UNCUYO
UNIVERSIDAD NACIONAL DE CUYO

FACULTAD DE INGENIERÍA

TESINA FINAL DE CARRERA

Deep Learning para el Análisis y Predicción de Accidentes Vehiculares en Mendoza

AUTOR: Andrés Maglione

TUTORA: Dra. Ana Carolina Olivera

2026

Página intencionalmente dejada en blanco.

Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas que hicieron posible la realización de este trabajo final de carrera. Sin su apoyo y colaboración, este proyecto no habría sido posible.

En primer lugar, agradezco a mi tutora de tesina, la Dra. Ana Carolina Olivera, por su constante guía, disponibilidad y apoyo a lo largo de todo el proceso. Su experiencia, compromiso y su presencia activa me ayudaron a superar los desafíos que surgieron durante el desarrollo de este trabajo y mantener la constancia en la investigación. También quiero agradecer al Dr. Ing. Pablo Javier Vidal, por sus valiosos aportes, sus sugerencias y su predisposición.

A la Dirección de Transformación Digital, Smart Cities y Gobierno Abierto de la Ciudad de Mendoza. En particular, al Geog. Luciano Pedro Santoni de la Dirección de Catastro Municipal por los datos de los accidentes vehiculares.

A todos los profesores de la carrera, quienes me brindaron las herramientas y conocimientos necesarios durante todos estos años de formación. Todos ellos han sido una gran inspiración y han contribuido de manera significativa a mi desarrollo académico y personal.

Mi agradecimiento más profundo es a mi familia por su apoyo incondicional en cada etapa de este proyecto. Gracias por acompañarme en cada paso de mi formación, por su comprensión, paciencia y por siempre estar ahí para celebrar los logros y brindar apoyo en los momentos difíciles.

A mis amigos, que me acompañaron en cada paso, me inspiraron e hicieron la experiencia universitaria mucho más enriquecedora.

¡Muchas gracias a todos!

Andrés Maglione
Mendoza, Marzo 2026

Resumen

Los accidentes de tránsito constituyen un problema global que afecta la seguridad, la salud pública y la economía de las ciudades, representando la principal causa de muerte entre niños y adultos jóvenes. A pesar de que algunas ciudades han logrado reducir significativamente el número de accidentes, otras, como la ciudad de Mendoza, continúan enfrentando retos considerables.

El desafío de disminuir el impacto de la accidentología vial es una tarea compleja que requiere la colaboración de diversos actores. Teniendo en cuenta los avances recientes en el campo del aprendizaje profundo, este trabajo se desarrolla una herramienta de predicción de riesgo de accidentes de tránsito en la ciudad de Mendoza utilizando un modelo basado en la arquitectura *Vision Transformer* (ViT) y otro basado en una arquitectura híbrida entre Memoria de Largo-Corto Plazo (*Long Short-Term Memory*, LSTM) y Red Neuronal Convolucional (*Convolutional Neural Network*, CNN). Para ello, se emplean datos históricos de accidentes de tránsito, información meteorológica y datos de tráfico, y se obtienen resultados competitivos con el estado del arte.

Por último, se construyó una aplicación web interactiva que permite visualizar el mapa de riesgo para cada día con el fin de dar apoyo a la toma de decisiones en lo que respecta al manejo de recursos como servicios de emergencia y prevención.

Abstract

Traffic accidents constitute a global problem affecting urban safety, public health, and the economy, representing the leading cause of death among children and young adults. While some cities have managed to significantly reduce their accident rates, others, such as Mendoza, Argentina, continue to face considerable difficulties in this regard.

Reducing the impact of traffic accidents is a complex challenge that requires collaboration among diverse stakeholders. Building on recent advances in deep learning, this work develops a traffic accident risk prediction tool for the city of Mendoza using two neural network architectures: one based on ViT and another on a hybrid LSTM–CNN design. The models are trained on historical accident records, meteorological data, and traffic information, achieving results competitive with the state of the art.

In addition, an interactive web application was built to visualize the risk map for each day, supporting decision making regarding the allocation of resources such as emergency response and prevention services.

Índice general

Índice de figuras

Índice de tablas

Siglas

AI Inteligencia Artificial. (*p. 7, 12*)
ANN Redes Neuronales Artificiales. (*p. 22, 26, 28, 34, 36*)
ANSV Agencia Nacional de Seguridad Vial. (*p. 8*)
API Interfaz de Programación de Aplicaciones. (*p. 10, 11, 19, 39, 60, 62–64*)
AUC Área Bajo la Curva. (*p. 18*)

CNN Red Neuronal Convolucional. (*p. iii, v, 7, 25, 26, 33, 34, 36, 53*)
CORS Intercambio de Recursos de Origen Cruzado. (*p. 63, 65*)
CRS Sistema de Referencia de Coordenadas. (*p. 9*)
CSR Renderizado del Lado del Cliente. (*p. 11*)
CSS Hojas de Estilo en Cascada. (*p. 11*)

DL Aprendizaje Profundo. (*p. 4, 5, 7, 20, 28–30, 33, 35–38, 51, 55, 58, 60, 67–69, 88, 102–104, 147*)

ETL Extracción, Transformación y Carga. (*p. 7, 19, 60*)

GCN Red Convolucional de Grafos. (*p. 35*)
GNN Red Neuronal de Grafos. (*p. 35, 36*)
GPU Unidad de Procesamiento Gráfico. (*p. 29, 60*)

HTML Lenguaje de Marcado de Hipertexto. (*p. 11, 12*)
HTTP Protocolo de Transferencia de Hipertexto. (*p. 10, 11*)

LLM Grandes Modelos de Lenguaje. (*p. 29, 30*)
LSTM Memoria de Largo-Corto Plazo. (*p. iii, v, 7, 26–29, 34–36, 53*)

MAP Precisión Promedio Media. (*p. 17, 18, 59, 70, 74–76, 88, 95*)
ML Aprendizaje Automático. (*p. 4, 5, 7, 12, 13, 15, 19, 20, 29, 33, 35, 36, 45, 46, 59, 60, 62, 63*)
MLP Perceptrón Multicapa. (*p. 22, 34, 35*)

NER Reconocimiento de Entidades Nombradas. (*p. 32*)
NLP Procesamiento de Lenguaje Natural. (*p. 12*, *32*)

POI Puntos de Interés. (*p. 39*, *41*, *48*, *50*, *59*, *65*)

ReLU Unidad Lineal Rectificada. (*p. 23*)
REST Transferencia de Estado Representacional. (*p. 11*, *60*)
RNN Red Neuronal Recurrente. (*p. 7*, *26*, *27*, *29*, *32*, *76*, *77*)
ROC Característica Operativa del Receptor. (*p. 18*)

SMN Servicio Meteorológico Nacional. (*p. 39*)
SSR Renderizado del Lado del Servidor. (*p. 11*, *12*, *62*, *65*)

URI Identificador Uniforme de Recursos. (*p. 11*)
UTM Universal Transversal de Mercator. (*p. 10*)

ViT *Vision Transformer*. (*p. iii*, *v*, *32*, *34–36*, *51*, *55*, *102*)
VPS Servidor Privado Virtual. (*p. 65*)

WAF Cortafuegos de Aplicación Web. (*p. 60*, *65*)
WGS84 Sistema Geodésico Mundial 1984. (*p. 10*)

1

Introducción

1.1. Motivación

Uno de los elementos que caracteriza a las ciudades es su tráfico vehicular. En los últimos años el aumento del parque automotriz en todo el mundo ha provocado un aumento en la circulación de los vehículos tanto públicos como privados. Los accidentes vehiculares se han vuelto algo cotidiano en las metrópolis por lo que los tomadores de decisiones deben tomar medidas para mitigar su impacto.

Los accidentes de tránsito son un problema global por su impacto multifacético. Según la Organización Mundial de la Salud (OMS), comprenden la principal causa de muerte entre niños y adultos jóvenes, y su impacto económico llega se estima entre el 1 y el 3 % de la economía de cada país [1].

A pesar de que las estadísticas muestran que la tasa de mortalidad por accidentes viales ha disminuido a nivel mundial en los últimos años, el continente americano representa la excepción, siendo, junto con África, las únicas regiones que han experimentado un aumento en la tasa de mortalidad por accidentes viales [1].

Según estadísticas del año 2024, en Argentina cerca de 16 personas fallecen por día a causa de accidentes de tránsito, lo que representa más de 5 900 víctimas fatales al año. También se estima que la cantidad de heridos supera a las 100 000 personas anualmente y el costo económico se encuentra por encima de los U$S 10 000 millones anuales [2].

Particularmente, en la Ciudad de Mendoza, se han registrado más de 9 000 accidentes de tránsito desde el año 2023 [3]. Este número solamente incluye los accidentes reportados a las autoridades, por lo que la cifra real podría ser considerablemente ma-

yor. Estos accidentes no solo afectan la seguridad y el bienestar de los ciudadanos, sino que también generan congestión vehicular y aumentan los costos asociados al sistema de salud y a la infraestructura vial, lo que nos lleva a preguntarnos: ¿es posible disminuir el impactos de los accidentes a través de medidas preventivas?

Esta problemática local coincide con la creciente disponibilidad de fuentes de datos urbanos en la ciudad. Los registros históricos de accidentes, información geoespacial detallada, sensores inteligentes distribuidos sobre la malla urbana y datos en tiempo real ofrecen una oportunidad para comprender y, fundamentalmente, predecir la ocurrencia de estos eventos, con el objetivo de implementar medidas preventivas y mejorar la seguridad vial.

Una herramienta preventiva puede ayudar a distribuir eficientemente los recursos de emergencia y preventivos, permitiendo una respuesta más rápida y efectiva ante posibles accidentes para minimizar su impacto. En la ciudad de Mendoza, podemos destacar la labor de los Preventores, agentes municipales de seguridad ciudadana que actúan como un primer anillo de contención y apoyo a la policía provincial. De contar con una herramienta apropiada, actores en puestos de toma de decisiones podrían optimizar la labor del cuerpo de preventores para atacar el problema de la accidentología vial.

El estudio de la accidentología vial es una rama establecida dentro de la ingeniería de tránsito y la seguridad vial. Actualmente, los enfoques basados en Aprendizaje Automático (ML) y particularmente en Aprendizaje Profundo (DL) se colocan en el centro de atención, debido a su capacidad para manejar grandes volúmenes de datos de diferentes fuentes y extraer patrones complejos que pueden no ser evidentes mediante métodos tradicionales. En este trabajo se busca estudiar técnicas que permitan predecir el riesgo de accidentes de manera espacio-temporal. Esto quiere decir que se busca identificar áreas y momentos de alto riesgo de accidentes de tránsito en entornos urbanos, particularmente en la Ciudad de Mendoza, Argentina. Para conseguirlo, se aplicarán modelos de Aprendizaje Profundo (DL) según lo descrito en el capítulo 2.

Además, se implementará una herramienta web que permita visualizar las predicciones generadas por los modelos en conjunto con otros datos de interés como la presencia de semáforos, colegios, hospitales, entre otros.

1.2. Objetivos

El objetivo general de este trabajo final de carrera es desarrollar una herramienta que permita clasificar, en base a la información de sucesos pasados, el riesgo de que ocurra un accidente de tránsito en cada sector de la ciudad de Mendoza, para un día en particular.

Como objetivos específicos podemos mencionar:

- Desarrollar una arquitectura de Aprendizaje Profundo (DL) que, a partir de los datos sobre accidentes viales y otra información ambiental, pueda predecir la probabilidad de que ocurra un accidente.
- Implementar un prototipo web que permita, a partir de datos de accidentes de tráfico en la ciudad de Mendoza, mostrar dónde ocurrirían accidentes con mayor probabilidad sobre un mapa de la Ciudad de Mendoza dividido en secciones.

El resultado de la tesina final de carrera será una aplicación web interactiva que muestre el mapa de la Ciudad de Mendoza con la información sobre la probabilidad de que ocurra un accidente vehicular en cada sector. Esto permitirá poner en alerta a servicios de emergencia y preventores o establecer políticas a largo plazo de acción como campañas preventivas.

1.3. Estructura del documento

El presente documento se encuentra organizado de la siguiente manera:

- **Capítulo 2: Marco Teórico.** En este capítulo se presenta una revisión de la literatura relevante sobre conceptos de Aprendizaje Automático (ML) y Aprendizaje Profundo (DL), así como estudios previos relacionados con la predicción de riesgo de accidentes de tránsito. También se describen los antecedentes de las arquitecturas presentadas en este trabajo, definiciones sobre el concepto de accidente y otras referencias bibliográficas esenciales para comprender el resto del trabajo.
- **Capítulo 3: Metodología.** Detalla el diseño y desarrollo del trabajo, indicando las decisiones tomadas en cada punto del mismo, desde la obtención de los datos hasta la visualización de las predicciones a través de la aplicación web.
- **Capítulo 4: Resultados.** Expone los principales resultados obtenidos durante la ejecución de experimentos y el diseño e implementación de la herramienta web.
- **Capítulo 5: Conclusiones.** Presenta las conclusiones del trabajo realizado, discutiendo los hallazgos más relevantes, las limitaciones del estudio y las posibles direcciones para trabajos futuros.

2

Marco Teórico y Antecedentes

Este capítulo presenta los fundamentos teóricos sobre los que se desarrolla el presente trabajo. El objetivo es proporcionar al lector los conceptos necesarios para comprender los métodos y técnicas utilizados en los capítulos posteriores. En la Sección 2.1, se introduce el concepto de accidentes de tránsito, su definición legal y estadísticas relevantes en Argentina. Luego, la Sección 2.2 presenta los conceptos básicos necesarios para trabajar con datos geoespaciales, mientras que la Sección 2.3 resume fundamentos de aplicaciones web relevantes para la implementación del sistema. A continuación, la Sección 2.4.4 introduce nociones de ingeniería de datos y procesos de Extracción, Transformación y Carga (*Extraction, Transformation, Loading*, ETL), relevantes para la construcción y mantenimiento del conjunto de datos. Posteriormente, en la Sección 2.4, se presentan los conceptos básicos de la Inteligencia Artificial (*Artificial Intelligence*, AI) y el ML, incluyendo las técnicas que se utilizan para entrenar y evaluar este tipo de modelos. Dicha sección concluye con el algoritmo *Random Forest* (Sección 2.4.5), utilizado como línea de base en este trabajo. Seguidamente, en la Sección 2.5 se profundiza en el campo del DL, presentando las arquitecturas de redes neuronales más relevantes para este trabajo: CNN, Red Neuronal Recurrente (*Recurrent Neural Network*, RNN) y Memoria de Largo-Corto Plazo (*Long Short-Term Memory*, LSTM). En la Sección 2.6, se introduce detalladamente la arquitectura *Transformer* y su aplicación en distintas disciplinas. Para concluir, la Sección 2.7 presenta una breve revisión del estado del arte en la predicción del riesgo de accidentes de tránsito utilizando técnicas de DL.

2.1. Accidentes de tránsito

Los accidentes de tránsito se ubican en el foco de este trabajo por su impacto significativo en la vida de las personas. Según la Organización Mundial de la Salud (OMS), comprenden la principal causa de muerte entre niños y adultos jóvenes (5-29 años), a pesar de que 9 de cada 10 víctimas fatales ocurren en el rango etario de 18 a 59 años [4].

2.1.1. Definición y clasificación según la Ley Nacional de Tránsito

La Ley de Tránsito N° 24.449 de Argentina define un accidente de tránsito como *todo hecho que produzca daño en personas o cosas como consecuencia de la circulación* [5]

Según la tipología más reciente definida por la Agencia Nacional de Seguridad Vial (ANSV), los accidentes de tránsito pueden clasificarse [6]:

- **Atropello a peatón o animal**: encuentro entre al menos un vehículo y un peatón o animal.
- **Choque**: siniestro vial que se produce entre un vehículo en movimiento y un objeto fijo, como puede ser infraestructura vial, elementos del entorno o elementos situados en la calzada.
- **Colisión**: situación en la que un vehículo contacta con otro, pudiendo alguno de ellos estar detenido o en movimiento. Según el sector en el que contactan los vehículos, pueden ser frontales, laterales o de alcance (impacto en la parte posterior).
- **Despeñamiento**: caída de un vehículo desde un lugar alto.
- **Despiste**: salida involuntaria de la calzada o trayectoria normal.

2.1.2. Estadísticas de Accidentes de Tránsito en Argentina

El Observatorio Vial, como parte de la Secretaría de Transporte de la Nación, reporta cada año estadísticas detalladas sobre accidentes a nivel nacional. Según la última publicación disponible, correspondiente al año 2024, se registraron 4 054 víctimas fatales por accidentes de tránsito, 159 de ellas en la provincia de Mendoza [7].

El informe de Agencia Nacional de Seguridad Vial [7] nos da una mirada detallada en cuanto a cuál es el tipo de accidente que predomina: 6 de cada 10 accidentes ocurren por una colisión (situación en la que un vehículo impacta contra otro, a diferencia de un choque, en el que un vehículo impacta con un objeto fijo), mientras que la mitad de los accidentes ocurren en rutas.

Otro dato importante tiene que ver con la distribución horaria: los períodos comprendidos entre las 6 y las 7 de la mañana y entre las 19 y las 21 horas son los momentos

del día en los que se producen la mayor cantidad de accidentes. Estos horarios coinciden con picos de comienzo y finalización de la jornada laboral, educativa y social. Además, en la mayoría de estos accidentes tiene como participantes a hombres jóvenes entre 15 y 34 años, y el 45 % de los mismos involucra a motocicletas.

Finalmente, la organización Luchemos por la Vida, dedicada a la prevención de accidentes de tránsito en Argentina, reporta que, mientras que países como Canadá, Holanda y Suecia han reducido la cantidad de víctimas fatales en al menos un 50 % en los últimos 35 años, Argentina ha visto una reducción de tan solo el 15 % en el mismo período [8]. A partir de la gráfica de la Figura 2.1, se puede apreciar que aún queda un largo camino por recorrer en materia de seguridad vial en el país, y que la aplicación de medidas preventivas es fundamental para lograrlo.

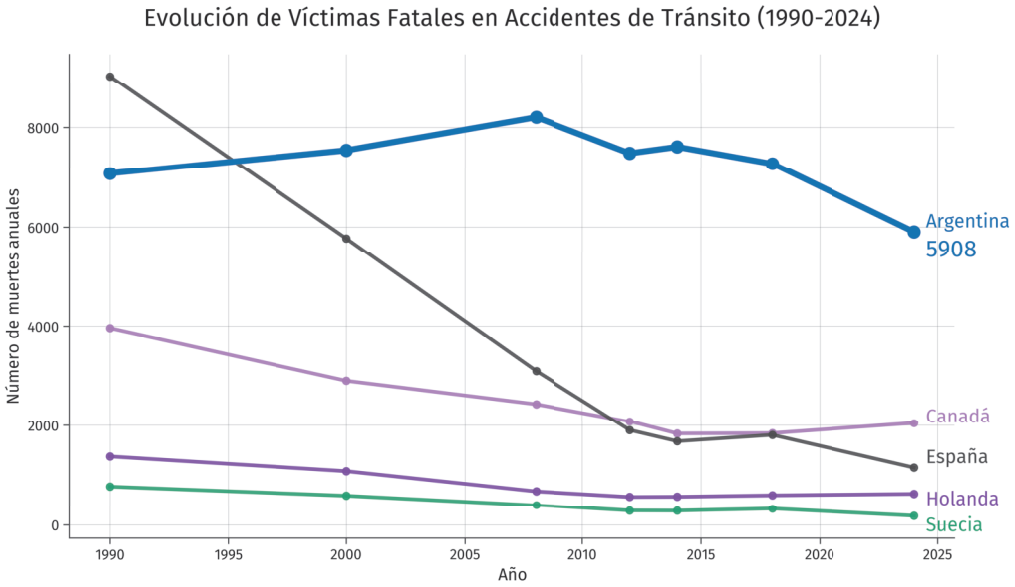


Figura 2.1: *Evolución de víctimas fatales en accidentes de tránsito en Argentina, Canadá, Holanda, Suecia y España (1990-2024).*

2.2. Sistemas de Referencia de Coordenadas

El análisis espacial de accidentes de tránsito requiere trabajar con datos geográficos que representan ubicaciones en la superficie terrestre. Para esto, es fundamental comprender cómo se representan y transforman las coordenadas geográficas mediante los Sistema de Referencia de Coordenadas (CRS).

2.2.1. Coordenadas geográficas y el sistema Sistema Geodésico Mundial 1984 (*World Geodetic System 1984*, WGS84)

El sistema de coordenadas geográficas más utilizado a nivel mundial es el Sistema Geodésico Mundial 1984 (*World Geodetic System 1984*, WGS84), identificado por el código

EPSG:4326. Este sistema representa puntos sobre la superficie terrestre mediante dos ángulos: la **latitud** (distancia angular al ecuador, entre -90° y 90°) y la **longitud** (distancia angular al meridiano de Greenwich, entre -180° y 180°).

WGS84 es el sistema de referencia estándar utilizado por los dispositivos GPS y por la mayoría de las fuentes de datos geográficos. Los datos de accidentes de tránsito obtenidos del sistema WebGIS de la Municipalidad de Mendoza se encuentran codificados en este sistema.

2.2.2. Necesidad de la transformación de coordenadas

A la hora de manipular los datos geográficos, se debe tener en cuenta cuál es el objetivo que se busca alcanzar con los mismos, y en función de eso elegir el sistema de referencia adecuado.

A pesar de que guardar datos en formato WGS84 es una práctica estándar y fácilmente comprensible, en este trabajo se requiere utilizar otras proyecciones

1. **Sistema Universal Transversal de Mercator (*Universal Transverse Mercator*, UTM)**: el sistema Universal Transversal de Mercator (UTM) divide la tierra en 60 zonas de 6° de longitud. Cada una de estas zonas presenta su propio sistema de coordenadas expresado en metro, por lo que este sistema de referencia se utiliza para realizar cálculos espaciales precisos.

2. **Proyección Web Mercator**: esta proyección, identificada por los códigos EPSG:3857 o EPSG:3395, es el estándar para mapas web interactivos. Se utiliza para poder representar el globo terrestre en un plano bidimensional para cualquier sección del mismo.

2.3. Fundamentos de Aplicaciones Web

Las aplicaciones web modernas se apoyan en un modelo cliente-servidor en el que el cliente (típicamente, un navegador) solicita recursos y servicios a un servidor mediante el Protocolo de Transferencia de Hipertexto (*Hypertext Transfer Protocol*, HTTP) [9]. Este esquema favorece la separación de responsabilidades entre la interfaz de usuario, la lógica de negocio y el acceso a datos, y permite exponer funcionalidades a través de una Interfaz de Programación de Aplicaciones (*Application Programming Interface*, API). En términos generales, una API establece un contrato de comunicación entre componentes de software, especificando las operaciones disponibles y el formato de los datos intercambiados.

2.3.1. REST

Un enfoque frecuentemente utilizado para el diseño de APIs es el estilo arquitectónico Transferencia de Estado Representacional (*Representational State Transfer*, REST), propuesto por Fielding [10]. En REST, la comunicación se organiza alrededor de *recursos* identificados mediante un Identificador Uniforme de Recursos (*Uniform Resource Identifier*, URI), y se emplean operaciones uniformes provistas por HTTP (por ejemplo, métodos como GET, POST, PUT o DELETE) para manipular representaciones de dichos recursos. Entre sus restricciones más relevantes se encuentran la separación cliente-servidor y la naturaleza *stateless* (sin estado) de las interacciones, lo que contribuye a la escalabilidad y a la simplicidad del sistema [11].

2.3.2. *Backend* y arquitectura en capas

El componente del lado del servidor, también conocido como *backend*, suele organizarse según una arquitectura en capas, separando responsabilidades como presentación (exposición de puntos de acceso o *endpoints* para que los clientes puedan consumirlos), lógica de negocio y acceso a datos [11]. En el contexto de una API basada en REST, estos *endpoints* representan rutas o puntos de acceso mediante los cuales el cliente interactúa usando HTTP para operar sobre *recursos*. Esta separación permite que el sistema sea más fácil de probar y mantener, al acotar el impacto de cambios dentro de cada capa.

2.3.3. *Frontend* y renderizado (CSR vs SSR)

El componente del lado cliente (*frontend*) es la porción de la aplicación que se ejecuta en el navegador y se apoya en estándares como Lenguaje de Marcado de Hipertexto (*HyperText Markup Language*, HTML) para estructurar el contenido, junto con Hojas de Estilo en Cascada (*Cascading Style Sheets*, CSS) y código que gestiona la interacción del usuario [12]. En este contexto, el *renderizado* refiere al proceso mediante el cual el estado y los datos de una aplicación se transforman en una representación visible de la interfaz para el usuario, generalmente a partir de la ejecución de código *JavaScript*. Esto se materializa en un documento HTML con estilos CSS que contiene los datos necesarios para el usuario. En aplicaciones web, este proceso puede realizarse principalmente en el cliente o bien delegarse, al menos en la primera entrega de contenido, al servidor.

En Renderizado del Lado del Cliente (*Client-Side Rendering*, CSR), el cliente recibe un documento inicial mínimo y construye la vista principalmente a partir de lógica ejecutada localmente, mientras que en Renderizado del Lado del Servidor (*Server-Side Rendering*, SSR) el servidor entrega HTML pre-renderizado. En este trabajo se adopta SSR con el objetivo de reducir la latencia percibida por el usuario y disminuir la cantidad de datos que se transmiten al cliente, de manera que se mejore el rendimiento en dispositivos de bajos recursos.

2.4. Inteligencia Artificial

La Inteligencia Artificial (*Artificial Intelligence*, AI) es un campo de la informática que se puede definir como el estudio y el diseño de aquellos sistemas que son capaces de realizar tareas que normalmente están relacionadas a la inteligencia del ser humano [13]. Esta definición, intencionalmente vaga, refleja el objetivo del campo de estudio, que se resume en crear máquinas con características humanas, como el razonamiento, la resolución de problemas, la comprensión del lenguaje natural y el aprendizaje.

La investigación en AI es un campo sumamente amplio, ya que el concepto puede aplicarse para resolver problemas en dominios muy diversos, abarcando desde áreas como la robótica [14], la visión por computadora [15], la planificación urbana [16], Procesamiento de Lenguaje Natural (*Natural Language Processing*, NLP) [17], entre otras. Para esto, se incorporan técnicas y enfoques de diversas disciplinas como la estadística, la lingüística y la psicología.

2.4.1. Aprendizaje Automático (*Machine Learning*, ML)

Una parte especialmente importante de la AI es el Aprendizaje Automático (*Machine Learning*, ML). Esta disciplina busca construir algoritmos que aprendan a partir de datos (ejemplos de cierto fenómeno). Estos ejemplos pueden ser obtenidos de la naturaleza, creados por un ser humano o diseñados por otro algoritmo [18].

Según el proceso que utilizan para aprender, los algoritmos de ML se pueden clasificar en tres grandes categorías: aprendizaje supervisado, no supervisado y por refuerzo.

El aprendizaje supervisado comprende aquellos algoritmos en los que se cuenta con un conjunto de datos etiquetados, es decir, cada ejemplo de entrenamiento tiene un valor asociado que indica la respuesta correcta. El objetivo del algoritmo es aprender una función que, para cualquier entrada (particularmente aquellas que nunca antes ha visto) mapee a la salida correcta.

El aprendizaje no supervisado, por otro lado, se refiere a aquellos algoritmos que trabajan con datos no etiquetados. En este caso, el fin del algoritmo es descriptivo: en lugar de asignar la etiqueta correcta a cada elemento de entrada, se busca describir cómo están organizados los datos, encontrando patrones o estructuras subyacentes [19].

Finalmente, en el aprendizaje por refuerzo el algoritmo aprende a partir de la interacción con un entorno. En este caso, se ve al sistema como un agente que toma acciones con el objetivo de maximizar una recompensa acumulada a lo largo del tiempo. A diferencia del aprendizaje supervisado, no se proporcionan datos etiquetados, sino que el agente debe descubrir qué acciones conducen a las mejores recompensas.

En este trabajo final de carrera, se lleva a cabo un enfoque de aprendizaje super-

visado para la clasificación multiclase: a partir de un conjunto de datos etiquetados que pueden pertenecer a más de dos clases (pero a un número finito de las mismas), se busca entrenar un modelo que sea capaz de predecir la clase correcta para nuevas muestras.

2.4.2. Entrenamiento de un modelo de ML

El entrenamiento de un modelo de ML implica encontrar aquellos parámetros del modelo que minimicen una función de pérdida, de manera que el mismo pueda generalizar el conocimiento obtenido a datos que no ha visto. Para esto, es necesario contar con un flujo de trabajo que permita obtener el mejor rendimiento posible, nos proporcione métricas de evaluación y garantice que el resultado final sea suficientemente bueno al evaluar datos nuevos.

En el contexto de este trabajo se trata con datos espacio-temporales. Cada muestra del conjunto de datos representa el riesgo de accidente de tránsito en una región de la ciudad de Mendoza para un día en particular. Esto quiere decir que el entrenamiento del modelo de ML corresponde a una tarea de series temporales, en la que cada muestra está asociada a un instante de tiempo y a una ubicación geográfica. A la hora de la evaluación y uso del modelo en un entorno real, el mismo se enfrentará a datos futuros, que no han sido vistos durante el entrenamiento.

División del conjunto de datos

Para llevar a cabo el entrenamiento, es necesario dividir el conjunto original en tres subconjuntos: conjunto de entrenamiento, conjunto de validación y conjunto de prueba. La idea básica es utilizar el conjunto de entrenamiento para ajustar los parámetros del modelo, el conjunto de validación para obtener métricas durante el entrenamiento (que permiten detectar problemas comunes en el modelo) y el conjunto de prueba para evaluar el rendimiento final del mismo.

Por lo general, se suele destinar entre el 60 % y el 80 % de los datos al conjunto de entrenamiento, entre el 10 % y el 20 % al conjunto de validación y el resto al conjunto de prueba [19], asignando cada muestra de forma aleatoria a uno de los tres conjuntos.

Existen técnicas como la validación cruzada que permiten aprovechar mejor los datos disponibles, utilizando distintas partes del conjunto de entrenamiento como conjunto de validación de forma alternada, lo que permite aprovechar al máximo los datos disponibles.

A la hora de trabajar con datos temporales, se debe ser mucho más cuidadoso para dividir el conjunto de datos: no se puede aplicar una división aleatoria para crear los conjuntos, ya que se perdería la secuencia temporal de los datos. En su lugar, se debe dividir el conjunto de datos en bloques temporales, utilizando los datos más antiguos para el entrenamiento y los datos más recientes para la validación y la prueba. De esta

forma, se garantiza que el modelo no pueda usar datos futuros, y las predicciones sean extrapolables a situaciones reales.

La Figura 2.2 muestra un ejemplo de cómo se puede dividir el conjunto de datos en bloques temporales, utilizando el 70 % de los datos más antiguos para el entrenamiento, el 15 % siguiente para la validación y el 15 % restante para la prueba.

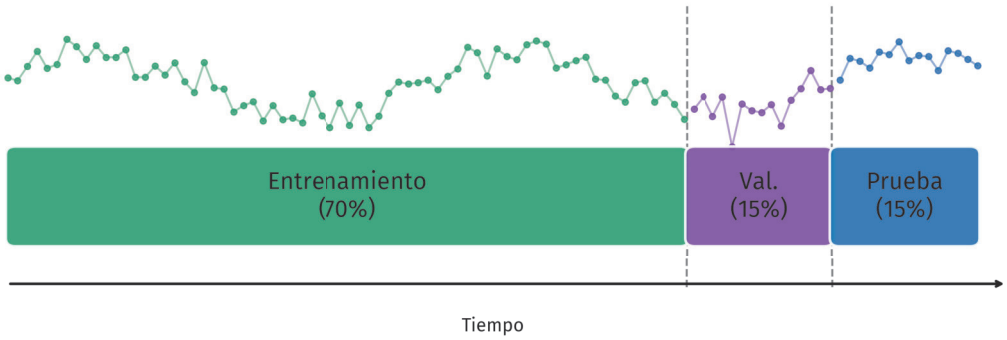


Figura 2.2: *Ejemplo de una división temporal del conjunto de datos en entrenamiento (70 %), validación (15 %) y prueba (15 %).*

Para el caso particular de los datos espacio-temporales, también es posible realizar una división espacial de los mismos, utilizando ciertas regiones geográficas para el entrenamiento y otras para la validación. Esto puede ser útil para evaluar la capacidad de generalización del modelo a nuevas regiones. En este trabajo se explora la división temporal como punto de partida, pero también se evalúa la división espacial en algunos experimentos.

La Figura 2.3 muestra un ejemplo de cómo se ve cada parte del conjunto de datos en la validación cruzada. La aplicación al dominio espacial es una extensión directa, en la que la división en *pliegues* o *particiones* se realiza en función de regiones geográficas en lugar de utilizar cualquier otro criterio (como el azar o la distribución temporal).

Datos desbalanceados

En muchos problemas del mundo real, la distribución de los datos en un problema de clasificación puede estar desbalanceada, es decir, algunas clases pueden tener muchas más muestras que otras. Este desbalance provoca que los modelos favorezcan a las clases mayoritarias, ya que al hacerlo pueden obtener una alta precisión sin necesidad de aprender a clasificar correctamente las clases minoritarias.

Para subsanar este problema, existen diversas técnicas que pueden aplicarse tanto a nivel de datos como a nivel de modelo.

A nivel de datos, se pueden aplicar técnicas como el sobremuestreo (*oversampling*) (aumentar la cantidad de muestras de las clases minoritarias) o el submuestreo (*undersampling*) (reducir la cantidad de muestras de las clases mayoritarias). También existen

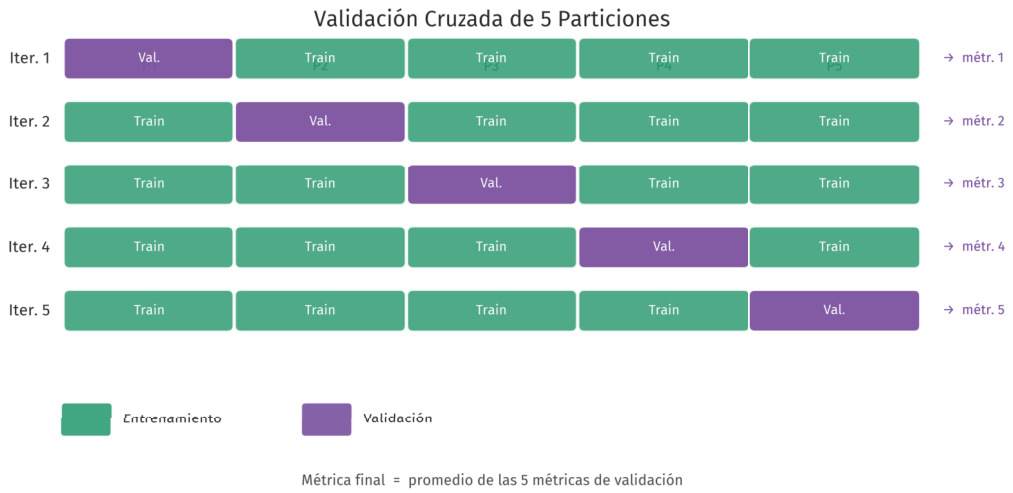


Figura 2.3: *Ejemplo de validación cruzada en un conjunto de datos.*

otra técnicas que crean datos sintéticos para las clases minoritarias, de manera que el conjunto de datos quede más balanceado.

Es evidente que este tipo de técnicas no pueden aplicarse directamente a datos temporales, ya que se perdería la secuencia temporal de los datos. En ese caso, podemos aplicar técnicas a nivel de modelo, como el uso de funciones de pérdida ponderadas, en las que se asigna un mayor peso a las clases minoritarias durante el entrenamiento. De esta forma, el modelo se ve incentivado a prestar más atención a estas clases.

2.4.3. Métodos de evaluación de modelos de ML

Una vez introducidos los conceptos básicos de ML, es importante definir cómo se evalúa el rendimiento de estos modelos. En este sentido, existen diversas métricas que permiten cuantificar la calidad de las predicciones realizadas por un modelo entrenado o comparar el rendimiento entre dos modelos diferentes. Esta sección se centra en las principales métricas utilizadas en tareas de clasificación.

Matriz de confusión

La matriz de confusión es una herramienta fundamental para medir el rendimiento de un modelo. Para un problema de clasificación multiclase, consiste en una tabla de dimensión $n \times n$, donde n es el número de clases. Cada fila de la matriz representa las instancias reales de una clase, mientras que cada columna representa las instancias predichas por el modelo (ver Figura 2.4). Los elementos de la matriz indican la cantidad de instancias que fueron clasificadas correctamente (diagonal principal) y las que fueron clasificadas incorrectamente (fuera de la diagonal).

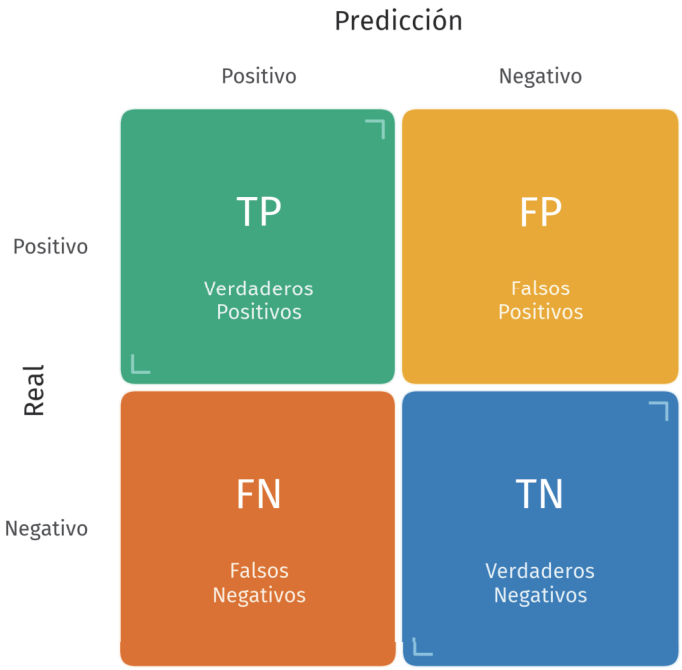


Figura 2.4: *Matriz de confusión para un problema de clasificación binaria.*

A partir de la matriz de confusión, se pueden definir los siguientes parámetros

- **Verdaderos Positivos (TP):** número de instancias correctamente clasificadas como positivas.
- **Falsos Positivos (FP):** número de instancias incorrectamente clasificadas como positivas.
- **Verdaderos Negativos (TN):** número de instancias correctamente clasificadas como negativas.
- **Falsos Negativos (FN):** número de instancias incorrectamente clasificadas como negativas.

Y con ellos, se pueden calcular diversas métricas de evaluación.

- **Exactitud (*Accuracy*)**: mide la proporción de predicciones correctas realizadas por el modelo. Se calcula como:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

Esta métrica es útil cuando las clases están balanceadas, pero puede ser engañosa en casos de clases desbalanceadas.

- **Sensibilidad** (***Recall***): mide la capacidad del modelo para identificar correctamente las instancias positivas. Se calcula como:

$$Recall = \frac{TP}{TP + FN} \tag{2.2}$$

El *recall* es especialmente importante en situaciones donde es crucial minimizar los falsos negativos.

- **Precisión** (***Precision***): mide la proporción de instancias clasificadas como positivas que son realmente positivas. Se calcula como:

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

Esta métrica es relevante cuando es importante minimizar los falsos positivos.

A partir de estas métricas, se pueden derivar otras más complejas

- **Puntuación F1** (***F1-Score***): es la media armónica entre la precisión y la sensibilidad. Se calcula como:

$$F1\text{-}Score = 2 \times \frac{Precision \cdot Recall}{Precision + Recall} \tag{2.4}$$

Esta métrica es especialmente útil cuando se busca un equilibrio entre precisión y sensibilidad.

- **MAP**: en el contexto de este trabajo, se utiliza para evaluar la calidad del ranking de riesgo asignado a las celdas de la grilla espacial. Se calcula como la media de la precisión promedio (AP) obtenida para cada unidad de tiempo. El AP se define como:

$$\text{AP} = \frac{1}{m} \sum_{k=1}^{n} P(k) \cdot \text{rel}(k) \tag{2.5}$$

Donde:

  - m es la cantidad total de muestras pertenecientes a la clase positiva en el conjunto evaluado.
  - n es el número total de muestras (instancias de la clase positiva y negativa).
  - $P(k)$ es la precisión calculada al considerar únicamente los primeros k elementos del ranking.
  - $rel(k)$ es una función indicadora que toma el valor 1 si la muestra en la posición k del ranking pertenece a la clase positiva, y 0 si pertenece a la clase negativa.

En el caso particular de la predicción del riesgo de accidentes de tránsito, las métricas más utilizadas son el *recall* (es importante detectar zonas de alto riesgo, a pesar

de que algunas zonas seguras sean clasificadas erróneamente como peligrosas) y MAP (para evaluar el rendimiento general del modelo).

La curva Característica Operativa del Receptor (*Receiver Operating Characteristic*, ROC) y el área bajo la curva

Muchos modelos de clasificación no producen directamente una etiqueta de clase, sino una probabilidad o puntuación continua a partir de la cual se deriva la predicción final mediante un umbral de decisión. Modificar dicho umbral altera el balance entre los tipos de error: reducirlo incrementa la sensibilidad (*recall*) a costa de aumentar los falsos positivos, mientras que elevarlo tiene el efecto contrario. La curva Característica Operativa del Receptor (*Receiver Operating Characteristic*, ROC) es una herramienta gráfica que permite visualizar simultáneamente ambos tipos de error para todos los valores de umbral posibles [20].

Formalmente, la curva ROC representa, para cada umbral posible, la **tasa de verdaderos positivos** (TPR) en el eje vertical, frente a la **tasa de falsos positivos** (FPR) en el eje horizontal:

$$\text{TPR} = \frac{TP}{TP + FN}, \qquad \text{FPR} = \frac{FP}{FP + TN} \tag{2.6}$$

La TPR coincide con el *recall* (Ecuación 2.2), mientras que la FPR es el complemento de la especificidad del clasificador. Un clasificador ideal, que no comete ningún error independientemente del umbral empleado, produce una curva que asciende verticalmente desde el origen hasta el punto $(0, 1)$ y luego avanza horizontalmente, abrazando la esquina superior izquierda del gráfico. En el extremo opuesto, un clasificador sin capacidad discriminatoria produce una curva que se aproxima a la diagonal principal, desde $(0, 0)$ hasta $(1, 1)$, representando un rendimiento equivalente al azar [20]. Un clasificador aleatorio, que asigna probabilidades de forma uniforme, se representaría exactamente por la diagonal (Área Bajo la Curva (*Area Under the Curve*, AUC) de $0{,}5$).

La Figura 2.5 muestra un ejemplo de curva ROC. La diagonal punteada representa el rendimiento de un clasificador aleatorio, mientras que la curva verde representa el rendimiento de un modelo entrenado. En este caso, el modelo entrenado tiene un rendimiento superior al azar, ya que su curva se encuentra por encima de la diagonal.

2.4.4. Ingeniería de datos y procesos ETL

En aquellos sistemas que consumen y producen información a partir de múltiples fuentes, la ingeniería de datos aborda el diseño, la construcción y el mantenimiento de flujos de datos (*data pipelines*) que permitan obtener los datos necesarios de manera confiable para el entrenamiento de modelos de ML. Esto involucra tareas como la ingesta, el modelado y la transformación de datos [21].

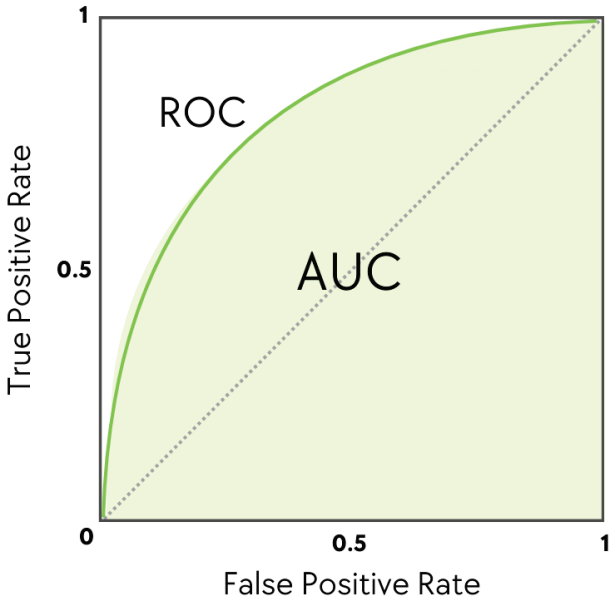


Figura 2.5: *Ejemplo de curva ROC. La diagonal punteada representa el rendimiento de un clasificador aleatorio, mientras que la curva verde representa el rendimiento de un modelo entrenado.*

En particular, cuando se construye un almacén de datos de este estilo, es habitual estructurar la preparación de datos mediante un proceso Extracción, Transformación y Carga (*Extraction, Transformation, Loading*, ETL). Dicho proceso comprende la extracción desde fuentes distintas fuentes de datos (por ejemplo, bases de datos transaccionales, o servicios web en forma de API), la transformación para estandarizar formatos, resolver inconsistencias, integrar entidades entre fuentes, y la carga del resultado en un repositorio destino. Además de la carga inicial, el ETL contempla el mantenimiento posterior y la actualización de los datos, lo cual suele representar una fracción sustantiva del esfuerzo total [22].

De acuerdo con Orallo et al. [22], un sistema ETL suele incluir, entre otras, las siguientes responsabilidades:

- **Extracción e incorporación de distintas fuentes**: lectura de datos transaccionales y adquisición de datos externos en formatos heterogéneos.
- **Limpieza, transformación e integración**: estandarización de medidas y fechas, tratamiento de valores faltantes, eliminación de redundancias y vinculación de registros que refieren a las mismas entidades.
- **Carga y mantenimiento**: definición del orden de carga para respetar restricciones de integridad y mecanismos de actualización (por ejemplo, cargas incrementales) para reflejar cambios.

En esta tesina final de carrera, estas responsabilidades resultan importantes para poder construir el conjunto de datos que alimenta a los modelos de ML a partir de fuentes

heterogéneas.

2.4.5. *Random Forest*

Un *árbol de decisión* es un modelo de ML que divide el espacio de características mediante reglas jerárquicas del tipo *si X <umbral, ve a la izquierda; si no, a la derecha*, asignando la predicción según la región en que cae cada instancia. Resultan intuitivos e interpretables, pero los árboles profundos tienden a sobreajustarse al conjunto de entrenamiento [20].

El algoritmo *Random Forest* mitiga este problema entrenando muchos árboles sobre subconjuntos aleatorios de los datos y combinando sus predicciones por mayoría de votos. Adicionalmente, en cada división se considera solo un subconjunto aleatorio de predictores, evitando que un único atributo dominante estructure todos los árboles. El resultado es un conjunto de árboles poco correlacionados cuyo promedio tiene menor varianza y mejor capacidad de generalización que cualquier árbol individual [20].

2.5. Deep Learning

El DL o aprendizaje profundo es una subárea del ML (Figura 2.6) que se enfoca en el desarrollo de modelos basados en redes neuronales artificiales con múltiples capas ocultas. Se conoce como aprendizaje *profundo* porque estas redes aplican muchas *capas* de transformaciones a los datos.

Los modelos de DL se han vuelto extremadamente populares en los últimos años, con diversas variantes en su arquitectura que han alcanzado grandes resultados en todo tipo de dominios [15] [17] [23]. Más allá del rendimiento alcanzado, hay otros motivos que explican su popularidad, como la capacidad de estos modelos de disminuir el trabajo de ingeniería de características que se realizaba manualmente en los enfoques tradicionales de ML, su habilidad para manejar volúmenes de datos gigantescos y la posibilidad de aprovechar al máximo los avances en computación paralela y *hardware* especializado [24].

2.5.1. El concepto de neurona

Una neurona artificial es la unidad básica de procesamiento de una red neuronal. Su funcionamiento se inspira en las neuronas biológicas del cerebro humano (Figura 2.7), aunque su implementación es mucho más simple. Matemáticamente, una neurona está compuesta por [20]:

- conexiones que llegan a ella a partir de otras neuronas, cada una con un peso asociado que indica la importancia de esa conexión

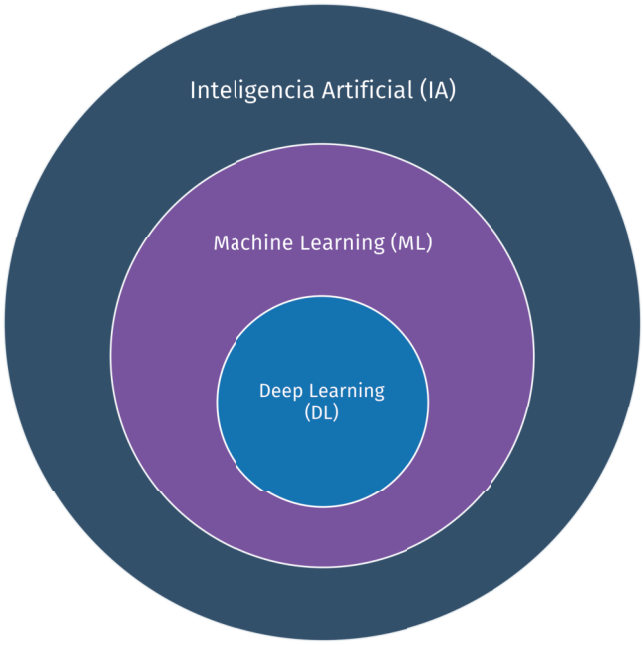


Figura 2.6: *Relación jerárquica entre Inteligencia Artificial, Machine Learning y Deep Learning.*

- un sesgo, que es un valor adicional que permite ajustar la salida de la neurona
- una función de activación g (ver Sección 2.5.3), que determina si la neurona se activa o no (y en qué nivel) en función de la suma ponderada de sus entradas más el sesgo

Dada una neurona j, que recibe señales de las neuronas $i_1 \dots i_n$, su salida o_j se calcula de la siguiente manera [13]:

$$o_j = g\left(\sum_{k=1}^{n} w_{i_k j} \cdot o_{i_k} + b_j\right) \tag{2.7}$$

Donde o_{i_k} es la salida de la neurona i_k, $w_{i_k j}$ es el peso de la conexión entre las neuronas i_k y j, b_j es el sesgo de la neurona j y g es la función de activación.

2.5.2. Redes Neuronales

Las Redes Neuronales Artificiales (*Artificial Neural Networks*, ANN) utilizan neuronas artificiales interconectadas de forma estructurada para modelar y resolver problemas complejos. Estas redes están organizadas en capas, donde cada capa contiene múltiples neuronas que procesan la información de manera paralela. La arquitectura básica de una ANN consta de una capa de entrada, una o más capas ocultas y una capa de salida, como se muestra en la Figura 2.8.

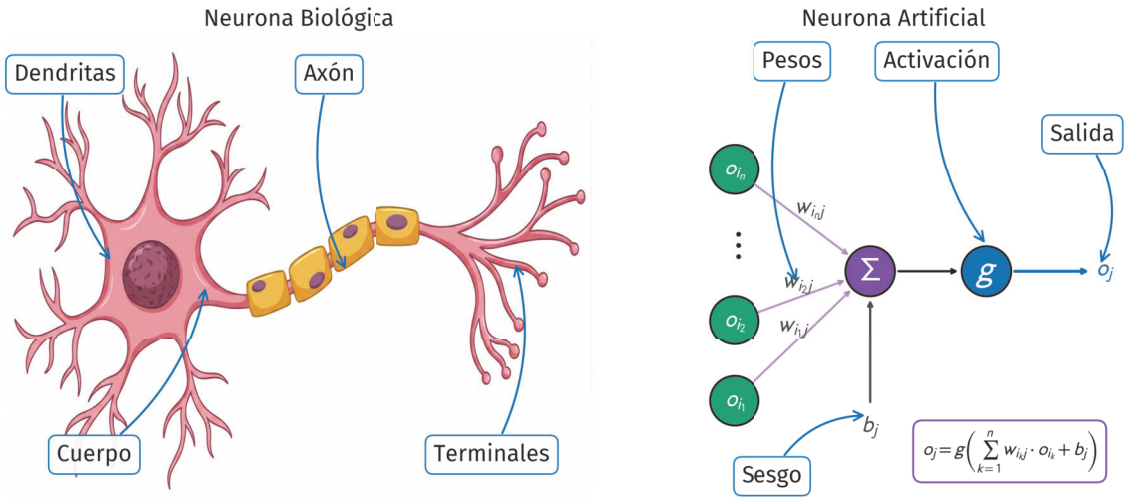


Figura 2.7: *Comparación entre una neurona biológica y una neurona artificial.*

- **Capa de entrada:** Es la primera capa de la red y recibe los datos de entrada. Cada neurona en esta capa representa una característica.
- **Capas ocultas:** Son las capas intermedias entre la capa de entrada y la capa de salida. Estas capas realizan la mayor parte del procesamiento y transformación de los datos. Una ANN puede tener múltiples capas ocultas, y cada una puede tener un número variable de neuronas.
- **Capa de salida:** Es la última capa de la red y produce el resultado final del modelo. El número de neuronas en esta capa depende del tipo de tarea que se esté realizando (clasificación binaria, clasificación multiclase, regresión, etc.).

Este tipo de redes son conocidas como *fully connected*, ya que cada neurona de una capa está conectada a todas las neuronas de la capa siguiente. Una ANN compuesta solo por capas *fully connected* también se conoce como Perceptrón Multicapa (*Multi-Layer Perceptron*, MLP).

2.5.3. Funciones de activación

Las funciones de activación son funciones matemáticas que se aplican a la señal que sale de cada neurona, permitiendo al modelo capturar relaciones no lineales entre las entradas y las salidas. Algunas de las funciones de activación más comunes son:

Función Unidad Lineal Rectificada (*Rectified Linear Unit*, ReLU)

La función ReLU es una de las funciones de activación más utilizadas en capas ocultas de redes neuronales debido a su simplicidad y efectividad. Se define como:

$$\text{ReLU}(x) = \max(0, x) \tag{2.8}$$

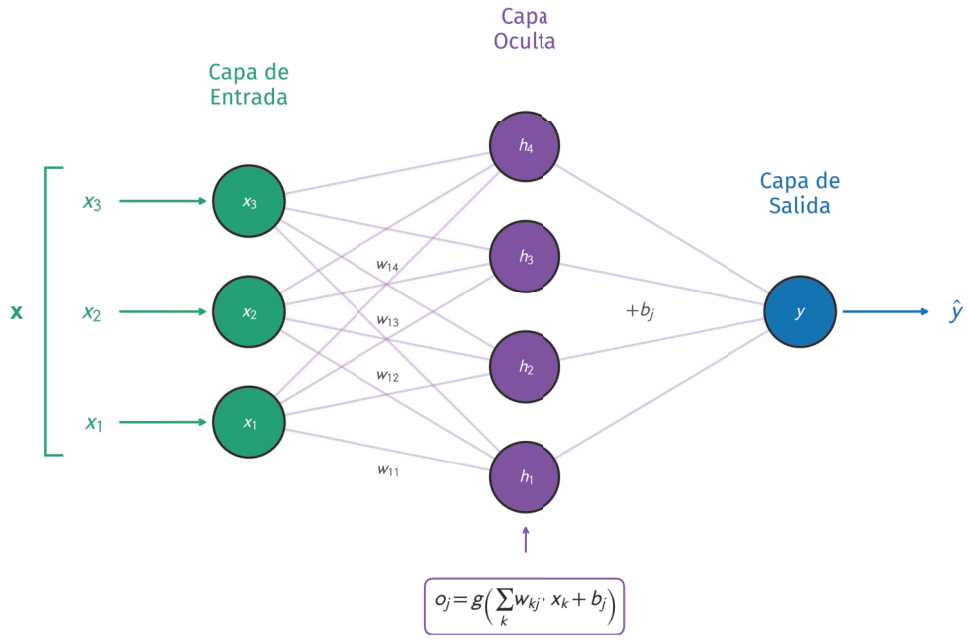


Figura 2.8: *Arquitectura básica de una red neuronal con una capa de entrada, una capa oculta y una capa de salida.*

Función Softmax

La función Softmax se utiliza comúnmente en la capa de salida de redes neuronales para tareas de clasificación multiclase. Convierte un vector de valores en un vector de probabilidades, donde cada valor representa la probabilidad de pertenecer a una clase específica. Se define como:

$$\text{Softmax}(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{2.9}$$

donde x_i es el valor de la neurona i y la suma en el denominador se realiza sobre todas las neuronas de la capa de salida.

2.5.4. Propagación hacia atrás y actualización de pesos

A partir de los conceptos presentados hasta ahora, se puede generar una idea de la forma en la que las redes neuronales aprenden. Si se realiza una composición de cada una de las funciones de las neuronas (definidas en la Ecuación 2.7), se puede obtener una fórmula completa que describe la salida de la red para un ejemplo de entrada.

$$y = g\left(\sum_{k=1}^{n} w_{i_k j} \cdot o_{i_k} + b_j\right) \tag{2.10}$$

Donde y es la salida de la red para un ejemplo de entrada, g es la función de activación, $w_{i_k j}$ son los pesos de las conexiones entre las neuronas, o_{i_k} son las salidas de las

neuronas de la capa de entrada y b_j es el sesgo de la neurona j.

Para llevar a cabo el entrenamiento, se aplica la función de pérdida a la salida de la red para obtener un valor que mida la discrepancia entre la predicción y el valor real. A partir de este valor, se puede calcular la derivada de la función de pérdida con respecto a los pesos de la red (el gradiente) y utilizar este valor para actualizar los pesos de manera que la predicción sea más precisa.

Ya que este proceso de actualización se lleva a cabo desde las capas más profundas (más cercanas a la salida) hacia las capas más superficiales (más cercanas a la entrada), se conoce como **propagación hacia atrás**.

2.5.5. Regularización y el concepto de *dropout*

Un escenario común a la hora de entrenar redes neuronales es el sobreajuste, que ocurre cuando el modelo aprende a memorizar los datos de entrenamiento en lugar de generalizar a datos nuevos. Para mitigar este problema, se pueden aplicar diversas técnicas de *regularización*, que buscan penalizar la complejidad del modelo o introducir ruido durante el entrenamiento para fomentar la generalización.

En este trabajo final de carrera, se utiliza la técnica de *dropout*, que consiste en apagar aleatoriamente un porcentaje de neuronas durante el entrenamiento. Esto obliga a la red a no depender demasiado de ninguna neurona en particular, lo que puede ayudar a mejorar la capacidad de generalización del modelo [24]. La Figura 2.9 muestra un ejemplo de cómo se ve una red neuronal antes y después de la aplicación del *dropout*.

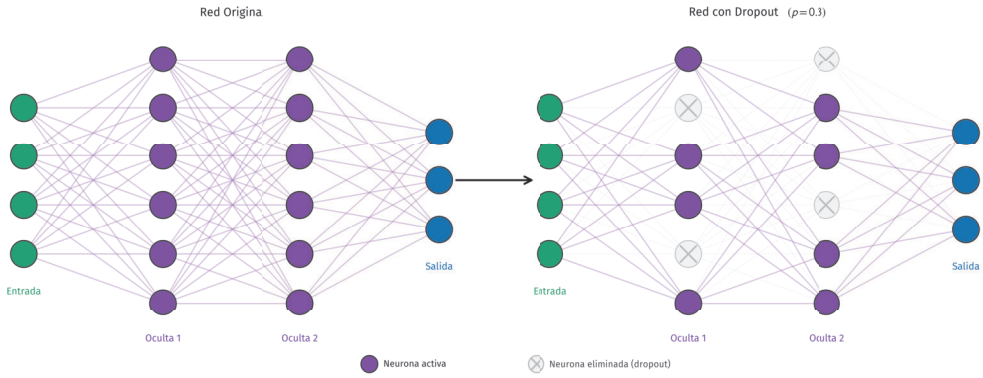


Figura 2.9: *Ejemplo de una red neuronal antes y después de la aplicación del dropout.*

Por lo general, el *dropout* se aplica solo a las capas ocultas de la red y se desactiva durante la fase de evaluación y su posterior uso para inferencia. Sin embargo, existen casos en los que se puede utilizar el *dropout* también durante la inferencia, particularmente en la técnica conocida como *Monte Carlo Dropout* [25]. La intuición detrás de esta técnica es que, al mantener el *dropout* activo durante la inferencia, se pueden obtener múltiples predicciones para una misma entrada, lo que permite estimar la incertidumbre del mo-

delo en sus predicciones. A partir de estos experimentos, se puede calcular la media y la varianza de las predicciones, lo que proporciona una medida de confianza en la salida del modelo.

2.5.6. Red Neuronal Convolucional (CNN)

Este tipo de redes neuronales está especialmente diseñada para procesar datos con una estructura de cuadrícula, como imágenes. Utilizan operaciones de convolución para extraer **características locales** de los datos de entrada, lo que les permite identificar patrones espaciales de una forma resistente a transformaciones como traslaciones o escalados [24].

Además de las capas *fully connected* presentes en una red neuronal tradicional, las CNN suelen incluir capas convolucionales y de *pooling*, como se puede apreciar en la Figura 2.10.

- **Capas Convolucionales**: son las que caracterizan a las CNN. Estas capas aplican filtros sobre la entrada para extraer características locales. Cada filtro se desplaza sobre la entrada, realizando una operación de convolución que produce un mapa de características. Estos mapas resaltan patrones específicos en los datos, como bordes, texturas o formas.
- **Capas de *Pooling***: estas capas se utilizan para reducir la dimensionalidad de los mapas de características generados por las capas convolucionales. El *pooling* ayuda a disminuir la cantidad de parámetros y cálculos en la red, además de proporcionar cierta invarianza a pequeñas transformaciones en los datos de entrada. Suelen aplicarse entre dos capas convolucionales.

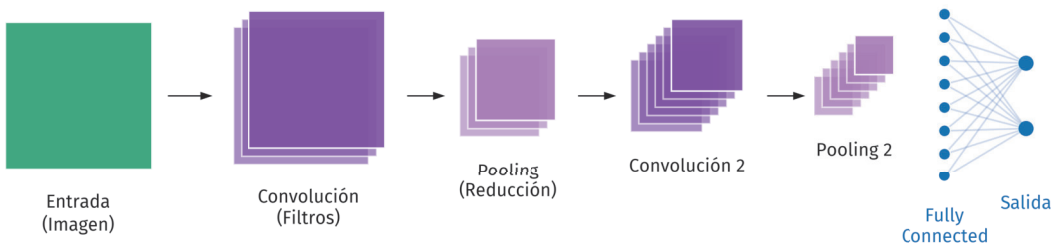


Figura 2.10: *Arquitectura de una Red Neuronal Convolucional (CNN)*

2.5.7. Red Neuronal Recurrente (RNN)

Las Redes Neuronales Recurrentes o RNN surgen como respuesta a la necesidad de modelar datos secuenciales, donde, idealmente, el modelo debería recordar información relevante de los datos de entrada anteriores para tomar decisiones informadas en

el presente. Este tipo de redes presenta mecanismos de retroalimentación a través de los cuales la salida producida por una entrada influye en el procesamiento de la próxima entrada.

Intuitivamente, las RNN necesitan el doble de parámetros entrenables que una red neuronal tradicional, ya que utilizan un conjunto de pesos que se aplica a la entrada inmediata y otro conjunto diferente que se aplica a la señal proveniente de pasos temporales anteriores.

El procesamiento de secuencias tiene aplicaciones en una amplia variedad de dominios, aunque la investigación central del área ha estado históricamente en el procesamiento del lenguaje natural.

Actualmente, las RNN han sido en gran medida reemplazadas por arquitecturas más avanzadas como los *Transformers*, que han demostrado al mismo tiempo mayor rendimiento y mayor capacidad de aprovechar el procesamiento paralelo: en un estudio comparativo llevado a cabo por Karita et al. [26], el modelo basado en *Transformers* superó a las RNN en 13/15 escenarios, e igualó su rendimiento ocho veces más rápido.

Un modelo base de RNN consta de una capa de entrada, una o más capas ocultas recurrentes y una capa de salida. En cada paso temporal, la red recibe una entrada y actualiza su estado interno (memoria) en función de la entrada actual y del estado anterior. Este estado interno se utiliza para generar la salida en cada paso temporal.

Las ANN que **no** cuentan con conexiones recurrentes se conocen como *feed-forward*.

2.5.8. Memoria de Largo-Corto Plazo (*Long Short-Term Memory*, LSTM)

Luego de su introducción, las RNN demostraron ser efectivas en una variedad de tareas, pero sus limitaciones técnicas se hicieron evidentes rápidamente. El principal inconveniente al aprender dependencias a largo plazo es el problema del desvanecimiento y explosión del gradiente. Trabajos como el de Bengio et al. [27] señalaron que las RNN tradicionales no son robustas frente al ruido y carecen de eficiencia para capturar relaciones en secuencias extensas.

Para mitigar esto, se propuso la arquitectura LSTM [28], la cual introduce un diseño de celda basado en un estado de memoria interno y mecanismos de control denominados puertas. Estas puertas regulan el flujo de información, permitiendo que la red decida qué datos conservar o descartar en cada paso temporal [24].

Como se observa en la Figura 2.11, la celda opera mediante tres componentes principales:

- **Puerta de olvido** (***forget gate***): Determina qué información del estado anterior (c_{t-1}) se descarta. Una función sigmoide procesa la entrada actual (x_t) y el estado

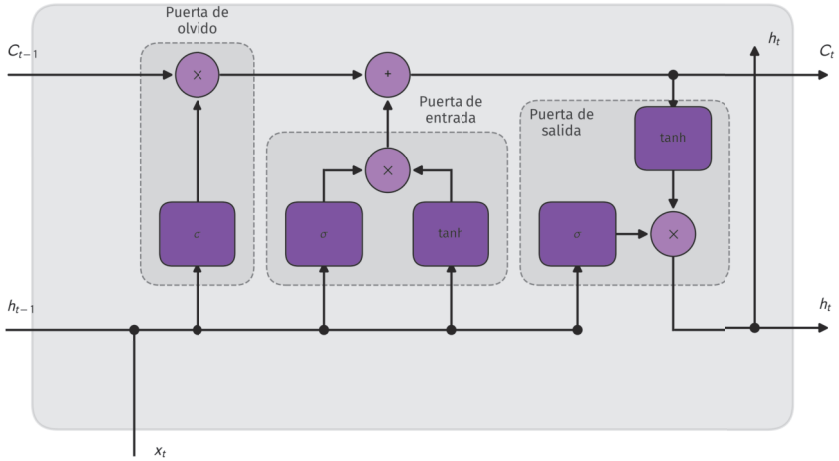


Figura 2.11: *Estructura interna de una celda LSTM y sus mecanismos de control.*

previo (h_{t-1}), generando valores entre 0 (olvido total) y 1 (conservación total).

- **Puerta de entrada** (***input gate***): Regula la incorporación de nueva información al estado de la celda. Combina una capa sigmoide, que decide qué valores actualizar, con una capa *tanh* que genera un vector de valores candidatos para ser sumados a la memoria.

- **Puerta de salida** (***output gate***): Decide qué parte del estado de la celda actualizada se exportará como el nuevo estado (h_t). El estado de la celda se normaliza mediante una función *tanh* y se filtra por una compuerta sigmoide, emitiendo únicamente la información relevante para el paso actual.

2.5.9. Consideraciones a la hora de entrenar modelos de DL

Para entrenar modelos basados en ANN, existe un número adicional de consideraciones que deben ser tenidas en cuenta, además de las que se describen en la sección 2.4.2.

En primer lugar, se introduce el concepto de *epoch*. Una *epoch* de entrenamiento se refiere a una iteración completa del conjunto de entrenamiento, donde cada muestra del conjunto de entrenamiento es utilizada exactamente una vez. Este parámetro puede combinarse con el de tamaño de lote (*batch size*), que define el número de muestras que se procesan simultáneamente. Si se utiliza un tamaño de lote mayor a 1, el modelo calcula el gradiente promedio de las muestras del lote y actualiza los parámetros del modelo en función de este gradiente promedio. Una ventaja de utilizar entrenamiento por lotes es la paralelización: ya que todas las muestras del lote se procesan en paralelo, el entrenamiento puede ser acelerado. Otra ventaja que presenta es que, ya que el gradiente se promedia entre todo el lote, el modelo puede converger más rápidamente al verdadero

gradiente del conjunto de datos [24]. Sin embargo, mientras más grande sea el tamaño de lote, los pesos de la red neuronal se actualizan menos veces por *epoch*. De este modo, se debe encontrar un equilibrio entre el tamaño de lote y el número de *epochs* para lograr una convergencia óptima [24].

Una forma de limitar la cantidad de *epochs* para prevenir que el modelo sobreajuste es mediante la detención temprana (*early stopping*). Esta técnica utiliza dos parámetros para detener el entrenamiento de forma anticipada si el desempeño del modelo en el conjunto de validación deja de mejorar. El primer parámetro es el umbral de mejora, que define el porcentaje mínimo de mejora que se debe lograr en cada *epoch* para continuar con el entrenamiento. El segundo parámetro, conocido como paciencia, define el número de *epochs* a esperar antes de detener el entrenamiento.

Por último, se introduce el concepto de tasa de aprendizaje (*learning rate*). Este parámetro define la magnitud de los cambios que se realizan en los parámetros del modelo durante el entrenamiento. Un valor más alto de la tasa de aprendizaje puede conducir a una convergencia más rápida, pero también puede conducir a oscilaciones en el desempeño del modelo. Por otro lado, un valor más bajo de la tasa de aprendizaje puede conducir a una convergencia más estable, pero también puede conducir a una convergencia más lenta [13].

2.6. *Transformers*

La arquitectura *transformer* fue introducida en 2017 [29] y representa un gran cambio en la estructura de redes neuronales que había estado dominando el campo hasta entonces. Desde la introducción de LSTM, la mayoría de los avances en el campo del DL se habían dado por el aumento en la capacidad de cómputo y de datos disponibles, pero la arquitectura subyacente permanecía relativamente sin cambios.

Aunque originalmente los *transformer* fueron propuestos para realizar traducción automática, han demostrado ser extremadamente efectivos en una amplia variedad de tareas, de forma tal que hoy en día dominan el campo del procesamiento del lenguaje natural y han sido adaptados con éxito a otros dominios como la visión por computadora y el procesamiento de series de tiempo en general [30]. Actualmente, los *transformer* se han vuelto la arquitectura predominante en el campo del DL, alcanzando popularidad masiva por sus aplicaciones a los Grandes Modelos de Lenguaje (*Large Language Model*, LLM).

El principal motivo por el que los *transformer* han superado a las RNN en la mayoría de las tareas es su capacidad para aprovechar el procesamiento paralelo. A diferencia de las RNN, que procesan secuencias de manera secuencial, los *transformer* pueden procesar todos los elementos de la secuencia simultáneamente y así aprovechar el cómputo en Unidad de Procesamiento Gráfico (*Graphics Processing Unit*, GPU).

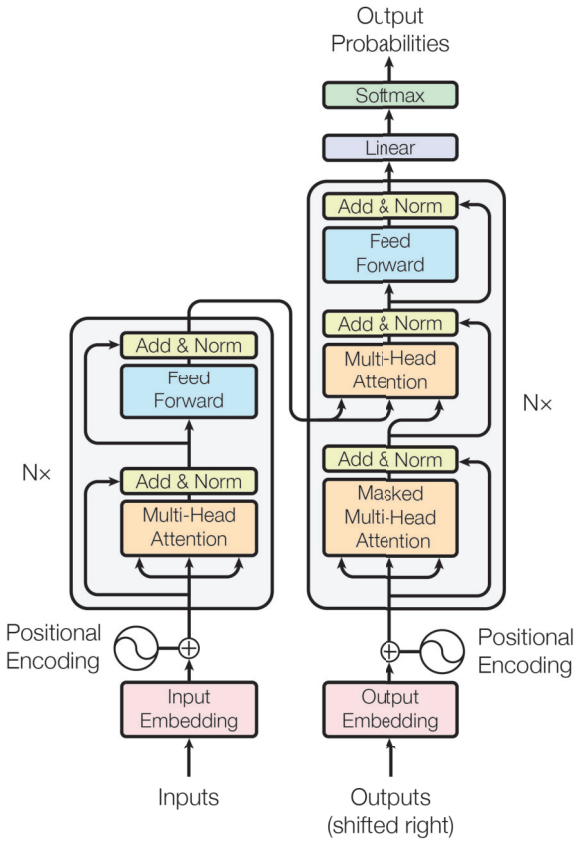


Figura 2.12: *Arquitectura Transformer canónica propuesta por Vaswani et al.* [29]

2.6.1. *Tokens*

Un *token* es la unidad básica de entrada y salida de un modelo de ML que procesa secuencias (como puede ser un *transformer* o LSTM), y tienen su origen en el campo del procesamiento del lenguaje natural. En este contexto, un *token* se considera una unidad atómica e indivisible de texto, que puede ser una palabra, un subpalabra o incluso un carácter individual, dependiendo del método de *tokenización* utilizado. Sin embargo, lo que un *token* representa es una decisión de diseño.

Ya que los modelos de DL trabajan con datos numéricos, la importancia de los *tokens* radica en su capacidad para transformar datos discretos (como texto) en representaciones numéricas que los modelos pueden procesar. Por lo tanto, el paso inicial a la hora de trabajar con secuencias es la *tokenización*, que consiste en convertir los datos de entrada en una secuencia de *tokens* a través de un diccionario o *lookup-table*, de manera que la operación pueda revertirse a la salida del modelo para obtener una salida que pertenezca al dominio original [24].

2.6.2. *Embeddings*

Los embeddings son representaciones vectoriales densas de datos discretos, como palabras, frases o incluso elementos no textuales como imágenes. Estas representacio-

nes permiten a los modelos de DL capturar relaciones semánticas y sintácticas entre los elementos, facilitando el aprendizaje y la generalización [31].

Cada token de la secuencia de entrada se mapea a un vector de tamaño d_{emb} a través de una matriz de $vocab\_size \times d_{emb}$, donde *vocab_size* es la cantidad de *tokens* únicos en el vocabulario del modelo. La construcción de estos embeddings puede verse como una capa adicional en la red neuronal o como una tarea de preentrenamiento que se realiza por separado (práctica habitual cuando se trabaja con LLM).

Cuando se trabaja en entornos de alta dimensionalidad, los embeddings son especialmente útiles, ya que permiten reducir la dimensionalidad de los datos de entrada sin perder información relevante.

2.6.3. *Encodings* posicionales

Considerando que los *transformers* procesan los elementos de la secuencia en paralelo, es necesario introducir alguna forma de información posicional para que el modelo pueda distinguir entre diferentes posiciones en la secuencia. Esto se logra mediante el uso de *encodings* posicionales, que son vectores que se suman a las representaciones de los elementos de la secuencia para incorporar información sobre su posición relativa.

Existen diferentes formas de generar estos *encodings* posicionales. Una de las más comunes es utilizar funciones trigonométricas (seno y coseno) [29] para crear vectores que varían de manera periódica con la posición en la secuencia. Otra opción es utilizar *embeddings* aprendidos, en los que los vectores posicionales se entrenan junto con el resto del modelo.

La implementación de *encodings* posicionales utilizando funciones trigonométricas puede definirse de la siguiente manera:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{emb}}}\right) \tag{2.11}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{emb}}}\right) \tag{2.12}$$

donde pos es la posición del elemento en la secuencia, i es la dimensión del vector de *encoding*, y d_{emb} es la dimensión de los embeddings del modelo.

Para cada *embedding* de entrada, se suma el vector de *encoding* posicional correspondiente antes de ser procesado por el mecanismo de atención.

2.6.4. Atención

El mecanismo de atención es el componente central de la arquitectura *transformer*. Permite al modelo asignar diferentes pesos a diferentes partes de la secuencia de entrada

al procesar cada elemento, lo que le permite enfocarse en las partes más relevantes para la tarea en cuestión [24].

De manera matemática, la atención se calcula utilizando una función de puntuación, que mide la relevancia de cada elemento de la secuencia con respecto a los demás. Esta puntuación se utiliza para asignar pesos a cada elemento, que luego se utilizan para calcular una representación ponderada de la secuencia de entrada.

Para esto, el *transformer* utiliza un mecanismo conocido como *self-attention* o autoatención, que permite al modelo considerar todas las partes de la secuencia de entrada al procesar cada elemento.

Este mecanismo se construye utilizando tres matrices de proyección lineal: *queries*, *keys* y *values*. Las *queries* representan la información que se busca en la secuencia, las *keys* representan la información disponible en la secuencia, y las *values* representan la información que se utilizará para generar la salida.

Cuando se procesa un elemento de la secuencia, se calcula la atención comparando la *query* de ese elemento con las *keys* de todos los demás elementos en la secuencia. Esto produce una serie de pesos que indican qué tan relevante es cada elemento para el elemento actual [32]. Estos pesos se utilizan luego para ponderar las *values* correspondientes y generar una representación contextualizada del elemento actual. La fórmula más popular para calcular la atención (utilizando el producto escalar) es la siguiente [29]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2.13}$$

Donde Q, K y V son las matrices de *queries*, *keys* y *values* respectivamente, y d_k es la dimensión de las *keys*. La normalización por $\sqrt{d_k}$ ayuda a estabilizar los gradientes durante el entrenamiento.

A la hora de calcular la atención, los *transformer* suelen utilizar múltiples cabezas de atención (*multi-head attention*), que permiten al modelo capturar diferentes tipos de relaciones entre los elementos de la secuencia (por ejemplo, relaciones a corto y largo plazo) de manera simultánea. Cada cabeza realiza su propio cálculo de atención, y los resultados se concatenan y se proyectan nuevamente para formar la salida final del mecanismo de atención (Figura 2.13).

Además, existen diferentes tipos de arquitecturas de *transformers*, que en gran medida definen cómo actúa el mecanismo de atención.

- **Encoder-Only**: Crean una representación rica de la secuencia de entrada. Para esto, utilizan un mecanismo de atención bidireccional en el que cada elemento de la secuencia puede atender a todos los demás elementos, tanto anteriores como posteriores. Esta arquitectura es adecuada para tareas de comprensión del lenguaje

(BERT) [33], Reconocimiento de Entidades Nombradas (NER) entre otras.

- **Decoder-Only**: Procesan información de izquierda a derecha para predecir el siguiente elemento de una secuencia. Utilizan un mecanismo de atención causal, en el que cada elemento solo puede atender a los elementos anteriores en la secuencia. Se utiliza principalmente para tareas generativas [17]. Un *transformer* con una arquitectura *decoder-only* es equivalente a una RNN multi-estado [34].
- **Encoder-Decoder**: Combina tanto las capas de codificación como las de decodificación del *transformer*. El *encoder* procesa la secuencia de entrada completa de forma bidireccional, y posteriormente el *decoder* genera la secuencia de salida de forma autoregresiva. Esta es la arquitectura original propuesta en Vaswani et al. [29]. Hoy en día, se utiliza menos que las dos anteriores.

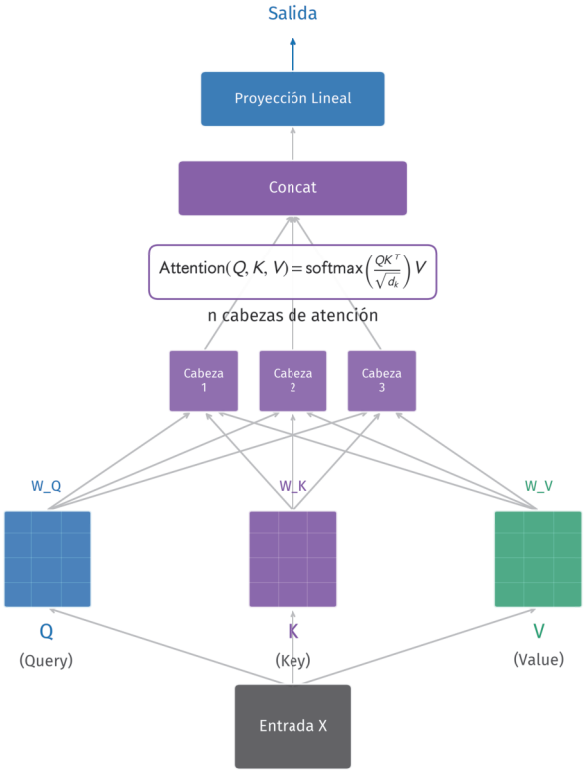


Figura 2.13: *Mecanismo de atención multi-cabeza y las matrices Query (Q), Key (K) y Value (V).*

2.6.5. *Vision Transformer* (ViT)

Los ViT adaptan la arquitectura *transformer* al dominio de la visión por computadora. En lugar de procesar secuencias de texto, los ViT dividen una imagen en parches (pequeñas regiones cuadradas) y tratan cada uno de estos parches como un token, de manera análoga a las palabras en un modelo de NLP. [15]

El modelo parte de una imagen de tamaño $H \times W \times C$ (altura, ancho y cantidad de canales) y la divide en parches de tamaño $p \times p$, donde p es un parámetro que debe ser

definido antes del entrenamiento.

Esto da como resultado $N = \frac{HW}{p^2}$ parches. Cada uno de estos se aplana y se proyecta linealmente a un espacio de dimensión D para crear una secuencia de embeddings.

A esta secuencia se le añade un token especial de clasificación (CLS) al inicio [33]. El estado de este token después de pasar por las capas del *transformer* se utiliza para realizar la clasificación final de la imagen.

Esta arquitectura ha demostrado resultados competitivos e incluso superiores a las CNN tradicionales en tareas de clasificación de imágenes, especialmente cuando se entrena con grandes cantidades de datos.

2.7. Deep Learning para la predicción espacio-temporal del riesgo de accidentes de tránsito

Debido al gran impacto social y económico de los accidentes de tránsito, la predicción y prevención de los mismos ha sido un área de investigación relevante desde la década de 1990. Originalmente, se modelaba el problema a través de métodos estadísticos como distribuciones de Poisson o modelos binomiales negativos [35] [36].

Con el avance de las técnicas de ML y DL, y la disponibilidad de mayores conjuntos de datos, se han desarrollado enfoques más sofisticados para abordar este problema.

Si nos enfocamos particularmente en la tarea de predicción tanto espacial como temporal del riesgo de accidentes de tránsito (es decir, predecir en qué zonas y en qué momentos es más probable que ocurran accidentes), podemos encontrar diversos enfoques en la literatura.

En Yuan et al. [37] se propone un modelo basado en ConvLSTM [38]. Una red ConvLSTM es una extensión de la arquitectura LSTM diseñada específicamente para modelar datos con dependencias tanto espaciales como temporales. Su principal innovación radica en que sustituye los productos matriciales de una LSTM convencional (que operan sobre vectores unidimensionales) por operadores de convolución. Este trabajo puede considerarse el primero que busca predecir mapas de riesgo de accidentes de tránsito que varíen en el tiempo según las condiciones. Los autores utilizan datos de accidentes, de precipitaciones, de volumen de tráfico y de estructura de la red vial para alimentar el modelo con datos de todo el estado de Iowa, en Estados Unidos. La principal innovación que presenta este trabajo es la presencia de un ensamble espacial de modelos ConvLSTM. La tarea de predicción se aplica sobre el área de un estado en lugar de una ciudad, por lo que se divide el área de estudio en subregiones y se entrena un modelo ConvLSTM independiente para cada una de ellas, teniendo en cuenta las diferentes características (principalmente de densidad poblacional) de cada subregión. Finalmente, se combinan las predicciones de cada modelo para generar un mapa de riesgo completo.

En Jin et al. [39] se presenta un modelo híbrido que combina CNN y ANN jerárquicas para llevar a cabo dos tareas en simultáneo: la predicción del riesgo de accidentes en una grilla y la clasificación del tipo de accidente más probable en cada celda de la grilla. El modelo utiliza datos históricos de la ciudad de Dajeon en Corea del Sur, junto con datos del flujo de tránsito y comportamiento de los conductores. Además, los autores analizan los pros y contras de utilizar un enfoque basado en grilla para dividir la ciudad en contraste con un modelo basado en grafos, en el que se modela cada segmento de calle como un nodo del grafo. El análisis concluye que ambos enfoques son válidos, pero que el enfoque basado en grillas puede ser más útil a la hora de detectar patrones espaciales, ya que los accidentes no ocurren de manera uniforme a lo largo de segmentos de calle específicos, sino que tienden a agruparse en ciertas áreas.

El trabajo de Lin et al. [40] justifica de manera similar la decisión de utilizar una teselación de la superficie de estudio en lugar de modelar la red vial como un grafo. En este caso, se aplica una arquitectura que combina CNN y LSTM para procesar los datos de accidentes y volumen de tráfico en la ciudad de Taipei, pero también se combina con un MLP que procesa datos de contexto externos como condiciones climáticas y eventos del calendario (días de semana, feriados, etc.). A la hora de analizar los resultados, se evalúan los mapas de riesgo generados por distintos modelos de manera tanto cuantitativa (a partir de una función de pérdida) como cualitativa (en función de qué tan útil es la predicción generada). En este caso en particular, el modelo basado en CNN+LSTM supera a la iniciativa basada en ConvLSTM presentada en Yuan et al. [37].

Más recientemente, se ha explorado el uso de otras arquitecturas más complejas para afrontar el problema. En Grigorev et al. [41] se propone un modelo basado en ViT que aprovecha la capacidad de estas arquitecturas para capturar relaciones espaciales complejas en los datos de accidentes. Los autores construyen un mapa de riesgo bidimensional que puede ser interpretado como una imagen multicanal, donde cada canal representa una ventana temporal diferente, de manera que el modelo pueda aprender patrones espaciales y temporales de manera conjunta. A su vez, se incorporan datos externos como condiciones climáticas y flujo de tráfico a través de un MLP que se concatena con las representaciones aprendidas por el ViT antes de la capa de salida.

En Al-Thani et al. [42] se utiliza un *transformer* con arquitectura *encoder-decoder* para enfocarse únicamente en la dimensión temporal del problema. Dicho trabajo compara el modelo basado en *transformers* con otros enfoques basados en LSTM y demuestra que el primero supera a los segundos especialmente a la hora de capturar dependencias a largo plazo entre los datos.

También es relevante mencionar los enfoques basados en Red Neuronal de Grafos (*Graph Neural Network*, GNN), que han ganado popularidad en los últimos años debido a su capacidad para modelar relaciones complejas en datos estructurados como grafos. En Wang et al. [43] se propone un modelo que utiliza Red Convolucional de Grafos (*Graph Convolutional Network*, GCN) para capturar las relaciones espaciales entre diferentes ubi-

caciones en una red vial, combinándolas con LSTM para modelar la dimensión temporal. De esta forma, se puede modelar la topología de la red vial de manera mucho más efectiva, no solo considerando la distancia geográfica entre puntos, sino también considerando la conectividad real entre los mismos.

En Yu et al. [44] se utilizan GCN para modelar las relaciones espaciales entre diferentes ubicaciones en un conjunto de datos propietario de la ciudad de Beijing. El modelo permite capturar datos heterogéneos como la estructura de la red vial, el flujo de tráfico y las condiciones climáticas y otros puntos de interés. Los resultados alcanzados resultan prometedores comparados con enfoques basados en LSTM y ML clásico, pero la naturaleza privativa del conjunto de datos dificulta la comparación directa con otros enfoques.

El trabajo de Liu et al. [45] explora la posibilidad de combinar GCN con un mecanismo basado en atención para refinar la salida de la red y capturar las relaciones que producen el mayor riesgo de accidente. También se utiliza una función de pérdida focal como una alternativa a la función de pérdida ponderada para manejar el desbalance de clases en el conjunto de datos. Tomando una superficie de estudio muy densamente poblada en la ciudad de Beijing, se construye un conjunto de datos con mediciones muy detalladas de la red vial donde cada segmento de calle es un nodo en el grafo, y encuentran resultados muy similares a implementaciones de GCN previas.

Por último, es interesante mencionar los resultados presentados en revisiones sobre el área de investigación como Behboudi et al. [46], donde se analizan diversos enfoques para estudiar el riesgo de accidentes de tránsito. Se destaca que, particularmente en los modelos basados en DL, la calidad de los datos de entrada es el factor determinante a la hora de conseguir buenos resultados. Esto es consistente con la idea general de los modelos basados en ANN, que permiten minimizar la tarea de ingeniería de características, pero requieren grandes cantidades de datos de alta calidad para aprender representaciones útiles.

En resumen, la predicción espacio-temporal del riesgo de accidentes de tránsito es un área de investigación activa que ha visto avances significativos gracias a la aplicación de técnicas de ML y DL. Los enfoques basados en CNN, LSTM, *transformers* y GNN representan el estado del arte en este campo, y continúan evolucionando a medida que se desarrollan nuevas técnicas y se dispone de conjuntos de datos más ricos y detallados. En particular, en este trabajo final de carrera se explora el uso de ViT para abordar este problema, considerando los datos disponibles para la ciudad de Mendoza, como clima e información diaria de accidentes vehiculares, junto con información contextual relevante como feriados, días de la semana y calendario académico.

3

Metodología y desarrollo del trabajo

En este capítulo se detalla el diseño metodológico adoptado para llevar a cabo el desarrollo de este trabajo. Inicialmente, en la sección 3.1 se detalla en profundidad la descripción de la tarea a realizar. En la Sección 3.2 se presenta el flujo de trabajo general del proyecto, describiendo las etapas principales desde la recolección de datos hasta la implementación de la aplicación web. En las secciones posteriores, se profundiza en cada una de las etapas, incluyendo la generación y preparación del conjunto de datos (3.3), la arquitectura de los modelos propuestos (3.4), la configuración de hiperparámetros (3.6), y finalmente la implementación de la aplicación web (3.9).

3.1. Definición formal de la tarea a realizar

A partir de la revisión bibliográfica realizada, se puede definir la tarea que se busca llevar a cabo de la siguiente manera:

Definición formal de la tarea: *en base a datos históricos de accidentes viales en la Ciudad de Mendoza, se busca entrenar modelos de DL que permitan predecir el riesgo de accidentes de tránsito en dos dimensiones: espacial (en qué zona de la ciudad) y temporal (cómo evoluciona el riesgo en el tiempo). El mapa de riesgo para el período actual debe poder visualizarse de forma interactiva a través de una aplicación web.*

Para esto, es necesario definir el concepto de riesgo de accidente de tránsito. En este trabajo, se plantea la tarea como un problema de clasificación multiclase, donde

el objetivo es predecir la clase de riesgo para cada celda de la grilla espacial en un día determinado. Se definen tres clases de riesgo:

- **Bajo** (clase 0)
- **Medio** (clase 1)
- **Alto** (clase 2)

Considerando que el riesgo se define tradicionalmente como la probabilidad de que ocurra un fenómeno, multiplicado por el impacto que tiene, se crea una función de riesgo que tiene en cuenta la cantidad de accidentes y la gravedad de los mismos. Esta función se aplica a cada celda del mapa de riesgo, y se utiliza para determinar la clase de riesgo de cada celda. La misma puede definirse de la siguiente manera:

$$Riesgo = min(3 \times N_{atropellos} + 2 \times N_{choques} + N_{colisiones} + N_{otros}, 2) \tag{3.1}$$

Donde $N_{atropellos}$, $N_{choques}$, $N_{colisiones}$ y N_{otros} representan la cantidad de accidentes de cada tipo en la celda según la clasificación propuesta en la Sección 2.1.1.

Esta ecuación permite obtener un valor de riesgo entre 0 y 2, donde 0 representa un riesgo bajo, 1 un riesgo medio y 2 un riesgo alto.

3.2. Flujo de trabajo del proyecto

El desarrollo de este proyecto sigue un flujo de trabajo estructurado que abarca desde la recolección de datos hasta la implementación de una aplicación web interactiva. La Figura 3.1 presenta un diagrama que ilustra las etapas principales del proceso.

La primera etapa consiste en la recolección de datos de múltiples fuentes, seguido de un análisis exploratorio para comprender las características del conjunto de datos. Posteriormente, se realiza el preprocesamiento necesario para transformar los datos en un formato adecuado para los modelos de aprendizaje profundo. Luego de dividir el dataset en conjuntos de entrenamiento, validación y prueba, se procede al entrenamiento de los modelos propuestos. Finalmente, se evalúan los resultados y se implementa una aplicación web que permite visualizar las predicciones de riesgo.

3.3. Generación y preparación del conjunto de datos

El primer paso para conseguir entrenar los modelos de DL consiste en la obtención y preparación del conjunto de datos. Esto es necesario para tomar decisiones sobre

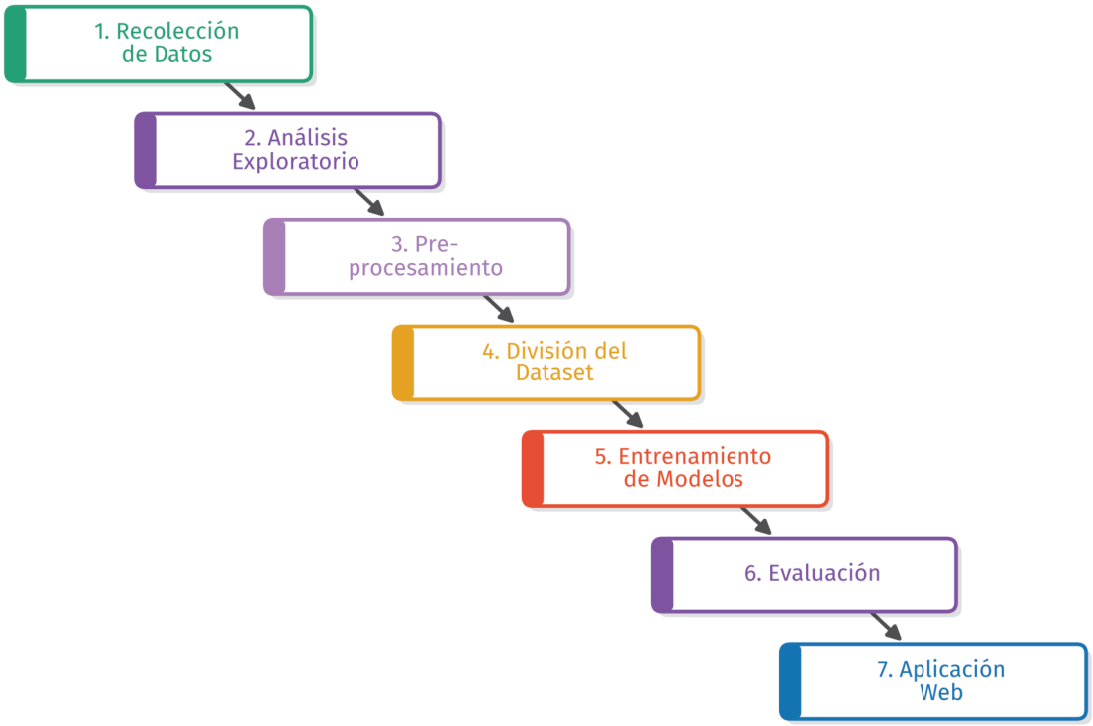


Figura 3.1: *Flujo de trabajo del proyecto*

qué tipo de modelos implementar, ajustar los mismos correctamente, y garantizar que se alcanza el mejor rendimiento posible en la práctica.

3.3.1. Recolección de datos

Este desarrollo utiliza datos de accidentes viales proporcionados por la Dirección de Catastro de la Ciudad de Mendoza a través de su API de WebGIS. El conjunto de datos comprende ~ 9 700 registros de accidentes viales, cada uno caracterizado por 35 atributos que describen aspectos temporales, espaciales y contextuales del siniestro. La Tabla 3.1 presenta una descripción detallada de las variables disponibles, organizadas por categoría.

A su vez, para complementar los datos de accidentes, se utilizan fuentes de datos externas de meteorología y POI. Los datos meteorológicos fueron proporcionados por el Servicio Meteorológico Nacional (SMN) y presentan una estructura como la descrita en la Tabla 3.2.

Los datos de POI corresponden a la localización geográfica de puntos relevantes para el análisis de accidentes, como hospitales, centros educativos, semáforos y otra señalética vial (discos pare). Estos datos son obtenidos a partir de archivos de capas (*shapefiles*) provistos también por la Dirección de Catastro. La Tabla 3.3 presenta un resumen de los conjuntos de datos de POI utilizados.

Tabla 3.1: *Descripción de las columnas del conjunto de datos de accidentes viales*

Categoría	Columna	Descripción
Identificación	Acta ID	Identificador único del acta de accidente
	Cantidad de vehículos	Número de vehículos involucrados
	Relevamiento	Tipo de relevamiento (lugar del hecho, dependencia)
	Presenta vehículo	Indica si hay vehículo presente en el registro
Temporal	Fecha siniestro	Fecha y hora del accidente
	Fecha actuación	Fecha y hora en la que se realizó la denuncia
Ubicación	Intersección	Calle de intersección
	Calle	Nombre de la calle principal
	Altura	Número de altura de la calle
	Calle compuesta	Dirección completa formateada
	x	Coordenada de longitud
	y	Coordenada de latitud
Tipo de accidente	Tipo de accidente	Categoría general (colisión, choque, etc.)
	Colisión	Subtipo de colisión (frontal, de alcance, etc.)
	Choque	Indica si hubo choque
	Atropello	Indica si hubo atropello
	Otro tipo	Descripción de otros tipos de accidente
Infraestructura	Calzada	Tipo de pavimento (asfalto, adoquín, etc.)
	Inclinación calzada	Grado de inclinación de la calzada
	Ciclovía	Presencia de ciclovía
	Calle ciclovía	Calle donde se ubica la ciclovía
Condiciones	Visibilidad e iluminación	Condiciones de visibilidad (buena, nocturno)
	Fenómenos climáticos	Condiciones climáticas al momento del accidente
	Otro fenómeno	Descripción de otros fenómenos relevantes
Señalización	Tipo de señal	Tipo de señalización presente
	Calle señal	Ubicación de la señal
	Visión señal	Visibilidad de la señal
	Estado semáforo	Estado operativo del semáforo
Otros elementos	Cámara	Presencia de cámara de vigilancia
	Estacionamiento medido	Presencia de estacionamiento medido
	Parada de colectivo	Cercanía a parada de transporte público
	Senda peatonal	Presencia de senda peatonal
	Calle senda	Ubicación de la senda peatonal
	Eje central	Tipo de eje central de la vía
	Nomenclador	Visibilidad del nomenclador de calles

Tabla 3.2: *Descripción de las variables meteorológicas obtenidas del SMN*

Variable	Descripción
date	Fecha de la observación (YYYY-MM-DD)
temp_max	Temperatura máxima registrada (°C)
temp_min	Temperatura mínima registrada (°C)
temp_avg	Temperatura media diaria (°C)
pressure	Presión atmosférica media a nivel de estación (hPa)
precipitation	Cantidad de precipitación acumulada (mm)
humidity	Humedad relativa promedio (%)
wind_speed_max	Velocidad máxima de ráfaga de viento (km/h)
wind_speed_avg	Velocidad media del viento (km/h)
wind_dir_X	Dirección predominante del viento (16 puntos cardinales)

Tabla 3.3: *Descripción de los conjuntos de datos de POI*

Conjunto de datos	Cantidad de entradas	Atributos principales
Semáforos	1213	Tipo de luz, tipo de semáforo, ubicación
Señales Pare/Ceda	307	Tipo de señal, ubicación
Centros de Salud	62	Nombre, tipo de establecimiento, ubicación
Edificios Educativos	91	Nombre del establecimiento, ubicación
Universidades	31	Nombre de la institución, ubicación

3.3.2. Análisis exploratorio de datos

Una vez obtenidos los datos provenientes de las tres distintas fuentes de información, es esencial analizar la distribución de los mismos a través de distintas variables con el objetivo de comprender en profundidad las tendencias. En particular, en esta sección se realiza un estudio de los datos de accidentes.

Distribución temporal de accidentes

La Figura 3.2 muestra cómo se distribuyen los accidentes en la ciudad de Mendoza según el día de la semana. La tendencia que puede apreciarse en la misma es que los días de semana presentan un mayor número de accidentes que los fines de semana, mostrando cantidades muy similares entre sí. Puede establecerse una correlación entre la cantidad de accidentes y la actividad humana típica de cada día, considerando que el flujo vehicular es mayor desde el lunes hasta el sábado por la mañana, y disminuye notablemente los domingos.

Por otro lado, la Figura 3.3 ilustra la distribución de accidentes según la hora del día. Nuevamente, la tendencia descrita por la imagen es clara: entre las 23hs y las 6hs puede apreciarse un mínimo en la cantidad de accidentes. La cantidad de accidentes crece rápidamente hasta alcanzar un punto máximo a las 13hs que baja inmediatamente, hasta que vuelve a subir alrededor de las 18hs.

También puede establecerse una correlación entre lo que muestran las estadísticas y los horarios de mayor actividad en la ciudad de Mendoza. El pico de las 13hs puede

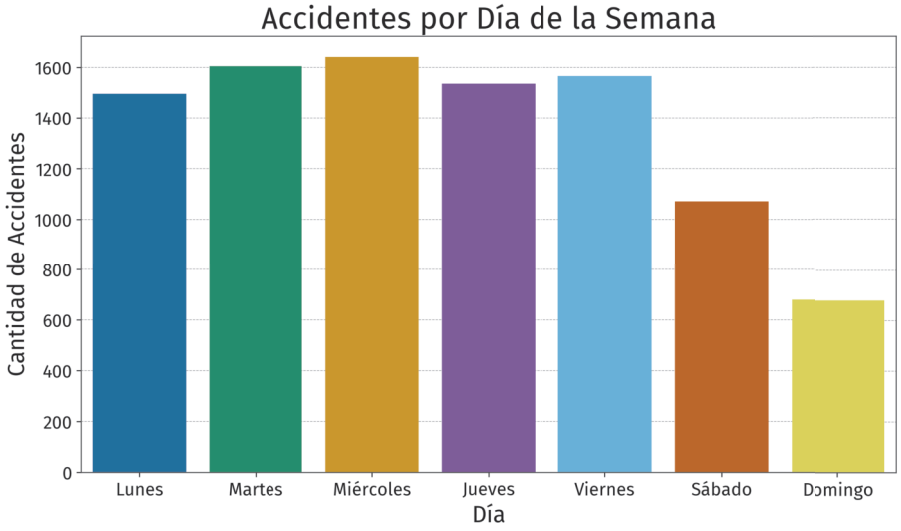


Figura 3.2: *Distribución de accidentes por día de la semana*

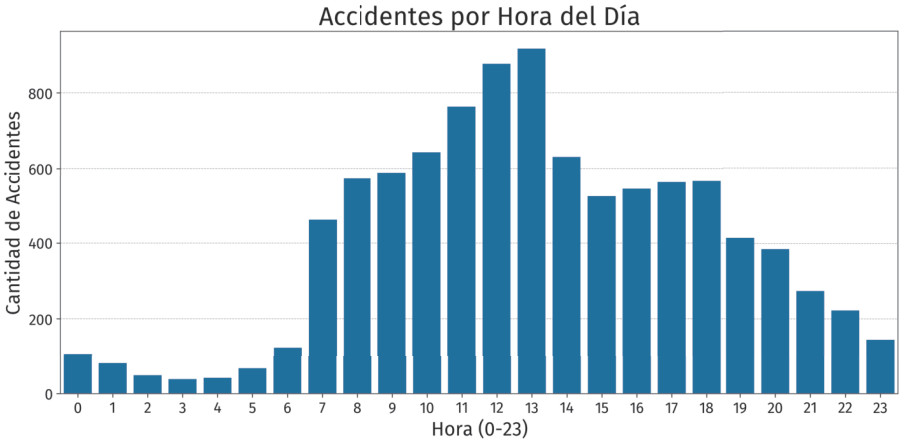


Figura 3.3: *Distribución de accidentes por hora del día*

relacionarse con el horario de salida del turno mañana en las escuelas y entrada del turno tarde, y con el corte de la jornada laboral para aquellos que no trabajan en horario corrido.

Por último, se analiza el promedio diario de accidentes a través de los meses como se muestra en la Figura 3.4. La tendencia en este caso es mucho menos clara, ya que cada año presenta un comportamiento ligeramente diferente. Sin embargo, pueden observarse algunos patrones que se repiten en todos ellos, como que el mes de enero representa el mes con menos accidentes de todo el año, mientras que el pico suele encontrarse alrededor de los meses de septiembre y octubre.

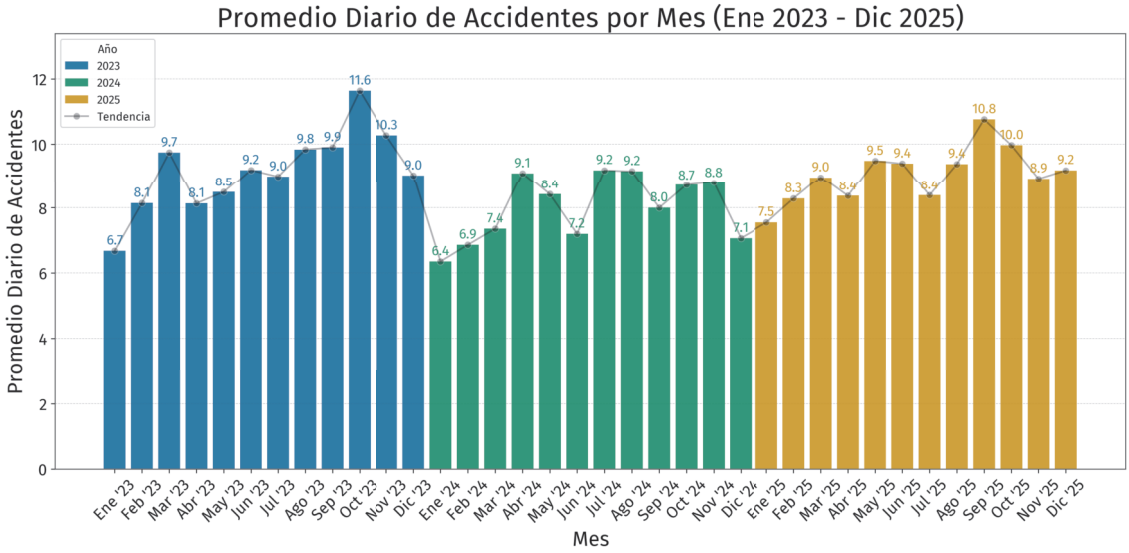


Figura 3.4: *Promedio diario de accidentes por mes*

Distribución de las clases

A partir de la clasificación de accidentes propuesta en la Sección 2.1.1, se analiza cómo se distribuyen los accidentes según su clase. La Figura 3.5 muestra que la enorme mayoría de los accidentes de tránsito corresponden a colisiones, mientas que se puede detectar una cantidad mucho menor de atropellos, choques, y finalmente existe una pequeña cantidad de accidentes clasificados como “otros”.

Distribución geográfica de los accidentes

La Figura 3.6 muestra la distribución espacial de todos los accidentes registrados sobre un mapa base de la Ciudad de Mendoza. Se puede observar que la gran mayoría de los accidentes se concentra en el centro urbano de la ciudad, formando una estructura que coincide con la disposición de las calles principales y avenidas.

Sin embargo, al analizar la zona oeste del mapa, se detecta un patrón diferente: los accidentes en esta región se encuentran altamente dispersos, con la excepción de algunos registros alineados sobre la Avenida Champagnat. Esta dispersión representa registros

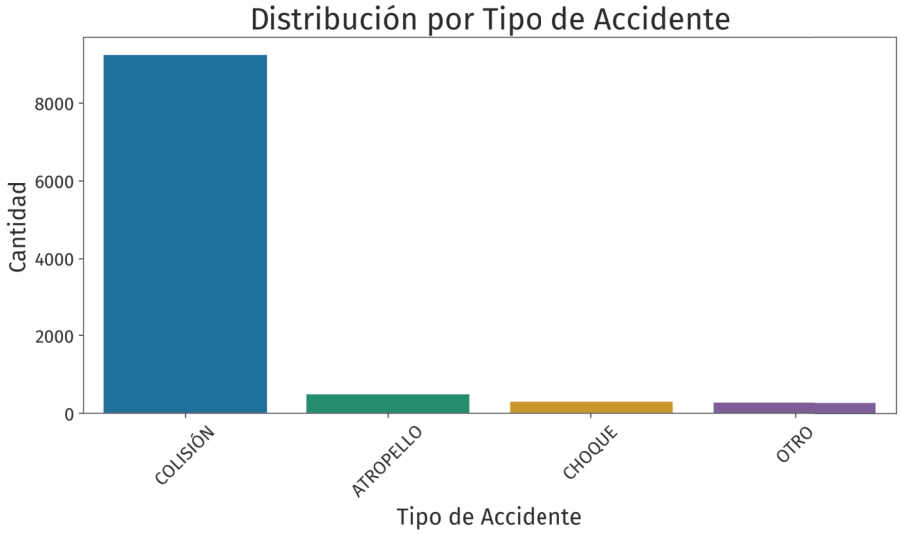


Figura 3.5: *Distribución de accidentes por tipo*

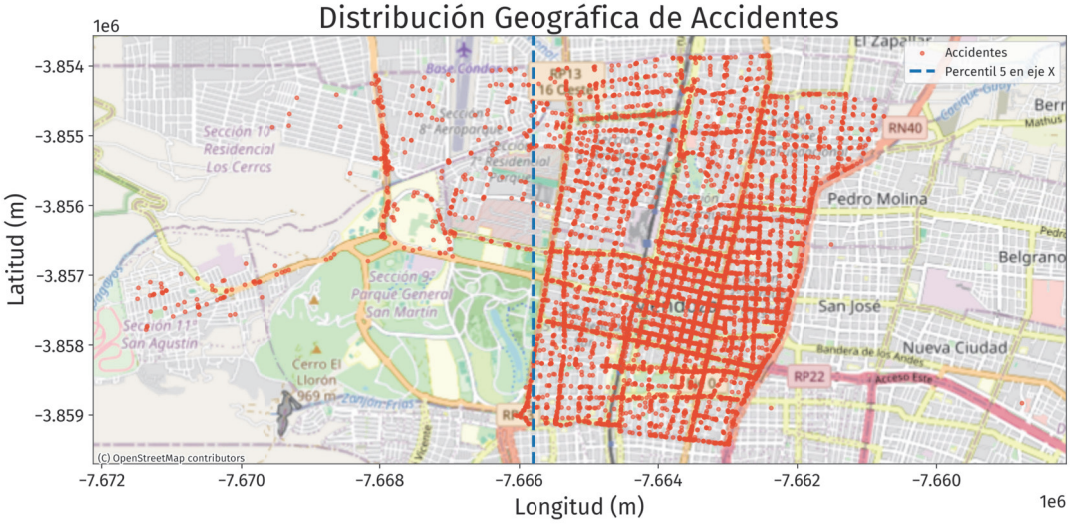


Figura 3.6: *Distribución geográfica de accidentes con línea de corte en el percentil 5*

atípicos (*outliers*) que corresponden a menor densidad urbana, donde los accidentes son esporádicos y no representativos del comportamiento general del tránsito urbano.

La Figura 3.7 presenta un diagrama de caja (*boxplot*) de la coordenada X (longitud) de todos los accidentes, donde puede apreciarse claramente la presencia de valores atípicos hacia el oeste (valores menores de X). La línea punteada indica el percentil 5, que se utiliza como umbral para el filtrado espacial.

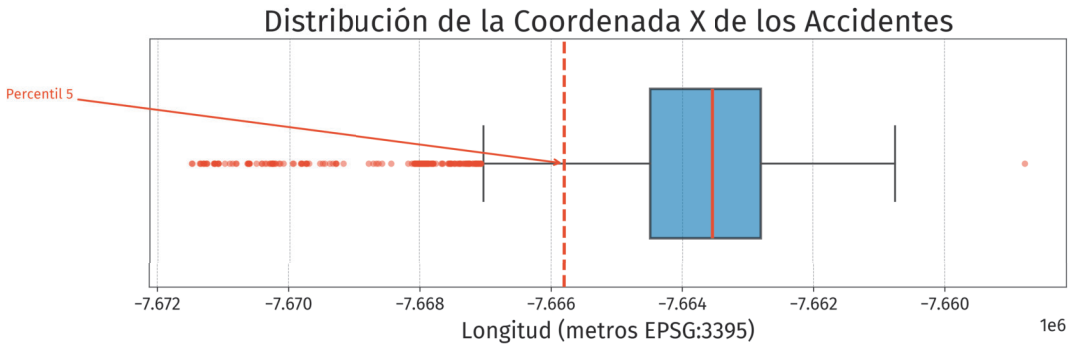


Figura 3.7: *Distribución de la coordenada X (longitud) de los accidentes. La línea punteada indica el percentil 5 utilizado como umbral de filtrado.*

Por esta razón, se aplica un filtro espacial basado en el percentil 5 del eje de longitud (eje X), eliminando el 5 % de los accidentes, para concentrar la tarea del modelo a desarrollar, asegurando que se concentre en las zonas con mayor densidad de accidentes. La intuición matemática para esto es que, si se elimina solo el 5 % de los eventos pero el 50 % de la superficie de trabajo, la densidad de accidentes por km^2 se estaría casi duplicando.

La Figura 3.8 presenta el resultado de este filtro, mostrando únicamente los accidentes dentro del área urbana más densa. A simple vista, puede apreciarse la mayor concentración de siniestros, muchos de ellos coincidiendo con las principales vías de la ciudad. Esto genera una representación de los datos más homogénea.

3.3.3. Preprocesamiento de datos

A partir del estudio de la composición de los datos, se buscar aplicar transformaciones que permitan procesarlos de manera espacio-temporal para alimentar un modelo de ML. El objetivo de esta etapa es diseñar y aplicar un formato estandarizado que capture los datos que se consideran más importantes de forma independiente al resto de las tareas.

El *pipeline* de procesamiento consta de cuatro etapas principales: generación de la grilla sobre la superficie de la ciudad, agregación temporal, ingeniería de características y conformación del tensor final.

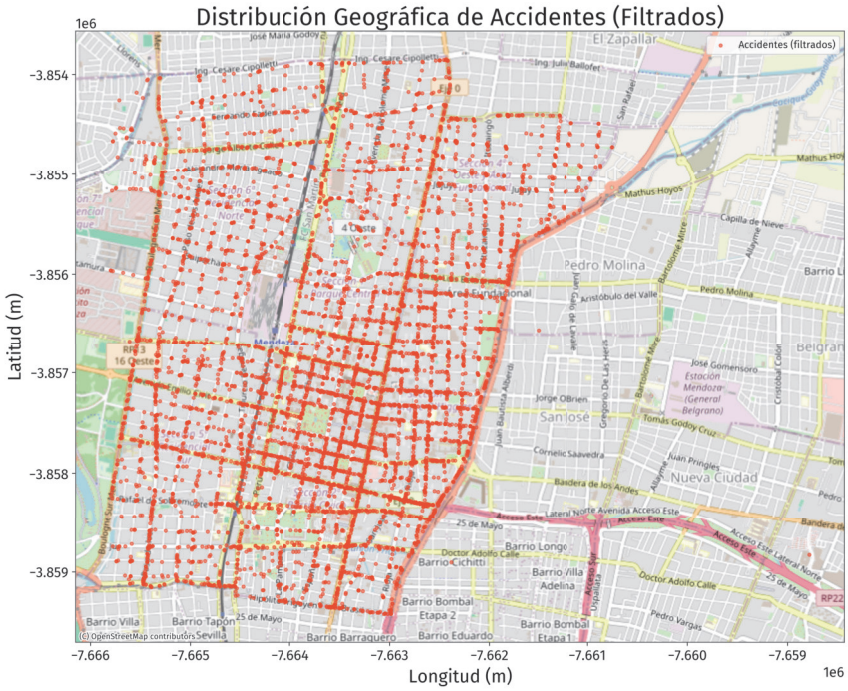


Figura 3.8: *Distribución geográfica de accidentes filtrados (percentil 5 en longitud)*

Generación de la grilla

Para manejar la naturaleza espacial de los accidentes, se divide el área de estudio (conformada por el mínimo rectángulo que contiene todos los accidentes) en una grilla regular formada por $L \times L$ celdas. Este enfoque favorece la más sencilla aplicación de algoritmos de ML posteriormente, ya que los mismos trabajan por defecto con datos tabulares.

El proceso de generación de la grilla sigue estos pasos:

1. Se filtran los accidentes atípicos que caen fuera de la zona urbana densa, utilizando un percentil espacial (por defecto el 5 %) para determinar los límites efectivos del área de estudio.

2. Se crea una teselación regular de celdas cuadradas de tamaño $L \times L$.

3. Cada accidente se asigna a una celda específica mediante una operación espacial de ”punto en polígono”.

Esta discretización permite tratar el problema como una tarea de predicción sobre una matriz de dimensiones $H \times W$, donde H y W son el número de filas y columnas de la grilla respectivamente.

Tamaño de celda

Un parámetro fundamental a la hora de generar la grilla es el tamaño de celda L. Este valor determina el nivel de detalle espacial del modelo, y debe ser seleccionado manteniendo un balance entre resolución, cantidad de datos y utilidad a la hora de interpretar los resultados.

La Figura 3.9 muestra una comparación de distintas configuraciones de grilla sobre el área de estudio filtrada, con grillas de 10×10, 12×12, 16×16 y 20×20 celdas. Puede observarse que a medida que la cantidad de celdas aumenta, el tamaño de cada celda disminuye, lo que incrementa la resolución espacial del modelo pero también aumenta la complejidad computacional y el riesgo de celdas con pocos o ningún accidente.

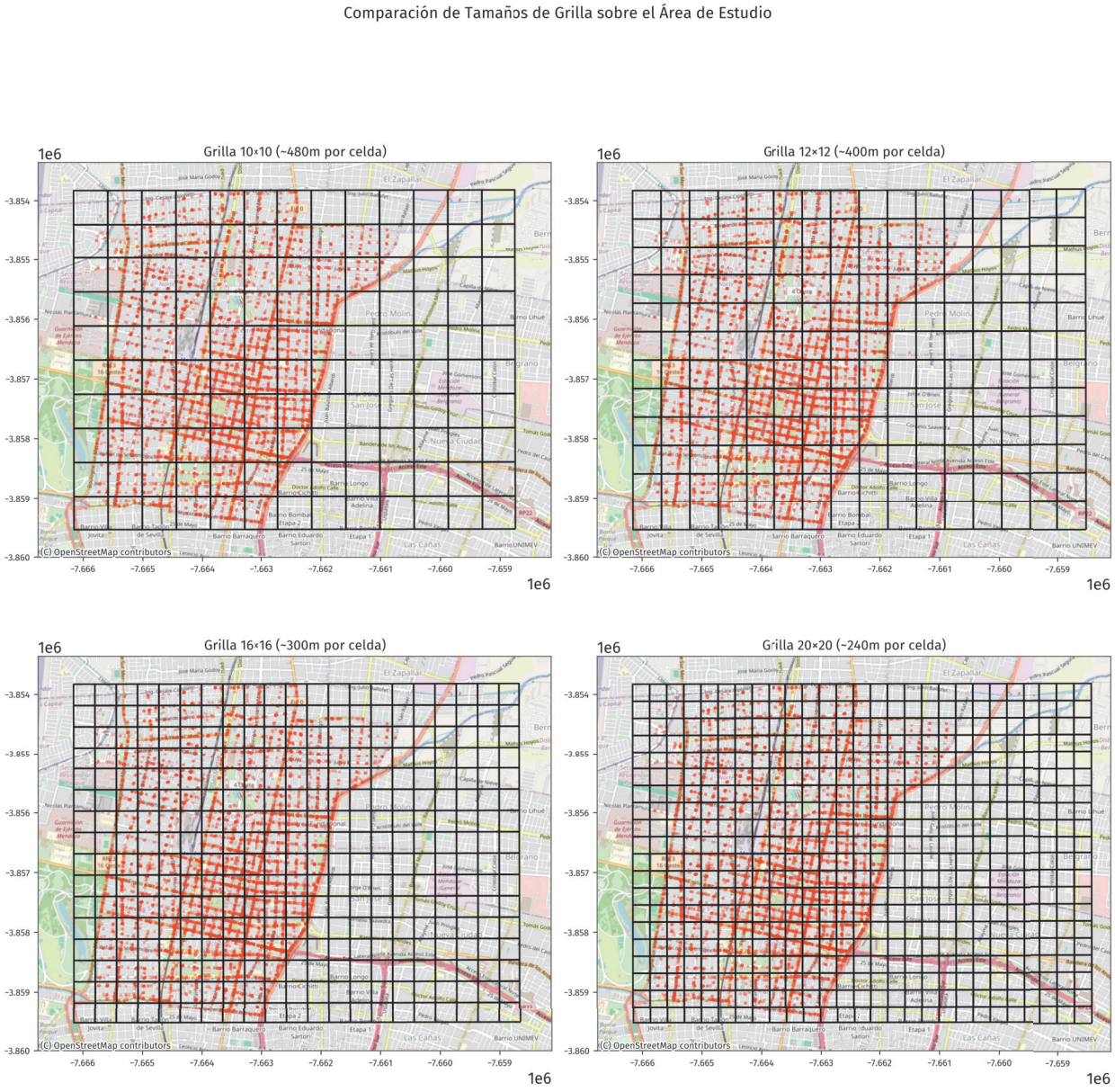


Figura 3.9: *Comparación de diferentes tamaños de grilla*

Adicionalmente al rendimiento que obtenga el modelo en una configuración de grilla en particular, la elección de un tamaño debe teniendo en cuenta la utilidad de las predicciones a la hora de tomar decisiones. Aunque trabajos como los de Grigorev et al. [41] o Wang et al. [43] utilizan grillas de 20×20, el área de la ciudad de Mendoza es mucho más pequeña que la de los conjuntos de datos utilizados en dichos estudios. Considerando esta diferencia de área, en conjunto con la baja densidad de accidentes que presenta la ciudad, se elige la configuración de 12×12 celdas con $L = 400m$ como foco central de este trabajo: esto quiere decir que la aplicación web y la búsqueda de hiperparámetros sobre los modelos se llevarán a cabo sobre la especificación mencionada. A pesar de que se elige la configuración de 12×12 para llevar a cabo el desarrollo, se realizan experimentos adicionales con otras configuraciones de grilla para analizar su impacto en el rendimiento del modelo: tomando los mejores modelos obtenidos, se utiliza su configuración para entrenar nuevas versiones que utilicen grillas de 10×10, 16×16 y 20×20 para comparar los resultados.

Agregación Temporal y Fusión de Datos

Una vez generada la división espacial de los datos, se procede a la agregación temporal. Se define una resolución temporal diaria, agrupando los accidentes ocurridos en cada celda por día. A diferencia de otros trabajos relacionados que utilizan una resolución horaria ([41], [43]), se elige una resolución diaria por los siguientes motivos:

- **Falta de datos de tráfico**: estos datos presentan la mayor variación intradiaria y no están disponibles para la ciudad de Mendoza. De estar disponibles, se debería considerar una resolución horaria.
- **Dispersión de los datos**: dado que se cuenta con un promedio de menos de 10 accidentes diarios en toda la ciudad, una resolución horaria resultaría en una gran cantidad de celdas vacías, dificultando el aprendizaje del modelo. Considerando que cada día tiene asociado no un solo punto de datos, sino una grilla completa de celdas, una resolución horaria implicaría más de un 99 % de celdas vacías.

Para cada par (día, celda), se construye un vector de características que combina información de múltiples fuentes:

- **Variables Estáticas - POI:** Se calculan conteos de elementos de infraestructura dentro de cada celda. Estas características son constantes en el tiempo para una celda dada.
  - Cantidad de semáforos.
  - Cantidad de señales de ”Pare”.
  - Cantidad de escuelas y universidades (edificios educativos).

- Cantidad de centros de salud.

- **Historial de Accidentes (Ventanas de Riesgo):** Para capturar la tendencia temporal, se calculan ventanas de riesgo históricas. Utilizando la función de riesgo definida en la Sección 3.1, se calcula el riesgo para cada celda en cada día.

  Luego, se calculan sumas móviles de este índice para diferentes ventanas de tiempo hacia el pasado (1, 2, 3, 4, 5, 7, 14, 28 días). Estas variables permiten al modelo entender tanto el riesgo inmediato como las tendencias de largo plazo de cada esquina o zona.

  La elección de estas ventanas se basa en el trabajo de Grigorev et al. [41] y tiene una interpretación en términos de patrones temporales. Las ventanas de hasta 5 días capturan patrones a muy corto plazo, mientras que las de 7, 14 y 28 días capturan patrones un plazo más lejano manteniendo la periodicidad semanal. De esta forma se puede capturar la esencia de la serie de tiempo.

Además, se integran las variables meteorológicas diarias correspondientes a la fecha del registro. Ya que estos datos son idénticos para cada una de las celdas, no se agregan a la grilla espacial, sino que se mantienen como un vector adicional para ser introducido en una etapa posterior del modelo.

A los datos meteorológicos, se le agregan más variables invariables en el espacio, entre las que se encuentran:

- Día de la semana (1-7)
- Mes del año (1-12)
- Si es fin de semana (sí-no)
- Si es feriado (sí-no)
- Si el día se encuentra dentro del período escolar (sí-no)

Formato de Salida

El resultado final del preprocesamiento es un conjunto de datos estructurado que representa un tensor espacio-temporal. Cada fila del *dataset* corresponde a una celda en un día específico, conteniendo:

- Identificadores: Fecha, ID de celda.
- Variable Objetivo (*Target*): Índice de riesgo calculado para el día actual (lo que el modelo debe predecir).

- Características de Entrada (*Features*): conteos de POI y ventanas históricas de riesgo (riesgo acumulado en los N días previos).

Esta estructura puede ser visualizada como una imagen multicanal, donde cada canal representa una característica distinta distribuida espacialmente, tal como se ilustra en la Figura 3.10.

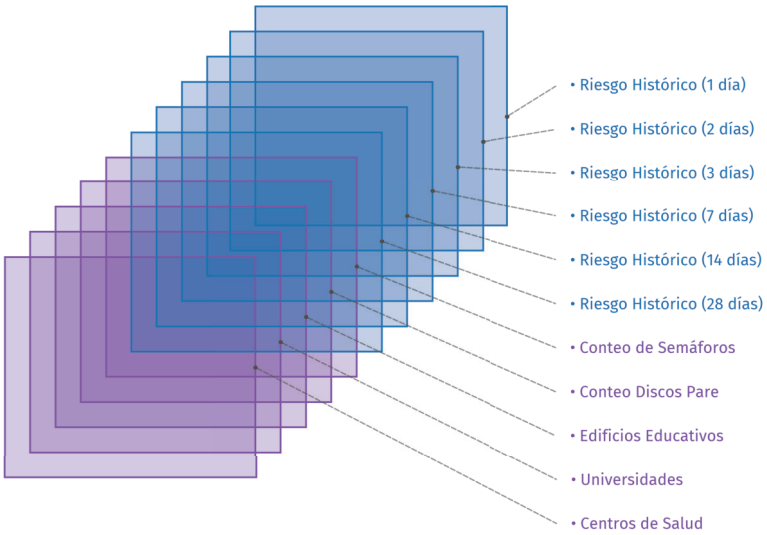


Figura 3.10: *Representación de los canales de entrada del modelo.*

También se genera el conjunto de datos meteorológico, que puede ser fusionado a la grilla a través de la fecha.

3.3.4. Análisis del conjunto de datos final

Una vez construido el conjunto de datos final, es importante analizar su distribución, principalmente para comprender cómo impacta la división espacial del área de estudio en la cantidad de muestras disponibles para el entrenamiento. La Figura 3.11 presenta la distribución de la variable objetivo (riesgo de accidentes) en el conjunto de datos final. Puede observarse que la mayoría de las muestras corresponden a riesgo bajo (clase 0), con una cantidad menor de muestras para riesgo medio (clase 1) y una cantidad aún menor para riesgo alto (clase 2). Esta distribución desigual debe ser tenida en cuenta durante el entrenamiento de los modelos, ya que puede afectar su capacidad para aprender a predecir correctamente las clases menos representadas.

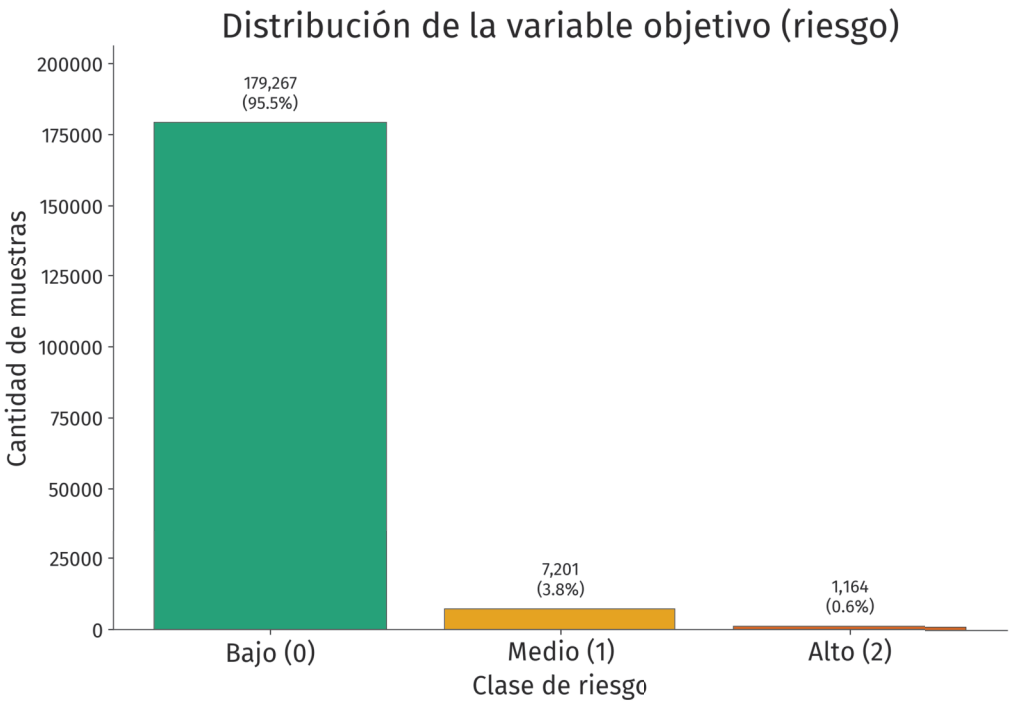


Figura 3.11: *Distribución de la variable objetivo (riesgo de accidentes) en el conjunto de datos final.*

3.4. Arquitectura de los modelos

En esta sección se describen las arquitecturas de modelos de DL que se busca comparar para la predicción del riesgo de accidentes viales.

3.4.1. Risk Vision Transformer (RiskViT)

En este trabajo final de carrera se construye un modelo basado en *Vision Transformer* (ViT), adaptado para procesar imágenes de una cantidad arbitraria de canales con el objetivo capturar la información como se propone en la Sección 3.3.3 (Figura 3.12).

Este modelo se basa fuertemente en la arquitectura propuesta por Grigorev et al. [41], para los tamaños de clase y características específicas del caso particular de la ciudad de Mendoza. La arquitectura completa consta de las siguientes etapas:

- **Extracción de parches:** La imagen de entrada, una grilla espacial de características de tamaño $W \times H \times C$, se divide en parches no solapados de tamaño $P \times P$. Una restricción importante de esta capa es que P debe ser un divisor exacto de las dimensiones espaciales la entrada. Para este desarrollo se considera solo el caso $H = W$, para trabajar con imágenes cuadradas.
- **Embeddings posicionales:** Cada parche es aplanado y proyectado linealmente a un espacio de dimensión D a través de una capa *fully connected*. De esta forma, se generan los *embeddings* aprendibles que serán la entrada del bloque *transformer*.

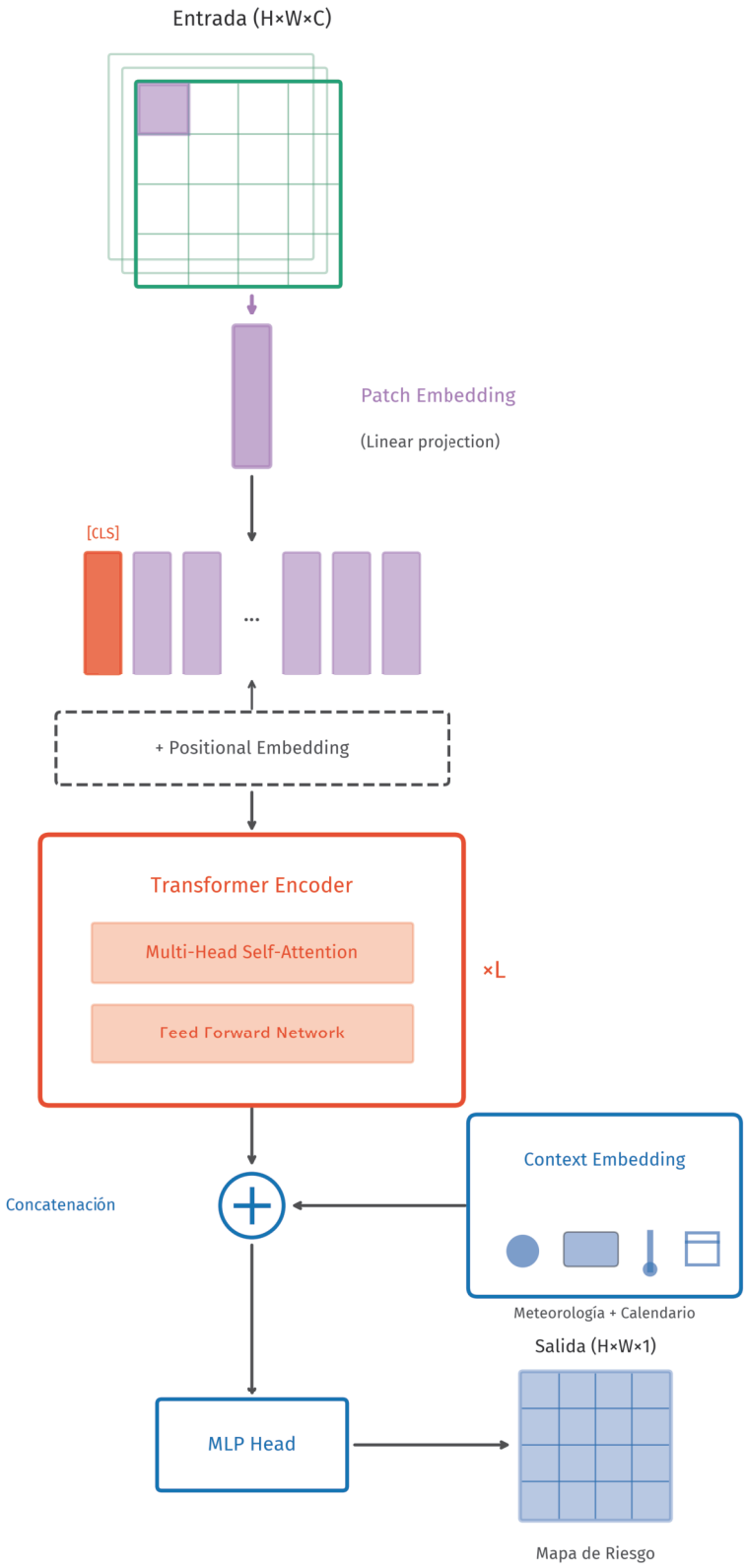


Figura 3.12: *Arquitectura del modelo RiskViT*

- ***Transformer encoder*:** Los embeddings de cada uno de los parches se introducen como una secuencia al encoder, precedidos por el token [CLS] que se utiliza para la clasificación final [33]. En este bloque, se presentan L cabezas de atención.

- **Fusión con datos meteorológicos:** La salida del *transformer* se concatena con el vector de características meteorológicas y temporales.

- **MLP final:** Finalmente, la representación combinada se pasa por una red neuronal feed-forward para producir la predicción final del riesgo de accidentes. La salida es un vector lineal de tamaño igual a la cantidad de celdas, que debe ser reestructurado a la forma de grilla.

3.4.2. CNN-LSTM

La otra arquitectura que se propone en contraste a RiskViT es una arquitectura híbrida que combina redes neuronales convolucionales (CNN) para la extracción de características espaciales, y redes neuronales recurrentes (LSTM) para modelar la dependencia temporal de los datos. La Figura 3.13 ilustra la arquitectura propuesta.

A diferencia del modelo RiskViT que procesa una única imagen multicanal (un solo mapa de riesgo), este modelo recibe una secuencia temporal de T mapas de riesgo, permitiendo capturar explícitamente la evolución temporal del riesgo. La arquitectura completa consta de las siguientes etapas:

- **Entrada:** El modelo recibe como entrada una secuencia de T mapas de riesgo, donde cada mapa corresponde a un día distinto. Esto permite observar la evolución del riesgo a lo largo del tiempo de manera explícita.

- **Capas convolucionales:** Cada mapa de la secuencia se procesa de forma independiente a través de una secuencia de capas convolucionales. En esta parte, se aplica una operación de convolución 2D, seguida de normalización por lotes y activación ReLU, finalizando con operaciones de *Max Pooling* y aplanado (*Flatten*). El resultado es un vector de características de dimensión d para cada paso temporal.

- **LSTM:** La secuencia de vectores de características se introduce en una red LSTM, que modela las dependencias temporales entre los días consecutivos. El estado oculto final h_T del LSTM representa una codificación de toda la historia temporal.

- **Fusión con datos contextuales:** Similar al modelo RiskViT, el estado oculto final del LSTM se concatena con el vector de características meteorológicas y temporales (temperatura, precipitación, día de la semana, etc.).

- **MLP final:** La representación concatenada se pasa por una red neuronal *feed-forward* para producir la predicción final del riesgo de accidentes. La salida es un vector li-

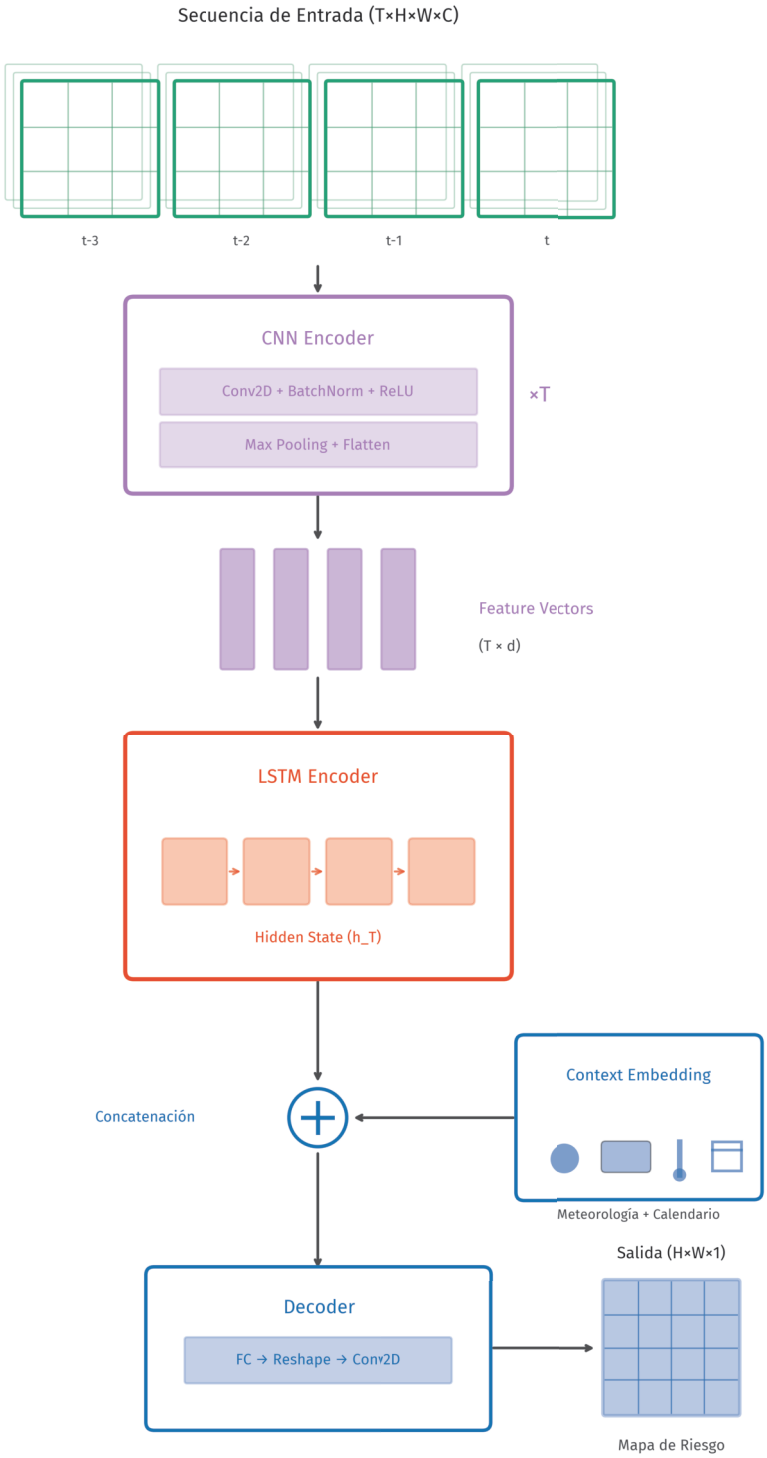


Figura 3.13: *Arquitectura del modelo CNN-LSTM*

neal de tamaño igual a la cantidad de celdas, que debe ser reestructurado a la forma de grilla.

Esta arquitectura permite modelar los análisis espaciales y temporales por separado de manera explícita. Así, se logra un modelo cuyo funcionamiento es más fácil de comprender, pero que tiene un mayor número de parámetros y, a su vez, es menos paralelizable, lo que aumenta el tiempo de entrenamiento.

3.5. Modelo *baseline*

A pesar de que en el presente trabajo se pretende estudiar la aplicación de modelos de DL para la predicción espacio-temporal de accidentes, se propone un modelo más sencillo que sirva como línea de base. Esto es importante para poder garantizar que el rendimiento obtenido a través de un enfoque complejo como un *pipeline* de DL se justifica frente a alternativas fáciles de implementar y comprender. En este caso, se plantea como línea de base un clasificador que prediga, para cada día, el mapa de riesgo del día anterior. Este modelo funciona como un punto de partida para comparar con métodos más avanzados.

La ecuación que representa este modelo puede definirse de la siguiente manera:

$$\hat{y}_{c,t} = y_{c,t-1} \tag{3.2}$$

donde $y_{c,t-1}$ representa el nivel de riesgo observado en la celda c el día anterior.

3.6. Configuración de hiperparámetros

Una vez definidos los modelos a utilizar, se diseña la estrategia para la optimización de hiperparámetros. Se define un conjunto de parámetros posibles para cada elemento de cada una de de las arquitecturas, así como parámetros comunes a ambas relacionadas con el proceso de entrenamiento.

3.6.1. Hiperparámetros de RiskViT

La Tabla 3.4 presenta los hiperparámetros configurables del modelo RiskViT y los valores evaluados durante la búsqueda de hiperparámetros. Algunos parámetros fueron fijados con valores estándar según la literatura de ViT, mientras que otros fueron explorados sistemáticamente.

Tabla 3.4: *Hiperparámetros del modelo RiskViT*

Parámetro	Descripción	Valores
`patch_size`	Tamaño de cada parche en que se divide la grilla	$\{1, 2, 4\}$
`dim`	Dimensión de los embeddings del *transformer*	$\{32, 64, 128\}$
`depth`	Número de capas del *transformer*	$\{4, 8\}$
`heads`	Número de cabezas de atención	$\{4, 8\}$
`mlp_dim`	Dimensión de la capa oculta del MLP	$\{32, 64, 128\}$
`context_dim`	Dimensión del vector de contexto (datos meteorológicos y temporales)	$\{0, 32\}$

3.6.2. Hiperparámetros de CNN-LSTM

La Tabla 3.5 presenta los hiperparámetros configurables del modelo CNN-LSTM. Esta arquitectura incluye parámetros tanto para la componente convolucional como para la recurrente, además de los parámetros relacionados con el manejo de secuencias temporales.

Tabla 3.5: *Hiperparámetros del modelo CNN-LSTM*

Parámetro	Descripción	Valores
`cnn_filters`	Número de filtros para las capas convolucionales	$\{[32, 64], [32, 64, 128]\}$
`lstm_hidden_size`	Tamaño del estado oculto del LSTM	$\{64, 128\}$
`lstm_num_layers`	Número de capas LSTM apiladas	$\{1\}$
`sequence_length`	Longitud de la secuencia temporal de entrada	$\{1, 3, 7, 14\}$
`context_dim`	Dimensión del vector de contexto (datos meteorológicos y temporales)	$\{0, 32\}$

3.6.3. Parámetros comunes

Además de los hiperparámetros específicos de cada arquitectura, existen parámetros de entrenamiento comunes a ambos modelos que controlan el proceso de aprendizaje. La Tabla 3.6 describe estos parámetros y sus valores utilizados.

Los pesos de clase se calculan inicialmente basado en la proporción utilizada en [41] y [43].

3.6.4. Estrategia de búsqueda de hiperparámetros

Para identificar la combinación óptima de hiperparámetros, se implementó una búsqueda de grilla (*grid search*) sobre el espacio de hiperparámetros definido en las Tablas 3.4, 3.5 y 3.6, dando lugar a un total de 1145 configuraciones para RiskViT y 192 para CNN-LSTM. Este método de búsqueda consiste en evaluar sistemáticamente todas las combinaciones posibles de los valores especificados para cada hiperparámetro según su performance en el conjunto de validación.

Tabla 3.6: *Parámetros de entrenamiento comunes*

Parámetro	Descripción	Valor
`grid_size`	Tamaño de la grilla espacial de entrada	$\{12\}$
`batch_size`	Número de muestras procesadas simultáneamente en cada iteración. Un valor mayor acelera el entrenamiento pero requiere más memoria. La investigación al respecto sugiere que, para este caso, un valor pequeño debería ser ideal [47]	$\{8, 16\}$
`learning_rate`	Tasa de aprendizaje que controla el tamaño del paso en la optimización. Valores altos aceleran el aprendizaje inicial pero pueden causar inestabilidad	$\{10^{-3}, 10^{-4}, 10^{-5}\}$
`epochs`	Número máximo de iteraciones completas sobre el conjunto de entrenamiento. El entrenamiento puede detenerse antes si se activa *early stopping*	Máximo de 200
`weights`	Pesos asignados a cada clase a la hora de aplicar la función de pérdida	$\{[1, 4, 10]\}$

3.7. Método de entrenamiento

A la hora de entrenar el modelo, se utiliza una división de los datos en conjuntos de entrenamiento, validación y prueba. En este caso, se utiliza una división temporal, manteniendo los datos más recientes para validación y prueba, y utilizando los datos más antiguos para el entrenamiento, con el objetivo de evitar que se filtre información del futuro al pasado.

Se decide utilizar una división del conjunto de datos que asigne un 70 % de los mismos al conjunto de entrenamiento, un 15 % al conjunto de validación y un 15 % al conjunto de prueba.

- **Conjunto de entrenamiento:** Datos desde el 01/01/2023 hasta el 06/03/2025
- **Conjunto de validación:** Datos desde el 07/03/2025 hasta el 23/08/2025
- **Conjunto de prueba:** Datos desde el 24/08/2025 hasta el 10/02/2025

A diferencia de otros trabajos relacionados que utilizan una mayor proporción de datos para validación y prueba (por ejemplo, [43] se utiliza 60 % para entrenamiento, 20 % para validación y 20 % para prueba), se opta por una división más conservadora para maximizar la cantidad de datos disponibles para el entrenamiento, teniendo en cuenta que el tamaño del conjunto de datos es limitado.

3.7.1. *Early Stopping*

Durante el proceso de entrenamiento se utiliza la técnica de *early stopping* (Sección 2.5.9) para evitar el sobreajuste. En este caso, se utiliza una paciencia de 10 épocas, y

un mínimo cambio de 0.1 %: si durante 10 épocas consecutivas no se observa una mejora en la métrica de validación (reducción de la función de pérdida) mayor al 0.1 %, el entrenamiento se detiene automáticamente.

3.7.2. Optimizador

El optimizador es el componente de un modelo de DL que se encarga de ajustar los pesos del modelo para minimizar la función de pérdida. En este caso, se utiliza el optimizador Adam, que es un optimizador adaptativo que combina los optimizadores AdaGrad y RMSProp.

3.8. Métricas

Basado en la naturaleza del problema de predicción de riesgo de accidentes, se seleccionan métricas que capturan tanto la exactitud de las predicciones como la capacidad del modelo para identificar correctamente las áreas de alto riesgo.

3.8.1. Función de pérdida

En primer lugar, durante el entrenamiento se utiliza una función de Entropía Cruzada Ponderada (*Weighted Cross-Entropy Loss*) para manejar el desbalance de clases en los datos. La entropía cruzada es la métrica estándar que se utiliza en problemas de clasificación multiclase, y la versión ponderada permite asignar mayor importancia a las clases minoritarias (aquellas celdas que presentan riesgo de accidentes). La función puede definirse de la siguiente manera:

$$\mathcal{L}_{\text{WCE}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} w_c \cdot y_{i,c} \cdot \log(\hat{p}_{i,c}) \tag{3.3}$$

donde N es el número total de muestras, C es el número de clases de riesgo (en este caso, 3), w_c representa el peso asignado a la clase c, $y_{i,c}$ es el indicador binario que vale 1 si la clase verdadera de la muestra i es c y 0 en caso contrario, y $\hat{p}_{i,c}$ es la probabilidad predicha por el modelo para la clase c en la muestra i.

Los pesos de cada clase se obtienen de los parámetros definidos en la Tabla 3.6.

3.8.2. Evaluación

Una vez entrenado el modelo, se evalúa su desempeño en el conjunto de prueba utilizando las siguientes métricas presentadas en la Sección 2.4.3:

- ***Recall***: Mide la capacidad del modelo para identificar correctamente las celdas de alto riesgo. Es especialmente importante en este contexto, ya que se busca minimizar los falsos negativos (celdas de alto riesgo clasificadas como bajas).
- **MAP**: Proporciona una medida global de la precisión del modelo en todas las clases, considerando tanto falsos positivos como falsos negativos.
- **Exactitud** (*accuracy*): Mide la capacidad del modelo para clasificar correctamente una instancia de celda cualquiera. La utilizamos para medir cómo cambia esta métrica a medida que cambia el *recall*.

Estas métricas son aplicables para este trabajo final de carrera, ya que el objetivo es detectar las celdas donde realmente ocurrirán accidentes, por lo que es importante minimizar los falsos negativos. Si el número de falsos positivos es relativamente alto, se considera un problema secundario: es preferible, en términos de seguridad de vial, asegurarnos de detectar la mayor cantidad de accidentes posibles, aunque esto implique generar algunas alertas falsas.

3.9. Desarrollo de la aplicación web

En esta tesina final de carrera se busca no solo evaluar el rendimiento cuantitativo de los modelos propuestos en este capítulo, sino también desarrollar una aplicación web para visualizar los resultados y facilitar la interpretación y el análisis por parte de usuarios no técnicos. A su vez, el desarrollo de esta aplicación incluye el diseño de un esquema de retroalimentación del modelo, que ingiere constantemente nuevos datos de accidentes para mejorar su desempeño a lo largo del tiempo.

Esta aplicación muestra como funcionalidad principal un mapa interactivo de la ciudad de Mendoza, en la que puede apreciarse, de manera superpuesta, el mapa de riesgo predicho por el modelo, así como los puntos de interés (POI) que se consideran en el modelo. De esta forma, se busca facilitar la interpretación de los resultados y permitir a los usuarios identificar patrones espaciales en las predicciones.

Cabe recalcar que la aplicación web hace referencia tanto a la interfaz de usuario (el *frontend*) como a la lógica de negocio y la gestión de los modelos de ML (el *backend*).

3.9.1. Arquitectura de la aplicación web

La Figura 3.14 muestra la arquitectura completa la aplicación. Al comienzo de la imagen, se observa que los usuarios interactúan con el *frontend*, una aplicación web desarrollada con un *framework* moderno que permite a los usuarios visualizar mapas interactivos con las predicciones de riesgo y POI que ayudan a interpretar los resultados. Para interactuar con este componente, las solicitudes pasan por dos capas intermedias:

la capa de Cortafuegos de Aplicación Web (*Web Application Firewall*, WAF), que aporta seguridad a nivel de aplicación y métricas, y un *reverse proxy* que se encarga de enrutar las solicitudes dentro del servidor.

Este servidor utiliza *Docker Compose* para orquestar los distintos contenedores que componen la aplicación. En este servidor se encuentran alojados tanto el *backend* como el *frontend*: ambos se comunican entre sí a través de la red interna de Docker, lo que permite una comunicación eficiente y segura.

La aplicación *backend* se encarga de manejar la lógica interna de la aplicación, particularmente la gestión de los modelos de ML. Este último bloque fue desarrollado como un módulo independiente, de manera que pueda utilizarse conectado al servidor web o de forma autónoma para agilizar la experimentación y el desarrollo local. Este *backend* expone una API REST que permite solicitar predicciones, obtener datos de contexto y solicitar un reentreno de los modelos usando los datos más recientes.

El bloque de ML está compuesto por el acceso a los modelos propiamente dichos y también por un *pipeline* de ETL, que se permite construir el conjunto de datos bajo demanda. Nuevamente, este bloque se diseñó de manera modular para permitir su uso tanto dentro del servidor web como de forma autónoma, y realiza solicitudes a servicios externos para obtener datos meteorológicos y de accidentes actualizados.

3.9.2. Selección de tecnologías

En esta sección se describen las tecnologías seleccionadas para cada componente del aplicación, junto con una justificación de su elección basada en los requisitos del trabajo final de carrera. Es importante mencionar que se prioriza la elección de software libre siempre que sea posible.

Frameworks para ML

A la hora de desarrollar y entrenar modelos de ML, se elige un *framework* de Python ampliamente utilizado en la comunidad, que ofrezca flexibilidad para implementar arquitecturas personalizadas y soporte para entrenamiento en GPU. Considerando esto, se elige PyTorch [48] como la base para el desarrollo de los modelos de DL. Hoy en día, PyTorch es el *framework* más utilizado en investigación y desarrollo de modelos de DL, ofreciendo una amplia variedad de herramientas por defecto que permiten construir arquitecturas de redes neuronales a partir de bloques básicos.

La Tabla 3.7 muestra una comparativa entre los principales *frameworks* de ML para Python en, según su popularidad [49].

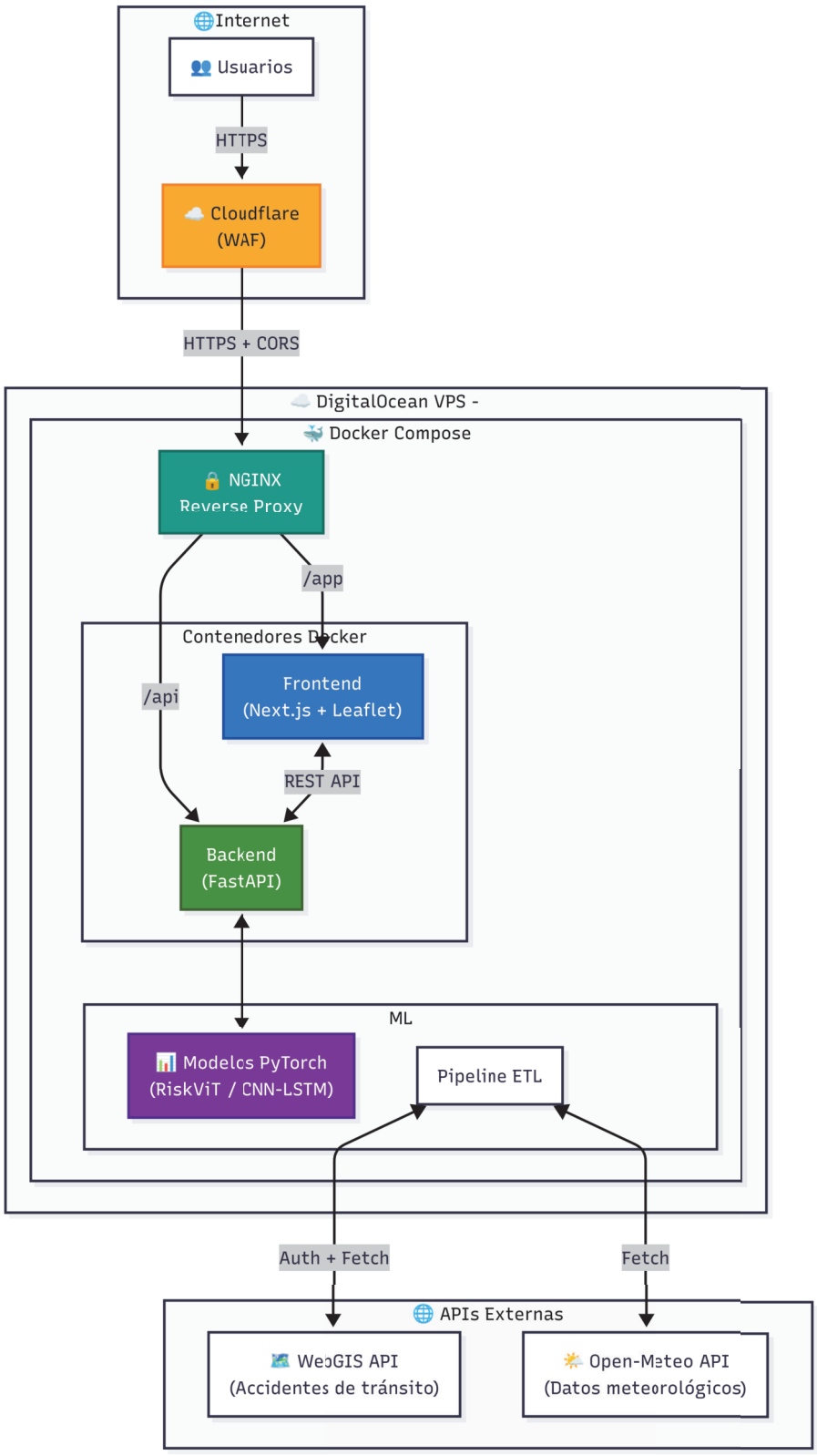


Figura 3.14: *Arquitectura completa de la aplicación web*

Tabla 3.7: *Cantidad de descargas mensuales y popularidad de los principales frameworks de ML en Python (Diciembre 2025)*

Framework	Cantidad de descargas
PyTorch	≈ 60 000 000
TensorFlow	≈ 20 000 000
Keras	≈ 15 000 000

Frontend

En cuanto a el desarrollo del *frontend*, se opta por utilizar Next.js [50], un *framework* basado en React que permite construir aplicaciones Finalmente, el *frontend* es una .web modernas con capacidades de SSR. Esta elección se basa principalmente en dos factores:

- **Experiencia con el *framework***: dado que anteriormente contaba con experiencia profesional desarrollando aplicaciones con Next.js, se consideró que esta elección permitiría acelerar el desarrollo y reducir la curva de aprendizaje, para poder enfocarse en los aspectos específicos del proyecto.
- **Capacidades de SSR**: la capacidad de renderizar páginas en el servidor acelera la carga del mapa inicial, mejorando la experiencia del usuario.

Visualización cartográfica

Dentro del *frontend*, se utiliza la librería Leaflet [51] para la visualización de datos geoespaciales. Leaflet es una librería de código abierto ampliamente utilizada para crear mapas interactivos en aplicaciones web. La misma permite utilizar una gran variedad de fuentes de mapas base, entre los que se destaca OpenStreetMap [52], un sistema de datos abiertos impulsado por la comunidad. Una característica importante de Leaflet es que permite combinar mapas base con capas de datos personalizadas, una funcionalidad esencial para poder superponer la cartografía de la ciudad de Mendoza con las predicciones de riesgo generadas por los modelos de ML.

Backend

El desarrollo del *backend* es necesario para poder conectar la visualización cartográfica con los modelos de ML. Para esta tarea, se elige FastAPI [53], un *framework* moderno para construir APIs en Python. Este componente de la arquitectura es responsable de comunicar al *frontend* con los modelos de forma segura, de manera que el usuario final no pueda acceder a todos los conjuntos de datos. También se utilizan *cronjobs* (tareas programadas) [54] para automatizar la obtención de nuevos datos y el reentrenamiento periódico de los modelos.

El backend fue construido siguiendo una arquitectura de capas que permite separar las responsabilidades entre presentación y lógica de negocios. La arquitectura se muestra en la Figura 3.15.

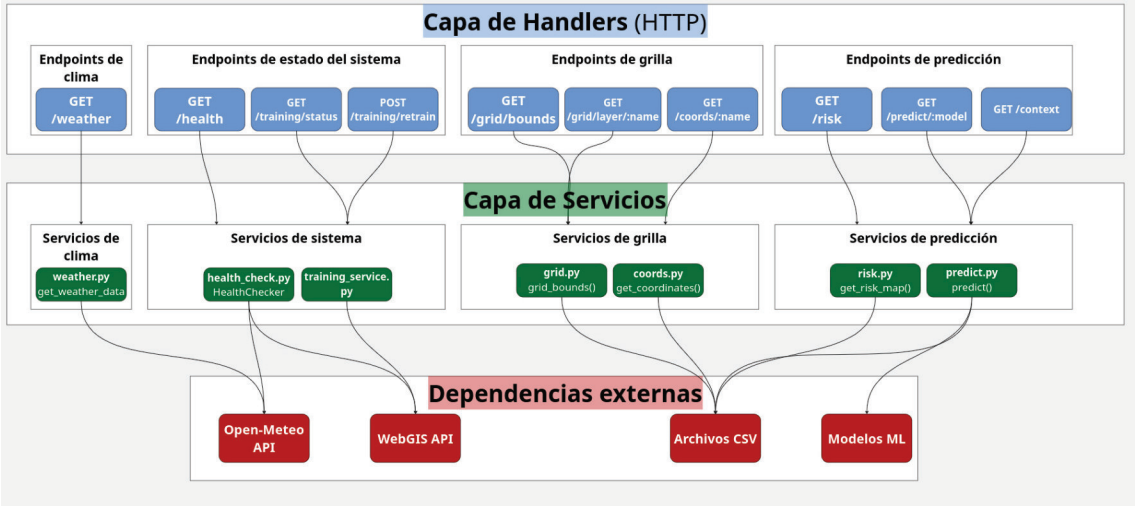


Figura 3.15: *Arquitectura del backend*

La capa de *handlers* recibe solicitudes *http*, las decodifica y se encarga de llamar a las funciones de lógica de negocios correspondientes. A su vez, formatea la respuesta para enviarla de vuelta al cliente y aplica seguridad a nivel de aplicación, como la implementación de políticas Intercambio de Recursos de Origen Cruzado (*Cross-Origin Resource Sharing*, CORS) (que solo permiten solicitudes desde dominios conocidos) y la validación de contraseña para reentrenar los modelos.

La capa de servicios se encarga de gestionar la lógica de negocios, agrupando operaciones de manejo de los modelos de ML y el pipeline de construcción del conjunto de datos. Esta capa se comunica con una capa final de clientes externos. La misma abstrae las fuentes de datos externas (como APIs de accidentes y meteorología) y los servicios de almacenamiento (como bases de datos o sistemas de archivos). De esta forma, se logra una arquitectura modular y desacoplada, que facilita el mantenimiento y la escalabilidad de la aplicación.

La documentación del *backend* fue construida usando el estándar OpenAPI. Se puede acceder a ella a través del siguiente enlace: `https://amaglione.tech/tesina/docs`.

3.9.3. Integración con fuentes de datos externas

Para poder alimentar el pipeline dentro de la aplicación web con datos actualizados de accidentes de tránsito y condiciones meteorológicas, es necesario acceder a fuentes de datos externas a través de APIs. En primera instancia, este proyecto fue diseñado para utilizar datos estáticos en forma de archivos CSV, pero este enfoque sólo permitía desarrollar la aplicación como una prueba de concepto. Para lograr un sistema funcional y que pueda ser utilizado en un entorno real, es fundamental que los datos se accedan dinámicamente.

API de accidentes de tránsito (WebGIS)

La principal fuente de datos de accidentes de tránsito es la API proporcionada por el portal WebGIS del Gobierno de Mendoza. Esta API ofrece acceso a los registros de accidentes en formato JSON, permitiendo filtrar por rango de fechas y obtener datos actualizados periódicamente.

Para poder utilizar esta API, es necesario implementar un proceso de autenticación mediante tokens: en primera instancia, debe hacerse una solicitud de autenticación con credenciales válidas para obtener un token de acceso. El mismo viene asociado a una fecha de vencimiento, a partir de la cual debe renovarse. Para encapsular este comportamiento, se diseña un módulo específico dentro del *pipeline* de construcción del conjunto de datos, que se encarga de gestionar la autenticación y realizar las solicitudes a la API.

Los datos disponibles en esta fuente se pueden adaptar fácilmente para respetar la misma configuración que se muestra en la Tabla 3.1, con la ventaja de que se puede solicitar a través de la API solamente los campos necesarios. En el Ejemplo 1 se muestra un accidente en formato JSON:

Ejemplo 1: *Registro de accidente en formato JSON provisto por la API WebGIS.*

```
{
    "objectid": 9708,
    "fecha_siniestro": 1768999500000,
    "tipo_accidente": "COLISIÓN",
    "coordenadas": "-68.855887438, -32.888849605",
    "calle": "MARTINEZ DE ROZAS",
    "altura": 1000
}
```

API meteorológica (Open-Meteo)

Para obtener datos meteorológicos históricos y pronósticos, se utiliza la API de Open-Meteo [55]. Este es un proyecto de software libre que ofrece acceso gratuito a datos meteorológicos globales para uso no-comercial.

Se elige utilizar este servicio por sobre los datos del Servicio Meteorológico Nacional Argentino (SMN) debido a que el SMN no ofrece una API pública para acceder a datos históricos de manera gratuita (aunque se pueden obtener solicitándolos). Open-Meteo, en cambio, es un estándar de la industria a la hora de trabajar con datos meteorológicos, y permitiría adaptar este proyecto a otras ciudades en el futuro sin depender de fuentes de datos locales.

3.9.4. Preparación para entorno de producción

Una vez verificado el funcionamiento de la aplicación web y la correcta conexión entre los componentes principales de su arquitectura, se llevan a cabo algunas modificaciones que permitan desplegar la aplicación en un entorno de producción más fácilmente.

En primer lugar, se utiliza *Docker* [56] plataforma de contenedorización, empaquetando tanto la aplicación frontend como backend de manera que la gestión del entorno de ejecución no sea un problema. Se utilizan variables de entorno para la configuración de ambos servicios.

La conexión entre los componentes se lleva a cabo utilizando *Docker Compose* [57]. Esta tecnología permite gestionar aplicaciones multicontenedor fácilmente, a partir de un solo archivo de configuración escrito de manera declarativa en formato YAML.

Las dos aplicaciones contenedorizadas se exponen a través del *reverse proxy Nginx* [58], que se encarga de mapear distintos subenlaces del host a puertos de los contenedores. También se toman medidas fundamentales de seguridad informática entre las que se destacan la creación de un certificado SSL que solo permita conexiones HTTPS a las aplicaciones, la implementación de políticas CORS y la instalación de Cloudflare como WAF [59].

Todo este ambiente de producción se despliega en la nube utilizando un Servidor Privado Virtual (*Virtual Private Server*, VPS) de la empresa DigitalOcean, con sistema operativo Ubuntu Server [60].

Optimizaciones de rendimiento

En esta etapa también se busca implementar algunas optimizaciones de rendimiento que mejoren la experiencia de usuario a la hora de utilizar la aplicación. En primer lugar, cuando se lanza el servicio de *backend*, se cargan ambos modelos en memoria y se mantienen activos. A su vez, se verifica que el conjunto de datos más reciente esté disponible para realizar predicciones inmediatas. Si el conjunto de datos no está disponible, se lanza automáticamente el *pipeline* de construcción del conjunto de datos para generarlo, de manera que el primer usuario no deba esperar a que todo el proceso se complete para poder utilizar la aplicación. También se agrega una mejora de rendimiento a la aplicación *frontend*: utilizando SSR, se obtienen las coordenadas de la grilla espacial y la ubicación de los POI directamente desde el servidor. Esto evita que la aplicación del lado del cliente deba hacer múltiples solicitudes de objetos pesados, lo que mejora significativamente el tiempo de carga inicial del mapa.

3.9.5. Entrenamiento automático

A pesar de que el modelo se evalúa en primera instancia utilizando un período fijo de fechas, se busca extender y generalizar este comportamiento para que el mismo pueda seguir aprendiendo de los datos sin necesidad de realizar acciones manuales. Una vez que el modelo se encuentra funcionando detrás de la capa de *backend*, se añade una funcionalidad que ejecuta automáticamente el proceso de entrenamiento de manera mensual, capturando lo datos más recientes de accidentes y meteorología. Si se tiene en cuenta que una de las limitaciones principales del desarrollo es la escasez de datos (tanto en la riqueza de los mismos como en el período de tiempo utilizado), se puede esperar que las predicciones sean más certeras a medida que el la cantidad de datos aumenta. También se disponibiliza una forma de realizar esta acción de manera manual.

4

Resultados y discusiones

En este capítulo se presentan y analizan los resultados obtenidos al entrenar y evaluar los modelos propuestos en este trabajo. En la sección 4.1 se muestran los resultados del modelo *baseline*, que sirve como referencia para evaluar el desempeño de los modelos de DL. Luego, en las secciones 4.3 y 4.4 se presentan los resultados obtenidos con los modelos RiskViT y CNN+LSTM, respectivamente. Posteriormente, en la sección 4.9, se realiza una comparación entre ambos modelos y con el estado del arte en la literatura. Para finalizar, se presenta el resultado de la implementación de la aplicación web que permite interactuar con los modelo (sección 4.11).

4.1. Resultados del modelo *baseline*

Como punto de partida, se utiliza el modelo *baseline* definido en la Sección 3.5 para evaluar una métrica inicial del rendimiento de los modelos. El mismo se ejecuta únicamente sobre el conjunto de datos de prueba, ya que no requiere ningún tipo de entrenamiento previo.

La Tabla 4.1 muestra las métricas obtenidas en este experimento.

4.1.1. Análisis de resultados

Los resultados obtenidos confirman algunas de las hipótesis planteadas en capítulos anteriores de esta tesina. En primer lugar, se puede apreciar que el desbalance de clases influye fuertemente en métricas como el *accuracy*. Un 90 % de exactitud significa que la enorme mayoría de las celdas no cambia de un día al otro, sino que se mantienen

Tabla 4.1: *Métricas del modelo base (persistencia del día anterior)*

Métrica	Valor
Accuracy	0.9091
Recall	0.0765
MAP	0.0352
F1	0.3620
Precision	0.3619
AUC	0.5294

constantes en el tiempo. Si consideramos la Figura 3.11, en la que se identifica un 95 % de celdas pertenecientes a la clase de riesgo bajo, la alta exactitud alcanzada por el modelo *baseline* puede explicarse por la mayoría de celdas de la clase 0.

Por otro lado, el *recall* permanece extremadamente bajo, con un valor cercano al 8 %. La interpretación de esta métrica es que solo en el 8 % de los casos se detectan aquellas celdas con un riesgo significativo. El hecho de el *recall* tenga un valor bajo en el modelo *baseline* significa que la ubicación de los accidentes tiene una variabilidad temporal considerable: los patrones de la serie de tiempo necesitan un modelo complejo para ser capturados.

A partir de estos resultados, podemos observar que el modelo no reporta resultados satisfactorios a nivel de la métrica *recall*, lo que sugiere que no está capturando la variabilidad propia de datos espacio-temporales. Por este motivo se introduce el uso de dos modelos de DL, cada uno con sus propias características.

4.2. Resultados del modelo Random Forest

Antes de entrenar los modelos de DL, se llevó a cabo un experimento con modelos de *Random Forest* para establecer una línea base adicional que permita evaluar el impacto del aprendizaje profundo en la tarea. Este modelo, más sencillo utiliza el mismo conjunto de datos y las mismas variables de entrada que los modelos de DL, con la diferencia de que, en lugar de realizar una única predicción global del mapa de riesgo de la Ciudad de Mendoza, solo tiene la capacidad de predecir el riesgo de una única celda a la vez.

Se realizaron 20 experimentos variando hiperparámetros como el número de árboles y la profundidad máxima. La Tabla 4.2 muestra un resumen de los resultados obtenidos.

Tabla 4.2: *Resultados del modelo Random Forest*

Configuración	Accuracy	Recall	MAP	F1	Precision	AUC
Sin datos contextuales	0.7027	0.1094	0.0479	0.3252	0.3506	0.7030
Con datos contextuales	**0.9466**	**0.1607**	**0.0793**	**0.3410**	**0.4197**	**0.7504**

4.2.1. Análisis de resultados

La mejora en las métricas clave es significativa con respecto a el modelo *baseline*, lo que confirma que la inclusión de variables contextuales aporta información relevante para la predicción del riesgo de accidentes. A su vez, este modelo sirve como una mejor línea de base para evaluar el impacto de los modelos de DL, que se espera que puedan capturar patrones más complejos y dependencias espacio-temporales que un modelo de *Random Forest* no puede modelar.

La Figura 4.1 muestra la curva ROC del modelo Random Forest, que presenta un AUC de 0.7504.

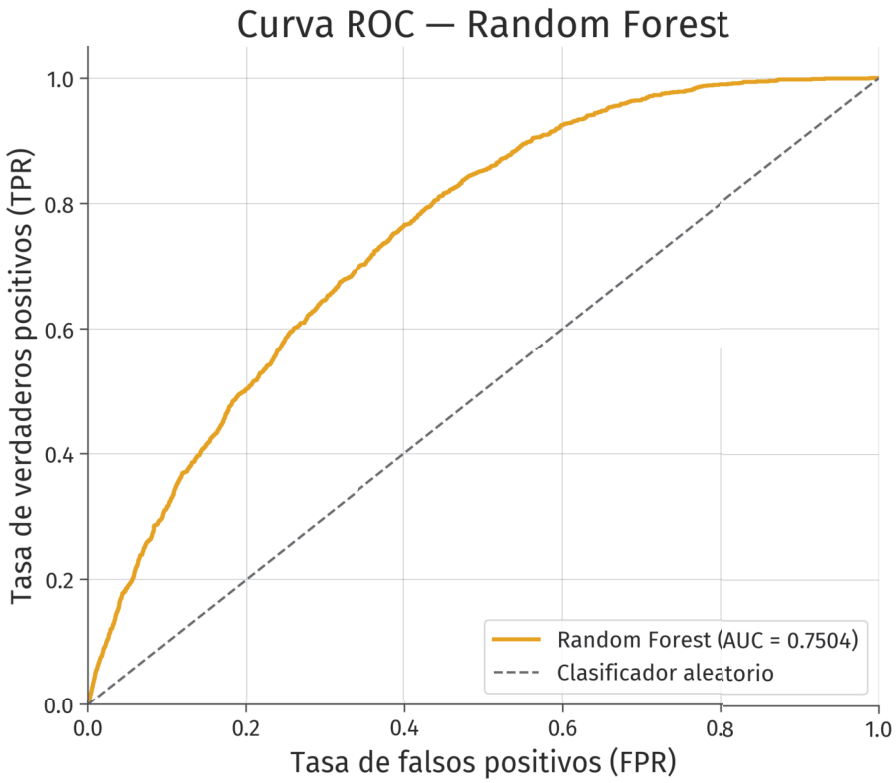


Figura 4.1: *Curva ROC del modelo Random Forest sobre el conjunto de prueba*

4.3. Resultados del modelo RiskViT

En esta sección se presentan los resultados obtenidos al entrenar el modelo RiskViT para la tarea de predicción de riesgo en una grilla espacial de 12×12 celdas. Se realizaron 1145 experimentos variando hiperparámetros, como se muestra en la Sección 3.6, y se reportan las métricas finales sobre el conjunto de prueba.

Las Tablas 4.3 y 4.4 muestran resúmenes agregados para analizar el impacto del *learning rate* y del *patch size* en el rendimiento. La Tabla 4.5 lista las configuraciones con mejor desempeño por *recall*.

Para ver una lista de todas las ejecuciones realizadas, con sus respectivos hiperparámetros y métricas de evaluación, se recomienda consultar el Anexo A, que contiene tablas detalladas con esta información.

Tabla 4.3: *Resumen de métricas por learning rate* (*ViT, grilla* 12×12)

learning_rate	N	MAP promedio	MAP máximo	*recall* promedio	*recall* máximo	*accuracy* promedio
10^{-5}	381	**0.0903**	0.0915	**0.1589**	0.1686	0.9479
10^{-4}	382	0.0900	0.0922	0.1579	0.1711	0.9476
10^{-3}	382	0.0884	**0.0933**	0.1567	**0.1807**	0.9383

Las métricas corresponden a la evaluación final sobre el conjunto de prueba.

Tabla 4.4: *Resumen de métricas por patch size* (*ViT, grilla* 12×12)

patch_size	N	MAP promedio	MAP máximo	*recall* promedio	*recall* máximo	*accuracy* promedio
1	432	**0.0896**	**0.0933**	**0.1580**	**0.1807**	0.9445
4	281	0.0896	0.0932	0.1580	0.1783	0.9455
2	432	0.0894	0.0928	0.1576	0.1706	0.9442

Tabla 4.5: *Top 10 configuraciones por Recall* (*ViT, grilla* 12×12)

model_id	learning_rate	batch_size	dim	depth	heads	mlp_dim	patch_size	*recall*	MAP	*accuracy*
`11b20bae`	10^{-3}	8	128	8	8	64	1	**0.1807**	0.0933	0.9257
`5ed034bc`	10^{-3}	8	128	8	8	128	4	0.1783	0.0888	0.9458
`c12714f7`	10^{-3}	16	128	8	4	128	1	0.1770	0.0918	0.9426
`d80dabfd`	10^{-3}	16	128	4	8	64	1	0.1765	0.0933	0.9335
`a939d712`	10^{-3}	16	64	4	8	32	1	0.1758	0.0918	0.9479
`2e06e128`	10^{-3}	16	64	8	8	64	1	0.1740	0.0929	0.9480
`157489e3`	10^{-3}	16	128	4	4	128	1	0.1724	0.0915	0.9193
`c9b03d6b`	10^{-3}	16	128	8	4	32	4	0.1724	0.0916	0.9426
`4b3504d9`	10^{-3}	16	128	4	4	32	1	0.1717	0.0903	0.9279
`0ad8a7be`	10^{-3}	8	32	8	8	32	4	0.1712	0.0932	0.9480

4.3.1. Mejor modelo y curvas de entrenamiento

Al analizar el barrido de hiperparámetros, la mejor configuración obtenida, maximizando el *recall* corresponde al modelo `11b20bae` (Tabla 4.5). Esta configuración obtiene **Accuracy=0.9257**, **Recall=0.1807**, **MAP=0.0933**, **F1=0.3724**, **Precision=0.3821** y **AUC=0.7822** en el conjunto de prueba, superando sustancialmente al *baseline* de la Sección 4.1.

Durante el entrenamiento se utilizó *early stopping* (*patience*=10, *min_delta*=0.001). En este caso, el criterio se activó en la época 22, algo esperable en un conjunto de datos pequeño [24]. La Figura 4.2 muestra las curvas de la función de pérdida para entrenamiento y validación del mejor modelo.

A partir de la curva obtenida, se pueden obtener conclusiones con respecto a la convergencia del modelo. En primer lugar, se puede apreciar que la pérdida baja rápidamente en el conjunto de datos de entrenamiento, y se mantiene estable en validación. Desde la época 15, la pérdida de validación comienza a subir ligeramente, lo que activa el criterio de *early stopping* en la época 22. Esto puede ser un indicio de que el modelo comienza a sobreajustar el conjunto de entrenamiento: a pesar de esto, este modelo obtuvo

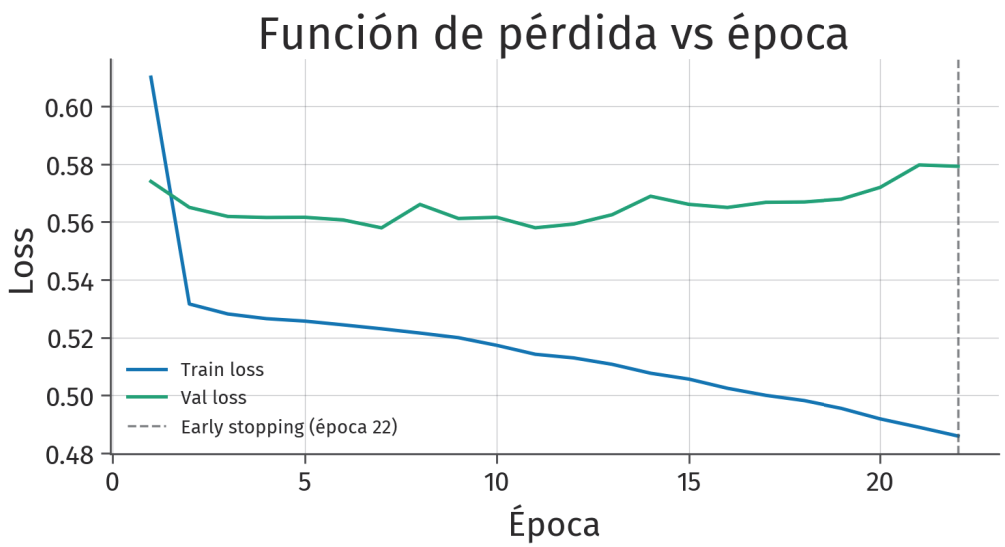


Figura 4.2: *Curvas de pérdida en entrenamiento y validación para el mejor modelo RiskViT (grilla* 12×12*)*

los mejores resultados en el conjunto de prueba a diferencia de otros que se entrenaron durante más épocas o presentaron una menor cantidad de parámetros.

La Figura 4.3 muestra la curva ROC del mejor modelo RiskViT. Con un AUC de 0.7822, el modelo demuestra una capacidad de discriminación superior a la de un clasificador aleatorio (AUC = 0.5) y ligeramente superior a la conseguida por los modelos *baseline* y *Random Forest*.

4.3.2. Estudio de ablación

Para validar la importancia de cada una de las variables de entrada al modelo, se agrupan las mismas en las siguientes categorías:

- **Riesgo:** ventanas de riesgo históricas para cada una de las celdas.
- **POI:** características relacionadas con puntos de interés.
- **Clima:** características meteorológicas (temperatura, precipitación, viento).
- **Temporal:** características de calendario (día de la semana, feriados, período escolar).

Se toman los 10 modelos con mejor desempeño en *recall* de la Tabla 4.5 y, para cada una de las configuraciones de hiperparámetros, se entrenan variantes con las siguientes combinaciones de características:

- `Sin clima`: se excluyen las características meteorológicas.
- `Sin POI`: se excluyen los puntos de interés.

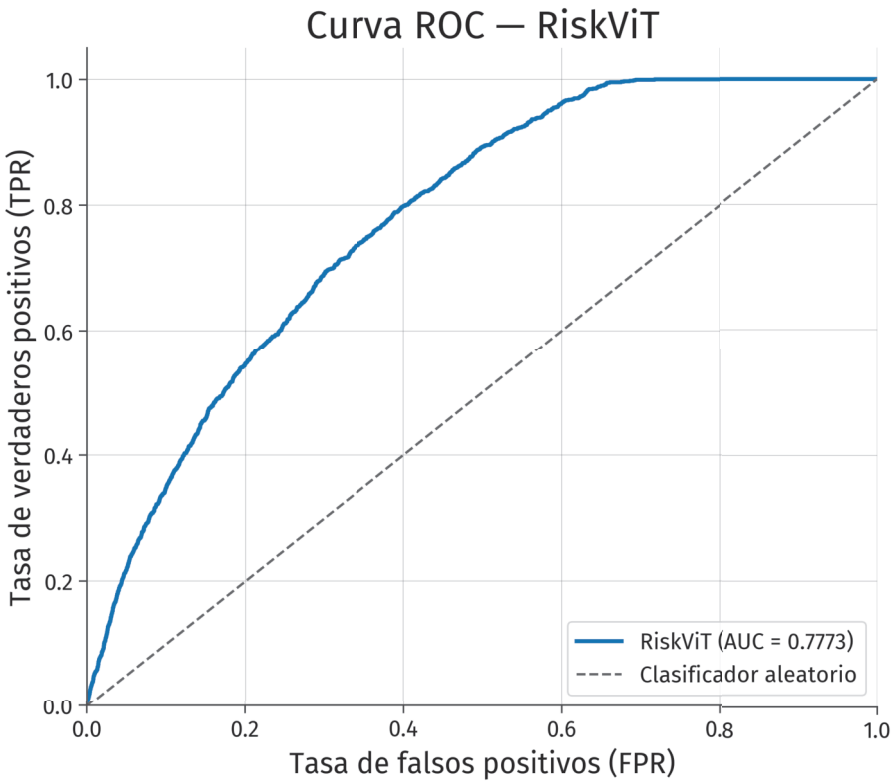


Figura 4.3: *Curva ROC del mejor modelo RiskViT (`11b20bae`) sobre el conjunto de prueba*

- `Sin riesgo`: se excluyen las ventanas de riesgo históricas.
- `Sin temporal`: se excluyen las características de calendario.
- `Solo riesgo`: se utilizan únicamente las ventanas de riesgo histórico.
- `Solo POI+clima`: se utilizan únicamente POI y características meteorológicas.
- `Completo`: modelo entrenado con todas las variables (referencia).

La Figura 4.4 muestra la distribución del *recall* para cada grupo de ablación sobre las 10 configuraciones evaluadas. Los grupos están ordenados de menor a mayor *recall* máximo, siendo el grupo `Completo` el de referencia y el que aparece a la derecha.

El análisis de la figura permite extraer conclusiones sobre la relevancia de cada categoría de variables. El grupo de modelos que utiliza todas las variables obtiene las métricas más elevadas tanto en el caso promedio como en el mejor caso, lo que indica que el modelo se beneficia de la combinación de todas las fuentes de información disponibles. Sin embargo, existen otras combinaciones que también obtienen resultados considerablemente buenos. En particular, se aprecia que el grupo que no incluye las ventanas de riesgo tiene un rendimiento similar al conjunto de datos completo, mientras que la combinación que no incluye datos climáticos ve una disminución importante en el rendimiento. Esto sugiere que el modelo RiskViT no utiliza las ventanas de riesgo como

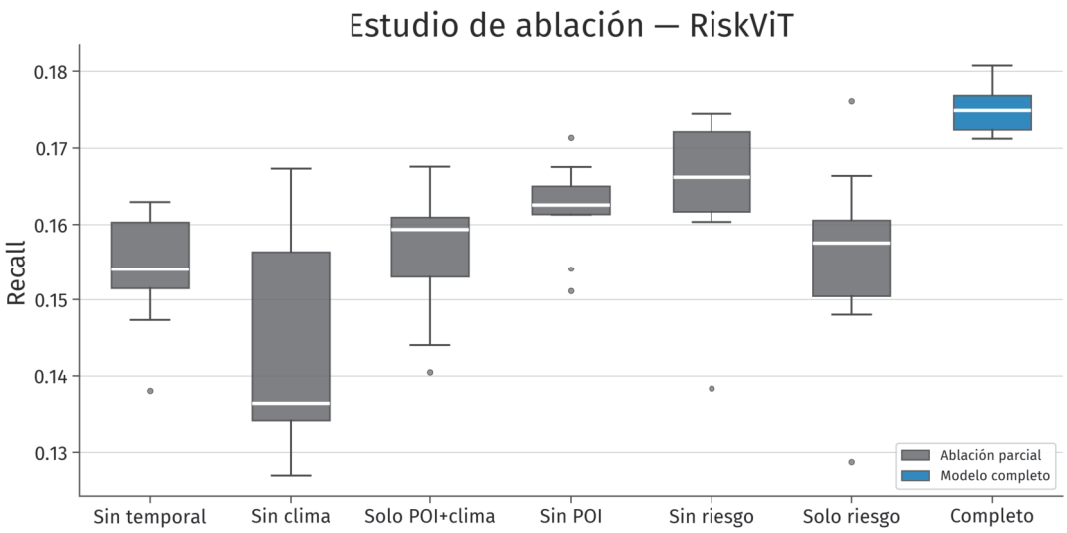


Figura 4.4: *Distribución del recall por grupo de ablación para RiskViT. El grupo* `Completo` *(azul) representa el modelo entrenado con todas las variables.*

fuente de información principal para realizar predicciones.

4.3.3. Análisis del tiempo de ejecución de RiskViT

Todos los experimentos de entrenamiento se llevaron a cabo en el clúster de cómputo de alto rendimiento Mendieta. Cada ejecución utilizó una fracción equivalente a un tercio de una GPU NVIDIA A30 (aproximadamente 8 GB de memoria de video), junto con recursos de CPU y memoria RAM compartidos, manteniendo el uso de RAM alrededor de los 2.5 GB. El mejor modelo, con **731 504 parámetros** entrenables, logró un tiempo de entrenamiento de aproximadamente 2.5 minutos.

La Figura 4.5 muestra la distribución del tiempo de entrenamiento de los 1145 experimentos realizados. Se puede observar que la mediana del tiempo de entrenamiento es de aproximadamente 1.5 minutos, con un rango que va desde menos de 1 minuto hasta aproximadamente 7 minutos. La variabilidad en los tiempos de entrenamiento se debe principalmente a las diferencias en la complejidad del modelo (número de capas, dimensión de las capas ocultas, etc.) y a la activación del *early stopping*.

4.4. Resultados del modelo CNN+LSTM

En esta sección se presentan los resultados obtenidos al entrenar el modelo CNN+LSTM para la tarea de predicción de riesgo en una grilla espacial de 12×12 celdas. Se realizaron 192 experimentos variando hiperparámetros, como se muestra en la Sección 3.6, y se reportan las métricas finales sobre el conjunto de prueba.

Las Tablas 4.6 y 4.7 muestran resúmenes agregados para analizar el impacto del *learning rate* y de la longitud de secuencia en el rendimiento. La Tabla 4.8 lista las confi-

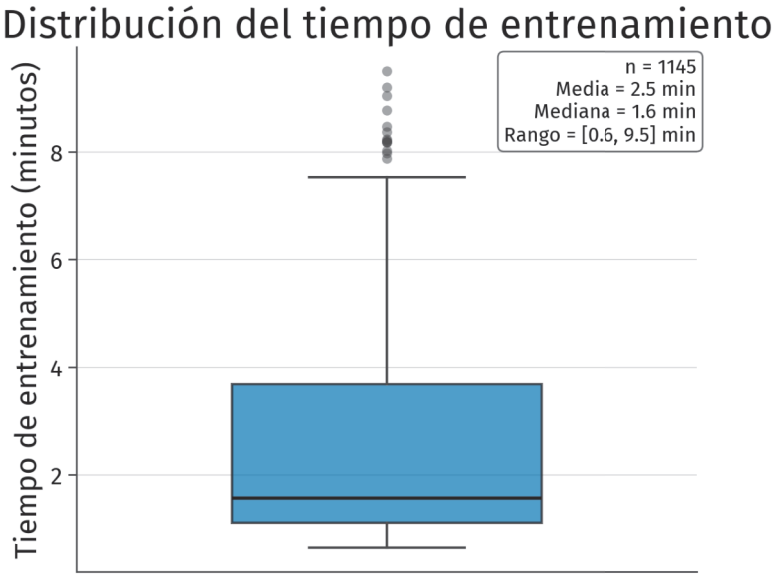


Figura 4.5: *Distribución del tiempo de entrenamiento para los 1145 experimentos del modelo RiskViT (grilla* 12×12*).*

guraciones con mejor desempeño por *recall*.

En el Anexo A se pueden encontrar tablas detalladas con todas las ejecuciones realizadas, incluyendo los hiperparámetros utilizados y las métricas de evaluación obtenidas para cada experimento.

Tabla 4.6: *Resumen de métricas por learning rate (CNN-LSTM, grilla* 12×12*)*

learning_rate	N	MAP promedio	MAP máximo	*recall* promedio	*recall* máximo	*accuracy* promedio
10^{-5}	64	**0.0900**	0.0928	0.1582	0.1740	0.9490
10^{-4}	64	0.0897	0.0927	**0.1582**	0.1690	0.9479
10^{-3}	64	0.0891	0.0949	0.1569	0.1833	0.9421

Las métricas corresponden a la evaluación final sobre el conjunto de prueba.

Tabla 4.7: *Resumen de métricas por longitud de secuencia (CNN-LSTM, grilla* 12×12*)*

sequence_length	N	MAP promedio	MAP máximo	*recall* promedio	*recall* máximo	*accuracy* promedio
7	48	**0.0905**	**0.0949**	**0.1605**	**0.1833**	0.9454
14	48	0.0903	0.0940	0.1598	0.1740	0.9469
1	48	0.0889	0.0923	0.1557	0.1686	0.9467
3	48	0.0887	0.0916	0.1553	0.1700	0.9463

4.4.1. Mejor modelo y curvas de entrenamiento

Al analizar el barrido de hiperparámetros, la mejor configuración obtenida, maximizando el *recall* corresponde al modelo `095d9665` (Tabla 4.8). Esta configuración obtiene **Accuracy=0.9206**, **Recall=0.1833**, **MAP=0.0949**, **F1=0.3819**, **Precision=0.3804** y **AUC=0.7774** en el conjunto de prueba, superando tanto al *baseline* de la Sección 4.1 como al modelo RiskViT presentado anteriormente.

Tabla 4.8: *Top 10 configuraciones por Recall (CNN-LSTM, grilla* 12×12*)*

model_id	learning_rate	batch_size	sequence_length	lstm_hidden_size	cnn_filters	*recall*	MAP	*accuracy*
`095d9665`	10^{-3}	8	7	128	[32, 64, 128]	**0.1833**	0.0949	0.9206
`e41829af`	10^{-5}	8	14	128	[32, 64]	0.1740	0.0928	0.9495
`20838ade`	10^{-3}	16	7	128	[32, 64, 128]	0.1735	0.0935	0.9492
`56b5dd4d`	10^{-3}	16	14	128	[32, 64]	0.1707	0.0915	0.9264
`89c73d5f`	10^{-3}	16	3	128	[32, 64]	0.1700	0.0907	0.9382
`c2f435a2`	10^{-4}	16	7	128	[32, 64]	0.1690	0.0911	0.9492
`a6017ccc`	10^{-5}	8	14	64	[32, 64, 128]	0.1687	0.0916	0.9496
`fec6dbf3`	10^{-4}	8	1	64	[32, 64]	0.1686	0.0899	0.9478
`d244cf4c`	10^{-5}	8	7	64	[32, 64, 128]	0.1685	0.0926	0.9484
`31b1403c`	10^{-3}	8	14	128	[32, 64]	0.1684	0.0940	0.9484

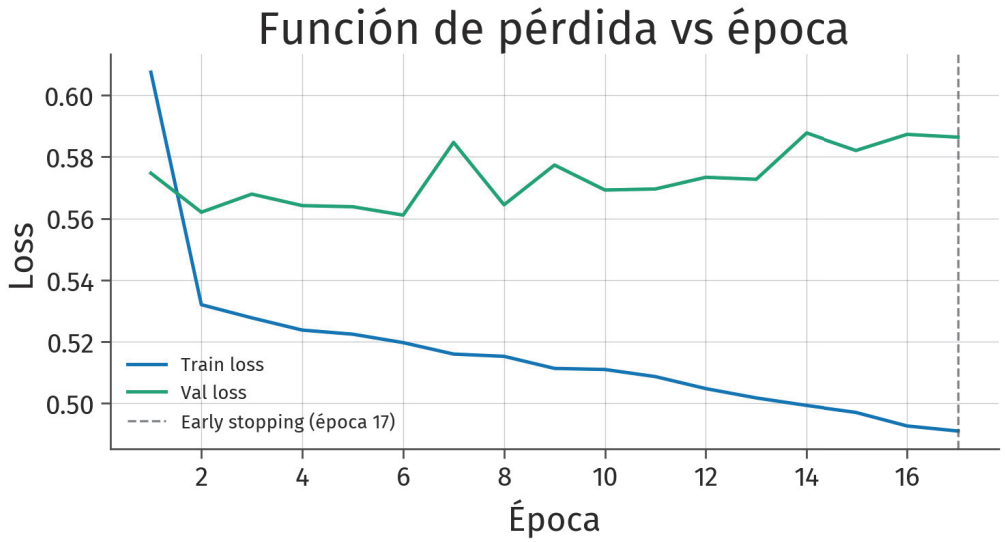


Figura 4.6: *Curvas de pérdida en entrenamiento y validación para el mejor modelo CNN+LSTM (grilla* 12×12*)*

A partir de la curva obtenida, se observa un comportamiento de convergencia similar al del modelo RiskViT: la pérdida disminuye rápidamente al inicio y luego se estabiliza, hasta que eventualmente la pérdida en validación comienza a subir. Esto sugiere que el modelo está sobreajustando a los datos de entrenamiento. Sin embargo, el modelo CNN+LSTM alcanza valores de *recall* y MAP ligeramente superiores en el conjunto de prueba, lo que es consistente con la hipótesis de que la arquitectura híbrida convolucional-recurrente modela con mayor eficacia las dependencias espacio-temporales del problema. En la sección 4.6 se realiza un análisis más detallado sobre el fenómeno de sobreajuste en ambos modelos.

La Figura 4.7 muestra la curva ROC del mejor modelo CNN+LSTM. Con un AUC de 0.7774, el modelo presenta una capacidad de discriminación comparable a la del modelo RiskViT, confirmando que ambas arquitecturas logran aprender patrones de riesgo de manera efectiva.

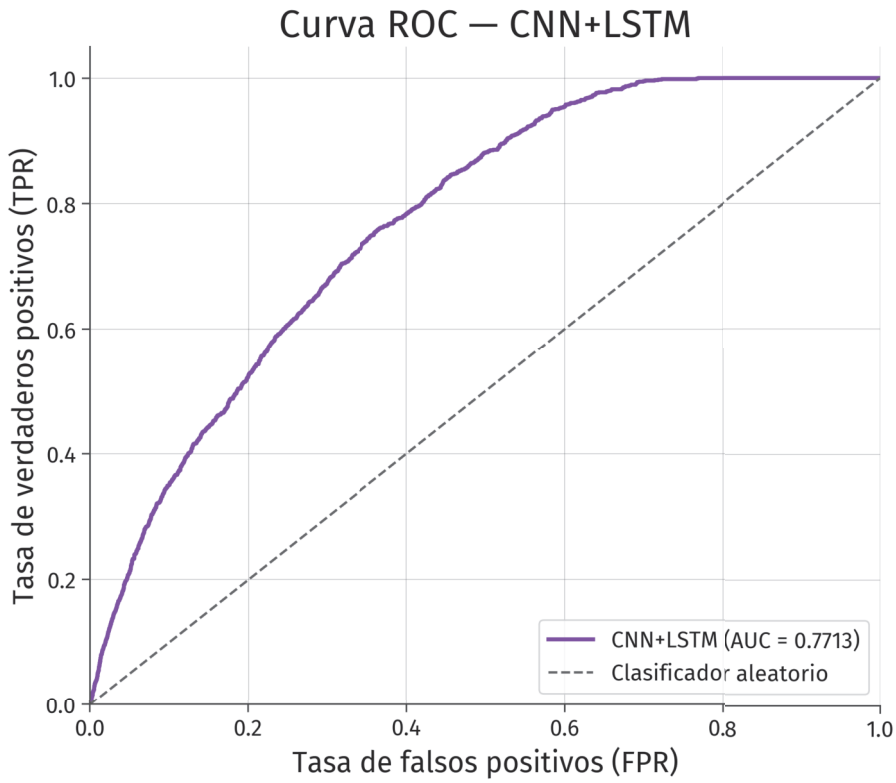


Figura 4.7: *Curva ROC del mejor modelo CNN+LSTM (`095d9665`) sobre el conjunto de prueba*

4.4.2. Análisis del tiempo de ejecución de CNN+LSTM

Al igual que con el modelo RiskViT, todos los experimentos de entrenamiento del modelo CNN+LSTM se llevaron a cabo en el clúster de cómputo de alto rendimiento Mendieta. El mejor modelo, con **1 020 048 parámetros** entrenables, requirió mayores tiempos de entrenamiento que el modelo RiskViT debido a la naturaleza secuencial de las RNN.

La Figura 4.8 muestra la distribución del tiempo de entrenamiento de los 192 experimentos realizados. Se puede observar que la mediana del tiempo de entrenamiento es de aproximadamente 6 minutos, con un rango que va desde poco más de 1 minuto hasta aproximadamente 56 minutos. La mayor variabilidad en los tiempos de entrenamiento, en comparación con el modelo RiskViT, se debe principalmente a las diferencias en la longitud de la secuencia temporal de entrada, ya que secuencias más largas requieren mayor procesamiento. Esto es consistente con la comparación realizada en el marco teórico, donde se resaltó la ventaja a nivel de eficiencia computacional de las arquitecturas basadas en *transformers* por sobre las basadas en RNN.

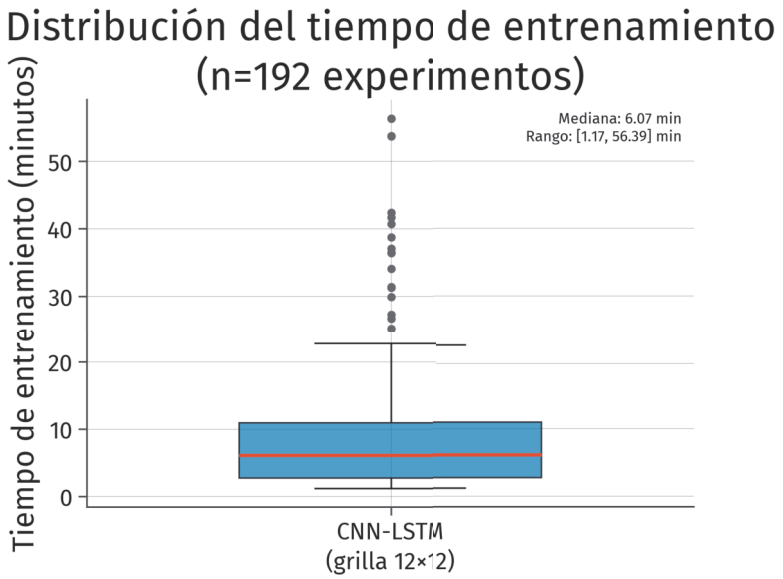


Figura 4.8: *Distribución del tiempo de entrenamiento para los 192 experimentos del modelo CNN+LSTM (grilla* 12×12*).*

4.4.3. Estudio de ablación

De manera análoga al estudio realizado para RiskViT (Sección 4.3.2), se realiza un estudio de ablación sobre los 10 modelos CNN+LSTM con mejor desempeño en *recall* de la Tabla 4.8. Se emplean las mismas combinaciones de características descritas anteriormente.

La Figura 4.9 muestra la distribución del *recall* para cada grupo de ablación.

El caso del modelo CNN+LSTM no es exactamente igual que el de RiskViT: existe una variabilidad mucho menor entre los diferentes grupos de ablación. Aunque el modelo con el máximo *recall* tanto en promedio como en máximo continúa siendo el que utiliza todas las variables, la diferencia con respecto a las otras configuraciones es mucho menor. Esto sugiere que el modelo CNN+LSTM es capaz de aprender patrones relevantes a partir de subconjuntos reducidos de características. Una explicación posible para esto es que, ya que este modelo procesa secuencias de mapas de riesgo, tiene acceso a un historial de datos contextuales que no están presentes en el caso de RiskViT.

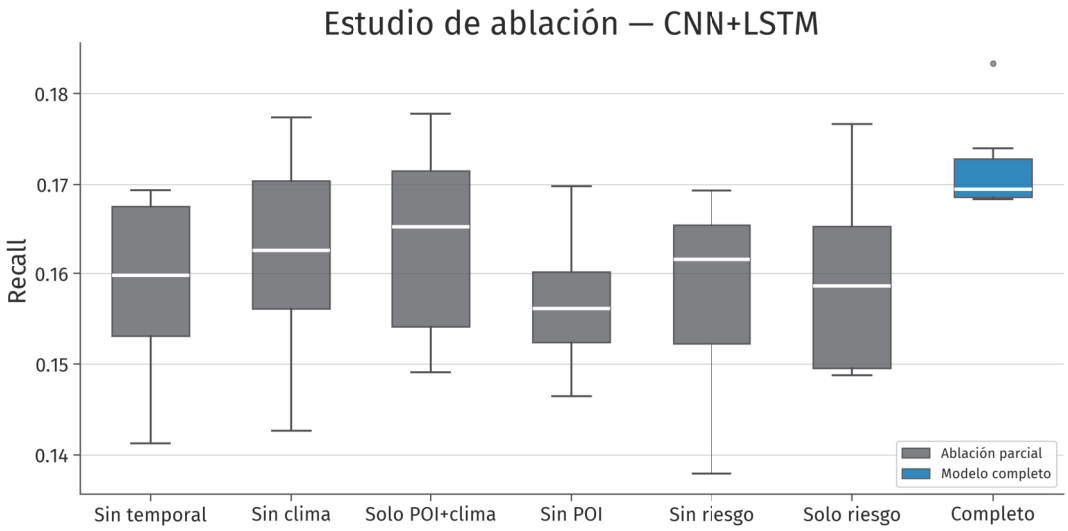


Figura 4.9: *Distribución del recall por grupo de ablación para CNN+LSTM. El grupo* `Completo` *(azul) representa el modelo entrenado con todas las variables.*

4.5. Validación Cruzada Espacial

Además de la división temporal del conjunto de datos en entrenamiento, validación y prueba, se aplica el concepto de validación cruzada espacial para evaluar la capacidad de generalización de los modelos a nuevas áreas geográficas dentro de la Ciudad de Mendoza. Para esto, se divide el mapa en regiones geográficas y se entrena el modelo utilizando datos de tres regiones mientras se evalúa en la región restante. Este proceso se repite para cada una de las regiones, obteniendo así una evaluación más robusta del desempeño del modelo en diferentes contextos espaciales.

En particular, se evalúan dos divisiones distintas. La primera de ellas, divide el área de estudio en **5 zonas**, de las cuales una corresponde al centro de la ciudad mientras que las demás corresponden a la periferia (norte, sur, este y oeste, como se ve en la Figura 4.10).

La segunda división, divide el área de estudio en **4 cuadrantes** (noroeste - NO, noreste - NE, suroeste - SO, sureste - SE), como se muestra en la Figura 4.11.

Dentro de las dos divisiones geográficas, los accidentes se distribuyen de manera heterogénea, lo que se espera impacte profundamente en los resultados. Las Tablas 4.9 y 4.10 muestran la distribución de accidentes para cada una de las zonas y cuadrantes, respectivamente. Nótese que los porcentajes en la Tabla 4.9 no suman a 100 % ya que las 5 zonas no son mutuamente excluyentes, sino que se superponen en sus límites.

En ambos casos, se entrena el modelo dejando uno de los segmentos fuera del conjunto de entrenamiento y se evalúa el desempeño en dicho segmento. Este proceso se repite para cada segmento, obteniendo así una evaluación más completa de la capacidad de generalización espacial del modelo. Las Tablas 4.11 y 4.12 muestran los resultados

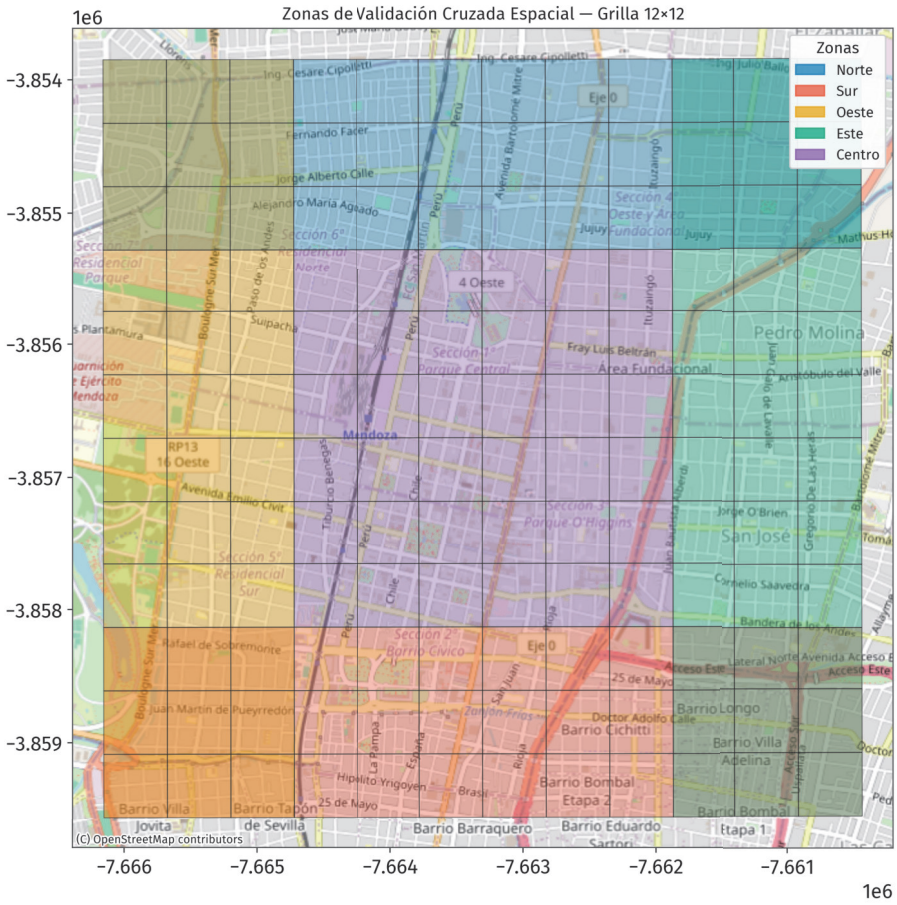


Figura 4.10: *División del mapa en 5 zonas para validación cruzada espacial*

Tabla 4.9: *Distribución de accidentes por zona*

Zona	Porcentaje de accidentes
Norte	17.06 %
Sur	22.88 %
Este	3.17 %
Oeste	17.68 %
Centro	48.66 %

Tabla 4.10: *Distribución de accidentes por cuadrante*

Cuadrante	Porcentaje de accidentes
Noroeste	38.41 %
Noreste	19.44 %
Suroeste	22.96 %
Sureste	19.19 %

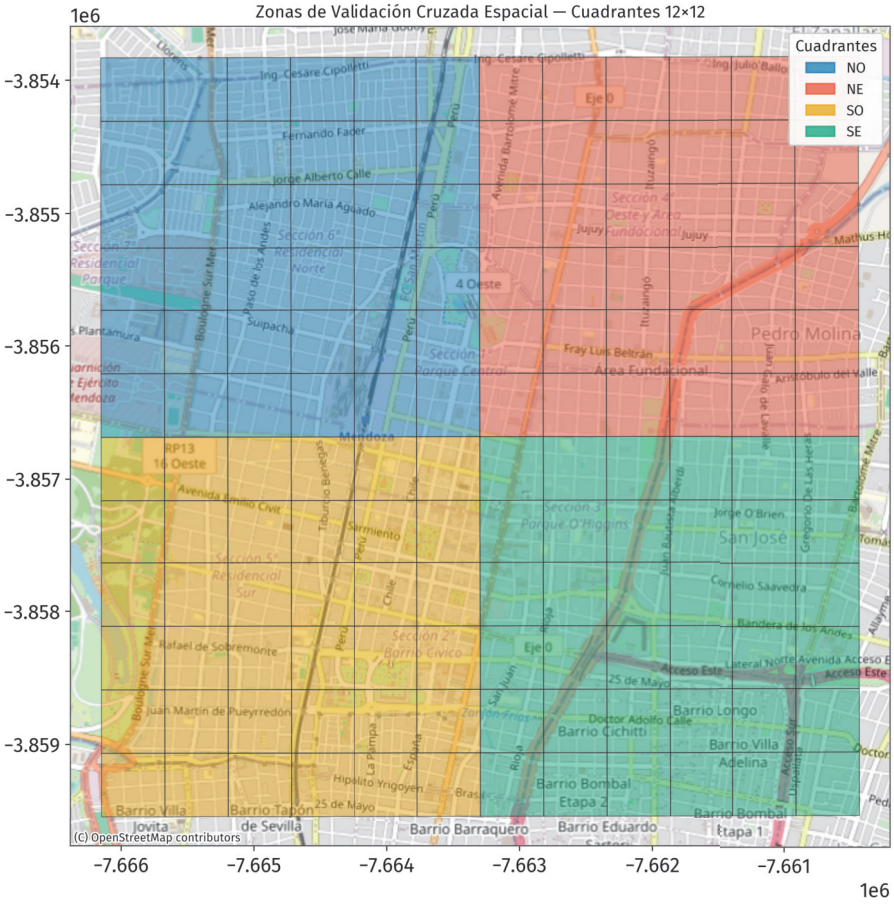


Figura 4.11: *División del mapa en 4 cuadrantes para validación cruzada espacial*

obtenidos para ambos modelos, evaluados *únicamente* sobre las celdas del segmento retenido (*zone metrics*).

Tabla 4.11: *Validación cruzada espacial - división en 5 zonas. Métricas evaluadas sobre las celdas del segmento retenido.*

Modelo	Zona	Accuracy	*recall*	MAP	F1	AUC
RiskViT	Norte	0.3851	0.0363	0.0190	0.1989	0.5485
	Sur	0.4090	0.1150	0.0673	0.2168	0.5948
	Este	0.4416	0.0000	0.0000	0.2110	0.4421
	Oeste	0.2406	0.0377	0.0160	0.1538	0.5314
	Centro	0.2454	0.0960	0.0449	0.1867	0.5035
CNN+LSTM	Norte	0.3335	0.0283	0.0134	0.1858	0.4755
	Sur	0.3274	0.0416	0.0306	0.1954	0.5582
	Este	0.4509	0.0147	0.0074	0.2164	0.7424
	Oeste	0.3181	0.0638	0.0421	0.1766	0.4813
	Centro	0.4140	0.0857	0.0414	0.2407	0.4707

Tabla 4.12: *Validación cruzada espacial - división en 4 cuadrantes. Métricas evaluadas sobre las celdas del segmento retenido.*

Modelo	Cuadrante	Accuracy	*recall*	MAP	F1	AUC
RiskViT	Noroeste	0.3673	0.0607	0.0253	0.2300	0.5135
	Noreste	0.2965	0.0433	0.0203	0.1761	0.4825
	Suroeste	0.3574	0.1203	0.0993	0.2068	0.5298
	Sureste	0.3062	0.0773	0.0616	0.1774	0.4644
CNN+LSTM	Noroeste	0.2842	0.1425	0.1087	0.1901	0.5581
	Noreste	0.2945	0.0195	0.0078	0.1772	0.4741
	Suroeste	0.2508	0.0391	0.0164	0.1623	0.5321
	Sureste	0.3760	0.0171	0.0151	0.2075	0.5439

Se puede apreciar una considerable caída del rendimiento al evaluar ambos modelos en segmentos geográficos que no formaron parte del conjunto de entrenamiento. Mientras que en la evaluación temporal estándar ambos modelos alcanzan valores de AUC en torno a 0.78, en la validación cruzada espacial los valores de AUC sobre la zona retenida caen al rango 0.44-0.59 para la división en 5 zonas y a 0.46-0.56 para la división en cuadrantes, valores cercanos a los de un clasificador aleatorio (AUC= 0.5). La caída en el *recall* es igualmente marcada: de ≈ 0.18 a valores inferiores a 0.14 en la validación espacial.

El rendimiento varía considerablemente entre zonas, lo que se explica en gran parte por la distribución heterogénea de los accidentes en el área de estudio. Alrededor del

50 % de los accidentes registrados corresponden a la zona central, mientras que tan solo el 3 % ocurrió en la zona este. Este desequilibrio repercute directamente en los resultados: para RiskViT, la zona este presenta un AUC de 0.44 (por debajo del azar) y un *recall* de 0.00, lo que indica que el modelo no logra identificar ninguna celda de riesgo en una zona que prácticamente no estaba representada durante el entrenamiento. Por el contrario, las zonas sur y suroeste, con mayor concentración de accidentes, obtienen los mejores valores de *recall* (0.115 y 0.120 respectivamente para RiskViT).

Un caso particular es la zona este del modelo CNN+LSTM, que obtiene un AUC de 0.742 pese a la escasez de eventos en esa región; sin embargo, su *recall* de 0.015 sugiere que este valor de AUC es resultado de un conjunto de datos de prueba extremadamente pequeño y no refleja una capacidad genuina de generalización.

En la división por cuadrantes, la segmentación más equilibrada en términos de área atenúa parcialmente las diferencias entre regiones, aunque el rendimiento continúa siendo bajo. El cuadrante noroeste favorece al CNN+LSTM (*recall*= 0.143, AUC= 0.558) mientras que el cuadrante suroeste es el más favorable para RiskViT (*recall*= 0.120, AUC= 0.530).

Estos resultados indican que ninguno de los dos modelos generaliza correctamente a nuevas áreas geográficas, lo que sugiere que ambos modelos dependen fuertemente de haber observado patrones en las mismas celdas durante el entrenamiento.

4.6. Análisis de sobreajuste

Como se pudo apreciar en las secciones 4.3.1 y 4.4.1, ambos modelos presentan un comportamiento de sobreajuste durante el entrenamiento, lo que se evidencia en el aumento de la pérdida en el conjunto de validación después de cierto número de épocas. En esta sección se estudia el resultado de aplicar diferentes técnicas de regularización para mitigar el sobreajuste y mejorar la capacidad de generalización de los modelos.

Como punto de partida, resulta interesante identificar el comportamiento de la función de pérdida en otros experimentos que presentan métricas ligeramente menores. Las figuras 4.12 y 4.13 muestran las curvas de pérdida para modelos RiskViT y CNN+LSTM que obtuvieron un *recall* de alrededor de 0.17 utilizando menos parámetros.

El modelo CNN+LSTM con menor cantidad de parámetros converge de una forma mucho más suave que el modelo RiskViT, y no muestra señales de sobreajuste, mientras que el modelo RiskViT con menor cantidad de parámetros presenta un comportamiento de sobreajuste similar al del mejor modelo, aunque con una pérdida de validación más elevada.

A partir de esto, se puede inferir que para la cantidad de datos disponibles con la distribución presentada anteriormente, un modelo más simple puede capturar mejor los

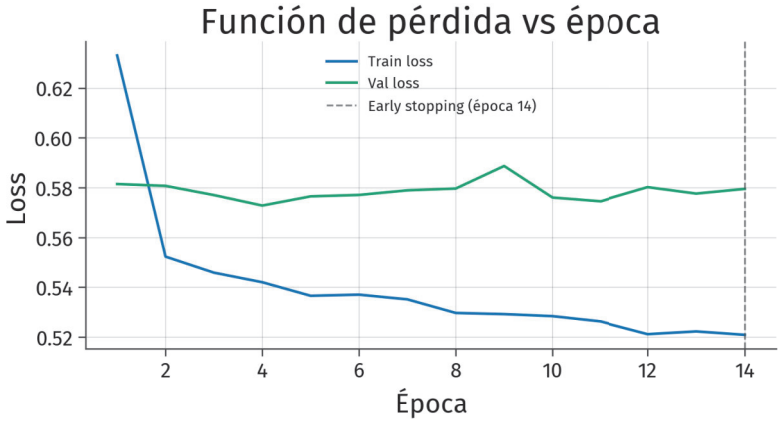


Figura 4.12: *Curvas de pérdida de RiskViT alrededor de 400 mil parámetros*

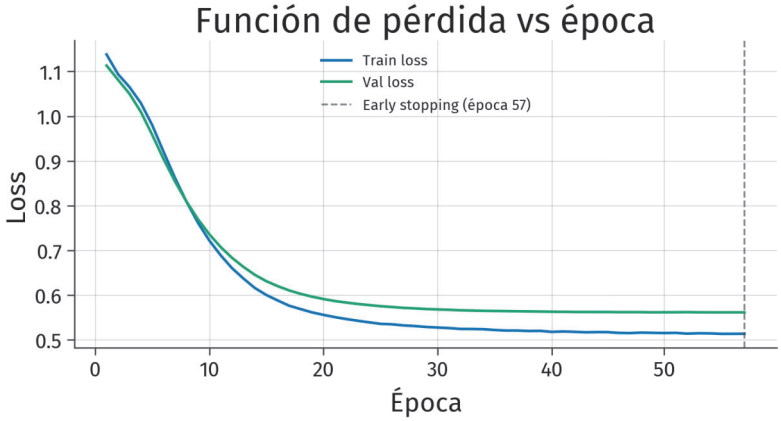


Figura 4.13: *Curvas de pérdida de CNN+LSTM con alrededor de 500 mil parámetros*

patrones de riesgo sin caer en el sobreajuste, mientras que un modelo más complejo como RiskViT, aunque tiene mayor capacidad de modelado, es más propenso a sobreajustar a los datos de entrenamiento. A pesar de que el modelo CNN+LSTM con menor cantidad de parámetros obtiene un *recall* ligeramente menor que el mejor modelo, su comportamiento sugiere que es un modelo más robusto y constituye una mejor base para predecir el riesgo de accidentes de tránsito en la ciudad de Mendoza.

4.6.1. Aumentación con conjuntos de datos externos

En primer lugar se evalúa la técnica de aumentación de datos utilizando conjuntos externos relacionados con accidentes de tránsito. A partir de un conjunto de datos que contiene información de accidentes en el departamento de Godoy Cruz, se obtienen aquellos accidentes que caen dentro del perímetro de la grilla definida anteriormente. Cabe recalcar que el conjunto de datos adicional no contiene todo el período temporal de interés: mientras que originalmente se contaba con datos desde enero de 2023 hasta marzo de 2026, el conjunto externo solo contiene datos desde enero de 2024 hasta diciembre de 2025. A pesar de esto, se añaden 203 accidentes adicionales a la grilla, lo que representa un aumento de tan solo 2 % del volumen de datos. La Figura 4.14 muestra la distribución espacial de los accidentes adicionales obtenidos a partir del conjunto externo.

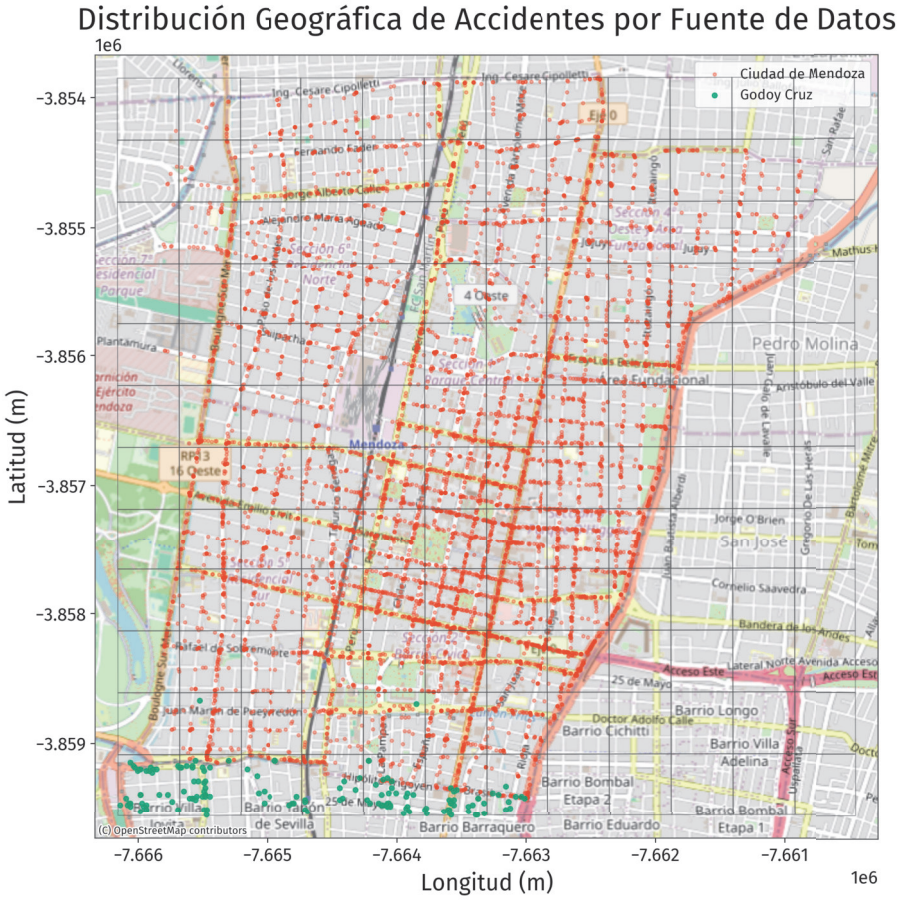


Figura 4.14: *Distribución espacial de los accidentes adicionales obtenidos a partir del conjunto externo.*

Considerando que el volumen de datos añadido es muy bajo, se estudia el rendimiento del modelo únicamente en la frontera sur, teniendo en cuenta las 9 celdas que presentan una cantidad relativamente alta de accidentes que provienen del conjunto de datos de Godoy Cruz. La Tabla 4.13 muestra las métricas obtenidas para el modelo RiskViT entrenado con el conjunto de datos original y con el conjunto de datos aumentado, evaluados únicamente sobre las celdas de la frontera sur.

Tabla 4.13: *Resultados del entrenamiento de RiskViT con el conjunto de datos original y con el conjunto de datos aumentado (frontera sur)*

Conjunto de datos	Recall	MAP	F1	AUC
Original	0.1915	0.1915	0.3262	0.554
Aumentado	**0.2073**	**0.1951**	**0.3273**	**0.6189**

El modelo entrenado con el conjunto de datos aumentado presenta una mejora en el *recall* y en el AUC, lo que sugiere que la adición de datos adicionales, aunque limitada, contribuye a mejorar la capacidad del modelo para identificar celdas de riesgo en la frontera sur. En este caso, la mejora es modesta, pero este experimento confirma la hipótesis de que un mayor volumen de datos puede ayudar a mejorar la capacidad de generalización del modelo.

4.6.2. Sobremuestreo (*oversampling*)

Otra técnica habitual para combatir el desbalance de clases es el *oversampling*: en lugar de penalizar la función de pérdida, se modifica la distribución de muestras que el modelo observa durante el entrenamiento. Ya se ha visto en secciones anteriores que la función de pérdida ponderada asigna mayor penalización a los errores cometidos sobre celdas de clase alta. El sobremuestreo actúa de forma complementaria, aumentando la frecuencia con la que el modelo procesa días con mayor cantidad de celdas de riesgo.

Para aplicar esta técnica sobre el conjunto de datos, se asigna un peso a cada muestra de entrenamiento (en este caso, cada día) en función de la cantidad de celdas de riesgo que contiene. Durante el proceso de carga de los datos, se utiliza un módulo de PyTorch llamado `WeightedRandomSampler`, que permite muestrear índices del conjunto de entrenamiento con reemplazo y con probabilidad proporcional al peso asignado a cada muestra. Se debe tener especial cuidado con las dependencias temporales para evitar la filtración de datos futuros y para asegurar que el modelo CNN+LSTM, que procesa secuencias de días, reciba secuencias coherentes durante el entrenamiento.

La Tabla 4.14 muestra los resultados del mejor modelo obtenido con sobremuestreo habilitado para cada arquitectura, comparados con el mejor modelo de referencia del entrenamiento principal (sin sobremuestreo).

Tabla 4.14: *Comparación del mejor modelo con y sin sobremuestreo para cada arquitectura.*

Modelo	Configuración	Recall	MAP	F1	Accuracy
RiskViT	Sin sobremuestreo (ref.)	**0.1807**	**0.0933**	**0.3724**	0.9257
	Con sobremuestreo	0.1799	0.0884	0.3447	**0.9448**
CNN+LSTM	Sin sobremuestreo (ref.)	**0.1833**	**0.0949**	**0.3819**	0.9206
	Con sobremuestreo	0.1808	0.0901	0.3655	**0.9379**

Los resultados indican que el sobremuestreo no aporta una mejora apreciable sobre la configuración de referencia: el *recall* disminuye levemente en ambas arquitecturas (menos de 0.003 puntos) y el MAP cae en torno a 0.005 puntos. El F1 también retrocede, especialmente en RiskViT. La única métrica que sube es la *accuracy*, pero dado el fuerte desbalance de clases, este indicador no es representativo de la capacidad de clasificación del modelo.

Una explicación posible es que la función de pérdida ponderada ya amplifica la señal de gradiente proveniente de las celdas de clase alta en cada paso de entrenamiento. En ese contexto, aumentar la frecuencia con la que el modelo ve los días de alto riesgo no añade información nueva: estos patrones ya reciben mayor atención a través de la función de pérdida.

La combinación de sobremuestreo con pérdida ponderada no supera al uso de la pérdida ponderada en solitario para este conjunto de datos y estas arquitecturas. El sobremuestreo queda descartado como técnica de regularización adicional en los experimentos posteriores.

4.6.3. *Dropout*

Se evaluó el efecto del *dropout* sobre el modelo RiskViT entrenando 48 variantes con probabilidades $p \in \{0.1, 0.2, 0.3, 0.5\}$. La Tabla 4.15 muestra los resultados del mejor modelo por tasa.

Tabla 4.15: *Resultados de RiskViT con distintas probabilidades de dropout (mejor modelo por tasa)*

Tasa (p)	Recall	MAP	F1	Val. loss
Sin dropout (ref.)	0.1807	0.0933	0.3724	—
0.1	0.1688	0.0838	0.3461	0.5847
0.2	0.1744	0.0870	0.3658	0.5915
0.3	0.1753	0.0922	0.3493	0.5899
0.5	**0.1834**	0.0894	0.3307	0.5795

El *dropout* no mejora las métricas de forma sistemática: el *recall* varía menos de 1.5 puntos respecto al modelo de referencia. La única tendencia clara es que la pérdida en validación disminuye (ligeramente) al aumentar p.

La Figura 4.15 muestra las curvas de pérdida para los extremos del rango evaluado

($p = 0.1$ y $p = 0.5$). En este caso, la dinámica es idéntica a la del modelo de referencia (Figura 4.2): la pérdida de entrenamiento desciende continuamente mientras la validación se estabiliza y activa el *early stopping* en pocas épocas. A diferencia de en el mejor modelo, la validación no vuelve a crecer pero tampoco muestra mejoras. Esto sugiere que el sobreajuste no está siendo mitigado por el *dropout*, y que la reducción de la pérdida en validación se debe principalmente a la aleatoriedad inherente al proceso de entrenamiento, más que a una mejora genuina en la capacidad de generalización del modelo. Es posible que la arquitectura RiskViT no consiga aprender los patrones de riesgo de manera efectiva, y que el modelo esté simplemente memorizando los datos de entrenamiento sin lograr una representación generalizable, a causa de la escasez de los datos disponibles.

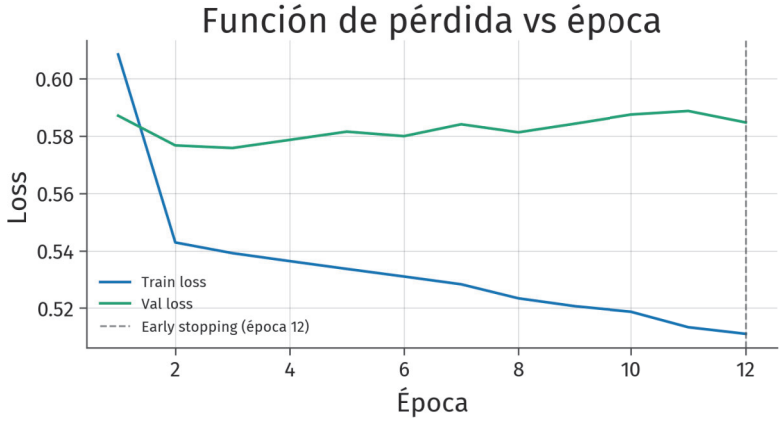


Figura 4.15: *Curvas de pérdida de RiskViT con* $p = 0.1$

De manera análoga, se evaluó el efecto del *dropout* sobre el modelo CNN+LSTM entrenando 48 variantes con las mismas tasas. La Tabla 4.16 muestra los resultados del mejor modelo por tasa.

Tabla 4.16: *Resultados de CNN+LSTM con distintas probabilidades de dropout (mejor modelo por tasa)*

Tasa (p)	Recall	MAP	F1	Val. loss
Sin dropout (ref.)	0.1833	0.0949	0.3819	—
0.1	0.1749	0.0912	0.3740	0.7543
0.2	**0.1767**	**0.0944**	0.3309	0.5770
0.3	0.1725	0.0872	0.3623	0.7435
0.5	0.1688	0.0894	0.3244	0.5769

A diferencia del modelo RiskViT, el modelo CNN+LSTM no muestra mejoras en la pérdida de validación al aumentar p, y las métricas de desempeño no presentan una tendencia clara. Según lo reportado en la Sección 4.6, simplemente reduciendo la cantidad de parámetros del modelo se obtiene un comportamiento de entrenamiento mucho más suave, sin señales de sobreajuste, y con un *recall* cercano al del mejor modelo, lo que sugiere que el *dropout* es innecesario para esta arquitectura.

4.7. Comparación de modelos

Para determinar cuál de las arquitecturas propuestas resulta más adecuada para el problema de predicción de accidentes en Mendoza, es necesario contrastar no solo las métricas de desempeño, sino también la complejidad de los modelos y el costo computacional asociado a su entrenamiento. La Tabla 4.17 resume los valores clave de los mejores modelos encontrados.

Tabla 4.17: *Comparación final entre Baseline, Random Forest, RiskViT y CNN+LSTM*

Modelo	Accuracy	Recall	MAP	F1	Precision	AUC	Parámetros	Tiempo (med.)
Baseline	0.9091	0.0765	0.0352	0.3620	0.3619	0.5294	-	-
Random Forest	0.9466	0.1607	0.0793	0.3410	**0.4197**	0.7519	-	**~3.8 s**
RiskViT	**0.9257**	0.1807	0.0933	0.3724	0.3821	**0.7822**	**731 504**	~1.5 min
CNN+LSTM	0.9206	**0.1833**	**0.0949**	**0.3819**	0.3804	0.7774	1 020 048	~6.0 min

Ambos modelos de DL superan holgadamente al modelo *baseline*, duplicando las métricas de *recall* y *MAP*, y obtienen resultados ligeramente mejores que *Random Forest*. Esto confirma que la inclusión de variables contextuales y el aprendizaje de patrones espacio-temporales aportan una ganancia de información significativa que puede ser capturada por algoritmos de DL, aunque la escasez de datos sigue siendo un factor limitante para alcanzar un desempeño más elevado.

Al comparar las arquitecturas entre sí, el modelo CNN+LSTM presenta una ligera ventaja en las métricas más relevantes. Con un MAP de 0.0949 frente a 0.09333 del RiskViT, la arquitectura híbrida aparenta ser ligeramente mejor para ordenar de manera correcta las celdas de mayor riesgo.

La Figura 4.16 muestra las curvas ROC de los modelos. Se puede observar que las curvas de RiskViT y CNN+LSTM están prácticamente superpuestas, confirmando que ambas arquitecturas tienen una capacidad de discriminación similar. Los valores de AUC cercanos a 0.78 indican que los modelos logran distinguir entre celdas con y sin riesgo de manera significativamente superior a un clasificador aleatorio (AUC = 0.5), aunque existe margen de mejora.

Por otro lado, el modelo RiskViT obtiene un *Accuracy* ligeramente mayor (0.9307). Sin embargo, dado el fuerte desbalance de clases discutido anteriormente, esta métrica es menos informativa que el *recall*, donde el RiskViT queda levemente por detrás (0.1782 vs 0.1813).

4.7.1. Análisis cualitativo de las predicciones

Más allá de las métricas obtenidas por cada modelo, resulta interesante estudiar cómo se ven las predicciones generadas por los mismos, y qué información accionable pueden aportar para personas en puestos de toma de decisiones. A continuación se analiza una serie de gráficos que comparan las predicciones de cada modelo con el mapa de

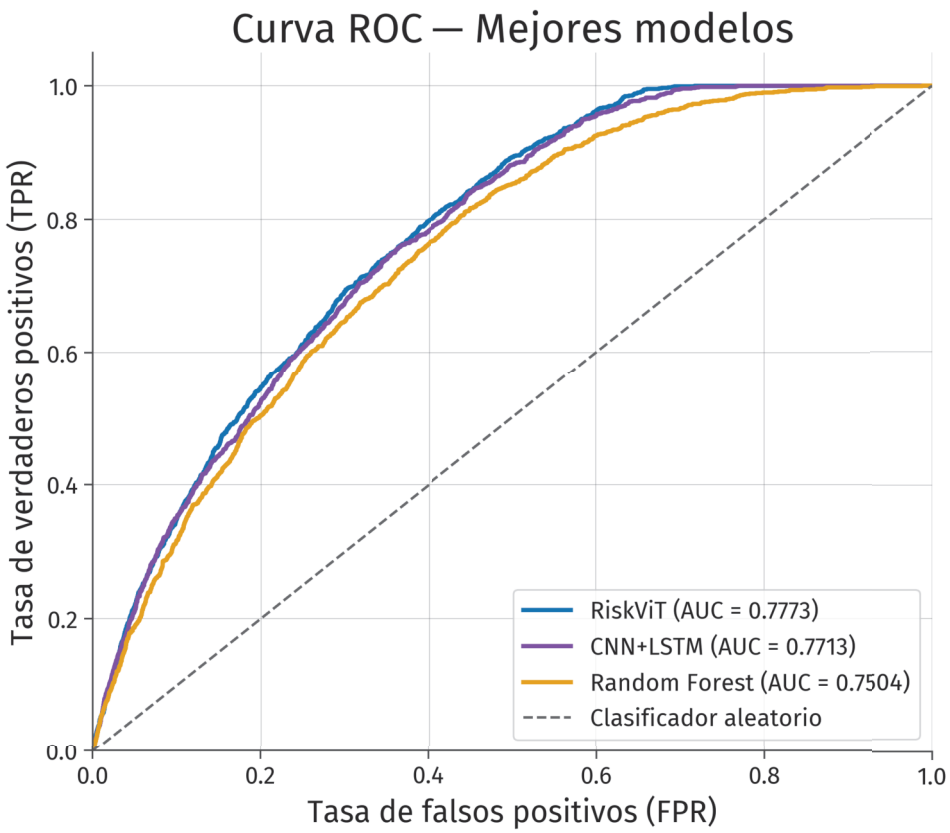


Figura 4.16: *Curvas ROC de los mejores modelos RiskViT y CNN+LSTM sobre el conjunto de prueba*

riesgo real para una selección de días del conjunto de prueba.

Para evaluar el desempeño práctico de los modelos, se seleccionaron cuatro muestras representativas del conjunto de prueba que permiten observar el comportamiento de las predicciones en diferentes contextos temporales y niveles de actividad. Las Figuras 4.17–4.20 muestran, para cada fecha, el mapa de riesgo real (objetivo) y las predicciones producidas por RiskViT y CNN+LSTM.

Análisis de la distribución espacial

Al observar las muestras de días laborales (por ejemplo, martes 30-09-2025 en la Figura 4.17 y miércoles 29-10-2025 en la Figura 4.18), se identifica una diferencia en la morfología de las predicciones.

En primer lugar, se observa un mayor *agrupamiento* en el modelo CNN+LSTM. En particular, en la Figura 4.18, este modelo tiende a generar predicciones conectadas entre sí, identificando una zona de riesgo medio-alto razonablemente similar al conjunto de accidentes reales. Este comportamiento sugiere que las capas convolucionales detectan patrones espaciales con las celdas adyacentes.

Por otro lado, el modelo RiskViT presenta predicciones más *dispersas*. Si bien logra detectar la presencia de riesgo en zonas generales, en ocasiones no captura la continui-

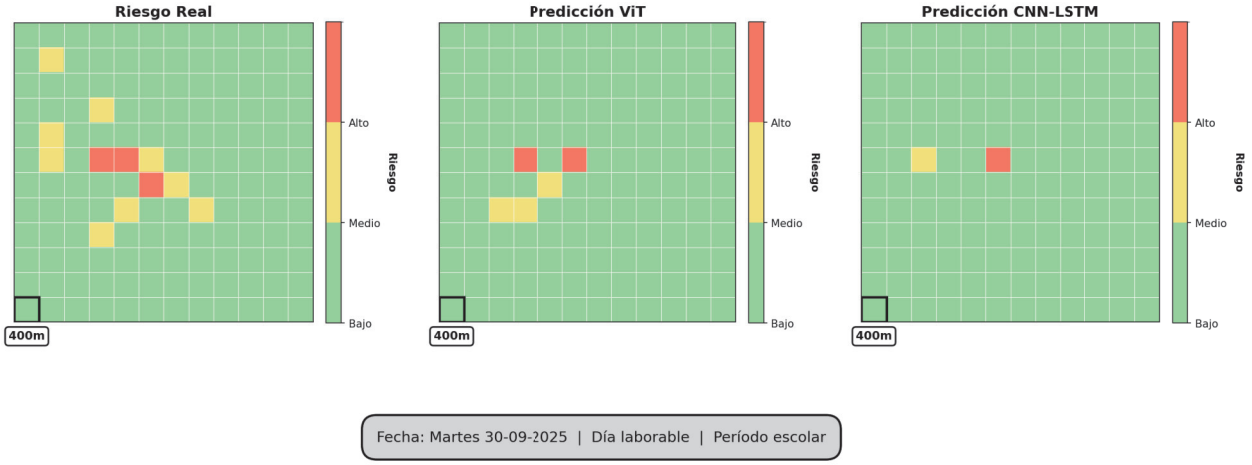


Figura 4.17: *Ejemplo de predicción (martes 30-09-2025)*

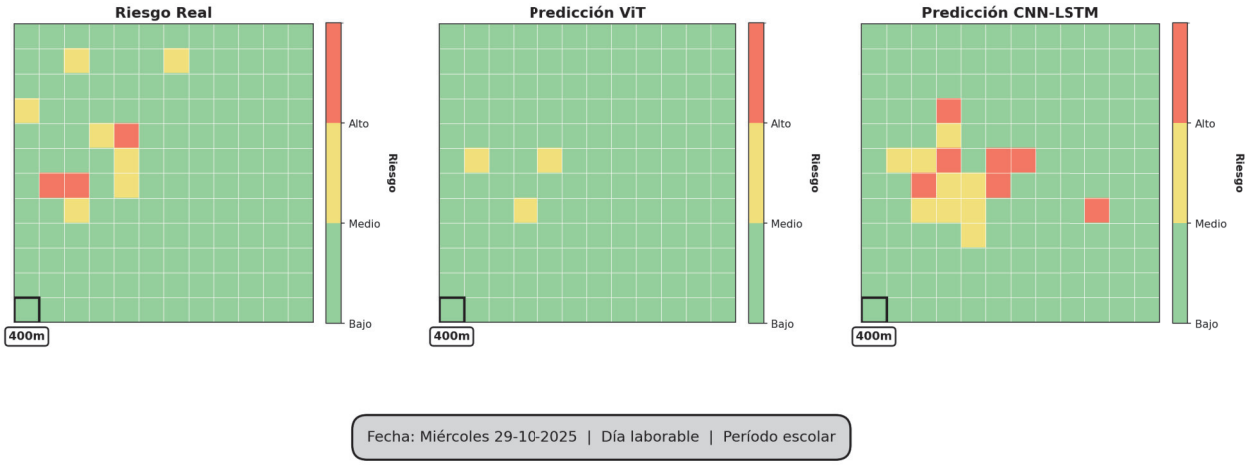


Figura 4.18: *Ejemplo de predicción (miércoles 29-10-2025)*

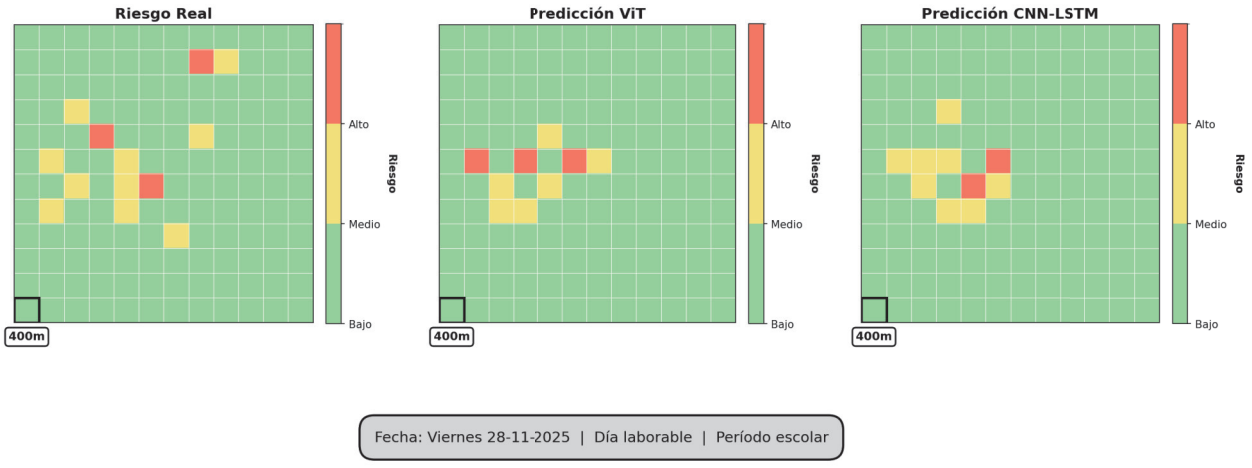


Figura 4.19: *Ejemplo de predicción (viernes 28-11-2025)*

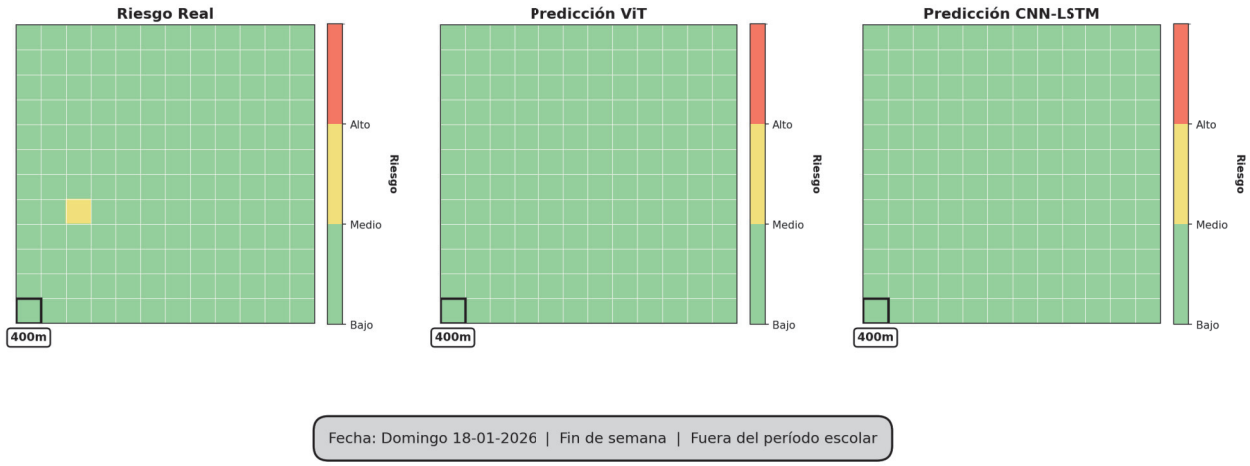


Figura 4.20: *Ejemplo de predicción (domingo 18-01-2026)*

dad del patrón espacial, marcando celdas aisladas. Esto puede explicarse por el mecanismo de atención propio de los *transformers*, que permite modelar dependencias de mayor alcance, pero potencialmente a costa de perder parte de la continuidad local que las arquitecturas convolucionales preservan.

Utilidad para la toma de decisiones

Un aspecto crítico para la implementación en entornos reales es la densidad de la predicción. En este contexto, el objetivo es contribuir a la asignación eficiente de recursos preventivos (por ejemplo, preventores, patrullas o ambulancias) sin diluir su impacto.

En primer lugar, ambos modelos demuestran ser conservadores y evitan *saturar* el mapa con celdas de riesgo alto. Este punto es relevante para prevenir la "fatiga de alerta" en los tomadores de decisiones: si el modelo marcara riesgo alto de forma generalizada en todo el mapa, la predicción perdería utilidad al no poder discriminar entre zonas de mayor o menor riesgo.

Por otro lado, en la muestra del viernes 28-11-2025 (Figura 4.19) se aprecia que ambos modelos coinciden en la ubicación general del riesgo, aunque CNN+LSTM tiende a reflejar con mayor precisión la intensidad del mismo, identificando zonas de clase más alta. Esta capacidad de discriminar no solo *dónde* actuar, sino también *con qué urgencia*, favorece el uso del modelo híbrido para la asignación preventiva.

Finalmente, es notable el desempeño de ambos modelos en días de baja siniestralidad, como el domingo 18-01-2026 (Figura 4.20), correspondiente a un fin de semana en período no escolar. En este caso, ambos modelos predicen correctamente una grilla casi totalmente libre de riesgo, lo que sugiere que han aprendido patrones de estacionalidad y factores externos para la predicción del riesgo.

4.8. Análisis de incertidumbre en las predicciones

Para evaluar la calidad de las predicciones generadas por los modelos desarrollados en este trabajo para la toma de decisiones, se estudia la incertidumbre asociada a dichas predicciones.

Una forma de realizar esta estimación sin necesidad de reentrenamiento consiste en aplicar la técnica de *Monte Carlo Dropout* (MC Dropout) explicada en la Sección X. En este caso, se deja el *dropout* activo durante la inferencia, lo que permite obtener una distribución de predicciones a partir de múltiples pasadas hacia adelante con la misma entrada. A partir de esta distribución, se calcula desviación estándar de las predicciones para cada celda, lo que cuantifica la incertidumbre de las mismas.

Se utilizaron $N = 30$ ejecuciones de inferencia (cada una de estas ejecuciones representa una predicción del modelo con *dropout* activo). La Figura 4.21 muestra la distribución de la desviación estándar σ_i para cada celda i del mapa, utilizando el modelo RiskViT. Se puede apreciar un pico de celdas con σ cercano a cero, lo que indica que el modelo tiene alta confianza en sus predicciones para estas zonas. Más a la derecha, aparece otro pico de celdas con σ alrededor de 0.045. Este segundo pico corresponde a las celdas con mayor varianza en las predicciones, lo que sugiere que el modelo tiene mayor incertidumbre en estas zonas. La presencia de estos dos picos indica que el modelo es capaz de distinguir entre zonas donde tiene alta confianza y zonas donde la incertidumbre es mayor, lo que es un resultado positivo para la utilidad de las predicciones en la toma de decisiones.

Una desviación estándar de 0.045 representa un valor relativamente bajo, lo que sugiere que incluso en las zonas con mayor incertidumbre, el modelo no muestra una variabilidad excesiva en sus predicciones. Esto coincide con la hipótesis de sobreajuste mencionada anteriormente: el modelo tiene un alto sesgo hacia los accidentes que ha visto durante el entrenamiento, lo que se traduce en una baja varianza en las predicciones, incluso en las zonas donde no tiene suficiente información para generalizar correctamente.

La Figura 4.22 muestra la distribución equivalente para el modelo CNN+LSTM. La distribución presenta un patrón bimodal similar al de RiskViT: un pico pronunciado en $\sigma \approx 0$ (celdas con alta confianza, en su mayoría de riesgo bajo) y un segundo pico más amplio centrado alrededor de $\sigma \approx 0.046$, que corresponde a las celdas con mayor incertidumbre. La media de σ es de 0.0345 y el percentil 90 es de 0.0586, lo que es comparable a los valores del modelo RiskViT y sugiere que ambas arquitecturas presentan un nivel de confianza similar en sus predicciones.

Finalmente, se calcula la correlación de Pearson entre la desviación estándar σ_i y el error de clasificación (celda correctamente clasificada o no) para cada par (i, t) del conjunto de prueba. Los valores obtenidos se muestran en la Tabla 4.18.

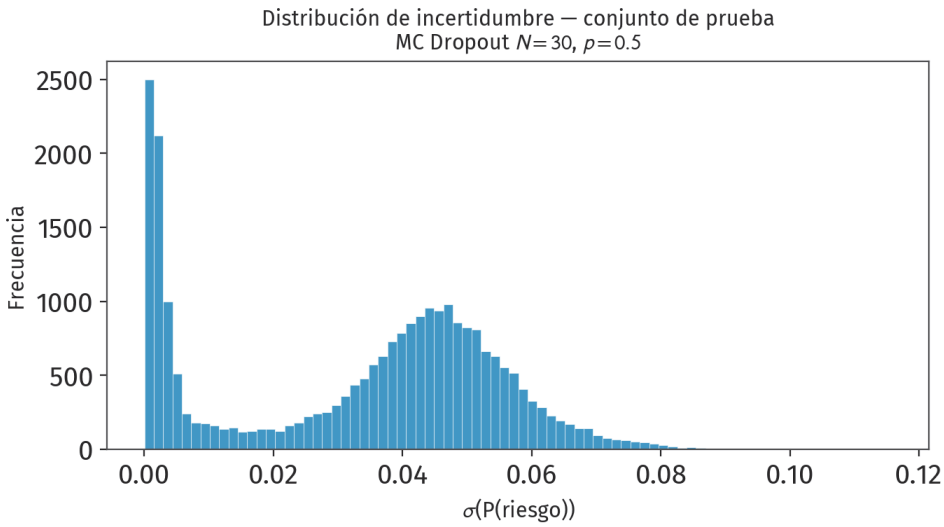


Figura 4.21: *Distribución de la incertidumbre de las predicciones del modelo RiskViT*

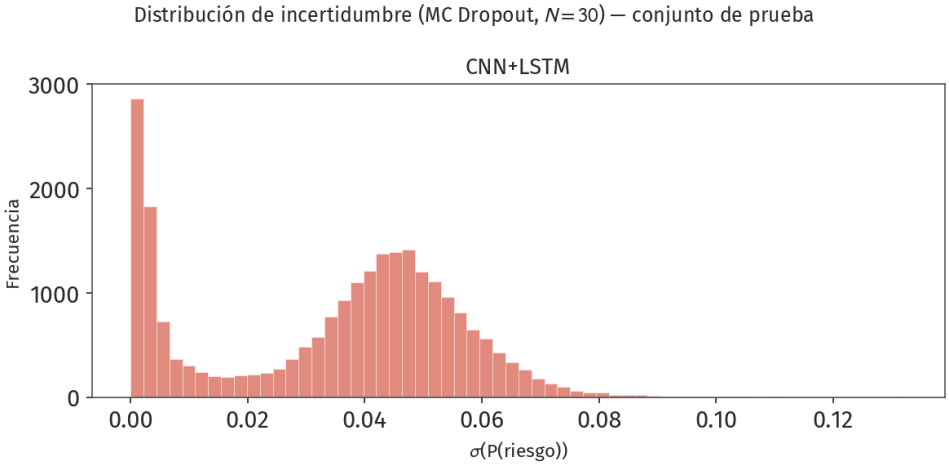


Figura 4.22: *Distribución de la incertidumbre de las predicciones del modelo CNN+LSTM*

Tabla 4.18: *Correlación de Pearson entre $\sigma(P(riesgo))$ y el error de clasificación en el conjunto de prueba.*

Modelo	**Correlación σ vs. error**
RiskViT	0.154
CNN+LSTM	0.147

Ambos modelos presentan una correlación positiva entre la desviación estándar de las predicciones y el error de clasificación, lo que indica que las celdas con mayor incertidumbre tienden a ser aquellas donde el modelo comete más errores. Sin embargo, la magnitud de esta correlación es relativamente baja (alrededor de 0.15), lo que sugiere que la incertidumbre estimada no es un predictor fuerte del error de clasificación. Nuevamente, esto puede estar relacionado con la hipótesis de sobreajuste: el modelo tiene un alto sesgo hacia los patrones de riesgo que ha visto durante el entrenamiento.

4.9. Análisis del estado del arte

En esta sección se analizan aproximaciones previas al problema de predicción espacio-temporal del riesgo de accidentes de tránsito utilizadas en otros casos de estudio. Teniendo en cuenta que los datos usados en dichos trabajos son muy distintos a los disponibles en la ciudad de Mendoza, es imposible hacer una comparación directa de los resultados. Aún así, es interesante estudiar cómo los modelos desarrollados en esta tesina obtienen resultados similares a los que reportan en otros desarrollos del estado del arte como GS-Net [43] o CViT [41] (Capítulo 2), a pesar que se cuenta con un volumen de datos mucho menor.

A continuación se resaltan las principales diferencias entre los datos utilizados en esta tesina y los que usan en los modelos mencionados anteriormente

Tabla 4.19: *Análisis de los de conjuntos de datos por ciudad*

Característica	Mendoza	Chicago	NY
Período	2023–2025	2016	2013
Accidentes	9 700	44 000	147 000
Frecuencia de predicciones	Diaria	Horaria	Horaria
Datos contextuales*	C, P	C, S, T	C, S, T, P
Grilla	12×12	20×20	20×20
Área por celda (km^2)	0.16	4	4

*__C__: Clima, __P__: Puntos de interés, __S__: Segmentos de calle, __T__: Taxímetros

Las diferencias en los conjuntos de datos son notables. En primer lugar, la cantidad de accidentes es significativamente menor en Mendoza que en Chicago y Nueva York. Por otro lado, la riqueza de los datos en las ciudades estadounidenses es considerable: las mismas cuentan con datos de taxímetros (**T**) y segmentos de calle (**S**) que permiten estimar el flujo vehicular sobre cada parte de la ciudad con una frecuencia horaria: esto constituye la diferencia más significativa con este trabajo, en el que se debió adaptar la tarea para poder aplicarla al contexto de Mendoza. Finalmente, resulta importante mencionar la diferencia en el área de cada una de las celdas (y, en consecuencia, el área de estudio de la ciudad), que se aprecia en cada uno de los experimentos. El tamaño de las celdas en las ciudades de Chicago y Nueva York es 25 veces mayor que el tamaño de las celdas utilizadas en Mendoza, y cada una de ellas, a su vez, contiene una cantidad

mucho mayor de accidentes. Esto implica que ninguno de estos dos casos de estudios es directamente comparable al caso de Mendoza.

Una vez mencionadas estas diferencias, se procede a analizar las métricas obtenidas. La Tabla 4.20 describe de los valores de *recall* obtenidos por cada modelo en cada ciudad, mientras que la tabla 4.21 realiza un estudio equivalente para los valores de MAP. Se incluyen los resultados de RiskViT y CNN+LSTM para Mendoza, y los resultados reportados por los modelos CViT y GSNet para Chicago y Nueva York.

Tabla 4.20: *Análisis de recall en el estado del arte*

Mendoza		**Chicago**		**Nueva York**	
RiskViT	CNN+LSTM	GSNet	CViT	GSNet	CViT
0.1807	**0.1833**	0.2069	**0.2093**	**0.3402**	0.3386

Tabla 4.21: *Análisis de MAP en el estado del arte*

Mendoza		**Chicago**		**Nueva York**	
RiskViT	CNN+LSTM	GSNet	CViT	GSNet	CViT
0.0933	**0.0949**	0.0903	**0.0980**	0.1861	**0.1875**

Analizando las tres ciudades, es interesante ver cómo impacta la cantidad de datos de la mismas en el *recall* obtenido. Aunque se trata de un caso de estudio con diferencias considerables, el hecho de que las métricas alcanzadas en el *dataset* de Chicago estén tan solo un 10 % por encima de las obtenidas demuestra que el enfoque tomado en esta tesina es efectivo para atacar el problema de la predicción espacio-temporal de accidentes de tránsito. Al mismo tiempo, los resultados de MAP también se encuentran en un rango similar: en particular, los dos modelos desarrollados en esta tesina superan al modelo GSNet en la ciudad de Chicago y se acercan a los resultados obtenidos por CViT, aunque el resultado se queda los de las métricas reportadas en Nueva York.

Por lo tanto, del análisis de los resultados obtenidos en Mendoza se espera que, a medida que los enfoques propuestos en este trabajo puedan acceder a mayores volúmenes de datos, el rendimiento de los modelos y la calidad de sus predicciones vean una mejora.

Finalmente, tras analizar todas las métricas obtenidas por los modelos y estudiar el comportamiento frente a diferentes experimentos, es posible concluir que para este caso particular en el que el conjunto de datos es acotado y desbalanceado un modelo más sencillo como CNN+LSTM es capaz obtener un rendimiento apenas superior al de un modelo más complejo como RiskViT, aunque presenta mejores condiciones para su desarrollo y mantenimiento a largo plazo, ya que presenta curvas de entrenamiento cerca del óptimo.

4.10. Experimentos sobre distintos tamaños de grilla

Además de los experimentos exhaustivos para ajustar los hiperparámetros de ambos modelos sobre la grilla de 12×12, se lleva a cabo una serie de experimentos más acotada para evaluar el rendimiento del modelo RiskVit sobre grillas de otros tamaños: 4×4 ($1.44\ km^2$ por celda), 8×8 ($0.36\ km^2$ por celda), 10×10 ($0.23\ km^2$ por celda) y 16×16 ($0.09\ km^2$ por celda). Cabe recalcar que estos experimentos se realizan con los parámetros que obtuvieron el mejor rendimiento en la grilla de 12×12, sin realizar un ajuste específico para cada tamaño de grilla, por lo que no representan un ajuste óptimo para cada caso.

En particular, en la Figura 4.23 se grafican los mejores modelos obtenidos para cada tamaño de grilla según su *recall*. Se incluye una línea para el *recall* y otra para la *accuracy*. Se puede apreciar que existe un compromiso entre ambas métricas: mientras que en los tamaños de celda más pequeños se puede obtener un *recall* más alto (superando el 50 %), en estos casos la *accuracy* baja considerablemente. A medida qe el tamaño de la grilla aumenta, la proporción de las métricas se invierte, obteniendo un *recall* apenas por encima del 10 % en la grilla de 16×16. Esto puede explicarse por el fenómeno del desbalance de clases. Mientras más fina es la granularidad con la que se divide el área de estudio de la ciudad de Mendoza, cada celda está asociada a un número menor de accidentes, y aumenta la proporción de celdas que pertenecen a la clase mayoritaria (riesgo muy bajo). La tarea se vuelve considerablemente más difícil de resolver para el modelo, que a pesar de estar preparado para lidiar con el desbalance no consigue obtener un buen rendimiento.

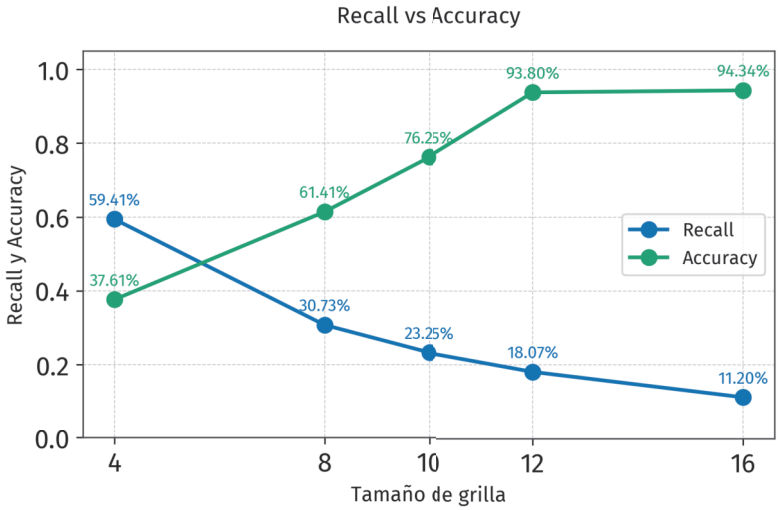


Figura 4.23: *Comparación de recall y accuracy para diferentes tamaños de grilla*

4.11. Resultados del desarrollo de la aplicación Web

Como parte del presente trabajo, se desarrolló una aplicación web interactiva que permite visualizar las predicciones generadas por los modelos entrenados. La misma fue diseñada con el objetivo de servir como una herramienta de apoyo para la toma de decisiones en materia de seguridad vial, facilitando el acceso y la interpretación de las predicciones de riesgo de accidentes de tránsito.

4.11.1. Diseño e interfaz de usuario

La aplicación cuenta con un diseño *responsive*, adaptable a pantallas de dispositivos móviles y de escritorio, lo que permite su uso en diferentes contextos y escenarios. La interfaz de usuario fue desarrollada priorizando la simplicidad y el minimalismo, de manera que sea intuitiva de usar por parte de cualquier usuario, incluso aquellos sin conocimiento técnico en el área. Se utilizó un estilo de mapa sobrio obtenido de *OpenStreetMap*, y colores fácilmente identificables para las distintas clases de riesgo.

La página principal de la aplicación (Figura 4.24) se centra en un mapa interactivo de la ciudad de Mendoza, sobre el cual se superponen las predicciones de riesgo generadas por los modelos. El mapa permite al usuario realizar operaciones de *zoom* y desplazamiento para explorar diferentes zonas de la ciudad con mayor detalle. También se solicita al usuario permiso de ubicación (opcional), que de ser aceptado muestra su ubicación actual en el mapa, facilitando la identificación de zonas de riesgo cercanas a su posición.

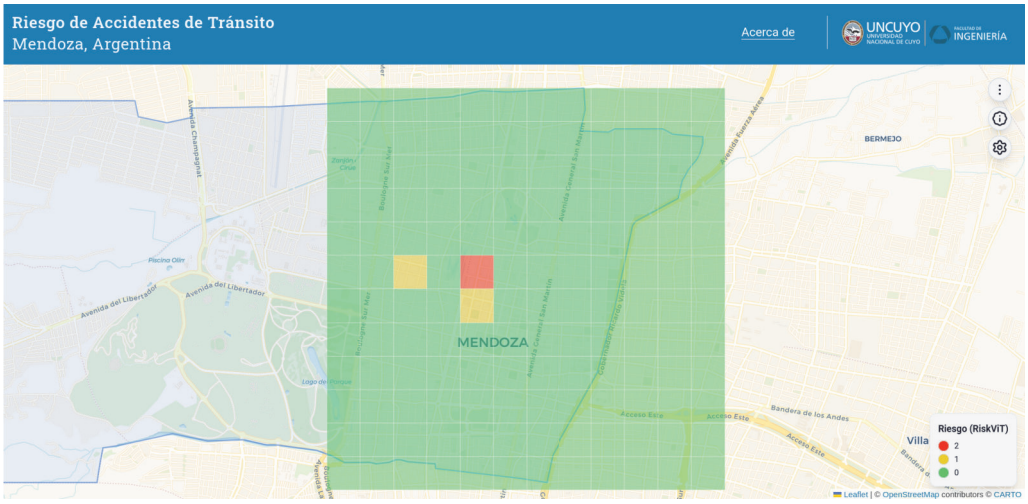


Figura 4.24: *Pantalla principal de la aplicación web*

Al costado del mapa se ubica un panel lateral con controles que permiten al usuario interactuar con las visualizaciones. Este panel incluye botones para activar o desactivar diferentes capas sobre el mapa, seleccionar el modelo activo (RiskViT o CNN+LSTM) y acceder a los datos de contexto utilizados para generar las predicciones. La Figura

4.25 muestra la aplicación con las capas de datos contextuales activadas, permitiendo visualizar la información meteorológica y los puntos de interés considerados por los modelos.

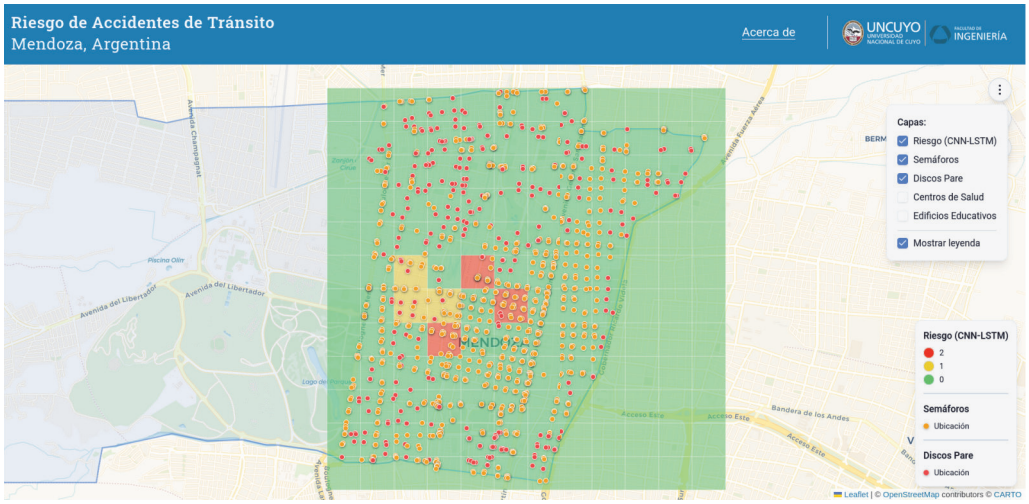


Figura 4.25: *Visualización de las capas contextuales en el mapa*

Además de la visualización de predicciones, la aplicación incluye funcionalidades para explorar los datos contextuales utilizados por los modelos. Esto permite a los usuarios conocer los factores que están siendo utilizados para realizar las predicciones. La Figura 4.26 muestra el panel de datos contextuales desplegado abajo a la izquierda.

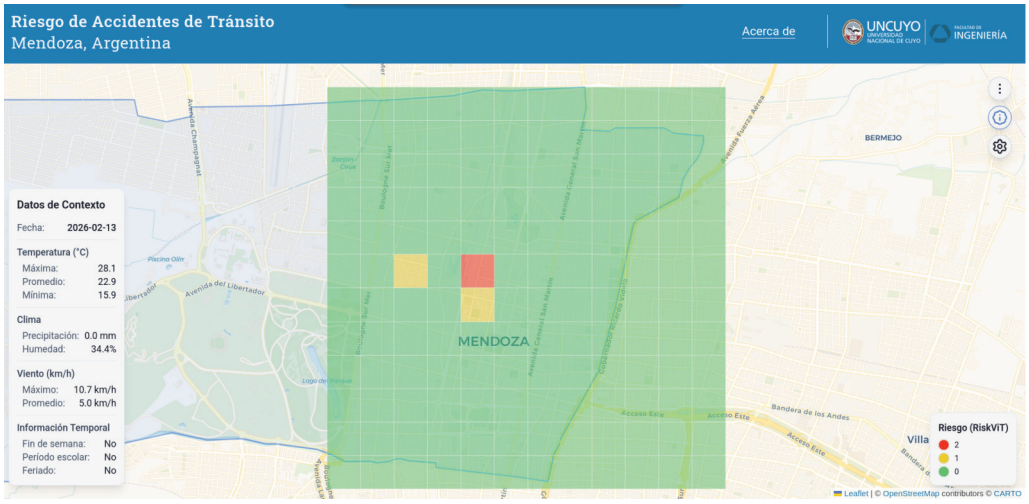


Figura 4.26: *Comparación visual de las predicciones de ambos modelos*

4.11.2. Adaptabilidad a dispositivos móviles

Un aspecto destacado del diseño es su capacidad de adaptación a pantallas de diferentes tamaños. La Figura 4.27 muestra la interfaz en un dispositivo móvil, donde los controles se reorganizan ligeramente y se centra el mapa de riesgo. Esta característica es especialmente relevante considerando que los usuarios potenciales de la herramienta podrían necesitar acceder a las predicciones mientras se encuentran en terreno.

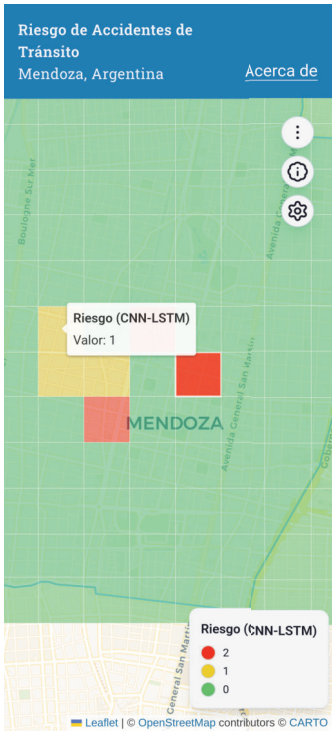


Figura 4.27: *Vista de la aplicación en dispositivo móvil*

4.11.3. Información del proyecto

Además de la página principal, la aplicación incluye una sección "Acerca de" (Figura 4.28) que proporciona información detallada sobre el proyecto a los usuarios. Esta página actúa como un breve resumen de los objetivos del trabajo, las metodologías empleadas y el contexto en el que se desarrolló. Esta sección se abre como un diálogo emergente la primera vez que un usuario accede a la aplicación, para asegurar el uso responsable y consciente de la herramienta, teniendo en cuenta las limitaciones de los modelos.

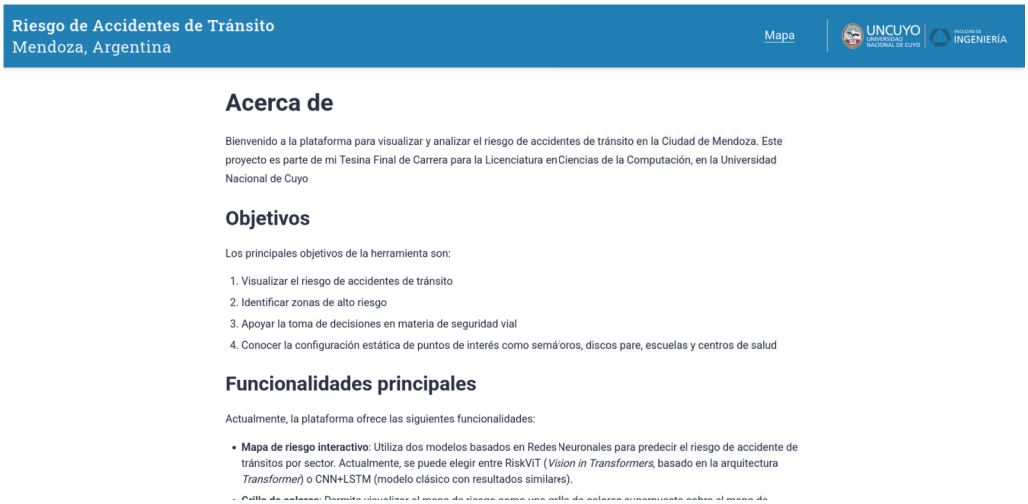


Figura 4.28: *Página "Acerca de" con información del proyecto*

5

Conclusiones, desafíos y perspectivas futuras

Este capítulo presenta las conclusiones obtenidas del presente trabajo final de carrera. Se resumen los principales objetivos cumplidos y hallazgos realizados (Sección 5.1), se discuten las dificultades encontradas durante el desarrollo del proyecto (Sección 5.2), se plantean posibles líneas de trabajo futuro para continuar con la investigación iniciada (Sección 5.3) y, por último, se presentan las conclusiones finales (Sección 5.4).

5.1. Objetivos Cumplidos

La predicción espacio-temporal del riesgo de accidentes de tránsito es un campo de estudio incipiente pero con un gran potencial para mejorar la seguridad vial. Particularmente en países en vías de desarrollo como la Argentina y el resto de América Latina, la implementación de modelos de predicción de accidentes puede contribuir significativamente a la reducción de siniestros viales, que representan una de las principales causas de mortalidad en la región. La investigación en el estado del arte se enfoca en grandes ciudades en Estados Unidos o países como Corea del Sur y Japón, pero hay una falta de estudios que aborden el contexto latinoamericano, que presenta características muy diferentes a nivel cultural, demográfico y de infraestructura vial.

Con respecto a los objetivos planteados al principio de este documento, podemos concluir:

- En cuanto a *Desarrollar una arquitectura de Aprendizaje Profundo (DL) que, a partir de*

los datos sobre accidentes viales y otra información ambiental, pueda predecir la probabilidad de que ocurra un accidente, se realizó un análisis exhaustivo del estado del arte en la predicción de accidentes de tránsito utilizando técnicas de DL. Se implementaron dos arquitecturas para abordar el problema: la arquitectura RiskViT, basada en ViT, y la arquitectura CNN+LSTM, un modelo híbrido que combina redes convolucionales para la extracción de características espaciales y redes recurrentes para modelar la dinámica temporal. Ambas arquitecturas fueron entrenadas y evaluadas a partir de un conjunto de datos diseñado en este mismo trabajo final de carrera, que combina datos históricos de accidentes de tránsito de la ciudad de Mendoza con información contextual adicional, como datos meteorológicos y puntos de interés para la infraestructura vial. Los resultados obtenidos muestran que ambas arquitecturas son capaces de predecir el riesgo de accidentes viales con un desempeño prometedor, que supera ampliamente al modelo *baseline* propuesto como punto de partida.

- Con respecto al segundo objetivo, *implementar un prototipo web que permita, a partir de datos de accidentes de tráfico en la ciudad de Mendoza, mostrar dónde ocurrirían accidentes con mayor probabilidad sobre un mapa de la Ciudad de Mendoza dividido en secciones*, se desarrolló una aplicación web interactiva que visualiza las predicciones generadas por los modelos de DL en un mapa de la ciudad de Mendoza. La aplicación permite a los usuarios explorar las predicciones de riesgo de accidentes en diferentes sectores de la ciudad, así como acceder a información contextual relevante, como la ubicación de semáforos, colegios y hospitales. La herramienta, diseñada utilizando tecnologías *open source* modernas, es accesible tanto desde computadoras de escritorio como desde dispositivos móviles, y está pensada para que tomadores de decisiones puedan utilizarla para el apoyo en la prevención de accidentes viales y la optimización de recursos de emergencia.

Al alcanzar estos objetivos y desarrollar las herramientas propuestas, este trabajo final de carrera contribuye a la investigación en el campo de la predicción de accidentes de tránsito utilizando técnicas de DL, especialmente en el contexto latinoamericano. La aplicación web constituye más que un prototipo académico, ya que está pensada para ser puesta en producción para su uso real sin necesidad de realizar grandes modificaciones: gracias a los mecanismos que automáticamente actualizan el conjunto de datos y reentrenan el modelo, se asegura que las predicciones sean relevantes aún en momentos futuros.

Además, tuve la posibilidad de presentar resultados preliminares de este trabajo en el IV Foro de Tecnologías para la Mejora Urbana, organizado por la Universidad Autónoma de México. Esta experiencia me permitió recibir retroalimentación de expertos en el área y me ayudó a mejorar la calidad de la investigación y la presentación de los resultados. La presentación está disponible en el siguiente enlace: `https://youtu.be/_3QFv4aWsTs?t=3207`.

5.2. Dificultades encontradas

El desarrollo de esta tesina final de carrera presentó numerosas dificultades, que debieron ser afrontadas a lo largo del proceso.

En primer lugar, la disponibilidad de datos distintos a los utilizados en otros estudios explorados durante el análisis de la literatura fue un gran desafío. La mayoría de estos trabajos están realizados sobre ciudades que cuentan con sistemas de recolección de datos de accidentes de tránsito muy completos, que incluyen información detallada del flujo de tránsito, datos de sensores en tiempo real, entre otros. En contraste, la ciudad de Mendoza no cuenta con este tipo de infraestructura de captura de datos, lo que llevó a la necesidad de adaptar los modelos de DL y realizar predicciones diarias. A su vez, el conjunto de datos fue diseñado específicamente para este trabajo, lo que implicó un importante esfuerzo en la recolección, limpieza y preprocesamiento de los datos, así como en preparación de un proceso automatizado para poder actualizarlo periódicamente.

Por otro lado, el proceso de entrenamiento de los modelos de DL también presentó dificultades, especialmente en la etapa de ajuste de hiperparámetros. Para poder ejecutar los experimentos en un entorno de computación de alto desempeño, se diseñaron diversos *scripts* para poder automatizar los barridos de hiperparámetros, la evaluación de los modelos y la recolección de resultados, teniendo en cuenta que los experimentos fueron ejecutados en un contexto de colas que no garantiza que los mismos se ejecuten de inmediato.

Finalmente, la integración de las predicciones generadas por los modelos de DL con la visualización en el mapa requirió un esfuerzo adicional para garantizar que las predicciones se muestren de manera coherente con la geografía de la ciudad. También fue necesario optimizar el rendimiento de la aplicación para que la experiencia de usuario no se vea comprometida, teniendo en cuenta que las predicciones deben ser actualizadas diariamente y que la aplicación debe ser accesible desde dispositivos móviles con diferentes capacidades de procesamiento.

5.3. Perspectivas futuras

Esta tesina final de carrera constituye un punto de partida para futuras investigaciones en el campo de la predicción de accidentes de tránsito utilizando técnicas de DL, y deja abiertas diversas líneas de trabajo que podrían explorarse posteriormente.

En primer lugar, sería esencial obtener datos más ricos (y posiblemente en tiempo real) para la ciudad de Mendoza que permitan mejorar el rendimiento de los modelos y disminuir la granularidad temporal. Si se contara con datos de flujo de tránsito, taxímetros u otros sensores, se podrían implementar algunas de las mejoras a los modelos propuestas en este trabajo. Además, considerando que el área de estudio constituye una

ciudad relativamente pequeña, sería interesante explorar la posibilidad de incorporar datos de los departamentos vecinos, como Guaymallén, Las Heras o la totalidad de Godoy Cruz, para aumentar la cantidad de datos disponibles y mejorar la capacidad de generalización de los modelos.

A su vez, se podrían explorar otras arquitecturas de DL que han sido utilizadas en la literatura para la predicción de accidentes de tránsito, como modelos basados en grafos o arquitecturas híbridas más complejas. La mayoría de estas técnicas han sido validadas principalmente en ciudades mucho más grandes que Mendoza, por lo que sería interesante evaluar su desempeño en un contexto con características diferentes.

En cuanto a la aplicación web, un aspecto que no fue abordado es la validación de las predicciones generadas por los modelos con usuarios reales, como personal de seguridad vial o servicios de emergencia. Sería importante realizar estudios de usabilidad y evaluar el impacto que la herramienta tiene en la toma de decisiones para la prevención de accidentes viales. A su vez, podrían implementarse funcionalidades extra, como alertas de riesgo en tiempo real o retroalimentación por parte de los usuarios que permita mejorar los modelos a lo largo del tiempo.

5.4. Conclusiones Finales

Desde un punto de vista personal, esta tesina final de carrera ha sido una experiencia de aprendizaje muy enriquecedora, que me ha permitido aplicar muchos de los conocimientos adquiridos a lo largo de la carrera, complementando con la investigación sobre el estado del arte y técnicas recientes en el área del DL. Personalmente, considero que la intersección entre la investigación académica y el desarrollo de herramientas prácticas es fundamental para que el conocimiento generado tenga un impacto real en la sociedad, y este trabajo final de carrera ha sido una oportunidad para explorar esa intersección.

Bibliografía

[1] World Health Organization. *Global status report on road safety 2023*. 2023. URL: https://iris.who.int/server/api/core/bitstreams/46275f9f-ef66-4892-8ddd-a496ef8c1b74/content (visitado 27-01-2026).

[2] Luchemos por la Vida. *Consideraciones generales acerca de los accidentes de tránsito*. es. Asociación Civil Luchemos por la Vida. URL: https://www.luchemos.org.ar/es/accidentes-argentina (visitado 12-11-2025).

[3] Municipalidad de la Ciudad de Mendoza. *GeoPortal de la Ciudad de Mendoza: Plataforma de información geográfica*. https://webgis.ciudaddemendoza.gob.ar/portal/home/. 2026. (Visitado 27-01-2026).

[4] World Health Organization. *Road safety*. 2023. URL: https://www.who.int/health-topics/road-safety#tab=tab_1 (visitado 17-11-2025).

[5] Congreso de la Nación Argentina. «Artículo 64: Presunciones». En: *Ley de Tránsito N° 24.449*. Boletín Oficial de la República Argentina, 1994. URL: http://servicios.infoleg.gob.ar/infolegInternet/anexos/0-4999/818/texact.htm#20 (visitado 17-11-2025).

[6] María Luján Medina et al. *Glosario de términos y definiciones relativas a la seguridad vial*. Inf. téc. Versión 1. Argentina: Dirección Nacional de Observatorio Vial, Agencia Nacional de Seguridad Vial, Ministerio de Transporte, ene. de 2021. URL: www.argentina.gob.ar/seguridadvial.

[7] Agencia Nacional de Seguridad Vial. *Informe preliminar de siniestralidad fatal 2024*. Informe estadístico. Datos parciales y preliminares actualizados a junio de 2025. Ministerio de Economía, Secretaría de Transporte, Argentina, jun. de 2025. URL: https://www.argentina.gob.ar/sites/default/files/2018/12/informe_preliminar_de_siniestralidad_fatal_2024_junio_2025_2.pdf.

[8] Luchemos por la Vida. *Comparación de Argentina con otros países*. es. Asociación Civil Luchemos por la Vida. URL: https://www.luchemos.org.ar/es/estadisticas/internacionales/comparacion-de-argentina-con-otros-paises (visitado 12-11-2025).

[9] Roy T. Fielding et al. *HTTP Semantics*. RFC 9110. Internet Engineering Task Force (IETF), 2022. URL: https://www.rfc-editor.org/rfc/rfc9110 (visitado 11-02-2026).

[10] Roy Thomas Fielding. «Architectural Styles and the Design of Network-based Software Architectures». Tesis doct. University of California, Irvine, 2000. URL: https://roy.gbiv.com/pubs/dissertation/fielding_dissertation.pdf (visitado 11-02-2026).

[11] Andrew S. Tanenbaum et al. *Distributed Systems: Principles and Paradigms*. 4.ª ed. Pearson Prentice Hall, 2023.

[12] Meta Open Source. *React*. 2026. URL: https://react.dev/ (visitado 11-02-2026).

[13] Stuart J. Russell et al. *Artificial Intelligence: a modern approach*. 3.ª ed. Pearson, 2009.

[14] Mohsen Soori et al. «Artificial Intelligence, Machine Learning and Deep Learning in Advanced Robotics, A Review». En: *Cognitive Robotics* (2023). URL: https://api.semanticscholar.org/CorpusID:258017769.

[15] Alexey Dosovitskiy et al. «An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale». En: *CoRR* abs/2010.11929 (2020). arXiv: 2010.11929. URL: https://arxiv.org/abs/2010.11929.

[16] Zhong-ren Peng et al. «The Pathway of Urban Planning AI: From Planning Support to Plan-Making». En: *Journal of Planning Education and Research* 44 (2023), págs. 2263-2279. URL: https://api.semanticscholar.org/CorpusID:259637390.

[17] Alec Radford et al. «Improving Language Understanding by Generative Pre-Training». En: 2018. URL: https://api.semanticscholar.org/CorpusID:49313245.

[18] Andriy Burkov. *The Hundred-Page Machine Learning Book*. 1.ª ed. Kindle Direct Publishing, 2019. ISBN: 9781790485000.

[19] Trevor Hastie et al. *The Elements of Statistical Learning*. Springer Series in Statistics. New York, NY, USA: Springer New York Inc., 2001.

[20] Gareth James et al. *Introduction*. en. Springer texts in statistics. Cham: Springer International Publishing, 2023, págs. 1-13.

[21] Matt Housley Joe Reis. *Fundamentals of Data Engineering: Plan and Build Robust Data Systems*. 1.ª ed. O'Reilly Media, 2022. ISBN: 9781098108304; 1098108302.

[22] José Hernández Orallo et al. *Introducción a la minería de datos*. es. Old Tappan, NJ: Prentice Hall, 2004.

[23] Jürgen Jänes et al. «Deep learning for protein structure prediction and design—progress and applications». En: *Molecular Systems Biology* 20.3 (ene. de 2024), págs. 162-169. ISSN: 1744-4292. DOI: 10.1038/s44320-024-00016-x.

[24] Aston Zhang et al. *Dive into Deep Learning*. https://D2L.ai. Cambridge University Press, 2023.

[25] Yarin Gal et al. «Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning». En: *Proceedings of the 33rd International Conference on Machine Learning (ICML)*. PMLR, 2016, págs. 1050-1059.

[26] Shigeki Karita et al. «A Comparative Study on Transformer vs RNN in Speech Applications». En: *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, dic. de 2019, págs. 449-456. DOI: 10.1109/asru46091.2019.9003750.

[27] Y. Bengio et al. «Learning long-term dependencies with gradient descent is difficult». En: *IEEE Transactions on Neural Networks* 5.2 (1994), págs. 157-166. DOI: 10.1109/72.279181.

[28] Sepp Hochreiter et al. «Long Short-Term Memory». En: *Neural Computation* 9.8 (1997), págs. 1735-1780. DOI: 10.1162/neco.1997.9.8.1735.

[29] Ashish Vaswani et al. «Attention is all you need». En: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., 2017, págs. 6000-6010. ISBN: 9781510860964.

[30] Haoyi Zhou et al. *Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting*. 2021. arXiv: 2012.07436 [cs.LG]. URL: https://arxiv.org/abs/2012.07436.

[31] Jeffrey Pennington et al. «Glove: Global Vectors for Word Representation». En: vol. 14. Ene. de 2014, págs. 1532-1543. DOI: 10.3115/v1/D14-1162.

[32] Dzmitry Bahdanau et al. «Neural Machine Translation by Jointly Learning to Align and Translate». En: *ArXiv* 1409 (sep. de 2014).

[33] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[34] Matanel Oren et al. «Transformers are Multi-State RNNs». En: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. Ed. por Yaser Al-Onaizan et al. Miami, Florida, USA: Association for Computational Linguistics, nov. de 2024, págs. 18724-18741. DOI: 10.18653/v1/2024.emnlp-main.1043.

[35] Venkataraman Shankar et al. «Effect of roadway geometrics and environmental factors on rural freeway accident frequencies». En: *Accident Analysis and Prevention* 27.3 (jun. de 1995), págs. 371-389. ISSN: 0001-4575. DOI: 10.1016/0001-4575(94)00078-z.

[36] Mark Poch et al. «Negative Binomial Analysis of Intersection-Accident Frequencies». En: *Journal of Transportation Engineering-asce - J TRANSP ENG-ASCE* 122 (mar. de 1996). DOI: 10.1061/(ASCE)0733-947X(1996)122:2(105).

[37] Zhuoning Yuan et al. «Hetero-ConvLSTM: A Deep Learning Approach to Traffic Accident Prediction on Heterogeneous Spatio-Temporal Data». En: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '18. ACM, jul. de 2018, págs. 984-992. DOI: 10.1145/3219819.3219922.

[38] Xingjian Shi et al. *Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting*. 2015. DOI: 10.48550/ARXIV.1506.04214.

[39] Zhixiong Jin et al. «From Prediction to Prevention: Leveraging Deep Learning in Traffic Accident Prediction Systems». En: *Electronics* 12.20 (oct. de 2023), pág. 4335. ISSN: 2079-9292. DOI: 10.3390/electronics12204335.

[40] Kun-Yu Lin et al. «Predicting Road Traffic Risks with CNN-and-LSTM Learning Over Spatio-Temporal and Multi-Feature Traffic Data». En: *2023 IEEE International Conference on Software Services Engineering (SSE)*. 2023, págs. 305-311. DOI: 10.1109/SSE60056.2023.00049.

[41] Artur Grigorev et al. «Traffic Accident Risk Forecasting using Contextual Vision Transformers with Static Map Generation and Coarse-Fine-Coarse Transformers». En: *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. 2023, págs. 4762-4769. DOI: 10.1109/ITSC57777.2023.10421915.

[42] Mansoor G. Al-Thani et al. «Traffic Transformer: Transformer-based framework for temporal traffic accident prediction». En: *AIMS Mathematics* 9.5 (2024), págs. 12610-12629. ISSN: 2473-6988. DOI: 10.3934/math.2024617.

[43] Beibei Wang et al. «GSNet: Learning Spatial-Temporal Correlations from Geographical and Semantic Aspects for Traffic Accident Risk Forecasting». En: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.5 (mayo de 2021), págs. 4402-4409. ISSN: 2159-5399. DOI: 10.1609/aaai.v35i5.16566.

[44] Le Yu et al. «Deep spatio-temporal graph convolutional network for traffic accident prediction». En: *Neurocomputing* 423 (ene. de 2021), págs. 135-147. ISSN: 0925-2312. DOI: 10.1016/j.neucom.2020.09.043.

[45] Xian Liu et al. «Attention based spatio-temporal graph convolutional network with focal loss for crash risk evaluation on urban road traffic network based on multi-source risks». En: *Accident Analysis and Prevention* 192 (nov. de 2023), pág. 107262. ISSN: 0001-4575. DOI: 10.1016/j.aap.2023.107262.

[46] Noushin Behboudi et al. *Recent Advances in Traffic Accident Analysis and Prediction: A Comprehensive Review of Machine Learning Techniques*. 2024. DOI: 10.48550/ARXIV.2406.13968.

[47] Nitish Shirish Keskar et al. «On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima». En: *arXiv preprint arXiv:1609.04836* (2016).

[48] Adam Paszke et al. «PyTorch: An Imperative Style, High-Performance Deep Learning Library». En: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, págs. 8024-8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[49] PyPI Stats Team. *PyPI Stats: Aggregate download statistics for Python packages*. https://pypistats.org/. 2026. (Visitado 19-01-2026).

[50] Vercel Inc. *Next.js*. https://github.com/vercel/next.js. 2026.

[51] Vladimir Agafonkin. *Leaflet: An open-source JavaScript library for mobile-friendly interactive maps*. https://leafletjs.com/. 2024. (Visitado 19-01-2026).

[52] OpenStreetMap contributors. *Planet dump retrieved from https://planet.openstreetmap.org*. https://www.openstreetmap.org. Copyright: OpenStreetMap is open data, licensed under the Open Data Commons Open Database License (ODbL) by the OpenStreetMap Foundation (OSMF). 2026.

[53] Sebastián Ramírez. *FastAPI*. https://github.com/fastapi/fastapi. 2026. URL: https://fastapi.tiangolo.com.

[54] Evi Nemeth et al. *UNIX and Linux system administration handbook*. 5.ª ed. Pearson Education, 2017. Cap. 9: Periodic Processes.

[55] Patrick Zippenfenig. *Open-Meteo.com Weather API*. Latest release of the Open-Meteo weather API. Licensed under CC-BY-4.0. 2026. DOI: 10.5281/zenodo.7970649.

[56] Dirk Merkel. «Docker: lightweight linux containers for consistent development and deployment». En: *Linux Journal* 2014.239 (2014), pág. 2.

[57] Docker Inc. *Docker Compose documentation*. 2024. URL: https://docs.docker.com/compose/.

[58] F5, Inc. *NGINX Documentation*. 2024. URL: https://nginx.org/en/docs/.

[59] OWASP Foundation. *Web Application Firewall (WAF)*. 2024. URL: https://owasp.org/www-community/Web_Application_Firewall.

[60] Canonical Ltd. *Ubuntu Server Documentation*. Version 22.04 LTS. 2024. URL: https://ubuntu.com/server/docs.

Página intencionalmente dejada en blanco.

Apéndices

Página intencionalmente dejada en blanco.

A

Tablas de resultados

En este capítulo se presentan tablas conteniendo todas las ejecuciones realizadas para cada modelo, con sus respectivos hiperparámetros y métricas de evaluación. El análisis de estos resultados puede encontrarse en el capítulo 4.

A.1. Resultados del modelo RiskViT

La Tabla A.1 presenta los resultados de las 1161 ejecuciones realizadas con el modelo RiskViT, incluyendo la configuración de hiperparámetros y las métricas de evaluación obtenidas.

Tabla A.1: *Resultados de todas las ejecuciones del modelo RiskViT (grilla 12×12), ordenados por Recall descendente*

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
`11b20bae`	10^{-3}	8	1	128	8	8	64	0.1807	0.9257	0.0933
`5ed034bc`	10^{-3}	8	4	128	8	8	128	0.1783	0.9458	0.0888
`c12714f7`	10^{-3}	16	1	128	8	4	128	0.1770	0.9426	0.0918
`d80dabfd`	10^{-3}	16	1	128	4	8	64	0.1765	0.9335	0.0933
`a939d712`	10^{-3}	16	1	64	4	8	32	0.1758	0.9479	0.0918
`2e06e128`	10^{-3}	16	1	64	8	8	64	0.1740	0.9480	0.0929
`157489000`	10^{-3}	16	–	128	4	4	128	0.1724	0.9193	0.0915
`c9b03d6b`	10^{-3}	16	4	128	8	4	32	0.1724	0.9426	0.0916
`4b3504d9`	10^{-3}	16	1	128	4	4	32	0.1717	0.9279	0.0903
`0ad8a7be`	10^{-3}	8	4	32	8	8	32	0.1712	0.9480	0.0932

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
c4b73238	10^{-3}	16	1	64	8	4	128	0.1711	0.9458	0.0908
4a665505	10^{-4}	8	1	64	8	4	128	0.1711	0.9471	0.0907
8b4ee400	10^{-3}	16	2	32	8	4	128	0.1706	0.9357	0.0875
7ad0adae	10^{-3}	16	4	128	8	4	64	0.1705	0.9480	0.0920
4e22cbd5	10^{-3}	8	4	64	8	4	128	0.1705	0.9404	0.0914
0182bfe3	10^{-3}	8	1	32	4	4	64	0.1703	0.9475	0.0910
3b04b29c	10^{-3}	8	4	64	4	8	32	0.1702	0.9480	0.0911
f586e722	10^{-3}	16	4	64	8	8	32	0.1700	0.9480	0.0909
1d32a15b	10^{-3}	16	4	32	4	8	32	0.1698	0.9258	0.0902
3cb346ad	10^{-3}	8	1	64	8	8	128	0.1697	0.9480	0.0903
9347f309	10^{-3}	16	4	64	8	4	32	0.1696	0.9473	0.0923
2578da83	10^{-3}	16	2	32	8	8	64	0.1695	0.9377	0.0908
25de8ca2	10^{-4}	8	1	128	4	4	32	0.1694	0.9466	0.0900
2b8778c4	10^{-3}	16	1	32	4	4	128	0.1694	0.9449	0.0903
110ffdfa	10^{-3}	16	4	32	8	4	64	0.1692	0.9423	0.0929
c1b9ea1f	10^{-4}	8	1	128	4	8	32	0.1688	0.9479	0.0904
6008858d	10^{-3}	8	4	128	8	4	32	0.1687	0.9480	0.0919
90f4feb9	10^{-5}	16	4	32	4	8	32	0.1686	0.9479	0.0906
1c24923e	10^{-3}	8	4	128	4	8	64	0.1684	0.9438	0.0898
30ebcfb5	10^{-3}	16	4	32	4	8	32	0.1682	0.9457	0.0894
cd1234de	10^{-3}	16	4	128	4	8	128	0.1680	0.9440	0.0912
3438dc07	10^{-4}	16	4	128	4	8	32	0.1678	0.9480	0.0916
fba7396b	10^{-3}	16	1	128	4	8	64	0.1677	0.9474	0.0902
82aa6c98	10^{-4}	8	2	128	4	4	64	0.1677	0.9480	0.0910
121c7d6f	10^{-3}	16	1	64	4	8	64	0.1676	0.9454	0.0901
c3401704	10^{-3}	8	2	64	4	4	32	0.1675	0.9425	0.0923
22a5298c	10^{-3}	8	2	128	8	8	64	0.1675	0.9292	0.0888
3d196dd3	10^{-3}	16	4	128	8	8	128	0.1673	0.9480	0.0904
9f399991	10^{-3}	8	2	32	8	4	64	0.1672	0.9258	0.0893
ae9f3166	10^{-3}	8	2	128	8	8	64	0.1670	0.9480	0.0916
50644ee7	10^{-3}	16	4	32	8	4	128	0.1669	0.9379	0.0897
f5c312d2	10^{-3}	16	2	64	8	8	128	0.1667	0.9195	0.0884
3098f14b	10^{-4}	16	1	32	4	8	64	0.1667	0.9480	0.0914
2f4e6e41	10^{-3}	8	1	32	4	8	64	0.1666	0.9246	0.0892
d0264e3b	10^{-4}	8	1	128	8	4	64	0.1666	0.9472	0.0912
a010a776	10^{-4}	8	2	64	4	8	64	0.1666	0.9475	0.0905
d7a5d299	10^{-3}	16	2	64	4	4	128	0.1662	0.9252	0.0881
c30c6608	10^{-4}	8	1	128	8	4	128	0.1662	0.9403	0.0905
88b574cd	10^{-3}	16	1	64	8	8	32	0.1661	0.9480	0.0902

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
a84be19c	10^{-3}	8	1	128	8	4	128	0.1660	0.9480	0.0917
36669a6d	10^{-3}	8	1	32	8	8	32	0.1660	0.9268	0.0894
a97ecd59	10^{-3}	8	2	128	4	8	64	0.1660	0.9191	0.0899
1afeb7c9	10^{-3}	8	2	64	4	8	32	0.1659	0.9480	0.0897
069ac190	10^{-4}	8	1	64	4	4	32	0.1658	0.9471	0.0917
2b3c7bf1	10^{-3}	16	1	64	4	8	128	0.1658	0.9476	0.0907
a6f937b0	10^{-3}	8	2	64	8	8	32	0.1657	0.9426	0.0924
e4ccd125	10^{-3}	8	1	32	8	4	128	0.1656	0.9465	0.0901
73d47eab	10^{-3}	16	4	32	8	4	64	0.1656	0.9322	0.0898
14594dfc	10^{-3}	16	4	128	4	4	32	0.1656	0.9405	0.0917
cc8c66cb	10^{-4}	16	4	128	8	4	128	0.1656	0.9454	0.0913
3448caa0	10^{-4}	8	4	32	4	8	32	0.1656	0.9479	0.0922
5d70737b	10^{-4}	16	1	128	4	8	128	0.1655	0.9467	0.0895
fdb6564b	10^{-3}	8	2	64	4	4	64	0.1650	0.9184	0.0898
ea605893	10^{-4}	16	1	64	8	8	32	0.1650	0.9479	0.0906
7ed47303	10^{-4}	8	4	64	4	8	64	0.1650	0.9479	0.0910
bd03b9e3	10^{-4}	16	1	128	4	4	32	0.1649	0.9480	0.0914
ed86d0ec	10^{-3}	16	1	32	4	8	64	0.1648	0.9434	0.0903
bb71df7c	10^{-3}	16	2	64	8	8	128	0.1648	0.9412	0.0887
b09d8a04	10^{-4}	16	4	64	8	8	128	0.1648	0.9480	0.0903
4ca79af9	10^{-4}	8	2	128	8	8	128	0.1646	0.9480	0.0903
2950c8ad	10^{-4}	8	1	32	8	4	32	0.1646	0.9478	0.0901
37ca0105	10^{-3}	8	1	64	4	8	128	0.1646	0.9476	0.0913
6e2e5127	10^{-3}	16	1	64	8	8	64	0.1645	0.9282	0.0898
56ab42eb	10^{-3}	16	2	64	8	4	128	0.1645	0.9463	0.0876
f0d4ea60	10^{-5}	16	4	32	4	4	64	0.1645	0.9480	0.0911
5a4bfe8c	10^{-3}	16	1	64	8	4	32	0.1644	0.9480	0.0914
976d8b56	10^{-3}	8	4	128	4	4	32	0.1643	0.9384	0.0881
1fefd820	10^{-3}	16	2	128	8	8	64	0.1643	0.9430	0.0899
836adf3d	10^{-4}	16	2	128	4	4	128	0.1643	0.9480	0.0909
7a746e5a	10^{-3}	8	1	64	8	4	128	0.1642	0.9275	0.0889
234c2bd4	10^{-4}	8	2	128	4	4	128	0.1642	0.9480	0.0907
e5f643e4	10^{-5}	16	2	64	4	8	32	0.1641	0.9480	0.0913
98384016	10^{-5}	8	4	32	4	8	32	0.1641	0.9480	0.0903
f58ccd1c	10^{-3}	16	1	32	8	8	32	0.1640	0.9480	0.0908
af706f3f	10^{-3}	8	4	128	4	8	32	0.1640	0.9393	0.0909
fed7a303	10^{-4}	8	2	64	8	4	128	0.1640	0.9473	0.0899
1426b20e	10^{-4}	16	4	32	4	8	128	0.1640	0.9480	0.0912
4b5d2842	10^{-5}	16	2	32	4	8	128	0.1639	0.9480	0.0902

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
3b0d2424	10^{-5}	16	2	64	4	4	64	0.1639	0.9480	0.0909
1d220578	10^{-3}	16	1	32	4	4	32	0.1639	0.9480	0.0893
22362c8f	10^{-3}	8	1	64	8	8	64	0.1639	0.9219	0.0907
871100000000	10^{-3}	16	–	32	8	4	64	0.1638	0.9480	0.0889
ce254064	10^{-3}	8	2	128	4	8	128	0.1638	0.9446	0.0881
290d9e43	10^{-4}	8	1	64	8	4	64	0.1637	0.9480	0.0910
ca14cfdc	10^{-3}	8	2	64	4	4	128	0.1637	0.9406	0.0907
ea12544d	10^{-3}	16	1	64	8	4	64	0.1637	0.9178	0.0891
20b33588	10^{-4}	8	2	64	4	4	64	0.1637	0.9472	0.0904
62cb95e7	10^{-4}	16	2	64	8	4	128	0.1635	0.9480	0.0911
70d02505	10^{-5}	16	1	128	4	8	32	0.1635	0.9479	0.0903
213677ab	10^{-5}	16	1	32	8	4	64	0.1635	0.9480	0.0907
4351dcb9	10^{-3}	16	1	128	8	8	128	0.1635	0.9216	0.0893
76f174aa	10^{-4}	16	2	128	4	4	64	0.1634	0.9453	0.0909
b7653462	10^{-5}	8	1	64	8	4	64	0.1633	0.9480	0.0909
82552657	10^{-4}	8	2	64	4	4	32	0.1633	0.9480	0.0912
64669650	10^{-5}	16	1	128	4	4	32	0.1633	0.9480	0.0902
6a1071a6	10^{-3}	8	1	64	4	4	128	0.1633	0.9462	0.0896
56eb17be	10^{-5}	8	1	128	8	4	128	0.1633	0.9480	0.0909
f27d125e	10^{-4}	8	2	64	8	4	64	0.1633	0.9478	0.0906
4c2ebf42	10^{-4}	16	2	128	4	8	64	0.1632	0.9479	0.0900
4590d554	10^{-5}	16	1	32	4	8	32	0.1632	0.9480	0.0902
d0a93289	10^{-3}	8	2	128	8	4	32	0.1631	0.9319	0.0900
9397c660	10^{-4}	16	1	64	4	8	128	0.1631	0.9479	0.0895
93d4c69d	10^{-5}	16	1	64	4	4	128	0.1631	0.9470	0.0906
b8d2e6ae	10^{-3}	16	2	128	8	4	64	0.1630	0.9478	0.0923
12b4b6b5	10^{-5}	8	2	64	4	8	32	0.1630	0.9479	0.0902
17171bb6	10^{-3}	8	2	64	8	8	32	0.1630	0.9338	0.0888
d6ceb36c	10^{-3}	8	4	64	4	8	64	0.1629	0.9461	0.0900
37c92919	10^{-3}	16	4	32	4	8	128	0.1629	0.9243	0.0897
c04aa3ce	10^{-4}	16	4	128	8	4	64	0.1629	0.9476	0.0901
4d53629e	10^{-4}	8	2	32	8	4	64	0.1629	0.9480	0.0912
b09d5ee0	10^{-3}	8	2	32	4	8	128	0.1628	0.9478	0.0898
d9a2fd49	10^{-4}	8	1	64	4	8	128	0.1628	0.9439	0.0901
d29ce6ec	10^{-3}	8	2	32	8	8	128	0.1628	0.9480	0.0905
af3fffd4	10^{-4}	16	2	128	4	4	32	0.1628	0.9480	0.0905
77daebb8	10^{-3}	8	4	64	4	4	64	0.1627	0.9383	0.0906
42c6e19e	10^{-5}	8	4	64	4	4	32	0.1627	0.9480	0.0908
373b4c96	10^{-3}	8	2	32	4	8	32	0.1627	0.9479	0.0907

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
9bce387f	10^{-4}	8	4	32	8	8	32	0.1627	0.9457	0.0895
47374064	10^{-4}	16	1	64	4	8	128	0.1627	0.9480	0.0909
8b1540c0	10^{-3}	8	2	128	8	8	128	0.1627	0.9480	0.0911
85df5ae5	10^{-4}	16	2	128	4	8	32	0.1627	0.9480	0.0906
81a0a464	10^{-3}	8	2	32	4	8	64	0.1627	0.9455	0.0892
7f75c99f	10^{-3}	16	4	32	4	8	64	0.1627	0.9475	0.0908
9186d1ee	10^{-5}	16	4	64	4	4	32	0.1627	0.9480	0.0909
f9a2f0f3	10^{-4}	16	4	32	4	4	64	0.1626	0.9480	0.0907
d1861e77	10^{-5}	8	4	32	8	4	32	0.1626	0.9479	0.0908
c1bed0bb	10^{-5}	8	1	32	4	4	128	0.1626	0.9479	0.0907
3499d9c7	10^{-4}	16	2	128	8	4	32	0.1626	0.9471	0.0889
918976b2	10^{-5}	16	1	32	8	4	128	0.1626	0.9480	0.0905
32cc509b	10^{-4}	16	2	64	4	8	32	0.1625	0.9479	0.0911
ce803a52	10^{-4}	8	2	128	8	4	128	0.1625	0.9452	0.0886
ae353343	10^{-4}	16	2	64	4	4	64	0.1625	0.9480	0.0910
6fbf1fd8	10^{-3}	8	2	128	8	4	32	0.1624	0.9480	0.0901
6672d905	10^{-5}	16	2	64	8	8	32	0.1624	0.9480	0.0915
4a30e828	10^{-3}	16	4	32	8	8	128	0.1624	0.9423	0.0907
2e03de15	10^{-5}	16	4	32	8	4	32	0.1624	0.9480	0.0909
f3e006d2	10^{-4}	8	2	64	4	8	32	0.1623	0.9479	0.0908
8d009276	10^{-4}	16	1	64	8	4	128	0.1623	0.9478	0.0902
917c1f0c	10^{-4}	8	2	64	4	4	32	0.1623	0.9475	0.0879
e78787ac	10^{-3}	16	2	64	8	8	32	0.1623	0.9254	0.0893
a6e47355	10^{-3}	16	1	32	4	8	128	0.1622	0.9292	0.0907
a570a79c	10^{-5}	8	4	32	4	4	32	0.1620	0.9480	0.0906
9d8c8077	10^{-4}	8	2	64	8	4	32	0.1620	0.9478	0.0910
c0fd1086	10^{-4}	8	2	32	4	4	32	0.1620	0.9480	0.0903
cdf73a94	10^{-3}	16	1	64	4	4	32	0.1620	0.9480	0.0902
ad422209	10^{-4}	16	1	64	4	8	64	0.1620	0.9479	0.0910
90499b10	10^{-3}	8	2	128	4	4	32	0.1619	0.9244	0.0872
6e699abc	10^{-5}	16	2	32	4	8	64	0.1619	0.9480	0.0896
9382f2c5	10^{-4}	8	1	128	8	8	64	0.1619	0.9475	0.0905
7016c999	10^{-5}	16	2	32	8	4	128	0.1618	0.9480	0.0907
8b43525e	10^{-3}	8	4	64	4	4	32	0.1618	0.9464	0.0900
421f8371	10^{-4}	16	2	32	4	4	32	0.1618	0.9480	0.0908
70d9b8f6	10^{-5}	16	2	128	4	8	128	0.1618	0.9480	0.0906
bbe04f7a	10^{-4}	8	4	64	4	4	128	0.1618	0.9479	0.0908
578b3392	10^{-5}	8	2	128	8	4	128	0.1618	0.9480	0.0903
083b71bd	10^{-3}	8	1	64	4	8	32	0.1618	0.9301	0.0877

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
b7511093	10^{-3}	16	1	32	8	8	64	0.1618	0.9374	0.0904
cbe4bd2d	10^{-3}	8	1	64	8	8	32	0.1617	0.9480	0.0907
aa5cbfa1	10^{-5}	16	4	32	4	4	128	0.1617	0.9480	0.0898
b1c8f4d7	10^{-3}	16	2	64	4	8	128	0.1617	0.9427	0.0898
2ea2902e	10^{-4}	16	4	32	8	8	32	0.1617	0.9463	0.0895
e9989d1d	10^{-3}	8	2	64	4	8	64	0.1617	0.9462	0.0905
18cca912	10^{-5}	8	4	128	8	8	64	0.1617	0.9480	0.0902
9f0b4d67	10^{-3}	16	1	64	8	4	128	0.1617	0.9308	0.0872
359a8776	10^{-4}	16	2	128	4	4	128	0.1617	0.9453	0.0893
b6a85c57	10^{-5}	16	2	32	4	8	32	0.1617	0.9480	0.0905
cf74ad94	10^{-3}	8	1	32	8	8	32	0.1617	0.9480	0.0907
a2527d15	10^{-4}	16	2	32	8	4	128	0.1616	0.9478	0.0897
b36cb173	10^{-5}	16	2	128	4	4	128	0.1616	0.9480	0.0902
85524369	10^{-5}	16	1	64	4	8	64	0.1616	0.9480	0.0907
1e20b5cb	10^{-4}	16	4	32	8	4	64	0.1616	0.9479	0.0913
43e19c8d	10^{-4}	16	1	64	4	4	128	0.1616	0.9480	0.0919
dd87ce26	10^{-5}	16	2	32	8	4	64	0.1616	0.9480	0.0907
1584f45a	10^{-4}	8	2	128	4	8	128	0.1616	0.9457	0.0892
e62f13f0	10^{-5}	16	1	128	4	8	64	0.1616	0.9479	0.0905
7ac4d669	10^{-5}	16	4	128	4	4	32	0.1616	0.9480	0.0914
518f122a	10^{-5}	8	1	64	4	8	32	0.1616	0.9480	0.0907
630aee0b	10^{-3}	8	2	32	8	8	32	0.1616	0.9403	0.0901
c9afdf47	10^{-4}	8	2	128	8	8	128	0.1616	0.9469	0.0894
7505b888	10^{-3}	16	2	64	8	4	32	0.1615	0.9377	0.0898
3.038E+019	10^{-5}	16	–	64	8	8	128	0.1615	0.9480	0.0902
722e825e	10^{-4}	16	4	32	8	8	64	0.1615	0.9480	0.0913
79cb30fb	10^{-4}	16	1	64	4	4	64	0.1614	0.9480	0.0907
50e3b79e	10^{-5}	8	1	128	4	4	32	0.1614	0.9480	0.0909
31d43397	10^{-4}	16	1	32	4	4	32	0.1614	0.9480	0.0904
478a020d	10^{-5}	16	1	64	4	4	32	0.1614	0.9479	0.0904
197e653d	10^{-5}	8	2	128	8	8	32	0.1614	0.9480	0.0908
46f096ca	10^{-4}	8	1	128	4	4	128	0.1613	0.9480	0.0901
b638392f	10^{-3}	16	2	32	8	4	128	0.1613	0.9458	0.0892
b85e54ed	10^{-5}	16	2	64	8	4	128	0.1613	0.9480	0.0907
34a60c00	10^{-5}	16	2	64	8	8	128	0.1613	0.9480	0.0907
07c8a9e8	10^{-5}	8	1	64	4	4	64	0.1613	0.9480	0.0897
8f5acfed	10^{-3}	8	2	32	8	4	128	0.1613	0.9263	0.0872
8dfa5219	10^{-5}	16	4	128	4	8	32	0.1613	0.9480	0.0911
d32dc637	10^{-5}	8	4	128	4	8	64	0.1612	0.9480	0.0907

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
afb71b59	10^{-5}	8	4	128	4	4	128	0.1612	0.9480	0.0907
1e89fb75	10^{-3}	8	1	32	4	8	32	0.1612	0.9480	0.0909
7757cac5	10^{-5}	8	1	32	8	4	128	0.1612	0.9479	0.0904
2a657e2c	10^{-5}	8	4	32	4	8	128	0.1612	0.9477	0.0905
50a7a23b	10^{-5}	8	1	32	8	4	128	0.1612	0.9480	0.0907
7194ea2a	10^{-5}	16	1	128	8	8	128	0.1612	0.9480	0.0907
07739aad	10^{-4}	8	2	32	4	4	32	0.1612	0.9480	0.0899
21311a3c	10^{-4}	16	2	32	4	8	32	0.1611	0.9480	0.0907
f5f3527d	10^{-4}	8	1	32	4	8	64	0.1611	0.9480	0.0899
006d4c1f	10^{-3}	8	2	128	8	8	32	0.1611	0.9358	0.0890
abcb688b	10^{-5}	8	1	128	8	8	32	0.1611	0.9480	0.0907
bfed2e23	10^{-3}	16	1	128	8	8	32	0.1611	0.9328	0.0904
fc41b227	10^{-5}	16	4	64	8	8	32	0.1611	0.9480	0.0908
83bf28ae	10^{-3}	16	2	32	4	4	64	0.1611	0.9480	0.0893
3c61d201	10^{-5}	8	1	128	4	8	64	0.1611	0.9478	0.0908
5c318766	10^{-4}	16	2	32	8	8	32	0.1610	0.9480	0.0905
f3a8a17c	10^{-4}	8	1	32	4	8	32	0.1610	0.9480	0.0904
3491097d	10^{-4}	8	4	64	8	8	64	0.1610	0.9447	0.0908
dbaa2c78	10^{-3}	16	2	64	4	4	64	0.1610	0.9434	0.0897
4321eefa	10^{-5}	8	2	128	8	4	64	0.1610	0.9480	0.0901
d456a61b	10^{-5}	16	4	32	4	4	128	0.1610	0.9480	0.0904
0a2ed705	10^{-4}	8	4	128	4	8	128	0.1610	0.9448	0.0879
144c4dec	10^{-4}	16	1	128	4	8	128	0.1610	0.9480	0.0909
4ae18f04	10^{-5}	16	1	32	8	8	128	0.1610	0.9480	0.0907
98b61ea1	10^{-5}	16	1	64	4	4	64	0.1610	0.9480	0.0907
e89be9b8	10^{-5}	8	2	128	4	8	64	0.1610	0.9480	0.0906
ee9c6ea2	10^{-5}	16	4	64	4	8	128	0.1610	0.9480	0.0904
df4a5c02	10^{-3}	8	1	64	8	4	64	0.1609	0.9260	0.0880
acb7957c	10^{-5}	8	4	32	4	8	128	0.1609	0.9480	0.0906
f0505f4b	10^{-5}	8	2	32	4	4	128	0.1609	0.9480	0.0907
0a1699a9	10^{-5}	16	1	128	8	4	64	0.1609	0.9480	0.0904
17f38144	10^{-4}	16	4	128	4	4	128	0.1609	0.9480	0.0904
56f4ae83	10^{-4}	16	2	64	8	4	32	0.1609	0.9480	0.0911
952264d1	10^{-3}	16	4	128	4	8	64	0.1609	0.9433	0.0917
6c53f3fe	10^{-3}	16	4	32	4	4	32	0.1609	0.9215	0.0884
8f239dd9	10^{-3}	16	1	32	4	8	128	0.1608	0.9413	0.0889
aa191726	10^{-4}	8	1	64	4	8	64	0.1608	0.9480	0.0906
6.3067E+019	10^{-3}	8	–	32	4	8	128	0.1608	0.9300	0.0876
96c5f7d7	10^{-3}	8	4	32	4	8	32	0.1608	0.9336	0.0903

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
2b14caad	10^{-4}	8	2	32	8	8	32	0.1608	0.9480	0.0905
f9f7ac16	10^{-5}	16	2	128	8	8	32	0.1608	0.9480	0.0901
2ab98316	10^{-4}	8	2	128	4	4	32	0.1608	0.9462	0.0912
9240a9b8	10^{-4}	8	1	32	4	4	32	0.1608	0.9480	0.0907
9fd2b510	10^{-5}	16	2	32	4	4	32	0.1608	0.9480	0.0900
3f12e869	10^{-5}	16	1	64	4	4	64	0.1607	0.9480	0.0897
3ed01ccc	10^{-3}	16	2	64	4	8	32	0.1607	0.9480	0.0903
7f015ae4	10^{-4}	16	2	64	4	4	32	0.1607	0.9480	0.0906
20cd6071	10^{-4}	16	1	128	8	8	64	0.1607	0.9480	0.0900
74228119	10^{-5}	16	2	32	8	8	128	0.1607	0.9480	0.0906
ae992adb	10^{-5}	16	2	64	4	4	128	0.1607	0.9480	0.0907
fd15ce40	10^{-5}	8	4	64	8	8	64	0.1607	0.9480	0.0906
6abecfdb	10^{-5}	8	2	32	8	4	64	0.1607	0.9480	0.0907
c45ec91b	10^{-3}	8	1	64	4	8	32	0.1607	0.9480	0.0909
256af771	10^{-5}	8	2	64	4	4	64	0.1607	0.9479	0.0902
090240b6	10^{-5}	16	2	128	8	4	64	0.1607	0.9480	0.0905
7e3c8e21	10^{-5}	16	2	32	8	8	64	0.1607	0.9480	0.0899
b4ae6f37	10^{-5}	8	4	32	8	4	64	0.1606	0.9480	0.0905
3f83e1ad	10^{-5}	8	2	32	8	8	128	0.1606	0.9480	0.0897
e3bf2a23	10^{-5}	16	2	64	8	8	64	0.1606	0.9479	0.0905
54d270f5	10^{-3}	8	1	64	4	4	32	0.1606	0.9480	0.0909
d8dd304a	10^{-4}	8	2	32	4	8	64	0.1606	0.9480	0.0897
3e8db1c6	10^{-4}	8	1	64	4	8	32	0.1606	0.9480	0.0904
97ed0eab	10^{-5}	8	2	64	4	4	32	0.1606	0.9480	0.0906
d00565f4	10^{-3}	16	2	32	8	8	128	0.1606	0.9362	0.0885
1ed13d45	10^{-5}	16	1	64	4	8	32	0.1606	0.9480	0.0905
6a90d1a3	10^{-4}	8	4	32	4	4	64	0.1606	0.9480	0.0909
da246fa1	10^{-5}	16	2	64	4	4	64	0.1606	0.9480	0.0905
5de3112b	10^{-4}	8	1	32	4	8	128	0.1606	0.9469	0.0900
86534454	10^{-5}	8	2	64	8	8	64	0.1605	0.9479	0.0903
1a9683b1	10^{-3}	16	1	32	8	8	128	0.1605	0.9480	0.0908
ecbc56fd	10^{-4}	16	1	128	8	8	32	0.1605	0.9480	0.0892
9a6c8080	10^{-4}	8	4	32	8	4	64	0.1605	0.9469	0.0895
28962d15	10^{-4}	16	1	32	8	4	32	0.1605	0.9480	0.0904
e1c2af3a	10^{-5}	16	2	32	8	8	32	0.1605	0.9479	0.0901
adf85e6f	10^{-5}	16	2	32	4	4	64	0.1605	0.9480	0.0897
c46ba303	10^{-3}	16	2	128	8	4	32	0.1605	0.9272	0.0928
475dacf1	10^{-5}	8	2	64	8	4	64	0.1605	0.9480	0.0894
be3be5d5	10^{-4}	8	4	128	4	4	32	0.1605	0.9462	0.0897

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
59d3f743	10^{-4}	8	1	128	8	4	128	0.1605	0.9455	0.0898
d7f8b97d	10^{-4}	16	1	128	4	8	32	0.1605	0.9475	0.0895
74e9d37c	10^{-3}	8	1	128	4	8	128	0.1605	0.9480	0.0895
daa61bb1	10^{-5}	16	2	128	8	4	32	0.1605	0.9478	0.0905
2b442f50	10^{-4}	16	2	128	8	8	64	0.1604	0.9479	0.0903
e40aac87	10^{-4}	16	1	64	8	8	64	0.1604	0.9479	0.0898
788fcb8e	10^{-4}	8	4	64	8	8	32	0.1604	0.9480	0.0904
fcbe62f5	10^{-5}	16	4	64	8	4	64	0.1604	0.9480	0.0904
a705d953	10^{-4}	8	4	64	4	8	128	0.1604	0.9480	0.0901
34188696	10^{-4}	16	4	64	8	8	64	0.1604	0.9480	0.0902
417d4f6b	10^{-4}	8	1	32	4	4	64	0.1604	0.9480	0.0907
92b446a0	10^{-4}	8	1	128	4	4	64	0.1604	0.9480	0.0900
00897a3a	10^{-4}	16	2	64	8	4	128	0.1604	0.9480	0.0905
d5714d3b	10^{-3}	16	2	32	4	4	128	0.1603	0.9480	0.0891
c5c98c94	10^{-5}	16	1	32	4	4	32	0.1603	0.9480	0.0907
fa474c16	10^{-4}	8	2	32	4	4	64	0.1603	0.9480	0.0906
a6fad404	10^{-4}	8	1	32	4	4	64	0.1603	0.9479	0.0905
ec5e9c58	10^{-5}	16	4	128	8	4	128	0.1603	0.9480	0.0906
dd56ec3e	10^{-5}	16	2	32	4	4	64	0.1603	0.9480	0.0906
794370eb	10^{-5}	16	2	32	4	8	128	0.1603	0.9480	0.0902
a7aa0561	10^{-5}	16	4	32	8	4	64	0.1603	0.9480	0.0905
9f530f74	10^{-5}	16	1	64	4	8	64	0.1603	0.9480	0.0906
240d0720	10^{-4}	8	1	32	8	4	128	0.1603	0.9480	0.0906
98e9cc3c	10^{-5}	16	2	128	4	4	32	0.1603	0.9480	0.0906
c2dc7687	10^{-3}	16	2	128	8	4	32	0.1603	0.9480	0.0892
da0416fd	10^{-4}	8	2	64	8	8	128	0.1603	0.9449	0.0900
7879674	10^{-5}	8	–	32	8	4	32	0.1603	0.9480	0.0905
91677c0b	10^{-4}	8	2	128	4	8	32	0.1603	0.9470	0.0902
b75cefdc	10^{-5}	8	2	64	8	8	64	0.1603	0.9480	0.0907
88e212c1	10^{-5}	8	4	32	4	4	128	0.1603	0.9480	0.0907
906d1e9b	10^{-5}	8	2	128	4	4	128	0.1603	0.9479	0.0905
f835cc2d	10^{-5}	8	1	32	8	8	64	0.1603	0.9480	0.0904
52ab3b1e	10^{-5}	8	1	128	8	8	64	0.1603	0.9480	0.0905
12208998	10^{-5}	16	1	128	4	8	128	0.1602	0.9480	0.0898
69b2fd36	10^{-5}	8	1	32	4	8	128	0.1602	0.9480	0.0899
6dabd1af	10^{-4}	16	4	64	8	4	32	0.1602	0.9480	0.0910
13652428	10^{-5}	16	4	32	4	8	128	0.1602	0.9480	0.0903
791adac9	10^{-4}	16	2	64	4	8	128	0.1602	0.9480	0.0902
38780aff	10^{-4}	8	2	64	4	8	128	0.1602	0.9478	0.0898

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
c75154a8	10^{-5}	16	1	64	4	4	128	0.1602	0.9480	0.0907
24fdd709	10^{-5}	8	2	64	8	8	128	0.1602	0.9480	0.0901
b33858cb	10^{-5}	8	1	128	8	8	64	0.1602	0.9480	0.0901
760b245d	10^{-4}	16	1	32	8	4	64	0.1602	0.9478	0.0915
3b6bb882	10^{-5}	8	1	64	8	8	32	0.1602	0.9480	0.0905
52b32600	10^{-4}	16	1	128	8	4	128	0.1601	0.9480	0.0908
c53e1896	10^{-5}	16	2	64	4	4	32	0.1601	0.9480	0.0905
e8bf3b6a	10^{-4}	16	1	32	4	4	32	0.1601	0.9480	0.0905
7313b5a4	10^{-4}	8	2	32	8	4	128	0.1601	0.9480	0.0907
f3f5b993	10^{-5}	8	1	128	4	4	128	0.1601	0.9480	0.0906
8e02b97a	10^{-5}	8	4	32	4	4	64	0.1601	0.9480	0.0905
2e78a963	10^{-5}	8	2	64	4	8	64	0.1601	0.9480	0.0904
74d75d99	10^{-5}	8	2	32	8	4	128	0.1601	0.9480	0.0905
4c802f16	10^{-5}	8	1	32	4	8	64	0.1601	0.9480	0.0906
ef47e27a	10^{-5}	16	1	64	8	8	128	0.1601	0.9480	0.0906
b1b9bf0c	10^{-5}	8	1	64	4	8	128	0.1601	0.9480	0.0907
acd20772	10^{-5}	8	1	32	4	4	32	0.1601	0.9480	0.0905
97187f31	10^{-5}	8	1	64	4	8	128	0.1601	0.9480	0.0900
2a1b415b	10^{-3}	16	4	64	8	8	64	0.1601	0.9451	0.0888
ad3b19c1	10^{-4}	16	1	64	4	8	32	0.1601	0.9480	0.0915
0d251010	10^{-5}	16	1	64	8	4	64	0.1601	0.9479	0.0900
a0f1f988	10^{-4}	16	2	128	8	8	32	0.1601	0.9476	0.0900
684e0bc6	10^{-3}	8	4	64	8	8	128	0.1601	0.9473	0.0901
6641ff58	10^{-4}	8	1	64	8	8	32	0.1601	0.9480	0.0894
9a0a0639	10^{-5}	8	1	64	4	8	64	0.1601	0.9479	0.0907
9a2cb16e	10^{-5}	16	4	64	8	4	128	0.1601	0.9480	0.0905
3fcd99cb	10^{-3}	16	2	32	8	8	64	0.1601	0.9480	0.0895
554b1178	10^{-4}	8	4	32	8	8	32	0.1601	0.9475	0.0899
2bac69d9	10^{-4}	16	2	32	8	4	32	0.1601	0.9480	0.0907
1887ed3b	10^{-5}	16	1	128	4	4	64	0.1601	0.9480	0.0905
8ef97d01	10^{-3}	8	1	32	4	8	128	0.1601	0.9469	0.0910
b1536cfc	10^{-4}	8	2	64	4	8	128	0.1601	0.9480	0.0899
044f4d0b	10^{-5}	8	4	128	4	4	64	0.1601	0.9480	0.0899
27568a92	10^{-5}	16	4	128	4	4	128	0.1600	0.9480	0.0907
2491a1ef	10^{-5}	8	1	64	8	4	32	0.1600	0.9480	0.0904
94e421e6	10^{-3}	16	1	64	4	4	128	0.1600	0.9477	0.0899
7fadfa7f	10^{-4}	8	2	32	8	4	64	0.1600	0.9480	0.0910
f4786d32	10^{-5}	16	2	128	8	4	64	0.1600	0.9479	0.0898
b726826f	10^{-3}	8	1	128	4	4	128	0.1600	0.9480	0.0900

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
28bd6275	10^{-5}	16	2	32	4	4	32	0.1600	0.9480	0.0909
90390110	10^{-3}	8	1	128	4	4	32	0.1600	0.9480	0.0908
82ee2044	10^{-4}	8	4	128	4	4	128	0.1600	0.9480	0.0915
22b8ad35	10^{-3}	8	2	64	4	4	64	0.1600	0.9448	0.0887
bdae6be4	10^{-5}	16	4	128	8	8	64	0.1600	0.9480	0.0905
f4a73492	10^{-5}	16	1	128	4	8	128	0.1599	0.9480	0.0900
05f9400e	10^{-5}	8	2	64	4	4	64	0.1599	0.9480	0.0907
011cafa5	10^{-5}	8	2	32	4	8	64	0.1599	0.9480	0.0904
83909688	10^{-4}	16	4	32	8	8	128	0.1599	0.9480	0.0900
ef3895a8	10^{-3}	8	2	32	4	8	64	0.1599	0.9276	0.0883
503316a9	10^{-5}	8	1	32	8	4	32	0.1599	0.9480	0.0893
941bb928	10^{-5}	8	4	128	4	8	32	0.1599	0.9480	0.0905
e3dca700	10^{-5}	16	1	128	8	8	32	0.1599	0.9480	0.0888
02bf0461	10^{-5}	16	1	64	8	8	32	0.1599	0.9480	0.0901
8293a2b5	10^{-5}	16	2	128	8	4	128	0.1599	0.9480	0.0894
b1581aa4	10^{-4}	16	1	32	8	4	32	0.1599	0.9480	0.0904
3c0ea1d2	10^{-5}	8	2	64	8	8	32	0.1599	0.9480	0.0901
bdaef1d4	10^{-5}	8	2	64	8	4	32	0.1599	0.9480	0.0906
d2429dcb	10^{-3}	8	4	32	8	8	128	0.1599	0.9460	0.0880
5e9a080a	10^{-5}	16	1	32	4	8	128	0.1598	0.9480	0.0905
381608eb	10^{-5}	16	2	32	8	8	32	0.1598	0.9480	0.0901
dc6d3953	10^{-4}	16	1	32	4	8	128	0.1598	0.9480	0.0906
a5adfd46	10^{-5}	16	1	32	8	8	128	0.1598	0.9480	0.0910
3bff0561	10^{-3}	8	1	32	8	8	128	0.1598	0.9480	0.0915
982d4a79	10^{-4}	16	1	32	4	8	64	0.1598	0.9480	0.0905
19327b74	10^{-5}	8	1	128	4	4	64	0.1598	0.9480	0.0903
8d9a963d	10^{-3}	8	4	128	8	8	64	0.1598	0.9480	0.0902
95e30e0e	10^{-5}	8	4	64	4	8	32	0.1597	0.9480	0.0907
9457480a	10^{-5}	8	2	32	8	4	128	0.1597	0.9480	0.0898
ed7ce5e2	10^{-4}	16	4	32	8	4	64	0.1597	0.9480	0.0904
743b2e67	10^{-5}	8	2	64	4	8	32	0.1597	0.9480	0.0905
00db2916	10^{-5}	8	2	32	8	8	128	0.1597	0.9480	0.0906
6997d9b1	10^{-5}	8	2	32	8	4	64	0.1597	0.9480	0.0904
232f3485	10^{-5}	8	2	32	4	4	64	0.1597	0.9480	0.0905
4c1bc5c5	10^{-5}	8	2	32	4	8	128	0.1597	0.9480	0.0906
009fbd7a	10^{-5}	16	1	32	8	4	128	0.1597	0.9480	0.0906
2d7bf049	10^{-5}	8	1	64	8	8	128	0.1597	0.9480	0.0905
06975fe9	10^{-5}	8	1	64	8	8	32	0.1597	0.9480	0.0903
8789531a	10^{-5}	8	1	32	4	4	128	0.1597	0.9480	0.0906

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
95cca6cc	10^{-5}	16	4	64	4	4	64	0.1597	0.9480	0.0903
5507b46f	10^{-4}	8	4	32	8	4	64	0.1597	0.9480	0.0903
60791b43	10^{-5}	8	1	64	4	8	64	0.1597	0.9480	0.0906
6d17ab7e	10^{-4}	16	4	32	4	8	32	0.1597	0.9477	0.0899
c95ed0d5	10^{-4}	16	1	32	4	4	128	0.1597	0.9480	0.0908
63b490ee	10^{-4}	8	1	64	4	4	64	0.1597	0.9480	0.0905
2b33ad6d	10^{-5}	8	1	64	4	8	32	0.1597	0.9480	0.0907
7d8694d5	10^{-3}	16	4	32	4	8	128	0.1597	0.9438	0.0891
5cff3a7a	10^{-4}	16	4	32	4	8	32	0.1597	0.9480	0.0908
5e0bc2ea	10^{-4}	16	1	128	8	8	64	0.1597	0.9478	0.0913
e3f5bdd4	10^{-4}	16	2	64	4	8	32	0.1597	0.9480	0.0907
318d3658	10^{-5}	16	2	128	4	8	128	0.1597	0.9480	0.0906
4577141b	10^{-5}	8	4	32	8	4	128	0.1597	0.9480	0.0908
b3476807	10^{-5}	16	2	32	4	8	64	0.1597	0.9480	0.0903
815ead42	10^{-4}	16	1	32	8	8	64	0.1597	0.9480	0.0908
a680e0b1	10^{-3}	8	2	64	4	8	128	0.1596	0.9449	0.0898
6574a97f	10^{-5}	16	2	128	8	8	64	0.1596	0.9480	0.0900
4370b360	10^{-3}	8	1	128	8	8	32	0.1596	0.9480	0.0902
65044dac	10^{-5}	16	2	128	4	4	64	0.1596	0.9479	0.0908
a8b02e02	10^{-3}	8	1	32	4	8	128	0.1596	0.9195	0.0863
1f4443bf	10^{-5}	8	2	128	8	4	32	0.1596	0.9480	0.0900
2ce81fd8	10^{-5}	8	2	32	4	8	64	0.1596	0.9479	0.0898
e1b33bdd	10^{-5}	8	4	64	4	4	128	0.1596	0.9480	0.0903
0a234c6a	10^{-3}	8	4	32	8	8	64	0.1596	0.9469	0.0878
bc08b006	10^{-4}	8	1	32	8	8	64	0.1596	0.9480	0.0900
02ad20bc	10^{-3}	16	2	32	4	8	128	0.1596	0.9187	0.0899
9a9c940a	10^{-3}	8	2	32	8	4	32	0.1596	0.9469	0.0900
ac8faf59	10^{-5}	16	1	128	8	8	64	0.1596	0.9479	0.0901
32c1df93	10^{-3}	8	1	128	4	8	32	0.1596	0.9480	0.0888
aead3826	10^{-4}	8	1	64	8	8	32	0.1596	0.9478	0.0915
c4e03c7c	10^{-5}	8	4	128	8	8	32	0.1596	0.9480	0.0900
480f010c	10^{-3}	16	2	64	8	4	32	0.1596	0.9480	0.0889
b8085960	10^{-3}	16	1	64	8	8	128	0.1595	0.9446	0.0891
9dee483e	10^{-4}	16	2	32	8	8	64	0.1595	0.9472	0.0893
46836eb5	10^{-5}	16	1	32	8	4	32	0.1595	0.9480	0.0913
5a952ba3	10^{-5}	8	1	128	4	4	64	0.1595	0.9480	0.0904
44aa1612	10^{-5}	16	2	128	8	4	128	0.1595	0.9480	0.0906
94983f34	10^{-5}	8	2	64	8	8	128	0.1595	0.9479	0.0900
cda9db35	10^{-3}	8	1	64	8	8	128	0.1595	0.9321	0.0891

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
15ecbef1	10^{-5}	8	1	64	8	8	64	0.1595	0.9480	0.0907
8f2d9a06	10^{-5}	8	1	128	4	8	32	0.1595	0.9480	0.0907
3259a7dd	10^{-5}	16	2	32	4	4	128	0.1595	0.9480	0.0908
6bd0c278	10^{-5}	8	4	64	8	8	32	0.1595	0.9480	0.0903
78f72c48	10^{-5}	8	4	32	8	4	32	0.1595	0.9480	0.0892
acce2a54	10^{-5}	16	4	32	4	8	64	0.1595	0.9480	0.0903
adddf125	10^{-5}	8	4	128	8	8	128	0.1595	0.9480	0.0904
9e2f6c72	10^{-4}	16	2	64	8	8	32	0.1595	0.9480	0.0896
7f49f4d2	10^{-4}	8	2	128	8	4	64	0.1595	0.9479	0.0892
c808d967	10^{-3}	8	2	128	4	4	128	0.1594	0.9461	0.0879
69f7e817	10^{-4}	8	2	32	4	8	64	0.1594	0.9480	0.0912
b6e66672	10^{-4}	8	4	32	4	4	32	0.1594	0.9480	0.0890
6efae33f	10^{-5}	8	1	128	8	8	128	0.1594	0.9479	0.0900
e0748468	10^{-3}	8	4	32	8	8	64	0.1594	0.9228	0.0861
1ce80a94	10^{-5}	8	2	128	4	4	32	0.1594	0.9480	0.0903
8819ae7f	10^{-4}	16	2	32	8	4	128	0.1594	0.9480	0.0900
cb7b89df	10^{-5}	16	1	64	8	4	128	0.1594	0.9480	0.0901
101d85aa	10^{-4}	16	1	32	8	4	64	0.1594	0.9480	0.0909
a5fdeda0	10^{-3}	16	1	128	4	4	128	0.1594	0.9480	0.0902
6ac59703	10^{-5}	16	2	32	8	4	32	0.1594	0.9480	0.0902
0c10ff10	10^{-5}	8	2	128	8	4	64	0.1594	0.9480	0.0898
a6f4bdb5	10^{-3}	16	1	32	8	4	128	0.1594	0.9480	0.0891
e0b49bfb	10^{-4}	8	2	32	8	8	64	0.1593	0.9479	0.0908
0718c28a	10^{-4}	16	2	32	4	8	32	0.1593	0.9480	0.0907
15e3f9d3	10^{-4}	16	1	64	4	4	32	0.1593	0.9480	0.0904
f5cf3887	10^{-5}	8	2	32	4	4	128	0.1593	0.9479	0.0900
0a7f0e81	10^{-5}	16	2	128	8	8	128	0.1593	0.9480	0.0907
5b6d07ad	10^{-4}	8	2	64	4	8	32	0.1593	0.9480	0.0897
ccb9555d	10^{-5}	8	2	128	8	8	64	0.1593	0.9480	0.0905
e1668d5c	10^{-5}	8	1	128	8	8	128	0.1593	0.9480	0.0902
b43d4143	10^{-4}	8	2	32	8	4	32	0.1593	0.9480	0.0909
ded60297	10^{-4}	16	2	64	4	8	64	0.1593	0.9480	0.0902
4dd601c9	10^{-5}	16	4	64	4	4	128	0.1593	0.9480	0.0906
c47228e9	10^{-5}	8	2	64	8	8	32	0.1593	0.9479	0.0896
08809a82	10^{-5}	8	1	32	8	8	32	0.1593	0.9480	0.0899
7799720a	10^{-5}	8	4	64	4	4	64	0.1592	0.9480	0.0907
181d2b95	10^{-4}	16	4	64	4	8	32	0.1592	0.9480	0.0905
2dfed371	10^{-4}	16	1	128	8	4	128	0.1592	0.9464	0.0900
7fabd954	10^{-5}	8	1	32	4	4	64	0.1592	0.9480	0.0906

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
2697a88e	10^{-5}	16	2	128	4	4	128	0.1592	0.9480	0.0907
8e1805bf	10^{-4}	8	1	64	8	8	64	0.1592	0.9479	0.0909
063c7d85	10^{-5}	8	1	64	8	8	64	0.1592	0.9479	0.0897
68f81856	10^{-5}	8	2	32	4	8	32	0.1592	0.9480	0.0907
40a013af	10^{-4}	8	1	32	4	8	128	0.1592	0.9480	0.0903
8d7795ae	10^{-5}	16	4	32	4	4	64	0.1592	0.9480	0.0906
84166e4f	10^{-4}	16	1	64	8	4	32	0.1591	0.9480	0.0904
2b34240e	10^{-5}	8	4	64	8	4	64	0.1591	0.9480	0.0903
14df1e6a	10^{-5}	16	4	32	8	8	64	0.1591	0.9480	0.0908
704d11d0	10^{-3}	16	1	128	4	8	128	0.1591	0.9378	0.0873
b2a5acb6	10^{-4}	8	1	64	8	8	128	0.1591	0.9480	0.0910
2f23ed4a	10^{-4}	8	1	32	8	8	64	0.1591	0.9480	0.0905
8404cd3b	10^{-5}	16	1	64	8	8	64	0.1591	0.9480	0.0898
cc843511	10^{-5}	8	2	32	8	4	32	0.1590	0.9480	0.0906
35fd06c8	10^{-5}	8	1	64	4	4	128	0.1590	0.9480	0.0907
b40ce6df	10^{-5}	16	2	32	8	4	128	0.1590	0.9480	0.0907
6123f18e	10^{-5}	16	1	32	4	4	128	0.1590	0.9480	0.0907
caf05c66	10^{-5}	8	1	32	8	8	128	0.1590	0.9480	0.0907
b77ebae9	10^{-4}	16	2	64	4	8	64	0.1590	0.9480	0.0899
9a25c4f4	10^{-5}	16	2	128	8	8	64	0.1590	0.9480	0.0905
afaf3156	10^{-5}	8	2	128	8	8	128	0.1590	0.9480	0.0902
d3e8aea7	10^{-4}	8	1	128	4	8	128	0.1590	0.9480	0.0907
cda11b32	10^{-4}	8	2	32	8	8	64	0.1590	0.9480	0.0900
2f837060	10^{-4}	16	4	128	8	4	32	0.1590	0.9415	0.0900
297a0698	10^{-5}	16	4	128	8	4	32	0.1590	0.9480	0.0903
20ec4711	10^{-4}	8	2	128	8	4	32	0.1590	0.9456	0.0885
2046a0d9	10^{-5}	8	2	128	8	8	128	0.1590	0.9480	0.0895
553a2458	10^{-5}	8	2	128	8	4	32	0.1590	0.9480	0.0907
2c957f4d	10^{-5}	8	2	128	8	8	32	0.1590	0.9480	0.0897
f5c5a6b2	10^{-5}	16	1	32	4	4	128	0.1590	0.9480	0.0904
c4ae9291	10^{-3}	16	1	64	8	8	128	0.1590	0.9175	0.0898
2861e991	10^{-4}	16	4	32	4	4	32	0.1590	0.9480	0.0907
732e62cc	10^{-4}	8	1	32	8	8	128	0.1590	0.9469	0.0889
876183ad	10^{-5}	8	1	32	8	4	32	0.1590	0.9480	0.0905
d3c52546	10^{-4}	16	4	128	8	8	128	0.1589	0.9480	0.0909
d41ff12f	10^{-4}	8	1	32	8	4	32	0.1589	0.9480	0.0902
fc90b621	10^{-5}	16	2	64	8	8	64	0.1589	0.9480	0.0905
f2e95d9a	10^{-5}	16	1	64	8	4	64	0.1589	0.9480	0.0906
aeee7bc7	10^{-5}	16	4	32	4	8	128	0.1589	0.9480	0.0907

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
c092cf30	10^{-5}	8	4	128	4	4	32	0.1589	0.9480	0.0905
b91f0d03	10^{-4}	16	4	128	8	8	32	0.1589	0.9475	0.0905
9538ff5e	10^{-4}	16	1	64	8	8	64	0.1589	0.9480	0.0902
88feac14	10^{-4}	8	2	32	4	8	32	0.1589	0.9480	0.0907
9420043f	10^{-4}	16	1	64	4	8	32	0.1589	0.9479	0.0906
1a4a8c4d	10^{-4}	8	1	32	4	4	128	0.1589	0.9480	0.0907
837887fe	10^{-4}	16	4	32	8	8	32	0.1589	0.9480	0.0908
7e1154b8	10^{-3}	16	2	32	4	8	64	0.1589	0.9470	0.0894
ccf5e849	10^{-5}	8	4	32	4	4	128	0.1589	0.9480	0.0903
b67fae70	10^{-4}	16	1	64	8	4	128	0.1589	0.9480	0.0903
47a43fab	10^{-5}	8	1	32	8	8	64	0.1589	0.9480	0.0901
82e5d174	10^{-5}	8	2	64	4	8	128	0.1589	0.9480	0.0903
18b072e7	10^{-4}	8	2	128	4	8	64	0.1589	0.9480	0.0912
8d847466	10^{-5}	8	1	128	4	4	128	0.1589	0.9480	0.0905
833e8e6a	10^{-5}	8	2	128	8	8	64	0.1588	0.9480	0.0906
e31f02a5	10^{-5}	8	4	64	8	4	32	0.1588	0.9480	0.0903
0cc7abf5	10^{-4}	16	4	64	8	4	128	0.1588	0.9432	0.0888
97467b8d	10^{-5}	8	2	64	8	4	128	0.1588	0.9480	0.0905
5ab594c7	10^{-3}	8	4	64	8	4	32	0.1588	0.9456	0.0916
a4e0446d	10^{-5}	16	1	32	4	4	32	0.1588	0.9480	0.0898
7139add0	10^{-4}	16	4	64	8	8	32	0.1588	0.9480	0.0900
b6756ae5	10^{-4}	8	4	64	4	4	32	0.1588	0.9480	0.0903
4237b8e2	10^{-3}	16	2	128	4	4	128	0.1588	0.9442	0.0909
b18c5879	10^{-5}	8	4	64	4	8	128	0.1587	0.9480	0.0905
2641dd3f	10^{-4}	8	1	64	4	4	32	0.1587	0.9480	0.0909
1eed4a91	10^{-5}	8	1	128	4	8	64	0.1587	0.9480	0.0901
74f123c3	10^{-5}	16	2	128	8	8	32	0.1587	0.9480	0.0907
800fc5ac	10^{-5}	8	4	32	4	8	64	0.1587	0.9480	0.0907
ee6c3880	10^{-4}	16	4	32	8	4	32	0.1587	0.9480	0.0908
cff35a01	10^{-5}	16	1	128	8	8	64	0.1587	0.9480	0.0903
5ffdd4e9	10^{-3}	16	1	64	8	4	32	0.1587	0.9371	0.0900
db3090d5	10^{-5}	16	1	128	4	4	32	0.1587	0.9480	0.0904
f2f44127	10^{-5}	16	1	128	4	4	128	0.1587	0.9480	0.0905
9ef5e80e	10^{-5}	8	4	32	8	8	32	0.1587	0.9480	0.0897
c609cc6f	10^{-5}	8	1	128	4	4	32	0.1587	0.9480	0.0907
b2698aeb	10^{-4}	8	4	32	4	4	64	0.1587	0.9480	0.0898
f00da8a8	10^{-4}	8	1	64	4	8	128	0.1587	0.9480	0.0906
a552141d	10^{-5}	8	4	128	8	4	64	0.1587	0.9480	0.0905
b7439d73	10^{-4}	16	2	32	4	4	64	0.1586	0.9480	0.0906

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
642bd912	10^{-5}	8	4	64	8	8	128	0.1586	0.9480	0.0904
2df75a02	10^{-5}	16	2	64	8	4	32	0.1586	0.9480	0.0908
cf1a4ef0	10^{-5}	8	1	128	8	4	128	0.1586	0.9480	0.0907
4a79cace	10^{-4}	8	4	32	4	8	64	0.1586	0.9479	0.0903
b90c705a	10^{-5}	16	1	64	8	8	32	0.1586	0.9480	0.0900
91a28831	10^{-5}	8	2	32	4	4	32	0.1586	0.9478	0.0893
e865793f	10^{-4}	8	2	64	8	8	32	0.1586	0.9480	0.0896
eecb14fd	10^{-4}	8	4	32	8	4	128	0.1586	0.9466	0.0893
bdf12c72	10^{-4}	8	2	32	8	4	128	0.1586	0.9475	0.0900
933fb7db	10^{-3}	16	2	32	4	8	32	0.1586	0.9286	0.0908
dd264f67	10^{-3}	8	1	128	8	4	32	0.1586	0.9480	0.0905
1245d42c	10^{-5}	8	1	64	8	4	128	0.1586	0.9480	0.0903
bf16598d	10^{-3}	16	1	128	4	8	128	0.1585	0.9280	0.0874
7fbe17b9	10^{-4}	16	1	64	8	8	128	0.1585	0.9480	0.0906
e438471b	10^{-5}	16	1	128	4	8	32	0.1585	0.9480	0.0903
8a87cd2e	10^{-5}	8	2	64	4	4	128	0.1585	0.9480	0.0896
8c731f39	10^{-5}	8	2	32	8	8	64	0.1585	0.9480	0.0905
80bce819	10^{-3}	16	4	64	4	4	32	0.1585	0.9435	0.0883
b41c26f4	10^{-5}	8	1	128	8	4	32	0.1585	0.9480	0.0903
59c7fdc9	10^{-5}	8	2	64	4	4	32	0.1585	0.9480	0.0901
d68f60fa	10^{-4}	16	1	32	8	8	128	0.1585	0.9473	0.0891
af55f1b9	10^{-5}	16	2	128	4	8	32	0.1585	0.9479	0.0909
82720fbb	10^{-3}	8	2	128	4	4	64	0.1585	0.9425	0.0900
b3f519e1	10^{-4}	8	1	32	8	8	32	0.1585	0.9480	0.0903
65429f05	10^{-3}	8	2	128	4	8	32	0.1584	0.9172	0.0887
c76046ba	10^{-5}	16	1	32	8	8	64	0.1584	0.9480	0.0889
db196950	10^{-5}	16	1	64	4	8	128	0.1584	0.9480	0.0906
085ec149	10^{-5}	16	4	128	4	8	128	0.1584	0.9480	0.0906
5dbb69e6	10^{-4}	16	4	32	4	8	64	0.1584	0.9480	0.0905
2628751	10^{-5}	8	–	32	8	4	64	0.1584	0.9475	0.0896
5b2870ff	10^{-5}	8	1	64	8	8	128	0.1584	0.9479	0.0914
c3539f75	10^{-5}	8	1	128	8	4	64	0.1583	0.9480	0.0901
5602f6c2	10^{-4}	16	2	32	8	8	128	0.1583	0.9480	0.0904
b244dca1	10^{-5}	16	2	64	4	8	128	0.1583	0.9480	0.0900
20bce9d2	10^{-3}	8	1	32	4	8	64	0.1583	0.9480	0.0913
b50c3efd	10^{-5}	16	1	64	4	8	32	0.1583	0.9480	0.0901
e3957a97	10^{-4}	8	4	32	4	8	64	0.1583	0.9480	0.0902
b4ce6c44	10^{-5}	8	4	32	8	8	128	0.1583	0.9480	0.0905
a4182baf	10^{-4}	16	2	64	4	4	128	0.1583	0.9480	0.0908

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
d89c8cb8	10^{-5}	16	2	64	8	4	32	0.1583	0.9480	0.0906
c2f20083	10^{-4}	8	1	64	8	4	64	0.1583	0.9480	0.0893
73ea9040	10^{-4}	16	1	32	4	8	32	0.1583	0.9480	0.0898
9b6c4cf6	10^{-3}	16	4	64	4	8	32	0.1583	0.9439	0.0891
4d43f9b6	10^{-5}	8	1	32	4	8	32	0.1582	0.9480	0.0903
57f1fd05	10^{-5}	16	4	128	4	8	64	0.1582	0.9480	0.0901
b2b8a414	10^{-5}	16	1	64	8	4	32	0.1582	0.9480	0.0901
7b6921b8	10^{-5}	8	1	32	4	8	64	0.1582	0.9480	0.0902
6602ed7d	10^{-4}	8	2	32	4	8	32	0.1582	0.9480	0.0903
0b7649f0	10^{-3}	16	2	64	4	4	64	0.1582	0.9299	0.0893
88af1866	10^{-4}	8	2	64	8	8	128	0.1582	0.9480	0.0899
835cc7c9	10^{-5}	16	4	32	8	4	128	0.1582	0.9480	0.0907
92645da9	10^{-5}	16	4	128	4	4	64	0.1581	0.9480	0.0900
7c2f3c13	10^{-5}	16	2	128	8	8	128	0.1581	0.9480	0.0898
cd1d8d01	10^{-5}	16	2	64	4	8	64	0.1581	0.9480	0.0899
14ef4925	10^{-5}	8	2	128	4	8	32	0.1581	0.9480	0.0900
14c45ab0	10^{-4}	8	2	128	8	4	64	0.1581	0.9480	0.0902
6046f6a0	10^{-4}	16	2	32	4	4	128	0.1581	0.9478	0.0903
98e8d279	10^{-3}	16	1	128	4	8	32	0.1581	0.9480	0.0906
486251ec	10^{-5}	16	2	128	4	8	32	0.1581	0.9480	0.0902
4bd29d88	10^{-5}	16	1	128	8	8	128	0.1581	0.9480	0.0902
b59d8fc3	10^{-5}	8	1	128	4	8	128	0.1580	0.9479	0.0903
f1d982c8	10^{-5}	8	4	64	8	4	128	0.1580	0.9480	0.0906
e88f489a	10^{-3}	8	1	128	8	4	64	0.1580	0.9303	0.0891
ef489ba3	10^{-4}	8	1	32	8	8	128	0.1580	0.9480	0.0894
49399f48	10^{-3}	16	1	32	4	8	64	0.1580	0.9303	0.0877
122108d2	10^{-3}	16	1	128	8	8	64	0.1580	0.9480	0.0905
7e8308ac	10^{-5}	8	1	128	8	4	64	0.1579	0.9480	0.0906
207ee875	10^{-5}	8	2	64	8	4	64	0.1579	0.9480	0.0904
c591cce8	10^{-5}	8	2	128	4	4	64	0.1579	0.9480	0.0904
b0b4748d	10^{-3}	16	4	32	8	4	32	0.1579	0.9480	0.0895
7b7897ba	10^{-5}	8	4	32	8	8	64	0.1579	0.9480	0.0900
1d1e5d6c	10^{-3}	8	1	128	4	8	64	0.1579	0.9480	0.0900
49b30016	10^{-5}	16	1	64	8	8	64	0.1579	0.9480	0.0903
d38f6a00	10^{-5}	16	2	32	8	8	128	0.1579	0.9480	0.0903
b9790cec	10^{-5}	8	4	32	8	8	64	0.1579	0.9480	0.0904
3e0ac294	10^{-5}	16	1	64	8	4	32	0.1579	0.9480	0.0901
60410047	10^{-4}	16	1	64	8	8	32	0.1579	0.9480	0.0902
3982637c	10^{-5}	8	4	64	4	8	64	0.1579	0.9480	0.0899

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
22916253	10^{-3}	16	4	64	8	4	64	0.1579	0.9402	0.0893
f5b94c6b	10^{-3}	8	2	128	8	8	32	0.1579	0.9480	0.0904
1c8a4a80	10^{-5}	8	2	128	4	8	32	0.1579	0.9480	0.0901
bf47708c	10^{-5}	16	1	32	4	8	64	0.1579	0.9479	0.0904
00930a75	10^{-5}	16	4	64	8	8	64	0.1579	0.9480	0.0911
edcd77cb	10^{-5}	8	1	64	4	4	32	0.1578	0.9480	0.0891
9c982253	10^{-4}	16	2	32	8	4	64	0.1578	0.9480	0.0904
d5625ab7	10^{-5}	8	1	32	8	4	64	0.1578	0.9480	0.0906
70995866	10^{-5}	8	1	64	4	4	64	0.1578	0.9480	0.0907
94cc2fa0	10^{-3}	8	1	32	4	4	128	0.1578	0.9273	0.0873
8bad21e7	10^{-5}	8	2	32	4	8	32	0.1578	0.9480	0.0906
d3ebe383	10^{-5}	8	1	32	4	4	64	0.1578	0.9480	0.0907
61d3d2b5	10^{-5}	16	1	32	4	8	64	0.1578	0.9480	0.0899
c81e86db	10^{-5}	8	1	64	4	4	32	0.1578	0.9480	0.0902
338e7738	10^{-5}	16	1	64	4	4	32	0.1578	0.9480	0.0900
c25b4ba4	10^{-5}	8	2	64	4	4	128	0.1578	0.9480	0.0899
dcd13dd1	10^{-5}	16	2	64	8	4	128	0.1578	0.9480	0.0901
70ace2cb	10^{-5}	8	2	128	4	4	128	0.1578	0.9480	0.0898
8af38019	10^{-5}	16	1	32	4	4	64	0.1578	0.9480	0.0905
e2a40854	10^{-4}	16	2	32	4	4	64	0.1577	0.9479	0.0897
37407980	10^{-4}	16	1	64	4	8	64	0.1577	0.9480	0.0900
8f036dab	10^{-4}	16	2	128	8	8	128	0.1577	0.9480	0.0896
dcb34c80	10^{-4}	8	4	64	4	4	64	0.1577	0.9480	0.0906
280abbb6	10^{-4}	8	1	32	8	4	64	0.1577	0.9480	0.0898
c7675d13	10^{-4}	8	4	128	8	4	128	0.1577	0.9480	0.0901
e45fdd81	10^{-3}	16	1	64	4	8	64	0.1577	0.9301	0.0889
17863937	10^{-5}	8	1	64	8	4	64	0.1577	0.9480	0.0898
df081065	10^{-4}	8	1	32	4	4	32	0.1577	0.9480	0.0902
d42d1cb9	10^{-3}	16	1	32	4	8	32	0.1576	0.9480	0.0903
90c595ee	10^{-4}	16	2	64	8	4	64	0.1576	0.9480	0.0900
06bc45e7	10^{-5}	8	2	64	8	4	128	0.1576	0.9480	0.0904
da372aa1	10^{-5}	8	4	32	4	4	32	0.1576	0.9480	0.0904
5771c42f	10^{-5}	16	1	64	8	8	128	0.1576	0.9480	0.0903
770914bd	10^{-4}	16	1	64	4	4	128	0.1576	0.9479	0.0902
945a28cc	10^{-4}	16	1	32	4	8	32	0.1576	0.9479	0.0903
c2208273	10^{-5}	16	4	128	8	4	64	0.1576	0.9480	0.0905
3d775e1b	10^{-5}	16	2	128	4	8	64	0.1576	0.9480	0.0905
d0776d3f	10^{-5}	8	2	128	8	4	128	0.1576	0.9480	0.0898
0c66f82a	10^{-5}	8	1	64	8	4	128	0.1576	0.9480	0.0906

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
0978f514	10^{-5}	8	2	128	4	8	128	0.1576	0.9480	0.0903
e7fd9c5e	10^{-4}	16	2	64	8	8	64	0.1575	0.9480	0.0903
79fddd60	10^{-4}	8	2	32	8	8	32	0.1575	0.9480	0.0898
89daae3c	10^{-5}	16	4	32	8	8	32	0.1575	0.9479	0.0898
0b3855e4	10^{-5}	8	2	32	4	4	32	0.1575	0.9480	0.0906
e3a4a315	10^{-4}	16	1	32	4	4	64	0.1575	0.9480	0.0901
a4abd86d	10^{-4}	8	4	32	8	8	64	0.1575	0.9464	0.0888
f26d76a0	10^{-5}	8	1	32	4	8	128	0.1575	0.9480	0.0899
9d403b67	10^{-5}	16	2	64	8	4	64	0.1575	0.9480	0.0896
6ee8e52b	10^{-3}	16	2	64	8	8	64	0.1575	0.9392	0.0892
b5d956b4	10^{-5}	16	2	128	4	4	32	0.1575	0.9479	0.0898
aacc1bdb	10^{-3}	8	4	64	4	8	128	0.1575	0.9477	0.0897
00b33fbf	10^{-5}	8	2	64	4	8	64	0.1575	0.9480	0.0900
1e4daf31	10^{-3}	16	2	64	4	4	128	0.1575	0.9450	0.0903
6b4c98f3	10^{-3}	8	1	64	8	4	128	0.1575	0.9480	0.0899
0c24bdff	10^{-5}	16	4	32	8	8	32	0.1574	0.9480	0.0895
68e9964f	10^{-5}	8	2	32	4	4	64	0.1574	0.9480	0.0906
c8048f1c	10^{-3}	8	4	128	8	4	64	0.1574	0.9478	0.0897
2e77ded4	10^{-4}	16	2	128	4	8	64	0.1574	0.9480	0.0905
1dba40e6	10^{-5}	8	1	64	8	4	32	0.1573	0.9480	0.0901
6cf0c721	10^{-4}	8	4	128	8	8	32	0.1573	0.9455	0.0893
f42208b4	10^{-3}	8	4	64	8	8	32	0.1573	0.9478	0.0879
7e0fc2b2	10^{-3}	8	4	32	4	8	128	0.1573	0.9446	0.0883
9816dec4	10^{-4}	8	1	32	4	8	32	0.1573	0.9479	0.0907
662591b1	10^{-5}	8	4	128	4	8	128	0.1573	0.9480	0.0907
7b5ab801	10^{-5}	16	1	128	8	4	128	0.1573	0.9480	0.0899
3acbe366	10^{-4}	16	1	64	8	4	32	0.1573	0.9478	0.0912
dd8de9be	10^{-4}	16	4	32	8	4	128	0.1573	0.9480	0.0895
f42526a5	10^{-3}	16	2	128	4	8	64	0.1573	0.9429	0.0885
73f518de	10^{-4}	8	1	128	4	4	32	0.1573	0.9480	0.0901
86c38d6a	10^{-5}	8	1	32	8	8	32	0.1573	0.9480	0.0902
e21a9392	10^{-5}	16	4	32	4	8	64	0.1573	0.9480	0.0908
fc6bb673	10^{-4}	8	2	64	8	8	32	0.1573	0.9480	0.0904
ec63c8fc	10^{-4}	16	1	64	8	4	64	0.1573	0.9480	0.0904
dfe934f9	10^{-5}	8	2	128	4	8	64	0.1573	0.9480	0.0901
2435e9b2	10^{-4}	16	2	32	4	4	32	0.1572	0.9480	0.0893
43b70a00	10^{-4}	16	1	128	4	8	32	0.1572	0.9480	0.0902
e7830d11	10^{-4}	16	2	64	4	4	64	0.1572	0.9480	0.0897
8587efa1	10^{-4}	16	1	32	8	8	32	0.1572	0.9480	0.0899

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
9655cfc8	10^{-5}	16	2	32	4	4	128	0.1572	0.9480	0.0905
25547c42	10^{-5}	16	1	32	8	4	64	0.1572	0.9480	0.0900
22ca8884	10^{-5}	16	1	32	8	8	64	0.1572	0.9480	0.0900
de0f04da	10^{-3}	16	2	32	8	4	32	0.1571	0.9480	0.0890
2e749e8f	10^{-5}	16	1	128	8	4	32	0.1571	0.9480	0.0898
ad4a644f	10^{-3}	8	2	128	4	8	64	0.1571	0.9469	0.0899
7880c661	10^{-4}	8	2	64	8	8	64	0.1571	0.9478	0.0898
38059154	10^{-5}	16	1	128	4	4	128	0.1571	0.9480	0.0900
41d1a0f7	10^{-5}	16	2	128	4	4	64	0.1571	0.9480	0.0899
7a4a026d	10^{-5}	8	2	32	8	8	64	0.1571	0.9480	0.0887
8312e6bb	10^{-3}	8	1	32	8	8	64	0.1571	0.9480	0.0900
a1b5a314	10^{-4}	8	2	32	4	4	64	0.1570	0.9480	0.0894
ae3f6f5e	10^{-4}	16	2	32	8	8	32	0.1570	0.9478	0.0897
73982098	10^{-4}	16	4	32	8	4	32	0.1570	0.9479	0.0902
121d7a34	10^{-3}	16	1	64	8	8	32	0.1570	0.9374	0.0888
5ec35c85	10^{-4}	16	1	128	4	4	128	0.1570	0.9480	0.0910
ef51435a	10^{-5}	16	2	64	4	8	128	0.1570	0.9480	0.0898
233aca6a	10^{-5}	8	2	32	4	8	128	0.1569	0.9480	0.0898
97da7386	10^{-4}	16	4	64	8	4	64	0.1569	0.9480	0.0902
4e209bc6	10^{-4}	8	4	32	4	4	32	0.1569	0.9480	0.0900
32fb48c1	10^{-4}	16	2	128	4	4	64	0.1569	0.9480	0.0900
65e1ac5d	10^{-4}	8	1	32	8	8	32	0.1569	0.9480	0.0897
029e8e01	10^{-5}	16	4	64	8	4	32	0.1569	0.9480	0.0899
aa61154a	10^{-4}	8	1	64	8	8	128	0.1569	0.9436	0.0892
6908b97f	10^{-3}	8	4	32	8	4	32	0.1569	0.9480	0.0895
09e3c696	10^{-5}	16	2	128	8	4	32	0.1569	0.9480	0.0907
d3eb2e20	10^{-4}	8	2	32	8	4	32	0.1569	0.9480	0.0899
41611435	10^{-4}	16	4	64	4	4	32	0.1569	0.9480	0.0898
9dbc4213	10^{-3}	8	1	128	4	4	64	0.1569	0.9480	0.0899
2b5d6c35	10^{-3}	8	1	128	4	8	128	0.1569	0.9194	0.0857
4dd90f0b	10^{-3}	8	1	128	4	8	32	0.1569	0.9388	0.0880
80f570ff	10^{-3}	8	2	128	4	4	64	0.1569	0.9247	0.0876
16e2514c	10^{-3}	16	2	32	8	4	32	0.1568	0.9286	0.0902
4799cc7c	10^{-3}	8	1	64	8	4	32	0.1568	0.9480	0.0893
38f73871	10^{-4}	16	2	128	4	8	32	0.1568	0.9480	0.0904
50681823	10^{-5}	8	2	64	4	8	128	0.1568	0.9479	0.0893
e18fda2c	10^{-4}	8	1	128	4	8	64	0.1568	0.9472	0.0910
3.4163E+071	10^{-3}	8	–	128	4	4	128	0.1568	0.9412	0.0863
739caa1d	10^{-5}	8	4	32	4	4	64	0.1568	0.9480	0.0902

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
6d1ad7aa	10^{-5}	16	4	64	4	8	32	0.1568	0.9480	0.0897
ad2bf98c	10^{-4}	16	1	128	4	4	64	0.1568	0.9480	0.0898
0b439b5d	10^{-5}	8	2	32	8	8	32	0.1568	0.9480	0.0903
64db013f	10^{-4}	16	1	32	8	4	128	0.1568	0.9480	0.0901
9731b2b4	10^{-5}	8	1	128	4	8	128	0.1568	0.9480	0.0900
93e9a378	10^{-5}	16	4	128	8	8	128	0.1568	0.9480	0.0902
cfe06263	10^{-5}	8	1	32	4	8	32	0.1568	0.9477	0.0900
a8d36129	10^{-5}	8	1	128	4	8	32	0.1568	0.9480	0.0905
94a67a14	10^{-4}	16	1	32	4	4	128	0.1567	0.9480	0.0900
30034d43	10^{-4}	8	2	32	4	8	128	0.1567	0.9480	0.0904
cd93dec1	10^{-5}	16	4	64	8	8	128	0.1567	0.9480	0.0905
61cec361	10^{-3}	8	2	64	8	4	64	0.1567	0.9343	0.0906
a4b7b189	10^{-3}	8	2	64	8	4	64	0.1567	0.9476	0.0889
6d505625	10^{-3}	16	4	64	8	4	128	0.1567	0.9476	0.0930
9266b9b8	10^{-4}	16	1	64	8	8	128	0.1567	0.9479	0.0904
864e8d7f	10^{-4}	8	2	128	4	4	128	0.1567	0.9476	0.0898
d9fa13bb	10^{-4}	16	1	32	8	8	128	0.1567	0.9480	0.0899
5ef43beb	10^{-5}	8	1	32	4	4	32	0.1567	0.9480	0.0913
c1c27fb6	10^{-4}	8	2	64	8	8	64	0.1566	0.9480	0.0901
b0be3d82	10^{-5}	16	1	64	8	4	128	0.1566	0.9480	0.0896
08296bc8	10^{-5}	16	4	32	8	8	128	0.1566	0.9480	0.0900
51cfa8e2	10^{-3}	8	4	32	4	4	128	0.1566	0.9224	0.0886
3b083685	10^{-4}	8	2	64	4	4	128	0.1566	0.9480	0.0892
2a729280	10^{-3}	8	4	32	8	4	64	0.1566	0.9225	0.0873
ad1d0f9c	10^{-5}	16	4	128	8	8	32	0.1566	0.9480	0.0901
f12a882c	10^{-3}	16	2	32	4	4	64	0.1566	0.9359	0.0881
d07deb2d	10^{-4}	16	4	64	4	4	128	0.1566	0.9480	0.0900
032adfa9	10^{-3}	16	1	64	8	4	64	0.1566	0.9480	0.0901
01f4e16a	10^{-4}	8	1	32	4	8	64	0.1566	0.9480	0.0903
9765024a	10^{-4}	8	4	128	8	4	64	0.1566	0.9388	0.0906
0b864451	10^{-5}	16	2	32	8	4	64	0.1566	0.9480	0.0904
c56c2946	10^{-5}	16	2	32	8	8	64	0.1566	0.9480	0.0898
aae4a0b6	10^{-4}	16	2	32	4	8	128	0.1565	0.9480	0.0901
c32a464f	10^{-3}	16	2	128	8	4	128	0.1565	0.9297	0.0874
c9a28468	10^{-4}	16	1	64	4	4	64	0.1565	0.9480	0.0891
efe839cf	10^{-4}	8	2	32	4	4	128	0.1565	0.9480	0.0904
1542afe4	10^{-5}	16	1	128	4	8	64	0.1565	0.9480	0.0903
e306b785	10^{-5}	16	2	64	8	4	64	0.1564	0.9480	0.0904
fa8aaf58	10^{-3}	8	4	32	4	4	128	0.1564	0.9434	0.0888

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
30f63c09	10^{-4}	8	1	64	8	8	64	0.1564	0.9480	0.0903
36eba93f	10^{-3}	8	1	128	8	8	128	0.1564	0.9480	0.0887
08fa144d	10^{-5}	16	1	128	8	4	32	0.1564	0.9479	0.0904
6534b336	10^{-4}	8	4	32	8	8	128	0.1564	0.9470	0.0885
f006e747	10^{-4}	8	2	128	8	4	128	0.1564	0.9480	0.0885
5168f3db	10^{-4}	8	2	128	8	8	32	0.1563	0.9479	0.0891
2f0113d7	10^{-3}	8	4	128	8	8	32	0.1563	0.9472	0.0878
4f1bb94f	10^{-5}	16	4	64	4	8	64	0.1563	0.9480	0.0904
d7a73a35	10^{-4}	16	1	128	8	8	32	0.1563	0.9479	0.0890
5bca393a	10^{-3}	8	1	32	8	4	128	0.1563	0.9286	0.0868
ba57e240	10^{-3}	16	1	128	8	4	64	0.1563	0.9480	0.0899
17b71667	10^{-3}	8	2	64	8	8	64	0.1563	0.9480	0.0908
9ebdb2d1	10^{-5}	8	2	32	8	8	32	0.1563	0.9480	0.0889
0cbc8359	10^{-3}	8	2	128	4	8	32	0.1563	0.9473	0.0910
349e779e	10^{-5}	8	1	128	8	8	32	0.1563	0.9480	0.0897
8e0f812f	10^{-5}	8	1	32	8	8	128	0.1563	0.9479	0.0900
4866754e	10^{-5}	16	2	64	4	4	32	0.1562	0.9480	0.0902
3aa14a05	10^{-4}	16	2	32	8	4	64	0.1562	0.9480	0.0900
d64d7168	10^{-3}	16	1	64	4	4	64	0.1562	0.9459	0.0902
16b6264f	10^{-4}	16	2	32	4	8	64	0.1562	0.9480	0.0900
246e297c	10^{-4}	16	2	32	4	8	128	0.1562	0.9480	0.0901
11e8acf9	10^{-4}	8	1	64	4	4	64	0.1561	0.9480	0.0887
d273597b	10^{-4}	16	2	128	8	4	64	0.1561	0.9479	0.0897
3b8fb591	10^{-3}	16	4	32	4	4	64	0.1561	0.9341	0.0879
cd4fce13	10^{-4}	8	1	128	8	4	32	0.1561	0.9480	0.0900
58544402	10^{-4}	16	2	128	8	4	128	0.1561	0.9480	0.0895
6d3b49ce	10^{-4}	8	4	32	8	4	32	0.1561	0.9479	0.0895
50344638	10^{-3}	8	1	64	8	4	64	0.1560	0.9480	0.0894
85256a8b	10^{-5}	8	4	32	4	8	32	0.1560	0.9480	0.0901
79b4e8ee	10^{-4}	16	2	64	8	4	64	0.1560	0.9480	0.0895
eb9f6139	10^{-4}	8	4	64	4	8	32	0.1560	0.9480	0.0913
8e0a5f16	10^{-4}	8	2	64	8	4	32	0.1560	0.9480	0.0904
dfe068dc	10^{-5}	16	2	64	4	4	128	0.1560	0.9480	0.0901
0719368e	10^{-3}	16	4	128	8	8	32	0.1560	0.9480	0.0909
7e15026f	10^{-5}	8	4	128	8	4	128	0.1560	0.9480	0.0901
c450f99d	10^{-4}	16	1	128	4	4	32	0.1560	0.9478	0.0898
2f21b86b	10^{-4}	8	1	128	8	4	64	0.1560	0.9480	0.0899
3893d3c1	10^{-3}	16	1	128	8	4	32	0.1560	0.9480	0.0891
2257570b	10^{-5}	8	2	128	4	8	128	0.1559	0.9480	0.0898

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
c044063d	10^{-4}	16	2	32	4	8	64	0.1559	0.9480	0.0900
888ffa1e	10^{-5}	16	2	64	4	8	32	0.1559	0.9480	0.0894
e20e8e03	10^{-3}	16	2	64	4	8	64	0.1559	0.9234	0.0891
f4c5eb77	10^{-5}	16	4	32	8	4	32	0.1559	0.9480	0.0899
6a353fd8	10^{-5}	16	1	128	8	4	64	0.1559	0.9480	0.0900
17509bf2	10^{-4}	16	4	32	4	4	32	0.1559	0.9480	0.0898
6c284019	10^{-4}	8	2	128	4	8	128	0.1559	0.9480	0.0902
409bac93	10^{-5}	8	4	32	8	4	128	0.1558	0.9480	0.0906
9cf1be22	10^{-3}	16	2	128	4	8	128	0.1558	0.9222	0.0867
78572cd6	10^{-5}	16	2	64	8	8	32	0.1558	0.9480	0.0903
3b80a056	10^{-3}	16	1	32	4	4	64	0.1558	0.9480	0.0893
ff7ae91c	10^{-3}	8	2	64	4	8	128	0.1558	0.9261	0.0872
dfd85ee3	10^{-5}	16	4	32	4	8	32	0.1558	0.9480	0.0892
b6269bf8	10^{-4}	8	4	32	4	4	128	0.1558	0.9480	0.0899
8e18a019	10^{-4}	16	1	64	4	4	32	0.1557	0.9480	0.0893
ad7de2af	10^{-4}	16	4	64	4	8	128	0.1557	0.9476	0.0898
5b2293d0	10^{-4}	16	1	128	8	4	32	0.1557	0.9480	0.0891
c2c152b6	10^{-5}	8	4	32	4	8	64	0.1557	0.9479	0.0892
71dded28	10^{-5}	16	1	128	8	8	32	0.1557	0.9480	0.0902
23a729c0	10^{-4}	8	2	64	4	4	64	0.1557	0.9480	0.0887
50119581	10^{-5}	16	2	64	4	8	64	0.1557	0.9480	0.0898
ed3f6449	10^{-3}	16	1	128	8	8	64	0.1556	0.9298	0.0848
89d1c4f3	10^{-3}	16	1	128	4	8	32	0.1556	0.9384	0.0889
fde34897	10^{-3}	8	2	128	8	4	64	0.1556	0.9267	0.0884
6d9d5cfa	10^{-4}	16	2	128	4	4	32	0.1555	0.9480	0.0895
fe426e10	10^{-4}	8	4	128	4	8	64	0.1555	0.9475	0.0868
74706011	10^{-5}	8	4	32	8	4	64	0.1555	0.9480	0.0895
182dd4fa	10^{-5}	8	2	128	4	4	32	0.1555	0.9480	0.0900
632a5b8c	10^{-4}	8	4	32	8	8	64	0.1555	0.9480	0.0900
30b7195b	10^{-5}	16	4	32	4	4	32	0.1555	0.9480	0.0897
67da0ca7	10^{-3}	8	1	32	8	4	32	0.1555	0.9480	0.0901
4e20e4d1	10^{-3}	8	2	32	4	4	32	0.1554	0.9475	0.0901
49e277e5	10^{-3}	16	2	128	8	8	64	0.1554	0.9282	0.0887
adaf84a9	10^{-3}	8	1	32	4	4	32	0.1554	0.9273	0.0867
4ca43fab	10^{-4}	16	4	32	4	4	128	0.1554	0.9480	0.0899
c6d2ba2a	10^{-4}	8	2	64	8	4	128	0.1554	0.9480	0.0895
ce174709	10^{-4}	16	2	64	8	8	64	0.1554	0.9480	0.0896
66ba3e15	10^{-3}	8	1	64	8	8	32	0.1554	0.9328	0.0850
019a6a28	10^{-4}	16	4	32	4	4	64	0.1554	0.9479	0.0899

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
06ef3d24	10^{-3}	8	4	32	8	4	64	0.1554	0.9469	0.0887
3ea7735e	10^{-4}	8	4	128	4	4	64	0.1553	0.9455	0.0909
c0985976	10^{-5}	16	4	32	8	4	128	0.1553	0.9480	0.0896
cabaa504	10^{-5}	16	1	32	4	8	128	0.1553	0.9480	0.0900
b9c54daf	10^{-4}	8	1	128	8	8	32	0.1553	0.9480	0.0903
0e3713b3	10^{-5}	8	1	64	4	4	128	0.1553	0.9480	0.0904
ce75131c	10^{-4}	16	2	64	8	4	32	0.1553	0.9478	0.0889
7009327c	10^{-5}	8	2	64	8	4	32	0.1553	0.9480	0.0901
2d4cb1c7	10^{-3}	8	1	64	4	4	64	0.1553	0.9422	0.0872
a71aca57	10^{-4}	8	1	64	4	8	32	0.1553	0.9479	0.0900
ee0eb36e	10^{-3}	16	2	64	8	4	64	0.1552	0.9480	0.0891
e31e1d4a	10^{-4}	16	2	32	8	8	128	0.1552	0.9479	0.0895
b8171147	10^{-3}	16	2	128	4	4	64	0.1552	0.9463	0.0876
6f61a3ba	10^{-4}	16	2	128	4	8	128	0.1552	0.9461	0.0895
4d8daf67	10^{-4}	16	2	128	4	8	128	0.1551	0.9480	0.0900
2eb58082	10^{-4}	8	2	128	8	8	32	0.1551	0.9470	0.0887
2cf0c1da	10^{-3}	8	2	64	8	4	32	0.1551	0.9403	0.0876
e4cb6cbd	10^{-4}	8	2	32	4	8	128	0.1551	0.9480	0.0899
5dd194ae	10^{-4}	16	1	32	8	4	128	0.1551	0.9478	0.0903
b4a248aa	10^{-4}	8	1	128	4	8	32	0.1551	0.9480	0.0889
84feae66	10^{-3}	8	2	32	8	8	64	0.1551	0.9470	0.0899
ef8880fe	10^{-4}	8	2	128	8	8	64	0.1551	0.9477	0.0897
3aef7b24	10^{-4}	8	1	128	4	8	64	0.1550	0.9480	0.0892
c296e1f4	10^{-4}	16	4	32	8	4	128	0.1550	0.9479	0.0890
4b86e210	10^{-4}	8	1	64	4	4	128	0.1550	0.9480	0.0900
8e1a1ca6	10^{-4}	16	2	32	4	4	128	0.1550	0.9480	0.0894
df056a84	10^{-3}	16	2	32	4	8	128	0.1550	0.9464	0.0879
ef40e0e8	10^{-5}	16	1	32	8	8	32	0.1550	0.9480	0.0903
12f74e1f	10^{-4}	16	2	64	8	8	128	0.1549	0.9477	0.0901
4388414d	10^{-4}	8	2	128	4	4	64	0.1549	0.9475	0.0893
6bf0a668	10^{-4}	8	4	128	8	8	64	0.1549	0.9457	0.0874
2732b7e6	10^{-4}	16	1	32	4	4	64	0.1549	0.9480	0.0899
0c6a2e4f	10^{-4}	16	2	64	8	8	128	0.1549	0.9480	0.0893
9acccc4e	10^{-4}	16	1	128	8	8	128	0.1549	0.9480	0.0906
48d82fcf	10^{-4}	8	2	128	8	8	64	0.1549	0.9480	0.0892
cb7a31b2	10^{-5}	16	1	128	8	4	128	0.1549	0.9480	0.0906
f43f6ad9	10^{-3}	8	2	128	4	4	32	0.1548	0.9480	0.0884
9bd919d4	10^{-4}	8	2	64	8	4	64	0.1548	0.9480	0.0889
ffc28acb	10^{-4}	16	1	32	4	8	128	0.1548	0.9480	0.0904

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
ebf28c9d	10^{-3}	8	2	32	4	4	128	0.1548	0.9141	0.0865
18dd6cca	10^{-5}	16	1	32	4	4	64	0.1548	0.9480	0.0900
b84b9173	10^{-4}	8	1	32	4	4	128	0.1548	0.9478	0.0897
4292031	10^{-4}	8	–	32	4	4	128	0.1548	0.9479	0.0892
c87388dd	10^{-3}	8	4	128	4	4	64	0.1547	0.9480	0.0890
a1fe7c34	10^{-3}	8	4	32	4	4	32	0.1547	0.9246	0.0853
e13b242c	10^{-3}	16	2	32	4	4	128	0.1547	0.9363	0.0871
d2533f4d	10^{-3}	16	4	32	8	8	64	0.1547	0.9478	0.0900
753c8c70	10^{-4}	16	1	32	8	8	64	0.1547	0.9479	0.0892
74299eb3	10^{-3}	16	4	64	4	8	64	0.1546	0.9417	0.0881
ec82a0ed	10^{-3}	8	1	128	8	4	64	0.1545	0.9480	0.0897
d0f902e7	10^{-4}	16	2	128	8	8	32	0.1545	0.9480	0.0900
bab94b75	10^{-3}	8	2	64	8	4	128	0.1545	0.9238	0.0871
c77dd285	10^{-4}	8	4	128	8	8	128	0.1545	0.9480	0.0901
b49f11d2	10^{-5}	16	1	64	4	8	128	0.1545	0.9480	0.0901
90150043	10^{-4}	8	4	64	8	4	32	0.1545	0.9480	0.0880
c84fe50c	10^{-3}	16	1	32	8	8	128	0.1545	0.9337	0.0862
62119303	10^{-3}	16	1	128	8	4	32	0.1545	0.9310	0.0863
6bd924fc	10^{-4}	8	4	32	4	8	128	0.1544	0.9480	0.0897
09df8564	10^{-3}	8	2	32	4	4	64	0.1544	0.9450	0.0879
c83d84de	10^{-3}	8	2	64	8	8	128	0.1544	0.9449	0.0867
66bd95b9	10^{-4}	16	4	32	4	8	64	0.1544	0.9480	0.0895
602abaec	10^{-5}	16	1	32	8	4	32	0.1544	0.9480	0.0896
9191fa7b	10^{-4}	16	4	128	4	4	64	0.1544	0.9480	0.0901
4f04576a	10^{-4}	16	1	128	8	4	64	0.1544	0.9480	0.0881
f7310f82	10^{-4}	8	1	128	8	4	32	0.1544	0.9480	0.0893
fa3011f6	10^{-3}	16	2	32	8	4	64	0.1544	0.9448	0.0875
17af7969	10^{-4}	8	2	128	4	4	32	0.1543	0.9480	0.0898
601d1457	10^{-4}	16	4	128	4	8	64	0.1543	0.9478	0.0898
a1c8107f	10^{-4}	16	4	64	4	8	64	0.1542	0.9480	0.0900
4fc101d8	10^{-3}	8	1	64	4	4	128	0.1542	0.9263	0.0866
9a0b4c74	10^{-3}	16	4	128	8	8	64	0.1542	0.9478	0.0900
98636378	10^{-4}	16	2	64	4	4	32	0.1542	0.9480	0.0891
05cabd29	10^{-5}	16	2	32	4	8	32	0.1542	0.9480	0.0895
b6d3e482	10^{-3}	8	1	64	4	4	32	0.1542	0.9270	0.0887
d9dbdb9f	10^{-4}	8	4	32	8	4	32	0.1542	0.9479	0.0893
b79d7c3c	10^{-4}	8	1	64	4	4	128	0.1542	0.9469	0.0898
2c32996f	10^{-5}	8	1	128	8	4	32	0.1542	0.9480	0.0894
eacfb7bb	10^{-4}	16	1	64	8	4	64	0.1541	0.9480	0.0894

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
538a071a	10^{-3}	16	2	64	4	8	64	0.1541	0.9442	0.0892
c296fd16	10^{-3}	16	1	128	4	4	32	0.1541	0.9480	0.0886
73317239	10^{-5}	8	4	32	8	8	32	0.1540	0.9480	0.0901
d3c23111	10^{-3}	8	4	128	4	8	128	0.1540	0.9352	0.0862
427ac2f8	10^{-3}	16	2	128	8	8	128	0.1540	0.9270	0.0872
45a9aa56	10^{-4}	16	1	128	4	4	64	0.1540	0.9480	0.0907
f4272919	10^{-3}	16	1	128	8	4	64	0.1540	0.9316	0.0912
e577edb1	10^{-5}	16	2	32	8	4	32	0.1540	0.9480	0.0898
01c631cb	10^{-3}	8	1	64	8	4	32	0.1539	0.9262	0.0885
015e52d6	10^{-4}	8	4	64	8	4	64	0.1539	0.9480	0.0885
80cf46e3	10^{-3}	8	4	64	8	4	64	0.1539	0.9419	0.0876
3f09af48	10^{-3}	8	2	32	4	8	32	0.1539	0.9231	0.0859
de6d63c3	10^{-3}	8	2	64	8	8	64	0.1539	0.9246	0.0875
ef0d0eee	10^{-4}	8	1	64	8	4	32	0.1539	0.9480	0.0893
e248307d	10^{-5}	16	1	128	4	4	64	0.1538	0.9480	0.0884
b0d7eed5	10^{-3}	16	2	128	4	8	32	0.1538	0.9330	0.0900
61b85a6b	10^{-3}	8	2	32	8	4	128	0.1538	0.9469	0.0898
a39816da	10^{-4}	8	4	128	8	4	32	0.1538	0.9438	0.0882
7024c46f	10^{-3}	16	2	128	8	8	32	0.1538	0.9480	0.0878
8d8d48a9	10^{-4}	16	4	64	4	4	64	0.1537	0.9480	0.0894
61a3376a	10^{-3}	8	2	64	4	8	64	0.1536	0.9265	0.0865
48db3d78	10^{-4}	16	4	128	4	8	128	0.1535	0.9480	0.0890
9548a11c	10^{-4}	8	1	32	8	4	128	0.1535	0.9480	0.0890
b2db58e7	10^{-5}	8	2	128	4	4	64	0.1535	0.9480	0.0898
42c3459d	10^{-3}	16	4	32	4	4	32	0.1535	0.9480	0.0875
816ad134	10^{-4}	8	2	64	4	8	64	0.1535	0.9480	0.0902
cf39904a	10^{-4}	16	2	32	8	4	32	0.1535	0.9479	0.0901
1e388b8d	10^{-3}	8	1	32	4	8	32	0.1534	0.9411	0.0875
b0eccf46	10^{-3}	16	1	32	8	8	64	0.1534	0.9480	0.0876
9b41608b	10^{-3}	8	1	128	4	4	128	0.1534	0.9295	0.0860
ec8e6868	10^{-3}	16	4	64	4	8	128	0.1533	0.9400	0.0876
83b1cbfe	10^{-3}	8	2	32	4	4	32	0.1533	0.9295	0.0871
3e2b7c89	10^{-4}	16	1	128	4	8	64	0.1532	0.9480	0.0895
39127a5c	10^{-4}	16	1	128	4	8	64	0.1532	0.9470	0.0889
617400d4	10^{-3}	8	2	128	8	4	128	0.1531	0.9480	0.0876
7e31e154	10^{-4}	16	2	128	8	4	64	0.1531	0.9480	0.0899
1e288d71	10^{-4}	16	4	32	8	8	64	0.1531	0.9480	0.0899
7f971897	10^{-4}	8	1	128	8	8	128	0.1531	0.9480	0.0890
e1abb7cf	10^{-3}	16	2	64	4	4	32	0.1531	0.9480	0.0896

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
e7a1edc7	10^{-3}	8	1	128	8	4	128	0.1530	0.9283	0.0860
4a4780b1	10^{-4}	16	2	128	8	8	128	0.1530	0.9478	0.0901
740a94fb	10^{-4}	8	4	64	8	8	128	0.1529	0.9480	0.0897
b56d6016	10^{-3}	16	2	64	4	8	32	0.1529	0.9273	0.0874
15a4336e	10^{-3}	8	2	64	8	4	128	0.1529	0.9473	0.0880
33001eb5	10^{-3}	16	2	128	4	8	128	0.1528	0.9453	0.0867
ae1d4d1b	10^{-4}	8	2	32	4	4	128	0.1528	0.9480	0.0896
6a794752	10^{-4}	8	1	64	4	8	64	0.1528	0.9479	0.0880
3b2930ca	10^{-4}	16	2	32	8	8	64	0.1528	0.9480	0.0896
b1acf574	10^{-5}	16	2	128	4	8	64	0.1527	0.9480	0.0890
ad18fcf2	10^{-4}	16	4	128	8	8	64	0.1526	0.9480	0.0900
3df3339c	10^{-4}	16	2	64	4	8	128	0.1526	0.9457	0.0884
5f5233f6	10^{-4}	16	2	128	8	4	128	0.1526	0.9479	0.0880
e24cb1c0	10^{-4}	16	1	32	8	8	32	0.1525	0.9480	0.0895
f210f2de	10^{-3}	8	2	64	8	4	32	0.1525	0.9480	0.0890
2797dd93	10^{-4}	8	4	64	8	4	128	0.1524	0.9442	0.0873
4dec833d	10^{-4}	16	4	32	4	4	128	0.1524	0.9478	0.0900
285fda14	10^{-4}	8	4	32	4	8	32	0.1524	0.9479	0.0903
0f369d26	10^{-3}	16	2	32	8	4	64	0.1524	0.9277	0.0875
d91a6305	10^{-3}	16	1	32	8	4	128	0.1523	0.9291	0.0870
d2902ba6	10^{-4}	16	2	64	4	4	128	0.1523	0.9479	0.0884
e52ce2a9	10^{-3}	16	2	64	8	4	64	0.1523	0.9286	0.0860
a3272f32	10^{-5}	16	4	32	8	4	64	0.1523	0.9480	0.0892
cc719868	10^{-3}	8	2	128	4	8	128	0.1522	0.9276	0.0855
fa94d21c	10^{-3}	16	1	32	8	4	32	0.1522	0.9480	0.0868
c9db1e8b	10^{-4}	8	1	32	8	4	64	0.1521	0.9480	0.0899
618e6ae3	10^{-4}	8	4	128	4	8	32	0.1521	0.9438	0.0873
b328c499	10^{-3}	16	1	128	8	8	128	0.1521	0.9480	0.0892
876d2568	10^{-3}	16	4	32	4	4	64	0.1521	0.9457	0.0889
a2dd56d1	10^{-3}	16	1	32	4	4	32	0.1520	0.9203	0.0876
bee25bce	10^{-3}	16	4	32	4	8	64	0.1519	0.9364	0.0862
8e618aaa	10^{-4}	8	1	128	8	8	64	0.1519	0.9478	0.0888
d0341018	10^{-3}	16	2	128	4	8	32	0.1519	0.9480	0.0889
7cd6f23d	10^{-3}	8	4	32	4	8	64	0.1518	0.9440	0.0883
b15000df	10^{-3}	16	1	32	8	4	32	0.1517	0.9376	0.0883
377a8c45	10^{-5}	8	4	128	8	4	32	0.1517	0.9480	0.0900
61838073	10^{-3}	16	2	32	8	8	32	0.1517	0.9261	0.0879
54ef36ec	10^{-4}	8	1	128	4	4	64	0.1517	0.9435	0.0890
d6296f9e	10^{-4}	16	4	128	4	4	32	0.1516	0.9479	0.0893

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
92b3d25a	10^{-3}	16	2	64	8	8	64	0.1516	0.9394	0.0837
c489f124	10^{-4}	8	4	32	4	8	128	0.1516	0.9480	0.0897
10a5060e	10^{-3}	8	1	128	8	4	32	0.1516	0.9320	0.0867
5410e2fd	10^{-4}	16	2	64	8	8	32	0.1516	0.9479	0.0893
5bb56c5b	10^{-3}	16	4	64	8	8	128	0.1515	0.9479	0.0882
4ccdde5a	10^{-3}	16	4	128	8	4	128	0.1515	0.9480	0.0912
5457c8f8	10^{-4}	8	1	64	8	4	128	0.1515	0.9480	0.0891
4fd0efbf	10^{-3}	16	2	128	4	4	64	0.1515	0.9289	0.0845
b08da6b0	10^{-3}	16	2	32	8	8	32	0.1514	0.9480	0.0883
255243fd	10^{-4}	8	2	32	8	8	128	0.1514	0.9480	0.0887
2511b0f9	10^{-5}	16	4	32	4	4	32	0.1514	0.9480	0.0882
dc0121a9	10^{-3}	16	4	32	8	8	32	0.1514	0.9480	0.0899
8b00ad0f	10^{-4}	16	4	32	4	8	128	0.1512	0.9478	0.0889
bb23ed92	10^{-4}	8	2	32	8	8	128	0.1512	0.9480	0.0888
cceac3af	10^{-3}	8	4	64	8	8	64	0.1512	0.9477	0.0890
ca09e9f4	10^{-4}	16	1	128	4	4	128	0.1511	0.9472	0.0906
27aca4bf	10^{-3}	16	2	32	4	8	32	0.1511	0.9464	0.0890
eb2c6791	10^{-3}	8	1	32	8	4	32	0.1511	0.9323	0.0861
bee953c3	10^{-4}	16	1	128	8	4	64	0.1510	0.9466	0.0888
1ab752e2	10^{-4}	8	1	64	8	4	32	0.1510	0.9476	0.0898
d6be5bb6	10^{-3}	8	2	32	8	8	32	0.1509	0.9294	0.0869
336fbd8e	10^{-3}	8	4	32	8	4	32	0.1509	0.9355	0.0879
9041630b	10^{-3}	16	4	32	4	4	128	0.1509	0.9475	0.0876
b50d0653	10^{-3}	8	1	128	8	8	32	0.1508	0.9442	0.0867
d1904bae	10^{-3}	16	4	128	4	4	64	0.1507	0.9441	0.0862
6.664E+063	10^{-3}	16	–	128	4	4	32	0.1507	0.9480	0.0895
fb198a5c	10^{-3}	16	2	32	8	8	128	0.1507	0.9463	0.0888
07277c7a	10^{-3}	8	1	32	4	4	128	0.1506	0.9480	0.0888
315c9eb7	10^{-3}	16	4	64	4	4	64	0.1505	0.9246	0.0865
f19e6a8b	10^{-3}	8	1	128	4	8	64	0.1505	0.9310	0.0869
c4d314ba	10^{-3}	16	1	128	8	8	32	0.1505	0.9480	0.0887
743c1ea6	10^{-4}	16	2	128	8	8	64	0.1505	0.9468	0.0887
f2be0e18	10^{-4}	8	2	128	4	8	32	0.1504	0.9480	0.0901
b6dd57c8	10^{-3}	8	2	32	8	4	32	0.1503	0.9350	0.0879
81bb128c	10^{-3}	8	2	128	8	4	64	0.1503	0.9477	0.0901
fd90c16c	10^{-3}	16	2	128	8	4	64	0.1502	0.9248	0.0847
65bc5ff3	10^{-3}	8	1	32	8	8	64	0.1502	0.9271	0.0880
f035a7a7	10^{-3}	16	4	32	4	4	128	0.1501	0.9347	0.0853
cd16bde7	10^{-5}	16	1	32	8	8	32	0.1500	0.9479	0.0901

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
00ebc49c	10^{-3}	16	4	64	4	4	128	0.1499	0.9391	0.0860
5af1ff67	10^{-4}	16	2	128	8	4	32	0.1499	0.9479	0.0900
ce7624d4	10^{-3}	16	2	128	8	8	128	0.1499	0.9479	0.0893
71da8ef4	10^{-3}	16	2	128	8	4	128	0.1499	0.9441	0.0874
b5cfcaab	10^{-3}	16	1	32	8	4	64	0.1497	0.9282	0.0870
9c7287ee	10^{-3}	16	4	128	4	4	128	0.1497	0.9437	0.0860
ade64837	10^{-3}	8	4	32	4	4	64	0.1497	0.9438	0.0873
24b457fa	10^{-3}	8	1	32	8	4	64	0.1497	0.9480	0.0887
429058a3	10^{-3}	16	1	64	4	4	64	0.1496	0.9240	0.0835
4649181f	10^{-3}	16	2	64	8	8	32	0.1494	0.9471	0.0873
19ed5d5b	10^{-3}	16	1	128	4	4	64	0.1494	0.9480	0.0883
a0295a78	10^{-3}	8	1	64	4	8	64	0.1493	0.9480	0.0873
6ff8b4ee	10^{-4}	8	1	128	4	4	128	0.1492	0.9471	0.0890
6241fc9a	10^{-3}	8	2	32	8	8	64	0.1491	0.9357	0.0878
8b91c49a	10^{-4}	8	2	128	8	4	32	0.1490	0.9480	0.0891
eba38d11	10^{-4}	8	2	64	4	4	128	0.1489	0.9474	0.0890
8d4d1fb3	10^{-3}	8	1	32	8	4	64	0.1489	0.9295	0.0880
7b04eff5	10^{-3}	16	4	32	8	4	128	0.1487	0.9480	0.0884
a6a92b02	10^{-5}	16	1	32	4	8	32	0.1487	0.9480	0.0883
9c3abe67	10^{-3}	8	1	32	4	4	32	0.1487	0.9480	0.0881
b38f9523	10^{-3}	8	1	64	4	8	64	0.1485	0.9206	0.0837
91bf0aa7	10^{-3}	16	1	32	4	4	64	0.1485	0.9379	0.0884
35afdf99	10^{-4}	8	4	32	8	4	128	0.1485	0.9480	0.0887
d6b8a657	10^{-3}	8	1	32	4	4	64	0.1483	0.9163	0.0868
3e10de7e	10^{-3}	16	4	32	8	4	32	0.1483	0.9292	0.0837
c2a28fb7	10^{-3}	16	4	128	4	8	32	0.1482	0.9430	0.0870
7f079bd6	10^{-3}	16	2	32	4	8	64	0.1482	0.9295	0.0852
506b61f5	10^{-3}	16	1	32	4	8	32	0.1481	0.9412	0.0868
840ae3ca	10^{-3}	8	4	32	8	4	128	0.1481	0.9475	0.0887
5d2a542f	10^{-3}	8	4	32	4	8	32	0.1480	0.9471	0.0882
2d374009	10^{-4}	16	1	128	8	8	128	0.1480	0.9475	0.0886
1e84baf9	10^{-4}	16	1	128	8	4	32	0.1480	0.9480	0.0882
6c79c53e	10^{-3}	16	1	128	8	4	128	0.1479	0.9191	0.0866
76a58097	10^{-3}	8	1	64	8	8	64	0.1475	0.9480	0.0857
54407887	10^{-4}	8	1	128	8	8	32	0.1474	0.9478	0.0895
cf611748	10^{-3}	16	1	64	4	8	32	0.1473	0.9269	0.0847
3b9eb414	10^{-3}	8	2	64	4	8	32	0.1471	0.9239	0.0877
18d541ff	10^{-3}	8	2	32	4	8	128	0.1470	0.9275	0.0838
b8b5b7a8	10^{-3}	8	4	32	4	4	32	0.1469	0.9480	0.0865

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
cc85f76d	10^{-3}	16	2	128	8	8	32	0.1467	0.9391	0.0860
cd7239e8	10^{-3}	8	4	128	8	4	128	0.1466	0.9480	0.0852
9f293ef0	10^{-4}	8	2	128	4	8	64	0.1466	0.9475	0.0890
490f4e8d	10^{-3}	16	2	128	4	4	128	0.1466	0.9091	0.0858
54b65a33	10^{-4}	8	1	128	4	8	128	0.1465	0.9473	0.0881
4ced5d4b	10^{-3}	8	2	128	8	4	128	0.1465	0.9150	0.0819
f483e4c5	10^{-3}	8	2	128	4	4	128	0.1463	0.9139	0.0821
70277091	10^{-3}	8	4	32	4	4	64	0.1461	0.9369	0.0850
a73a77c9	10^{-3}	8	4	32	4	8	64	0.1460	0.9306	0.0843
ed8b0065	10^{-3}	16	4	32	8	8	64	0.1459	0.9343	0.0866
c1d482a3	10^{-3}	8	4	32	8	4	128	0.1459	0.9245	0.0834
a73b81bf	10^{-3}	8	2	64	4	4	32	0.1459	0.9319	0.0864
b775c58e	10^{-3}	8	2	64	4	4	128	0.1457	0.9197	0.0828
dc88a17c	10^{-3}	8	2	32	8	4	64	0.1456	0.9475	0.0875
d28c2d83	10^{-3}	8	4	64	4	4	128	0.1455	0.9282	0.0838
ff7f020d	10^{-3}	16	2	128	4	8	64	0.1455	0.9313	0.0832
65e188ad	10^{-3}	8	2	32	4	4	64	0.1453	0.9272	0.0849
0abfbc68	10^{-3}	16	1	64	4	4	128	0.1452	0.9144	0.0833
dc42aaf2	10^{-3}	8	1	64	4	8	128	0.1451	0.9242	0.0847
ad549d5b	10^{-3}	16	1	64	4	8	128	0.1450	0.9237	0.0847
b2652ebc	10^{-3}	16	1	32	8	8	32	0.1449	0.9266	0.0870
4b47a56f	10^{-3}	16	2	64	4	8	128	0.1448	0.9292	0.0859
e44c11b7	10^{-3}	8	2	32	8	8	128	0.1448	0.9214	0.0853
7293391b	10^{-3}	16	1	64	4	4	32	0.1443	0.9356	0.0872
d6d54898	10^{-3}	8	1	128	4	4	32	0.1442	0.9347	0.0855
69c69932	10^{-3}	16	2	64	4	4	32	0.1440	0.9325	0.0847
41fa95f2	10^{-4}	8	1	128	8	8	128	0.1439	0.9473	0.0864
abee9d2b	10^{-3}	8	2	32	4	4	128	0.1437	0.9471	0.0869
fa6ee440	10^{-3}	16	2	32	4	4	32	0.1437	0.9278	0.0870
ed9b361f	10^{-3}	16	2	64	8	4	128	0.1435	0.9322	0.0839
7635759d	10^{-3}	8	1	128	8	8	64	0.1435	0.9480	0.0883
03c20a1b	10^{-3}	8	1	128	8	8	128	0.1433	0.9108	0.0809
ab89c521	10^{-3}	16	2	32	4	4	32	0.1431	0.9476	0.0872
3f7b9a00	10^{-3}	8	1	64	4	4	64	0.1427	0.9298	0.0843
80c07eef	10^{-3}	8	1	128	4	4	64	0.1423	0.9348	0.0820
986eb435	10^{-3}	16	1	32	4	4	128	0.1412	0.9320	0.0830
f35f870b	10^{-3}	16	4	32	8	8	32	0.1401	0.9363	0.0846
0397e5ef	10^{-3}	16	2	128	4	4	32	0.1387	0.9336	0.0837
8c64bb84	10^{-3}	8	4	32	8	8	32	0.1381	0.9317	0.0825

Continúa en la siguiente página

Tabla A.1 – continuación

ID	LR	BS	Patch	Dim	Depth	Heads	MLP	Recall	Acc.	MAP
b869398a	10^{-3}	8	2	128	8	8	128	0.1365	0.9154	0.0772
3422bd20	10^{-3}	16	1	128	4	4	64	0.1352	0.9187	0.0826
7780cc69	10^{-3}	8	1	32	8	8	128	0.1346	0.9134	0.0819
c7457f2e	10^{-3}	8	2	64	8	8	128	0.1337	0.9179	0.0775

A.2. Resultados del modelo CNN-LSTM

La Tabla A.2 presenta los resultados de las 192 ejecuciones realizadas con el modelo CNN-LSTM, incluyendo la configuración de hiperparámetros y las métricas de evaluación obtenidas.

Tabla A.2: *Resultados de todas las ejecuciones del modelo CNN-LSTM, ordenados por Recall descendente*

ID	LR	BS	Seq	LSTM	Filtros CNN	Ctx	Recall	Acc.	mAP
095d9665	10^{-3}	8	7	128	[32,64,128]	–	0.1833	0.9206	0.0949
e41829af	10^{-5}	8	14	128	[32,64]	–	0.1740	0.9495	0.0928
20838ade	10^{-3}	16	7	128	[32,64,128]	–	0.1735	0.9492	0.0935
56b5dd4d	10^{-3}	16	14	128	[32,64]	–	0.1707	0.9264	0.0915
89c73d5f	10^{-3}	16	3	128	[32,64]	–	0.1700	0.9382	0.0907
c2f435a2	10^{-4}	16	7	128	[32,64]	–	0.1690	0.9492	0.0911
a6017ccc	10^{-5}	8	14	64	[32,64,128]	–	0.1687	0.9496	0.0916
fec6dbf3	10^{-4}	8	1	64	[32,64]	–	0.1686	0.9478	0.0899
d244cf4c	10^{-5}	8	7	64	[32,64,128]	–	0.1685	0.9484	0.0926
31b1403c	10^{-3}	8	14	128	[32,64]	–	0.1684	0.9484	0.0940
cecdc96c	10^{-3}	8	7	128	[32,64]	–	0.1683	0.9491	0.0936
afe5088d	10^{-5}	16	14	128	[32,64,128]	–	0.1680	0.9495	0.0899
7aa2e19b	10^{-4}	8	14	128	[32,64,128]	–	0.1678	0.9445	0.0925
5832c405	10^{-5}	16	7	64	[32,64]	–	0.1670	0.9492	0.0922
cca77519	10^{-5}	8	7	128	[32,64]	–	0.1657	0.9483	0.0917
d49fbfc0	10^{-4}	8	7	128	[32,64]	–	0.1655	0.9456	0.0927
903a2bbe	10^{-4}	8	7	128	[32,64,128]	–	0.1651	0.9489	0.0917
f5800dd5	10^{-4}	16	14	128	[32,64]	–	0.1647	0.9498	0.0907
91503a26	10^{-5}	16	14	128	[32,64]	–	0.1646	0.9490	0.0914
0b555f40	10^{-5}	16	7	64	[32,64,128]	–	0.1644	0.9492	0.0909
d99e2924	10^{-4}	8	14	128	[32,64]	–	0.1643	0.9488	0.0916
192af301	10^{-3}	8	7	128	[32,64]	–	0.1643	0.9326	0.0880
742cd4f3	10^{-3}	16	14	128	[32,64]	–	0.1643	0.9490	0.0935
4e74744d	10^{-4}	16	1	128	[32,64]	–	0.1643	0.9478	0.0901
9677e9c0	10^{-4}	16	14	64	[32,64]	–	0.1642	0.9490	0.0900

Continúa en la siguiente página

Tabla A.2 – continuación

ID	LR	BS	Seq	LSTM	Filtros CNN	Ctx	Recall	Acc.	mAP
3c4ed18c	10^{-5}	16	7	64	[32,64,128]	–	0.1642	0.9473	0.0905
7d1aad3e	10^{-4}	16	14	64	[32,64,128]	–	0.1642	0.9462	0.0906
b6e318f1	10^{-5}	16	7	128	[32,64]	–	0.1641	0.9483	0.0916
6c7e2b62	10^{-3}	16	7	128	[32,64]	–	0.1640	0.9472	0.0923
2bde5dcc	10^{-5}	8	7	128	[32,64]	–	0.1640	0.9492	0.0908
b14b5776	10^{-3}	8	7	64	[32,64,128]	–	0.1639	0.9221	0.0892
4e9704f5	10^{-3}	8	7	64	[32,64,128]	–	0.1634	0.9492	0.0912
e7fb23cd	10^{-3}	16	3	128	[32,64,128]	–	0.1634	0.9382	0.0887
dcc61b70	10^{-4}	8	1	128	[32,64]	–	0.1632	0.9493	0.0895
b6099749	10^{-5}	16	14	128	[32,64]	–	0.1630	0.9498	0.0916
e7e326fd	10^{-5}	8	14	64	[32,64]	–	0.1628	0.9498	0.0917
42e89611	10^{-3}	16	3	64	[32,64]	–	0.1628	0.9494	0.0907
e81a8b14	10^{-5}	16	14	64	[32,64,128]	–	0.1627	0.9498	0.0912
a33fc61d	10^{-4}	16	14	64	[32,64]	–	0.1627	0.9498	0.0910
d5e6711f	10^{-3}	16	1	128	[32,64]	–	0.1626	0.9493	0.0904
dddea486	10^{-4}	16	1	128	[32,64,128]	–	0.1622	0.9493	0.0900
d5300650	10^{-3}	8	14	64	[32,64]	–	0.1619	0.9498	0.0917
28838365	10^{-5}	16	14	128	[32,64,128]	–	0.1618	0.9498	0.0910
5f15d21a	10^{-4}	16	7	64	[32,64,128]	–	0.1618	0.9491	0.0921
2dc18817	10^{-4}	16	3	128	[32,64,128]	–	0.1617	0.9483	0.0900
90745938	10^{-3}	16	3	128	[32,64]	–	0.1616	0.9486	0.0907
a1bfa34e	10^{-4}	8	3	128	[32,64]	–	0.1616	0.9449	0.0885
8799aa32	10^{-3}	8	1	64	[32,64,128]	–	0.1616	0.9493	0.0903
e1470211	10^{-4}	8	14	64	[32,64]	–	0.1616	0.9495	0.0912
28320934	10^{-4}	8	3	64	[32,64,128]	–	0.1615	0.9444	0.0876
2c635a35	10^{-5}	16	14	64	[32,64]	–	0.1615	0.9498	0.0903
7b2e2ed2	10^{-5}	16	7	128	[32,64,128]	–	0.1614	0.9492	0.0916
5f661492	10^{-3}	16	3	64	[32,64,128]	–	0.1614	0.9484	0.0896
89f5c155	10^{-3}	16	3	128	[32,64,128]	–	0.1613	0.9487	0.0892
bdbbb41e	10^{-4}	8	3	128	[32,64]	–	0.1613	0.9493	0.0888
169ba2b8	10^{-5}	16	1	128	[32,64,128]	–	0.1612	0.9478	0.0912
de8db31c	10^{-4}	8	7	128	[32,64,128]	–	0.1611	0.9464	0.0896
5e4e2251	10^{-4}	16	3	64	[32,64]	–	0.1609	0.9489	0.0893
ce2be9d2	10^{-4}	8	14	128	[32,64]	–	0.1609	0.9456	0.0905
ce752ebf	10^{-5}	16	3	64	[32,64,128]	–	0.1608	0.9481	0.0898
bb63521c	10^{-4}	16	14	128	[32,64,128]	–	0.1608	0.9480	0.0903
d53d3b92	10^{-3}	16	14	64	[32,64,128]	–	0.1608	0.9498	0.0901
746f3f15	10^{-5}	8	14	128	[32,64,128]	–	0.1607	0.9498	0.0909
efd0f700	10^{-5}	16	7	128	[32,64]	–	0.1604	0.9492	0.0905

Continúa en la siguiente página

Tabla A.2 – continuación

ID	LR	BS	Seq	LSTM	Filtros CNN	Ctx	Recall	Acc.	mAP
d5f4b776	10^{-5}	8	7	128	[32,64,128]	–	0.1604	0.9492	0.0917
2596734	10^{-5}	8	–	–	–	–	0.1603	0.9478	0.0892
4ed2dda7	10^{-5}	16	1	64	[32,64,128]	–	0.1603	0.9482	0.0883
6c0deb35	10^{-3}	8	14	64	[32,64,128]	–	0.1603	0.9329	0.0898
95838813	10^{-3}	16	1	128	[32,64,128]	–	0.1602	0.9420	0.0883
12721bda	10^{-4}	8	7	128	[32,64]	–	0.1601	0.9492	0.0903
9ca8e2b8	10^{-4}	16	7	64	[32,64]	–	0.1601	0.9492	0.0897
3342e6ac	10^{-3}	8	14	128	[32,64,128]	–	0.1601	0.9475	0.0918
6c51a212	10^{-4}	8	7	64	[32,64]	–	0.1597	0.9489	0.0911
6cd5b832	10^{-5}	8	1	128	[32,64,128]	–	0.1597	0.9493	0.0897
eacf4674	10^{-4}	16	7	128	[32,64,128]	–	0.1597	0.9482	0.0900
33b385e1	10^{-4}	16	1	64	[32,64]	–	0.1596	0.9493	0.0896
3c1887b7	10^{-4}	8	7	64	[32,64,128]	–	0.1596	0.9470	0.0926
8fd5c2d1	10^{-3}	16	14	128	[32,64,128]	–	0.1595	0.9365	0.0870
6f19de89	10^{-3}	8	1	64	[32,64]	–	0.1595	0.9493	0.0906
60e4133f	10^{-4}	8	7	64	[32,64,128]	–	0.1594	0.9492	0.0907
8659e830	10^{-5}	16	14	64	[32,64]	–	0.1593	0.9493	0.0907
d08d9388	10^{-3}	8	7	64	[32,64]	–	0.1593	0.9492	0.0897
39ef8810	10^{-5}	8	3	64	[32,64,128]	–	0.1592	0.9492	0.0909
878d8459	10^{-5}	16	1	128	[32,64]	–	0.1589	0.9493	0.0895
f6172ecd	10^{-4}	8	7	64	[32,64]	–	0.1588	0.9463	0.0888
8937cb96	10^{-5}	8	1	128	[32,64]	–	0.1588	0.9493	0.0900
4b841463	10^{-5}	8	14	128	[32,64]	–	0.1586	0.9498	0.0916
fa4a53dd	10^{-3}	8	1	128	[32,64,128]	–	0.1586	0.9379	0.0854
c9e6a22e	10^{-5}	8	1	64	[32,64]	–	0.1585	0.9493	0.0889
7137543d	10^{-4}	16	14	128	[32,64]	–	0.1585	0.9486	0.0887
48b584ec	10^{-5}	16	3	64	[32,64]	–	0.1585	0.9494	0.0902
44ae815e	10^{-5}	8	1	64	[32,64]	–	0.1584	0.9492	0.0896
7c0c169f	10^{-4}	8	1	128	[32,64,128]	–	0.1584	0.9458	0.0923
2fa3add7	10^{-3}	8	3	128	[32,64]	–	0.1584	0.9488	0.0916
78a5d183	10^{-3}	8	3	128	[32,64]	–	0.1583	0.9382	0.0876
264a52e3	10^{-3}	16	7	64	[32,64,128]	–	0.1580	0.9481	0.0917
c575cdb2	10^{-4}	16	3	128	[32,64,128]	–	0.1579	0.9438	0.0901
0b55a491	10^{-4}	8	3	128	[32,64,128]	–	0.1579	0.9485	0.0906
a1b0a254	10^{-4}	16	7	64	[32,64]	–	0.1578	0.9475	0.0904
6abc3d68	10^{-4}	8	14	128	[32,64,128]	–	0.1578	0.9495	0.0903
f6b7eb38	10^{-5}	8	3	128	[32,64]	–	0.1578	0.9492	0.0892
29a90885	10^{-3}	16	1	64	[32,64,128]	–	0.1577	0.9493	0.0887
e0d6fec2	10^{-5}	8	3	128	[32,64]	–	0.1575	0.9494	0.0904

Continúa en la siguiente página

Tabla A.2 – continuación

ID	LR	BS	Seq	LSTM	Filtros CNN	Ctx	Recall	Acc.	mAP
217ac93c	10^{-3}	16	14	128	[32,64,128]	–	0.1575	0.9456	0.0893
7444d81b	10^{-5}	8	14	64	[32,64,128]	–	0.1575	0.9496	0.0902
6f21baa9	10^{-4}	8	1	64	[32,64]	–	0.1573	0.9493	0.0900
fabd2dae	10^{-5}	16	14	64	[32,64,128]	–	0.1570	0.9493	0.0906
81efe63f	10^{-4}	8	3	64	[32,64]	–	0.1568	0.9472	0.0886
b011c6ba	10^{-5}	8	1	128	[32,64,128]	–	0.1568	0.9484	0.0884
5e567521	10^{-5}	16	1	64	[32,64,128]	–	0.1567	0.9493	0.0897
8f1ef1f0	10^{-3}	16	7	64	[32,64]	–	0.1567	0.9412	0.0877
34aa3541	10^{-5}	16	3	128	[32,64]	–	0.1567	0.9480	0.0879
59959afa	10^{-3}	16	1	64	[32,64]	–	0.1566	0.9493	0.0888
b56da17b	10^{-5}	16	3	128	[32,64]	–	0.1566	0.9494	0.0893
25a8bbbb	10^{-4}	16	3	128	[32,64]	–	0.1562	0.9494	0.0899
d5764a90	10^{-3}	16	1	64	[32,64]	–	0.1559	0.9410	0.0892
58ad046b	10^{-4}	8	14	64	[32,64]	–	0.1559	0.9463	0.0879
98849166	10^{-5}	8	1	128	[32,64]	–	0.1558	0.9486	0.0889
0bfad8ea	10^{-5}	8	3	64	[32,64]	–	0.1558	0.9494	0.0898
93405a12	10^{-5}	8	14	64	[32,64]	–	0.1555	0.9494	0.0900
5bd31905	10^{-3}	16	7	64	[32,64,128]	–	0.1555	0.9266	0.0886
4c42e611	10^{-5}	16	7	128	[32,64,128]	–	0.1554	0.9488	0.0883
08b022d3	10^{-4}	16	3	64	[32,64]	–	0.1554	0.9494	0.0898
4a2d8da4	10^{-3}	8	3	128	[32,64,128]	–	0.1554	0.9277	0.0894
59bb5c6b	10^{-3}	16	14	64	[32,64]	–	0.1552	0.9498	0.0892
f42e6e90	10^{-5}	8	7	64	[32,64]	–	0.1551	0.9492	0.0908
83668d2a	10^{-4}	16	7	64	[32,64,128]	–	0.1549	0.9486	0.0879
474046	10^{-3}	8	–	–	–	–	0.1549	0.9481	0.0896
726b025c	10^{-5}	8	1	64	[32,64,128]	–	0.1549	0.9493	0.0892
710e1931	10^{-3}	16	7	64	[32,64]	–	0.1548	0.9492	0.0908
6430a4a4	10^{-4}	16	7	128	[32,64,128]	–	0.1548	0.9469	0.0903
23e0da4d	10^{-3}	16	7	128	[32,64,128]	–	0.1547	0.9344	0.0877
e32cd78a	10^{-4}	8	1	128	[32,64,128]	–	0.1546	0.9493	0.0893
b41cdae5	10^{-4}	8	14	64	[32,64,128]	–	0.1546	0.9494	0.0900
35001377	10^{-4}	8	3	64	[32,64,128]	–	0.1546	0.9483	0.0894
f1afc77a	10^{-3}	8	14	64	[32,64,128]	–	0.1544	0.9498	0.0888
ac00f1b4	10^{-4}	8	3	64	[32,64]	–	0.1544	0.9483	0.0881
544f0926	10^{-4}	16	1	64	[32,64]	–	0.1542	0.9489	0.0884
a9bfc6cb	10^{-3}	8	7	64	[32,64]	–	0.1542	0.9277	0.0890
4e12e3d3	10^{-5}	8	14	128	[32,64,128]	–	0.1541	0.9486	0.0902
ad3200e0	10^{-5}	16	1	64	[32,64]	–	0.1541	0.9493	0.0905
27c3a0fb	10^{-4}	8	14	64	[32,64,128]	–	0.1540	0.9476	0.0898

Continúa en la siguiente página

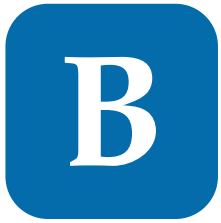
Tabla A.2 – continuación

ID	LR	BS	Seq	LSTM	Filtros CNN	Ctx	Recall	Acc.	mAP
8d8a19e3	10^{-4}	16	1	64	[32,64,128]	–	0.1537	0.9487	0.0875
3e9c1597	10^{-5}	8	7	64	[32,64]	–	0.1536	0.9479	0.0901
0fe40a5a	10^{-5}	8	7	64	[32,64,128]	–	0.1536	0.9492	0.0904
41305989	10^{-5}	8	3	64	[32,64]	–	0.1535	0.9491	0.0883
a575bf1f	10^{-4}	16	1	128	[32,64,128]	–	0.1534	0.9476	0.0879
d8325a7a	10^{-4}	8	1	128	[32,64]	–	0.1534	0.9478	0.0871
16fb5fa5	10^{-4}	16	14	64	[32,64,128]	–	0.1530	0.9498	0.0907
23a4f791	10^{-4}	8	1	64	[32,64,128]	–	0.1529	0.9469	0.0884
af0801d7	10^{-5}	16	7	64	[32,64]	–	0.1529	0.9483	0.0895
932cfc7f	10^{-4}	16	3	64	[32,64,128]	–	0.1527	0.9471	0.0882
432819bc	10^{-4}	16	3	64	[32,64,128]	–	0.1527	0.9494	0.0900
6.1511E+103	10^{-5}	16	–	–	–	–	0.1526	0.9493	0.0891
26e5ba2b	10^{-5}	16	3	128	[32,64,128]	–	0.1526	0.9494	0.0878
d6d71968	10^{-4}	16	3	128	[32,64]	–	0.1524	0.9439	0.0852
d6dd46a2	10^{-3}	8	14	128	[32,64]	–	0.1524	0.9328	0.0887
5ef3a1af	10^{-4}	16	14	128	[32,64,128]	–	0.1523	0.9450	0.0889
ab1d3378	10^{-3}	16	1	64	[32,64,128]	–	0.1522	0.9394	0.0880
2ddf8356	10^{-5}	16	3	64	[32,64,128]	–	0.1518	0.9494	0.0893
43b825a0	10^{-3}	8	1	128	[32,64]	–	0.1517	0.9479	0.0903
dac825b6	10^{-3}	16	14	64	[32,64]	–	0.1516	0.9463	0.0884
3c042440	10^{-5}	8	3	128	[32,64,128]	–	0.1515	0.9492	0.0887
a894e458	10^{-3}	8	1	128	[32,64]	–	0.1515	0.9333	0.0859
af218fb9	10^{-3}	8	14	64	[32,64]	–	0.1514	0.9432	0.0908
215ffafb	10^{-4}	16	1	128	[32,64]	–	0.1513	0.9491	0.0893
67944f5e	10^{-5}	16	3	64	[32,64]	–	0.1513	0.9491	0.0890
5b515380	10^{-4}	16	7	128	[32,64]	–	0.1512	0.9481	0.0882
1574139c	10^{-4}	16	1	64	[32,64,128]	–	0.1511	0.9478	0.0882
7777400e	10^{-3}	8	7	128	[32,64,128]	–	0.1511	0.9480	0.0889
e3f6e5ae	10^{-5}	16	3	128	[32,64,128]	–	0.1510	0.9484	0.0878
2c19ba47	10^{-3}	16	14	64	[32,64,128]	–	0.1507	0.9447	0.0858
5aac9c08	10^{-5}	16	1	128	[32,64]	–	0.1507	0.9493	0.0885
dcd6ed1d	10^{-4}	8	1	64	[32,64,128]	–	0.1506	0.9493	0.0890
3d833d35	10^{-3}	8	3	64	[32,64,128]	–	0.1502	0.9494	0.0879
46b79e1b	10^{-3}	8	3	64	[32,64]	–	0.1499	0.9283	0.0885
321dd1bf	10^{-5}	16	1	64	[32,64]	–	0.1494	0.9489	0.0895
20ee969c	10^{-5}	8	3	64	[32,64,128]	–	0.1491	0.9477	0.0882
50e217ad	10^{-5}	8	3	128	[32,64,128]	–	0.1491	0.9492	0.0893
bb1af2d0	10^{-3}	16	1	128	[32,64]	–	0.1488	0.9371	0.0868
f413fdfb	10^{-3}	8	1	64	[32,64]	–	0.1487	0.9407	0.0875

Continúa en la siguiente página

Tabla A.2 – continuación

ID	LR	BS	Seq	LSTM	Filtros CNN	Ctx	Recall	Acc.	mAP
c237e616	10^{-3}	16	3	64	[32,64,128]	–	0.1481	0.9404	0.0860
47c5e7a5	10^{-3}	8	3	64	[32,64,128]	–	0.1480	0.9441	0.0848
d2edd373	10^{-3}	8	1	64	[32,64,128]	–	0.1478	0.9331	0.0876
1af2a6f3	10^{-3}	16	7	128	[32,64]	–	0.1477	0.9364	0.0859
74beaa6b	10^{-3}	8	3	128	[32,64,128]	–	0.1473	0.9470	0.0870
c08dc279	10^{-3}	8	3	64	[32,64]	–	0.1464	0.9494	0.0869
ddabb4ea	10^{-5}	8	1	64	[32,64,128]	–	0.1457	0.9487	0.0869
1d26a861	10^{-4}	8	3	128	[32,64,128]	–	0.1450	0.9448	0.0867
02ed031b	10^{-3}	16	1	128	[32,64,128]	–	0.1427	0.9385	0.0838
c0cf3767	10^{-3}	8	14	128	[32,64,128]	–	0.1427	0.9371	0.0838
52ac33fc	10^{-3}	16	3	64	[32,64]	–	0.1356	0.9408	0.0840



Código fuente del proyecto

El código fuente del proyecto puede encontrarse en el siguiente repositorio alojado en GitFront: `https://gitfront.io/r/andresmaglione/9pWX2NZ4SEnM/tesina-lcc/`. Allí, puede encontrarse el código completo, incluyendo análisis exploratorio de datos, implementación de modelos de DL, desarrollo de la aplicación web, scripts para la automatización de experimentos y los archivos LaTeX utilizados para la redacción de este documento. Para mayor información sobre la estructura del repositorio, se recomienda consultar el archivo *README.md* incluido en el mismo, que contiene detalles sobre la organización del proyecto.

Página intencionalmente dejada en blanco.