



UNIVERSIDAD DE CONCEPCIÓN  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL



Comparación de modelos de aprendizaje profundo para la clasificación y  
segmentación de áreas de interés forestal

POR

Ricardo Antonio Hernández Alarcón

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción  
para optar al título profesional de Ingeniero Civil Industrial

Profesor Guía

Guillermo Cabrera Vives, Ph.D.

Octubre, 2022

Concepción, Chile

©2022 Ricardo Antonio Hernández Alarcón

Ninguna parte de esta tesis puede reproducirse o transmitirse bajo ninguna forma o por ningún medio o procedimiento, sin permiso por escrito del autor.

# Resumen

En la industria forestal existe una necesidad latente de monitorear las superficies de cultivo con el objetivo de segmentar y clasificar los distintos tipos de suelo para la toma de decisiones corporativas respecto al uso de estas áreas. Actualmente, se utilizan técnicas manuales para el desarrollo de estas actividades, y para grandes extensiones de terrenos resultan ser poco eficientes y con altos costos en tiempo y recursos. Sin embargo, los avances en aprendizaje profundo han mostrado tener un gran potencial para el desarrollo de soluciones automatizadas, transformándose en el estado del arte para estas tareas. Por esto, el objetivo de esta memoria de título es implementar, comparar y evaluar dos modelos de aprendizaje profundo, Fully Convolutional Network y Multi-path Residual Network, para la tarea de clasificación y segmentación de los distintos tipos de áreas de interés forestal como cultivos de distintas edades, tanto de pino como de eucaliptos, bosques nativos y caminos. Esto a partir de imágenes aéreas georeferenciadas de predios previamente etiquetados. Los principales resultados muestran que no existen grandes diferencias entre los rendimientos de un modelo sobre el otro. Cada modelo resultó ser mejor en dos de los cuatro predios forestales del experimento. Se concluye que ambos modelos tienen un gran potencial para la clasificación de uso de suelo, por sobre la evidencia de rendimientos deficientes en algunas categorías en predios específicos. Debido a que estos ocurren en presencia de una gran similitud natural entre categorías y/o un desbalance de categorías desfavorable, a pesar de los esfuerzos por mitigar los efectos del desbalance mediante la técnica de aumentación de datos “Copy-Paste”.

# Abstract

In the forestry industry there is a latent need to monitor crop areas in order to segment and classify the different types of soil for corporate decision making regarding the use of these areas. Currently, manual techniques are used for the development of these activities, and for large extensions of land they are inefficient and costly in terms of time and resources. However, advances in deep learning have shown great potential for the development of automated solutions, becoming the state of the art for these tasks. Therefore, the objective of this report is to implement, compare and evaluate two deep learning models, Fully Convolutional Network and Multi-path Residual Network, for the task of classification and segmentation of different types of forest areas of interest such as crops of different ages, both pine and eucalyptus, native forests and roads. This is based on geo-referenced aerial images of previously labeled properties. The main results show that there are no major differences between the yields of one model over the other. Each model proved to be better in two of the four forest plots of the experiment. It is concluded that both models have great potential for land use classification, over the evidence of poor yields in some categories in specific plots. Because these occur in the presence of a high natural similarity between categories and/or an unfavorable category imbalance, despite efforts to mitigate the effects of the imbalance using the “Copy-Paste” data augmentation technique.

---

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Revisión Bibliográfica . . . . .	2
1.2. Objetivos . . . . .	3
1.3. Organización de la memoria de título . . . . .	4
<b>2. Marco Teórico</b>	<b>5</b>
2.1. Inteligencia artificial . . . . .	5
2.1.1. Aprendizaje Automático . . . . .	6
2.2. Aprendizaje profundo . . . . .	8
2.2.1. Redes Neuronales . . . . .	8
2.2.2. Redes Neuronales convolucionales . . . . .	12
2.2.3. Autoencoders . . . . .	15
2.3. Segmentación semántica . . . . .	17
2.3.1. Fully Convolutional Network . . . . .	18
2.3.2. Multi-path Residual Network . . . . .	20
2.4. Técnicas en aprendizaje profundo . . . . .	22
2.4.1. K-Fold Cross Validation . . . . .	22
2.4.2. Stop-Early . . . . .	22
2.4.3. Data Augmentation . . . . .	23
2.4.4. Fine-Tuning . . . . .	24
<b>3. Materiales y métodos</b>	<b>26</b>
3.1. Datos . . . . .	26

---

3.2. Preprocesamiento de datos . . . . .	28
3.3. Métodos . . . . .	31
3.4. Metodología . . . . .	31
3.5. Hiperparámetros . . . . .	32
3.6. Entrenamiento . . . . .	32
3.7. Validación . . . . .	33
3.8. Evaluación . . . . .	33
<b>4. Resultados experimentales y discusión</b>	<b>35</b>
4.1. Segmentación semántica . . . . .	35
4.1.1. Predio 1 . . . . .	36
4.1.2. Predio 2 . . . . .	37
4.1.3. Predio 3 . . . . .	38
4.1.4. Predio 4 . . . . .	39
4.2. Observaciones adicionales . . . . .	40
4.3. Visualización de predicciones . . . . .	41
<b>5. Conclusión</b>	<b>43</b>
<b>Anexo</b>	<b>49</b>
A.1. Búsqueda de hiperparámetros . . . . .	49
A.2. Gráficos de entrenamiento . . . . .	50
A.3. Matriz de confusión . . . . .	55
A.4. F1 score, Precision y Recall . . . . .	62
A.5. Predicciones . . . . .	64
A.6. Resumen FI . . . . .	68

# Índice de Figuras

2.1. Subáreas de la Inteligencia Artificial. . . . .	6
2.2. Técnicas de aprendizaje automático. . . . .	7
2.3. Diagrama de procesos de ejecución de un algoritmo de aprendizaje supervisado. . . . .	7
2.4. Concepto del perceptrón . . . . .	8
2.5. Perceptrón multicapa con una sola capa oculta. . . . .	9
2.6. Derivada de la función de pérdida $E$ en dos puntos distintos de la curva. . . . .	10
2.7. Propagación hacia adelante. . . . .	11
2.8. Retropropagación del error. . . . .	12
2.9. Ejemplo de una red convolucional alternando capas de convolución y pooling. . . . .	13
2.10. Proceso de convolución. . . . .	14
2.11. Ejemplo de agrupación máxima. . . . .	14
2.12. Estructura codificador-decodificador. . . . .	15
2.13. Codificador-decodificador convolucional. . . . .	15
2.14. La operación de convolución transpuesta. . . . .	16
2.15. Ejemplo de max pooling y max unpooling. . . . .	16
2.16. Ejemplo de bilinear interpolation. . . . .	17
2.17. Tareas de la visión por computador. . . . .	17
2.18. Estructura de la FCN-32s. . . . .	18
2.19. FCN-32s detallada. . . . .	19
2.20. Fusión para FCN-16s y FCN-8s. . . . .	19
2.21. Estructura codificador-decodificador y conexiones de salto a través del cuello de botella. . . . .	20

---

2.22. Comparación de distintas variaciones de la FCN. . . . .	20
2.23. Arquitectura MP-ResNet. . . . .	21
2.24. Bloque de deconvolución de características del MP-ResNet. . . . .	21
2.25. 10-Fold Cross Validation. . . . .	22
2.26. Ejemplo de Stop-Early. . . . .	23
2.27. Aumentaciones de datos estandar. Fuente: Géron (2019). . . . .	24
2.28. Aumentación Copy-Paste. Fuente: Ghiasi et al. (2021). . . . .	24
2.29. Proceso Fine-Tuning. Fuente: Vogado et al. (2021). . . . .	25
3.1. Predios forestales. . . . .	27
3.2. Imagen satelital del predio 1 y la visualización de las áreas etiquetadas según el tipo de uso de suelo. . . . .	28
3.3. Extractos etiquetados de los predios forestales. . . . .	29
3.4. División del predio 1 en ventanas de $448 \times 448$ pixeles. . . . .	30
3.5. Subdivisión con solape de las ventanas. . . . .	31
4.1. Matriz de confusión sobre el conjunto de test del predio 1, FCN iteración 1. . . . .	37
4.2. Matriz de confusión sobre el conjunto de test del predio 2, MP-ResNet iteración 4. . . . .	38
4.3. Matriz de confusión sobre el conjunto de test del predio 3, MP-ResNet iteración 2. . . . .	39
4.4. Matriz de confusión sobre el conjunto de test del predio 4, FCN iteración 1. . . . .	40
4.5. Predicciones: (a) Predio 1; (b) Predio 2; (c) Predio 3; (d) Predio 4. . . . .	42
A.1. Desempeño de los hiperparámetros sobre el conjunto de validación. . . . .	49
A.2. Entrenamiento: Predio 1 — FCN. . . . .	50
A.3. Entrenamiento: Predio 1 — MP-ResNet. . . . .	51
A.4. Entrenamiento: Predio 2 — FCN. . . . .	51
A.5. Entrenamiento: Predio 2 — MP-ResNet. . . . .	52
A.6. Entrenamiento: Predio 3 — FCN. . . . .	52
A.7. Entrenamiento: Predio 3 — MP-ResNet. . . . .	53
A.8. Entrenamiento: Predio 4 — FCN. . . . .	53
A.9. Entrenamiento: Predio 4 — MP-ResNet. . . . .	54



---

A.10.Matriz de confusión — Predio 1 — FCN. . . . .	55
A.11.Matriz de confusión — Predio 1 — MP-ResNet. . . . .	56
A.12.Matriz de confusión — Predio 2 — FCN. . . . .	57
A.13.Matriz de confusión — Predio 2 — MP-ResNet. . . . .	58
A.14.Matriz de confusión — Predio 3 — FCN. . . . .	59
A.15.Matriz de confusión — Predio 3 — MP-ResNet. . . . .	60
A.16.Matriz de confusión — Predio 4 — FCN. . . . .	61
A.17.Matriz de confusión — Predio 4 — MP-ResNet. . . . .	61
A.18.Predicciones: Predio 1. . . . .	64
A.19.Predicciones: Predio 2. . . . .	65
A.20.Predicciones: Predio 3. . . . .	66
A.21.Predicciones: Predio 4. . . . .	67

---

# Índice de Tablas

3.1. Taxonomía utilizada. . . . .	28
3.2. Distribución de categorías. . . . .	29
3.3. Búsqueda de hiperparámetros. . . . .	32
4.1. Cross validation de las métricas sobre el conjunto de test. . . . .	36
4.2. Cross validation F1 (%) de los modelos sobre el conjunto de test. . . . .	36
A.1. Desempeño de los hiperparámetros. . . . .	49
A.2. Predio 1—F1 score, Precision y Recall. . . . .	62
A.3. Predio 2—F1 score, Precision y Recall. . . . .	62
A.4. Predio 3—F1 score, Precision y Recall. . . . .	62
A.5. Predio 4—F1 score, Precision y Recall. . . . .	63

# Capítulo 1

## Introducción

La industria forestal en Chile es uno de los sectores productivos de gran importancia del país, en el año 2017 representó el 1.9% del PIB nacional. A diciembre de 2018 reporta un total de 2.303.886 hectáreas de superficie de plantaciones forestales acumulada (Cardemil, 2021). Esta industria tiene una necesidad latente de monitorear las superficies de cultivo con el objetivo de segmentar y clasificar los distintos tipos de suelo para la toma de decisiones corporativas respecto al uso de estas áreas. Actualmente, se utilizan técnicas manuales para el desarrollo de estas actividades, las que para grandes extensiones de terrenos resultan ser poco eficientes y con altos costos en tiempo y recursos. Por ello, es de interés reemplazar las técnicas actuales por soluciones automatizadas.

En este contexto, la Unidad de Data Science de la Universidad de Concepción se encuentra desarrollando Deep-Hub, una plataforma basada en inteligencia artificial para el análisis de imágenes aéreas de interés forestal. Esta plataforma permite el conteo automático de árboles desde imágenes de drones, clasificación de uso de suelo y detección de cambios de vegetación en el tiempo usando imágenes satelitales.

En Deep-Hub, la tarea de clasificación de uso de suelo está planificada para ser llevada a cabo mediante modelos de aprendizaje profundo de segmentación semántica, los cuales tienen el propósito de determinar que píxeles de una imagen pertenece a una etiqueta o categoría. De esta forma, con estos modelos es posible clasificar y segmentar distintos tipos de áreas de interés, como el tipo de uso de suelo. Dada la gran cantidad de modelos existentes, hay una necesidad

de explorar opciones de modelos, seleccionar los que tengan características más interesantes y/o mayor potencial e implementarlos en un ambiente de investigación con imágenes de interés forestal. Así, poder evaluarlos y compararlos para analizar sus oportunidades de implementación a un nivel productivo.

La presente memoria de título se ha definido por la Unidad de Data Science en apoyo a la toma de decisión, sobre la selección de un modelo para la implementación de este en una etapa productiva para Deep-Hub. Por tanto, este trabajo consiste en la implementación, comparación y evaluación de dos arquitecturas de aprendizaje profundo que son la Fully Convolutional Network (Long et al., 2015) y la Multi-path Residual Network (Ding et al., 2021) en la tarea de clasificación y segmentación de los distintos tipos de áreas de interés forestal, a partir de imágenes satelitales georeferenciadas de predios previamente etiquetados. Las áreas de interés consideran cultivos de distintas edades, tanto de pino como de eucaliptos, bosques nativos y caminos.

## 1.1. Revisión Bibliográfica

En el estado del arte de la segmentación semántica se encuentra una variedad de arquitecturas de aprendizaje profundo con distintas características, algunas con enfoque en el rendimiento como Fully Convolutional Network (FCN; Long et al.), U-Net (Ronneberger et al., 2015), Segnet (Badrinarayanan et al., 2017) y Deeplab (Chen et al., 2017), las cuales han estado presente como referencia en una gran cantidad de estudios. Además, otras arquitecturas como Fast-SCNN (Poudel et al., 2019) y ContextNet (Poudel et al., 2018) tienen un enfoque en la velocidad. Una de las arquitecturas más recientes es la Multi-path Residual Network (MP-ResNet; Ding et al.), que presenta ventajas significativas sobre FCN en imágenes de radar polarimétrico de apertura sintética de alta resolución, las que se caracterizan por tener una gran cantidad de ruido moteado o de manchas.

Las técnicas de aprendizaje profundo sobre imágenes obtenidas por sensores remotos han sido utilizadas para llevar a cabo distintas tareas vinculadas al monitoreo de cultivos, como la detección, clasificación, segmentación por instancia y segmentación semántica de especies. Por ejemplo, en Poblete-Echeverría et al. (2017) consideran una red neuronal para la detección y

segmentación del dosel de la vid a partir de imágenes RGB. En Fricker et al. (2019), aplican una red neuronal convolucional (CNN) para reconocer diferentes especies de pinos, utilizando imágenes RGB e hiperespectrales. En Zhao et al. (2018) aplican una CNN y una totalmente convolucional para la segmentación de las copas de los árboles de granada, utilizando imágenes ópticas. Con métodos similares, en Onishi y Ise (2021) y Surovỳ et al. (2018) se realiza segmentación de las copas de los árboles en bosques nativos y en Wallace et al. (2014) para eucaliptos. En Nguyen et al. (2020) proponen un método de clasificación multitemporal de alta resolución espacial con una red neuronal profunda espacio-temporal-espectral avanzada, con el fin de ubicar arrozales a nivel de pixel durante todo un año y para cada instancia temporal. Finalmente, en M Rustowicz et al. (2019) y Du et al. (2019) aplican aprendizaje profundo para la segmentación semántica de áreas de cultivo de pequeños agricultores.

## 1.2. Objetivos

### Objetivo General

Revisión, implementación y comparación de los modelos de aprendizaje profundo Fully Convolutional Network y Multi-path Residual Network, para la tarea de clasificación y segmentación de áreas de interés forestal.

### Objetivos Específicos

- i) Preprocesar imágenes satelitales georeferenciadas de predios forestales para la obtención de conjuntos de imágenes adecuados para entrenar, validar y testear los modelos.
- ii) Implementar y ajustar los modelos Fully Convolutional Network y Multi-path Residual Network.
- iii) Comparar los desempeños de Fully Convolutional Network y Multi-path Residual Network sobre el conjunto de test, considerando las métricas más importantes.
- iv) Obtener conclusiones a partir de los resultados, en apoyo la toma de decisión respecto a la implementación de Fully Convolutional Network o Multi-path Residual Network en

---

Deep-Hub.

### 1.3. Organización de la memoria de título

El resto de este trabajo está estructurado de la siguiente manera: Primero, en el Capítulo 2, se presenta el marco teórico tras los modelos de aprendizaje profundo de segmentación semántica. Luego, en el Capítulo 3, se presentan los materiales y métodos para la implementación y comparación de los modelos Fully Convolutional Network y Multi-path Residual Network. Posteriormente, en el Capítulo 4, se presentan los resultados experimentales y discusión. Por último, las conclusiones se presentan en el Capítulo 5.

# Capítulo 2

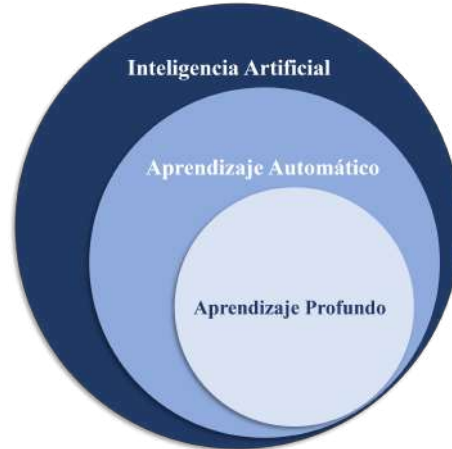
## Marco Teórico

En este capítulo se presentan los tópicos relacionados al aprendizaje profundo y a la segmentación semántica para entender el trabajo desarrollado.

### 2.1. Inteligencia artificial

La inteligencia artificial (IA) es una disciplina académica relacionada con la teoría de la computación cuyo objetivo es emular algunas de las facultades intelectuales humanas en sistemas artificiales. Con inteligencia humana se refiere típicamente a procesos de percepción sensorial (visión, audición, etc) y a sus consiguientes procesos de reconocimiento de patrones, por lo que las aplicaciones más habituales de la IA son el tratamiento de datos y la identificación de sistemas (Benítez et al., 2014).

Dentro de la IA encontramos dos subáreas, Aprendizaje Automático y Aprendizaje Profundo más conocidas por sus nombres en inglés “Machine Learning” y “Deep Learning” respectivamente. Donde el Aprendizaje Profundo es parte del Aprendizaje Automático como se puede apreciar en la Figura 2.1.



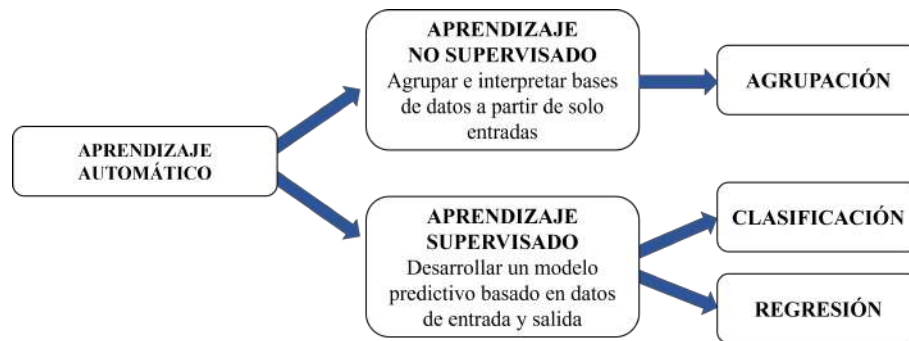
**Figura 2.1:** Subáreas de la Inteligencia Artificial.

### 2.1.1. Aprendizaje Automático

En la segunda mitad del siglo veinte, el aprendizaje automático evolucionó como un subcampo de la IA que involucraba algoritmos de autoaprendizaje, que derivaban el conocimiento a partir de datos para crear predicciones. Así, en lugar de necesitar al hombre para derivar de forma manual las reglas y crear modelos a partir del análisis de grandes cantidades de datos, el aprendizaje automático ofrece una alternativa más eficiente para capturar el conocimiento en datos, mejorar gradualmente el rendimiento de los modelos predictivos y tomar decisiones basadas en esos datos. El aprendizaje automático no solo es cada vez más importante en la investigación científica computacional, sino que juega un papel cada vez más importante en nuestra vida diaria. Gracias al aprendizaje automático, disfrutamos filtros potentes para el correo no deseado, software práctico de reconocimiento de voz y texto, motores de búsqueda fiables, desafiantes programas para jugar el ajedrez y, esperemos que muy pronto, eficientes autos de conducción autónoma (Mirjalili y Raschka, 2020).

El Aprendizaje automático se divide principalmente en dos técnicas, el aprendizaje supervisado y el aprendizaje no supervisado como se observa en el esquema de la Figura 2.2.

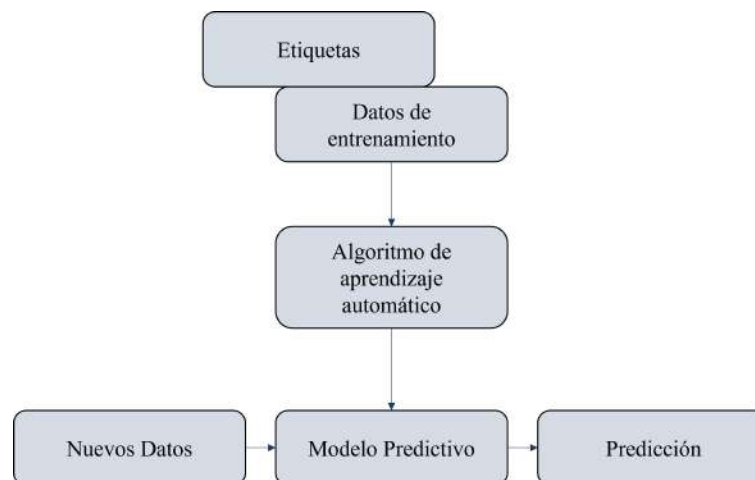




**Figura 2.2:** Técnicas de aprendizaje automático.

### Aprendizaje Supervisado

Los algoritmos de aprendizaje supervisado producen una función que establece una correspondencia entre las entradas y las salidas deseadas del sistema. Estos trabajan con un aprendizaje basado en ejemplos donde el nuevo conocimiento es inducido a partir de una serie de ejemplos y contraejemplos. Un ejemplo de este tipo de algoritmos, es el problema de clasificación, donde el sistema de aprendizaje trata de etiquetar una serie de vectores utilizando una entre varias categorías (Arteaga, 2015). Otra subcategoría del aprendizaje supervisado es la regresión, donde la señal resultante es un valor continuo (Mirjalili y Raschka, 2020). En la Figura 2.3 se muestra el proceso de ejecución de un algoritmo de aprendizaje supervisado.



**Figura 2.3:** Diagrama de procesos de ejecución de un algoritmo de aprendizaje supervisado.

## Aprendizaje no Supervisado

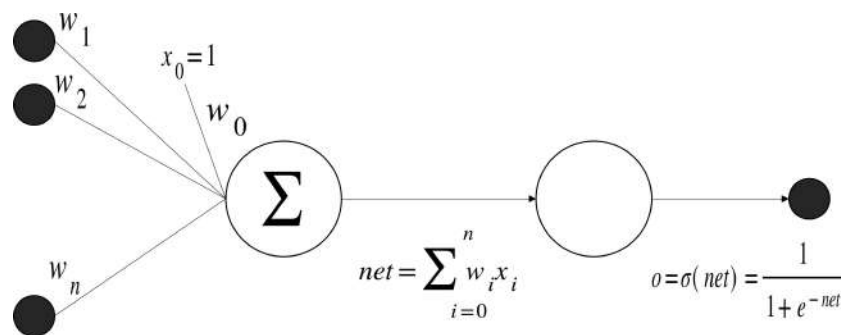
Todo el proceso de modelado se lleva a cabo sobre un conjunto de ejemplos formado tan solo por entradas al sistema. No se tiene información sobre las categorías de esos ejemplos. Constituye un tipo de aprendizaje por observación y descubrimiento, donde el sistema de aprendizaje analiza una serie de entidades y determina que algunas tienen características comunes, por lo que pueden ser agrupadas formando un concepto (Arteaga, 2015).

## 2.2. Aprendizaje profundo

El término Aprendizaje Profundo agrupa una parte de técnicas de aprendizaje automático que se basan en modelos de redes neuronales y cuya aplicabilidad se ha mostrado recientemente en multitud de usos por parte de la industria (Torres, 2020).

### 2.2.1. Redes Neuronales

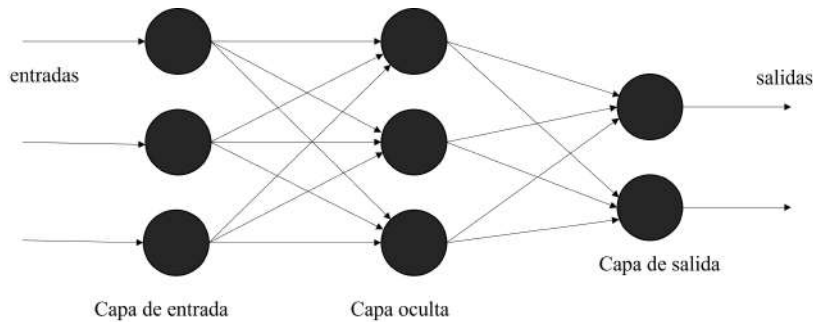
La red neuronal más simple posible es el perceptrón y fue definido por Rosenblatt (1958). En el contexto del aprendizaje supervisado y la clasificación binaria (0 o 1), esta puede utilizarse para predecir si una muestra pertenece a una clase u otra. El perceptrón (Figura 2.4) considera una suma ponderada de sus entradas y aplica una función sigmoide a esta suma. Así se obtiene la salida de la red, la cual durante la fase de aprendizaje se utiliza para calcular el error de la predicción y actualizar los pesos.



**Figura 2.4:** Concepto del perceptrón

En la Figura 2.5 se muestra un ejemplo de red neuronal multicapa. Este tipo de estructura

permite el aprendizaje automático de funciones complejas que mapean de forma no lineal desde el espacio de entrada hasta el espacio de salida (Mirjalili y Raschka, 2020).



**Figura 2.5:** Perceptrón multicapa con una sola capa oculta.

La fase de aprendizaje o entrenamiento de la red neuronal consiste en encontrar los valores de los pesos que minimizan el error de las predicciones. Esto se lleva a cabo utilizando comúnmente el algoritmo del descenso del gradiente (Robbins y Monro, 1951) junto con el algoritmo de retropropagación del error (Rumelhart et al., 1985).

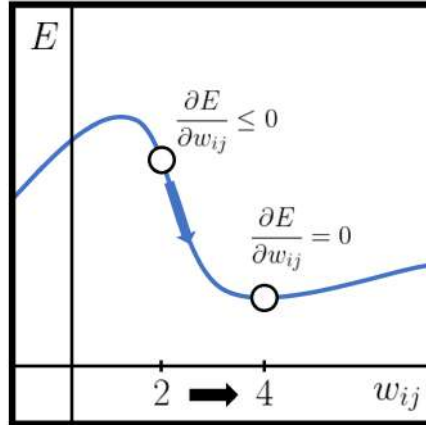
### Descenso del gradiente

El algoritmo del descenso del gradiente (Robbins y Monro, 1951) es un enfoque de optimización iterativo, el cual ajusta gradualmente los parámetros del modelo para minimizar la función de pérdida sobre el conjunto de datos de entrenamiento. La actualización de los parámetros en cada iteración está dado por la Ecuación 2.1. Donde  $w_{ij}$  es un parámetro,  $E$  es la función de pérdida y  $\alpha$  es un hiperparámetro conocido como la tasa de aprendizaje o learning rate. La tasa de aprendizaje determina la magnitud de la actualización del parámetro  $w_{ij}$  en cada iteración. La actualización de los parámetros puede realizarse en lotes de muestras, utilizando el promedio de los gradientes. El tamaño del lote es un hiperparámetro también conocido como batch size.

$$w_{ij} = w_{ij} - \alpha \left( \frac{\partial E}{\partial w_{ij}} \right) \quad (2.1)$$

En la Figura 2.6 se representa visualmente la función de pérdida para un modelo simple con un solo parámetro, y la derivada de esta, en dos puntos distintos. Como se puede observar, la

función alcanza su menor valor en donde la derivada es igual a 0, por tanto, el desplazamiento del valor del parámetro se debe realizar en dirección contraria al gradiente (Domke, 2012).



**Figura 2.6:** Derivada de la función de pérdida  $E$  en dos puntos distintos de la curva.

### Retropropagación del error

El algoritmo de retropropagación del error o la regla delta generalizada (Rumelhart et al., 1985), aplica de forma efectiva el método del gradiente utilizando la regla de la cadena para todos los parámetros de la red. Antes de la retropropagación del error, se debe obtener la activación de la capa de salida, realizando propagación hacia adelante, utilizando las Ecuaciones 2.2, 2.3, 2.4 y 2.5, para el caso de un perceptrón multicapa de solo una capa oculta. Donde,  $\mathbf{a}$  es un vector de características de una muestra  $\mathbf{x}$  más una unidad de sesgo,  $\mathbf{W}$  es una matriz de pesos,  $\mathbf{z}$  es la matriz de entrada a una capa,  $\phi$  es la función de activación, y los superíndices  $in$ ,  $h$  y  $0$  indican las capas de entrada, oculta y de salida, respectivamente.

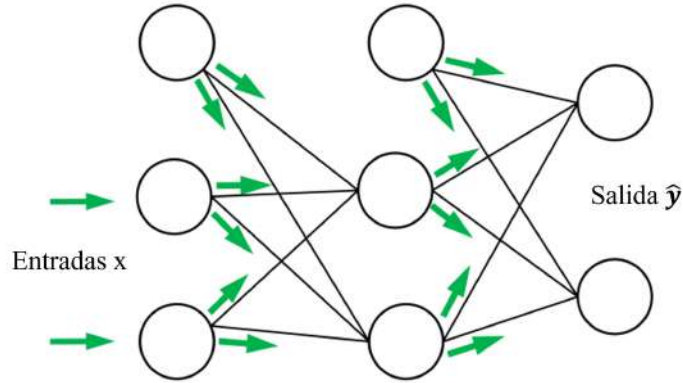
$$\mathbf{z}^{(h)} = \mathbf{a}^{(in)} \mathbf{W}^{(h)} \quad (\text{Entrada de la capa de oculta}) \quad (2.2)$$

$$\mathbf{a}^{(h)} = \phi(\mathbf{z}^{(h)}) \quad (\text{Activación de la capa de oculta}) \quad (2.3)$$

$$\mathbf{z}^{(0)} = \mathbf{a}^{(h)} \mathbf{W}^{(0)} \quad (\text{Entrada de la capa de salida}) \quad (2.4)$$

$$\mathbf{a}^{(0)} = \phi(\mathbf{z}^{(0)}) \quad (\text{Activación de la capa de salida}) \quad (2.5)$$

En la Figura 2.7 se muestra la propagación hacia adelante en un perceptron multicapa de una capa oculta, hasta obtener la estimación del vector  $\mathbf{y}$ .



**Figura 2.7:** Propagación hacia adelante.

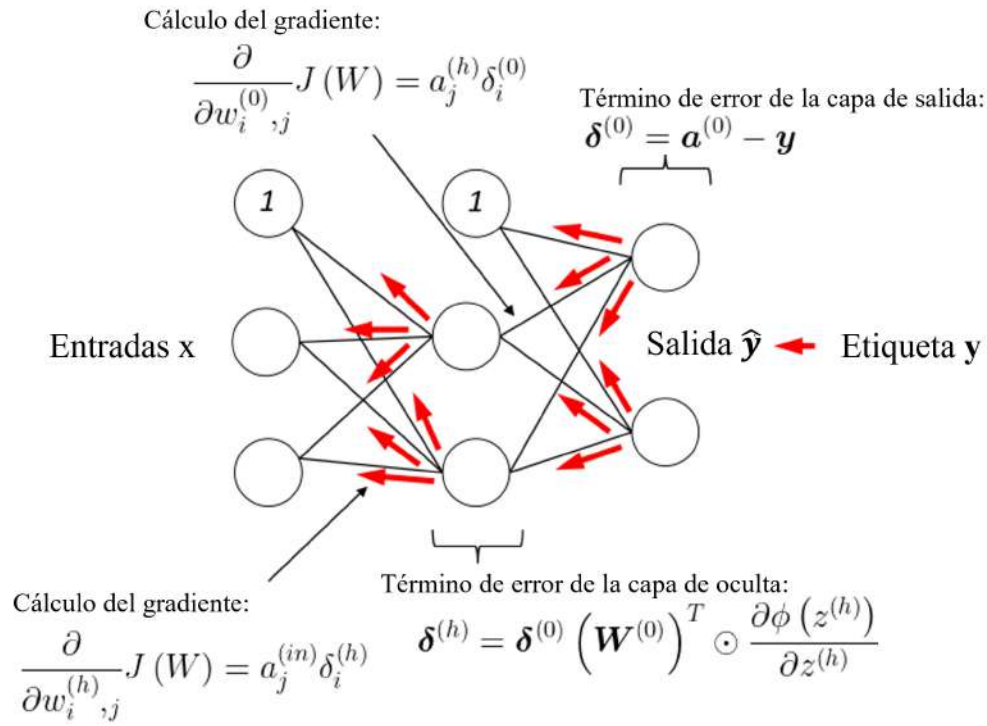
La retropropagación comienza con el cálculo del vector de error de la capa da salida definido en la Ecuación 2.6. Luego, se calcula el término de error de la capa oculta, mediante la Ecuación 2.7. Luego, las derivadas de la función de pérdida  $J(W)$  para los parámetros se obtienen mediante las Ecuaciones 2.8 y 2.9. Finalmente, la Figura 2.8 resume el algoritmo de retropropagación.

$$\boldsymbol{\delta}^{(0)} = \mathbf{a}^{(0)} - \mathbf{y} \quad (2.6)$$

$$\boldsymbol{\delta}^{(h)} = \boldsymbol{\delta}^{(0)} (\mathbf{W}^{(0)})^T \odot \frac{\partial \phi(z^{(h)})}{\partial z^{(h)}} \quad (2.7)$$

$$\frac{\partial}{\partial w_{i,j}^{(0)}} J(W) = a_j^{(h)} \delta_i^{(0)} \quad (2.8)$$

$$\frac{\partial}{\partial w_{i,j}^{(h)}} J(W) = a_j^{(in)} \delta_i^{(h)} \quad (2.9)$$



**Figura 2.8:** Retropropagación del error.

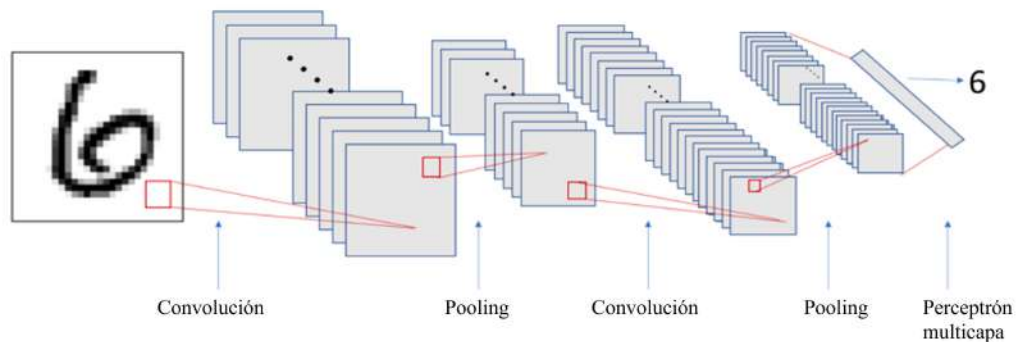
### 2.2.2. Redes Neuronales convolucionales

Las Redes Neuronales Convolucionales (Le Cun et al., 1989) son un tipo de red neuronal que se diferencia de las mencionadas anteriormente, en la suposición explícita de que las entradas son imágenes. Esto permite codificar ciertas propiedades en las arquitecturas para reconocer elementos concretos en las imágenes (Mirjalili y Raschka, 2020).

La idea de las redes neuronales convolucionales comenzó con el trabajo de Fukushima (1980), donde se desarrolló el neocognitron, una red neuronal de tipo backpropagation que imita el proceso del cortex visual. Luego, en Le Cun et al. (1989) se entrenó la red denominada LeNet, que emplea una estructura de red neuronal convolucional y consiguió clasificar imágenes de dígitos escritos a mano. Pero, fue en Krizhevsky et al. y la creación de la red AlexNet, cuando se observó el potencial de las redes neuronales convolucionales para la clasificación de imágenes.

Normalmente, las redes convolucionales están formadas de muchas capas convolucionales

y capas de submuestreo (también llamadas agrupación o pooling), que están seguidas por un perceptrón multicapa. Las primeras capas de una red convolucional aprenden características generales altamente genéricas, después, gradualmente en capas sucesivas se van aprendiendo características más concretas del dominio del problema que se trata, es decir, conceptos más abstractos. En la Figura 2.9 se muestra un ejemplo de red convolucional, para el problema de reconocimiento de dígitos escritos a mano del conjunto de datos MNIST (Deng, 2012).



**Figura 2.9:** Ejemplo de una red convolucional alternando capas de convolución y pooling.

## Convolución

La convolución es la operación de una capa convolucional. Esta usa un kernel o filtro con parámetros entrenables. Este tiene el objetivo de extraer características relevantes de un espacio conservando la disposición espacial de estas. En la Figura 2.10 se muestra el kernel en color amarillo con sus parámetros en rojo. Este opera desde la esquina superior izquierda del espacio de características, obteniendo la suma de valores ponderados por los parámetros del filtro, para almacenarlo en el siguiente espacio de características. El kernel se desplaza de izquierda a derecha y de arriba abajo hasta recorrer todo el espacio.

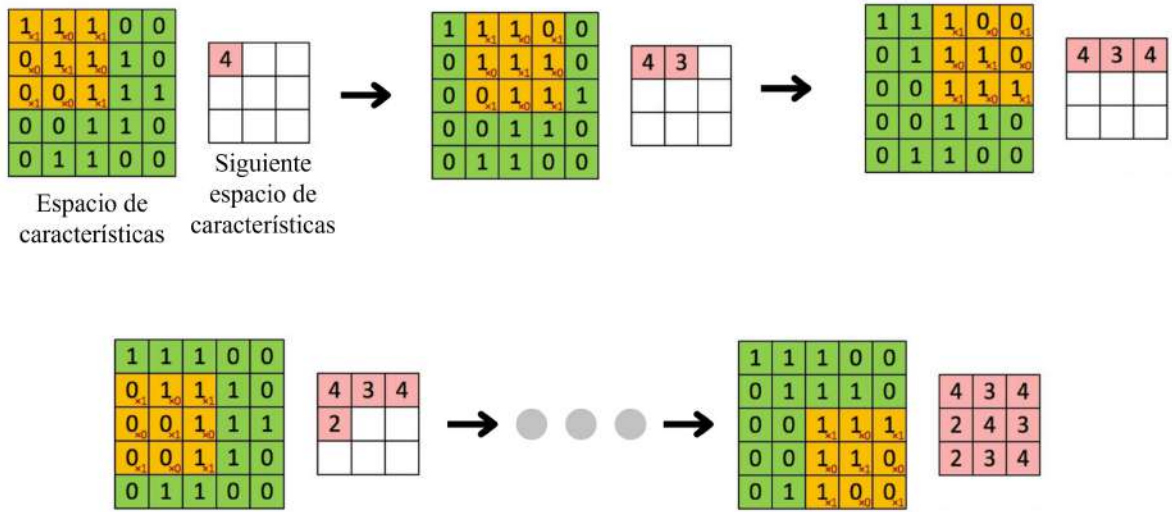


Figura 2.10: Proceso de convolución.

### Pooling

La agrupación o “Pooling” se realiza en redes neuronales para reducir la varianza y la complejidad del cálculo. Los tres tipos de operaciones de agrupación son: agrupación máxima, que selecciona el valor máximo de pixeles del lote; agrupación mínima, que selecciona el valor mínimo de pixel del lote y la agrupación promedio, que selecciona el valor promedio de todos los pixeles en el lote. En la Figura 2.11 se muestra la agrupación máxima.

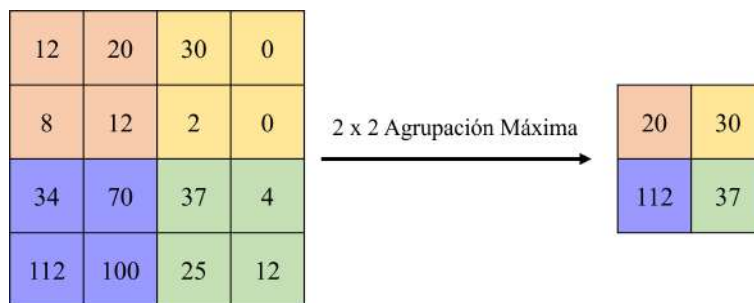
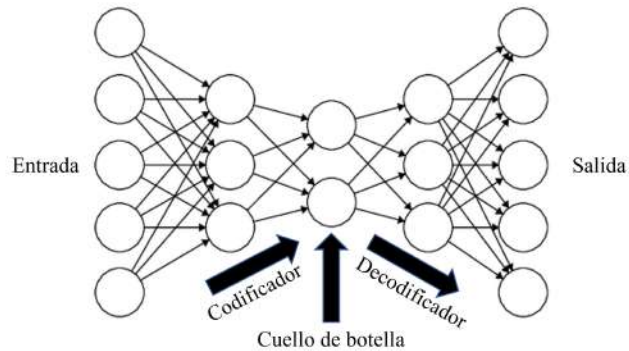


Figura 2.11: Ejemplo de agrupación máxima.

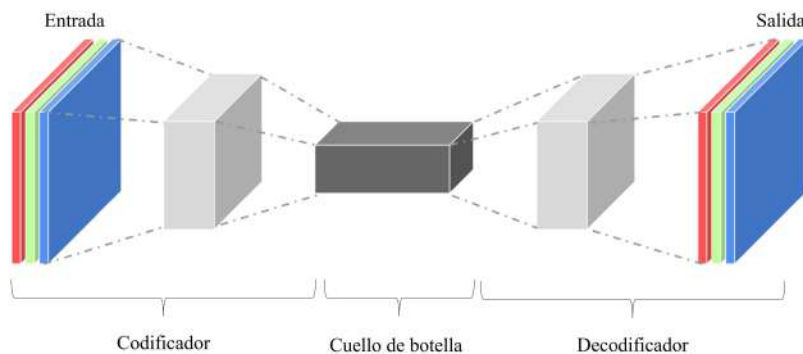


### 2.2.3. Autoencoders

Las redes neuronales autoencoder poseen una arquitectura que se compone de tres elementos, un codificador, un cuello de botella y un decodificador (Figura 2.12). El codificador comprime los datos de entradas a un espacio de características de menor dimensión llamado cuello de botella o código. El decodificador es el componente de la red encargado de interpretar el código aumentando su dimensionalidad. Para el caso de las redes convolucionales autoencoders (Figura 2.13), el decodificador contiene capas de sobre muestreo (como capas deconvolucionales y de unpooling), y permite desarrollar tareas como la reconstrucción de imágenes o segmentación semántica.



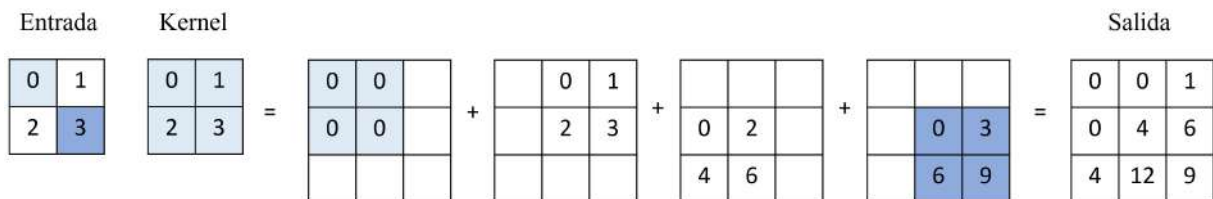
**Figura 2.12:** Estructura codificador-decodificador.



**Figura 2.13:** Codificador-decodificador convolucional.

## Deconvolución

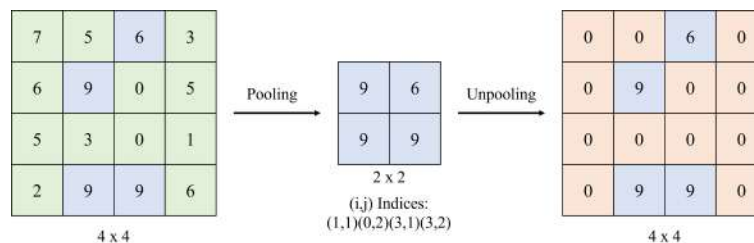
La deconvolución es una técnica de sobremuestreo que utiliza un kernel o filtro de parámetros que pueden ser aprendidos. En la Figura 2.14 muestra un ejemplo de mapeo de un input de dimensión  $2 \times 2$  hasta un espacio de características de dimensión  $3 \times 3$ . El primer paso considera multiplicar el valor de la esquina superior izquierda del input por cada elemento del filtro, para luego almacenar cada valor resultante en la posición correspondiente del espacio de características de salida. Este proceso continúa desplazándose de izquierda a derecha y de arriba abajo tanto para los valores del input, como para las posiciones de almacenamiento, sumando los valores superpuestos.



**Figura 2.14:** La operación de convolución transpuesta.

## Max Unpooling

Es una técnica de sobremuestreo, que recuerda las posiciones de los valores máximos antes de la operación max pooling. En la Figura 2.16 se muestra un ejemplo de inputs y outputs de las operaciones max pooling y max unpooling.



**Figura 2.15:** Ejemplo de max pooling y max unpooling.

### Bilinear interpolation

Es una técnica de sobremuestreo, que aplica interpolación lineal en dos direcciones, usando los cuatro vecinos más cercanos tomando el promedio de estos. En la Figura 2.16 se muestra un ejemplo de esta operación.

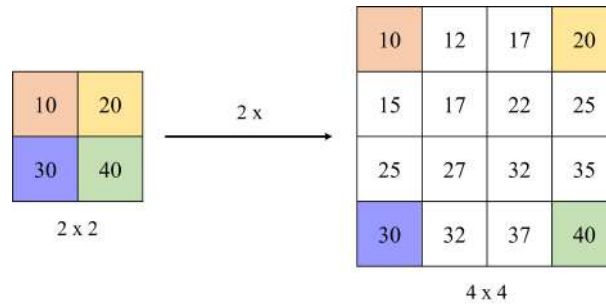


Figura 2.16: Ejemplo de bilinear interpolation.

## 2.3. Segmentación semántica

La segmentación semántica es la tarea de determinar la etiqueta o categoría para cada pixel de una imagen. En la Figura 2.17 se visualiza el resultado de la segmentación semántica y otras tareas de la visión por computador como lo es la clasificación y localización, detección de objetos y la segmentación por instancia.

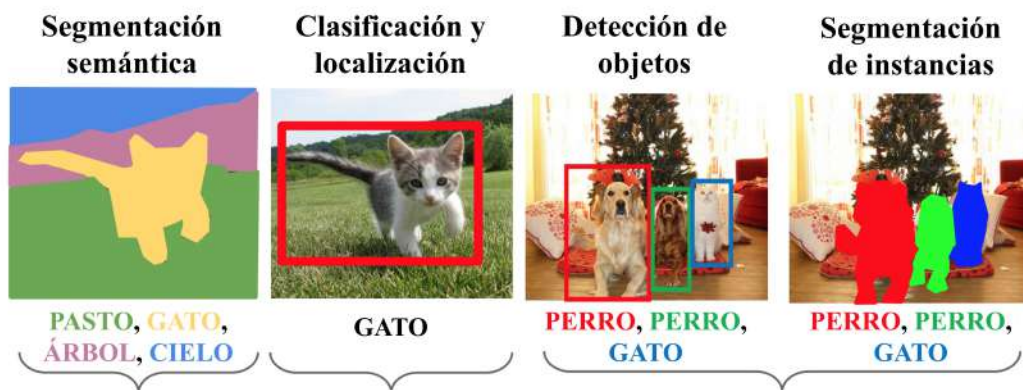


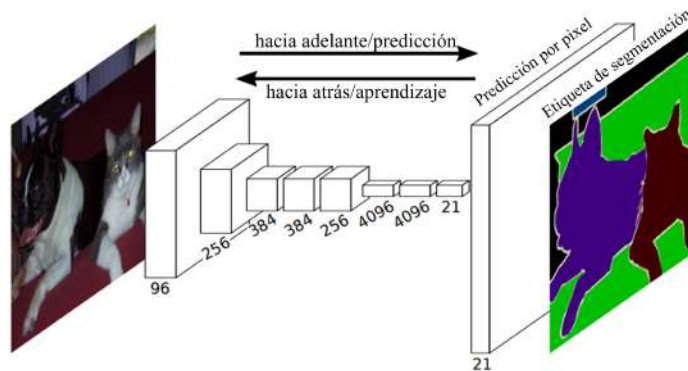
Figura 2.17: Tareas de la visión por computador.

La segmentación semántica como se menciona en la Sección 2.2.3, se puede resolver mediante modelos de aprendizaje profundo que posean una estructura codificador-decodificador. Ejemplo de estos son: Fully Convolutional Network y Multi-path Residual Network.

### 2.3.1. Fully Convolutional Network

La arquitectura Fully Convolutional Network (FCN) fue propuesta por Long et al. (2015) y es utilizada principalmente para realizar segmentación semántica. Esta emplea únicamente capas convolucionales, pooling y de sobremuestreo. Lo que implica que puede procesar imágenes de distintos tamaños, ya que carece de capas densas.

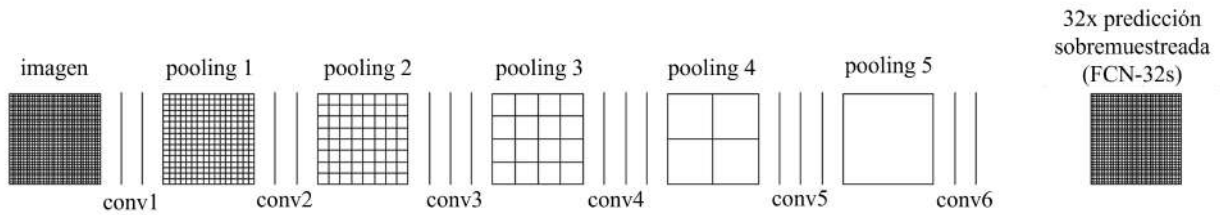
FCN consta de un codificador y un decodificador. El primero se utiliza para extraer características y conservar la relación contextual de estas. El segundo realiza una reconstrucción de la imagen pero segmentada según las categorías de interés. La estructura codificador-decodificador de la FCN se puede apreciar en la Figura 2.18, donde el decodificador contiene una capa de sobremuestreo o deconvolución, la que realiza un muestreo superior aumentando el tamaño de salida.



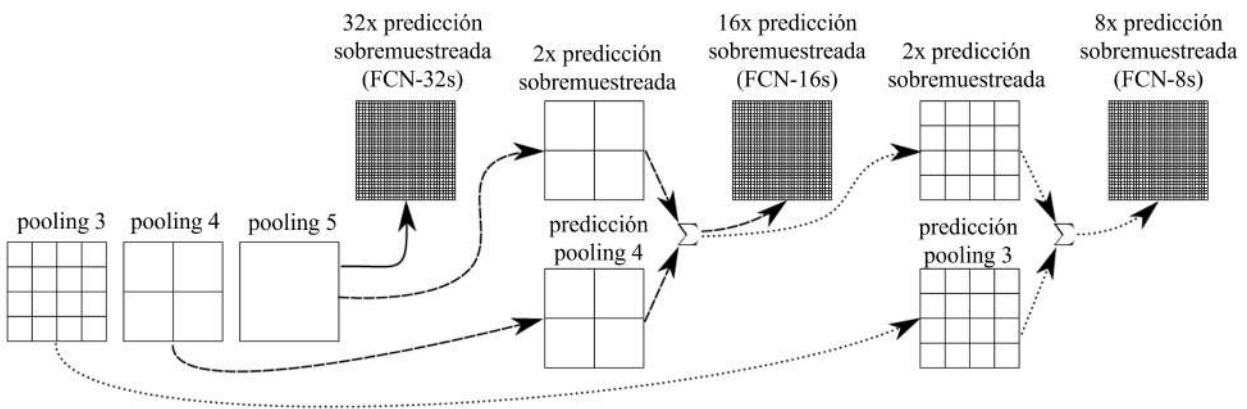
**Figura 2.18:** Estructura de la FCN-32s.

Los autores de la FCN presentan tres versiones de esta, la primera y más sencilla es la FCN-32s, su arquitectura se detalla en la Figura 2.19. La FCN-16s y la FCN-8s son variaciones más complejas de la FCN-32s como se muestra en la Figura 2.20. Estas contienen una mayor cantidad de capas deconvolucionales para el sobremuestreo progresivo de las características en

el decodificador. Además, al mismo tiempo implementa conexiones de salto para disminuir la pérdida de información.

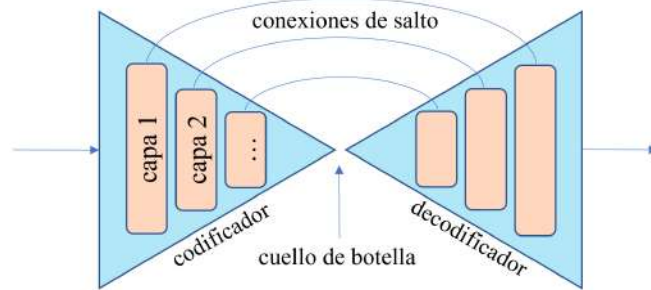


**Figura 2.19:** FCN-32s detallada.

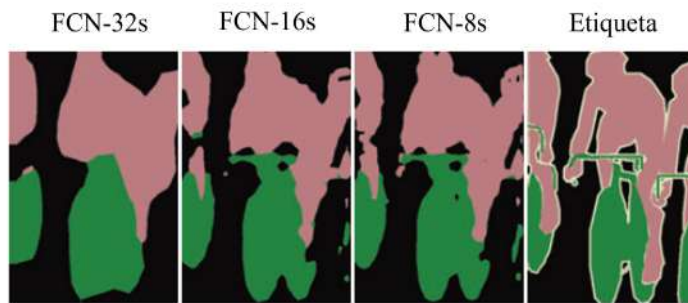


**Figura 2.20:** Fusión para FCN-16s y FCN-8s.

Debido a que en un codificador-decodificador operan capas convolucionales, pooling y de sobremuestreo, existe una pérdida de información atribuida a los cambios de dimensionalidad de los espacios de características. Las conexiones de salto permiten enviar mapas de características del codificador hacia al decodificador (Figura 2.21) disminuyendo la pérdida de información. Por tanto, el desempeño del modelo mejora, debido a el mejoramiento de la descompresión en el decodificador (Dong et al., 2018). En la Figura 2.22 se muestran los resultados para distintas variaciones de la FCN, donde se puede apreciar que la FCN-8s tiene el mejor resultado.



**Figura 2.21:** Estructura codificador-decodificador y conexiones de salto a través del cuello de botella.



**Figura 2.22:** Comparación de distintas variaciones de la FCN.

### 2.3.2. Multi-path Residual Network

La arquitectura de Multi-path Residual Network (MP-ResNet) fue propuesta por Ding et al. (2021) y se inspira en la Fully Convolutional Network (FCN). MP-ResNet fue diseñada para mejorar la segmentación semántica de imágenes de radar polarimétrico de apertura sintética, que es caracterizada por tener una gran cantidad de ruido moteado o de manchas. Esta red propone una forma de aprender características discriminatorias de un contexto de imagen más grande, de modo que se puedan mitigar los efectos de los ruidos moteados. Como se puede apreciar en la Figura 2.23 esta adopta la ResNet (He et al., 2016) como extractor de características y utiliza ramas paralelas de múltiples escalas para mejorar el aprendizaje de contexto semántico.

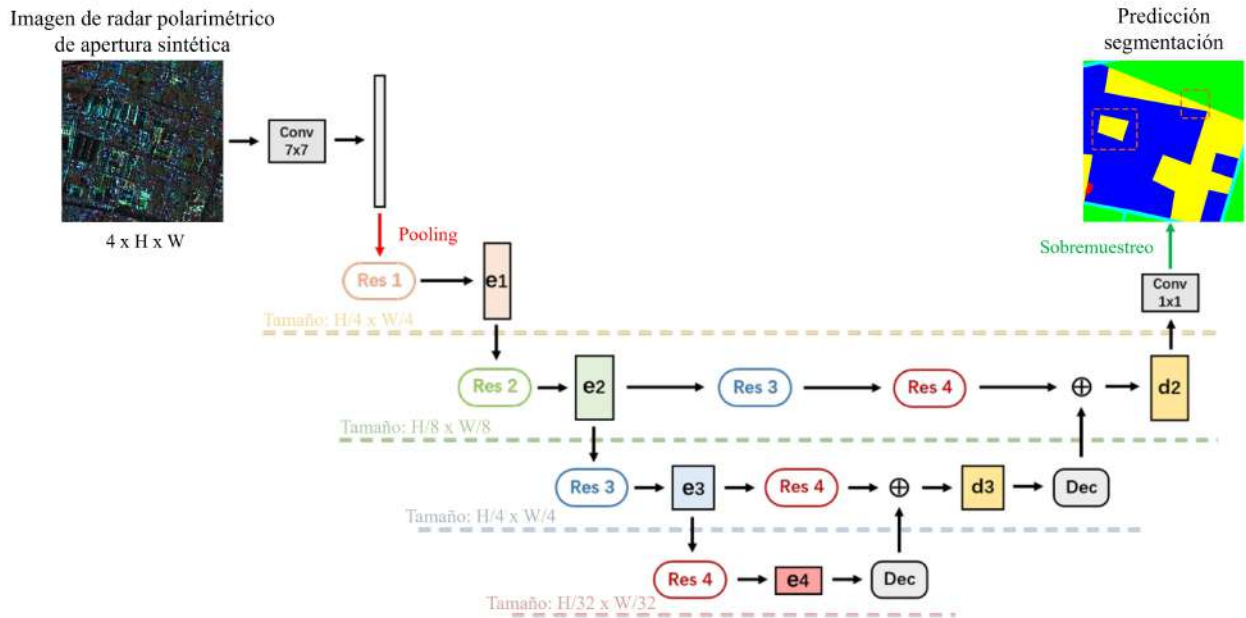


Figura 2.23: Arquitectura MP-ResNet.

MP-ResNet contiene un módulo de deconvolución de características caracterizado por tener un diseño de “cuello de botella” por canal para reducir el cálculo y refinar la información semántica crucial. De esta manera, las características de múltiples ramas se fusionan de manera gruesa a fina en el decodificador. El módulo de deconvolución se muestra en la Figura 2.24.

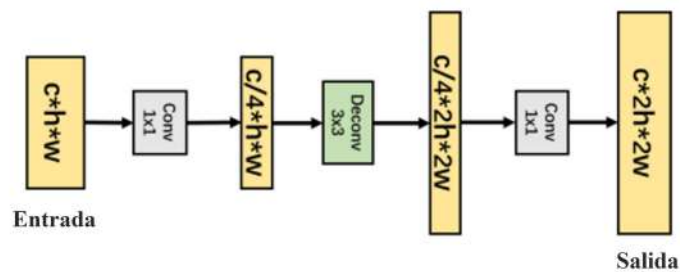


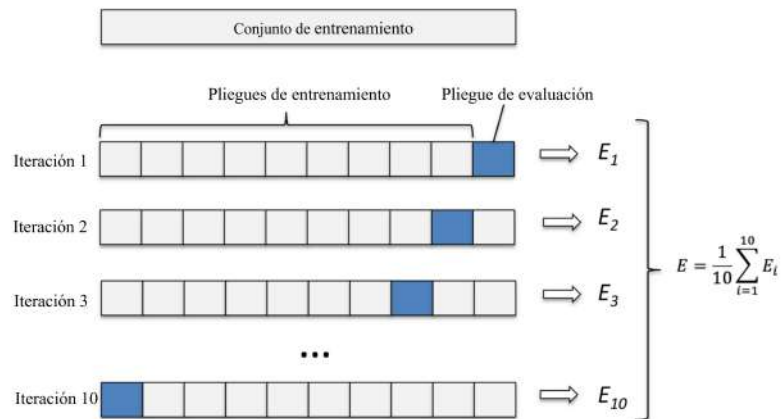
Figura 2.24: Bloque de deconvolución de características del MP-ResNet.

## 2.4. Técnicas en aprendizaje profundo

En esta sección se exponen las técnicas de aprendizaje profundo utilizadas en el presente trabajo, para la evaluación de modelos, regularización y reutilización de modelos.

### 2.4.1. K-Fold Cross Validation

$K$ -Fold Cross Validation es una técnica de evaluación de modelos. En la que se divide el conjunto de datos de entrenamiento en  $k$  pliegues o folds sin reemplazo, donde  $k - 1$  pliegues se utilizan para el entrenamiento del modelo y un pliegue para la evaluación del rendimiento. Este proceso se itera  $k$  veces para obtener  $k$  modelos y estimaciones de rendimiento (Mirjalili y Raschka, 2020). Esta técnica se ve representada de buena manera con un 10-Fold Cross Validation de ejemplo en la Figura 2.25.



**Figura 2.25:** 10-Fold Cross Validation.

### 2.4.2. Stop-Early

La técnica Stop-Early o Parada temprana consiste en detener el entrenamiento tan pronto como el error de validación alcance un mínimo. La Figura 2.26 muestra un modelo que a medida que pasan las épocas, el algoritmo aprende y su error de predicción disminuye en este ejemplo, basado en la raíz del error cuadrático medio (RMSE) en el conjunto de entrenamiento. Además, también disminuye su error de predicción en el conjunto de validación. Sin embargo, después



de un tiempo, el error de validación deja de disminuir y en realidad comienza a aumentar. Esto indica que el modelo ha comenzado a sobre ajustarse a los datos de entrenamiento. Con la parada anticipada, simplemente deja de entrenar tan pronto como el error de validación alcanza el mínimo (Géron, 2019).

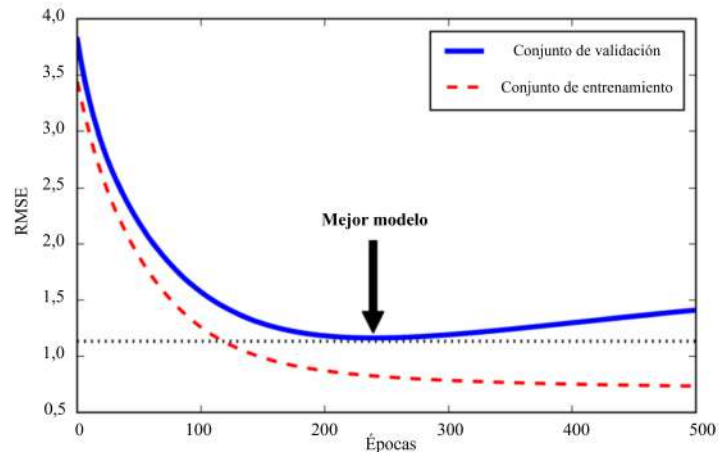
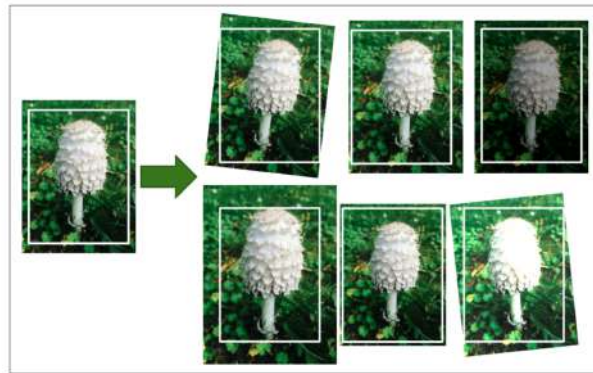


Figura 2.26: Ejemplo de Stop-Early.

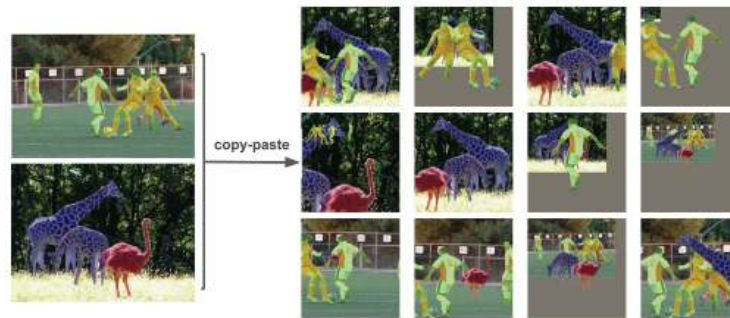
### 2.4.3. Data Augmentation

Data augmentation o aumento de datos es una técnica de regularización, consiste en generar nuevas instancias de entrenamiento a partir de las existentes, aumentando artificialmente el tamaño del conjunto de entrenamiento. Como se muestra en la Figura 2.27 se puede desplazar, rotar y cambiar ligeramente el tamaño de cada imagen, generar imágenes con varios contrastes y suponiendo que los objetivos sean simétricos, también puede voltear las imágenes horizontalmente entre otras transformaciones (Géron, 2019).



**Figura 2.27:** Aumentaciones de datos estandar. Fuente: Géron (2019).

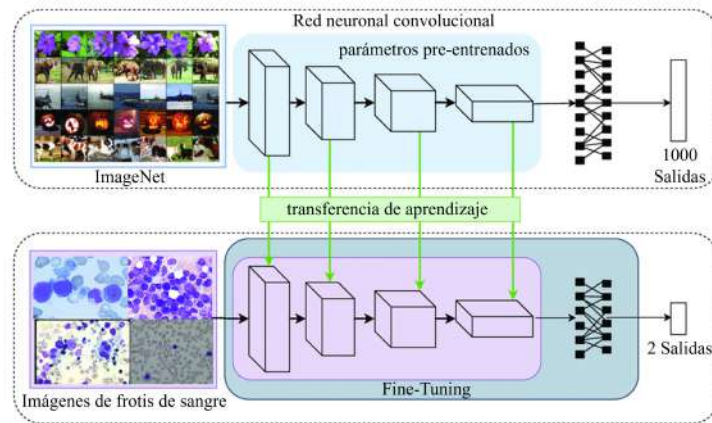
Una aumentación menos común es la aumentación Copy-Paste utilizada en (Ghiasi et al., 2021) para la segmentación por instancia. Consiste en recortar un objeto de una imagen y pegarla en otra como muestra en la Figura 2.28. Esta aumentación podría utilizarse también en segmentación semántica.



**Figura 2.28:** Aumentación Copy-Paste. Fuente: Ghiasi et al. (2021).

#### 2.4.4. Fine-Tuning

Fine-Tuning es una técnica de reutilización de modelos y consiste en transferir los parámetros de un modelo pre-entrenado a otro modelo. Pero, como punto de partida para el entrenamiento del modelo con un conjunto de datos generalmente más pequeño y/o en otro dominio para una tarea determinada (Torres, 2020). El proceso de Fine-Tuning se aprecia en la Figura 2.29.



**Figura 2.29:** Proceso Fine-Tuning. Fuente: Vogado et al. (2021).

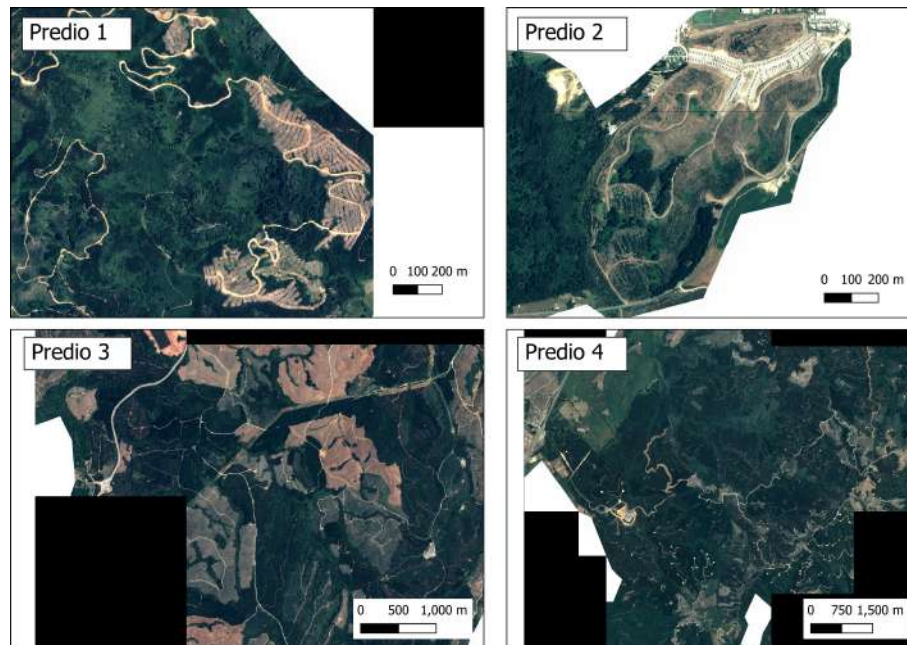
# Capítulo 3

## Materiales y métodos

En este capítulo se presentan imágenes satelitales de predios forestales y el preprocesamiento para la obtención de conjuntos de imágenes, también se explican los métodos y metodología para la comparación de los modelos Fully Convolutional Network y Multi-path Residual Network.

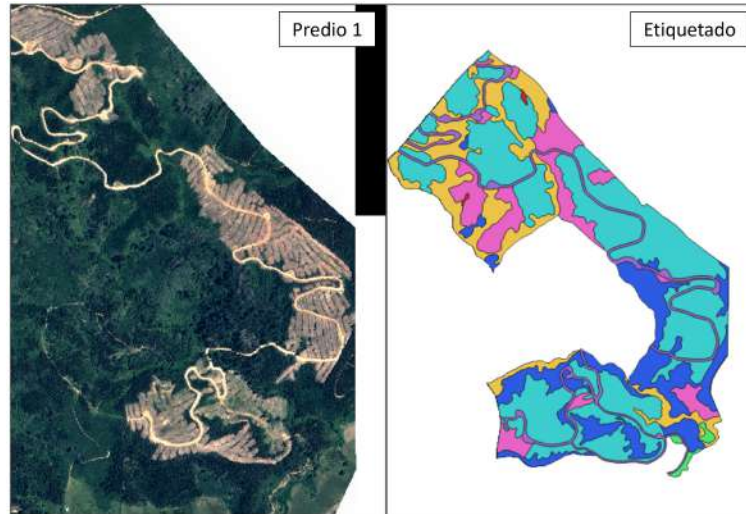
### 3.1. Datos

Para el desarrollo de esta memoria de título se han proporcionado cuatro imágenes satelitales RGB correspondiente a cuatro predios forestales distintos, que están parcialmente etiquetados por personal experto de Arauco S.A. En la Figura 3.1 se visualizan estos predios a los que llamaremos predio 1, 2, 3 y 4.



**Figura 3.1:** Predios forestales.

Las imágenes satelitales de los predios fueron proporcionadas en formato `tif` y etiquetadas como un archivo `shapefile` asociado a cada predio. El etiquetado consiste en polígonos que demarcan áreas dentro del predio y contienen dos atributos importantes: uso de suelo y edad de plantación, esta última exclusivamente para las áreas que contienen cultivos. A modo de ejemplo, en la Figura 3.2 se muestra una visualización de las áreas etiquetadas según el tipo de uso de suelo para el predio 1.



**Figura 3.2:** Imagen satelital del predio 1 y la visualización de las áreas etiquetadas según el tipo de uso de suelo.

### 3.2. Preprocesamiento de datos

Como se menciona en la Sección 3.1, los polígonos del archivo `shapefile` contienen dos atributos: uso de suelo y la edad de plantación. Estos se utilizan para crear la taxonomía de la Tabla 3.1, según los intereses de la industria forestal.

**Tabla 3.1:** Taxonomía utilizada.

Categoría	Valor	Descripción
NODATA	0	Contenido sin valor
EU46	1	Eucaliptus de 4 a 6 años
EU68	2	Eucaliptus de 6 a 8 años
EU810	3	Eucaliptus de 8 a 10 años
EU10inf	4	Eucaliptus de 10 o más años
PIRA47	5	Pino Radiata de 4 a 7 años
PIRA710	6	Pino Radiata de 7 a 10 años
PIRA10inf	7	Pino Radiata de 10 o más años
EUPI04	8	Eucaliptus o Pino Radiata menor a 4 años
BNSM	9	Bosque Nativo
FCAM	10	Camino
OTROS	11	Todas las demás clases

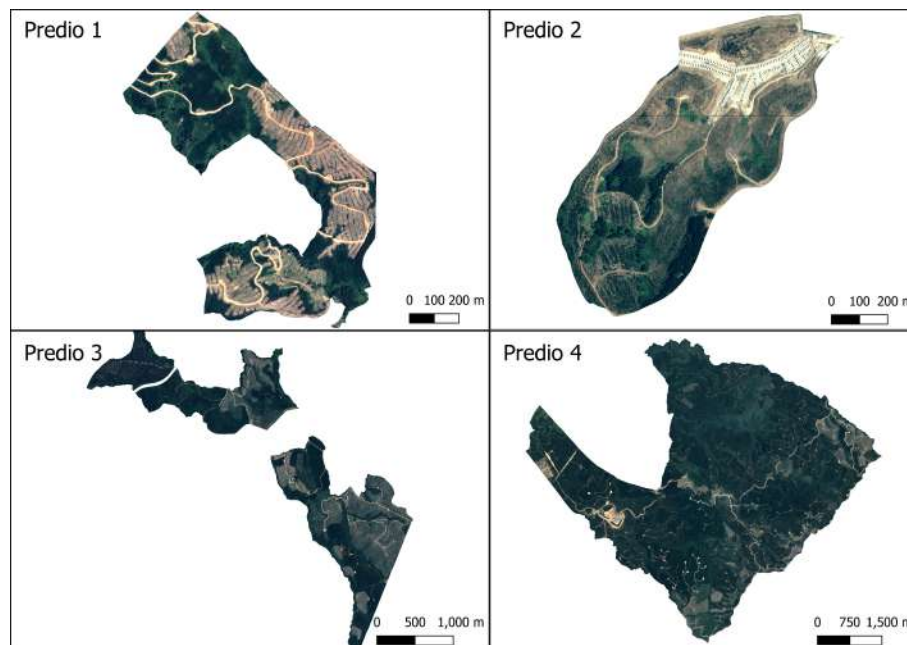
En la Tabla 3.2 se muestra la distribución de categorías resultantes en porcentaje de píxeles y cantidad de píxeles para cada predio. Se puede apreciar un desbalance importante entre las

distintas categorías, incluso en color rojo se puede apreciar que en la categoría PIRA710 del predio 4 solo se tienen 1466 píxeles.

**Tabla 3.2:** Distribución de categorías.

PREDIO		1		2		3		4	
DISTRIBUCIÓN		%	Píxeles	%	Píxeles	%	Píxeles	%	Píxeles
Eti.	EU46	0	0	0	0	32,47	4599029	14,12	15148613
	EU68	11,96	263724	0	0	0	0	27,72	29738524
	EU810	0	0	0	0	0	0	5,74	6162287
	EU10inf	0	0	0	0	10,11	1432405	11,73	12580840
	PIRA47	0	0	22,80	501839	0	0	0,19	204410
	PIRA710	0	0	15,96	351311	2,57	364067	0,00	1466
	PIRA10inf	16,50	363788	7,18	157953	32,63	4622493	16,30	17488597
	EUPI04	36,48	804504	0	0	2,13	302071	1,19	1279209
	BNSM	15,53	342493	5,60	123248	13,86	1962758	14,53	15590885
	FCAM	7,11	156753	3,20	70343	3,48	493593	3,94	4228861
OTROS	12,42	273811	45,27	996345	2,74	388711	4,54	4870870	

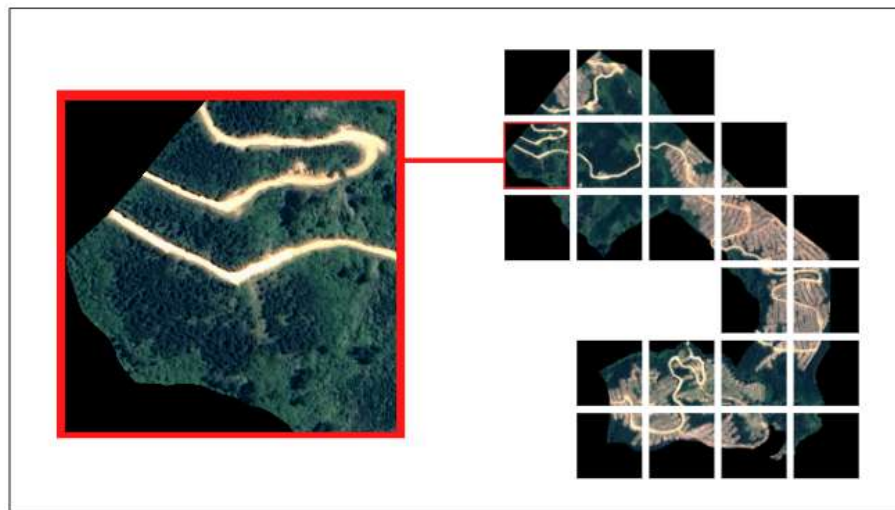
Los predios son enmascarados con los polígonos del archivo `shapefile`, mediante un procedimiento proporcionado por la Unidad de Data Science. En la Figura 3.3 se observan los extractos resultantes del enmascaramiento.



**Figura 3.3:** Extractos etiquetados de los predios forestales.

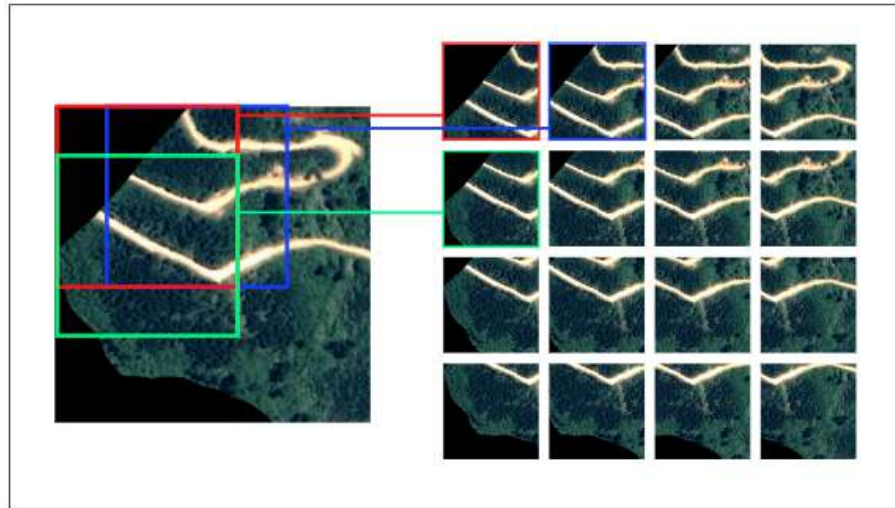
Para los extractos de los predios 1, 2 y 3, se realiza el siguiente proceso: Primero se divide en ventanas de  $448 \times 448$  píxeles, como muestra en la Figura 3.4. Las ventanas se distribuyen

en conjunto de entrenamiento y testeo con aproximadamente 80 % y 20 % de la cantidad total de ventanas, respectivamente. Después, cada ventana se subdivide con solapamiento generando subimágenes de  $256 \times 256$  píxeles. Este proceso se realiza desde izquierda a derecha y desde arriba hacia abajo con un solapamiento de 64 píxeles como se aprecia en la Figura 3.5. Generando un máximo de 16 subimágenes por ventana, ya que las subimágenes se descartan si es que no contienen píxeles correspondientes al predio. Para el predio 4, no se consideró necesario realizar el primer paso de división de ventanas debido al gran tamaño del extracto etiquetado, se optó por generar directamente las subimágenes de  $256 \times 256$  píxeles sin solapamiento desde el extracto etiquetado. Adicionalmente, se descartan las subimágenes que contienen la categoría PIRA710, dado su mínima aparición en el predio según lo visto en la Tabla 3.2. Luego, se distribuyen en conjunto de entrenamiento y testeo con aproximadamente 80 % y 20 % de la cantidad total de subimágenes generadas. Para todos los conjuntos de entrenamiento se considera un 25 % de él como conjunto de validación.



**Figura 3.4:** División del predio 1 en ventanas de  $448 \times 448$  píxeles.





**Figura 3.5:** Subdivisión con solape de las ventanas.

### 3.3. Métodos

Se implementan y comparan dos modelos de aprendizaje profundo supervisado para la segmentación semántica utilizando principalmente el lenguaje de programación Python 3.7 y el framework de Pytorch en una instancia de Google Colab. Los modelos son la Fully Convolutional Network 8s y la Multi-path Residual Network (MPResNet), ambos modelos utilizando como extractor de características la red ResNet34 pre-entrenada.

### 3.4. Metodología

El conjunto de entrenamiento se emplea para ajustar los parámetros del modelo. El conjunto de validación se utiliza para evaluar el desempeño del modelo en cada época del entrenamiento y detener el entrenamiento cuando en este conjunto el desempeño no ha mejorado en una cantidad de épocas. Se utiliza el conjunto de test para evaluar el desempeño del modelo sobre datos anteriormente no observados.

### 3.5. Hiperparámetros

Los hiperparámetros son parámetros externos al modelo en sí mismo. Estos tienen un gran impacto en la precisión de la red neuronal y puede haber diferentes valores óptimos para diferentes hiperparámetros (Torres, 2020).

En este trabajo se evalúan distintas combinaciones de hiperparámetros utilizando 4-Fold CV en el conjunto de entrenamiento del predio 1, con el objetivo de encontrar buenos hiperparámetros para los modelos. Los valores de los hiperparámetros evaluados y la mejor combinación seleccionada se muestran en la Tabla 3.3. Adicionalmente, en la Tabla A.1 y Figura A.1 se pueden ver los resultados de las combinaciones de hiperparámetros.

**Tabla 3.3:** Búsqueda de hiperparámetros.

Hiperparámetro	Valores	Valor seleccionado
Learning rate	{0,0001; 0,001; 0,005; 0,01}	0,005
Batch size	{4; 8; 16; 32; 48}	FCN: 16; MP-ResNet: 4

### 3.6. Entrenamiento

Con el objetivo de lograr un buen desempeño para los modelos en el conjunto de test, se entrena con los mejores hiperparámetros encontrados en la búsqueda de hiperparámetros. Además, se utiliza Data Augmentation y Fine-Tuning para mejorar la generalización de los modelos.

Las aumentaciones de datos utilizados son Random-Flipping (se activa una de las siguientes transformaciones, Flip vertical, Flip Horizontal, Flip Vertical y Horizontal, cada una con probabilidad de 25 %), Random-Rotation (se activa una rotación con los siguientes grados, 90, 180 y 270, cada uno con probabilidad de 25 %) y Copy-Paste con una probabilidad de activación de 25 %.

La aumentación Copy-Paste se aplica de forma estratégica para contrarrestar los efectos del desbalance de clases, de la siguiente forma: Primero se selecciona una de las categorías presente en el predio de forma aleatoria ponderada por los coeficientes de desbalance definidos por la Ecuación 3.1, donde  $t_i$  es la cantidad de pixeles de la categoría  $i$ . Posteriormente, se selecciona una imagen aleatoria que contenga la categoría y se realiza una copia de los pixeles

de esa categoría. Luego, la copia de esos píxeles sufren una transformación Random-Flipping y/o Random-Rotation. Finalmente, los píxeles se pegan sobre la imagen, para la cual se activó la aumentación Copy-Paste.

$$w_i = \frac{\max\{t\}}{t_i} \quad (3.1)$$

### 3.7. Validación

Para validar la efectividad de los modelos en la segmentación semántica, se utiliza la métrica Frequency-weighted IoU (FwIoU), como en Ding et al. (2021). Esta métrica está definida por la Ecuación 3.2.

$$FwIoU = \frac{1}{\sum_{j=1}^k t_j} \sum_{j=1}^k t_j \frac{n_{jj}}{n_{ij} + n_{ji} + n_{jj}}, \quad i \neq j \quad (3.2)$$

Donde  $n_{jj}$  es el total de píxeles con etiqueta de la clase  $j$  que fueron clasificados como clase  $j$ .  $n_{ij}$  es el total de píxeles con etiqueta de la clase  $i$  que fueron clasificados como clase  $j$ .  $n_{ji}$  es el total de píxeles con etiqueta de la clase  $j$  que fueron clasificados como clase  $i$  y  $t_j$  es el total de píxeles etiquetados como clase  $j$ . Esta métrica pondera las clases por importancia (su valor de  $t_j$ ).

La validación del modelo se realiza dentro del entrenamiento. Se utiliza el conjunto de validación para evaluar el desempeño del modelo en la métrica FwIoU, después de cada época. Además, para detener el entrenamiento si el desempeño no ha mejorado en una cierta cantidad de épocas, fijada en 15, es decir, usando Stop-Early.

### 3.8. Evaluación

Para comparar el desempeño de los modelos se utilizan tres conocidas métricas, Accuracy, FwIoU y F1-score, se evalúan estos sobre el conjunto de test con 4-Fold CV, mediante la métrica Accuracy que es definida como el número de píxeles clasificado correctamente sobre el número

total de píxeles, FwIoU como se explica anteriormente y F1-score se define según la Ecuación 3.3.

$$F1 - score = 2 \times \frac{Precisión \times Recall}{Precisión + Recall} \quad (3.3)$$

Donde, Precisión se calcula según la Ecuación 3.4 y Recall según la Ecuación 3.5.

$$Precisión = \frac{n_{jj}}{n_{ij} + n_{jj}} \quad i \neq j \quad (3.4)$$

$$Recall = \frac{n_{jj}}{n_{ji} + n_{jj}} \quad i \neq j \quad (3.5)$$

De las métricas, Accuracy y F1-score, se excluye la categoría NODATA para una mejor interpretación del rendimiento de los modelos sobre las categorías de interés forestal.

## Capítulo 4

# Resultados experimentales y discusión

En este capítulo se presentan y comparan los desempeños de los modelos, de forma general, por predio y por categoría. A partir de esto, se realizan observaciones que explican el comportamiento de los modelos. Finalmente se exponen visualmente las predicciones de los modelos.

### 4.1. Segmentación semántica

Los resultados obtenidos por los modelos FCN y MP-ResNet en los distintos predios se muestran en la Tabla 4.1, se puede apreciar que ninguno de los dos modelos tienen una ventaja clara por sobre el otro en términos generales. FCN muestra ventaja en el predio 2 y 4 en las tres métricas utilizadas, pero no así en los predios 1 y 3, donde MP-ResNet tiene los mejores valores. Para ambos modelos en los distintos predios, FwIoU tiene un rendimiento promedio mayor a 78,84 % y menor e igual a 87,21 %. Mientras que Accuracy tiene un rendimiento promedio mayor a 65,81 % y menor e igual a 84,25 %. Finalmente, mF1 (F1-score promedio de las categorías) tiene un rendimiento promedio mayor a 51,95 % y menor e igual a 79,29 %. Las curvas de aprendizaje de los modelos presentes se encuentran en las figuras del Anexo A.2.

**Tabla 4.1:** Cross validation de las métricas sobre el conjunto de test.

Modelo	FCN			MP-ResNet		
Predio	Accuracy (%)	mF1 (%)	FwIoU (%)	Accuracy (%)	mF1 (%)	FwIoU (%)
<b>1</b>	65,81 ± 4,15	60,58 ± 7,30	86,41 ± 1,78	<b>68,07 ± 3,05</b>	<b>62,58 ± 6,06</b>	<b>87,21 ± 1,10</b>
<b>2</b>	<b>82,75 ± 1,11</b>	<b>64,21 ± 1,87</b>	<b>79,82 ± 1,26</b>	82,08 ± 1,49	60,22 ± 1,06	78,89 ± 0,74
<b>3</b>	78,13 ± 1,01	51,95 ± 1,07	79,87 ± 0,73	<b>79,27 ± 1,34</b>	<b>52,82 ± 1,31</b>	<b>80,17 ± 1,36</b>
<b>4*</b>	<b>84,25 ± 0,06</b>	<b>79,29 ± 0,6</b>	<b>81,32 ± 0,11</b>	81,29 ± 0,87	76,41 ± 0,25	78,84 ± 0,51

\* Se utilizó 2 iteraciones del 4-Fold CV.

En la Tabla 4.2 se muestra el rendimiento de los modelos en los distintos predios para cada categoría sobre los conjuntos de test para la métrica de F1-score. Esta tabla evidencia algunas situaciones que se discuten a continuación. Adicionalmente, los resultados de F1-score, Precisión y Recall se encuentran disponibles en las tablas de el Anexo A.4.

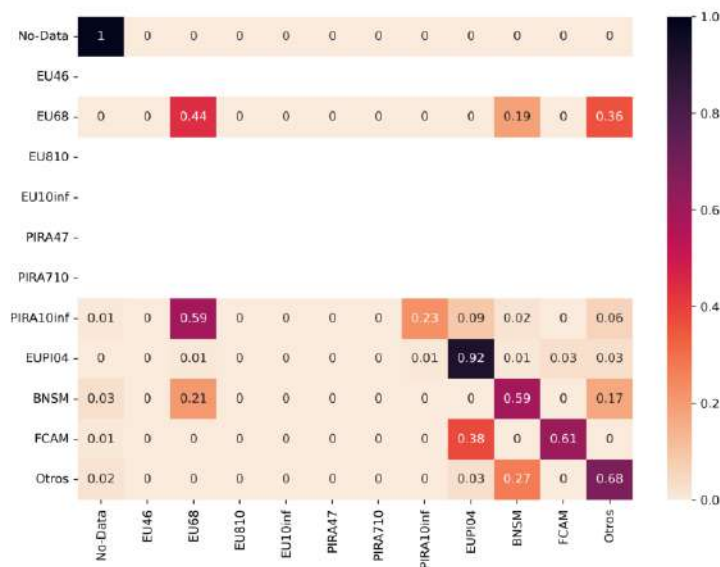
**Tabla 4.2:** Cross validation F1 (%) de los modelos sobre el conjunto de test.

Predio	1		2		3		4		
	FCN	MP-ResNet	FCN	MP-ResNet	FCN	MP-ResNet	FCN	MP-ResNet	
F1 (%)	EU46	-	-	-	94,12 ± 1,36	<b>94,68 ± 0,82</b>	<b>88,96 ± 0,43</b>	86,04 ± 1,86	
	EU68	34,87 ± 27,5	<b>38,88 ± 23,91</b>	-	-	-	<b>88,75 ± 0,24</b>	86,91 ± 0,47	
	EU810	-	-	-	-	-	<b>71,52 ± 0,33</b>	67,73 ± 0,68	
	EU10inf	-	-	-	-	<b>76,01 ± 2,12</b>	74,62 ± 11,06	<b>82,98 ± 0,02</b>	80,2 ± 0,30
	PIRA47	-	-	90,91 ± 1,09	<b>91,14 ± 1,23</b>	-	-	<b>95,84 ± 1,13</b>	91,57 ± 5,34
	PIRA710	-	-	<b>69,84 ± 4,99</b>	68,57 ± 1,46	0,00 ± 0,00	0,00 ± 0,01	-	-
	PIRA10inf	65,45 ± 18,28	<b>75,98 ± 7,67</b>	<b>31,01 ± 7,36</b>	18,88 ± 3,73	<b>87,41 ± 0,57</b>	87,0 ± 2,91	90,05 ± 0,10	<b>90,39 ± 0,68</b>
	EUPI04	91,61 ± 0,35	<b>91,88 ± 0,48</b>	-	-	10,74 ± 7,49	<b>12,75 ± 2,95</b>	51,55 ± 4,99	<b>60,58 ± 5,89</b>
	BNSM	<b>55,77 ± 4,32</b>	50,14 ± 3,36	<b>48,52 ± 10,03</b>	43,78 ± 1,99	61,63 ± 2,01	<b>62,85 ± 1,38</b>	<b>86,19 ± 0,00</b>	81,85 ± 1,19
	FCAM	63,04 ± 3,61	<b>65,90 ± 2,53</b>	<b>55,56 ± 6,54</b>	49,80 ± 1,86	75,41 ± 1,29	<b>78,19 ± 0,71</b>	<b>69,85 ± 0,05</b>	68,98 ± 1,43
OTROS	52,71 ± 10,99	<b>52,73 ± 5,24</b>	<b>89,44 ± 1,35</b>	89,16 ± 0,71	10,25 ± 10,78	<b>12,44 ± 6,42</b>	<b>67,22 ± 0,13</b>	49,86 ± 2,28	

#### 4.1.1. Predio 1

En el predio 1 para F1-score, se puede apreciar que MP-ResNet obtiene mejores rendimientos en cinco de las seis categorías presentes. Tanto FCN como MP-ResNet tienen buenos rendimientos promedios sobre EUPI04 (sobre el 91,61%) seguido por PIRA10inf (65,45% para FCN y 75,98% para MP-ResNet). Dado lo evidenciado en la Tabla 3.2, EUPI04 es la categoría predominante del predio con 36,48% de los pixeles etiquetados, seguida por PIRA10inf con un 16,50%. Considerando lo anterior pareciera haber una estrecha relación entre la cantidad de datos por categoría y el rendimiento de estas. Pero, esto no ocurre igual para todas, FCAM tiene el tercer mejor rendimiento promedio igual a 63,04% siendo la categoría con menor cantidad de pixeles (7,11%). Esto evidencia que además de la cantidad de pixeles por categoría, la naturaleza de estas también es un factor importante para el

aprendizaje de los modelos. La categoría más débil en este predio es EU68 en el modelo FCN, esto se debe a la dificultad que presenta el modelo para diferenciar entre esta y otras categorías. Se puede evidenciar en la matriz de confusión de la Figura 4.1, donde solo el 44 % de EU68 fue clasificado correctamente, un 19 % fue clasificado como BNSM y un 36 % como OTROS, y un 59 % de PIRA10inf y un 21 % de BSNM fueron clasificados como EU68. Todas las matrices de confusión se encuentran disponibles en el Anexo A.10.

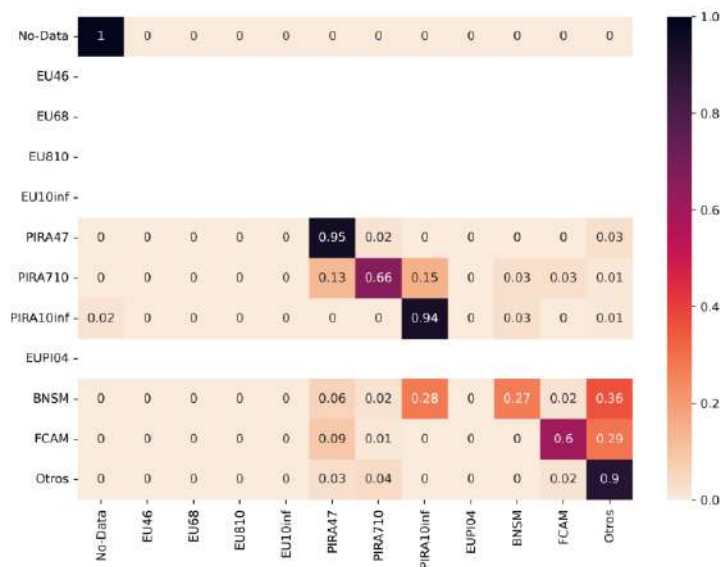


**Figura 4.1:** Matriz de confusión sobre el conjunto de test del predio 1, FCN iteración 1.

#### 4.1.2. Predio 2

En el predio 2 para F1-score, se puede apreciar que existe una superioridad del modelo FCN en cinco de las seis categorías presentes. En este predio se pueden apreciar buenos rendimientos promedios para F1-score en ambos modelos sobre PIRA47 y OTROS, aproximadamente 91 % y 89 %, respectivamente. La categoría que tiene peor rendimiento promedio es PIRA10inf con un 31,01 % para FCN y apenas un 18,88 % para MP-ResNet, también es la categoría que presenta mayor diferencia de rendimiento entre modelos. PIRA10inf (5,02 % de presencia en el predio, Tabla 3.2) muestra peor rendimiento que las categorías menos presentes del predio, BNSM y FCAM con 5,60 % y 3,20 % en el predio respectivamente. En la Figura 4.2 se puede apreciar

una iteración de MP-ResNet, donde el 94 % de PIRA10inf es clasificado correctamente, pero el modelo ha clasificado un 15 % de PIRA710 y un 28 % de BSNM como PIRA10inf, es decir, falsos positivos. Este comportamiento es similar para las demás iteraciones y provoca que la precisión promedio de F1-score sea deficiente, un 10,51 %. En el Anexo A.13 se puede encontrar las matrices de confusión para las iteraciones de MP-ResNet, y en el Anexo A.3, la tabla de F1-Score, Precisión y Recall.



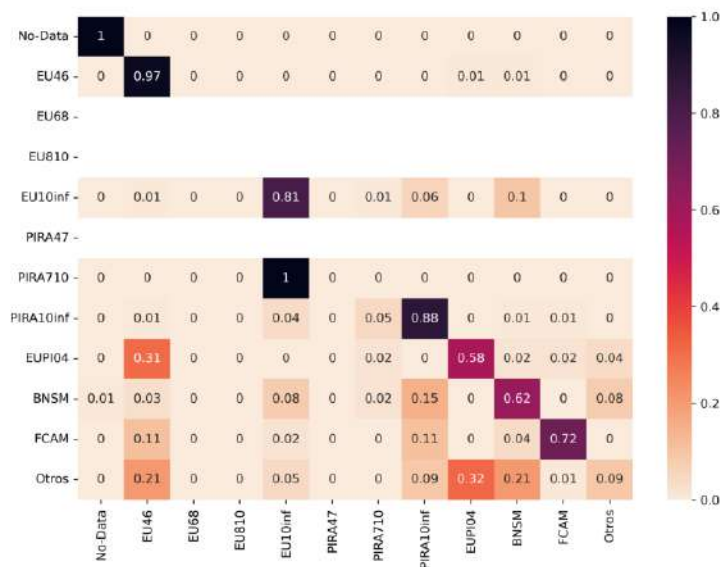
**Figura 4.2:** Matriz de confusión sobre el conjunto de test del predio 2, MP-ResNet iteración 4.

### 4.1.3. Predio 3

En el predio 3 para F1-score, se puede apreciar que MP-ResNet tiene la ventaja en cinco de las nueve categorías. Los modelos presentan diferencias de rendimientos promedios entre categorías menores a 3 puntos porcentuales. Las categoría con mejor rendimiento promedio es EU46 (94,12 % para FCN y 94,68 % para MP-ResNet), seguido por PIRA10inf (87,41 % para FCN y 87,0 % para MP-ResNet). Hay tres categorías con bajos desempeños en F1-score y son, EUPI04 (10,74 % para FCN y 12,75 % para MP-ResNet), OTROS (10,25 % para FCN y 12,44 % para MP-ResNet), y PIRA710 (0 % para ambos modelos). En la matriz de confusión de la Figura



4.3 se muestra la confusión entre las categorías. Entre las más importantes se encuentra que el 31 % de EUPI04 que ha sido clasificado como EU46, 32 % de BNSM ha sido clasificado como EUPI04, OTRO se ha distribuido mayoritariamente entre EUPI04, EU46 y BNSM y el 100 % de PIRA710 ha sido clasificado como EU10inf. Observando la Tabla 3.2 se puede apreciar que las tres categorías con más bajos desempeños también tienen los menores porcentajes de píxeles en el predio.



**Figura 4.3:** Matriz de confusión sobre el conjunto de test del predio 3, MP-ResNet iteración 2.

#### 4.1.4. Predio 4

En el predio 4, FCN tiene la ventaja en F1-score en ocho de diez categorías. FCN tiene un rendimiento promedio de 51,55 % en EUPI04, siendo su categoría más débil. Las demás categorías tienen un rendimiento promedio mayor a 67 % y menor e igual a 95,84 %. En la Figura 4.4 se muestra la matriz de confusión de una iteración de FCN. La matriz de confusión muestra la confusión que existe entre la clase EUPI04 y EU46, donde el 48 % de EUPI04 es clasificado como EU46.

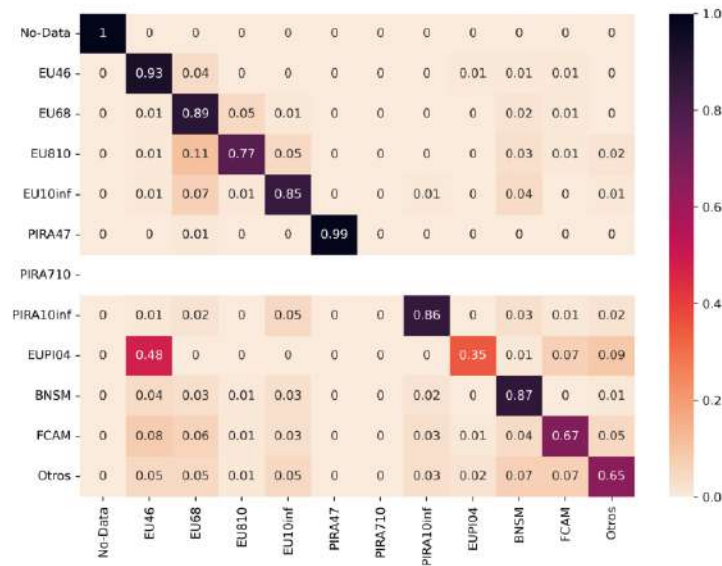


Figura 4.4: Matriz de confusión sobre el conjunto de test del predio 4, FCN iteración 1.

## 4.2. Observaciones adicionales

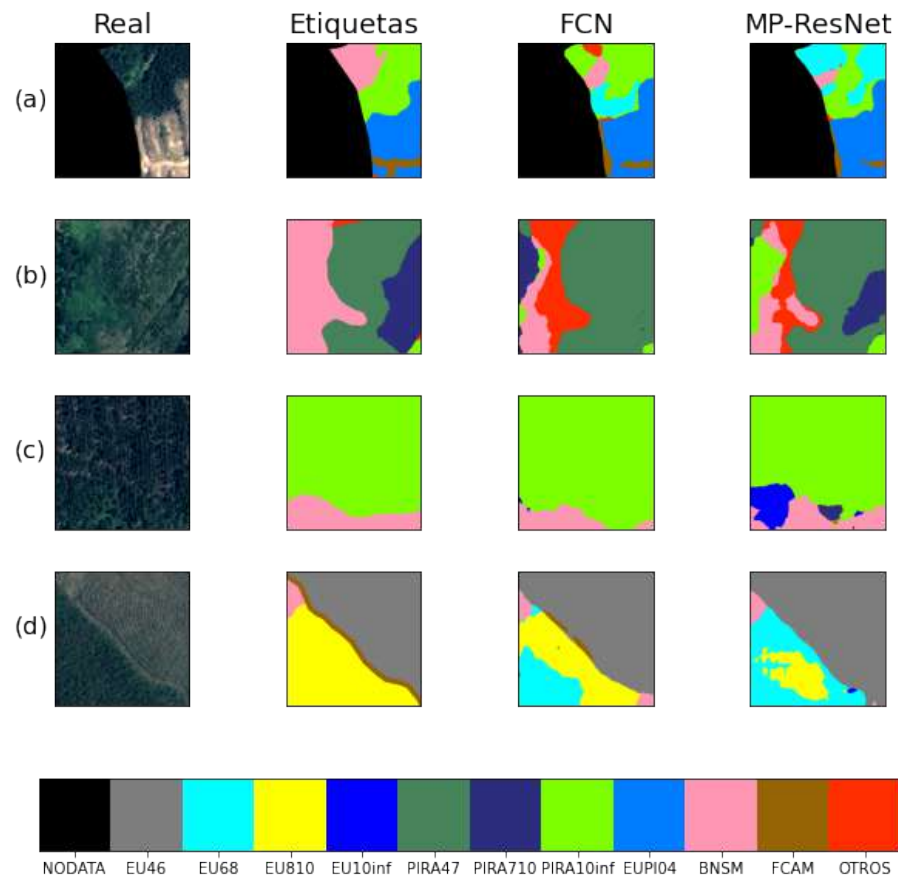
En los distintos predios ha sido común encontrar relación entre el rendimiento de los modelos en ciertas categorías con la cantidad de píxeles de esta en el predio. Esto da a entender que una mayor cantidad de píxeles es favorable para el aprendizaje de la categoría. Así se demuestra con EUPI04 en el predio 1, PIRA47 y OTROS en el predio 2, EU46 y PIRA10inf en el predio 3. Para el predio 4, cada categoría considerada tiene una gran cantidad de píxeles, por lo que existen altos valores de F1-score para la mayoría de estas. Aun así, hay problemas con algunas categorías producto de la naturaleza de las categorías como lo es EUPI04 que se confunde con EU46. Es natural esta confusión si se piensa que dentro de la categoría EUPI04 existen eucaliptus cercanos a los cuatro años con características similares a los eucaliptus de entre cuatro y seis años. Además, hay una alta relación entre los peores rendimientos y la escasez de etiquetas a pesar de los esfuerzos por mitigar los efectos del desbalance mediante la técnica de aumentación de datos “Copy-Paste”. En el predio 3 también se confunde EUPI04 con EU46, pero en este caso el modelo tiene muchos menos píxeles de esta categoría para aprender, obteniendo un peor rendimiento. En todos los predios hay confusiones que se pueden tratar de explicar por la

naturaleza de las categorías. Por ejemplo en el predio 2, la confusión entre PIRA47, PIRA710 y PIRA10inf, es entre árboles de la misma especie que difieren en edad con límites rígidos entre una categoría y otra. Esto mismo sucede para EU46, EU68 y EU10inf en el predio 4.

Dado que en este trabajo no se encontraron grandes diferencias entre los dos modelos y se evidencia malos rendimientos en presencia de pocas etiquetas para algunas categorías. Puede ser interesante en un trabajo futuro, entrenar un único modelo que utilice un único conjunto de datos formados por imágenes provenientes de todos los predios, para ser evaluado sobre cada predio de forma independiente y pueda ser comparado con el enfoque desarrollado en el presente trabajo.

### 4.3. Visualización de predicciones

En la Figura 4.5, se muestra un ejemplo de las predicciones realizadas por los modelos en los distintos predios. En esta, se puede apreciar de forma visual algunos de los errores. Por ejemplo: Para el predio 1, FCN y MP-ResNet, clasifican gran parte de PIRA10inf como EU68. Para el predio 2, FCN y MP-ResNet, clasifican gran parte de BNSM como OTROS. Para el predio 3, parte de BNSM es clasificada como PIRA10inf por FCN y como EU10inf por MP-ResNet. Finalmente en el predio 4, FCN y MP-Resnet, clasifican parte de EU810 como EU68. Adicionalmente, en el Anexo A.5 se encuentra una mayor cantidad de predicciones por predio, modelo e iteración sobre sus respectivos conjuntos de test.



**Figura 4.5:** Predicciones: (a) Predio 1; (b) Predio 2; (c) Predio 3; (d) Predio 4.

# Capítulo 5

## Conclusión

En este trabajo, se aborda el problema de la clasificación y segmentación de uso de suelo para cuatro predios forestales. Mediante el uso de imágenes satelitales y modelos de aprendizaje profundo de segmentación semántica. Los modelos usados son la Fully Convolutional Network (FCN) y Multi-path Residual Network (MP-ResNet). Estos se implementan, entrenan y evalúan con una metodología que permite una comparación justa entre estos a nivel de predios y para cada categoría considerada. Para el entrenamiento se utilizan las técnicas de Data augmentation (Random-Flipping, Random-Rotate y Copy-Past) y la técnica Fine-Tuning, utilizando las capas de la ResNet34 como extractor de características en ambos modelos para mejorar el desempeño de la generalización de los modelos. Además, se realiza una búsqueda de hiperparámetros y para la evaluación de los modelos se utiliza la técnica de 4-Fold Cross Validation.

Los principales resultados de este trabajo, muestran que ninguno de los dos modelos tiene una ventaja clara sobre el otro en términos generales. Debido a que cada uno tiene el mejor desempeño en dos predios. Para ambos modelos en los distintos predios, la métrica FwIoU, tiene un rendimiento promedio mayor a 78,84% y menor e igual a 87,21%. Mientras, Accuracy tiene un rendimiento promedio mayor a 65,81% y menor e igual a 84,25%, y mF1 un rendimiento promedio mayor a 51,95% y menor e igual a 79,29%. Los modelos muestran un gran potencial para la tarea de clasificar usos de suelo, ya que si bien existen rendimientos deficientes en algunas categorías. Estas se atribuyen a una gran similitud natural con otras y/o a la escasez

---

de presencia en el predio, a pesar de la implementación estratégica de la técnica Copy-Paste.

# Bibliografía

- H. C. Arteaga. Técnicas de aprendizaje supervisado y no supervisado para el aprendizaje automatizado de computadoras. En *Memorias del primer Congreso Internacional de Ciencias Pedagógicas: Por una educación integral, participativa e incluyente*, páginas 549–564. Instituto Superior Tecnológico Bolivariano, 2015.
- V. Badrinarayanan, A. Kendall, y R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.
- R. Benítez, G. Escudero, S. Kanaan, y D. M. Rodó. *Inteligencia artificial avanzada*. Editorial UOC, 2014.
- M. Cardemil. Industria forestal en Chile (serie minutas n° 68-21), 2021. URL [https://www.bcn.cl/obtienearchivo?id=repositorio/10221/32419/1/N\\_68\\_21\\_Industria\\_Forestal\\_en\\_Chile.pdf](https://www.bcn.cl/obtienearchivo?id=repositorio/10221/32419/1/N_68_21_Industria_Forestal_en_Chile.pdf).
- L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, y A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.
- L. Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.
- L. Ding, K. Zheng, D. Lin, Y. Chen, B. Liu, J. Li, y L. Bruzzone. Mp-resnet: Multipath residual network for the semantic segmentation of high-resolution polsar images. *IEEE Geoscience and Remote Sensing Letters*, PP:1–5, 05 2021.

- J. Domke. Generic methods for optimization-based modeling. En N. D. Lawrence y M. Girolami, editores, *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volumen 22 of *Proceedings of Machine Learning Research*, páginas 318–326, La Palma, Canary Islands, 21–23 Apr 2012. PMLR.
- L.-F. Dong, Y.-Z. Gan, X.-L. Mao, Y.-B. Yang, y C. Shen. Learning deep representations using convolutional auto-encoders with symmetric skip connections. En *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, páginas 3006–3010. IEEE, 2018.
- Z. Du, J. Yang, C. Ou, y T. Zhang. Smallholder crop area mapped with a semantic segmentation deep learning method. *Remote Sensing*, 11(7), 2019.
- G. A. Fricker, J. D. Ventura, J. A. Wolf, M. P. North, F. W. Davis, y J. Franklin. A convolutional neural network classifier identifies tree species in mixed-conifer forest from hyperspectral imagery. *Remote Sensing*, 11(19), 2019.
- K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36:193–202, 1980.
- A. Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. "O'Reilly Media, Inc.", 2019.
- G. Ghiasi, Y. Cui, A. Srinivas, R. Qian, T.-Y. Lin, E. D. Cubuk, Q. V. Le, y B. Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. En *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, páginas 2918–2928, 2021.
- K. He, X. Zhang, S. Ren, y J. Sun. Deep residual learning for image recognition. En *Proceedings of the IEEE conference on computer vision and pattern recognition*, páginas 770–778, 2016.
- A. Krizhevsky, I. Sutskever, y G. E. Hinton. Imagenet classification with deep convolutional neural networks. En F. Pereira, C. Burges, L. Bottou, y K. Weinberger, editores, *Advances in Neural Information Processing Systems*. Curran Associates, Inc.



- Y. Le Cun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, y W. Hubbard. Handwritten digit recognition: Applications of neural network chips and automatic learning. *IEEE Communications Magazine*, 27(11):41–46, 1989.
- J. Long, E. Shelhamer, y T. Darrell. Fully convolutional networks for semantic segmentation. páginas 3431–3440, 2015.
- R. M Rustowicz, R. Cheong, L. Wang, S. Ermon, M. Burke, y D. Lobell. Semantic segmentation of crop type in africa: A novel dataset and analysis of deep learning methods. En *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, páginas 75–82, 2019.
- V. Mirjalili y S. Raschka. *Python machine learning*. Marcombo, 2020.
- T. T. Nguyen, T. D. Hoang, M. T. Pham, T. T. Vu, T. H. Nguyen, Q.-T. Huynh, y J. Jo. Monitoring agriculture areas with satellite images and deep learning. *Applied Soft Computing*, 95:106565, 2020. ISSN 1568-4946.
- M. Onishi y T. Ise. Explainable identification and mapping of trees using UAV RGB image and deep learning. *Scientific reports*, 11(1):1–15, 2021.
- C. Poblete-Echeverría, G. F. Olmedo, B. Ingram, y M. Bardeen. Detection and segmentation of vine canopy in ultra-high spatial resolution RGB imagery obtained from unmanned aerial vehicle (UAV): A case study in a commercial vineyard. *Remote Sensing*, 9(3), 2017.
- R. P. Poudel, U. Bonde, S. Liwicki, y C. Zach. Contextnet: Exploring context and detail for semantic segmentation in real-time. *arXiv preprint arXiv:1805.04554*, 2018.
- R. P. Poudel, S. Liwicki, y R. Cipolla. Fast-scnn: Fast semantic segmentation network. *arXiv preprint arXiv:1902.04502*, 2019.
- H. Robbins y S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, páginas 400–407, 1951.

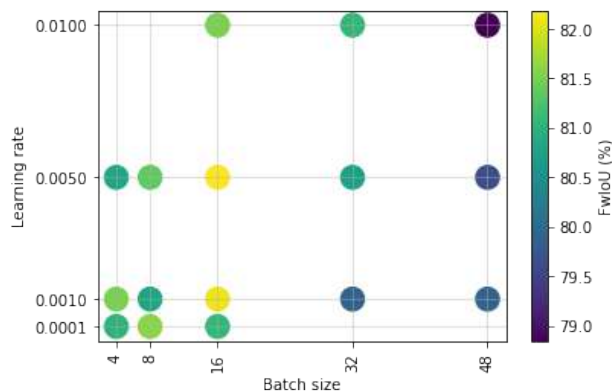
- O. Ronneberger, P. Fischer, y T. Brox. U-net: Convolutional networks for biomedical image segmentation. En *International Conference on Medical image computing and computer-assisted intervention*, páginas 234–241. Springer, 2015.
- F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.
- D. E. Rumelhart, G. E. Hinton, y R. J. Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.
- P. Surovỳ, N. Almeida Ribeiro, y D. Panagiotidis. Estimation of positions and heights from uav-sensed imagery in tree plantations in agrosilvopastoral systems. *International Journal of Remote Sensing*, 39(14):4786–4800, 2018.
- J. Torres. *Python deep learning: introducción práctica con Keras y TensorFlow 2*. Marcombo, 2020.
- L. Vogado, R. Veras, K. Aires, F. Araújo, R. Silva, M. Ponti, y J. M. R. Tavares. Diagnosis of leukaemia in blood slides based on a fine-tuned and highly generalisable deep learning model. *Sensors*, 21(9):2989, 2021.
- L. Wallace, R. Musk, y A. Lucieer. An assessment of the repeatability of automatic forest inventory metrics derived from UAV-borne laser scanning data. *IEEE Transactions on Geoscience and Remote Sensing*, 52(11):7160–7169, 2014.
- T. Zhao, Y. Yang, H. Niu, Y. Chen, y D. Wang. Comparing u-net convolutional networks with fully convolutional networks in the performances of pomegranate tree canopy segmentation. página 64, 10 2018.

# Anexo

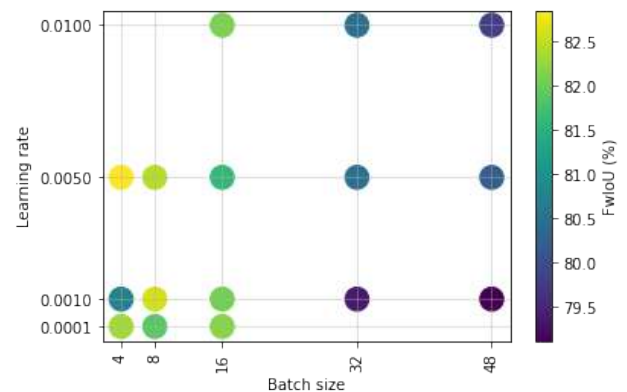
## A.1. Búsqueda de hiperparámetros

Tabla A.1: Desempeño de los hiperparámetros.

FCN					MP-ResNet				
Lr	Bz	Accuracy (%)	mF1 (%)	FwIoU (%) ▲	Lr	Bz	Accuracy (%)	mF1 (%)	FwIoU (%) ▲
0,0050	16,0	72,18 ± 9,43	69,89 ± 4,4	82,19 ± 3,97	0,0050	4,0	73,87 ± 8,07	66,75 ± 8,17	82,85 ± 1,65
0,0010	16,0	72,23 ± 10,1	68,2 ± 6,37	82,11 ± 3,72	0,0010	8,0	73,79 ± 8,61	67,99 ± 6,94	82,61 ± 2,28
0,0001	8,0	70,93 ± 10,51	64,82 ± 9,79	81,61 ± 4,7	0,0050	8,0	73,17 ± 7,63	66,74 ± 6,23	82,43 ± 3,76
0,0010	4,0	71,08 ± 10,84	64,08 ± 7,13	81,52 ± 4,31	0,0001	4,0	72,85 ± 7,16	66,2 ± 6,81	82,33 ± 2,19
0,0100	16,0	70,58 ± 9,33	66,31 ± 5,49	81,52 ± 4,2	0,0001	16,0	73,02 ± 8,65	67,54 ± 7,62	82,18 ± 3,37
0,0050	8,0	70,38 ± 9,76	67,13 ± 7,27	81,37 ± 3,62	0,0100	16,0	72,37 ± 7,31	67,36 ± 4,95	82,11 ± 2,34
0,0001	16,0	71,0 ± 10,09	65,14 ± 7,75	81,07 ± 5,45	0,0010	16,0	72,39 ± 7,36	65,09 ± 8,51	82,06 ± 1,87
0,0100	32,0	69,36 ± 10,71	66,63 ± 5,97	81,07 ± 3,28	0,0001	8,0	73,19 ± 6,35	67,02 ± 3,6	81,89 ± 3,66
0,0001	4,0	70,59 ± 10,71	63,47 ± 9,8	81,0 ± 5,58	0,0050	16,0	71,93 ± 8,78	66,92 ± 6,69	81,62 ± 2,59
0,0050	4,0	70,15 ± 8,24	66,0 ± 6,09	80,84 ± 5,3	0,0010	4,0	71,48 ± 7,93	64,27 ± 7,45	80,77 ± 2,52
0,0010	8,0	69,47 ± 9,17	65,25 ± 5,54	80,82 ± 4,93	0,0050	32,0	70,66 ± 8,16	64,12 ± 6,12	80,52 ± 2,93
0,0050	32,0	70,4 ± 9,81	66,27 ± 6,6	80,75 ± 5,57	0,0100	32,0	70,74 ± 7,35	64,61 ± 4,65	80,47 ± 3,47
0,0010	48,0	69,27 ± 9,57	63,55 ± 6,81	79,89 ± 6,13	0,0050	48,0	70,32 ± 6,68	63,54 ± 6,48	80,22 ± 3,28
0,0010	32,0	69,01 ± 10,62	63,27 ± 5,55	79,88 ± 4,2	0,0100	48,0	69,98 ± 7,27	64,05 ± 6,79	79,8 ± 2,55
0,0050	48,0	67,84 ± 11,45	64,78 ± 6,39	79,64 ± 4,31	0,0010	32,0	69,31 ± 8,42	63,63 ± 4,9	79,37 ± 3,34
0,0100	48,0	66,43 ± 11,15	61,72 ± 6,51	78,84 ± 5,81	0,0010	48,0	68,55 ± 6,67	62,78 ± 5,06	79,11 ± 3,65



(a) FCN



(b) MP-ResNet

Figura A.1: Desempeño de los hiperparámetros sobre el conjunto de validación.

## A.2. Gráficos de entrenamiento

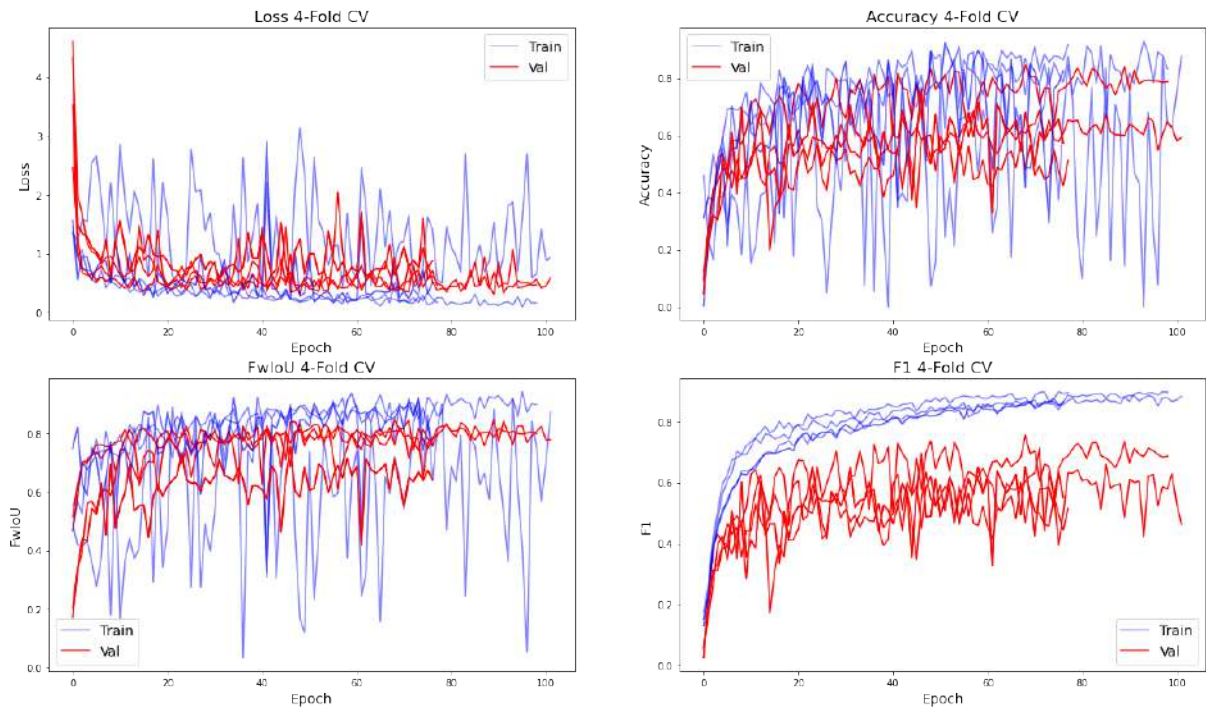


Figura A.2: Entrenamiento: Predio 1 — FCN.

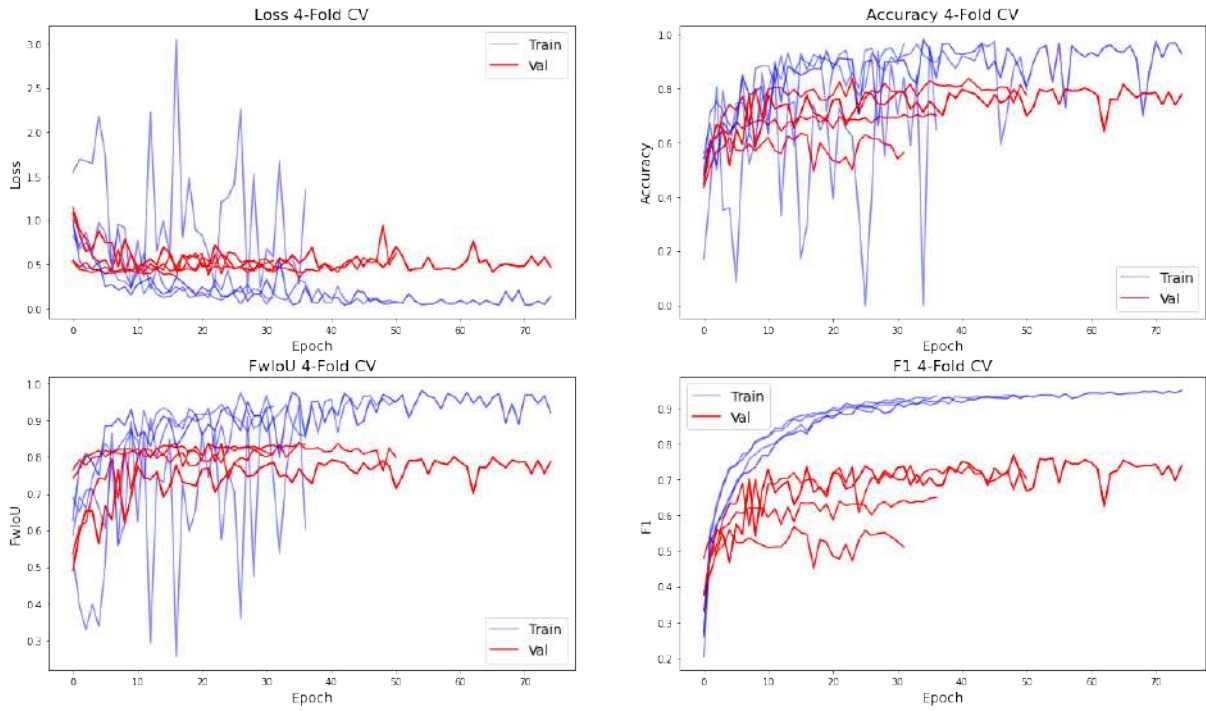


Figura A.3: Entrenamiento: Predio 1 — MP-ResNet.

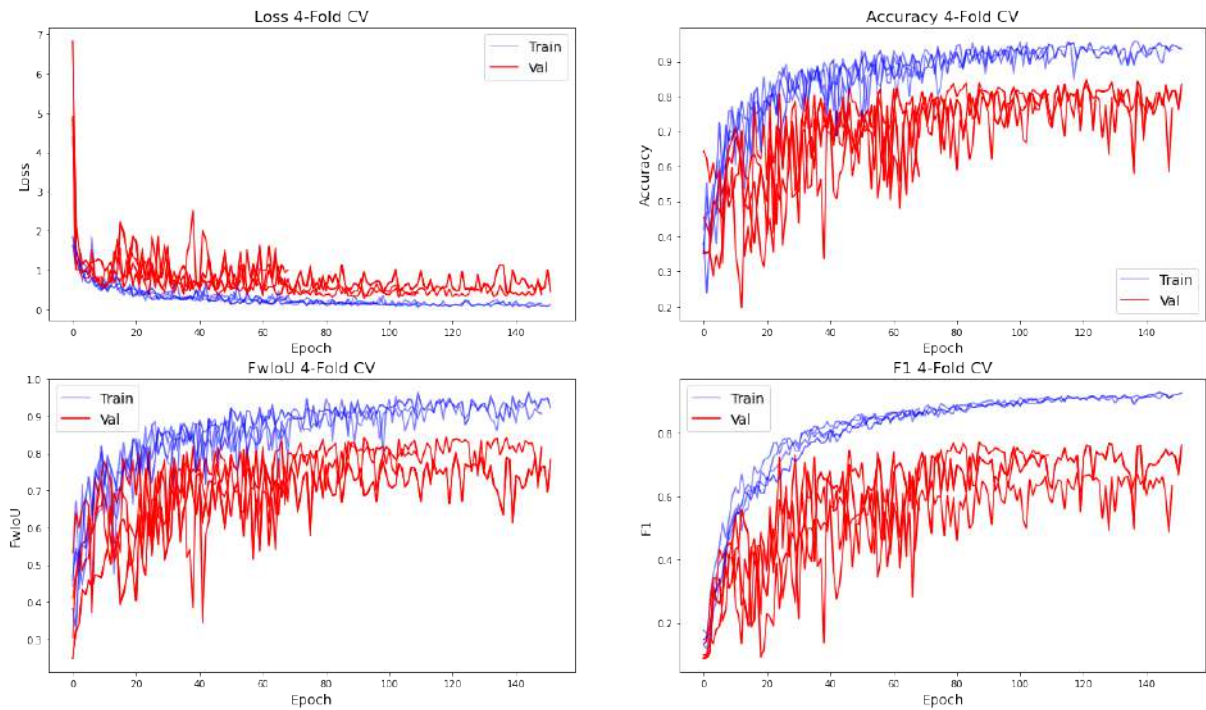


Figura A.4: Entrenamiento: Predio 2 — FCN.

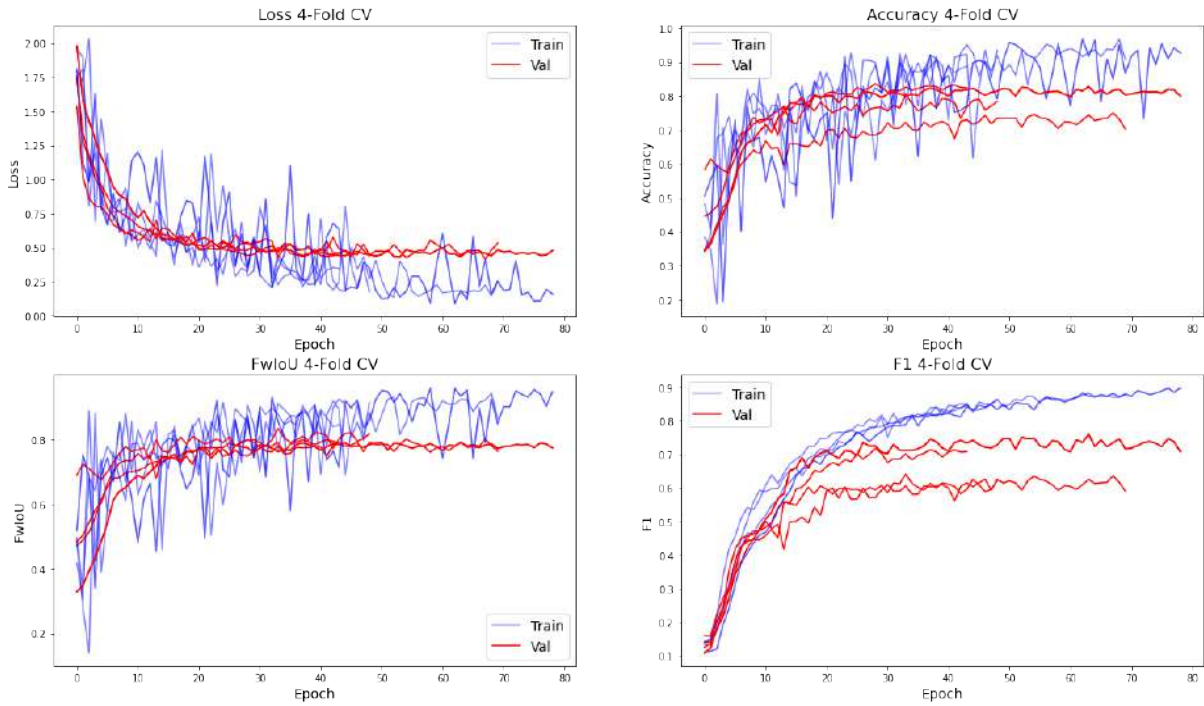


Figura A.5: Entrenamiento: Predio 2 — MP-ResNet.

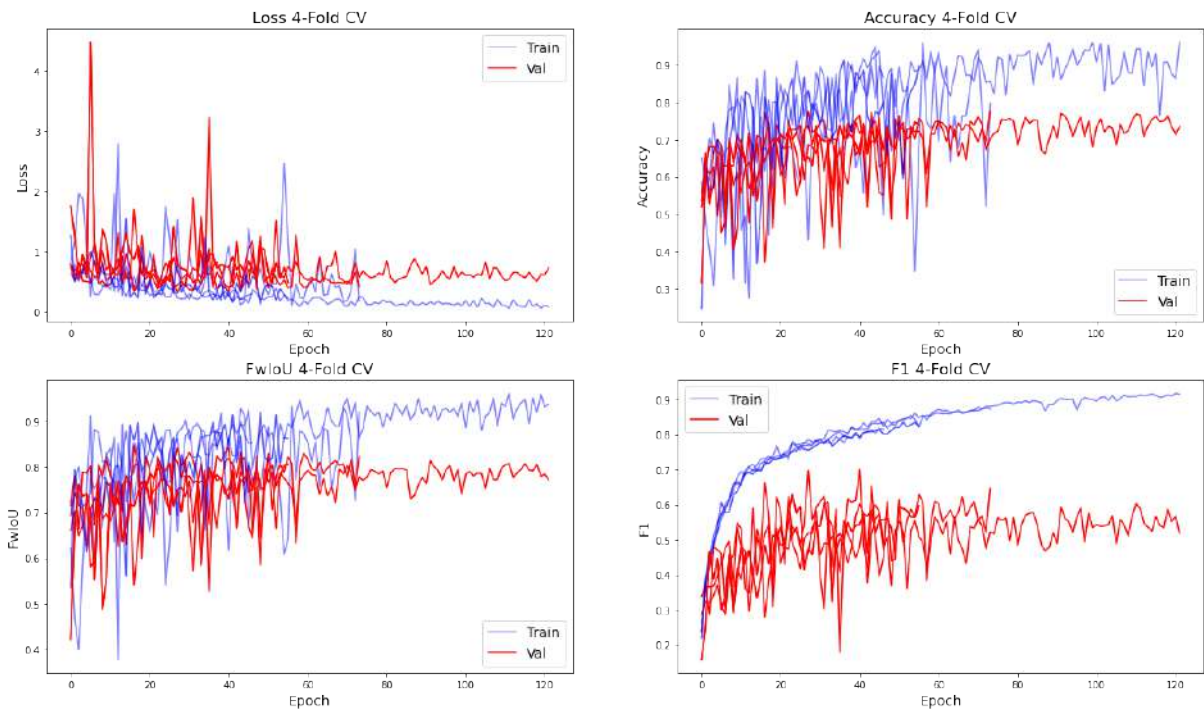


Figura A.6: Entrenamiento: Predio 3 — FCN.

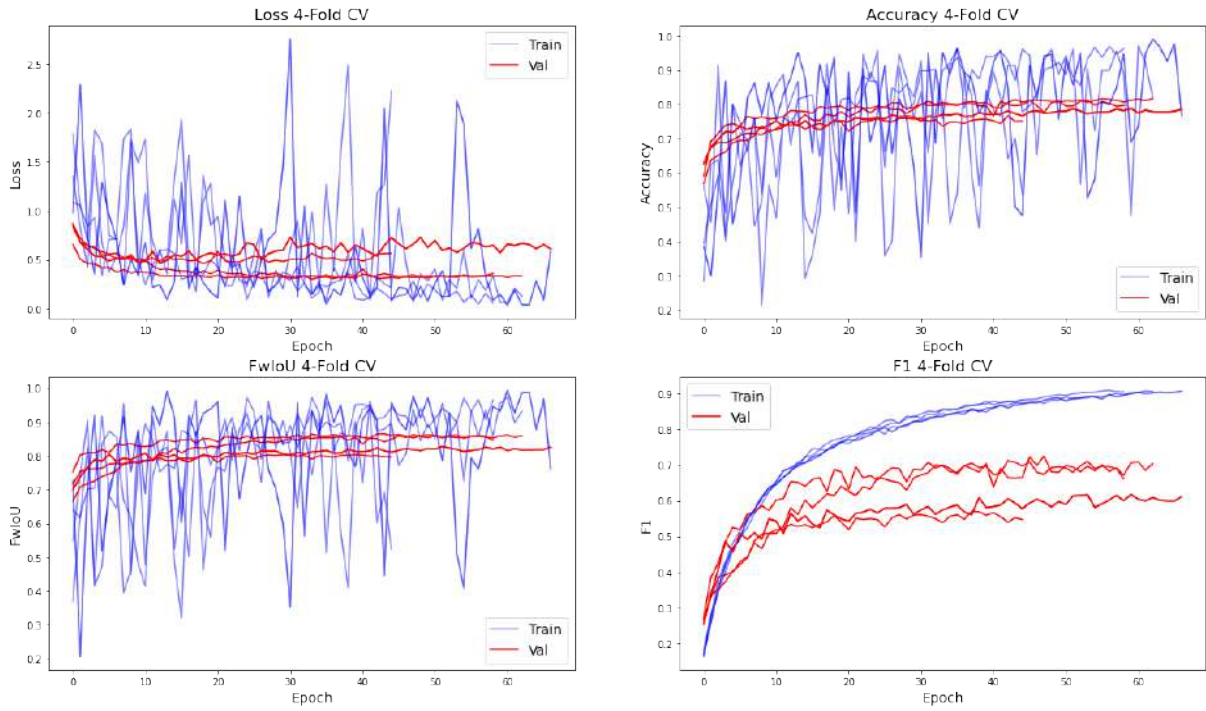


Figura A.7: Entrenamiento: Predio 3 — MP-ResNet.

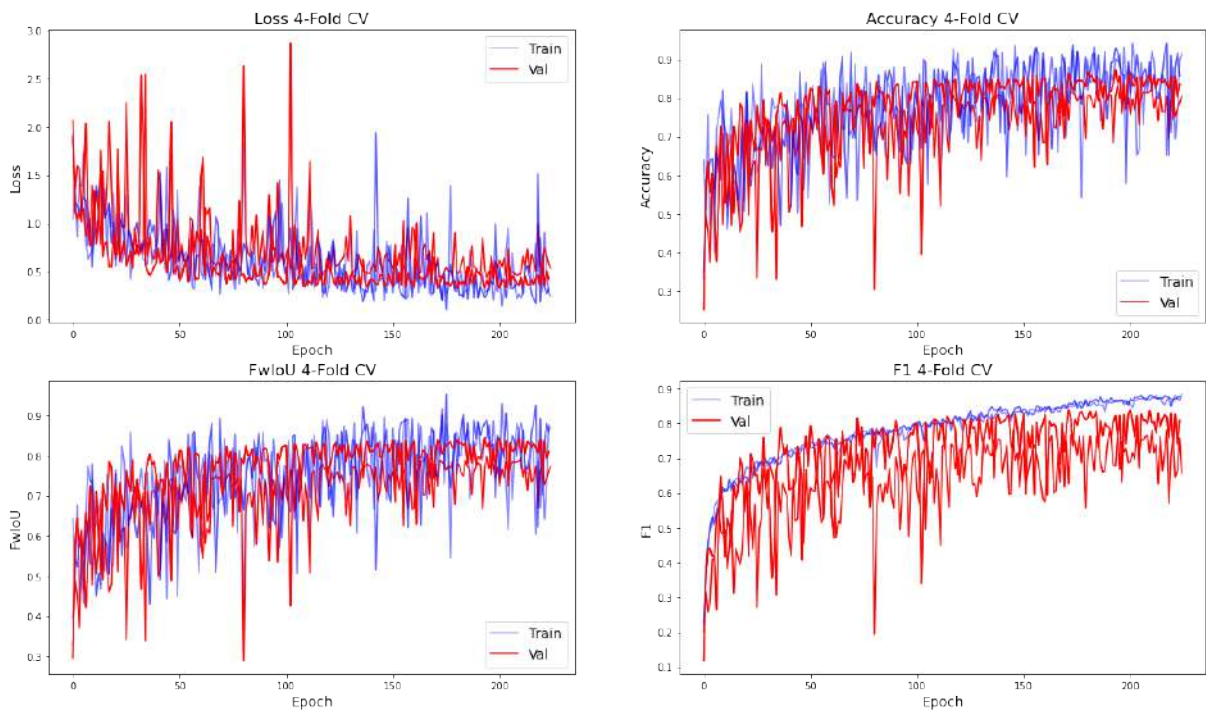


Figura A.8: Entrenamiento: Predio 4 — FCN.

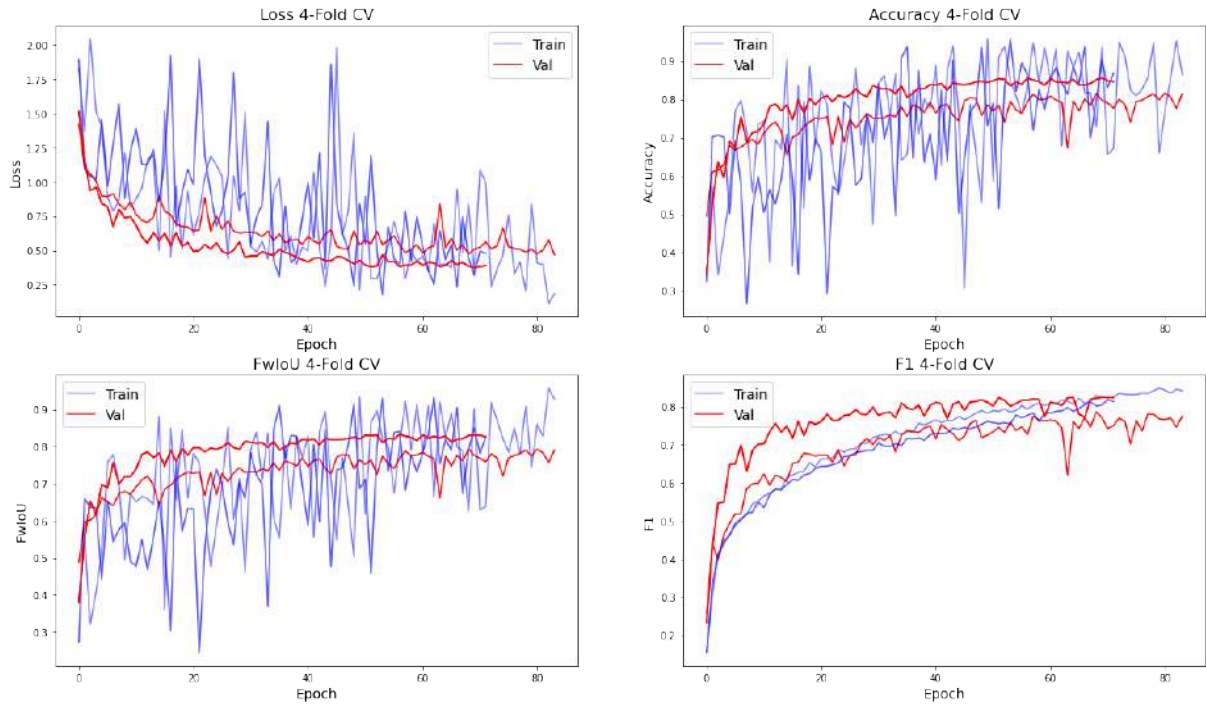
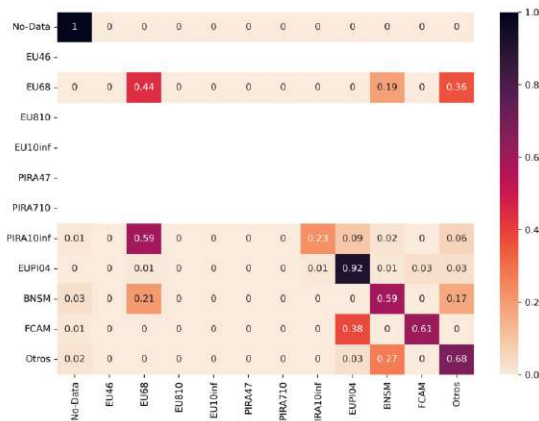


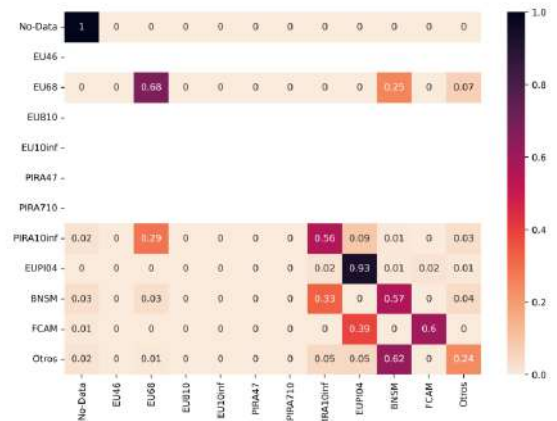
Figura A.9: Entrenamiento: Predio 4 — MP-ResNet.



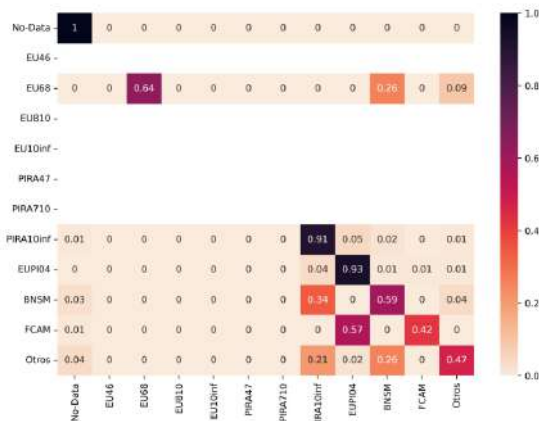
### A.3. Matriz de confusión



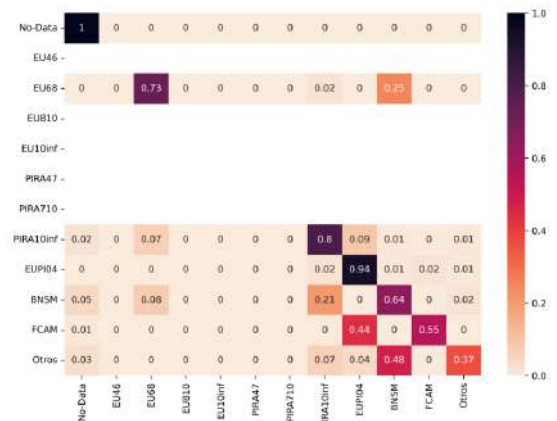
(a) Iteración 1



(b) Iteración 2

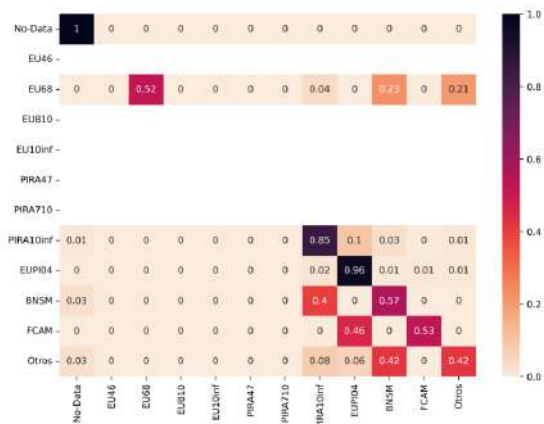


(c) Iteración 3

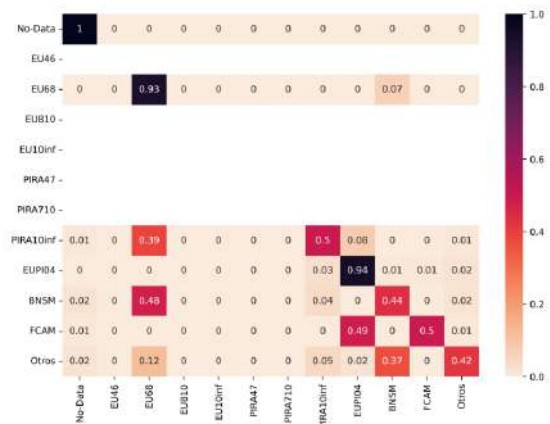


(d) Iteración 4

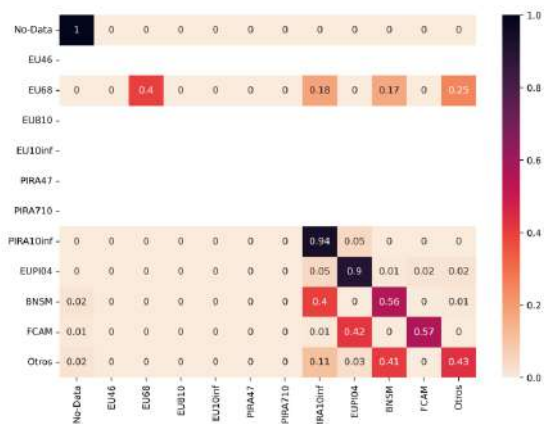
Figura A.10: Matriz de confusión — Predio 1 — FCN.



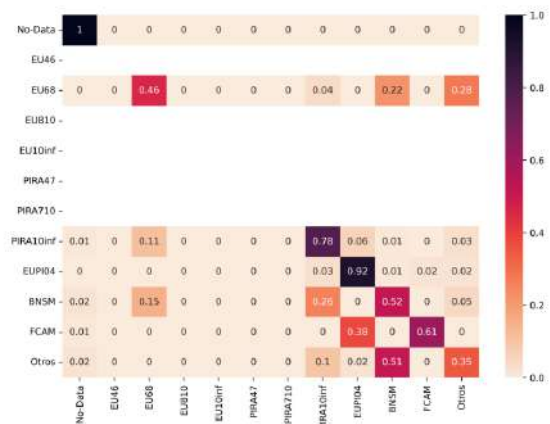
(a) Iteración 1



(b) Iteración 2



(c) Iteración 3



(d) Iteración 4

Figura A.11: Matriz de confusión — Predio 1 — MP-ResNet.



(a) Iteración 1



(b) Iteración 2



(c) Iteración 3

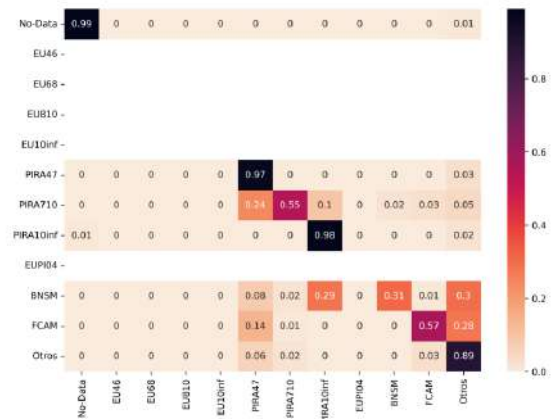


(d) Iteración 4

Figura A.12: Matriz de confusión — Predio 2 — FCN.



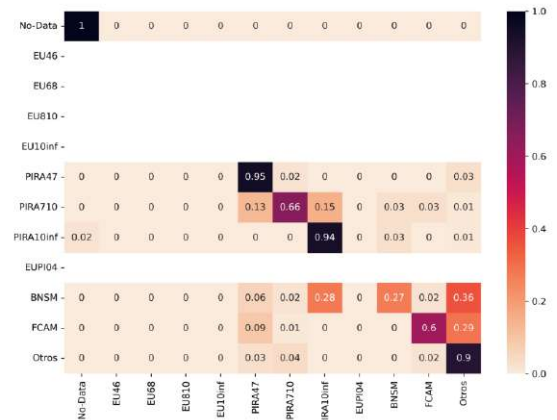
(a) Iteración 1



(b) Iteración 2

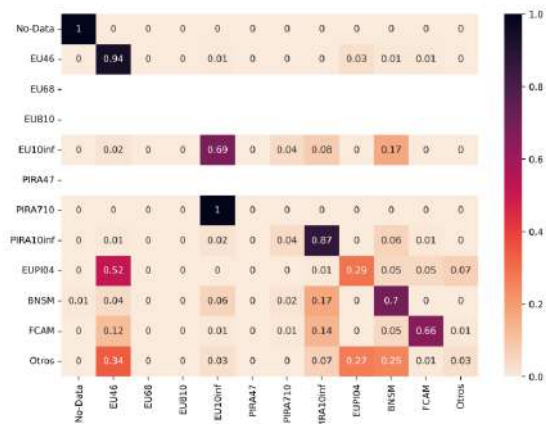


(c) Iteración 3



(d) Iteración 4

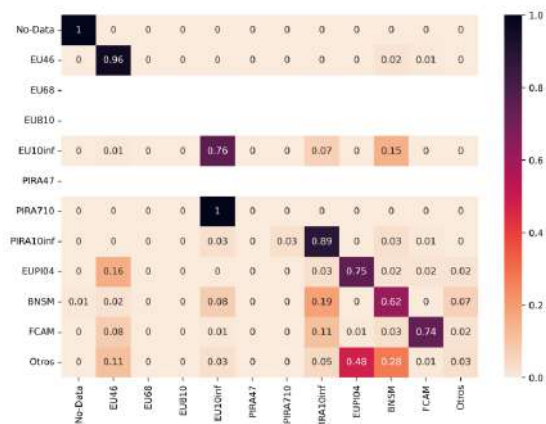
Figura A.13: Matriz de confusión — Predio 2 — MP-ResNet.



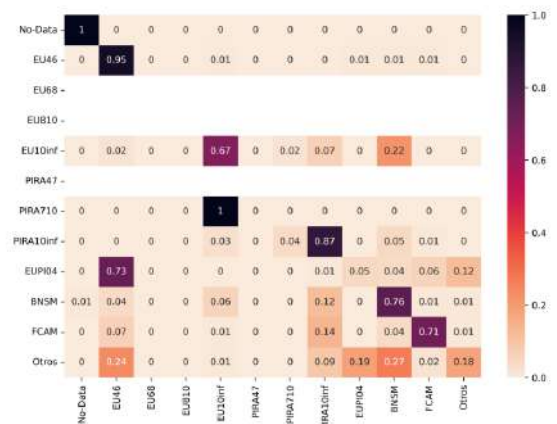
(a) Iteración 1



(b) Iteración 2

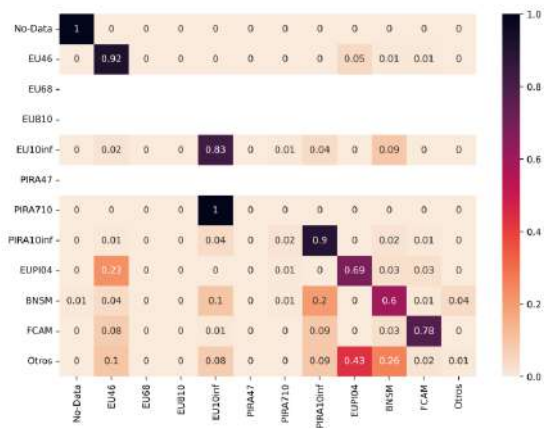


(c) Iteración 3

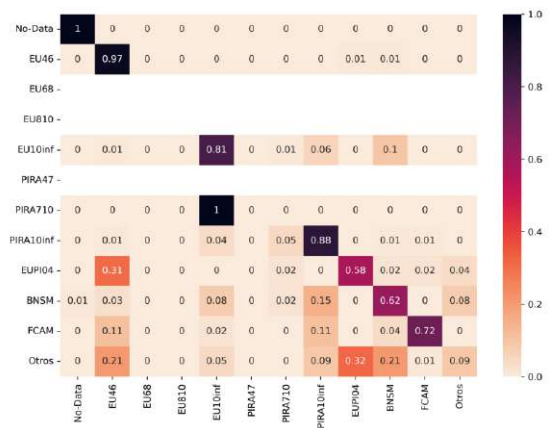


(d) Iteración 4

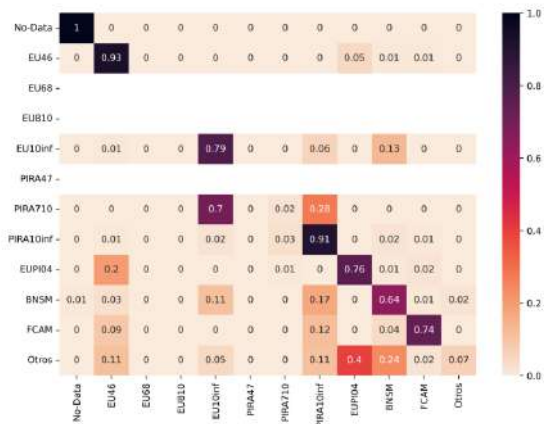
Figura A.14: Matriz de confusión — Predio 3 — FCN.



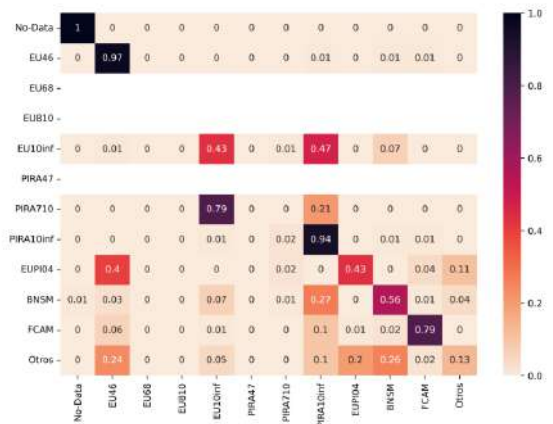
(a) Iteración 1



(b) Iteración 2



(c) Iteración 3



(d) Iteración 4

Figura A.15: Matriz de confusión — Predio 3 — MP-ResNet.

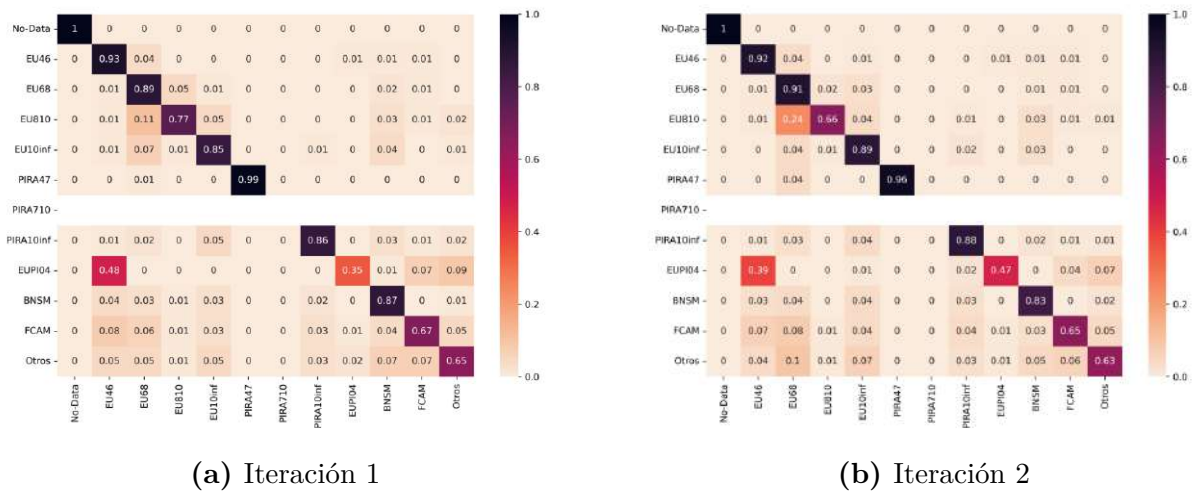


Figura A.16: Matriz de confusión — Predio 4 — FCN.

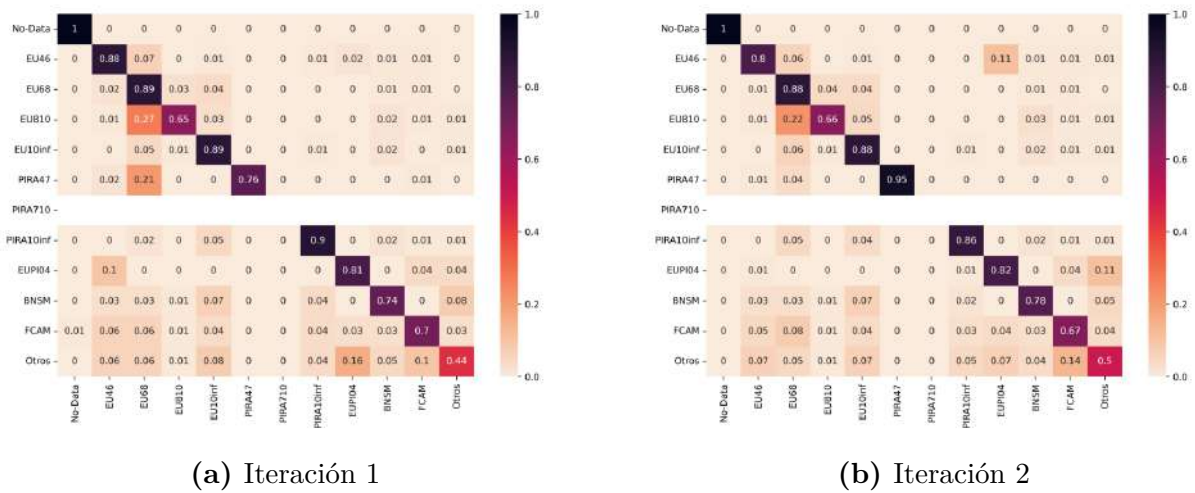


Figura A.17: Matriz de confusión — Predio 4 — MP-ResNet.

## A.4. F1 score, Precision y Recall

**Tabla A.2:** Predio 1—F1 score, Precision y Recall.

MODELO	FCN			MP-ResNet		
METRICA	F1 (%)	P (%)	R (%)	F1 (%)	P (%)	R (%)
EU46	-	-	-	-	-	-
EU68	34,87 ± 27,5	34,54 ± 38,28	62,56 ± 11,03	38,88 ± 23,91	53,54 ± 44,5	57,55 ± 20,84
EU810	-	-	-	-	-	-
EU10inf	-	-	-	-	-	-
PIRA47	-	-	-	-	-	-
PIRA710	-	-	-	-	-	-
PIRA10inf	65,45 ± 18,28	80,56 ± 9,12	62,37 ± 26,15	75,98 ± 7,67	78,21 ± 4,19	76,73 ± 16,38
EUPI04	91,61 ± 0,35	90,35 ± 0,3	92,9 ± 0,86	91,88 ± 0,48	90,86 ± 1,3	92,98 ± 1,98
BNSM	55,77 ± 4,32	52,57 ± 6,93	59,96 ± 2,75	50,14 ± 3,36	48,46 ± 2,86	52,21 ± 5,21
FCAM	63,04 ± 3,61	77,19 ± 7,04	54,48 ± 7,63	65,9 ± 2,53	82,21 ± 2,41	55,2 ± 4,22
OTROS	52,71 ± 10,99	72,26 ± 8,27	44,11 ± 15,75	52,73 ± 5,24	75,54 ± 9,68	40,55 ± 3,49

**Tabla A.3:** Predio 2—F1 score, Precision y Recall.

MODELO	FCN			MP-ResNet		
METRICA	F1 (%)	P (%)	R (%)	F1 (%)	P (%)	R (%)
EU46	-	-	-	-	-	-
EU68	-	-	-	-	-	-
EU810	-	-	-	-	-	-
EU10inf	-	-	-	-	-	-
PIRA47	90,91 ± 1,09	88,43 ± 3,71	93,75 ± 2,29	91,14 ± 1,23	87,63 ± 2,74	95,01 ± 1,03
PIRA710	69,84 ± 4,99	77,5 ± 5,59	64,42 ± 8,74	68,57 ± 1,46	78,13 ± 4,49	61,49 ± 4,08
PIRA10inf	31,01 ± 7,36	19,21 ± 5,67	86,13 ± 8,2	18,88 ± 3,73	10,51 ± 2,32	96,9 ± 1,77
EUPI04	-	-	-	-	-	-
BNSM	48,52 ± 10,03	80,98 ± 5,35	35,39 ± 10,49	43,78 ± 1,99	85,0 ± 3,74	29,51 ± 1,58
FCAM	55,56 ± 6,54	54,95 ± 11,32	58,46 ± 7,77	49,8 ± 1,86	43,5 ± 2,0	58,31 ± 2,25
OTROS	89,44 ± 1,35	86,85 ± 2,5	92,25 ± 1,3	89,16 ± 0,71	88,27 ± 0,91	90,08 ± 1,48

**Tabla A.4:** Predio 3—F1 score, Precision y Recall.

MODELO	FCN			MP-ResNet		
METRICA	F1 (%)	P (%)	R (%)	F1 (%)	P (%)	R (%)
EU46	94,12 ± 1,36	93,36 ± 1,75	94,89 ± 0,98	94,68 ± 0,82	94,54 ± 0,6	94,87 ± 2,2
EU68	-	-	-	-	-	-
EU810	-	-	-	-	-	-
EU10inf	76,01 ± 2,12	81,46 ± 2,2	71,33 ± 3,24	74,62 ± 11,06	80,07 ± 1,43	71,78 ± 16,72
PIRA47	-	-	-	-	-	-
PIRA710	0,0 ± 0,0	0,0 ± 0,0	0,0 ± 0,0	0,0 ± 0,01	0,0 ± 0,0	0,48 ± 0,84
PIRA10inf	87,41 ± 0,57	86,43 ± 2,05	88,5 ± 1,71	87,0 ± 2,91	84,17 ± 6,5	90,5 ± 2,06
EUPI04	10,74 ± 7,49	6,23 ± 4,48	42,48 ± 27,44	12,75 ± 2,95	7,31 ± 2,09	61,59 ± 12,59
BNSM	61,63 ± 2,01	56,81 ± 1,34	67,69 ± 5,69	62,85 ± 1,38	65,59 ± 1,69	60,45 ± 2,91
FCAM	75,41 ± 1,29	81,95 ± 1,36	69,92 ± 2,75	78,19 ± 0,71	81,28 ± 2,52	75,5 ± 3,0
OTROS	10,25 ± 10,78	45,8 ± 21,6	6,21 ± 6,9	12,44 ± 6,42	37,64 ± 17,79	7,68 ± 4,2



Tabla A.5: Predio 4—F1 score, Precision y Recall.

MODELO	FCN			MP-ResNet		
METRICA	F1 (%)	P (%)	R (%)	F1 (%)	P (%)	R (%)
<b>EU46</b>	88,96 ± 0,43	85,89 ± 1,01	92,27 ± 0,24	86,04 ± 1,86	88,71 ± 0,53	83,64 ± 3,97
<b>EU68</b>	88,75 ± 0,24	87,68 ± 1,37	89,88 ± 0,96	86,91 ± 0,47	85,46 ± 0,48	88,4 ± 0,46
<b>EU810</b>	71,52 ± 0,33	72,45 ± 4,77	71,3 ± 5,28	67,73 ± 0,68	70,18 ± 2,31	65,5 ± 0,75
<b>EU10inf</b>	82,98 ± 0,02	79,14 ± 1,58	87,29 ± 1,87	80,2 ± 0,3	73,28 ± 0,01	88,57 ± 0,7
<b>PIRA47</b>	95,84 ± 1,13	94,74 ± 3,73	97,12 ± 1,61	91,57 ± 5,34	98,9 ± 0,07	85,73 ± 9,34
<b>PIRA710</b>	-	-	-	-	-	-
<b>PIRA10inf</b>	90,05 ± 0,1	93,72 ± 1,13	86,69 ± 1,14	90,39 ± 0,68	92,83 ± 0,53	88,11 ± 1,77
<b>EUPI04</b>	51,55 ± 4,99	70,05 ± 0,75	41,06 ± 6,0	60,58 ± 5,89	48,72 ± 7,7	81,45 ± 0,58
<b>BNSM</b>	86,19 ± 0,0	87,28 ± 1,99	85,21 ± 1,9	81,85 ± 1,19	88,49 ± 0,18	76,17 ± 2,2
<b>FCAM</b>	69,85 ± 0,05	74,09 ± 1,21	66,1 ± 0,88	68,98 ± 1,43	69,64 ± 1,48	68,32 ± 1,38
<b>OTROS</b>	67,22 ± 0,13	70,63 ± 0,72	64,13 ± 0,83	49,86 ± 2,28	53,04 ± 1,64	47,06 ± 2,77

## A.5. Predicciones

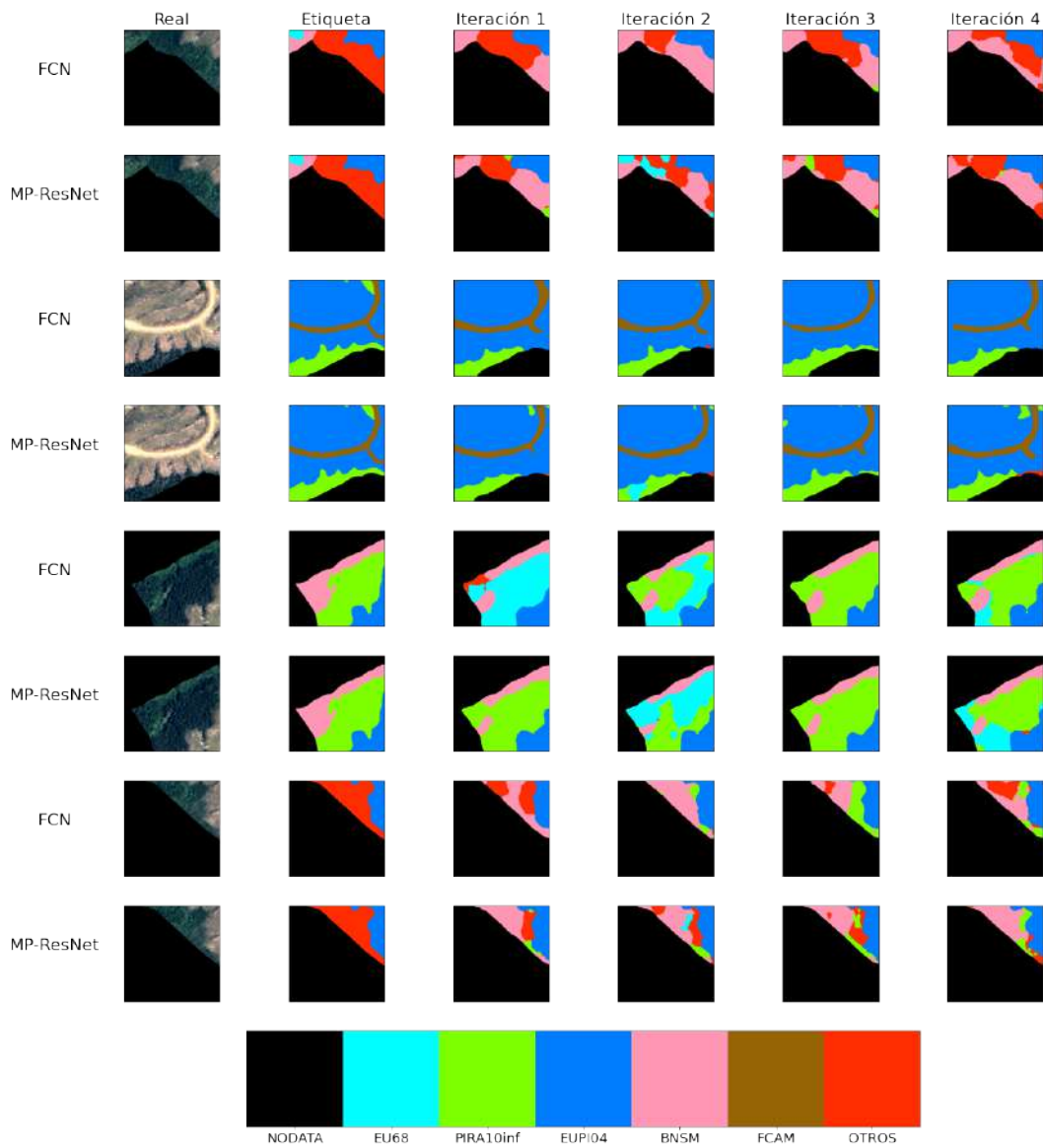


Figura A.18: Predicciones: Predio 1.

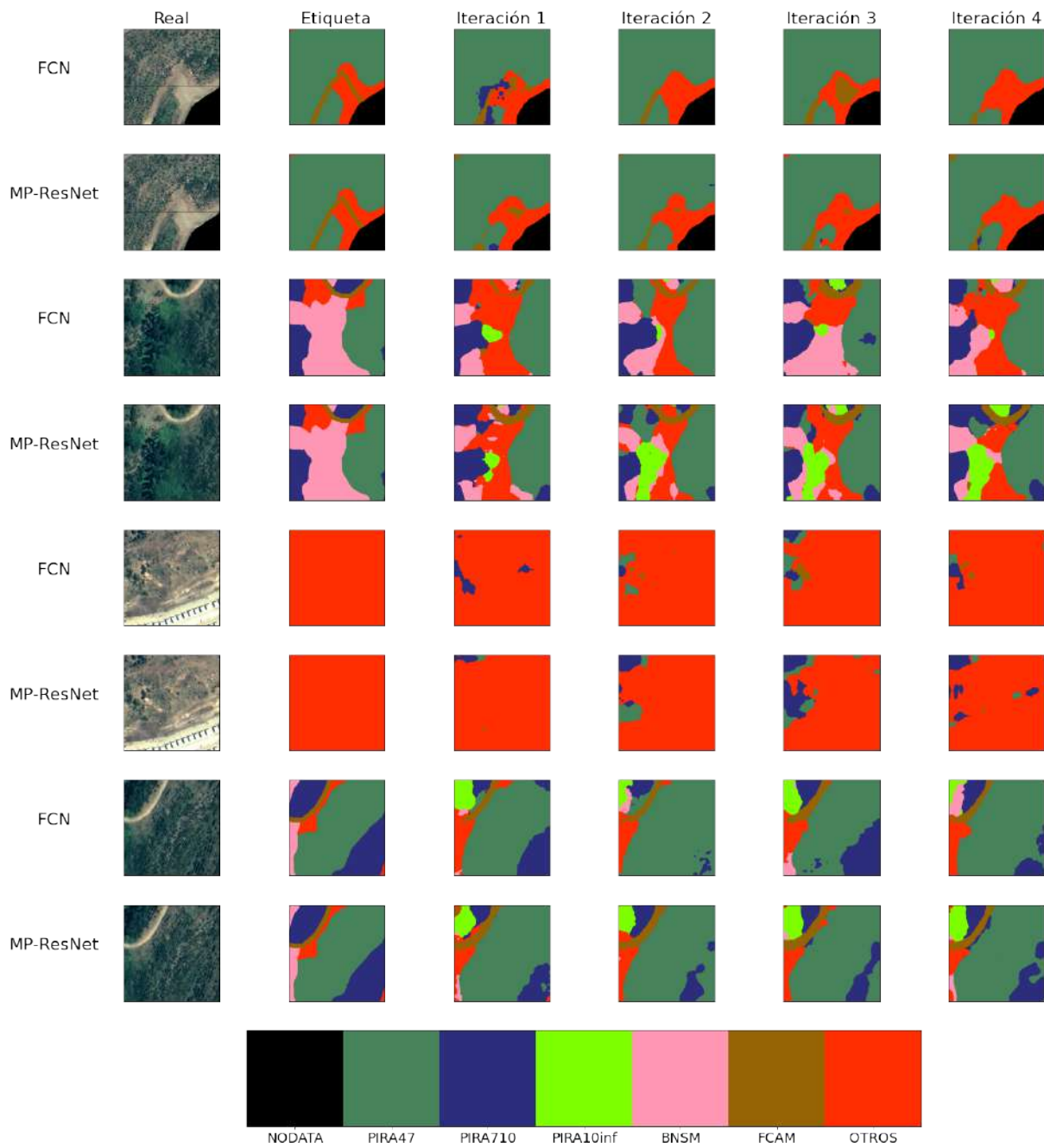


Figura A.19: Predicciones: Predio 2.

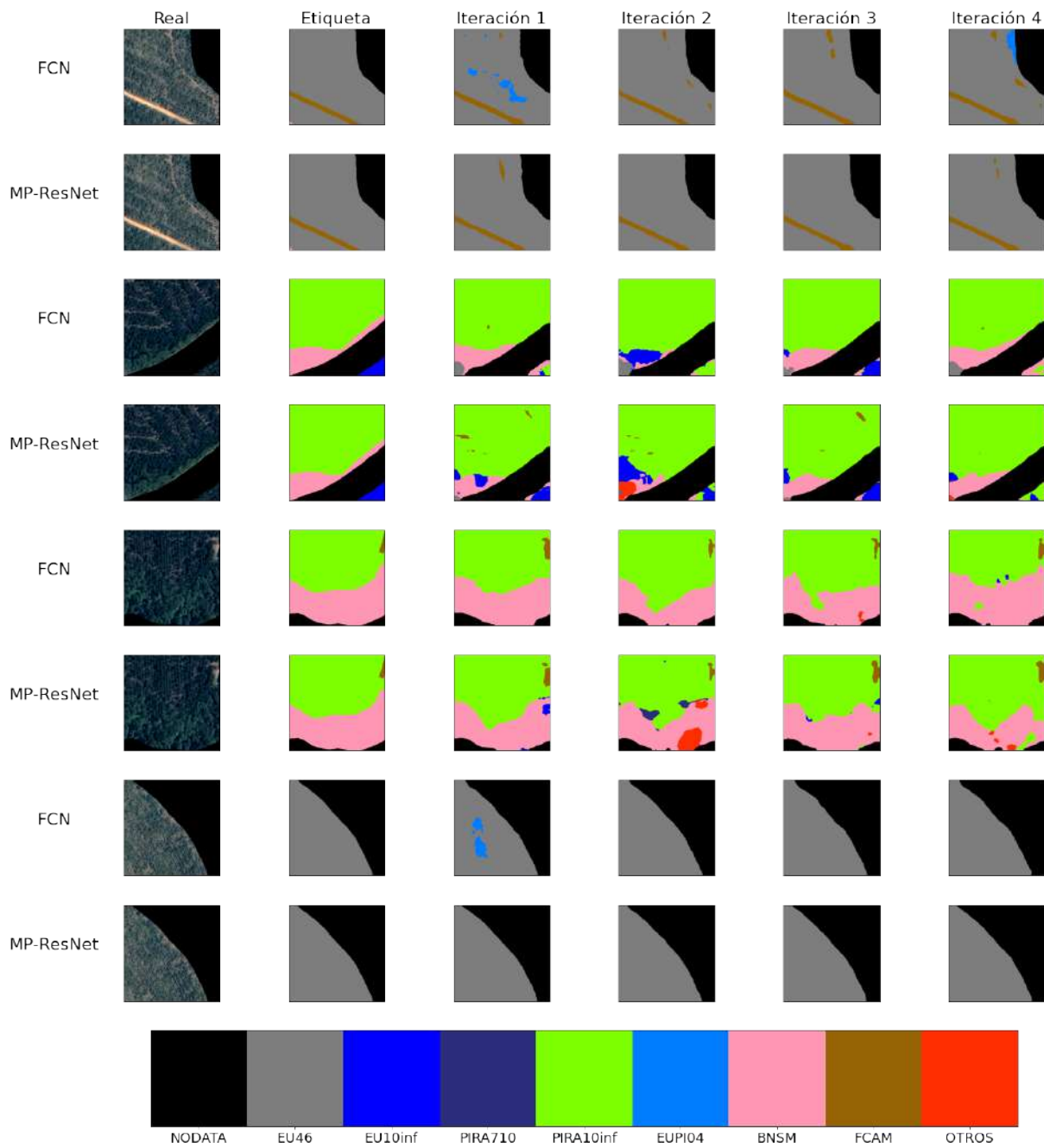


Figura A.20: Predicciones: Predio 3.

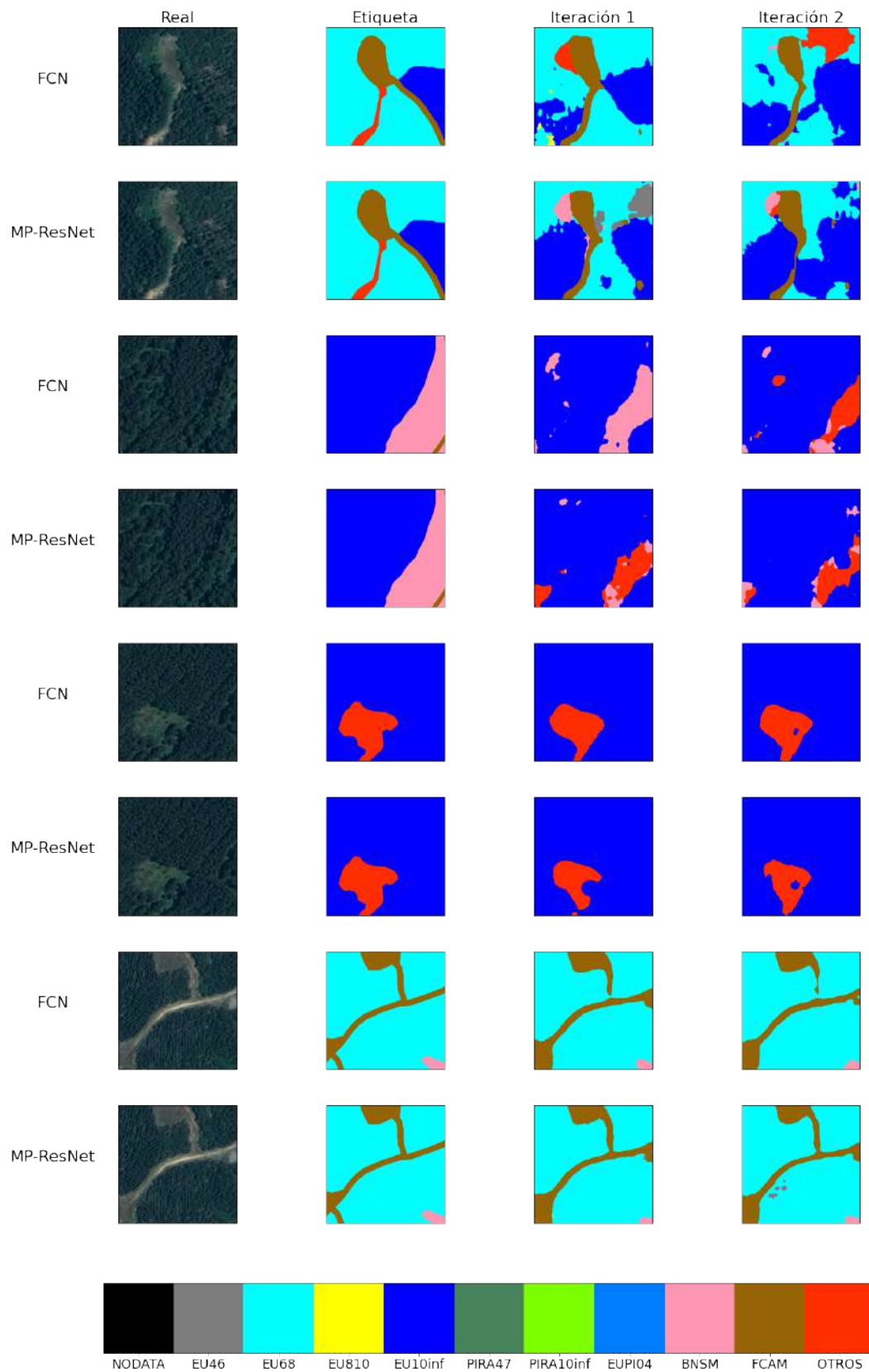


Figura A.21: Predicciones: Predio 4.

## A.6. Resumen FI

### UNIVERSIDAD DE CONCEPCION – FACULTAD DE INGENIERIA RESUMEN DE MEMORIA DE TITULO

**Departamento:** Departamento de Ingeniería Industrial

**Carrera:** Ingeniería Civil Industrial

**Nombre del memorista:** Ricardo Antonio Hernández Alarcón

**Título de la memoria:** Comparación de modelos de aprendizaje profundo para la clasificación y segmentación de áreas de interés forestal

**Fecha de la presentación oral:**

**Profesor(es) Guía:** Guillermo Cabrera Vives, Ph.D.

**Profesor(es) Revisor(es):**

**Concepto:**

**Calificación:**

**Resumen:**

La presente memoria de título se realiza en torno a la necesidad latente de la industria forestal de monitorear superficies de cultivo para la toma de decisiones corporativas respecto a su uso. Esta propone implementar, comparar y evaluar dos modelos de aprendizaje profundo para la tarea de clasificación y segmentación de distintos tipos de áreas de interés forestal, como cultivos de pino y eucaliptos de distintas edades, bosques nativos y caminos. Tarea que habitualmente es realizada de manera manual por la industria forestal. Los modelos considerados son Fully Convolutional Network y la Multi-path Residual Network y trabajan a partir de imágenes aéreas georeferenciadas de diferentes predios previamente etiquetados. Los resultados muestran que ambos modelos tienen comportamientos similares en base a las métricas utilizadas. Además, se evidencian rendimientos deficientes sobre algunas categorías en predios específicos, principalmente, en presencia de una gran similitud natural entre categorías y/o un desbalance de categorías desfavorable.

