# Modelo de aprendizaje profundo para la clasificación automática de escenas en imágenes satelitales (MAPCES)

Kmila Andrea Contreras Amaya y Monica Cristina Peña Rincon

Trabajo de Grado para optar al título de ingeniería electrónica

Director

Hans Yecid Garcia Arenas

Ingeniero Electrónico y Doctor en Ingeniería

Universidad Industrial de Santander

Facultad de fisico mecanica

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones

Bucaramanga

2026

## Dedicatoria

Dedicamos este trabajo a nuestras familias, por su amor, apoyo incondicional y confianza en cada etapa de nuestra formación.

A nuestros docentes y a todas las personas que, con sus enseñanzas, consejos y acompañamiento, contribuyeron a nuestro crecimiento personal y profesional.

Finalmente, a nosotras mismas, por la constancia, el esfuerzo y la dedicación que hicieron posible culminar este proceso.

## Agradecimientos

Agradecemos a la Universidad Industrial de Santander por la formación académica recibida y por los espacios brindados para nuestro crecimiento profesional.

Asimismo, expresamos nuestro agradecimiento a nuestros docentes y director de proyecto, por su orientación, acompañamiento y valiosos aportes durante el desarrollo de este trabajo.

De manera especial, agradecemos a nuestras familias por su apoyo, comprensión y motivación constante a lo largo de este proceso.

Finalmente, agradecemos a todas las personas que, de una u otra manera, contribuyeron al desarrollo y culminación satisfactoria de este proyecto.

**Tabla de Contenido**

**Pág.**

**Introducción** ..... 15

**1. Objetivos** ..... 19

1.1 Objetivo general ..... 19

1.2 Objetivos específicos ..... 19

**2. Conceptos previos** ..... 20

2.1 Clasificación de escenas en imágenes satelitales ..... 20

2.2 Dataset UC Merced ..... 21

2.3 Fundamentos esenciales de redes neuronales convolucionales ..... 21

2.4 Regularización y generalización del modelo ..... 22

2.5 Métricas de evaluación en clasificación multiclase ..... 23

**3. Preprocesamiento y preparación del conjunto de datos** ..... 24

3.1 Descripción general del flujo de preparación de datos ..... 24

3.2 Organización del conjunto de datos y mapeo de clases ..... 24

3.3 Preprocesamiento aplicado a las imágenes ..... 25

3.4 Partición estratificada del conjunto de datos ..... 26

3.5 Codificación de etiquetas para clasificación multiclase ..... 26

**4. Diseño e implementación del modelo de aprendizaje profundo** ..... 28

4.1 Criterios generales de diseño del modelo ..... 28

4.2 Baseline reproducido ..... 28

4.3 Modelo propuesto ..... 29

4.4 Comparación estructural y selección final ..... 30

**5. Entrenamiento, optimización y reproducibilidad del modelo** ..... 31

5.1 Formulación del proceso de entrenamiento ..... 31

5.2 Configuración del optimizador y función de pérdida ..... 32

5.3 Aumentación de datos en línea ..... 32

5.4 Estrategias de control y optimización del entrenamiento ..... 33

5.5 Curvas de entrenamiento y criterio de convergencia ..... 33

5.6 Entorno de ejecución ..... 35

5.7 Reproducibilidad del experimento ..... 35

**6. Resultados y evaluación del desempeño del modelo** ..... 36

6.1 Protocolo de evaluación del modelo ..... 36

6.2 Desempeño global en entrenamiento, validación y prueba ..... 37

6.3 Análisis mediante matrices de confusión ..... 38

6.4 Síntesis de métricas por clase y promedios globales ..... 39

6.5 Discusión general de resultados ..... 40

**7. Evaluación de robustez del modelo propuesto ante perturbaciones controladas** ..... 41

7.1 Planteamiento general del análisis de robustez ..... 41

7.2 Resultados globales ante perturbaciones controladas ..... 41

7.3 Discusión general del análisis de robustez ..... 43

**8. Interpretabilidad de resultados mediante mapas de saliencia** ..... 44

8.1 Propósito de la interpretabilidad en el estudio ..... 44

8.2 Síntesis del análisis interpretativo ..... 45

8.3 Alcances y limitaciones del análisis interpretativo ..... 45

**9. Evaluación de un modelo robusto inspirado en la literatura** ..... 47

9.1 Síntesis de la implementación adaptada ..... 47

9.2 Resultados globales del modelo robusto adaptado ..... 48

9.3 Discusión general ..... 49

**Referencias Bibliográficas** ..... 53

**Apéndices** ..... 57

## Lista de Figuras

**Pág.**

**Figura 1.** Flujo metodológico general implementado en el estudio. . . . . 25

**Figura 2.** Curvas de entrenamiento del modelo propuesto. . . . . 34

**Figura 3.** Matriz de confusión del modelo final. . . . . 38

**Figura 4.** Mapa de saliencia para una muestra de la clase *airplane*. . . . . 59
**Figura 5.** Mapa de saliencia para una muestra de la clase *mediumresidential*. . . . . 60
**Figura 6.** Mapa de saliencia para un caso de error de clasificación. . . . . 61
**Figura 7.** Arquitectura del modelo *baseline*. . . . . 63
**Figura 8.** Ejemplos representativos de clases del conjunto de datos. . . . . 65
**Figura 9.** Matriz de confusión del modelo en validación. . . . . 90
**Figura 10.** Matriz de confusión del modelo en entrenamiento. . . . . 91

## Lista de Tablas

**Pág.**

**Tabla 1.** Resumen de la partición estratificada del conjunto de datos. . . . . 26

**Tabla 2.** Resultados globales del modelo en entrenamiento, validación y prueba. . . . . 37

**Tabla 3.** Resumen global del desempeño del modelo propuesto ante perturbaciones controladas. . . . . 42

**Tabla 4.** Resumen global del desempeño del modelo robusto adaptado inspirado en DSFATN. . . . . 48

**Tabla 5.** Comparación estructural entre el baseline reproducido y el modelo propuesto. . . 64

**Tabla 6.** Resultados globales del modelo propuesto ante reducción de resolución. . . . . 70

**Tabla 7.** Resultados globales del modelo propuesto ante ruido AWGN con SNR controlada. 72

**Tabla 8.** Resultados globales del modelo propuesto ante desenfoque por *kernels* promedio. 73

**Tabla 9.** Correspondencia entre el enfoque DSFATN original y la adaptación implementada en esta tesis. . . . . 85

**Tabla 10.** Resultados globales del modelo robusto adaptado basado en DSFATN. . . . . 88

**Tabla 11.** Métricas por clase del modelo en el conjunto de prueba. . . . . 92

## Lista de Apéndices

**Pág.**

Apéndice A. Análisis ampliado de interpretabilidad mediante mapas de saliencia ..... 57
Apéndice B. Arquitectura del baseline reproducido y comparación estructural de modelos ... 63
Apéndice C. Ejemplos representativos del conjunto de datos..... 65
Apéndice D. Estado del arte ampliado sobre CNN aplicadas a UC Merced..... 67
Apéndice E. Evaluación ampliada de robustez del modelo propuesto ante perturbaciones controladas ..... 69
Apéndice F. Fundamentos complementarios de aprendizaje profundo para visión..... 76
Apéndice G. Fundamentos de interpretabilidad mediante mapas de saliencia..... 80
Apéndice H. Fundamentos de robustez en modelos de visión ..... 82
Apéndice I. Implementación adaptada y evaluación ampliada del enfoque DSFATN ..... 84
Apéndice J. Matrices de confusión complementarias..... 90
Apéndice K. Métricas por clase del modelo en el conjunto de prueba..... 92
Apéndice L. Repositorio del proyecto y material complementario..... 93

## Glosario

**Accuracy** métrica de evaluación que representa la proporción de predicciones correctas realizadas por el modelo respecto al total de muestras evaluadas.

**AWGN** sigla de *Additive White Gaussian Noise*; corresponde a un modelo de ruido aditivo, blanco y gaussiano utilizado para simular degradaciones aleatorias en señales e imágenes.

**Batch normalization** técnica de normalización por lotes que estabiliza y acelera el entrenamiento de redes neuronales al normalizar las activaciones internas de una capa. En entrenamiento usa la media y varianza del lote actual, mientras que en inferencia utiliza estadísticas móviles aprendidas durante el entrenamiento, junto con parámetros de escala y desplazamiento.

**Benchmark** conjunto de referencia utilizado para evaluar y comparar el desempeño de modelos o metodologías bajo condiciones experimentales definidas.

**CNN** sigla de *Convolutional Neural Network*; corresponde a una red neuronal convolucional, adecuada para el procesamiento de imágenes debido a su capacidad para extraer patrones espaciales como bordes, texturas y formas.

**Conjunto de prueba** subconjunto de datos no utilizado durante el entrenamiento, empleado para estimar el desempeño final del modelo.

**Conjunto de validación** subconjunto de datos utilizado para monitorear el desempeño del modelo durante el entrenamiento y apoyar la selección de hiperparámetros.

**Data augmentation** conjunto de técnicas utilizadas para generar nuevas muestras de entrenamiento a partir de las imágenes originales mediante transformaciones controladas, con el fin de mejorar la capacidad de generalización del modelo.

**Dropout** técnica de regularización que consiste en desactivar aleatoriamente una fracción de neuronas durante el entrenamiento, con el fin de reducir el sobreajuste.

**F1-score** métrica de evaluación que combina precisión y *recall* en un único valor, siendo útil cuando se desea analizar el equilibrio entre ambas.

**Hiperparámetro** parámetro definido antes del proceso de entrenamiento, como la tasa de aprendizaje, el tamaño de lote o el número de épocas.

**Matriz de confusión** tabla que permite comparar las clases reales con las clases predichas por el modelo, facilitando la identificación de aciertos y errores de clasificación.

**One-hot encoding** forma de representación de etiquetas categóricas en la que cada clase se expresa como un vector binario, donde solo una posición toma el valor uno y las demás toman valor cero.

**Overfitting** fenómeno que ocurre cuando un modelo aprende excesivamente los patrones particulares del conjunto de entrenamiento, logrando buen desempeño en esos datos pero baja capacidad de generalización sobre datos no vistos.

**Precisión** métrica que indica la proporción de muestras predichas como pertenecientes a una clase que realmente corresponden a dicha clase.

**Recall** métrica que representa la proporción de muestras de una clase que fueron correctamente identificadas por el modelo.

**Regularización** conjunto de técnicas orientadas a controlar la complejidad del modelo y mejorar su capacidad de generalización.

**Robustez** capacidad de un modelo para mantener un desempeño adecuado frente a perturbaciones o degradaciones en los datos de entrada.

**Saliency map** representación visual que resalta las regiones de una imagen que más influyen en la decisión tomada por el modelo durante el proceso de clasificación.

**SNR** sigla de *Signal-to-Noise Ratio* o relación señal a ruido; es una medida que compara la potencia de la señal útil con la potencia del ruido presente.

**Transfer learning** estrategia en la que se aprovechan modelos previamente entrenados en grandes conjuntos de datos para adaptarlos a una nueva tarea.

## Resumen

**Título:** Modelo de aprendizaje profundo para la clasificación automática de escenas en imágenes satelitales [*]

**Autor:** Kmila Andrea Contreras Amaya y Mónica Cristina Peña Rincón [*]

**Palabras Clave:** aprendizaje profundo, imágenes satelitales, clasificación de escenas, redes neuronales convolucionales, UC Merced, robustez, mapas de saliencia, reproducibilidad.

**Descripción:** Este trabajo de grado presenta la implementación y evaluación de un modelo de inteligencia artificial basado en aprendizaje profundo para la clasificación automática de escenas en imágenes satelitales. El estudio se desarrolló utilizando el conjunto de datos UC Merced Land Use, compuesto por 21 categorías de escenas, con imágenes organizadas y etiquetadas para tareas de clasificación supervisada multiclase. La metodología incluyó la verificación del conjunto de datos, la carga de imágenes en formato RGB, la normalización de intensidades, la codificación de etiquetas mediante representación *one-hot* y la partición estratificada en subconjuntos de entrenamiento, validación y prueba.

Posteriormente, se implementó una arquitectura basada en redes neuronales convolucionales, entrenada mediante estrategias de regularización, ajuste de hiperparámetros y monitoreo del desempeño. El modelo fue evaluado mediante métricas como exactitud, precisión, *recall*, F1-score y matrices de confusión. Adicionalmente, se analizaron escenarios de robustez ante perturbaciones controladas, incluyendo reducción de resolución, ruido AWGN y desenfoque mediante *kernels* promedio. También se evaluó una adaptación del modelo DSFATN reportado en la literatura, con el fin de contrastar el desempeño del modelo base frente a una estrategia robusta. Finalmente, se incorporaron mapas de saliencia para apoyar la interpretación cualitativa de las regiones relevantes utilizadas por el clasificador. Como complemento, se consolidó un repositorio en GitHub para favorecer la trazabilidad y reproducibilidad del experimento.

[*]Trabajo de grado

[*]Facultad de Ingenierías Físico-Mecánicas. Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones. Programa académico: Ingeniería Electrónica. Director: Hans Yecid García Arenas. Ingeniero Electrónico y Doctor en Ingeniería.

## Abstract

**Title:** Deep learning model for the automatic classification of scenes in satellite images [*]

**Author:** Kmila Andrea Contreras Amaya and Mónica Cristina Peña Rincón [*]

**Keywords:** deep learning, satellite images, scene classification, convolutional neural networks, UC Merced, robustness, saliency maps, reproducibility.

**Description:** This degree work presents the implementation and evaluation of an artificial intelligence model based on deep learning for the automatic classification of scenes in satellite images. The study was developed using the UC Merced Land Use dataset, composed of 21 scene categories, with images organized and labeled for supervised multiclass classification tasks. The methodology included dataset verification, RGB image loading, intensity normalization, label encoding using a *one-hot* representation, and stratified partitioning into training, validation, and test subsets.

Subsequently, a convolutional neural network architecture was implemented and trained using regularization strategies, hyperparameter adjustment, and performance monitoring. The model was evaluated through metrics such as accuracy, precision, *recall*, F1-score, and confusion matrices. In addition, robustness scenarios were analyzed under controlled perturbations, including resolution reduction, AWGN noise, and blurring using average *kernels*. An adaptation of the DSFATN model reported in the literature was also evaluated to compare the base model against a robust strategy. Finally, saliency maps were incorporated to support the qualitative interpretation of the relevant regions used by the classifier. As a complementary contribution, a GitHub repository was consolidated to promote traceability and reproducibility of the experimental workflow.

[*]Degree Work

[*]Faculty of Physical-Mechanical Engineering. School of Electrical, Electronics and Telecommunications Engineering. Academic Program: Electronic Engineering. Advisor: Hans Yecid García Arenas. Electronic Engineer and PhD in Engineering.

# Introducción

## Contexto: teledetección e imágenes satelitales

La teledetección satelital se ha consolidado como una tecnología habilitadora para la observación y el análisis de la superficie terrestre a gran escala. Su amplia cobertura espacial, periodicidad de captura y niveles de resolución crecientes han impulsado el uso de imágenes satelitales en aplicaciones como monitoreo ambiental, planificación territorial y análisis geoespacial. Sin embargo, el aumento sostenido en la disponibilidad de datos plantea un desafío operativo: la interpretación manual de grandes volúmenes de imágenes resulta lenta, costosa y difícil de escalar, especialmente cuando se requieren resultados consistentes y reproducibles.

En este contexto, la *clasificación automática de escenas* consiste en asignar a cada imagen una etiqueta semántica asociada al uso o la cobertura del suelo (p. ej., zonas urbanas, áreas agrícolas o cuerpos de agua). Mientras que los enfoques tradicionales dependían de características diseñadas manualmente y clasificadores convencionales, las redes neuronales convolucionales (CNN) permiten implementar modelos de aprendizaje profundo capaces de aprender representaciones discriminativas para distinguir la *escena* como un todo, aprovechando patrones espaciales y texturales presentes en la imagen Nogueira, Penatti, y dos Santos (2017). Esta capacidad es especialmente relevante cuando existen clases visualmente similares o una alta variabilidad dentro de una misma categoría Cole (2018).

### Planteamiento y formulación del problema

A pesar del potencial de las CNN, la clasificación de escenas satelitales presenta desafíos inherentes a la naturaleza del dato visual. En escenarios multiclase, es común observar: (i) **variabilidad intra–clase**, donde imágenes de una misma categoría difieren significativamente en estructura espacial, textura y apariencia; y (ii) **similitud inter–clase**, donde categorías distintas comparten rasgos visuales que incrementan la probabilidad de confusión Nogueira y cols. (2017). Estas condiciones hacen necesario evaluar, bajo un protocolo controlado, si un modelo de aprendizaje profundo basado en CNN logra separar de manera fiable múltiples categorías de uso y cobertura del suelo.

En consecuencia, la pregunta de investigación que guía este trabajo es: *¿en qué medida un modelo de aprendizaje profundo basado en CNN puede clasificar automáticamente escenas en imágenes satelitales en un escenario multiclase y cómo se caracteriza su desempeño mediante métricas estándar?*

Como hipótesis de trabajo, se plantea que el modelo alcance una exactitud (*accuracy*) igual o superior al 90 % bajo el protocolo experimental definido. Para caracterizar el comportamiento del clasificador más allá de la exactitud global, la evaluación se complementa con métricas estándar como precisión y matriz de confusión, permitiendo analizar el desempeño por categoría y reconocer patrones de error en el conjunto de clases Nogueira y cols. (2017).

### Justificación, motivación e importancia del trabajo

Este proyecto se justifica por su relevancia científica y práctica. En primer lugar, la clasificación de escenas satelitales es un problema vigente en teledetección cuyo desempeño depende

de decisiones críticas en la preparación de datos, la arquitectura del modelo y la configuración del entrenamiento. La literatura reporta que los modelos basados en CNN, entrenados bajo protocolos adecuados, pueden mejorar el rendimiento frente a enfoques tradicionales en tareas de reconocimiento de escenas Cole (2018); Nogueira y cols. (2017).

En segundo lugar, desde una perspectiva de ingeniería, es valioso estructurar un flujo experimental reproducible que documente de forma verificable las decisiones de implementación: preparación y particionado de la base de datos, definición del modelo multiclase, ajuste de parámetros de entrenamiento y evaluación con métricas estándar. Para favorecer la comparabilidad con trabajos previos, el estudio se apoya en un conjunto de datos de referencia ampliamente utilizado, el *UC Merced Land Use Dataset* Yang y Newsam (2010), el cual facilita la replicación y contrastación de resultados en la literatura.

A partir de lo anterior, este trabajo adquiere importancia al integrar, en un mismo proceso experimental, el desarrollo e implementación de un modelo de aprendizaje profundo basado en redes neuronales convolucionales (CNN) para la clasificación multiclase de escenas satelitales, el ajuste de sus parámetros de entrenamiento para optimizar su desempeño y la evaluación mediante métricas estándar que permitan caracterizar su comportamiento global y por clase, en coherencia con el objetivo general y los objetivos específicos formulados en el Capítulo 1.

### Alcance, limitaciones y restricciones del estudio

***Alcance.*** El alcance de esta investigación comprende la preparación de una base de datos de imágenes satelitales etiquetadas por escenas y su adecuación para el entrenamiento y validación de un modelo de aprendizaje profundo. Posteriormente, se implementa una arquitectura basada en CNN con capacidad de clasificar múltiples categorías de escenas, y se entrena ajustando parámetros

del proceso de aprendizaje para optimizar el desempeño. Finalmente, el modelo se evalúa mediante métricas estándar de clasificación, incluyendo exactitud, precisión y matriz de confusión, con el propósito de caracterizar su comportamiento en cada categoría analizada. La fase experimental se realiza sobre el *UC Merced Land Use Dataset* Yang y Newsam (2010).

***Limitaciones.*** Los resultados se limitan al conjunto de datos y al protocolo experimental definido; por tanto, la generalización a otras fuentes satelitales, resoluciones, condiciones de captura o regiones geográficas requiere validaciones adicionales. Asimismo, el estudio se desarrolla en un entorno controlado y no contempla despliegue en tiempo real ni integración directa con sistemas productivos.

***Restricciones.*** Las decisiones prácticas de entrenamiento (p. ej., tamaño de lote, número máximo de épocas y alcance del ajuste de hiperparámetros) están condicionadas por la disponibilidad de recursos computacionales en entornos locales y/o en la nube, lo cual limita la exploración exhaustiva del espacio de configuraciones.

Con base en lo anterior, este trabajo se orienta a desarrollar, implementar, entrenar y evaluar un modelo de aprendizaje profundo basado en CNN para clasificación automática de escenas satelitales en un escenario multiclase, utilizando el dataset UC Merced y métricas estándar para caracterizar el desempeño del modelo. En coherencia con este planteamiento, en el Capítulo 2 se presentan el objetivo general y los objetivos específicos que guían el desarrollo metodológico del documento.

# 1. Objetivos

## 1.1. Objetivo general

Implementar un modelo de inteligencia artificial, fundamentado en técnicas de aprendizaje profundo, para la clasificación automática de escenas en imágenes satelitales.

## 1.2. Objetivos específicos

Preparar una base de datos de imágenes satelitales existentes, etiquetadas por escenas, para su uso en el entrenamiento del modelo.

Implementar un modelo de aprendizaje profundo con capacidad de reconocer y clasificar múltiples categorías de escenas satelitales.

Entrenar el modelo de aprendizaje profundo implementado utilizando la base de datos de imágenes satelitales construida.

Evaluar el desempeño del modelo mediante métricas de clasificación como exactitud, precisión y matriz de confusión.

# 2. Conceptos previos

En este capítulo se presentan los conceptos previos necesarios para comprender el problema de clasificación automática de escenas en imágenes satelitales y las decisiones metodológicas adoptadas en esta tesis. En particular, se describe la formulación de la tarea de clasificación de escenas, el conjunto de datos de referencia utilizado y los fundamentos esenciales de las redes neuronales convolucionales y de las métricas empleadas para evaluar el desempeño del modelo. Los desarrollos teóricos complementarios, así como la revisión ampliada de algunos temas específicos, se presentan en los anexos del documento.

## 2.1. Clasificación de escenas en imágenes satelitales

La clasificación de imágenes consiste en asignar una etiqueta a una imagen completa a partir de su contenido visual. En el contexto de la teledetección, esta formulación se diferencia de la segmentación semántica, en la cual la etiqueta se asigna a nivel de píxel. En esta tesis se adopta un esquema de clasificación de etiqueta única por imagen (*single-label image classification*), donde cada recorte satelital se asocia con una sola categoría representativa de la escena (Cole, 2018).

En imágenes satelitales, este tipo de problema suele presentar dificultades asociadas a la variabilidad intraclase, la similitud interclase y la presencia de patrones espaciales complejos. Por ello, resulta pertinente emplear modelos capaces de aprender representaciones discriminativas de alto nivel. En este trabajo, la clasificación de escenas se aborda como una tarea multiclase sobre un conjunto de datos estandarizado, con el fin de garantizar comparabilidad y reproducibilidad experimental.

## 2.2. Dataset UC Merced

El *UC Merced Land Use Dataset* es uno de los conjuntos de datos de referencia más utilizados en clasificación de escenas de teledetección con imágenes RGB (Cheng, Xie, Han, Guo, y Xia, 2020; Tombe y Viriri, 2023; Yang y Newsam, 2010). El conjunto está conformado por 21 categorías de uso o cobertura del suelo, con 100 imágenes por clase, para un total de 2100 muestras. Cada imagen corresponde, en su versión más difundida, a un recorte de $256 \times 256$ píxeles (Yang y Newsam, 2010).

La adopción de este conjunto de datos en la presente tesis se justifica por tres razones principales: su carácter público, su uso extendido como *benchmark* en la literatura especializada y la posibilidad de contrastar resultados bajo un protocolo experimental conocido. En consecuencia, UC Merced constituye una base adecuada para evaluar el comportamiento del modelo propuesto en una tarea multiclase de clasificación de escenas.

## 2.3. Fundamentos esenciales de redes neuronales convolucionales

Las redes neuronales convolucionales (CNN) son modelos de aprendizaje profundo diseñados para extraer automáticamente patrones espaciales a partir de imágenes. Su funcionamiento se basa en la aplicación sucesiva de filtros convolucionales que permiten detectar desde características simples, como bordes o texturas, hasta representaciones más abstractas en niveles profundos (Keras Team, s.f.-a; Zhao y cols., 2024).

De manera general, una CNN combina capas de convolución, funciones de activación no lineales y mecanismos de reducción espacial. Las capas convolucionales permiten construir mapas de características; las activaciones, como ReLU, introducen no linealidad; y operaciones como *max*

*pooling* reducen la dimensión espacial de las representaciones, favoreciendo una extracción más compacta de información relevante (Keras Team, s.f.-b; TensorFlow Developers, s.f.-c).

Para tareas multiclase, la capa de salida suele emplear activación *softmax*, la cual transforma las salidas del modelo en una distribución de probabilidad sobre las clases. En esta tesis, estos elementos constituyen la base conceptual del clasificador implementado para reconocer escenas satelitales.

### 2.4. Regularización y generalización del modelo

En problemas de clasificación con conjuntos de datos relativamente limitados, como UC Merced, es importante incorporar mecanismos que favorezcan la generalización del modelo. Entre las estrategias más utilizadas se encuentran *batch normalization*, *dropout* y el aumento de datos (*data augmentation*) (Hao y cols., 2023; Ioffe y Szegedy, 2015; Srivastava, Hinton, Krizhevsky, Sutskever, y Salakhutdinov, 2014).

*Batch normalization* contribuye a estabilizar el entrenamiento al normalizar activaciones intermedias, mientras que *dropout* reduce el sobreajuste al desactivar aleatoriamente parte de las unidades durante la fase de entrenamiento. Por su parte, el aumento de datos introduce transformaciones controladas sobre las imágenes de entrada, incrementando la diversidad efectiva del conjunto de entrenamiento sin modificar su etiqueta semántica.

En esta tesis, la regularización se asume como un componente central del diseño experimental, ya que busca mejorar la capacidad de generalización del clasificador frente a variaciones espaciales y texturales presentes en las escenas.

## 2.5. Métricas de evaluación en clasificación multiclase

La evaluación del modelo requiere métricas que permitan analizar su desempeño global y por clase. En este trabajo se emplean la exactitud (*accuracy*), la precisión, el *recall*, el *F1-score* y la matriz de confusión (scikit-learn developers, s.f.-a; Stout, 2025).

La exactitud permite cuantificar la proporción total de predicciones correctas, mientras que la precisión, el *recall* y el *F1-score* ofrecen una visión más detallada del comportamiento del clasificador en cada categoría. Complementariamente, la matriz de confusión permite identificar patrones de error y clases que tienden a confundirse entre sí.

Estas métricas sustentan el protocolo de evaluación adoptado en la tesis y permiten interpretar el desempeño del modelo más allá de un único indicador global.

# 3. Preprocesamiento y preparación del conjunto de datos

Este capítulo describe el proceso seguido para preparar el conjunto de datos utilizado en la clasificación multiclase de escenas satelitales. En correspondencia con el primer objetivo específico de la tesis, se presentan la organización del *dataset*, el preprocesamiento aplicado a las imágenes, la estrategia de partición en subconjuntos de entrenamiento, validación y prueba, y la codificación de etiquetas requerida para el entrenamiento supervisado del modelo.

## 3.1. Descripción general del flujo de preparación de datos

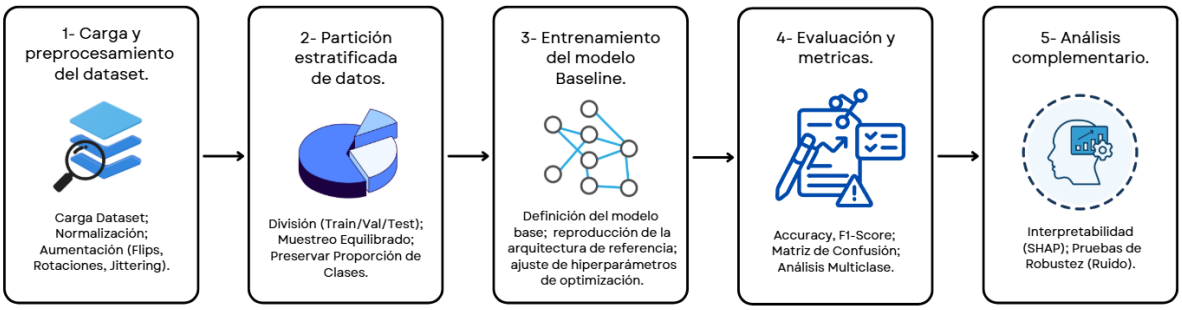
La preparación del conjunto de datos se realizó como una etapa previa al diseño, entrenamiento y evaluación del modelo. De forma general, el procedimiento consistió en verificar la estructura del *dataset* y la correspondencia entre carpetas y clases, cargar las imágenes en formato RGB conservando su tamaño original, normalizar los valores de intensidad y efectuar una partición estratificada en subconjuntos de entrenamiento, validación y prueba. La Figura **??** resume el flujo seguido en esta etapa.

## 3.2. Organización del conjunto de datos y mapeo de clases

El conjunto de datos se organizó siguiendo la estructura estándar *folder-per-class*, en la cual cada categoría se almacena en una carpeta independiente con sus respectivas imágenes. Para mantener consistencia entre el entrenamiento y el análisis posterior de resultados, las carpetas se recorrieron en orden alfabético y se construyó un diccionario de clases que establece el mapeo entre índice numérico y etiqueta semántica. Este mapeo se utilizó posteriormente para interpretar las salidas del modelo, generar matrices de confusión y elaborar reportes por clase. Para facili-

**Figura 1.**

*Flujo metodológico general implementado en el estudio.*




*Nota.* La figura sintetiza el proceso metodológico desarrollado en el trabajo, desde la carga y preprocesamiento del conjunto de datos hasta la partición estratificada, el entrenamiento del modelo, la evaluación mediante métricas estándar y el análisis complementario de interpretabilidad y robustez.

tar la comprensión visual de las categorías consideradas, en el Anexo C se presentan ejemplos representativos de las clases incluidas en el conjunto de datos.

### 3.3. Preprocesamiento aplicado a las imágenes

Las imágenes del conjunto de datos fueron cargadas en formato RGB, conservando su tamaño original con el fin de mantener la resolución nativa del *dataset* y la información espacial disponible en cada muestra. Posteriormente, los valores de intensidad de cada píxel se normalizaron al intervalo $[0, 1]$ mediante división por 255, y los arreglos de imagen se convirtieron al tipo de dato `float32`. Este preprocesamiento se aplicó de forma consistente a todos los subconjuntos, garantizando un mismo esquema de representación de entrada en entrenamiento, validación y prueba.

### 3.4. Partición estratificada del conjunto de datos

Con el propósito de asegurar una distribución balanceada de clases en cada subconjunto, se definió una partición estratificada del conjunto de datos en entrenamiento, validación y prueba, con proporciones de 80 %, 10 % y 10 %, respectivamente. Para garantizar reproducibilidad en la separación de muestras, se empleó `random_state=42`. El subconjunto de entrenamiento se destinó al ajuste de los parámetros del modelo, el de validación al monitoreo del proceso de entrenamiento y el de prueba al reporte final del desempeño del clasificador. Como resultado de esta partición, se obtuvieron 1680 imágenes para entrenamiento, 210 imágenes para validación y 210 imágenes para prueba.

**Tabla 1.**

*Resumen de la partición estratificada del conjunto de datos.*

| Subconjunto | Proporción | Número de imágenes | Uso en el estudio |
|---|---|---|---|
| Entrenamiento | 80 % | 1680 | Ajuste de parámetros del modelo |
| Validación | 10 % | 210 | Monitoreo del proceso de entrenamiento |
| Prueba | 10 % | 210 | Evaluación final del modelo |

### 3.5. Codificación de etiquetas para clasificación multiclase

Dado que el problema abordado en esta tesis corresponde a una tarea de clasificación multiclase con 21 categorías, las etiquetas se transformaron a codificación *one-hot*. En esta representación, cada muestra se asocia con un vector binario de dimensión $C = 21$, en el que únicamente la posición correspondiente a la clase verdadera toma valor uno, mientras que las restantes toman valor cero. Esta codificación permitió adaptar las salidas del conjunto de datos a la formulación

supervisada empleada durante el entrenamiento del modelo, manteniendo coherencia con la capa de salida multiclase y la función de pérdida utilizada posteriormente.

# 4. Diseño e implementación del modelo de aprendizaje profundo

Este capítulo presenta el diseño e implementación del modelo de aprendizaje profundo utilizado para la clasificación multiclase de escenas en imágenes satelitales. En correspondencia con el segundo objetivo específico de la tesis, se describe la arquitectura de referencia reproducida y la propuesta final desarrollada en este trabajo. La representación gráfica del *baseline* y la comparación estructural detallada entre ambas arquitecturas se presentan en el Anexo B.

## 4.1. Criterios generales de diseño del modelo

El diseño del clasificador se planteó bajo una formulación de clasificación supervisada multiclase, coherente con la naturaleza del conjunto de datos utilizado. Dado que el problema consiste en asignar una única etiqueta entre 21 categorías posibles a cada imagen de entrada, la arquitectura se definió con una capa de salida de 21 neuronas y activación *softmax*.

Como criterio general, se buscó una arquitectura capaz de extraer características espaciales discriminativas de las imágenes satelitales, manteniendo al mismo tiempo una complejidad razonable frente al tamaño del conjunto de datos. Para ello, se tomó como punto de partida un modelo convolucional secuencial de referencia y, a partir de él, se introdujeron ajustes orientados a mejorar la capacidad de generalización y la estabilidad del entrenamiento.

## 4.2. Baseline reproducido

Como referencia metodológica se reprodujo un *baseline* a partir de una implementación pública disponible en GitHub, específicamente el cuaderno `LandCoverDetection.ipynb`. Dentro de dicha referencia se adoptó el denominado **Modelo 3**, utilizado en este trabajo como arquitectura

base de comparación.

La reproducción del modelo base mantuvo como entrada imágenes de dimensión $256 \times 256 \times 3$ y una salida multiclase de 21 neuronas con activación *softmax*. Asimismo, se adaptó la lectura del conjunto de datos a la organización local definida en el capítulo anterior y se mantuvo un protocolo homogéneo de partición de datos para permitir comparaciones posteriores bajo las mismas condiciones experimentales.

De forma resumida, el modelo de referencia corresponde a una red neuronal convolucional secuencial compuesta por tres bloques convolucionales con filtros $32/64/128$, seguida de una etapa de aplanamiento mediante `Flatten` y una cabeza densa con capas de 256 y 128 neuronas. El modelo también incorpora `BatchNormalization` y `Dropout`.

### 4.3. Modelo propuesto

A partir del *baseline* reproducido, se implementó un modelo final orientado a mejorar la capacidad de representación del clasificador y reducir su sensibilidad al sobreajuste. La estrategia adoptada consistió en refinar los componentes principales de la arquitectura base mediante ajustes en profundidad efectiva, regularización y agregación de características.

En primer lugar, se incrementó el número de filtros en los bloques convolucionales, pasando de una configuración $32/64/128$ a $64/128/256$, con el fin de ampliar la capacidad del modelo para capturar patrones espaciales y texturales de mayor complejidad.

En segundo lugar, se reemplazó la capa `Flatten` por `GlobalAveragePooling2D`, lo que permitió reducir el número de parámetros en la transición hacia la etapa de clasificación y favorecer una representación más compacta de las características extraídas.

Adicionalmente, se incorporó regularización L2 tanto en capas convolucionales como den-

sas. En las capas convolucionales se utilizó $\lambda = 5 \times 10^{-4}$, mientras que en la parte densa se empleó $\lambda = 10^{-3}$. Asimismo, se ajustaron las tasas de *dropout*, utilizando valores de 0.2 en la parte convolucional y 0.3 en la etapa densa.

En conjunto, estas modificaciones dieron lugar a una arquitectura que conserva la lógica general del *baseline*, pero incorpora decisiones orientadas a mejorar la estabilidad del modelo y su capacidad de generalización sobre datos no vistos.

### 4.4. Comparación estructural y selección final

La comparación entre el *baseline* reproducido y el modelo propuesto permite identificar con claridad las decisiones de diseño introducidas como aporte metodológico de esta tesis. Aunque la arquitectura final emplea un mayor número de filtros en la etapa convolucional, el reemplazo de `Flatten` por `GlobalAveragePooling2D` reduce de forma importante el número total de parámetros, lo que mejora la eficiencia estructural del clasificador.

En consecuencia, el modelo propuesto combina una mayor capacidad de extracción de características con una estrategia más eficiente de agregación y regularización. Esta configuración se adoptó como base experimental del desarrollo principal de la tesis, y sobre ella se realizaron los análisis de entrenamiento, evaluación e interpretabilidad presentados en los capítulos posteriores.

La representación gráfica del *baseline* y la comparación estructural detallada entre ambas arquitecturas se incluyen en el Anexo B.

# 5. Entrenamiento, optimización y reproducibilidad del modelo

Este capítulo describe el proceso seguido para entrenar el modelo de aprendizaje profundo propuesto, así como las estrategias de optimización, control y reproducibilidad empleadas durante esta etapa. En correspondencia con el tercer objetivo específico de la tesis, se presentan la formulación del entrenamiento supervisado, la configuración del optimizador y la función de pérdida, la aumentación de datos en línea, los mecanismos de control de convergencia y el entorno de ejecución utilizado.

## 5.1. Formulación del proceso de entrenamiento

El entrenamiento del clasificador se planteó como un problema de aprendizaje supervisado multiclase, coherente con la naturaleza del conjunto de datos y con la capa de salida definida en el capítulo anterior. En este contexto, el modelo recibe como entrada imágenes satelitales preprocesadas y produce una distribución de probabilidad sobre las 21 categorías de escena consideradas en el estudio.

El proceso de entrenamiento tuvo como propósito ajustar los parámetros de la red neuronal convolucional de manera que la diferencia entre las predicciones del modelo y las etiquetas reales fuera progresivamente minimizada. Para ello, se utilizó el subconjunto de entrenamiento definido previamente, mientras que el subconjunto de validación se reservó para monitorear el comportamiento del modelo durante el ajuste y apoyar las decisiones de control del entrenamiento.

### 5.2. Configuración del optimizador y función de pérdida

La configuración adoptada para el entrenamiento del modelo se definió de acuerdo con la formulación multiclase del problema. Se empleó el optimizador Adam con una tasa de aprendizaje inicial de $\alpha = 10^{-3}$, mientras que la función de pérdida utilizada fue la entropía cruzada categórica, apropiada para tareas de clasificación supervisada con etiquetas codificadas en formato *one-hot*. La métrica monitoreada durante el ajuste fue la exactitud (*accuracy*), utilizada como indicador del desempeño del modelo sobre los subconjuntos de entrenamiento y validación.

Los principales hiperparámetros de entrenamiento fueron: optimizador Adam con learning_rate=0.001, función de pérdida `categorical_crossentropy`, métrica de seguimiento `accuracy`, tamaño de lote de 32 y un máximo de 500 épocas. Esta configuración constituyó el punto de partida del ajuste del modelo, mientras que la estabilidad y convergencia efectiva dependieron además de las estrategias de regularización y control descritas en las secciones siguientes.

### 5.3. Aumentación de datos en línea

Con el fin de incrementar la variabilidad efectiva del conjunto de entrenamiento y reducir el riesgo de sobreajuste, se aplicó una estrategia de aumentación de datos en línea mediante `ImageDataGenerator`. A diferencia del preprocesamiento fijo descrito en el Capítulo 3, estas transformaciones no se almacenaron como nuevas imágenes, sino que fueron generadas dinámicamente durante el proceso de entrenamiento.

Las transformaciones implementadas fueron de naturaleza geométrica y buscaron preservar la semántica global de la escena, introduciendo al mismo tiempo variaciones controladas de orientación, posición y escala. Los parámetros finales empleados fueron: `rotation_range=20`,

`width_shift_range=0.2`, `height_shift_range=0.2`, `horizontal_flip=True`, `zoom_range=0.2`, `shear_range=0.2` y `fill_mode='nearest'`. La adopción de este esquema permitió que el modelo observara variaciones de una misma escena a lo largo de las distintas épocas, favoreciendo una mayor robustez frente a cambios moderados de orientación y encuadre en las imágenes satelitales.

### 5.4. Estrategias de control y optimización del entrenamiento

Con el propósito de controlar el sobreajuste y mejorar la estabilidad de la convergencia, se implementaron mecanismos de monitoreo sobre el conjunto de validación utilizando `val_loss` como criterio principal de seguimiento. En particular, se emplearon dos *callbacks*: `EarlyStopping`, configurado con `patience=25` y `restore_best_weights=True`, y `ReduceLROnPlateau`, con `factor=0.5`, `patience=12` y `min_lr=1e-5`. En conjunto, estas estrategias permitieron detener el entrenamiento cuando el proceso dejaba de mejorar y reducir automáticamente la tasa de aprendizaje en fases de estancamiento, favoreciendo un ajuste más estable y eficiente.

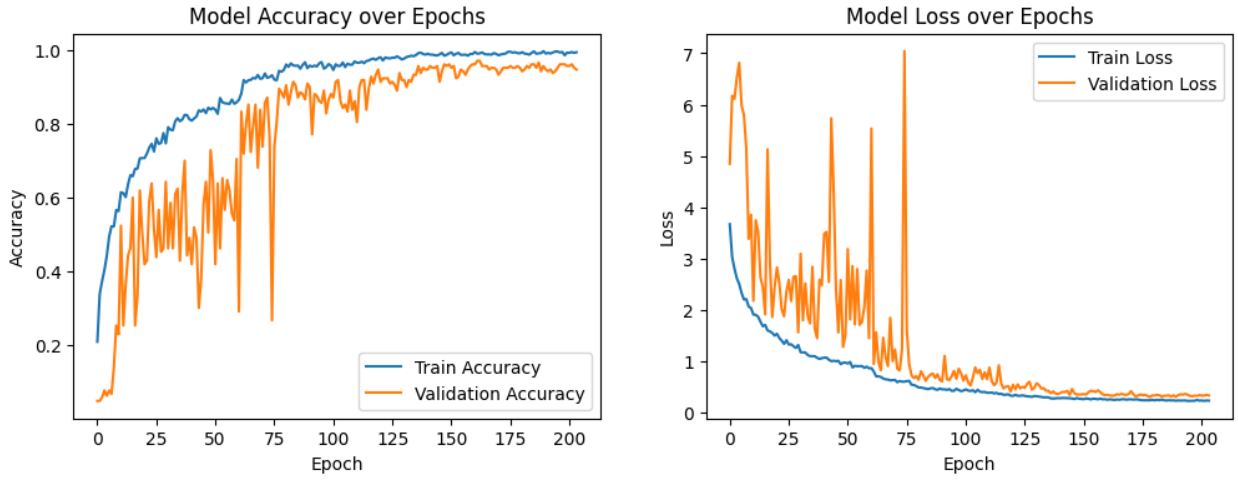
### 5.5. Curvas de entrenamiento y criterio de convergencia

Aunque el número máximo de épocas se fijó en 500, el entrenamiento efectivo del modelo propuesto se detuvo en la época 204 mediante *early stopping*, lo que indica que el ajuste alcanzó convergencia antes de agotar el límite inicialmente establecido.

La Figura 2 muestra la evolución de la exactitud y la pérdida en entrenamiento y validación. En términos generales, las curvas de entrenamiento presentan una tendencia favorable: la exactitud aumenta globalmente y la pérdida disminuye. Sin embargo, en validación se observan oscilaciones importantes y picos abruptos, especialmente durante las primeras épocas y con mayor notoriedad aproximadamente entre las épocas 40 y 80, lo que evidencia que el proceso no fue completamente estable en toda la fase de ajuste.

**Figura 2.**

*Curvas de entrenamiento del modelo propuesto.*




*Nota.* La figura presenta la evolución de la exactitud y la pérdida durante el entrenamiento y la validación del modelo propuesto. Estas curvas permiten analizar el comportamiento del aprendizaje, la convergencia del modelo y la posible presencia de sobreajuste.

Estos episodios de inestabilidad pueden asociarse con la complejidad de la tarea, el tamaño del conjunto de datos, la variabilidad introducida por la aumentación en línea y la alta capacidad de representación de la arquitectura. Además, dado que el entrenamiento incorporó el *callback* `ReduceLROnPlateau`, es razonable interpretar que la estabilización observada en etapas posteriores estuvo favorecida por la reducción progresiva de la tasa de aprendizaje, aunque la figura por sí sola no permite identificar con exactitud las épocas específicas de cada ajuste.

A partir de aproximadamente la época 80, las curvas de validación muestran un comportamiento más regular y una tendencia global más consistente, lo que sugiere una fase de ajuste más controlada. En consecuencia, la figura respalda una convergencia global favorable del modelo propuesto, aunque precedida por una etapa inicial e intermedia de inestabilidad que debe reconocerse explícitamente.

### 5.6. Entorno de ejecución

El proceso de entrenamiento se llevó a cabo en el entorno de Google Colab, mientras que el conjunto de datos fue accedido desde Google Drive mediante el montaje del directorio correspondiente. Esta configuración permitió disponer de un entorno flexible y de fácil acceso para la experimentación y el entrenamiento del modelo.

Dado que Google Colab proporciona recursos de hardware y software que pueden variar entre sesiones, se procedió a registrar las características específicas del entorno utilizadas durante la ejecución de los experimentos. En particular, el entrenamiento se realizó utilizando un acelerador gráfico (GPU) NVIDIA Tesla T4, con una memoria dedicada aproximada de 13.6 GB. Asimismo, el entorno contó con aproximadamente 7.7 GB de memoria RAM del sistema y 43.4 GB de almacenamiento en disco.

En cuanto al entorno de software, la ejecución se realizó sobre Python 3.12.13. Para la implementación del modelo se utilizó TensorFlow 2.19.0 junto con su API integrada Keras (versión 3.13.2). Adicionalmente, se emplearon bibliotecas complementarias como NumPy 2.0.2, scikit-learn, Matplotlib y Seaborn para el procesamiento de datos, evaluación y visualización de resultados.

### 5.7. Reproducibilidad del experimento

Con el fin de garantizar la trazabilidad y reproducibilidad del trabajo, se consolidó un repositorio en GitHub que integra los recursos del desarrollo experimental, incluyendo preparación de datos, implementación de modelos, entrenamiento, evaluación, análisis de robustez y generación de resultados. El enlace y la descripción general se presentan en el Anexo L.

# 6. Resultados y evaluación del desempeño del modelo

Este capítulo presenta los resultados obtenidos por el modelo de aprendizaje profundo propuesto y desarrolla la evaluación cuantitativa de su desempeño en la tarea de clasificación multiclase de escenas satelitales. En correspondencia con el cuarto objetivo específico de la tesis, se reportan los resultados alcanzados en los subconjuntos de entrenamiento, validación y prueba, así como el análisis mediante matrices de confusión y métricas globales de desempeño.

A diferencia de los capítulos previos, centrados en la preparación del conjunto de datos, el diseño arquitectónico y el proceso de entrenamiento, en este capítulo el énfasis se sitúa en la interpretación de los resultados obtenidos por el clasificador bajo el protocolo experimental definido.

## 6.1. Protocolo de evaluación del modelo

La evaluación del modelo se realizó sobre los subconjuntos de entrenamiento, validación y prueba definidos mediante partición estratificada 80–10–10. La métrica principal utilizada para el análisis global fue la exactitud (*accuracy*), complementada con matrices de confusión y métricas por clase, específicamente precisión, *recall* y F1-score bajo el esquema *one-vs-rest*.

Dado que el conjunto de datos presenta una distribución balanceada entre categorías, también se consideraron promedios globales de tipo *macro* y *weighted*, con el propósito de ofrecer una caracterización más completa del comportamiento multiclase del modelo.

### 6.2. Desempeño global en entrenamiento, validación y prueba

Bajo la configuración final reportada en esta tesis, el modelo alcanzó una exactitud de 100.00 % sobre el conjunto de entrenamiento, 95.71 % en validación y 94.29 % en prueba. Estos resultados muestran que la red logró un ajuste muy alto sobre los datos de entrenamiento y, al mismo tiempo, mantuvo un desempeño elevado sobre datos no vistos previamente.

En términos de pérdida, se obtuvieron valores de 0.2163 en entrenamiento, 0.3056 en validación y 0.4536 en prueba. Aunque la pérdida en prueba es superior a la observada en entrenamiento y validación, el nivel de exactitud alcanzado sugiere que el modelo conserva una capacidad de generalización adecuada para la tarea planteada.

**Tabla 2.**

*Resultados globales del modelo en entrenamiento, validación y prueba.*

| **Subconjunto** | **Accuracy ( %)** | **Loss** |
|---|---|---|
| Entrenamiento | 100.00 | 0.2163 |
| Validación | 95.71 | 0.3056 |
| Prueba | 94.29 | 0.4536 |

De manera general, la diferencia entre el desempeño en entrenamiento y los resultados obtenidos en validación y prueba indica que, aunque el modelo presenta un ajuste muy alto sobre las muestras vistas durante el aprendizaje, su comportamiento sobre datos no observados sigue siendo sólido y consistente con el objetivo de clasificación multiclase planteado.

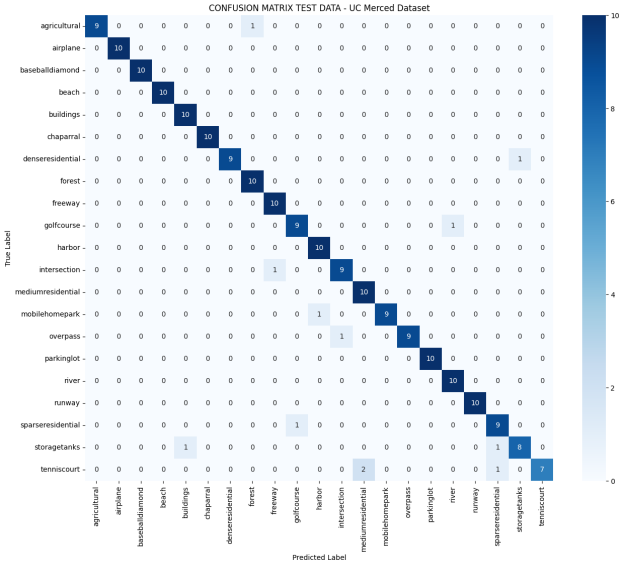
### 6.3. Análisis mediante matrices de confusión

Con el fin de examinar el comportamiento del clasificador más allá de la exactitud global, se construyeron matrices de confusión comparando las etiquetas reales con las etiquetas predichas por el modelo. En este trabajo, la matriz de confusión del conjunto de prueba se considera el resultado principal de evaluación, mientras que las matrices correspondientes a entrenamiento y validación se utilizan como apoyo para diagnosticar el ajuste alcanzado por la red.

La Figura 3 presenta la matriz de confusión del conjunto de prueba. Su análisis permite identificar tanto las clases correctamente diferenciadas como las confusiones residuales entre categorías visualmente cercanas.

**Figura 3.**

*Matriz de confusión del modelo final.*




*Nota.* La figura presenta la matriz de confusión obtenida por el modelo final sobre el conjunto de prueba. En ella se comparan las clases reales con las clases predichas, lo que permite identificar los aciertos de clasificación en la diagonal principal y los errores entre clases fuera de dicha diagonal.

De manera complementaria, las matrices de confusión correspondientes a los subconjuntos de validación y entrenamiento se presentan en el Anexo J. Estas permiten contrastar el comportamiento del modelo en distintas etapas del proceso de aprendizaje y analizar la consistencia de las predicciones.

En términos generales, dichas matrices evidencian un comportamiento favorable, con predominio de valores en la diagonal principal, lo que indica que el clasificador logró distinguir correctamente la mayoría de las categorías del problema.

### 6.4. Síntesis de métricas por clase y promedios globales

Con el propósito de analizar el desempeño final del modelo a nivel de categoría, se calculó el `classification_report` sobre el conjunto de prueba. Los resultados muestran un comportamiento global favorable, con una exactitud de 0.94 y promedios *macro* y *weighted* de 0.94 en F1-score.

En términos generales, varias clases alcanzaron métricas perfectas o casi perfectas, como *agricultural*, *airplane*, *beach*, *freeway*, *golfcourse*, *harbor*, *mobilehomepark*, *parkinglot*, *runway* y *tenniscourt*, lo cual evidencia que el modelo logra discriminar con alta confiabilidad escenas con patrones visuales claramente distintivos.

No obstante, el desempeño no fue completamente uniforme entre todas las categorías. Las principales reducciones en *recall* se observaron en *river*, *baseballdiamond* y *sparseresidential*, mientras que las menores precisiones se presentaron en *buildings*, *forest* y *storagetanks*. Esto sugiere que los errores residuales tienden a concentrarse en un subconjunto reducido de clases con mayor ambigüedad visual o similitud interclase.

La tabla detallada de métricas por clase del modelo en el conjunto de prueba se presenta en

el Anexo K.

### 6.5. Discusión general de resultados

En conjunto, los resultados obtenidos muestran que el modelo propuesto fue capaz de resolver con alto desempeño la tarea de clasificación multiclase de escenas satelitales sobre el conjunto UC Merced. La exactitud superior al 94 % en prueba indica que la red logró aprender representaciones discriminativas útiles para diferenciar las 21 categorías consideradas en el estudio.

Por otra parte, la diferencia entre entrenamiento y prueba sugiere la existencia de cierto grado de especialización sobre los datos de entrenamiento, algo esperable en modelos de alta capacidad. Sin embargo, el comportamiento observado en validación y prueba indica que las estrategias de regularización y control del entrenamiento implementadas fueron suficientes para sostener una generalización adecuada.

Finalmente, el análisis por clase y las matrices de confusión permiten concluir que el modelo no solo alcanzó un buen desempeño global, sino que además mostró consistencia en la mayoría de las categorías, concentrando sus errores en un número reducido de escenas visualmente más exigentes. Este resultado aporta evidencia de que la arquitectura y la estrategia de entrenamiento adoptadas fueron apropiadas para el problema planteado.

# 7. Evaluación de robustez del modelo propuesto ante perturbaciones controladas

Este capítulo evalúa la sensibilidad del modelo propuesto frente a perturbaciones controladas en las imágenes de entrada, considerando degradaciones que pueden afectar su generalización en escenarios reales.

El análisis se realizó directamente sobre la arquitectura propuesta, utilizando versiones perturbadas del *UC Merced Land Use Dataset* y manteniendo fija la configuración del modelo. Las perturbaciones consideradas fueron: (i) reducción de resolución espacial, (ii) ruido AWGN con SNR controlada y (iii) desenfoque mediante *kernels* promedio.

## 7.1. Planteamiento general del análisis de robustez

El propósito de este análisis fue determinar en qué medida el modelo propuesto conserva su capacidad de clasificación cuando las imágenes de entrada son degradadas de forma controlada. En todos los experimentos se mantuvo fija la arquitectura del modelo y la configuración general de entrenamiento, de manera que las diferencias observadas entre escenarios pudieran atribuirse directamente a la perturbación introducida y no a cambios en el diseño del clasificador.

Este enfoque permite interpretar el desempeño observado como una medida de la sensibilidad del modelo frente a cambios en la calidad de la información visual.

## 7.2. Resultados globales ante perturbaciones controladas

Como referencia, en condiciones nominales el modelo propuesto alcanzó una exactitud de prueba de 94.29 %. La Tabla 3 resume los resultados globales obtenidos en los distintos escenarios analizados.

**Tabla 3.**

*Resumen global del desempeño del modelo propuesto ante perturbaciones controladas.*

| **Tipo de perturbación** | **Escenario** | **Exactitud de prueba (%)** | **Comportamiento general** |
|---|---|---|---|
| Resolución espacial | $128 \times 128$ | 95.71 | Desempeño comparable al nominal |
| Resolución espacial | $64 \times 64$ | 92.38 | Degradación moderada |
| Resolución espacial | $32 \times 32$ | 87.62 | Caída evidente por pérdida de detalle |
| Ruido AWGN | 30 dB | 93.33 | Comportamiento estable |
| Ruido AWGN | 20 dB | 93.81 | Comportamiento estable |
| Ruido AWGN | 10 dB | 69.05 | Deterioro severo |
| Ruido AWGN | 5 dB | 38.57 | Deterioro crítico |
| Desenfoque por *kernel* | $2 \times 2$ | 93.33 | Alta tolerancia |
| Desenfoque por *kernel* | $4 \times 4$ | 94.76 | Alta tolerancia |
| Desenfoque por *kernel* | $8 \times 8$ | 91.90 | Degradación moderada |
| Desenfoque por *kernel* | $16 \times 16$ | 86.67 | Degradación marcada |

En el experimento de reducción de resolución espacial, el modelo mantuvo un desempeño competitivo para reducciones moderadas, alcanzando incluso una exactitud de prueba de 95.71 % en $128 \times 128$. Sin embargo, al disminuir la resolución a $64 \times 64$ y $32 \times 32$, la exactitud descendió a 92.38 % y 87.62 %, respectivamente. Esto evidencia una sensibilidad creciente frente a la pérdida de detalle espacial.

En el caso del ruido AWGN, el modelo conservó un comportamiento satisfactorio para niveles altos de SNR, con exactitudes de prueba de 93.33 % y 93.81 % en 30 dB y 20 dB, respectivamente. No obstante, cuando la relación señal a ruido disminuyó a 10 dB y 5 dB, el rendimiento

cayó de forma severa hasta 69.05 % y 38.57 %, lo cual indica una alta sensibilidad frente a degradaciones intensas por ruido.

Por su parte, en la evaluación mediante desenfoque por *kernels* promedio, el modelo mostró una buena tolerancia para niveles leves de suavizado, con exactitudes de prueba de 93.33 % y 94.76 % para *kernels* de $2 \times 2$ y $4 \times 4$. Sin embargo, al aumentar el tamaño del *kernel* a $8 \times 8$ y $16 \times 16$, el desempeño disminuyó progresivamente hasta 91.90 % y 86.67 %, lo cual sugiere que la pérdida de nitidez afecta la discriminación de texturas, bordes y estructuras espaciales relevantes.

### 7.3. Discusión general del análisis de robustez

En conjunto, los resultados permiten concluir que el modelo propuesto presenta un comportamiento satisfactorio frente a perturbaciones suaves, pero no mantiene la misma robustez cuando las degradaciones son severas. Las mayores dificultades se observan cuando la perturbación elimina o distorsiona información visual fina, particularmente en escenas cuya clasificación depende de contornos, textura y organización espacial.

Estos hallazgos caracterizan de manera directa una limitación importante del modelo propuesto: su capacidad de generalización depende en buena medida de la preservación de rasgos visuales discriminativos de alta frecuencia. En consecuencia, el análisis de robustez complementa la evaluación nominal del clasificador y aporta evidencia sobre sus alcances y limitaciones en condiciones degradadas.

El desarrollo ampliado de este análisis, incluyendo diseño experimental detallado, métricas por clase y discusión específica por perturbación, se presenta en el Anexo E.

# 8. Interpretabilidad de resultados mediante mapas de saliencia

Este capítulo presenta un análisis interpretativo complementario de los resultados obtenidos por el modelo de clasificación de escenas satelitales. Con este propósito, se emplearon mapas de saliencia basados en gradiente como herramienta de inspección visual, con el fin de identificar qué regiones de la imagen influyen con mayor intensidad en la predicción del clasificador. El desarrollo ampliado de este análisis, incluyendo el procedimiento de generación, los casos cualitativos representativos y la discusión detallada de sus alcances y limitaciones, se presenta en el Anexo A.

## 8.1. Propósito de la interpretabilidad en el estudio

La incorporación de mapas de saliencia en esta tesis tuvo como finalidad complementar la evaluación cuantitativa del modelo mediante un análisis visual de sus predicciones. En particular, esta herramienta permitió explorar si las decisiones del clasificador estaban asociadas con regiones coherentes de la escena o, por el contrario, con patrones poco interpretables o potencialmente espurios.

Desde esta perspectiva, la interpretabilidad no se aborda como un mecanismo de reemplazo de las métricas tradicionales, sino como una estrategia complementaria para enriquecer el análisis del comportamiento del modelo. De esta manera, los mapas de saliencia aportan una visión adicional sobre la relación entre la imagen de entrada y la clase predicha, facilitando una lectura más profunda de los resultados obtenidos.

### 8.2. Síntesis del análisis interpretativo

Los mapas de saliencia se generaron sobre muestras del conjunto de prueba mediante un enfoque basado en gradientes, coherente con el flujo de evaluación utilizado por el modelo. A partir de esta estrategia se analizaron tres tipos de casos representativos: (i) un ejemplo correctamente clasificado de una categoría con buen desempeño global, (ii) un ejemplo perteneciente a una clase visualmente más exigente y (iii) un caso de error de clasificación.

En los casos correctamente clasificados, el análisis mostró que el modelo tiende a concentrar su atención sobre regiones visualmente relevantes para la escena, ya sea en estructuras dominantes claramente distinguibles o en patrones espaciales distribuidos sobre varias zonas de la imagen. Esto sugiere que, cuando el clasificador acierta, su predicción suele apoyarse en información consistente con la semántica de la categoría evaluada.

En el caso de error analizado, la distribución de la saliencia sugirió que la predicción equivocada no fue aleatoria, sino posiblemente inducida por similitudes geométricas compartidas entre la clase real y la clase predicha. Este resultado aporta evidencia útil para interpretar ciertas confusiones del modelo en categorías con patrones visuales estructuralmente cercanos.

En conjunto, el análisis interpretativo permite complementar la evaluación cuantitativa del clasificador, ya que ofrece una lectura visual de los rasgos a los que el modelo parece responder tanto en escenarios de acierto como en casos de confusión.

### 8.3. Alcances y limitaciones del análisis interpretativo

Si bien los mapas de saliencia ofrecen información útil sobre la sensibilidad local del modelo frente a la entrada, su interpretación debe realizarse con cautela. Estos mapas no constituyen

una explicación causal de la predicción, sino una aproximación basada en gradientes que resalta regiones con alta influencia sobre la salida del clasificador.

Asimismo, la apariencia del mapa puede depender de decisiones de implementación como el método de agregación entre canales, la normalización aplicada o el tipo de visualización utilizado. Por esta razón, los resultados obtenidos mediante saliencia deben entenderse como evidencia cualitativa complementaria y no como un criterio único de validación del modelo.

Aun con estas limitaciones, el uso de mapas de saliencia en esta tesis resulta valioso porque permite enriquecer la discusión de resultados, aportando una perspectiva visual sobre los patrones que la red emplea para discriminar entre categorías de escena.

# 9. Evaluación de un modelo robusto inspirado en la literatura

En el capítulo anterior se analizó el comportamiento del modelo propuesto frente a distintas perturbaciones aplicadas sobre las imágenes de entrada. Para complementar dicho análisis, en esta etapa se incorporó una segunda aproximación experimental inspirada en la literatura especializada, orientada a evaluar una estrategia diseñada para mejorar la tolerancia del clasificador ante ruido y degradación visual.

Como referencia se tomó el modelo *Deep Salient Feature Based Anti-Noise Transfer Network* (DSFATN), propuesto por Gong, Xie, Liu, Shi, y Zheng (2018) para clasificación de escenas en imágenes de percepción remota bajo variaciones de escala y ruido. En esta tesis no se realizó una réplica exacta del método original, sino una implementación adaptada de sus ideas principales al entorno experimental del trabajo. Por ello, los resultados deben interpretarse como una referencia comparativa orientativa y no como una validación estricta del método original. El desarrollo ampliado de esta adaptación y de la comparación detallada se presenta en el Anexo I.

## 9.1. Síntesis de la implementación adaptada

La adaptación realizada conservó tres elementos generales del enfoque original: el uso del conjunto UC Merced como base experimental, la extracción de representaciones profundas mediante VGG-19 preentrenada en ImageNet y la incorporación de una restricción anti-ruido dentro del proceso de entrenamiento. Sin embargo, varios componentes del método original no fueron reproducidos de manera estricta. Entre ellos se encuentran el módulo de saliencia PBVS derivado de GBVS, la formulación exacta de la *joint loss*, la evaluación multibase sobre SIRI-WHU y SAT-6,

y la comparación con variantes como TN-1 y TN-2.

En consecuencia, la implementación desarrollada en esta tesis debe entenderse como una adaptación funcional orientada por el enfoque DSFATN, útil como referencia comparativa dentro del marco experimental definido, pero limitada frente al alcance metodológico del trabajo original.

### 9.2. Resultados globales del modelo robusto adaptado

La Tabla 4 resume los resultados obtenidos por la implementación adaptada inspirada en DSFATN sobre el conjunto original y en los escenarios perturbados mediante reducción de resolución, ruido AWGN y desenfoque por *kernels* promedio.

**Tabla 4.**

*Resumen global del desempeño del modelo robusto adaptado inspirado en DSFATN.*

| **Escenario** | **Accuracy promedio (%)** | **Desv. estándar (%)** |
|---|---|---|
| Original | 81.95 | 1.08 |
| Resolución $128 \times 128$ | 82.33 | 0.70 |
| Resolución $64 \times 64$ | 74.96 | 1.39 |
| Resolución $32 \times 32$ | 63.10 | 1.56 |
| SNR 30 dB | 80.67 | 1.23 |
| SNR 20 dB | 76.71 | 0.49 |
| SNR 10 dB | 63.00 | 1.84 |
| SNR 5 dB | 50.90 | 1.67 |
| Kernel $2 \times 2$ | 82.67 | 1.20 |
| Kernel $4 \times 4$ | 82.43 | 0.74 |
| Kernel $8 \times 8$ | 74.95 | 0.46 |
| Kernel $16 \times 16$ | 66.10 | 2.29 |

En condiciones originales, el modelo robusto adaptado alcanzó una exactitud promedio de 81.95 %, con una desviación estándar de 1.08 %. Este resultado muestra que la implementación desarrollada conserva una capacidad de clasificación funcional sobre UC Merced dentro del

protocolo experimental adoptado en esta tesis, aunque con un desempeño inferior al del modelo propuesto.

En escenarios de reducción de resolución, el comportamiento se mantuvo cercano al nominal para $128 \times 128$, pero presentó caídas importantes en $64 \times 64$ y especialmente en $32 \times 32$. Frente al ruido AWGN, el modelo mostró cierta tolerancia para 30 dB y 20 dB, pero el deterioro fue marcado para 10 dB y crítico para 5 dB. De manera similar, en el caso del desenfoque por *kernels* promedio, el desempeño se mantuvo cercano al escenario original para perturbaciones leves, pero disminuyó de forma clara para *kernels* de mayor tamaño.

### 9.3. Discusión general

Los resultados obtenidos indican que la implementación adaptada inspirada en DSFATN conserva un desempeño relativamente estable ante perturbaciones leves, pero su rendimiento disminuye de forma marcada cuando la degradación se vuelve severa. Esto sugiere que, aun cuando la estrategia incorporó una restricción anti-ruido, la capacidad de discriminación del clasificador sigue dependiendo en buena medida de la preservación de información espacial fina.

Bajo las condiciones experimentales definidas en esta tesis, la adaptación inspirada en DSFATN no alcanzó el desempeño global del modelo propuesto. No obstante, esta observación debe entenderse en el contexto de una implementación parcial del enfoque original. Por tanto, el valor principal de este capítulo radica en aportar una referencia comparativa orientativa basada en la literatura, más que en establecer una jerarquía definitiva entre ambos métodos.

En conjunto, este análisis permite situar el desempeño del modelo propuesto frente a una implementación funcional inspirada en DSFATN y resalta la importancia de reproducir con alto grado de fidelidad los módulos originales cuando se pretende realizar comparaciones metodológi-

cas estrictas.

## Conclusiones

En este trabajo se desarrolló y evaluó un modelo de aprendizaje profundo para la clasificación automática de escenas satelitales utilizando el conjunto de datos UC Merced Land Use Dataset. La metodología permitió construir un flujo experimental completo, desde la preparación de los datos y la reproducción de un modelo base, hasta la implementación de una arquitectura CNN mejorada y su evaluación bajo diferentes condiciones de degradación visual.

Los resultados obtenidos evidencian que el modelo propuesto alcanzó un desempeño adecuado en el escenario nominal de clasificación multiclase, con una exactitud de prueba de 94.29 %. Este comportamiento indica que las modificaciones introducidas en la arquitectura, especialmente el aumento progresivo de filtros convolucionales, el uso de `GlobalAveragePooling2D`, la regularización L2 y el uso de *dropout*, contribuyeron a mejorar la capacidad de generalización del clasificador frente al modelo base reproducido.

El análisis por clase permitió identificar que el modelo presenta un comportamiento sólido en la mayoría de categorías del conjunto de datos. Sin embargo, también se observaron algunas confusiones en clases con patrones visuales similares, especialmente en escenas residenciales, urbanas o con estructuras espaciales parecidas. Esto confirma que la clasificación de escenas satelitales no depende únicamente de objetos individuales, sino también de la textura, la distribución espacial y la composición global de la imagen.

Las pruebas de robustez mostraron que el desempeño del modelo se mantiene relativamente estable ante degradaciones leves o moderadas, como reducción de resolución a $128 \times 128$, ruido

AWGN con SNR altos y desenfoque mediante *kernels* pequeños. No obstante, el rendimiento disminuye de forma considerable cuando las perturbaciones son severas, particularmente en imágenes de baja resolución, ruido intenso y desenfoque fuerte. Esto permite concluir que, aunque el modelo presenta buena capacidad de clasificación en condiciones controladas, su desempeño sigue siendo sensible a la pérdida de información espacial y visual.

La implementación adaptada inspirada en el enfoque DSFATN aportó una referencia comparativa adicional basada en la literatura especializada. Bajo las condiciones experimentales definidas en esta tesis, dicha adaptación presentó un desempeño inferior al del modelo propuesto. Sin embargo, esta observación debe interpretarse con cautela, ya que la implementación realizada no correspondió a una reproducción exacta del método original, sino a una adaptación parcial de sus principios principales. Por ello, la comparación permite situar el desempeño del modelo propuesto frente a una referencia metodológica inspirada en DSFATN, pero no establecer una conclusión definitiva sobre el comportamiento del método original reportado en la literatura.

En términos generales, el trabajo cumplió con el propósito de implementar, entrenar y evaluar modelos de aprendizaje profundo para clasificación de escenas satelitales, incorporando tanto métricas cuantitativas como análisis de robustez. Los resultados obtenidos constituyen una base experimental útil para futuros estudios orientados al diseño de modelos más robustos, transferibles y aplicables a imágenes satelitales capturadas bajo condiciones reales de variabilidad, ruido y degradación.

## Referencias

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., y Kim, B. (2018). *Sanity checks for saliency maps.* arXiv preprint. Descargado de https://arxiv.org/abs/1810.03292

Castelluccio, M., Poggi, G., Sansone, C., y Verdoliva, L. (2015). Land use classification in remote sensing images by convolutional neural networks. *arXiv preprint*, *arXiv:1508.00092*. Descargado de https://arxiv.org/abs/1508.00092

Cheng, G., Xie, X., Han, J., Guo, L., y Xia, G.-S. (2020). Remote sensing image scene classification meets deep learning: Challenges, methods, benchmarks, and opportunities. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *13*, 3735–3756. Descargado de https://arxiv.org/abs/2005.01094 doi: 10.1109/JSTARS.2020.3005403

Cole, R. M. (2018). *A brief introduction to satellite image classification with neural networks.* Medium. Descargado de https://medium.com/@robmarkcole/a-brief-introduction-to-satellite-image-classification-with-neuralnetworks-3ce28be15683 (Retrieved January 27, 2026)

Datla, R., Perveen, N., y Krishna Mohan, C. (2024). Learning scene-vectors for remote sensing image scene classification. *Neurocomputing*, *587*, 127679. Descargado de https://doi.org/10.1016/j.neucom.2024.127679 doi: 10.1016/j.neucom.2024.127679

Gong, X., Xie, Z., Liu, Y., Shi, X., y Zheng, Z. (2018). Deep salient feature based anti-noise transfer network for scene classification of remote sensing imagery. *Remote Sensing*, *10*(3),

410. Descargado de https://doi.org/10.3390/rs10030410 doi: 10.3390/rs10030410

Hao, X., Liu, L., Yang, R., Yin, L., Zhang, L., y Li, X. (2023). A review of data augmentation methods of remote sensing image target recognition. *Remote Sensing*, *15*(3), 827. Descargado de https://doi.org/10.3390/rs15030827 doi: 10.3390/rs15030827

Hendrycks, D., y Dietterich, T. G. (2019). Benchmarking neural network robustness to common corruptions and perturbations. En *International conference on learning representations (iclr)*. Descargado de https://openreview.net/forum?id=HJz6tiCqYm

Hu, F., Xia, G.-S., Hu, J., y Zhang, L. (2015). Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing*, *7*(11), 14680–14707. Descargado de https://www.mdpi.com/2072-4292/7/11/14680 doi: 10.3390/rs71114680

Ioffe, S., y Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. En *Proceedings of the 32nd international conference on machine learning (icml)* (pp. 448–456). PMLR. Descargado de https://proceedings.mlr.press/v37/ioffe15.html

Keras Team. (s.f.-a). *Conv2d layer*. Keras documentation. Descargado de https://keras.io/api/layers/convolution_layers/convolution2d/ (Retrieved January 24, 2026)

Keras Team. (s.f.-b). *Maxpooling2d layer*. Keras documentation. Descargado de https://keras.io/api/layers/pooling_layers/max_pooling2d/ (Retrieved January 24, 2026)

Nogueira, K., Penatti, O. A. B., y dos Santos, J. A. (2017). Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognition*, *61*, 539–556. Descargado de https://doi.org/10.1016/j.patcog.2016.07.001 doi:

10.1016/j.patcog.2016.07.001

Patel, M. (2023). *The complete guide to image preprocessing techniques in python.* Medium. Descargado de https://medium.com/@maahip1304/the-complete-guide-to-image-preprocessing-techniques-in-python-dca30804550c (Retrieved January 27, 2026)

scikit-learn developers. (s.f.-a). *precision_recall_fscore_support.* scikit-learn documentation. Descargado de https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html (Retrieved January 27, 2026)

scikit-learn developers. (s.f.-b). *train_test_split.* scikit-learn documentation. Descargado de https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html (Retrieved January 27, 2026)

Simonyan, K., Vedaldi, A., y Zisserman, A. (2013). *Deep inside convolutional networks: Visualising image classification models and saliency maps.* arXiv preprint. Descargado de https://arxiv.org/abs/1312.6034

Smilkov, D., Thorat, N., Kim, B., Viégas, F., y Wattenberg, M. (2017). *Smoothgrad: Removing noise by adding noise.* arXiv preprint. Descargado de https://arxiv.org/abs/1706.03825

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., y Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958. Descargado de https://www.jmlr.org/papers/v15/srivastava14a.html

Stout, D. W. (2025). *Weighted metrics for multi-class models explained.* Magai. Descargado de https://magai.co/weighted-metrics-for-multi-class-models-explained/ (Retrieved January 27, 2026)

TensorFlow Developers. (s.f.-a). *Categoricalcrossentropy.* TensorFlow documentation. Descargado de https://www.tensorflow.org/api_docs/python/tf/keras/losses/CategoricalCrossentropy (Retrieved January 27, 2026)

TensorFlow Developers. (s.f.-b). *tf.gradienttape.* TensorFlow documentation. Descargado de https://www.tensorflow.org/api_docs/python/tf/GradientTape (Retrieved January 27, 2026)

TensorFlow Developers. (s.f.-c). *tf.keras.activations.relu.* TensorFlow documentation. Descargado de https://www.tensorflow.org/api_docs/python/tf/keras/activations/relu (Retrieved January 24, 2026)

Tombe, R., y Viriri, S. (2023). Remote sensing image scene classification: Advances and open challenges. *Geomatics*, *3*(1), 137–155. Descargado de https://doi.org/10.3390/geomatics3010007 doi: 10.3390/geomatics3010007

Yang, Y., y Newsam, S. (2010). *Uc merced land use dataset.* University of California, Merced. Descargado de http://weegee.vision.ucmerced.edu/datasets/landuse.html (Retrieved January 27, 2026)

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., y Parmar, M. (2024). A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, *57*, 99. Descargado de https://doi.org/10.1007/s10462-024-10721-6 doi: 10.1007/s10462-024-10721-6

# Apéndice A. Análisis ampliado de interpretabilidad mediante mapas de saliencia

Este apéndice amplía el análisis interpretativo del modelo mediante mapas de saliencia. Incluye el procedimiento de generación, los casos cualitativos y la discusión de sus alcances y limitaciones.

## A.1. Propósito de la interpretabilidad en el estudio

La incorporación de mapas de saliencia en esta tesis tuvo como finalidad complementar la evaluación cuantitativa del modelo mediante un análisis visual de sus predicciones. En particular, esta herramienta permitió explorar si las decisiones del clasificador estaban asociadas con regiones coherentes de la escena o, por el contrario, con patrones poco interpretables o potencialmente espurios.

## A.2. Procedimiento de generación de mapas de saliencia

Los mapas de saliencia se generaron a partir de muestras del conjunto de prueba, utilizando un enfoque basado en gradientes calculados con `tf.GradientTape`. Para cada imagen de entrada $x$, se determinó primero la clase predicha por el modelo y, a continuación, se calculó el gradiente del puntaje asociado a dicha clase respecto a la imagen de entrada.

Si $\hat{c} = \arg\max f(x)$ representa la clase predicha por el modelo, se tomó como puntaje de interés $s = f_{\hat{c}}(x)$, y posteriormente se obtuvo el gradiente $\nabla_x s$. A partir de este gradiente, se construyó un mapa escalar de saliencia combinando los tres canales RGB mediante el máximo del valor absoluto, según:

$$M(x) = \max_{k \in \{R,G,B\}} \left| \frac{\partial s}{\partial x_k} \right|.$$

Una vez obtenido el mapa escalar, este se normalizó al intervalo $[0,1]$ y se aplicó una corrección gamma de la forma $M \leftarrow M^{0,5}$, con el objetivo de mejorar el contraste visual de las regiones más relevantes. Finalmente, el mapa se superpuso sobre la imagen original utilizando una escala de color tipo *heatmap* y una transparencia fija.

Este procedimiento se aplicó sobre imágenes previamente preprocesadas a $256 \times 256 \times 3$ y normalizadas en el rango $[0,1]$, manteniendo coherencia con el flujo de evaluación utilizado por el modelo.

## A.3. Análisis cualitativo de casos representativos

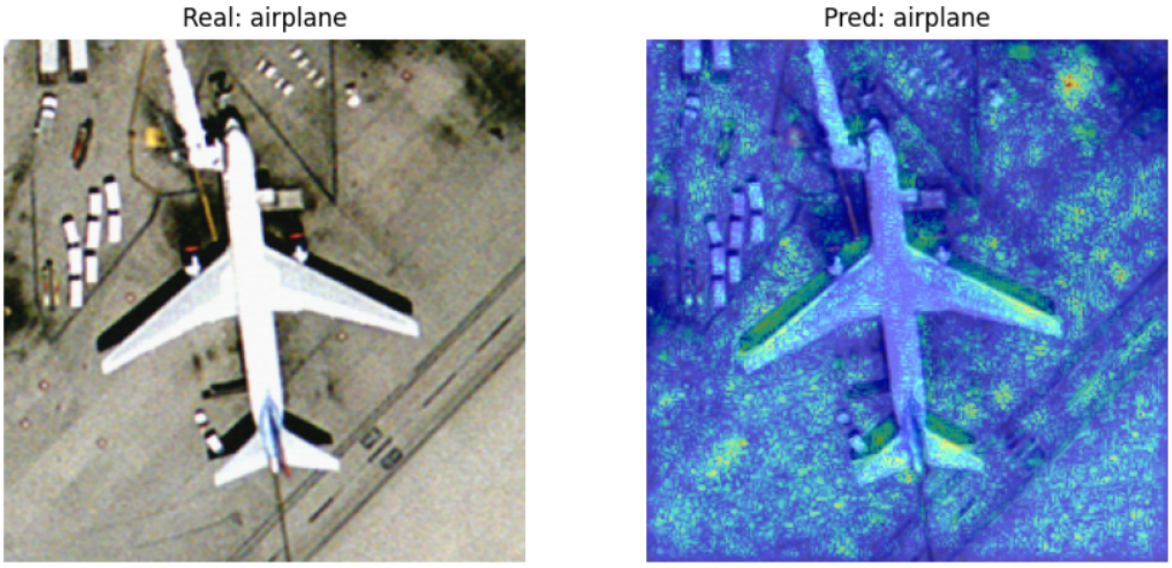
Con el fin de profundizar el análisis interpretativo del modelo, se seleccionaron tres tipos de casos del conjunto de prueba: (i) un ejemplo correctamente clasificado de una categoría con buen desempeño global, (ii) un ejemplo perteneciente a una clase visualmente más exigente y (iii) un caso de error de clasificación.

### *A.3.1. Caso de acierto en una clase bien reconocida*

La Figura 4 presenta un ejemplo correctamente clasificado correspondiente a la clase *airplane*. El análisis sugiere que la activación se concentra sobre estructuras visuales dominantes asociadas con la aeronave y su entorno inmediato, aportando evidencia de que el modelo utiliza información visual consistente con la semántica de la categoría.

**Figura 4.**

*Mapa de saliencia para una muestra de la clase airplane.*



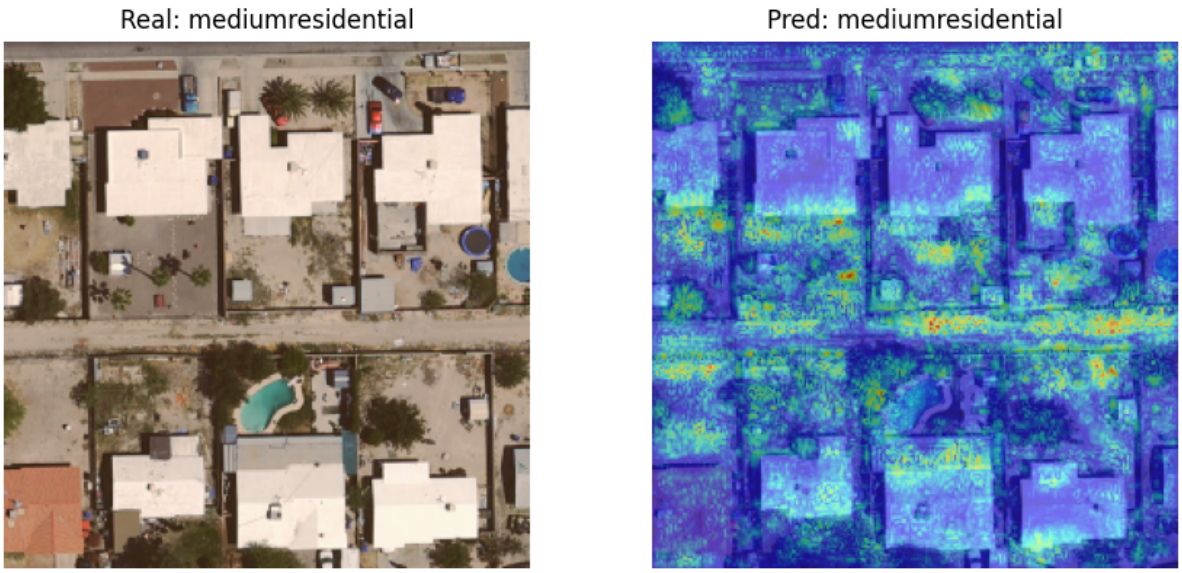

*Nota.* La figura presenta un ejemplo correctamente clasificado de la clase *airplane*. A la izquierda se muestra la imagen original y a la derecha la superposición del mapa de saliencia, el cual resalta las regiones que tuvieron mayor influencia en la decisión del modelo.

### *A.3.2. Caso de una clase visualmente problemática*

La Figura 5 presenta un ejemplo correctamente clasificado de la clase *mediumresidential*. En este caso, la saliencia aparece repartida sobre varias regiones de la escena, especialmente en cubiertas de edificaciones, límites entre construcciones y franjas viales. Esto sugiere que la decisión del modelo se apoya en una combinación de patrones espaciales repetitivos y relaciones de vecindad entre múltiples estructuras.

**Figura 5.**

*Mapa de saliencia para una muestra de la clase mediumresidential.*



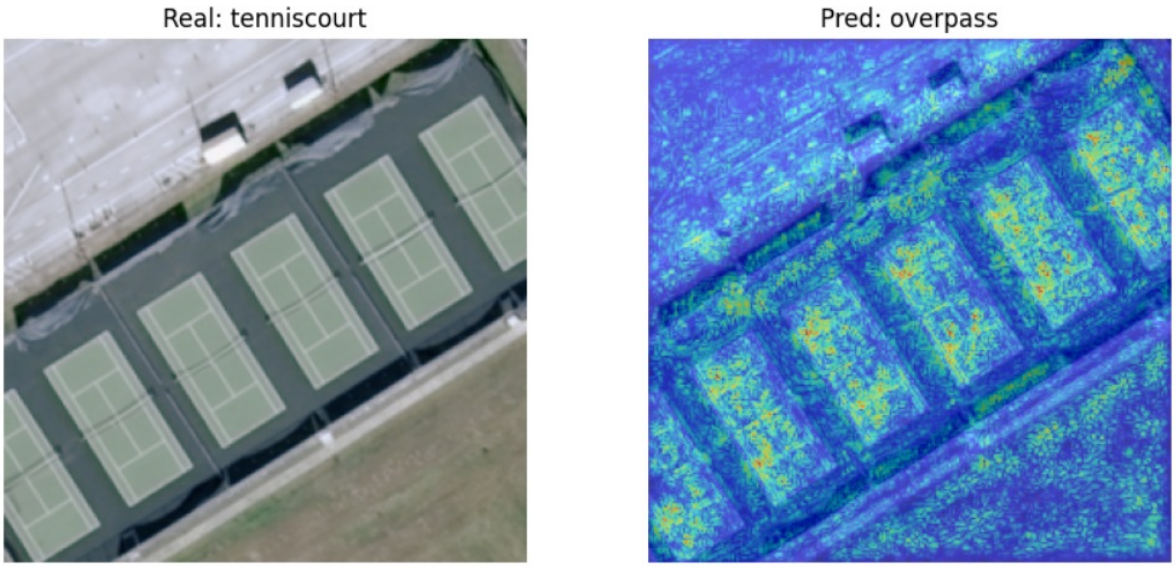

*Nota.* La figura presenta un ejemplo correctamente clasificado de la clase *mediumresidential*. A la izquierda se muestra la imagen original y a la derecha la superposición del mapa de saliencia, en la cual se resaltan las regiones de la escena que tuvieron mayor influencia en la decisión del modelo.

### *A.3.3. Caso de error de clasificación*

La Figura 6 presenta un ejemplo mal clasificado por el modelo, cuya clase real corresponde a *tenniscourt*, mientras que la predicción generada fue *overpass*. En este ejemplo, la saliencia se concentra sobre varias estructuras rectangulares y lineales distribuidas de forma paralela, lo que sugiere que la predicción errónea pudo estar inducida por similitudes geométricas compartidas entre ambas categorías.

**Figura 6.**

*Mapa de saliencia para un caso de error de clasificación.*




*Nota.* La figura presenta un ejemplo de error de clasificación en el que la clase real corresponde a *tenniscourt*, mientras que la clase predicha por el modelo fue *overpass*. A la izquierda se muestra la imagen original y a la derecha la superposición del mapa de saliencia, la cual resalta las regiones que influyeron en la decisión del modelo.

En conjunto, este análisis complementa la evaluación cuantitativa, ya que permite interpretar cómo se distribuye la atención del modelo tanto en casos de acierto como en escenarios de confusión.

### A.4. Alcances y limitaciones del análisis interpretativo

Si bien los mapas de saliencia ofrecen información útil sobre la sensibilidad local del modelo frente a la entrada, su interpretación debe realizarse con cautela. Estos mapas no constituyen una explicación causal de la predicción, sino una aproximación basada en gradientes que resalta

regiones con alta influencia sobre la salida del clasificador.

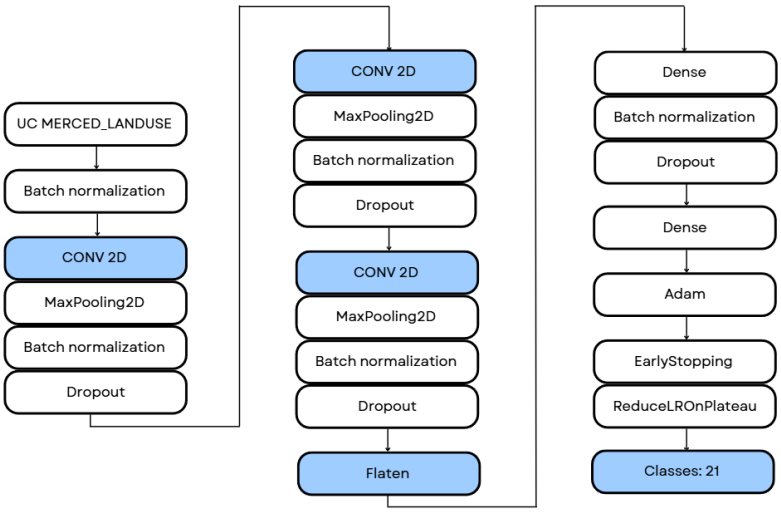
Asimismo, la apariencia del mapa puede depender de decisiones de implementación como el método de agregación entre canales, la normalización aplicada o el tipo de visualización utilizado. Por ello, los resultados obtenidos mediante saliencia deben entenderse como evidencia cualitativa complementaria y no como un criterio único de validación del modelo.

## Apéndice B. Arquitectura del baseline reproducido y comparación estructural de modelos

Este apéndice documenta la arquitectura del *baseline* reproducido y la comparación estructural detallada con el modelo propuesto.

**Figura 7.**

*Arquitectura del modelo baseline.*




*Nota.* La figura presenta la arquitectura del modelo *baseline* reproducido y utilizado como referencia experimental en el estudio. Este modelo permite establecer un punto de comparación inicial frente al modelo propuesto para la clasificación multiclase de escenas satelitales.

**Tabla 5.**

*Comparación estructural entre el baseline reproducido y el modelo propuesto.*

| **Componente** | **Baseline reproducido** | **Modelo propuesto** |
|---|---|---|
| Entrada | $256 \times 256 \times 3$ | $256 \times 256 \times 3$ |
| Bloques convolucionales | 3 bloques, filtros $32/64/128$ | 3 bloques, filtros $64/128/256$ |
| Activación / *padding* | ReLU / `same` | ReLU / `same` |
| Normalización | `BatchNormalization` | `BatchNormalization` |
| *Pooling* | `MaxPooling2D` | `MaxPooling2D` |
| Capa de agregación | `Flatten` | `GlobalAveragePooling2D` |
| Cabeza densa | `Dense(256)` + `Dense(128)` | `Dense(256)` + `Dense(128)` |
| *Dropout* | 0.25 en convolucional y 0.5 en densa | 0.2 en convolucional y 0.3 en densa |
| Regularización L2 | No | Conv.: $\lambda = 5 \times 10^{-4}$; densa: $\lambda = 10^{-3}$ |
| Salida | `Dense(21)` + *softmax* | `Dense(21)` + *softmax* |
| Parámetros (totales / entrenables) | 33,880,629 / 33,878,965 | 1,251,925 / 1,249,365 |

*Nota*. Comparación elaborada a partir de la arquitectura reproducida y del modelo final propuesto en este estudio.

## Apéndice C. Ejemplos representativos del conjunto de datos

Este apéndice presenta una muestra visual de algunas de las clases incluidas en el conjunto de datos empleado en el estudio.

### Ejemplos de clases

La Figura 8 permite observar la diversidad visual presente en el conjunto de datos utilizado. Entre las clases representadas se evidencian diferencias en la organización espacial de los objetos, la presencia de estructuras geométricas, la textura del terreno y las variaciones de color, aspectos que inciden directamente en la complejidad de la tarea de clasificación multiclase.

**Figura 8.**

*Ejemplos representativos de clases del conjunto de datos.*


*Nota.* Las imágenes corresponden a muestras representativas del conjunto de datos empleado en el estudio y se incluyen con fines ilustrativos para apoyar la comprensión visual de las categorías analizadas.

**Observaciones cualitativas**

A partir de la inspección visual de las imágenes presentadas, se destacan los siguientes aspectos:

- Algunas clases presentan patrones claramente diferenciables, caracterizados por estructuras geométricas bien definidas.
- Existen categorías con alta variabilidad visual, lo que incrementa la complejidad de la tarea de clasificación.
- Se observan similitudes entre ciertas clases, lo que puede inducir confusión en el modelo durante el proceso de aprendizaje.
- Las características relacionadas con textura, color y disposición espacial constituyen elementos relevantes para la discriminación entre categorías.

Estas observaciones complementan el análisis cuantitativo presentado en el cuerpo principal del documento.

# Apéndice D. Estado del arte ampliado sobre CNN aplicadas a UC Merced

Este apéndice amplía la revisión bibliográfica sobre el uso de redes neuronales convolucionales en clasificación de escenas de teledetección sobre UC Merced.

## D.1. CNNs aplicadas a UC Merced

El uso de redes neuronales convolucionales (CNN) para clasificación de escenas en teledetección ha sido ampliamente explorado sobre el conjunto UC Merced debido a su carácter de *benchmark*. Trabajos tempranos mostraron que las CNN entrenadas o adaptadas pueden superar enfoques basados en descriptores manuales al capturar patrones espaciales y texturales relevantes para diferenciar escenas. Por ejemplo, Castelluccio, Poggi, Sansone, y Verdoliva (2015) estudiaron el uso de CNN para clasificación de uso del suelo en imágenes de teledetección; Nogueira y cols. (2017) analizaron estrategias para explotar mejor CNN en clasificación de escenas remotas; y Hu, Xia, Hu, y Zhang (2015) mostraron que el ajuste fino (*fine-tuning*) de redes preentrenadas puede ser efectivo cuando la disponibilidad de datos etiquetados es limitada.

De forma complementaria, revisiones más recientes han sintetizado la evolución del área y han señalado retos persistentes como la similitud interclase, la variabilidad intraclase y la fuerte dependencia de los protocolos experimentales y de partición de datos (Cheng y cols., 2020; Tombe y Viriri, 2023). Asimismo, trabajos recientes continúan utilizando UC Merced como conjunto de referencia para evaluar nuevos métodos de clasificación de escenas, lo que confirma su vigencia dentro de la literatura especializada (Datla, Perveen, y Krishna Mohan, 2024).

***Aplicación en este trabajo.*** Este marco de trabajos previos sirve como referencia para justificar el uso de CNN sobre un *benchmark* multiclase y para motivar una metodología reproducible, en la que el rendimiento se reporta mediante métricas estándar y análisis por clase.

# Apéndice E. Evaluación ampliada de robustez del modelo propuesto ante perturbaciones controladas

Este apéndice presenta el desarrollo ampliado del análisis de robustez del modelo propuesto, incluyendo diseño experimental, resultados por perturbación, métricas por clase y discusión específica.

## E.1. Planteamiento general del análisis de robustez

El propósito de este análisis fue determinar en qué medida el modelo propuesto conserva su capacidad de clasificación cuando las imágenes de entrada son degradadas de forma controlada. Para ello, se siguió un mismo criterio experimental en todos los casos: se generaron nuevas versiones del conjunto de datos a partir de transformaciones específicas y, posteriormente, se evaluó el comportamiento del modelo propuesto manteniendo fija su arquitectura y la configuración general de entrenamiento. En consecuencia, las diferencias observadas entre escenarios se atribuyen a la naturaleza de la perturbación introducida y no a cambios en el diseño del clasificador.

Este enfoque permite interpretar el desempeño observado como una medida de la sensibilidad del modelo propuesto frente a cambios en la calidad de la información visual, manteniendo constante la configuración experimental del clasificador. En este apéndice no se plantea una comparación directa entre el modelo base y el modelo propuesto, sino la caracterización de la robustez del modelo propuesto frente a perturbaciones controladas.

## E.2. Robustez frente a reducción de resolución espacial

### *E.2.1. Diseño del experimento*

Con el fin de analizar el efecto de la baja resolución sobre la capacidad de generalización del modelo, se generaron tres versiones del conjunto de datos a partir del redimensionamiento de las imágenes originales. Las resoluciones consideradas fueron $128 \times 128$, $64 \times 64$ y $32 \times 32$ píxeles.

### *E.2.2. Resultados globales*

Como referencia, en condiciones nominales el modelo propuesto alcanzó una exactitud de prueba de 94.29 %. La Tabla 6 resume los resultados globales obtenidos para cada resolución evaluada.

**Tabla 6.**

*Resultados globales del modelo propuesto ante reducción de resolución.*

| Resolución | Exactitud de entrenamiento (%) | Exactitud de validación (%) | Exactitud de prueba (%) |
|---|---|---|---|
| $128 \times 128$ | 100.00 | 95.24 | 95.71 |
| $64 \times 64$ | 99.94 | 97.14 | 92.38 |
| $32 \times 32$ | 98.57 | 89.05 | 87.62 |

Los resultados muestran que el modelo mantiene un comportamiento sólido cuando la resolución se reduce a $128 \times 128$. Sin embargo, al disminuir la resolución a $64 \times 64$ y $32 \times 32$, el desempeño se deteriora, con una caída más evidente en la condición más severa.

#### *E.2.3. Análisis de métricas por clase*

Para la resolución de $64 \times 64$, el *classification report* sobre validación mostró una exactitud global de 0.97, con promedios *macro* y *weighted* de 0.97. En este caso, la mayoría de las clases conservaron un desempeño alto, aunque comenzaron a aparecer reducciones puntuales en métricas como precisión y *recall* en categorías tales como *baseballdiamond*, *freeway*, *runway* y *tenniscourt*.

En la resolución más severa, $32 \times 32$, la degradación fue más marcada. El reporte por clase mostró una exactitud global de 0.89, con promedios *macro* y *weighted* de 0.89. En este escenario, varias clases presentaron una reducción importante en precisión, *recall* y F1-score, particularmente *denseresidential*, *freeway*, *golfcourse*, *intersection*, *river* y *sparseresidential*.

#### *E.2.4. Discusión parcial*

En conjunto, los resultados evidencian que el modelo propuesto presenta una sensibilidad creciente frente a la pérdida de resolución espacial. Aunque una reducción moderada todavía permite conservar un desempeño competitivo, la degradación progresiva termina afectando la capacidad del modelo para diferenciar escenas con estructuras complejas o con alta similitud interclase.

### E.3. Robustez frente a ruido AWGN con SNR controlada

#### *E.3.1. Diseño del experimento*

Se evaluó el desempeño del modelo frente a la presencia de ruido gaussiano blanco aditivo (AWGN), generando versiones del conjunto de datos con niveles de SNR de 30 dB, 20 dB, 10 dB y 5 dB.

### *E.3.2. Resultados globales*

Como referencia, en condiciones nominales el modelo propuesto alcanzó una exactitud de prueba de 94.29 %. La Tabla 7 resume los resultados globales obtenidos para los diferentes niveles de SNR.

**Tabla 7.**

*Resultados globales del modelo propuesto ante ruido AWGN con SNR controlada.*

| SNR | Exactitud de entrenamiento ( %) | Exactitud de validación ( %) | Exactitud de prueba ( %) |
|---|---|---|---|
| 30 dB | 99.82 | 95.24 | 93.33 |
| 20 dB | 99.88 | 93.33 | 93.81 |
| 10 dB | 75.06 | 70.00 | 69.05 |
| 5 dB | 38.39 | 36.19 | 38.57 |

Los resultados indican que, para niveles altos de SNR, el modelo conserva un desempeño elevado. No obstante, al disminuir la SNR a 10 dB y 5 dB se observa una caída severa del rendimiento.

### *E.3.3. Análisis de métricas por clase*

Para 30 dB, el reporte de métricas mostró una exactitud global cercana a 0.95, con promedios *macro* y *weighted* también cercanos a 0.95. Para 20 dB, el comportamiento global continuó siendo alto, aunque comenzaron a aparecer ligeras reducciones adicionales en algunas clases.

El caso de 10 dB reveló una degradación mucho más significativa. La exactitud global del reporte fue cercana a 0.70, con descenso importante en varias clases. Finalmente, en 5 dB la degradación fue crítica, con exactitud global aproximada de 0.36.

#### *E.3.4. Discusión parcial*

Los resultados obtenidos frente a ruido AWGN muestran que el modelo propuesto es relativamente tolerante a perturbaciones leves o moderadas, pero altamente sensible cuando la relación señal a ruido disminuye de forma significativa. Esto motiva explorar estrategias de entrenamiento robusto o mecanismos de preprocesamiento específicos para escenarios ruidosos.

### E.4. Perturbación mediante desenfoque por *kernels* promedio

#### *E.4.1. Diseño del experimento*

Como tercera línea de análisis, se evaluó el efecto del desenfoque sobre el desempeño del modelo mediante filtrado lineal por convolución usando *kernels* promedio. Para ello, se definieron *kernels* de tamaño $2 \times 2$, $4 \times 4$, $8 \times 8$ y $16 \times 16$, construidos como matrices uniformes normalizadas.

#### *E.4.2. Resultados globales*

Como referencia, en condiciones nominales el modelo propuesto alcanzó una exactitud de prueba de 94.29 %. La Tabla 8 resume los resultados globales obtenidos para los diferentes tamaños de *kernel* evaluados.

**Tabla 8.**

*Resultados globales del modelo propuesto ante desenfoque por kernels promedio.*

| Kernel | Exactitud de entrenamiento (%) | Exactitud de validación (%) | Exactitud de prueba (%) | Macro avg | Weighted avg |
|---|---|---|---|---|---|
| $2 \times 2$ | 100.00 | 94.76 | 93.33 | 0.95 | 0.95 |
| $4 \times 4$ | 100.00 | 94.76 | 94.76 | 0.95 | 0.95 |
| $8 \times 8$ | 99.82 | 90.95 | 91.90 | 0.91 | 0.91 |
| $16 \times 16$ | 97.56 | 85.24 | 86.67 | 0.85 | 0.85 |

Los resultados muestran que el modelo conserva un desempeño alto bajo niveles de desenfoque leves, particularmente para *kernels* de $2 \times 2$ y $4 \times 4$. Sin embargo, al incrementar el tamaño del *kernel* a $8 \times 8$ y $16 \times 16$, se observa una degradación progresiva del rendimiento.

#### *E.4.3. Análisis de métricas por clase*

En el caso de $2 \times 2$, el modelo mantuvo un comportamiento global favorable, con una exactitud cercana a 0.95. Para el kernel $4 \times 4$, el desempeño global continuó siendo alto, aunque algunas clases mostraron ligeras reducciones en precisión, *recall* o F1-score.

Con el kernel $8 \times 8$, la degradación se hizo más evidente. El modelo alcanzó una exactitud global aproximada de 0.91, y varias categorías presentaron caídas apreciables. Finalmente, en el caso más severo, correspondiente al kernel $16 \times 16$, la exactitud global descendió a aproximadamente 0.85, confirmando que el desenfoque intenso deteriora la información visual discriminativa necesaria para clasificar adecuadamente escenas complejas o visualmente similares.

#### *E.4.4. Discusión parcial*

En términos generales, los resultados obtenidos con *kernels* promedio muestran que el modelo propuesto es relativamente tolerante a niveles leves de desenfoque, pero su desempeño disminuye conforme aumenta la severidad del suavizado espacial. Este hallazgo complementa los resultados observados en los experimentos de reducción de resolución y ruido AWGN.

### E.5. Discusión general del análisis de robustez

A partir de los resultados obtenidos, se observa que el modelo propuesto mantiene un desempeño satisfactorio bajo perturbaciones leves, tanto en escenarios de reducción moderada de resolución como ante niveles altos de SNR y desenfoque suave por *kernels* pequeños. Sin embargo, conforme aumenta la severidad de la degradación, el rendimiento disminuye de manera progresiva.

En conjunto, estos resultados permiten concluir que el modelo propuesto es funcional bajo perturbaciones suaves, pero no presenta una robustez suficiente frente a degradaciones intensas. Las mayores dificultades se observan en escenas que dependen de bordes, textura fina y estructura espacial, lo cual sugiere que el clasificador extrae una parte importante de su capacidad discriminativa a partir de dichos rasgos visuales.

# Apéndice F. Fundamentos complementarios de aprendizaje profundo para visión

Este apéndice reúne desarrollos teóricos complementarios sobre preprocesamiento, formulación multiclase, convolución, *padding* y aprendizaje por transferencia.

## F.1. Preprocesamiento de imágenes

El preprocesamiento busca convertir imágenes en una representación numérica consistente para el entrenamiento de modelos de aprendizaje profundo. En términos generales, incluye operaciones como redimensionamiento y normalización, con el propósito de controlar variaciones de escala y asegurar compatibilidad dimensional con la arquitectura utilizada (Patel, 2023).

Una práctica común en imágenes RGB es escalar intensidades a un rango numérico controlado (por ejemplo, $[0,1]$) mediante normalización lineal. Sea $x \in \{0,\dots,255\}^{H\times W\times C}$ una imagen RGB; una normalización simple puede definirse como

$$\tilde{x} = \frac{x}{255}, \tag{1}$$

de modo que $\tilde{x} \in [0,1]^{H\times W\times C}$.

Para evaluación experimental, es habitual dividir los datos en subconjuntos de entrenamiento, validación y prueba, aplicando estratificación cuando se desea preservar la proporción de clases por partición (scikit-learn developers, s.f.-b). En problemas multiclase con $K$ clases, las etiquetas suelen representarse mediante codificación *one-hot*, es decir, $y \in \{0,1\}^K$ con $\sum_{k=1}^{K} y_k = 1$.

En este contexto, el modelo produce *logits* $z \in \mathbb{R}^K$ y se aplica *softmax* para obtener proba-

bilidades:

$$p(y = k \mid x) = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}. \tag{2}$$

El entrenamiento se realiza con funciones de pérdida basadas en entropía cruzada categórica (TensorFlow Developers, s.f.-a), que para un conjunto de $N$ muestras puede expresarse como

$$\mathscr{L}_{CE} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log \big(p(y = k \mid x_i)\big). \tag{3}$$

***Aplicación en este trabajo.*** Estos conceptos sustentan el *pipeline* de preparación de datos y la formulación multiclase del entrenamiento supervisado del modelo.

### F.2. Convolución 2D y mapas de características

Las redes neuronales convolucionales (CNN) construyen representaciones jerárquicas aplicando operaciones locales sobre la imagen. En una capa convolucional 2D, se aprende un conjunto de filtros (*kernels*) que se desplazan sobre la entrada y generan mapas de características (*feature maps*), cuya profundidad corresponde al número de filtros. Este mecanismo permite capturar patrones desde descriptores simples hasta estructuras más abstractas en capas profundas (Zhao y cols., 2024). En la práctica, una convolución 2D se parametriza por número de filtros, tamaño de *kernel* y *stride*, pudiendo incluir una función de activación (Keras Team, s.f.-a).

De forma general, sea $X \in \mathbb{R}^{H \times W \times C}$ la entrada con $C$ canales. Para un filtro $m$ con pesos $W^{(m)} \in \mathbb{R}^{h \times w \times C}$ y sesgo $b^{(m)}$, la salida (antes de activación) en la posición espacial $(i, j)$ puede expresarse como

$$Z_{i,j}^{(m)} = \sum_{c=1}^{C} \sum_{u=1}^{h} \sum_{v=1}^{w} W_{u,v,c}^{(m)} X_{i+u,\, j+v,\, c} + b^{(m)}. \tag{4}$$

### F.3. *Padding* y control de dimensionalidad

El *padding* controla la dimensión espacial de la salida. El modo `valid` no utiliza relleno, mientras que `same` aplica relleno para conservar el tamaño espacial cuando *stride* es 1, evitando pérdidas tempranas de resolución (Keras Team, s.f.-a). Esta decisión es relevante cuando se busca mantener información local en etapas iniciales de extracción.

De manera ilustrativa, para una convolución 2D con *stride* $s$, tamaño de *kernel* $k$ y *padding* $p$, el tamaño espacial de salida se relaciona con la entrada como

$$H_{\text{out}} = \left\lfloor \frac{H + 2p - k}{s} \right\rfloor + 1, \qquad W_{\text{out}} = \left\lfloor \frac{W + 2p - k}{s} \right\rfloor + 1. \tag{5}$$

### F.4. Aprendizaje por transferencia (contexto y alcance)

El aprendizaje por transferencia (*transfer learning*) es una estrategia frecuente cuando el dominio objetivo dispone de menos datos etiquetados: se aprovechan representaciones aprendidas en un dominio fuente y luego se adaptan a la tarea objetivo. En clasificación de escenas en teledetección, se ha reportado que el ajuste fino de CNN preentrenadas puede mejorar el desempeño, particularmente en conjuntos relativamente pequeños (Hu y cols., 2015).

***Nota de alcance.*** En esta tesis, el aprendizaje por transferencia se presenta como contexto del estado del arte y alternativa metodológica; no obstante, el enfoque principal se centra en el diseño y evaluación de un *pipeline* reproducible con una CNN supervisada bajo el protocolo

experimental definido.

# Apéndice G. Fundamentos de interpretabilidad mediante mapas de saliencia

Este apéndice presenta los fundamentos teóricos de los mapas de saliencia como herramienta de interpretabilidad en modelos de visión.

## G.1. Concepto y propósito

En modelos de visión basados en redes neuronales convolucionales, los mapas de saliencia (*saliency maps*) buscan identificar qué regiones de la entrada influyen con mayor intensidad en la predicción del modelo. En su formulación clásica, se cuantifica la sensibilidad de una puntuación de clase respecto a pequeñas variaciones en la imagen de entrada, mediante gradientes calculados por diferenciación automática (Simonyan, Vedaldi, y Zisserman, 2013). Este tipo de explicación resulta útil para: (i) inspeccionar la evidencia visual utilizada por el clasificador, (ii) analizar patrones de error, y (iii) complementar el análisis cuantitativo por clase con evidencia cualitativa.

## G.2. Saliencia basada en gradiente respecto a la entrada

Sea $f_\theta(\cdot)$ un modelo de clasificación multiclase con parámetros $\theta$, y sea $z(x) \in \mathbb{R}^K$ el vector de puntuaciones por clase para una imagen $x \in \mathbb{R}^{H\times W\times C}$. Para una clase objetivo $c \in \{1,\dots,K\}$, el mapa de saliencia se define a partir del gradiente de la puntuación $z_c(x)$ respecto a la entrada:

$$G_c(x) = \frac{\partial z_c(x)}{\partial x}. \tag{6}$$

Dado que en imágenes RGB ($C = 3$) el gradiente es un tensor tridimensional, es habitual reducir la dimensión de canales para obtener un mapa bidimensional. Una agregación común con-

siste en tomar el máximo del valor absoluto sobre canales:

$$S_c(x)_{i,j} = \max_{k \in \{1,\ldots,C\}} \left| G_c(x)_{i,j,k} \right| . \tag{7}$$

Para visualización, el mapa se normaliza y se superpone sobre la imagen original como *heatmap*.

### G.3. Implementación mediante diferenciación automática

La obtención de $G_c(x)$ se implementa con diferenciación automática usando `tf.GradientTape`, registrando las operaciones del modelo y calculando el gradiente de la puntuación de la clase objetivo respecto a la entrada (TensorFlow Developers, s.f.-b). La clase objetivo puede definirse como la clase predicha para explicar una decisión concreta, o como la clase real para analizar fallos de clasificación.

### G.4. Limitaciones y buenas prácticas

Los mapas basados en gradiente describen sensibilidad local y no establecen causalidad; además, pueden depender de normalizaciones y de la elección de salida. Por ello, se recomienda interpretar resultados junto con casos representativos y, cuando sea pertinente, considerar verificaciones metodológicas (*sanity checks*) (Adebayo y cols., 2018) o extensiones como SmoothGrad para reducir ruido visual (Smilkov, Thorat, Kim, Viégas, y Wattenberg, 2017).

***Aplicación en este trabajo.*** La interpretabilidad mediante saliencia se incorpora para complementar la evaluación cuantitativa, ayudando a explicar confusiones entre clases y a inspeccionar si el modelo atiende a regiones coherentes con la escena.

# Apéndice H. Fundamentos de robustez en modelos de visión

Este apéndice reúne el marco conceptual asociado a la robustez de modelos de visión frente a perturbaciones controladas.

## H.1. Perturbaciones frecuentes y su interpretación

La robustez en modelos de visión se refiere a la estabilidad del desempeño cuando la entrada presenta variaciones o perturbaciones que no deberían cambiar la etiqueta semántica. En escenarios reales, estas perturbaciones pueden provenir de cambios de resolución, degradaciones de adquisición, ruido o procesos de compresión y filtrado. En visión por computador, se han propuesto *benchmarks* de corruptelas comunes para evaluar sistemáticamente este tipo de degradaciones y su efecto sobre modelos entrenados (Hendrycks y Dietterich, 2019).

***Cambios de escala y redimensionamiento.*** En imágenes satelitales es común encontrar variaciones de escala efectiva. El redimensionamiento puede suavizar texturas, alterar bordes y modificar patrones espaciales finos, lo cual afecta la separabilidad de escenas con detalles sutiles.

***Ruido aditivo.*** El ruido introduce perturbaciones de alta frecuencia que pueden degradar señales texturales. Evaluar el comportamiento del modelo bajo ruido permite estimar sensibilidad a degradaciones y a variaciones de calidad de imagen.

***Filtrado y kernels.*** Filtros de suavizado o realce pueden modificar la distribución de frecuencias de la imagen. En clasificación de escenas, esto puede cambiar la percepción de textura y contornos, afectando clases donde esos patrones son discriminativos.

### H.2. Criterios de análisis

Un análisis de robustez puede basarse en: (i) comparar métricas globales antes y después de la perturbación, (ii) analizar qué clases se degradan más, y (iii) inspeccionar si las confusiones aumentan en pares de clases visualmente cercanas. Este enfoque permite complementar la métrica global con un diagnóstico por categoría.

***Aplicación en este trabajo.*** El concepto de robustez se usa como fundamento para evaluar la estabilidad del clasificador ante perturbaciones controladas, reportando su impacto con métricas estándar y análisis por clase.

# Apéndice I. Implementación adaptada y evaluación ampliada del enfoque DSFATN

Este apéndice presenta el desarrollo ampliado de la implementación adaptada inspirada en el modelo *Deep Salient Feature Based Anti-Noise Transfer Network* (DSFATN), así como sus resultados y discusión detallada.

## I.1. Implementación adaptada del enfoque DSFATN

En esta tesis no se desarrolló una réplica exacta del modelo DSFATN original, sino una implementación adaptada de sus ideas principales al entorno experimental del trabajo. En consecuencia, los resultados obtenidos deben interpretarse como una referencia comparativa orientativa inspirada en la literatura, y no como una validación estricta del método reportado por Gong y cols. (2018).

La adaptación realizada conservó tres elementos fundamentales del enfoque original: el uso del conjunto UC Merced como base de trabajo, la extracción de representaciones profundas mediante VGG-19 preentrenada en ImageNet y la incorporación de una restricción anti-ruido dentro de la función de entrenamiento. Sin embargo, varios componentes del método original no fueron reproducidos de manera completa.

En primer lugar, el módulo de saliencia no fue replicado de forma estricta. El artículo original construye la saliencia mediante un esquema PBVS derivado de GBVS, apoyado en mapas de activación, normalización basada en grafos y una cadena de Markov. En la implementación desarrollada en esta tesis, la saliencia se obtuvo mediante un procedimiento simplificado y heurístico. Aunque se mantuvo la idea general de extraer parches relevantes, no se reprodujeron exactamente

ni el algoritmo PBVS ni sus ecuaciones originales.

En segundo lugar, la estrategia de transferencia fue reproducida solo parcialmente. El artículo utiliza VGG-19 preentrenada en ImageNet y toma la activación de la primera capa totalmente conectada (*fc1*, 4096 dimensiones) como representación profunda, aspecto que sí fue preservado en la implementación. No obstante, la comparación entre múltiples CNN preentrenadas reportada en el artículo no fue reproducida en esta tesis.

En tercer lugar, la función de pérdida original tampoco fue reproducida de manera exacta. La implementación desarrollada debe entenderse como una aproximación inspirada en la formulación original, y no como una reproducción idéntica del artículo.

Finalmente, el protocolo experimental del paper tampoco fue reproducido en toda su extensión. El trabajo original evalúa el método en tres conjuntos de datos, utiliza validación cruzada de cinco particiones, analiza distintos niveles de ruido y compara variantes como TN-1 y TN-2. En esta tesis se trabajó únicamente con UCM y no se reprodujeron de forma íntegra los experimentos multiescala ni la evaluación en los otros datasets.

Con el fin de delimitar con mayor precisión el alcance metodológico de la comparación, la Tabla 9 resume qué componentes del enfoque original fueron reproducidos, cuáles fueron adaptados y cuáles no se implementaron en esta tesis.

**Tabla 9.**

*Correspondencia entre el enfoque DSFATN original y la adaptación implementada en esta tesis.*

| **Componente del enfoque original** | **Estado en esta tesis** | **Observación** |
|---|---|---|
| Clasificación multiclase de escenas de percepción remota | Reproducido | Se mantuvo el objetivo general de clasificación de escenas satelitales. |

*Continúa en la siguiente página*

*Tabla 9 (continuación)*

| **Componente del enfoque original** | **Estado en esta tesis** | **Observación** |
|---|---|---|
| Conjunto de datos UC Merced | Reproducido | Se trabajó con UCM como base experimental, en concordancia con una de las bases utilizadas en el artículo original. |
| Extracción profunda con VGG-19 preentrenada en ImageNet | Reproducido parcialmente | Se utilizó VGG-19 como extractor fijo de características profundas, tomando como referencia la representación asociada a FC1. No se reprodujo la comparación entre múltiples CNN preentrenadas reportada en el artículo. |
| Módulo de saliencia PBVS derivado de GBVS | No reproducido estrictamente | La detección de regiones salientes y el muestreo de parches se implementaron mediante un procedimiento simplificado y heurístico, inspirado en la idea general del paper, pero sin replicar el algoritmo PBVS ni sus ecuaciones originales. |
| Muestreo de parches salientes multiescala | Adaptado | Se conservaron decisiones inspiradas en el artículo, como el uso de $\alpha = 9$ parches y escalamiento aproximado entre el 30 % y el 80 % de la región saliente, aunque el procedimiento de selección no fue idéntico al original. |
| Representación DSF por parche | Adaptado | Los vectores extraídos de los parches se agregaron en un único vector por escena, lo cual constituye una aproximación práctica, pero no una reproducción literal de la formulación original a nivel de parche. |
| *Joint loss* con término de clasificación y restricción anti-ruido | Reproducido parcialmente | Se mantuvo la idea de combinar clasificación con una restricción anti-ruido, pero la implementación concreta de la pérdida corresponde a una aproximación funcional y no a una réplica exacta de la formulación original. |
| Ubicación exacta de la restricción anti-ruido dentro de la red | Adaptado | En esta tesis la restricción se implementó sobre la representación oculta de FC1, decisión consistente con la interpretación adoptada del artículo, aunque el paper presenta ambigüedad entre la descripción textual y la formulación matemática. |
| Evaluación en UCM, SIRI-WHU y SAT-6 | No reproducido completamente | La implementación se evaluó únicamente sobre UC Merced. No se reprodujo la evaluación en los otros dos conjuntos de datos. |

*Continúa en la siguiente página*

*Tabla 9 (continuación)*

| **Componente del enfoque original** | **Estado en esta tesis** | **Observación** |
|---|---|---|
| Validación cruzada de cinco particiones | Reproducido parcialmente | Se mantuvo el esquema general de cinco *folds*, pero se añadió una partición interna de validación dentro del protocolo experimental de esta tesis. |
| Análisis multiescala y variantes TN-1 / TN-2 | No reproducido | No se reprodujeron los experimentos multiescala completos ni la comparación formal con las variantes reportadas en el artículo original. |

*Nota*. Elaboración propia a partir del artículo de Gong y cols. (2018) y de la implementación adaptada desarrollada en esta tesis.

En síntesis, la implementación desarrollada en esta tesis debe entenderse como una adaptación funcional orientada por el enfoque DSFATN, útil como referencia comparativa, pero limitada frente al alcance metodológico del trabajo original.

### I.2. Resultados del modelo robusto adaptado

La Tabla 10 resume los resultados obtenidos por la implementación adaptada inspirada en DSFATN sobre el conjunto original y en los escenarios perturbados mediante reducción de resolución, ruido AWGN y desenfoque por *kernels* promedio.

**Tabla 10.**

*Resultados globales del modelo robusto adaptado basado en DSFATN.*

| **Escenario** | **Accuracy promedio (%)** | **Desv. estándar (%)** |
|---|---|---|
| Original | 81.95 | 1.08 |
| Resolución $128 \times 128$ | 82.33 | 0.70 |
| Resolución $64 \times 64$ | 74.96 | 1.39 |
| Resolución $32 \times 32$ | 63.10 | 1.56 |
| SNR 30 dB | 80.67 | 1.23 |
| SNR 20 dB | 76.71 | 0.49 |
| SNR 10 dB | 63.00 | 1.84 |
| SNR 5 dB | 50.90 | 1.67 |
| Kernel $2 \times 2$ | 82.67 | 1.20 |
| Kernel $4 \times 4$ | 82.43 | 0.74 |
| Kernel $8 \times 8$ | 74.95 | 0.46 |
| Kernel $16 \times 16$ | 66.10 | 2.29 |

En condiciones originales, el modelo robusto adaptado alcanzó una exactitud promedio de 81.95 %, con una desviación estándar de 1.08 %. Este resultado muestra que la implementación desarrollada conserva una capacidad de clasificación funcional sobre UC Merced dentro del protocolo experimental adoptado en esta tesis. No obstante, su desempeño permanece por debajo del modelo propuesto desarrollado previamente.

En la reducción de resolución, el modelo mantuvo un desempeño cercano al escenario nominal para $128 \times 128$, pero la exactitud descendió de forma importante en $64 \times 64$ y $32 \times 32$. Frente al ruido AWGN, el rendimiento disminuyó progresivamente a medida que se redujo la relación señal a ruido. En el caso del desenfoque por *kernels* promedio, los resultados fueron favorables para perturbaciones leves, pero empeoraron en condiciones de desenfoque severo.

### I.3. Discusión

Los resultados obtenidos indican que la implementación adaptada inspirada en DSFATN conserva un desempeño relativamente estable ante perturbaciones leves. Sin embargo, el rendimiento disminuye de forma marcada cuando la degradación se vuelve severa, particularmente en los escenarios de SNR 5 dB, resolución $32 \times 32$ y *kernel* $16 \times 16$.

Las clases con mayor afectación se asociaron principalmente con escenas residenciales, intersecciones, canchas, edificios y patrones urbanos complejos. Esto sugiere que, aun cuando la implementación incorpora una estrategia anti-ruido, su capacidad de discriminación sigue dependiendo en buena medida de la preservación de información espacial fina.

Bajo las condiciones experimentales definidas en esta tesis, la adaptación inspirada en DSFATN no alcanzó el desempeño global del modelo propuesto. Sin embargo, esta observación debe entenderse en el contexto de una implementación parcial del enfoque original. Por tanto, el valor principal de este apéndice radica en aportar una referencia comparativa orientativa basada en la literatura, más que en establecer una jerarquía definitiva entre ambos métodos.

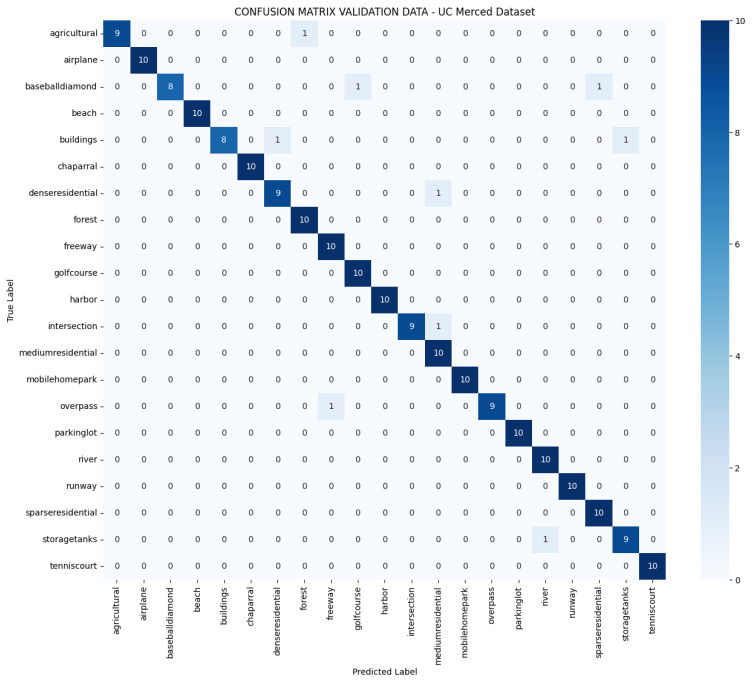
En este sentido, los resultados permiten situar el desempeño del modelo propuesto frente a una implementación funcional inspirada en DSFATN y, al mismo tiempo, ponen de relieve la importancia de reproducir con alto grado de fidelidad los módulos originales cuando se pretende realizar comparaciones metodológicas estrictas.

## Apéndice J. Matrices de confusión complementarias

Este apéndice reúne las matrices de confusión de validación y entrenamiento como apoyo al análisis de desempeño del modelo.

**Figura 9.**

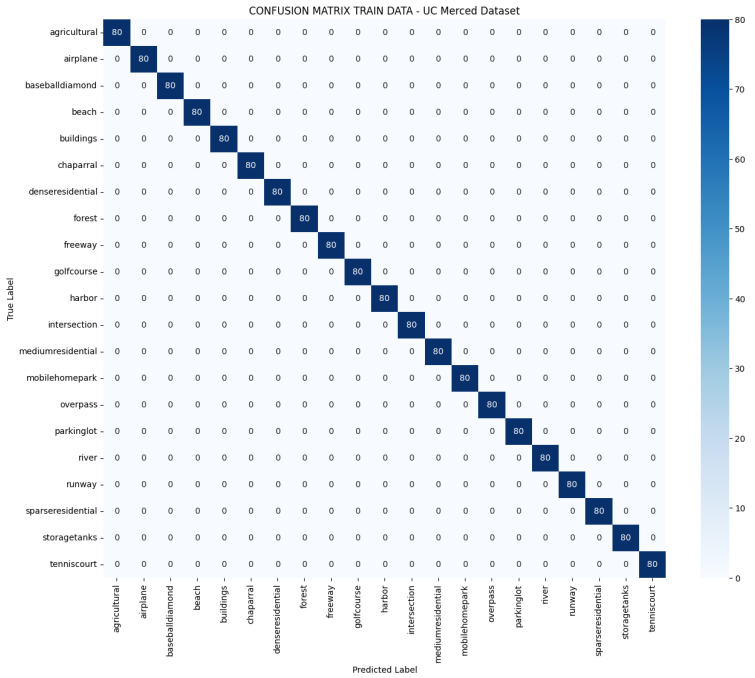
*Matriz de confusión del modelo en validación.*




*Nota.* La figura presenta la matriz de confusión obtenida por el modelo final sobre el conjunto de validación. Esta representación permite analizar la correspondencia entre las clases reales y las clases predichas durante la etapa de ajuste y seguimiento del desempeño del modelo.

**Figura 10.**

*Matriz de confusión del modelo en entrenamiento.*




*Nota.* La figura presenta la matriz de confusión obtenida por el modelo final sobre el conjunto de entrenamiento. Esta matriz permite verificar el comportamiento del modelo frente a las muestras utilizadas durante el proceso de aprendizaje e identificar posibles patrones de clasificación correcta o confusión entre clases.

### Análisis complementario

El análisis conjunto de estas matrices permite evidenciar un comportamiento consistente del modelo entre los distintos subconjuntos. En particular, se observa una alta concentración de valores en la diagonal principal en el conjunto de entrenamiento, lo que indica un ajuste elevado del modelo. En validación se presentan ligeras dispersiones fuera de la diagonal, coherentes con el proceso de generalización frente a datos no vistos durante el entrenamiento.

## Apéndice K. Métricas por clase del modelo en el conjunto de prueba

Este apéndice presenta la tabla detallada de métricas por clase del modelo sobre el conjunto de prueba.

**Tabla 11.**

*Métricas por clase del modelo en el conjunto de prueba.*

| Clase | Precision | Recall | F1-score | Support |
|---|---|---|---|---|
| agricultural | 1.00 | 1.00 | 1.00 | 10 |
| airplane | 1.00 | 1.00 | 1.00 | 10 |
| baseballdiamond | 1.00 | 0.80 | 0.89 | 10 |
| beach | 1.00 | 1.00 | 1.00 | 10 |
| buildings | 0.69 | 0.90 | 0.78 | 10 |
| chaparral | 0.91 | 1.00 | 0.95 | 10 |
| denseresidential | 1.00 | 0.90 | 0.95 | 10 |
| forest | 0.71 | 1.00 | 0.83 | 10 |
| freeway | 1.00 | 1.00 | 1.00 | 10 |
| golfcourse | 1.00 | 1.00 | 1.00 | 10 |
| harbor | 1.00 | 1.00 | 1.00 | 10 |
| intersection | 0.90 | 0.90 | 0.90 | 10 |
| mediumresidential | 0.90 | 0.90 | 0.90 | 10 |
| mobilehomepark | 1.00 | 1.00 | 1.00 | 10 |
| overpass | 1.00 | 0.90 | 0.95 | 10 |
| parkinglot | 1.00 | 1.00 | 1.00 | 10 |
| river | 1.00 | 0.70 | 0.82 | 10 |
| runway | 1.00 | 1.00 | 1.00 | 10 |
| sparseresidential | 1.00 | 0.80 | 0.89 | 10 |
| storagetanks | 0.82 | 0.90 | 0.86 | 10 |
| tenniscourt | 1.00 | 1.00 | 1.00 | 10 |
| Accuracy | | 0.94 | | 210 |
| Macro avg | 0.95 | 0.94 | 0.94 | 210 |
| Weighted avg | 0.95 | 0.94 | 0.94 | 210 |

## Apéndice L. Repositorio del proyecto y material complementario

Este apéndice reúne el repositorio de GitHub con los recursos computacionales empleados en el desarrollo experimental de la tesis.

Como material complementario del trabajo de grado, se consolidó un repositorio en GitHub con los recursos computacionales empleados durante el desarrollo experimental de la tesis. Este repositorio incluye archivos relacionados con la preparación del conjunto de datos, la implementación del modelo base, el entrenamiento, la evaluación del desempeño, el análisis de robustez, la generación de imágenes perturbadas y la documentación general del proyecto.

El repositorio del proyecto se encuentra disponible públicamente en GitHub bajo el nombre *Satellite-image-classification-using-deep-learning*, en la siguiente dirección electrónica:

`https://github.com/AndreaContre14/Satellite-image-classification-using-deep-learning`

La disponibilidad de este material fortalece la trazabilidad de los resultados presentados y favorece la reproducibilidad del procedimiento experimental desarrollado.