campo de visión computacional, tuvo que ser implementada desde cero, capa por capa. La implementación de CloudNet en PyTorch se encuentra en el script “cloudnet.py”¹².

En segundo lugar, las arquitecturas ResNet-50 e Inception v3 son modelos utilizados ampliamente en el campo de visión computacional, por este motivo cuentan con implementaciones optimizadas en distintos frameworks de aprendizaje profundo. En PyTorch, el uso de estas arquitecturas es bastante directo, solo es necesario importar la clase que contiene el modelo y cambiar la última capa por una capa con la cantidad de neuronas como clases que se desean predecir.

¹² <https://github.com/TheBlueStuff/cloud-classification/blob/main/models/cloudnet.py>

Finalmente, se implementó un flujo de entrenamiento para cada arquitectura que se encuentra en los siguientes archivos: “CloudNet.ipynb”¹³, “Resnet-50.ipynb”¹⁴ e “Inception-v3.ipynb”¹⁵.

5.2.1. Flujo de entrenamiento de redes neuronales convolucionales

En primer lugar, fue necesario realizar un preprocesamiento de las imágenes, de manera que sean del tamaño adecuado para ser procesadas por la red neuronal convolucional. Además, se aplicaron técnicas de aumentación de datos en el conjunto de train de manera distinta en cada iteración de entrenamiento. De esta manera, la red neuronal procesó una imagen diferente en cada época con el fin de mitigar el sobreajuste y hacer al modelo más robusto, tal y como se menciona en la sección 3.5.3. En consecuencia, fue necesario aplicar transformaciones a las imágenes, algunas que se aplicaron siempre como el redimensionamiento de las imágenes y otras que se aplicaron de manera distinta en cada época como la aumentación de datos. Se usó la librería Albumentations en Python para gestionar las transformaciones, para el conjunto de train se usaron todas las transformaciones; en cambio, para el conjunto de validation y test se usó solo la transformación de redimensionamiento. A continuación, se describen las transformaciones aplicadas.

- **Resize:** Redimensionamiento de la imagen al largo y ancho especificados (Buslaev et al., 2020).
- **ShiftScaleRotate:** Aplicar aleatoriamente transformaciones afines: trasladar, escalar y rotar la entrada (Buslaev et al., 2020).
- **RandomCrop:** Recortar una parte aleatoria de la imagen (Buslaev et al., 2020).

¹³ <https://github.com/TheBlueStuff/cloud-classification/blob/main/notebooks/CloudNet.ipynb>

¹⁴ <https://github.com/TheBlueStuff/cloud-classification/blob/main/notebooks/ResNet-50.ipynb>

¹⁵ <https://github.com/TheBlueStuff/cloud-classification/blob/main/notebooks/Inception-v3.ipynb>

- RGBShift: Desplazar aleatoriamente los valores de cada canal de la imagen de entrada: rojo, verde y azul (Buslaev et al., 2020).
- RandomBrightnessContrast: Cambia aleatoriamente el brillo y contraste de la imagen de entrada (Buslaev et al., 2020).

En segundo lugar, se entrenó los pesos de la red neuronal haciendo experimentos a modo de prueba con distintas configuraciones y monitoreando los resultados con la herramienta Weights & Biases donde se puede visualizar un panel con gráficos que muestran indicadores relacionados con el aprendizaje del modelo tanto en el conjunto de train como en el conjunto de validation. Se calculó el accuracy promedio durante el entrenamiento (ver Figura 9), la función de pérdida en el conjunto de train (ver Figura 10) y la función de pérdida en el conjunto de validation (ver Figura 11). El experimento mostrado en este caso toma los siguientes hiperparámetros: batch size 32, learning rate 0.0001, optimizador SGD y arquitectura ResNet-50. Además, se empleó la técnica early stopping para mitigar el sobreajuste; en consecuencia, el modelo terminó de entrenar en la época 26.

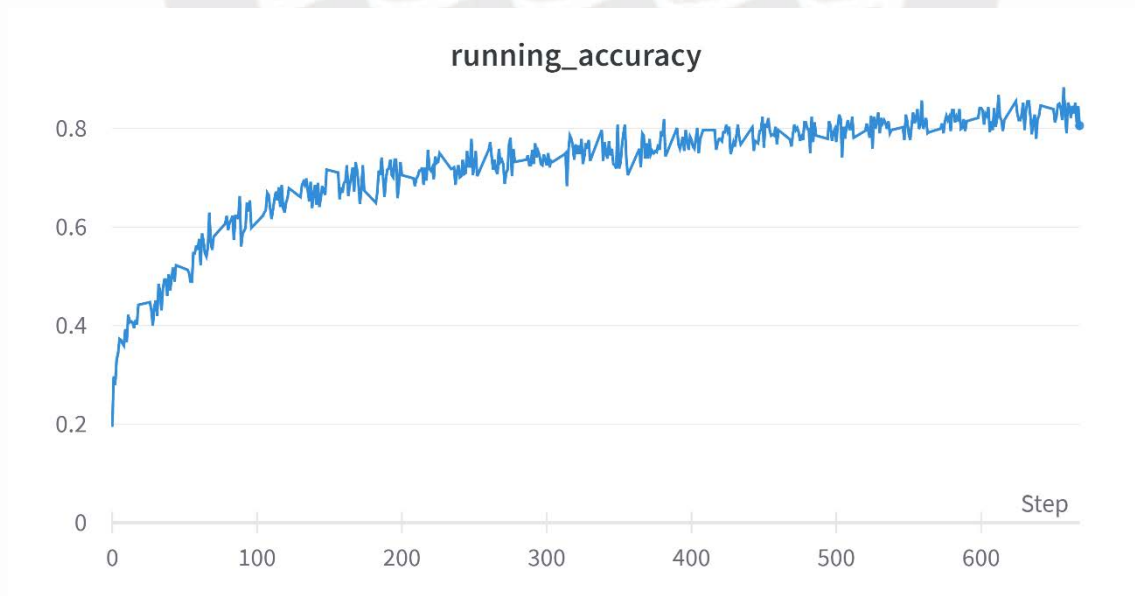


Figura 9. Gráfico de accuracy durante en el entrenamiento.

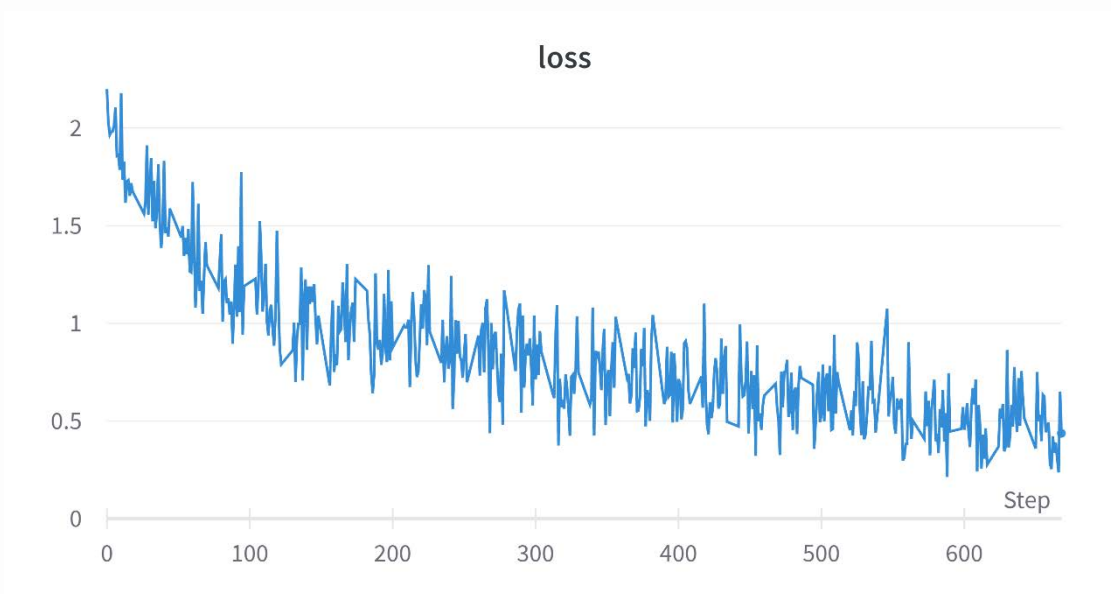


Figura 10. Gráfico de la función de pérdida en el conjunto de *train*.

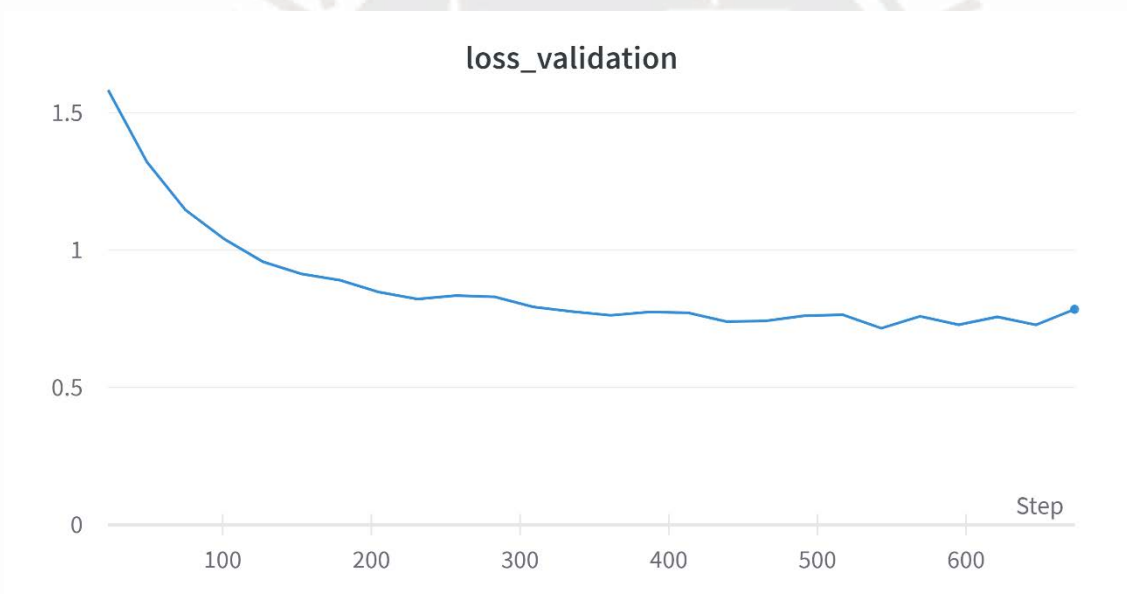


Figura 11. Gráfico de la función de pérdida en el conjunto de *validation*.

Finalmente, para evaluar el desempeño del modelo se hace uso del conjunto de test para simular un ambiente productivo, las métricas usadas son la matriz de confusión (ver Figura 12), el F-score micro (0.77) y el F-score macro (0.65).

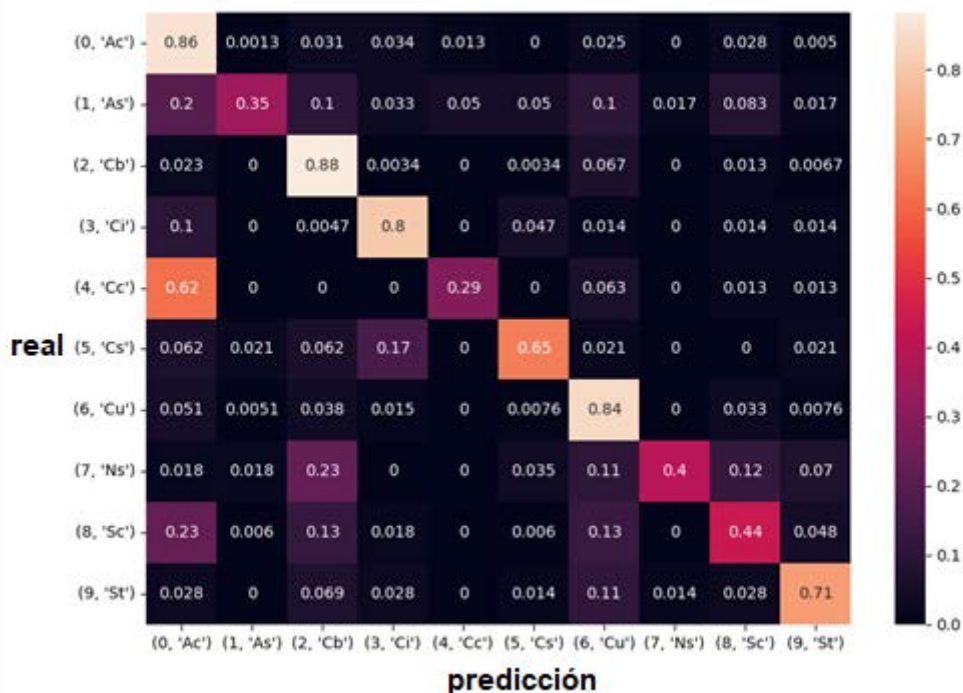


Figura 12. Matriz de confusión en el conjunto de *test*.

5.3. Resultado 5: Evaluación comparativa del desempeño de distintas arquitecturas de redes neuronales convolucionales en la base de datos

En los resultados anteriores, se generó una base de datos de imágenes de tipos de nubes y se implementaron 3 arquitecturas de redes neuronales convolucionales: CloudNet, Resnet-50 e Inception v3. El objetivo de este resultado es ejecutar el flujo de entrenamiento definido en el capítulo 5.2.1 con distintas configuraciones en cada una de las arquitecturas implementadas sobre la base de datos generada. De esta manera, se podrá obtener la mejor versión de cada modelo en el conjunto de validation y se elegirá la arquitectura que mejor desempeño obtuvo.

Se utilizó la herramienta *sweep* de Weights & Biases para ejecutar una búsqueda de cuadrícula; es decir, entrenar el modelo con todas las combinaciones posibles en un conjunto de configuraciones de hiperparámetros. En la siguiente tabla se detallan los hiperparámetros que se afinaron y los valores que tomaron.

Tabla 13

Hiperparámetros que se afinaron y los valores que tomaron

| | Optimizador | Batch size | Learning rate |
|----------------|--------------------|---------------------|----------------------------------|
| Valores | - Adam - SGD | - 8 - 16 - 32 | - 0.0001 - 0.0003 - 0.0005 |

En consecuencia, se realizaron 18 experimentos por cada arquitectura para analizar qué hiperparámetros afectan de mayor manera a la métrica final: F-score macro en el conjunto de validation. El código fuente para realizar de manera automática los experimentos mencionados se encuentra en el repositorio de GitHub del proyecto, específicamente en el archivo “train.py”¹⁶. La herramienta sweep de Weights & Biases permite ejecutar experimentos mediante el uso de agentes, es posible iniciar varios agentes en paralelo con combinaciones de hiperparámetros que aún no han sido usadas. En este caso, se usaron hasta 4 agentes en paralelo, ya que los servidores del grupo IA-PUCP cuentan con 4 tarjetas gráficas y es posible ejecutar un experimento en cada una de ellas. A continuación, se describen los experimentos realizados en cada una de las arquitecturas.

- CloudNet (Zhang et al., 2018)

CloudNet fue la primera arquitectura con la cual se realizaron experimentos. Esta arquitectura fue entrenada desde cero y los resultados obtenidos se pueden observar en el siguiente gráfico de coordenadas paralelas elaborado en la Figura 13.

¹⁶ <https://github.com/TheBlueStuff/cloud-classification/blob/main/training/train.py>

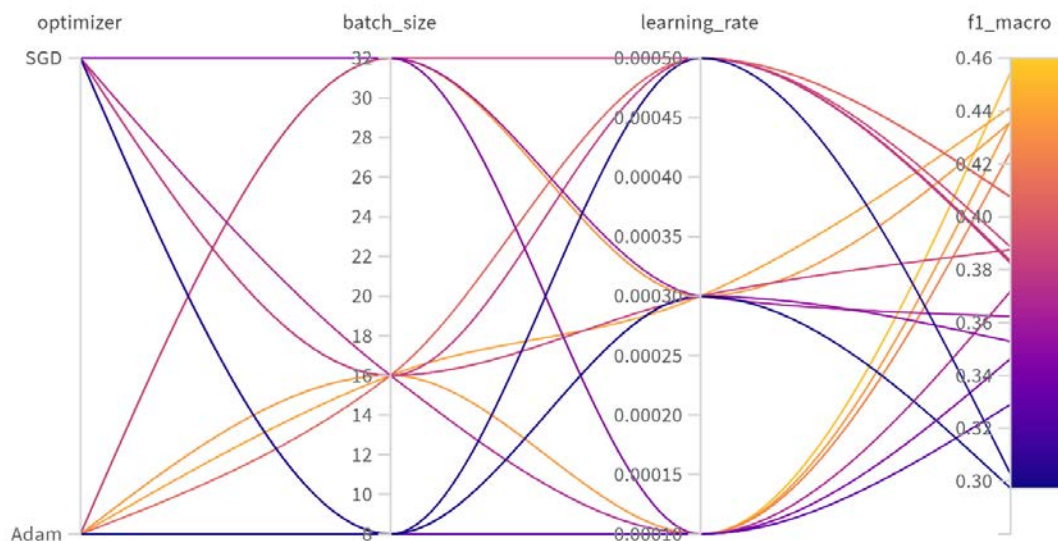


Figura 13. Gráfico de coordenadas paralelas de los experimentos realizados en CloudNet.

En este caso, el mejor resultado (0.45) se obtuvo con la siguiente configuración: optimizador Adam, batch size 32 y learning rate 0.0001. En la tabla 5.2, se observan los indicadores más relevantes en promedio de todos los experimentos. Finalmente, el reporte completo de los experimentos realizados en CloudNet se encuentra disponible en Google Drive y está disponible en el archivo “wandb_cloudnet.csv”¹⁷.

Tabla 14

Indicadores promedio de 18 experimentos en CloudNet

| Arquitectura | Tiempo (minutos) | F-score macro | Épocas | Loss |
|--------------|------------------|---------------|--------|------|
| CloudNet | 83.53 | 0.38 | 55.22 | 1.45 |

- ResNet-50 (He et al., 2016)

En segundo lugar, se usó ResNet-50 preentrenada, los resultados obtenidos se pueden observar en un gráfico de coordenadas paralelas en la Figura 14.

¹⁷ https://drive.google.com/file/d/1Ar9UJmkGrnmFqblFYh_AqP8CU3LXfpWN/view?usp=share_link

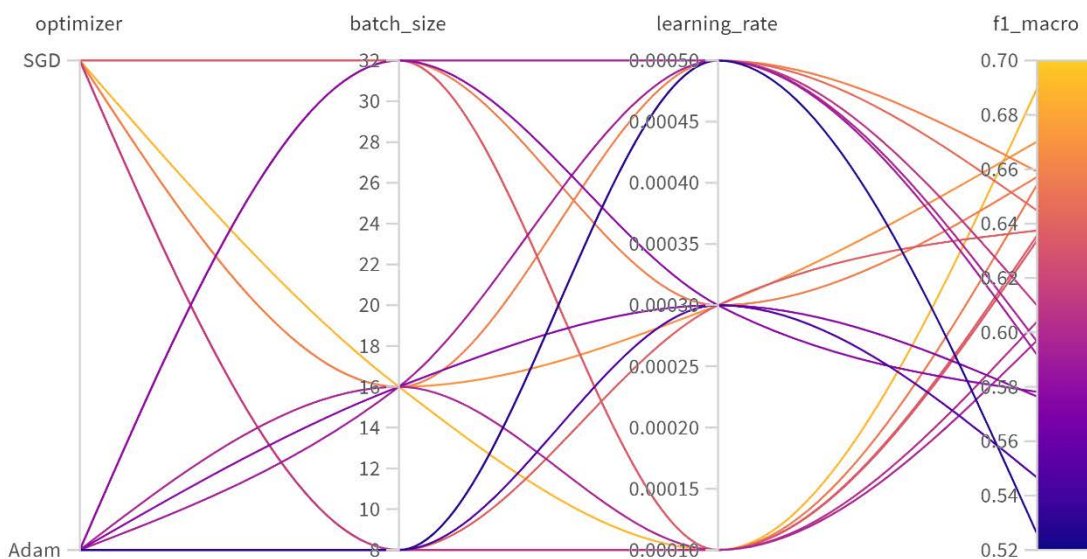


Figura 14. Gráfico de coordenadas paralelas de los experimentos realizados en ResNet-50.

En este caso, el mejor resultado (0.68) se obtuvo con la siguiente configuración: optimizador SGD, batch size 16 y learning rate 0.0001. En la tabla 5.3, se observan los indicadores más relevantes en promedio de todos los experimentos. Finalmente, el reporte completo de los experimentos realizados en ResNet-50 se encuentra disponible en Google Drive y está disponible en el archivo “wandb_resnet_50.csv”¹⁸

Tabla 15

Indicadores promedio de 18 experimentos en ResNet-50

| Arquitectura | Tiempo (minutos) | F-score macro | Épocas | Loss |
|--------------|------------------|---------------|--------|------|
| ResNet-50 | 42.85 | 0.63 | 15.5 | 0.5 |

- Inception v3 (Szegedy et al., 2015)

En tercer lugar, se usó Inception v3 preentrenada, los resultados obtenidos se pueden observar en un gráfico de coordenadas paralelas en la Figura 15.

¹⁸ https://drive.google.com/file/d/1aeyjn1Yf8No_lcqzjQSvyC5ap9vZW8-/view?usp=share_link

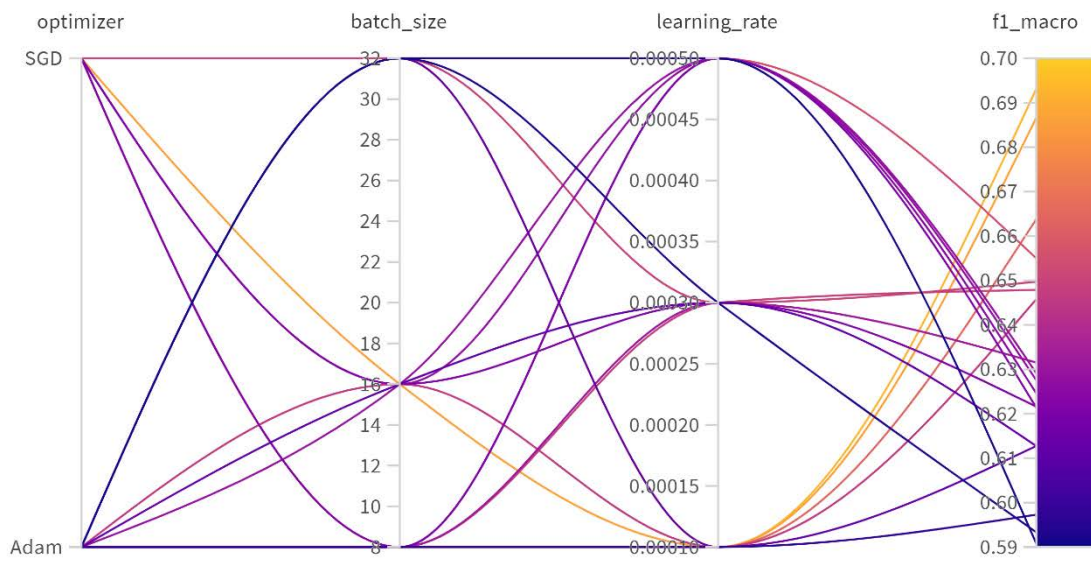


Figura 15. Gráfico de coordenadas paralelas de los experimentos realizados en Inception v3.

En este caso, el mejor resultado (0.69) se obtuvo con la siguiente configuración: optimizador SGD, batch size 32 y learning rate 0.0001. En la tabla 5.4, se observan los indicadores más relevantes en promedio de todos los experimentos. Finalmente, el reporte completo de los experimentos realizados en Inception v3 se encuentra disponible en Google Drive y está disponible en el archivo “wandb_inception_v3.csv”¹⁹.

Tabla 16

Indicadores promedio de 18 experimentos en Inception v3

| Arquitectura | Tiempo (minutos) | F-score macro | Épocas | Loss |
|--------------|------------------|---------------|--------|------|
| Inception v3 | 48.16 | 0.66 | 16.83 | 0.40 |

¹⁹ https://drive.google.com/file/d/1pldGNLjfnkxUPjSCZ_RCtp3XZ09hKE6f/view?usp=share_link

En la Figura 16, se compara la métrica F-score macro sobre el conjunto validation a lo largo del entrenamiento de cada una de las arquitecturas en su mejor versión encontrada, se puede observar que el modelo Inception v3 supera a ResNet-50 y este a su vez, a CloudNet. En el caso de CloudNet, se observa, al ser entrenada desde cero, necesita más épocas para converger. En cambio, ResNet-50 e Inception v3 convergen más rápido, en este particular caso, ResNet-50 empezó a mejorar de manera más pronta que Inception v3; sin embargo, se detuvo automáticamente debido a que se estaba aproximando al sobreajuste. En consecuencia, fue superada por Inception v3 al final.

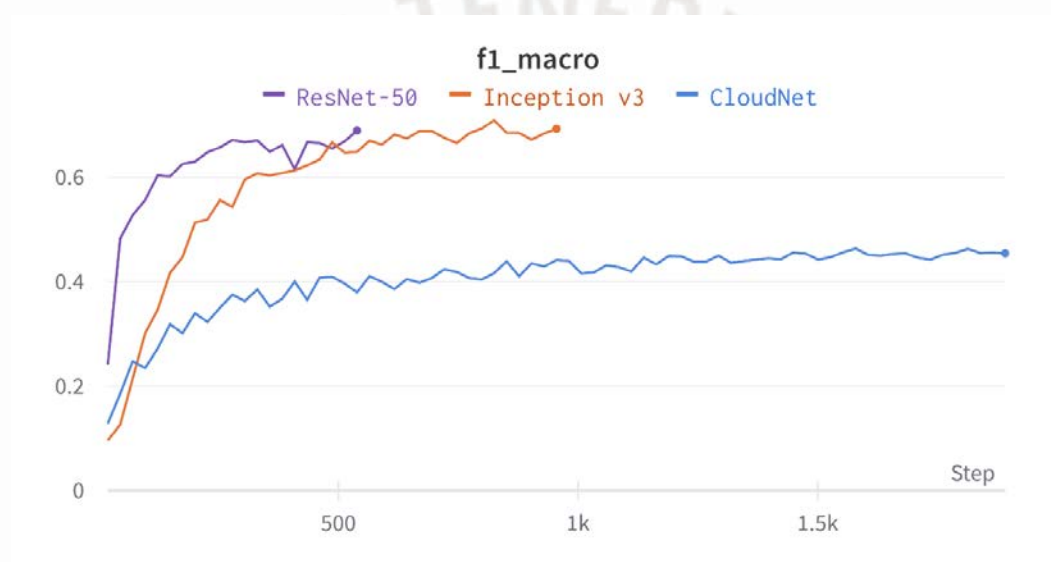


Figura 16. F-score macro a lo largo del entrenamiento de las 3 arquitecturas.

Finalmente, se evaluaron las tres mejores versiones obtenidas en el conjunto de test. En la siguiente tabla se detallan los resultados.

Tabla 17

Resultados de la mejor versión de cada arquitectura

| Arquitectura | F-score macro |
|--------------|---------------|
| CloudNet | 0.46 |
| ResNet-50 | 0.68 |
| Inception v3 | 0.70 |

Se eligió la mejor versión encontrada de Inception v3 como el modelo final, debido a que obtuvo mejores métricas en el conjunto de test y fue mejor durante en el entrenamiento, ya que, en promedio, obtuvo mejor F-score macro en el conjunto de validation. Además, su tiempo de entrenamiento promedio fue similar al de ResNet-50 y fue 2 veces más rápido que CloudNet. La matriz de confusión obtenida por dicho modelo en el conjunto de test se muestra en la figura a continuación:

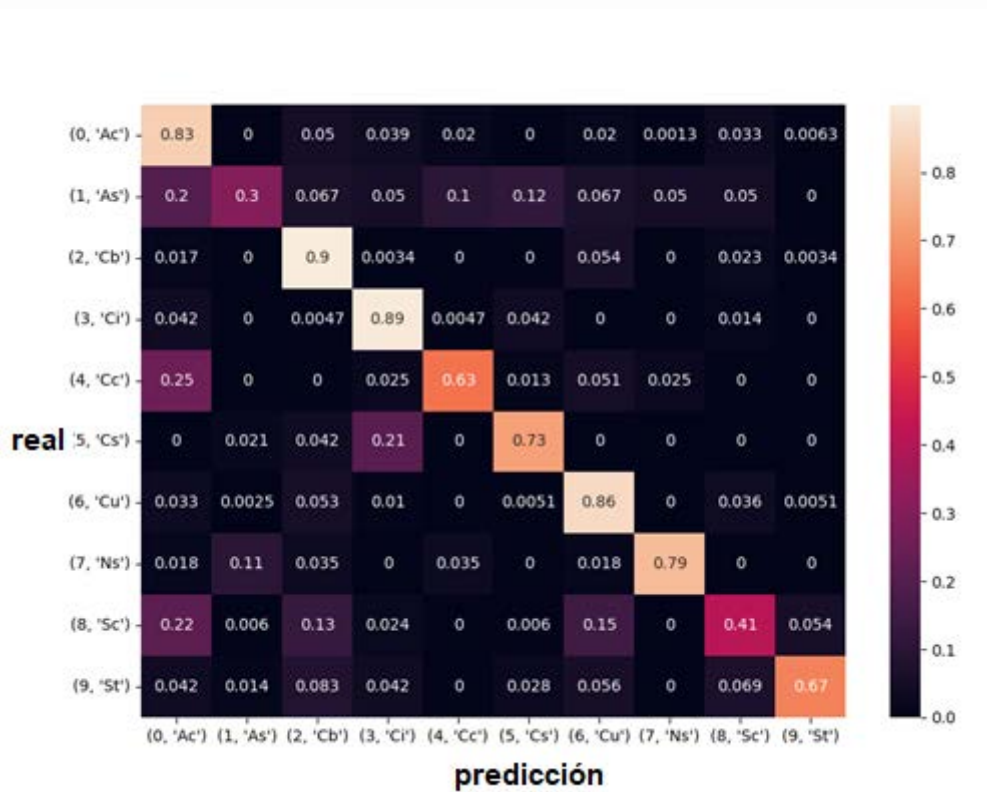


Figura 17. Matriz de confusión en el conjunto de *test* de la mejor configuración de Inception v3.

Métricas e indicadores adicionales se pueden observar en un reporte elaborado con la herramienta Weights & Biases²⁰.

²⁰ <https://wandb.ai/mario-andonaire/cloud-classification/reports/Reporte-de-evaluaci-n-comparativa-de-redes-neuronales-convolucionales--VmlldzoyOTE0MTgw?accessToken=38tnnjh77ylnhns20tq9hop2qaa1eu9674ntx9xevogj5nfict79493xudpfa33y>

La evaluación comparativa de arquitecturas de redes neuronales convolucionales descrita en el presente capítulo fue validada por un experto en aprendizaje de máquina. El acta de validación se encuentra en el Anexo H.

5.4. Resultado 6: Desarrollo de una interfaz web que muestre el funcionamiento del modelo de red neuronal convolucional implementado

Después de implementar y optimizar los modelos de redes neuronales convolucionales propuestos en este trabajo, se eligió el que obtuvo mejores resultados y desempeño. Para mostrar su funcionamiento, es necesario desarrollar una interfaz web, la cual se definió mediante el planteamiento de requerimientos funcionales y no funcionales. Además, se estableció la arquitectura que se usará para el despliegue de esta interfaz en AWS.

5.4.1. Lista de requerimientos funcionales de la interfaz web.

- La interfaz debe permitir al usuario cargar una o varias imágenes de nubes en formato JPG, JPEG o PNG.
- La interfaz debe procesar la imagen o imágenes mediante una API para determinar el tipo de nube presente.
- En caso de subir una sola imagen, la interfaz debe mostrarla e indicar el tipo de nube inferido junto con una corta descripción.
- En caso de subir múltiples imágenes, la interfaz debe permitir descargar un archivo .csv con los resultados inferidos.
- La interfaz debe mostrar una barra de progreso que indique el estatus de la predicción del tipo de nube.
- La interfaz debe mostrar la probabilidad de la predicción realizada.

5.4.2. Lista de requerimientos no funcionales de la interfaz web.

- La interfaz web debe ser responsive.

- La interfaz web debe ser accesible desde cualquier navegador.
- La interfaz web debe ser de acceso público.
- La interfaz web debe ser desarrollada usando Vue.js.
- La interfaz web deberá usar una API programada en Python usando la librería FastAPI para la obtención de la predicción del tipo de nube.
- La interfaz web será desplegada en AWS S3 y la API en AWS EC2.

5.4.3. Arquitectura de despliegue de la interfaz web en AWS.

La interfaz web realiza la inferencia del tipo de nube mediante el uso de una API que integra el modelo de red neuronal convolucional desarrollado. De esta manera, se implementaron 2 módulos: el módulo *frontend* que corresponde a la interfaz web misma y el módulo *backend* que implementa la API necesaria para inferir el tipo de nube presente en las imágenes proveídas por el usuario. El frontend fue desarrollado en Javascript con el framework Vue.js y fue desplegado en un contenedor AWS S3; asimismo, el backend fue desarrollado en Python usando la librería FastAPI y desplegado en una instancia de cómputo AWS EC2.

En la siguiente figura se puede observar un diagrama de la arquitectura definida para el despliegue de la interfaz web haciendo uso de AWS.

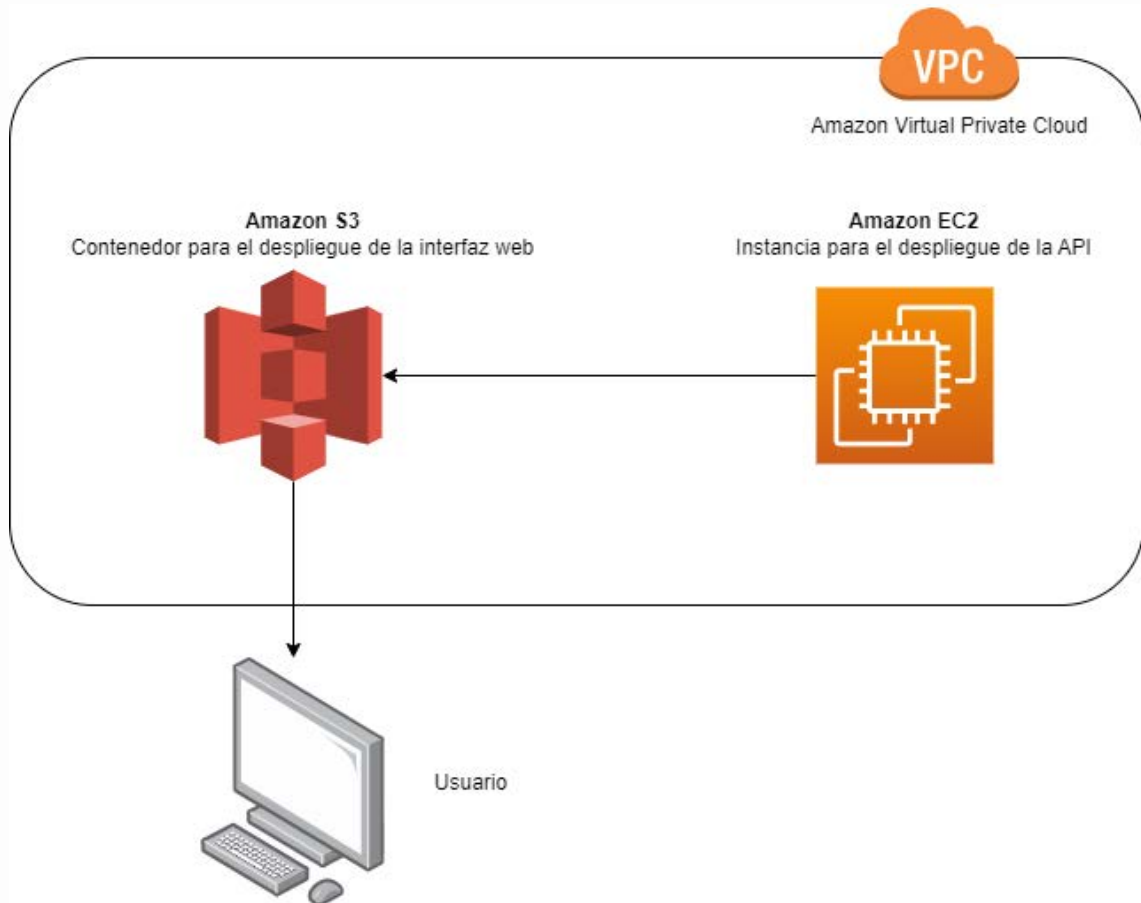


Figura 18. Gráfico de la arquitectura definida para el despliegue de la interfaz web.

5.4.4. Desarrollo del módulo backend.

La implementación del módulo backend tiene como objetivo poner a disposición el modelo de red neuronal convolucional entrenado para realizar predicciones (“model.pth”²¹). Esto se realizó en Python con la librería FastAPI mediante el despliegue de una interfaz que proporciona 2 servicios: el primer servicio tiene como objetivo realizar la inferencia del tipo de nube para una sola imagen, por lo cual recibe una sola imagen y retorna la clase a la cual pertenece; por otro lado, el segundo servicio realiza la predicción para varias imágenes, por lo que retorna un archivo csv que compuesto por los nombres de las imágenes y los tipos de nubes

²¹ <https://github.com/TheBlueStuff/cloud-classification/blob/main/Cloud-API/pretrained/model.pth>

que contienen. Estos servicios fueron desplegados en una instancia de cómputo de AWS EC2 y la documentación de estos servicios se puede encontrar en el Anexo I. Además, el código fuente de la API y las instrucciones para su configuración se encuentran en el repositorio de GitHub del proyecto en la carpeta “Cloud-API”²².

5.4.5. Desarrollo del módulo frontend.

El módulo frontend tiene como objetivo proveer una interfaz en donde el usuario pueda subir una o varias imágenes de nubes para poder recibir la predicción del tipo de nube presente. Esta interfaz fue desarrollada con el lenguaje Javascript y el framework Vue.js, el código fuente y las instrucciones para su configuración se encuentran en el repositorio de GitHub del proyecto en la carpeta “CloudClassificationApp”²³.

La interfaz web tiene 2 modos de uso diferentes. En la Figura 19, se observa el primer modo de uso; en este caso, el usuario cargará una imagen de una nube y recibirá como resultado el tipo de nube inferido, una breve descripción de la clase y la probabilidad con la cual el modelo hizo la predicción. Asimismo, en la Figura 20, se observa el segundo modo de uso; en este caso, el usuario cargará múltiples imágenes de nubes y recibirá como resultado un archivo csv (ver Figura 21) que contiene los nombres de los archivos cargados junto con la clase a la que pertenecen. El módulo frontend fue desplegado en un *bucket* de acceso público de Amazon S3²⁴.

²² <https://github.com/TheBlueStuff/cloud-classification/tree/main/Cloud-API>

²³ <https://github.com/TheBlueStuff/cloud-classification/tree/main/CloudClassificationApp>

²⁴ <http://cloud-classification-app.s3-website-us-east-1.amazonaws.com/>

Nimbus

Sube las imágenes de nubes que deseas clasificar:

[Click aquí para seleccionar las imágenes a subir](#)



Cu_001.jpg (116.3 kB)

1 files (116.3 kB in total)



Tipo de Nube: Cúmulos

Descripción: Nubes aisladas, densas y con contornos bien definidos. Se desarrollan verticalmente en forma de montículos cúpulas o torres, son nubes de color blanco brillante y su base es oscura y horizontal. Puede causar precipitación en forma de chubascos.

Probabilidad: 0.76

Figura 19. Primer modo de uso de la interfaz web.

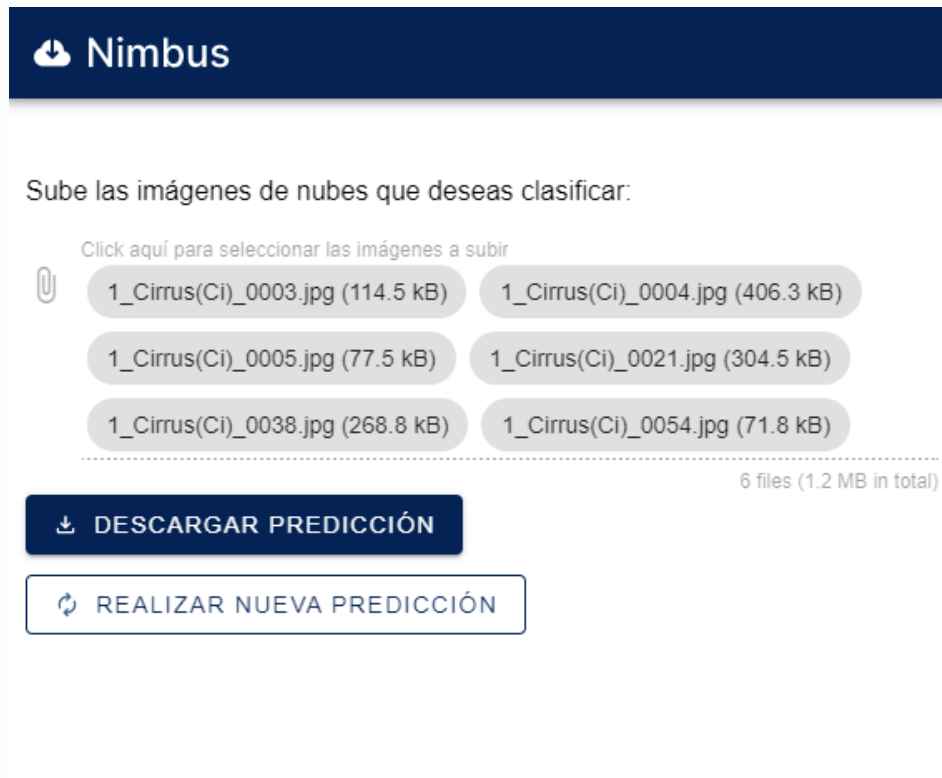


Figura 20. Segundo modo de uso de la interfaz web.

```
File Name, Class
1_Cirrus(Ci)_0005.jpg, Cirros
1_Cirrus(Ci)_0054.jpg, Cirros
1_Cirrus(Ci)_0038.jpg, Cirros
1_Cirrus(Ci)_0003.jpg, Cirros
1_Cirrus(Ci)_0021.jpg, Cirros
1_Cirrus(Ci)_0004.jpg, Cirros
```

Figura 21. Ejemplo de archivo csv que contiene los nombres de archivos y tipos de nubes inferidos

5.4.6. Pruebas funcionales de la interfaz web.

La interfaz web cuenta con un registro para cargar la imagen o imágenes de nubes. En la siguiente tabla se muestra la clase de equivalencia.

Tabla 18

Clase de equivalencia del registro de imagen o imágenes

| Entrada | Clase válida | Clase no válida |
|---------|---|--|
| Imagen | 1. Una o más imágenes (JPG, JPEG o PNG) | 1. Por lo menos un archivo a enviar no es una imagen (JPG, JPEG o PNG) |

Después de definir la clase de equivalencia para el registro de la imagen o imágenes, se define el catálogo de pruebas correspondiente (Tablas 19, 20 y 21).

Tabla 19

Escenario 1 de pruebas funcionales

| | | |
|------------------------------------|--|----------------|
| Objetivo de la prueba | Cargar una imagen para obtener el tipo de nube presente en ella. | |
| Escenario | El usuario carga a la interfaz web una imagen de una nube de tipo JPG. | |
| Resultado esperado | El usuario obtiene el tipo de nube inferido, una breve descripción de esa clase y la probabilidad con la que el modelo hizo la predicción. | |
| Ejemplo de datos ingresados | Dato | Entrada |
| | 1 imagen JPG | imagen.jpg |

Tabla 20

Escenario 2 de pruebas funcionales

| | | |
|------------------------------------|--|---------------------------------------|
| Objetivo de la prueba | Cargar varias imágenes para obtener el tipo de nube presente en cada una de ellas. | |
| Escenario | El usuario carga a la interfaz web 3 imágenes de nubes de tipo JPG. | |
| Resultado esperado | El usuario obtiene un archivo csv que contiene el nombre de cada imagen cargada junto con el tipo de nube presente en ella | |
| Ejemplo de datos ingresados | Dato | Entrada |
| | 3 imágenes | imagen1.jpg, imagen2.jpg, imagen3.jpg |

Tabla 21

Escenario 3 de pruebas funcionales

| | | |
|------------------------------------|---|--|
| Objetivo de la prueba | Cargar varias imágenes para obtener el tipo de nube presente en cada una de ellas. | |
| Escenario | El usuario carga a la interfaz web 2 imágenes de nubes de tipo jpg y un archivo xlsx. | |
| Resultado esperado | El usuario obtiene un mensaje de error: “Error al generar la predicción del tipo de nube: tipo de archivo incorrecto” | |
| Ejemplo de datos ingresados | Dato | Entrada |
| | 2 imágenes y 1 hoja de cálculo | imagen1.jpg, imagen2.jpg, archivo.xlsx |

Los requerimientos para la interfaz web definidos en el presente capítulo y las pruebas funcionales correspondientes fueron validados por un experto en aprendizaje de máquina, el acta de validación se encuentra en el Anexo J.

5.5. Discusión sobre el desarrollo del segundo objetivo

En este objetivo, se implementó una arquitectura de red neuronal convolucional para la clasificación automática de tipos de nubes. Este fin se logró mediante la revisión de la literatura para dar con las arquitecturas que se usaron previamente resolver el problema en cuestión. Posteriormente, se implementaron dichas arquitecturas en Python para luego poder realizar una evaluación comparativa entre ellas y elegir la mejor. Finalmente, se implementó y desplegó una interfaz web que integra el mejor modelo obtenido durante los pasos anteriores con el objetivo de mostrar el funcionamiento de la red neuronal convolucional. Después de realizar las actividades mencionadas, es importante mencionar lo siguiente:

En primer lugar, se eligieron 3 arquitecturas de las más relevantes usadas en la revisión sistemática de la literatura, donde una de ellas fue un modelo entrenado desde cero y las 2 restantes modelos preentrenados. Se concluye que las arquitecturas preentrenadas fueron capaces de aprender más rápido y de manera más efectiva, ya que aprovechan las características

aprendidas previamente en ImageNet (Deng, Dong, Socher, Li, Kai & Li, 2009) y se pone en evidencia así la razón por la cual este enfoque es ampliamente usado en visión computacional; por otro lado, la arquitectura entrenada desde cero aprendía más lento y alcanzaba métricas considerablemente menores.

En segundo lugar, se realizaron 18 experimentos por arquitectura, donde el modelo Inception v3 obtuvo mejores resultados tanto en promedio como globales. La optimización de hiperparámetros resultó una tarea determinante, ya que ayudó a encontrar la mejor configuración para cada arquitectura. En este caso, solamente se afinaron los hiperparámetros más importantes; es decir, los que tenían mayor impacto en la métrica final; sin embargo, existe una oportunidad de mejorar el modelo aún más incluyendo configuraciones que afinen hiperparámetros adicionales. Además, la técnica early stopping ayudó a mitigar el sobreajuste deteniendo el entrenamiento oportunamente, esto se puede comprobar mediante la similitud de la métrica F-score en el grupo de validation y en el grupo de test.

Finalmente, la implementación de la interfaz web pone a disposición del público el modelo para ser probado. Este podrá ser empleado por personal del SENAMHI, de manera que se obtenga retroalimentación para su posterior mejora y puesta en producción.

Conclusiones y trabajos futuros

Conclusiones

Con base en los resultados obtenidos en el desarrollo de este trabajo, se puede llegar a las siguientes conclusiones:

1. Con respecto a la preparación de la base de datos de imágenes de tipos de nubes, se definió un pipeline. Este consistió en obtener ejemplares de 2 fuentes: la web mediante web scraping y bases de datos publicadas en artículos científicos, para fines de este trabajo, se eligieron las bases de datos que proveen imágenes variadas; es decir, con diferentes ángulos de orientación de cámara, tomadas en distintos lugares y que incluyen parte del horizonte. Posteriormente, se dividió la base de datos pertinentemente en los conjuntos de train, validation y test. Finalmente, se aplicó la técnica de data augmentation en el conjunto de train. El pipeline definido garantiza la generación de una base de datos convenientemente separada para el proceso de entrenamiento y prueba de un modelo de aprendizaje de máquina y, además, está estructurado adecuadamente para evitar el problema conocido como data leakage.

2. Con respecto al desarrollo de un modelo de red neuronal convolucional para la clasificación automática de tipos de nubes, se hizo una revisión sistemática de la literatura para determinar las 3 arquitecturas más relevantes. Posteriormente, se implementaron dichas arquitecturas, se definió un flujo de entrenamiento sobre la base de datos creada y se realizó una optimización de hiperparámetros en cada una de las arquitecturas. Finalmente, se realizó una evaluación comparativa y como resultado se eligió la mejor configuración obtenida de Inception v3 como modelo final para este trabajo. Durante todo el proceso de entrenamiento y prueba, se usó la métrica F-score macro como objetivo, ya que es una métrica más robusta con respecto al desbalance de clases en la base de datos.

3. Con respecto a la implementación de una interfaz web que muestre el funcionamiento del modelo de red neuronal convolucional implementado, se desarrollaron 2 módulos que funcionan en conjunto: frontend y backend. El módulo backend provee una API que integra el modelo de red neuronal entrenado; por otro lado, el módulo frontend proporciona una interfaz web de usuario donde es posible subir imágenes de nubes para recibir la inferencia del modelo. Ambos módulos son de acceso público y se encuentran desplegados en AWS S3 y AWS EC2 respectivamente.

De acuerdo con los puntos mencionados anteriormente, se puede afirmar que este trabajo ayuda a mitigar el problema definido inicialmente: la clasificación de tipos de nubes en las estaciones meteorológicas del SENAMHI es un proceso que requiere estrictamente de un observador, por lo que es propensa a errores y laboriosa. Esto se ha logrado a través del desarrollo de un modelo de red neuronal convolucional que permite clasificar imágenes de nubes automáticamente, la solución mencionada será puesta a disposición del SENAMHI para su pase a producción, mantenimiento y mejora continua.

Trabajos futuros

Después de finalizar el presente proyecto, es posible extender su alcance por medio de la realización de los siguientes trabajos:

En primer lugar, la base de datos de imágenes de tipos de nubes generada aún presenta desbalance, ya que existen clases que poseen pocos ejemplares. Sería posible recolectar más imágenes de tipos de nubes mediante la implementación de una aplicación en la cual expertos puedan fotografiar nubes e indicar el tipo al cual pertenecen. Asimismo, se podría aprovechar las fotografías recolectadas en las webcams de las estaciones meteorológicas del SENAMHI para que sean etiquetadas por expertos de la misma institución.

En segundo lugar, el proyecto Meteo-Huascarán contempla la puesta en producción de este modelo, esto se realizará a través de la implementación de una API, la cual será integrada con los sistemas pertinentes. Además, el modelo podría ser ajustado usando las imágenes recolectadas por las webcams del SENAMHI, después de que estas hayan sido etiquetadas por expertos, esto permitiría mejorar aún más el desempeño de la red neuronal, ya que sería entrenada con imágenes que tienen las características del ambiente productivo propio del SENAMHI.

Finalmente, en este trabajo se usan imágenes variadas, tomadas en distintas locaciones, con distintos ángulos y que incluyen parte del horizonte. El hecho de incluir parte del horizonte en el entrenamiento del modelo implica un sesgo, ya que este es variable y de poca utilidad para determinar el tipo de nube. El problema mencionado podría ser solucionado mediante la implementación de una herramienta de segmentación automática para separar el cielo del horizonte. Así, se podría entrenar un modelo solamente con contenido de nubes y realizar la inferencia del mismo modo.

Referencias bibliográficas

- Ackerman, S., & Knox, J. (2011). *Meteorology*. Jones & Bartlett Publishers.
- Amazon Web Services Inc. (s.f.-a). Amazon EC2. Disponible 17 de junio de 2022, de <https://aws.amazon.com/ec2/>
- Amazon Web Services Inc. (s.f.-b). Amazon S3. Disponible 2 de noviembre de 2022, de <https://aws.amazon.com/s3/>
- Ambildhuke, G. M., & Banik, B. G. (2021). Transfer Learning Approach - An Efficient Method to Predict Rainfall Based on Ground-Based Cloud Images. *Ingenierie Des Systemes d'Information*, 26(4), 345–356. <https://doi.org/10.18280/ISI.260402>
- Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2020). Albumentations: Fast and Flexible Image Augmentations. *Information*, 11(2), 125. <https://doi.org/10.3390/info11020125>
- Chen, T., Rossow, W. B., & Zhang, Y. (2000). Radiative Effects of Cloud-Type Variations. *Journal of Climate*, 13(1), 264–286. [https://doi.org/10.1175/1520-0442\(2000\)013<0264:REOCTV>2.0.CO;2](https://doi.org/10.1175/1520-0442(2000)013<0264:REOCTV>2.0.CO;2)
- Cloud Appreciation Society. (2022). *Cloud Appreciation Society Photo Gallery*. <https://cloudappreciationsociety.org/gallery/>
- Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei. (2009). ImageNet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 248–255. <https://doi.org/10.1109/CVPR.2009.5206848>
- Dev, S., Lee, Y. H., & Winkler, S. (2015). Categorization of cloud image patches using an improved texton-based approach. *2015 IEEE International Conference on Image Processing (ICIP)*, 422–426. <https://doi.org/10.1109/ICIP.2015.7350833>

- Diouf, R., Sarr, E. N., Sall, O., Birregah, B., Bousso, M., & Mbaye, S. N. (2019). Web Scraping: State-of-the-Art and Areas of Application. *2019 IEEE International Conference on Big Data (Big Data)*, 6040–6042.
<https://doi.org/10.1109/BigData47090.2019.9005594>
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778. <https://doi.org/10.1109/CVPR.2016.90>
- Kapoor, S., & Narayanan, A. (2022). *Leakage and the Reproducibility Crisis in ML-based Science*.
- Kobrosly, W., & Vassiliadis, S. (1988). *Survey of software functional testing techniques*. 127–134. <https://doi.org/10.1109/stier.1988.95474>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.
<https://doi.org/10.1038/nature14539>
- LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324.
<https://doi.org/10.1109/5.726791>
- Li, M., Liu, S., & Zhang, Z. (2019). Dual Guided Loss for Ground-Based Cloud Classification in Weather Station Networks. *IEEE Access*, *7*, 63081–63088.
<https://doi.org/10.1109/ACCESS.2019.2916905>
- Li, M., Liu, S., & Zhang, Z. (2020). Deep tensor fusion network for multimodal ground-based cloud classification in weather station networks. *Ad Hoc Networks*, *96*, 101991.
<https://doi.org/https://doi.org/10.1016/j.adhoc.2019.101991>

- Liu, S., Duan, L., Zhang, Z., & Cao, X. (2019). Hierarchical Multimodal Fusion for Ground-Based Cloud Classification in Weather Station Networks. *IEEE Access*, 7, 85688–85695. <https://doi.org/10.1109/ACCESS.2019.2926092>
- Liu, S., Duan, L., Zhang, Z., Cao, X., & Durrani, T. S. (2020). Multimodal Ground-Based Remote Sensing Cloud Classification via Learning Heterogeneous Deep Features. *IEEE Transactions on Geoscience and Remote Sensing*, 58(11), 7790–7800. <https://doi.org/10.1109/TGRS.2020.2984265>
- Liu, S., Duan, L., Zhang, Z., Cao, X., & Durrani, T. S. (2021). Ground-Based Remote Sensing Cloud Classification via Context Graph Attention Network. *IEEE Transactions on Geoscience and Remote Sensing*. <https://doi.org/10.1109/TGRS.2021.3063255>
- Liu, S., Lehman, C., & Alegib, G. (2020). *ROBUSTNESS AND OVERFITTING BEHAVIOR OF IMPLICIT BACKGROUND MODELS*.
- Liu, S., & Li, M. (2018). Deep multimodal fusion for ground-based cloud classification in weather station networks. *EURASIP Journal on Wireless Communications and Networking*, 2018(1), 48. <https://doi.org/10.1186/s13638-018-1062-0>
- Liu, S., Li, M., Zhang, Z., Cao, X., & Durrani, T. S. (2020). Ground-Based Cloud Classification Using Task-Based Graph Convolutional Network. *Geophysical Research Letters*, 47(5). <https://doi.org/10.1029/2020GL087338>
- Liu, S., Li, M., Zhang, Z., Xiao, B., & Cao, X. (2018). Multimodal Ground-Based Cloud Classification Using Joint Fusion Convolutional Neural Network. *Remote Sensing*, 10(6), 822. <https://doi.org/10.3390/rs10060822>

- Liu, S., Li, M., Zhang, Z., Xiao, B., & Durrani, T. S. (2020). Multi-Evidence and Multi-Modal Fusion Network for Ground-Based Cloud Recognition. *Remote Sensing*, *12*(3), 464. <https://doi.org/10.3390/rs12030464>
- Liu, Z., Zhou, S., Wang, M., Peng, S., Shen, A., & Zhou, S. (2019). Ground-based Visible-light Cloud Image Classification based on a Convolutional Neural Network. *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, 108–112. 8//7
- Londoño Palacio, O. L., Maldonado Granados, L. F., & Calderón Villafañez, L. C. (2014). *Guías para construir estados del arte*.
- Prechelt, L. (1998). Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, *11*(4), 761–767.
- Ramírez, S. (s.f.). *FastAPI*. Retrieved October 23, 2022, from <https://fastapi.tiangolo.com/>
- Ranjit, M. P., Ganapathy, G., Sridhar, K., & Arumugham, V. (2019). Efficient deep learning hyperparameter tuning using cloud infrastructure: Intelligent distributed hyperparameter tuning with Bayesian optimization in the cloud. *IEEE International Conference on Cloud Computing, CLOUD, 2019-July*, 520–522. <https://doi.org/10.1109/CLOUD.2019.00097>
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, *61*, 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>
- Servicio Nacional de Meteorología e Hidrología del Perú. (s.f.). *¿Quiénes somos?* Disponible 23 de abril de 2022, de <https://www.senamhi.gob.pe/?p=senamhi>
- Servicio Nacional de Meteorología e Hidrología del Perú. (2013). *ESTANDARIZACIÓN PARA EL REGISTRO DE LA METADATA Manual Técnico: MT-DRD-004*. <https://www.senamhi.gob.pe/load/file/00711SENA-73.pdf>

- Servicio Nacional de Meteorología e Hidrología del Perú. (2021). *Meteo-Huascarán*.
- Shi, C., Wang, C., Wang, Y., & Xiao, B. (2017). Deep Convolutional Activations-Based Features for Ground-Based Cloud Classification. *IEEE Geoscience and Remote Sensing Letters*, 14(6), 816–820. <https://doi.org/10.1109/LGRS.2017.2681658>
- Sinko, M., Kamencay, P., Sykora, P., Benco, M., & Hudec, R. (2019). Cloud-type Classification of Ground-Based Images using Deep Learning. *2019 17th International Conference on Emerging ELearning Technologies and Applications (ICETA)*, 715–720. <https://doi.org/10.1109/ICETA48886.2019.9040149>
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2015). *Rethinking the Inception Architecture for Computer Vision*.
- Tillmann, N., De Halleux, J., & Xie, T. (2010). Parameterized unit testing: Theory and practice. *Proceedings - International Conference on Software Engineering*, 2, 483–484. <https://doi.org/10.1145/1810295.1810441>
- Wang, M., Zhou, S., Yang, Z., & Liu, Z. (2020). Clouda: A ground-based cloud classification method with a convolutional neural network. *Journal of Atmospheric and Oceanic Technology*, 37(9), 1661–1668. <https://doi.org/10.1175/JTECH-D-19-0189.1>
- World Meteorological Organization. (2017). *International Cloud Atlas Manual on the Observation of Clouds and Other Meteors*. <https://cloudatlas.wmo.int/>
- World Meteorological Organization. (2019). *Vision for the WMO Integrated Global Observing System in 2040* (Vol. 1243).
- Ye, L., Cao, Z., & Xiao, Y. (2017). DeepCloud: Ground-Based Cloud Image Categorization Using Deep Convolutional Features. *IEEE Transactions on Geoscience and Remote Sensing*, 55(10), 5729–5740. <https://doi.org/10.1109/TGRS.2017.2712809>

- Yu, A., Tang, M., Li, G., Hou, B., Xuan, Z., Zhu, B., & Chen, T. (2021). A novel robust classification method for ground-based clouds. *Atmosphere*, 12(8).
<https://doi.org/10.3390/atmos12080999>
- Zhang, J., Liu, P., Zhang, F., Iwabuchi, H., de Moura, A. A. de H. e A., & de Albuquerque, V. H. C. (2021). Ensemble Meteorological Cloud Classification Meets Internet of Dependable and Controllable Things. *IEEE Internet of Things Journal*, 8(5), 3323–3330. <https://doi.org/10.1109/JIOT.2020.3043289>
- Zhang, J., Liu, P., Zhang, F., & Song, Q. (2018). CloudNet: Ground-Based Cloud Classification With Deep Convolutional Neural Network. *Geophysical Research Letters*, 45(16), 8665–8672. <https://doi.org/10.1029/2018GL077787>
- Zhang, L., Jia, K., Liu, P., & Fang, C. (2020). Cloud Recognition Based on Lightweight Neural Network. *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 1033–1042.
- Zhang, Z., Li, D., & Liu, S. (2018). Salient Dual Activations Aggregation for Ground-Based Cloud Classification in Weather Station Networks. *IEEE Access*, 6, 59173–59181.
<https://doi.org/10.1109/ACCESS.2018.2874994>
- Zhu, T., Wei, L., & Guo, Y. (2021). Cloud classification of ground-based cloud images based on convolutional neural network. *Journal of Physics: Conference Series*, 2035(1).
<https://doi.org/10.1088/1742-6596/2035/1/012020>

Anexos

Anexo A. Páginas web de las herramientas empleadas en el proyecto

Pueden encontrar las páginas de las herramientas usadas en este trabajo a continuación:

- Python: <https://www.python.org/about/>
- Open CV: <https://opencv.org/>
- Jupyter Lab: <https://jupyter.org/>
- Lucid Chart: <https://www.lucidchart.com/pages/es>
- Git: <https://git-scm.com/>
- Google Drive <https://www.google.com/intl/es/drive/>
- Mendeley <https://www.mendeley.com/>
- PyTorch <https://pytorch.org/>
- Vue.js. <https://vuejs.org/>
- Weights & Biases <https://wandb.ai/>



Anexo B. Formulario de extracción de datos

El formulario de extracción de datos completo se puede visualizar a través del vínculo:

https://drive.google.com/file/d/1MunF4A9nte34JY1u9mQtqv0iq8L_Pfwu/view?usp=sharing



Anexo C. Plan de Proyecto

● **Justificación**

Las nubes influyen en los fenómenos climáticos y meteorológicos; por lo tanto, su observación precisa es esencial para el análisis y la predicción del tiempo (Ye et al., 2017). En Perú, el Servicio Nacional de Meteorología e Hidrología (SENAMHI) es la entidad encargada del monitoreo meteorológico, el cual se realiza por medio de una red de estaciones meteorológicas distribuidas en todo el país (Servicio Nacional de Meteorología e Hidrología del Perú, s.f.). Las estaciones mencionadas están interconectadas y reportan a una central de procesamiento las variables medidas u observadas (Servicio Nacional de Meteorología e Hidrología del Perú, 2013). Una de las variables tomadas en cuenta son los tipos de nubes que se pueden observar en la atmósfera en un momento determinado, el registro de esta variable requiere estrictamente de un observador, por lo que es un proceso propenso a errores y laborioso. En consecuencia, este trabajo de tesis busca contribuir a la automatización de la detección de los tipos de nubes en las estaciones meteorológicas del SENAMHI por medio de una herramienta de clasificación automática de tipos de nubes basada en una red neuronal convolucional. Tener automatizado este proceso contribuiría a poder recolectar datos en las estaciones meteorológicas que cuentan con webcams y de esta manera proporcionar información útil que podrá ser usada para el análisis meteorológico e investigación científica. Además, la automatización de este proceso ayudaría al SENAMHI a adaptarse de manera más fácil a las necesidades futuras de los usuarios en materia de observación meteorológica, ya que en los próximos años se espera mejorar los procesos relacionados con automatización y tecnología, de manera que los diversos sistemas de observación existentes tengan un mayor nivel de interoperabilidad (World Meteorological Organization, 2019). Finalmente, este trabajo se realiza en colaboración con el SENAMHI y el grupo de investigación IA-PUCP en contexto del proyecto Meteo-Huascarán: Ecoturismo seguro con monitoreo y pronóstico meteorológico automatizado con machine learning en el Parque Nacional Huascarán. Los resultados de este trabajo contribuirán al diseño de un sistema de observación meteorológica automática para el monitoreo y pronóstico de peligros de origen hidrometeorológico en el Parque Nacional Huascarán (Servicio Nacional de Meteorología e Hidrología del Perú, 2021).

● **Viabilidad**

En primer lugar, este trabajo es viable en términos de recursos, ya que se realiza en el contexto del proyecto Meteo-Huascarán; por lo tanto, se contará con el apoyo de expertos en

meteorología de parte del SENAMHI. Además, el grupo de investigación IA-PUCP proporcionará el hardware necesario para el entrenamiento de las redes neuronales convolucionales, la implementación de todo el proyecto se realizará con la ayuda de herramientas gratuitas y el producto final será desplegado en la nube usando los créditos disponibles en AWS Academy proporcionados por la universidad.

En segundo lugar, el presente trabajo es viable en términos de conocimientos, ya que el autor tiene experiencia en el desarrollo de interfaces web, modelos de aprendizaje profundo y aprendizaje de máquina. Además, el asesor y el coasesor son expertos en temas de ciencias de la computación, de esta manera ellos también aportarán al desarrollo del proyecto.

Finalmente, este trabajo es viable en términos de acceso a la información, ya que la biblioteca de la universidad cuenta con diversos recursos en línea que permiten el uso de bases de datos de artículos científicos en temas de ciencias de la computación tales como IEEE Electronic Library, Scopus y Web of Science.

- **Alcance**

Este trabajo pertenece al área de ciencias de la computación, específicamente a visión computacional, se hará uso de la técnica aprendizaje supervisado para desarrollar una herramienta basada en una red neuronal convolucional que permita clasificar tipos de nubes de manera automática.

En consecuencia, este proyecto contempla, en primer lugar, la preparación de una base de datos representativa, suficientemente heterogénea y grande para el entrenamiento adecuado de un modelo de clasificación automática de tipos de nubes, esta base de datos usará imágenes de 2 fuentes: bases de datos definidas en artículos científicos e imágenes obtenidas en la web por medio de *web scraping*. En segundo lugar, se realizará una revisión sistemática de tipo estado del arte para encontrar las arquitecturas de redes neuronales convolucionales que han sido usadas para la clasificación automática de tipos de nubes, se implementarán las 3 arquitecturas más relevantes y se seleccionará una por medio de una evaluación comparativa de su desempeño. Finalmente, una interfaz web será desarrollada para mostrar el funcionamiento del modelo de red neuronal convolucional seleccionado, esta integrará el modelo y lo usará para inferir el tipo de nube presente en la imagen que será subida por el usuario.

- **Limitaciones**

La implantación de la herramienta implementada y la puesta en producción de la misma en las estaciones meteorológicas del SENAMHI no está contemplada en este proyecto.

● **Identificación de los riesgos del proyecto**

En la Tabla B.1, se definirán los riesgos de este proyecto. Se incluirá la descripción de cada riesgo, sus síntomas, la probabilidad de que suceda, el impacto que tendría en el proyecto si ocurriera, la severidad (probabilidad x impacto), las acciones de mitigación y las acciones de contingencia. Los valores que pueden tomar los campos de probabilidad e impacto son los siguientes: 1 (bajo), 2 (medio), 3 (alto) y 4 (muy alto).

Tabla B1

Análisis de riesgos del proyecto

| | Riesgo 1 | Riesgo 2 | Riesgo 3 | Riesgo 4 |
|---------------------|---|--|--|--|
| Descripción | Pérdida de avances del documento escrito o de código por fallo del ordenador personal | Falta de disponibilidad de recursos de hardware del grupo IA-PUCP para el entrenamiento de los modelos | Las actividades del proyecto toman más tiempo de lo planificado | Los especialistas y/o expertos designados para apoyar en el proyecto no disponen de tiempo |
| Síntomas | <ul style="list-style-type: none"> El ordenador personal es antiguo y presenta fallas constantemente El estado de salud del disco duro del ordenador no es bueno | Las tarjetas gráficas están siendo usadas constantemente a su máxima capacidad | Retrasos en completar las primeras actividades del proyecto | <ul style="list-style-type: none"> Demoras en la respuesta de correos Ausencia en las reuniones planificadas |
| Probabilidad | 2 | 1 | 2 | 1 |
| Impacto | 4 | 2 | 3 | 4 |
| Severidad | 8 | 2 | 6 | 4 |
| Mitigación | <ul style="list-style-type: none"> Usar un programa de control de versiones como Git para guardar el código Usar un recurso en la nube como Google drive para guardar los avances del documento escrito | <ul style="list-style-type: none"> Coordinar con los investigadores que usan constantemente los recursos de hardware con el objetivo de determinar horarios para su uso | Hacer una planificación detallada que tome en cuenta unos días adicionales por cada actividad, para esto se podría abarcar días previos al inicio del semestre | Planificar horarios fijos para reuniones semanales, quincenales o mensuales con los expertos y/o especialistas |

| | | | | |
|---------------------|--|---|--|---|
| Contingencia | Usar un ordenador alternativo o uno de la universidad para continuar con el desarrollo del proyecto desde la última versión disponible | Usar recursos alternativos para el entrenamiento de los modelos como Google Colab o AWS Sagemaker | Cambiar la planificación a partir de la última actividad completada, de manera que se incluyan más horas de trabajo en el proyecto | Coordinar con el responsable del proyecto para obtener ayuda de otros expertos y/o especialistas que cuenten con tiempo |
|---------------------|--|---|--|---|

- Estructura de descomposición del trabajo (EDT)



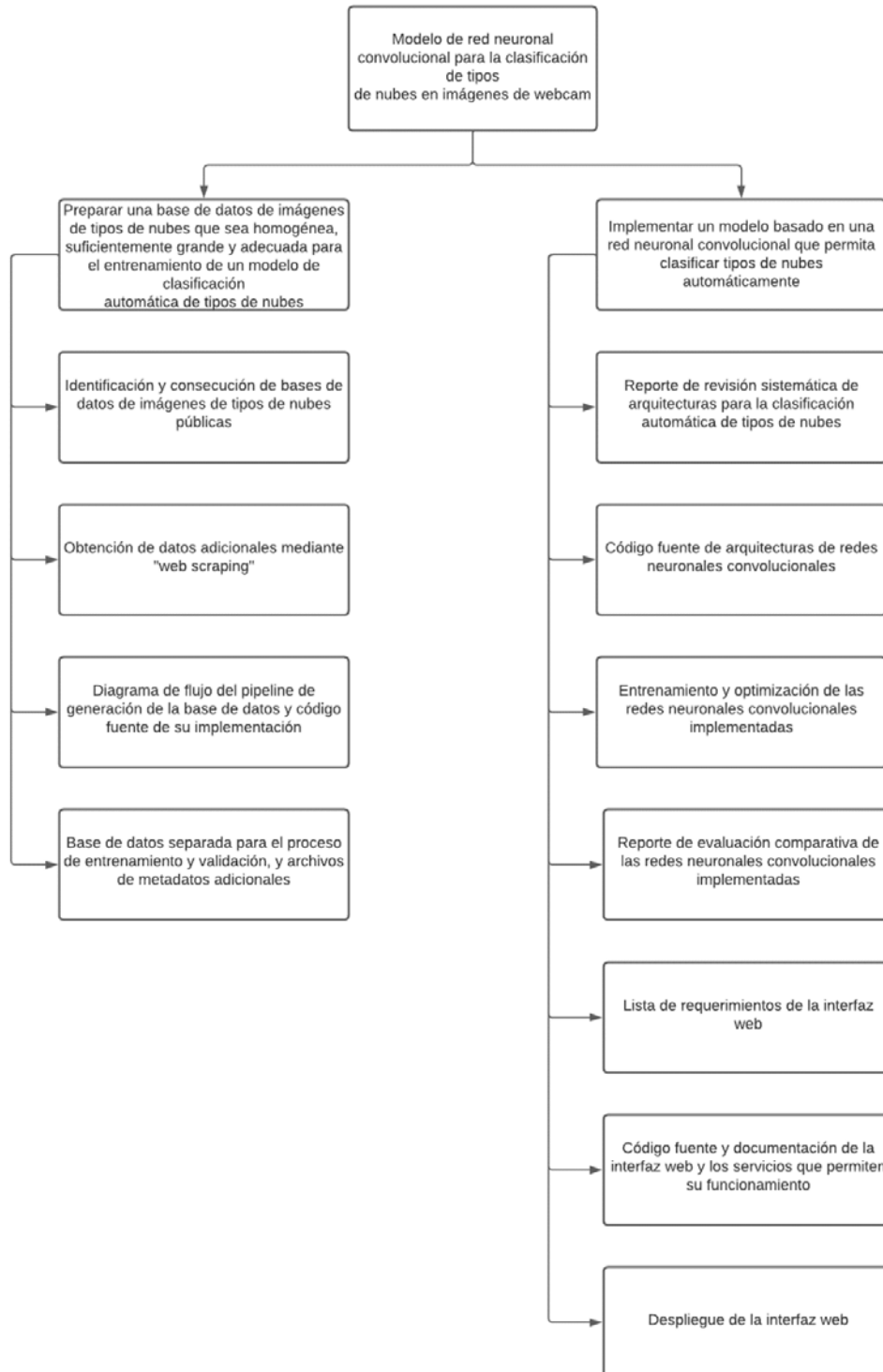


Figura B.1. Estructura de descomposición del trabajo

- **Lista de tareas**

Tabla B2

Lista de tareas del proyecto

| Id | Tarea | Duración en semanas | Esfuerzo en horas-persona | Costo estimado en soles |
|-----------|--|----------------------------|----------------------------------|--------------------------------|
| T1 | Identificación y consecución de bases de datos de imágenes de tipos de nubes públicas (R1) | 1 | 10 | 400 |
| T2 | Obtención de datos adicionales mediante web scraping (R1) | 1 | 10 | 400 |
| T3 | Diagrama de flujo del pipeline de generación de la base de datos y código fuente de su implementación (R1) | 2 | 20 | 800 |
| T4 | Base de datos separada para el proceso de entrenamiento y prueba, y archivos de metadatos adicionales (R2) | 1 | 10 | 400 |
| T5 | Reporte de revisión sistemática de arquitecturas para la clasificación automática de tipos de nubes (R3) | 1 | 10 | 400 |
| T6 | Código fuente de arquitecturas de redes neuronales convolucionales (R4) | 2 | 20 | 800 |
| T7 | Entrenamiento y optimización de las redes neuronales convolucionales implementadas (R5) | 2 | 20 | 800 |
| T8 | Reporte de evaluación comparativa de las redes neuronales convolucionales implementadas (R5) | 2 | 20 | 800 |
| T9 | Lista de requerimientos de la interfaz web (R6) | 0.5 | 5 | 200 |
| T10 | Código fuente y documentación de la interfaz web y los servicios que permiten su funcionamiento (R6) | 1 | 10 | 400 |
| T11 | Despliegue de la interfaz web (R6) | 0.5 | 5 | 200 |
| A1 | Asesorías | 12 | 24 | 2400 |

- **Cronograma del proyecto**

Tabla B3

Cronograma del proyecto

| Id | Tarea | Fecha de inicio | Fecha de fin | Dependencia |
|-----------|--|------------------------|---------------------|--------------------|
| T1 | Identificación y consecución de bases de datos de imágenes de tipos de nubes públicas (R1) | 01/08/2022 | 07/08/2022 | Ninguna |
| T2 | Obtención de datos adicionales mediante web scraping (R1) | 01/08/2022 | 07/08/2022 | Ninguna |
| T3 | Diagrama de flujo del pipeline de generación de la base de datos y código fuente de su implementación (R1) | 08/08/2022 | 21/08/2022 | T1, T2 |
| T4 | Base de datos separada para el proceso de entrenamiento y prueba, y archivos de metadatos adicionales (R2) | 22/08/2022 | 28/08/2022 | T3 |
| T5 | Reporte de revisión sistemática de arquitecturas para la clasificación automática de tipos de nubes (R3) | 29/08/2022 | 04/09/2022 | Ninguna |
| T6 | Código fuente de arquitecturas de redes neuronales convolucionales (R4) | 05/09/2022 | 18/09/2022 | T5 |
| T7 | Entrenamiento y optimización de las redes neuronales convolucionales implementadas (R5) | 19/09/2022 | 02/10/2022 | T6 |
| T8 | Reporte de evaluación comparativa de las redes neuronales convolucionales implementadas (R5) | 03/10/2022 | 16/10/2022 | T7 |
| T9 | Lista de requerimientos de la interfaz web (R6) | 17/10/2022 | 20/10/2022 | Ninguna |
| T10 | Código fuente y documentación de la interfaz web y los servicios que permiten su funcionamiento (R6) | 21/10/2022 | 28/10/2022 | T9 |

| | | | | |
|-----|------------------------------------|------------|------------|------------|
| T11 | Despliegue de la interfaz web (R6) | 29/10/2022 | 01/11/2022 | T10 |
| A1 | Asesorías | 01/08/2022 | 01/11/2022 | Recurrente |

- **Lista de recursos**

- **Personas involucradas y necesidades de capacitación**

- **Asesor:** Dr. César Armando Beltrán Castañón
- **Coasesor:** Dr. Edwin Rafael Villanueva Talavera
- **Tesista:** Mario Alejandro Andonaire Tuesta

- **Materiales requeridos para el proyecto**

No aplica

- **Estándares utilizados en el proyecto**

No aplica

- **Equipamiento requerido**

- Computador personal con acceso a internet
- Servidores con unidades de procesamiento gráfico

- **Herramientas requeridas**

- Python
- Open CV
- Jupyter Lab
- Lucid Chart
- Git
- Google Drive
- Mendeley
- Pytorch
- Visual Studio Code
- Vue.js
- FastAPI

- AWS EC2
- Weights and biases

- **Costeo del Proyecto**

Tabla B4

Costeo del proyecto

| Ítem | Descripción | Unidad | Cantidad | Valor Unidad (S/.) | Monto Parcial (S/.) | Monto Total (S/.) | | |
|-----------|--|--------------|---------------|--------------------|---------------------|-------------------|-----|------------|
| 0 | Costo total del proyecto | --- | --- | --- | --- | 9,710 | | |
| 1. | Estudiantes o tesistas | --- | --- | --- | --- | 9,160 | | |
| 1.1 | Asesor de tesis | Horas | 12 | 100 | 1200 | | | |
| 1.2 | Coasesor de tesis | Horas | 12 | 80 | 960 | | | |
| 1.3 | Tesista | Horas | 140 | 50 | 7000 | | | |
| 3. | Servicios y consultoría (en caso aplique) | --- | --- | --- | --- | 80 | | |
| 3.1 | Amazon EC2 | Informe | 1000 | 0.08 | 80 | | | |
| 4. | Bienes y equipos | Unid1 | Cant1- | Unid2 | Cant 2- | - | - | 470 |
| 4.1 | Computador personal | Equipo | 1 | Horas | 140 | 2.5 | 350 | |
| 4.2 | Servidor IA-PUCP | Equipo | 1 | Horas | 60 | 2 | 120 | |

Anexo D. Acta de validación del resultado 1

El acta de validación del resultado 1: Pipeline para la creación de la base de datos de imágenes de tipos de nubes se puede visualizar a través del vínculo:

<https://drive.google.com/file/d/11ovRJO96xsWMY3jwLWhcpNFBExT9jLXr/view?usp=sharing>



Anexo E. Bases de datos encontradas en la revisión sistemática de la literatura

En esta sección, se provee los detalles de cada base de datos obtenida; además se explica la forma cómo se consiguió.

- SWIMCAT (Dev et al., 2015)

Se completó un formulario en <http://vintage.winklerbros.net/swimcat.html> y posteriormente se recibió el enlace de descarga vía correo electrónico. En la siguiente tabla, se muestra la cantidad de ejemplares por clase que contiene este conjunto de imágenes.

Tabla D1

Cantidad de ejemplares por clase de Swimcat

| Clase | Número de ejemplares |
|----------------|----------------------|
| Despejado | 224 |
| Patrón | 89 |
| Oscuras densas | 251 |
| Blancas densas | 135 |
| Velo | 85 |

Esta base de datos cuenta, además, con información adicional como la hora, la elevación y el azimut. El archivo con la base de datos “swimcat.zip” se puede obtener mediante el vínculo:

<https://drive.google.com/file/d/1M8JVRAAnolCYM4idb5TKX1THqedKLTarD/view?usp=sharing>

- CCSN (Zhang et al., 2018)

Esta base de datos se obtuvo mediante el vínculo:

<https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/CADDPD>

En la siguiente tabla se muestra la cantidad de ejemplares por clase que contiene este conjunto de imágenes.

Tabla D2

Cantidad de ejemplares por clase de CCSN

| Clase | Número de ejemplares |
|--------------------|----------------------|
| Altocúmulos | 221 |
| Altoestratos | 188 |
| Cumulonimbos | 242 |
| Cirrocúmulos | 268 |
| Cirros | 139 |
| Cirrostratos | 287 |
| Cúmulos | 187 |
| Nimbostratos | 274 |
| Estratocúmulos | 340 |
| Estratos | 202 |
| Nubes artificiales | 200 |

El archivo con la base de datos “CCSN_V2.zip” se puede obtener mediante el vínculo:

<https://drive.google.com/file/d/1dzu0S4xNpIejFx3Tc-SWTBNFE94hZB9P/view?usp=sharing>

- GCD (Liu et al., 2021)

Se completó el formulario especificado en <https://github.com/shuangliutjnu/TJNU-Ground-based-Cloud-Dataset>, se envió por correo electrónico a los autores y se obtuvo el enlace para descargar.

En la siguiente tabla, se muestra la cantidad de ejemplares por clase que contiene este conjunto de imágenes.

Tabla D3

Cantidad de ejemplares por clase de GCD

| Clase | Número de ejemplares |
|---|----------------------|
| Cúmulos | 1525 |
| Altocúmulos y cirrocúmulos | 1475 |
| Cirros y cirrostratos | 1906 |
| Despejado | 3739 |
| Estratocúmulos, estratos y altoestratos | 3636 |
| Cumulonimbos y nimbostratos | 5764 |
| Mixto | 955 |

El archivo con la base de datos “GCD.zip” se puede obtener mediante el vínculo:

https://drive.google.com/file/d/1Wk4XQNw-dzyshOMxg89Fw_ZSHt0Isy1J/view?usp=sharing

- GRSCD (Liu et al., 2020)

Se completó el formulario especificado en <https://github.com/shuangliutjnu/TJNU-Ground-based-Remote-Sensing-Cloud-Database>, se envió por correo electrónico a los autores y se obtuvo el enlace para descargar.

En la siguiente tabla, se muestra la cantidad de ejemplares por clase que contiene este conjunto de imágenes.

Tabla D4

Cantidad de ejemplares por clase de GRSCD

| Clase | Número de ejemplares |
|---|----------------------|
| Cúmulos | 1438 |
| Altocúmulos y cirrocúmulos | 731 |
| Cirros y cirrostratos | 1323 |
| Despejado | 1338 |
| Estratocúmulos, estratos y altoestratos | 963 |

| | |
|-----------------------------|------|
| Cumulonimbos y nimbostratos | 1187 |
| Mixto | 1020 |

El archivo con la base de datos “GRSCD.zip” se puede obtener mediante el vínculo:
https://drive.google.com/file/d/1ZENig0C_kBfpdiNCtnnZ3qbwHSXwTRM3/view?usp=sharing

- MGCD (Liu et al., 2020)

Se completó el formulario especificado en <https://github.com/shuangliutjnu/Multimodal-Ground-based-Cloud-Database>, se envió por correo electrónico a los autores y se obtuvo el enlace para descargar.

En la siguiente tabla, se muestra la cantidad de ejemplares por clase que contiene este conjunto de imágenes.

Tabla D5

Cantidad de ejemplares por clase de MGCD

| Clase | Número de ejemplares |
|---|----------------------|
| Cúmulos | 1438 |
| Alto cúmulos y cirro cúmulos | 731 |
| Cirros y cirrostratos | 1323 |
| Despejado | 1338 |
| Estratocúmulos, estratos y altoestratos | 963 |
| Cumulonimbos y nimbostratos | 1187 |
| Mixto | 1020 |

Este conjunto de datos incluye, además, información del tiempo en el momento en que se tomó la imagen (temperatura, humedad, precipitación y velocidad del viento).

El archivo con la base de datos “MGCD.zip” se puede obtener mediante el vínculo:
<https://drive.google.com/file/d/1MhJID-7EhPavkiOIkcctd9KBiPIWpjdPV/view?usp=sharing>

- HBMCD (Zhang et al., 2020)

Esta base de datos se obtuvo del siguiente repositorio de GitHub (<https://github.com/SadaharuZL/HuaYun-BJUT-MIP-Cloud-Dataset>), en el cual el autor publicó un enlace de descarga.

En la siguiente tabla, se muestra la cantidad de ejemplares por clase que contiene este conjunto de imágenes.

Tabla D6

Cantidad de ejemplares por clase de HBMCD

| Clase | Número de ejemplares |
|-----------------|----------------------|
| Altocúmulos | 3114 |
| Altoestratos | 3047 |
| Cumulonimbos | 3547 |
| Cirrocúmulos | 3290 |
| Cirros | 3607 |
| Cirrostratos | 3292 |
| Cúmulos | 3526 |
| Nimbostratos | 3735 |
| Estratocúmulos | 3232 |
| Estratos | 3791 |
| Sol (despejado) | 4000 |

El archivo con la base de datos “HBMCD.zip” se puede obtener mediante el vínculo:

<https://drive.google.com/file/d/1nWKLq4ymopzwNjKvpXbSQKVAVSZV6vK/view?usp=sharing>

Anexo F. Base de datos de Cloud Appreciation Society

En la siguiente tabla, se muestra la cantidad de ejemplares por clase que contiene este conjunto de imágenes.

Tabla E1

Cantidad de ejemplares por clase de CAS

| Clase | Número de ejemplares |
|----------------|-----------------------------|
| Altocúmulos | 3997 |
| Altoestratos | 112 |
| Cumulonimbos | 1490 |
| Cirrocúmulos | 127 |
| Cirros | 1073 |
| Cirrostratos | 236 |
| Cúmulos | 1969 |
| Nimbostratos | 10 |
| Estratocúmulos | 833 |
| Estratos | 359 |

Anexo G. Acta de validación del resultado 3

El acta de validación del resultado 3: Revisión sistemática de arquitecturas de redes neuronales convolucionales que han sido usadas para la clasificación automática de tipos de nubes se puede visualizar a través del vínculo:

https://drive.google.com/file/d/1S2UfjB56YAnYiWMfpdvkutmFnkpW6p/view?usp=share_link



Anexo H. Acta de validación del resultado 5

El acta de validación del resultado 5: Evaluación comparativa del desempeño de distintas arquitecturas de redes neuronales convolucionales en la base de datos se puede visualizar a través del vínculo:

https://drive.google.com/file/d/1ztpPi_y0wRQ2WmnFBdpoXTYqWe7k84pH/view?usp=share_link



Anexo I. Documentación de servicios del módulo backend

La documentación de los servicios del módulo backend se puede encontrar en el vínculo:

<http://ec2-18-211-27-134.compute-1.amazonaws.com:8000/docs>



Anexo J. Acta de validación del resultado 6

El acta de validación del resultado 6: Desarrollo de una interfaz web que muestre el funcionamiento del modelo de red neuronal convolucional implementado se puede visualizar a través del vínculo:

https://drive.google.com/file/d/1RMSrLygQSY3sk3Ri8oX7Xp2XHoEzl-0d/view?usp=share_link

