





Universidad Complutense de Madrid (UCM)
Universidad Nacional de Educación a Distancia (UNED)

Máster en Ingeniería de Sistemas y de Control
Curso 2020 – 2021, Trabajo Fin de Máster

# Análisis de técnicas de segmentación semántica en imágenes aéreas mediante aprendizaje profundo

Esther Gascó Marco
Director: Gonzalo Pajares Martinsanz
Junio 2021



Esta obra está sujeta a una licencia de

Reconocimiento-NoComercial-

SinObraDerivada 3.0 España de Creative Commons

*La elaboración de este trabajo no hubiese sido posible sin la inestimable ayuda de mi director, el profesor Pajares, quien me guio durante todo el proceso y mostró una gran disponibilidad ante cualquier duda.*

## Resumen

La visión por computador es el campo de la ciencia de la computación que surge para hacer frente a la necesidad de dotar de la capacidad visual a las computadoras con el fin de comprender el contenido de las imágenes o vídeos digitales, pero la complejidad de la percepción en un mundo físico, dinámico e infinitamente variable dificulta esta tarea. Para alcanzar este propósito es necesario el desarrollo de modelos especializados de aprendizaje profundo, redes neuronales y algoritmos que en la actualidad siguen siendo un gran desafío. En este punto es donde se centra el presente trabajo el cual trata la percepción del contenido de la imagen a nivel de píxel, conocido computacionalmente como segmentación semántica. Se abordan una serie de experimentos a partir de imágenes aéreas donde se asigna una categoría de cobertura terrestre a cada píxel resultando en una tarea compleja debido a las grandes diferencias en la apariencia del entorno terrestre. Con el fin de acometer esta problemática, se comparan analíticamente los distintos modelos de segmentación, basados en aprendizaje profundo, más relevantes en la actualidad: DeepLabv3, FCN, SegNet y UNet. Para cada uno de tales métodos se estudia su comportamiento tanto desde el punto de vista de la fase de entrenamiento como de clasificación. Así, en el primer caso, se analizan comportamientos según distintos parámetros de aprendizaje, mientras que en el segundo se estudia su desempeño en función de los resultados del aprendizaje.

**Palabras clave:** visión artificial, redes neuronales convolucionales, segmentación semántica, imágenes aéreas.

## Abstract

Computer vision is the field of computer science that addresses the need to provide computers with visual capacity to understand the content of digital images or videos. However, the complexity of perception in a physical, dynamic, and infinitely variable world makes this task difficult. To achieve this purpose, the development of specialized deep learning models, neural networks and algorithms is necessary, even if it is still a great challenge nowadays. This study focuses on that development and deals with the perception of image content at pixel level, computationally known as semantic segmentation. A series of experiments are approached from aerial images where a category of land cover is assigned to each pixel, resulting in a complex task due to the great differences in the appearance of the terrestrial environment. To tackle this problem, the most relevant different segmentation models based on deep learning are analytically compared: DeepLabv3, FCN, SegNet and UNet. For each of these methods, the behavior is studied from both the point of view of the training phase and the phase of classification. Thus, in the first case, behaviors are analyzed according to different learning parameters, while in the second, their performance is studied based on the learning outcomes.

**Keywords:** computer vision, convolutional neural networks, semantic segmentation, aerial images.

# Índice

1 Aproximación a la visión por computador bajo aprendizaje profundo ..... 1
1.1 Estudio evolutivo de las redes neuronales artificiales ..... 2
1.1.1 Parametrización de los modelos ..... 3
1.1.2 Generalización de la Inteligencia Artificial ..... 4
1.1.2 El futuro de la Inteligencia Artificial ..... 6
1.2 Inteligencia artificial y percepción humana ..... 7
1.2.1 Detección de contornos cerrados ..... 8
1.2.2 Razonamiento sobre las imágenes ..... 9
1.2.3 Alcance de la visión por computador ..... 9
1.3 El paradigma de los algoritmos en la visión por computador ..... 10
1.3.1 Física intuitiva ..... 12
1.3.2 Causalidad ..... 13
1.3.3 Funcionalidad ..... 14
1.3.4 Intención ..... 15
1.3.5 Utilidad ..... 16
1.4 Evolución de los modelos de segmentación semántica basados en ..... 17
1.5 Objetivos y metodología ..... 19
1.6 Organización del trabajo ..... 20
2 La segmentación semántica en las CNN ..... 21
2.1 Segmentación semántica basada en FCN ..... 22
2.1.1 Conceptos ..... 25
2.2 UNet ..... 31
2.2.1 Arquitectura ..... 32
2.2.2 Cálculo de pérdidas en UNet ..... 33
2.3 DeepLab ..... 34
2.3.1 DeepLabv3 ..... 34
2.4 SegNet ..... 37
2.4.1 Arquitectura ..... 38

3 La segmentación semántica como mecanismo de prevención de incendios forestales ..... 41
3.1 Materiales y Recursos utilizados ..... 42
3.1.1 Descripción de las imágenes ..... 42
3.1.2 Recursos Hardware/Software ..... 43
3.2 Entrenamiento ..... 45
3.2.1 DeepLabv3 ..... 51
3.2.2 SegNet ..... 57
3.2.3 UNet ..... 60
3.2.4 FCNNet ..... 62
3.3 Clasificación ..... 64
3.3.1 DeepLabv3 ..... 65
3.3.2 SegNet ..... 74
3.3.3 UNet ..... 80
3.3.4 FCNNet ..... 82
4 Análisis comparativo de los modelos de segmentación semántica: conclusiones y trabajo futuro ..... 86
4.1 Comparativa de los métodos de segmentación semántica para la clasificación del entorno de Abadín ..... 87
4.2 Estudios en el campo de la segmentación semántica para el tratamiento de imágenes aéreas ..... 89
4.2.1 Segmentación semántica de imágenes aéreas con un conjunto de CNN ..... 89
4.2.2 Segmentación semántica de imágenes aéreas con redes neuronales convolucionales barajadas ..... 90
4.2.3 Aprendizaje profundo de múltiples tareas para la segmentación semántica de imágenes aéreas regularizadas geográficamente ..... 91
4.2.4 Red híbrida de atención múltiple para la segmentación semántica en imágenes aéreas ..... 91
4.2.5 Mejora de la segmentación semántica de imágenes aéreas con neuronas inhibidoras ..... 92
4.3 Conclusiones ..... 93
4.4 Trabajo futuro ..... 94
Bibliografía ..... 95

# Listado de figuras

Figura 1.1: Representación gráfica de extrapolación e interpolación. ..... 6
Figura 1.2: Ejemplo de dos problemas de los 23 de la prueba SVRT. ..... 9
Figura 1.3: Probabilidad de BagNet-33 de clase correcta para muestras con distinto tamaño y resolución decreciente. ..... 10
Figura 2.1: Arquitectura FCN: Submuestreo y sobremuestreo ..... 25
Figura 2.2: Representación de la función ReLU ..... 26
Figura 2.3: Capa de convolución ..... 29
Figura 2.4: Capa de convolución y agrupación ..... 29
Figura 2.5: Desagrupación “el vecino más próximo” ..... 30
Figura 2.6: Desagrupación “cama de clavos” ..... 30
Figura 2.7: Agrupación y desagrupación máxima. ..... 30
Figura 2.8: Representación y ejemplo de la arquitectura de UNet. ..... 33
Figura 2.9: Arquitecturas alternativas para capturar contexto multiescala. ..... 35
Figura 2.10: Convolución atrous ..... 36
Figura 2.11: SegNet. ..... 38
Figura 2.12: Ejemplo de proceso de decodificación en SegNet. ..... 39
Figura 3.1: Fotografía aérea de la región de Abadín ..... 43
Figura 3.2: Image Labeler ..... 44
Figura 3.3: Etiquetado de las fotografías ..... 50
Figura 3.4: Entrenamiento desfavorable en DeepLabv3 ..... 55
Figura 3.5: Entrenamiento favorable en DeepLabv3 ..... 56
Figura 3.6: Resultado desfavorable con L2Regularization igual a 0.1 ..... 57
Figura 3.7: Resultado favorable con L2Regularization igual a 0.005 ..... 58
Figura 3.8: Obtención de NaN durante el entrenamiento ..... 60
Figura 3.9: Resultado desfavorable obtenido con UNet ..... 60
Figura 3.10: Resultado desfavorable con FCNNet ..... 62
Figura 3.11: Clasificación favorable de imágenes con ResNet18 ..... 71
Figura 3.12: Clasificación favorable de imágenes con InceptionResnetv2 ..... 74
Figura 3.13: Clasificación favorable de imágenes con SegNet ..... 76

Figura 3.14: Clasificación desfavorable de imágenes con SegNet..... 78
Figura 3.15: Clasificación desfavorable de imágenes con UNet ..... 80
Figura 3.16: Clasificación desfavorable de imágenes con FCN-8s ..... 82
Figura 3.17: Clasificación desfavorable de imágenes con FCN-32s ..... 83
Figura 4.1: Comparativa de los resultados de clasificación..... 88

# Listado de tablas

Tabla 3.1: Resultados del entrenamiento con DeepLabv3. ..... 54
Tabla 3.2: Mejores resultados del entrenamiento con DeepLabv3..... 56
Tabla 3.3: Mejores resultados del entrenamiento con SegNet. ..... 58
Tabla 3.4: Resultados del entrenamiento con SegNet ..... 59
Tabla 3.5: Resultados del entrenamiento con UNet ..... 61
Tabla 3.6: Resultados del entrenamiento con FCNNet ..... 63
Tabla 3.7: Métricas de evaluación IoU para DeepLabv3 ..... 68
Tabla 3.8: Parámetros favorables empleados para la clasificación con ResNet18..... 69
Tabla 3.9: Métricas IoU obtenidas para el ejemplo ResNet18..... 69
Tabla 3.10: Matriz de confusión con SegNet18 ..... 71
Tabla 3.11: Parámetros favorables empleados para la clasificación con InceptionResNetv2 ... 72
Tabla 3.12: Métricas IoU obtenidas para el ejemplo InceptionResNetv2. ..... 72
Tabla 3.13: Matriz de confusión con InceptionResnetV2..... 74
Tabla 3.14: Parámetros favorables empleados para la clasificación con SegNet..... 75
Tabla 3.15: Métricas IoU obtenidas para el ejemplo SegNet..... 75
Tabla 3.16: Matriz de confusión con SegNet. ..... 77
Tabla 3.17: Parámetros desfavorables empleados para la clasificación con SegNet. ..... 77
Tabla 3.18: Métricas IoU obtenidas para el ejemplo SegNet..... 77
Tabla 3.19: Métricas de evaluación IoU para SegNet ..... 79
Tabla 3.20: Parámetros desfavorables empleados para la clasificación con UNet..... 80
Tabla 3.21: Métricas IoU obtenidas para el ejemplo UNet ..... 80
Tabla 3.22: Métricas de evaluación IoU para UNet ..... 81
Tabla 3.23: Parámetros desfavorables empleados para la clasificación con FCN-8s..... 82
Tabla 3.24: Métricas IoU obtenidas para el ejemplo FCN-8s ..... 82
Tabla 3.25: Parámetros desfavorables empleados para la clasificación con FCN-32s..... 83
Tabla 3.26: Métricas IoU obtenidas para el ejemplo FCN-32s ..... 83
Tabla 3.27: Métricas de evaluación IoU para FCNNet ..... 84

# CAPÍTULO 1

## Aproximación a la visión por computador bajo aprendizaje profundo

Cuando nos encontramos enfrente de una imagen, lo primero que hacemos es distinguir cada uno de los elementos que la forman, de manera inconsciente obtenemos un listado de palabras relacionadas con esta. Si se nos pidiera que describiéramos de forma general dicha imagen, probablemente elegiríamos un término que la represente sin pensarlo dos veces. Ambas tareas descritas pueden resultar cuestiones muy triviales que cualquier persona con una mínima madurez intelectual podría llevar a cabo, sin embargo, detrás de ello se encuentra un proceso muy complejo: la visión humana.

Esta engloba toda una red de tecnología orgánica capaz de adquirir y procesar las diferentes imágenes captadas por los ojos, quienes son los encargados de difundir la información más importante a través del nervio óptico al cerebro, tal como formas, contrastes, detección de bordes, colores relativos, etc. El cerebro es el encargado de combinar y encajar las visiones de ambos ojos, formando una única imagen en tres dimensiones. Su interpretación se obtiene gracias a los modelos mentales de objetos que hemos ido formando según la comprensión abstracta de los conceptos y las experiencias personales vividas mediante millones de interacciones con el mundo a lo largo de nuestras vidas.

La visión prima en el proceso de intercambio de información entre el organismo y el medio ambiente siendo su tarea fundamental la de reflejar adecuadamente el mundo exterior con el fin

de identificar la situación dada para tomar una decisión sobre su comportamiento posterior y, de esta manera, entender el mundo.

La visión por computador es el campo de la ciencia de la computación que surge para hacer frente a la necesidad de dotar de la capacidad de "ver" a las computadoras con el fin de comprender el contenido de las imágenes digitales. En sus inicios se encontraron dificultades en la simulación de la percepción apartándose de esta exégesis hacia finales de la década de 1960. A partir de entonces la percepción ha sido tratada durante mucho tiempo como un problema de interpretación de imágenes estáticas bidimensionales, las cuales se presentan en forma de matriz de entrada en configuración de píxeles o valores numéricos que representan cada uno de los colores siendo estos elementos sensibles a la luz. Hoy en día, la visión artificial ha vuelto a tratar cuestiones de comprensión y no de las imágenes en sí mismas, sino del mundo que reflejan con el objetivo de modelar el entorno desde el proceso de comprensión de la realidad.

La complejidad de la percepción de la visión en un mundo físico dinámico e infinitamente variable dificulta esta tarea. Para alcanzar este fin es necesario el desarrollo de modelos especializados de aprendizaje profundo, redes neuronales y algoritmos que en la actualidad siguen siendo un gran desafío.

En este punto es donde se centra el presente trabajo el cual tratará la percepción del contenido de la imagen a nivel de píxel, conocido computacionalmente como segmentación semántica. Las redes neuronales artificiales (Artificial Neural Networks, ANN), conocidas como redes neuronales convolucionales (Convolutional Neural Networks, CNN) dentro del paradigma del Aprendizaje Profundo, serán la herramienta que se empleará para tal fin.

En lo que resta de este capítulo se realiza una breve revisión de los aspectos expresados previamente.

## 1.1 Estudio evolutivo de las redes neuronales artificiales

La comparación entre las ANN, que se emplean en el aprendizaje profundo, y las redes neuronales biológicas (Biological Neural Network, BNN), propias del cerebro humano, son motivo de discusión recurrente en el campo de la inteligencia artificial. Muchos científicos coinciden en que el funcionamiento de las ANN son una imitación aproximada a la estructura del cerebro, pero otros se basan en la idea de que en realidad son motores de inferencia estadística cuya estructura no refleja muchas de las funciones del cerebro ya que este va más allá de la pura conexión neurológica.

De la misma forma que la investigación en neurociencia y psicología avanza, los modelos de redes neuronales artificiales alcanzan un elevado rendimiento en muchas tareas con similitud a

la conducta humana, por ejemplo, encontramos trabajos de investigación recientes sobre el procesamiento del lenguaje (Radford et al., 2019), la jugabilidad compleja (Jaderberg et al., 2019) y reconocimiento facial (Taigman et al., 2014), entre otros. Estos modelos optimizan millones de parámetros sobre millones de ejemplos, sin embargo, han evolucionado en el modelado de procesos cognitivos, desde una interpretación de conexión y procesamiento paralelo distribuido a una visión más moderna que precisa de modelos interpretables de escala relativamente pequeña con entradas bien controladas (McClelland y Rogers, 2003).

Del mismo modo que las BNN, las artificiales están capacitadas para realizar acciones a través de datos multidimensionales obtenidos que se encuentran en los distintos entornos de la vida real. Aunque los dos tipos de redes pueden diferenciarse respecto a su arquitectura, reglas de aprendizaje y funciones objetivas (Richards et al., 2019), ambas emplean un proceso de optimización repetitivo para obtener un propósito final dada una entrada.

### 1.1.1 Parametrización de los modelos

A través de los modelos neurocientíficos es posible obtener tanto la interpretabilidad de estos, relacionada con la descripción explícita de las causas implícitas, como la generalización, cuyo significado hace referencia a la capacidad de predicción del modelo en entornos nuevos (Neumann, 1955). En la práctica, ambos conceptos presentan discrepancias puesto que, por un lado, los modelos interpretables muestran escasa predicción y, por otro lado, los modelos predictivos o de generalización son difíciles de interpretar (Breiman, 2001; Shmueli, 2010; Yarkoni y Westfall, 2017). Esta cuestión también destaca en el modelado del cerebro humano con el fin de orquestar el comportamiento. Se considera que el cerebro contiene una capacidad de 300 millones de parámetros adaptables en cada milímetro cúbico de corteza, debido al número de neuronas que posee las cuales son capaces de sostener gran cantidad de sinapsis ajustables (Azevedo et al., 2009; Kandel et al, 2013). Este órgano sobreparametrizado es la respuesta evolutiva al comportamiento maleable en un mundo arduo. Esta complejidad se pretende reducir experimentalmente en situaciones controladas para examinar el comportamiento del cerebro de tal forma que sea posible aumentar la interpretabilidad de los distintos procesos neuronales observados.

La disponibilidad de grandes recursos y los avances en hardware de procesamiento, ha facilitado el desarrollo de redes neuronales de gran tamaño en plazos apropiados para llevar a cabo tareas complejas como son el procesamiento del lenguaje natural y la visión por computador. Los modelos de aprendizaje profundo hacen uso del concepto de sobreparametrización mediante el entrenamiento por grandes cantidades de datos. De este modo, los modelos así entrenados dominan frente a los modelos simples, que se basan en el rendimiento conductual a nivel humano en diversas tareas. Existe un amplio debate respecto a este tema en el que se cuestiona si la

sobreparametrización en exceso dentro del aprendizaje automático proporciona información sobre el código neuronal implícito de los organismos biológicos (Marcus, 2018).

El concepto de "fuerza bruta[1]", empleado comúnmente en informática, puede encajar en este contexto, considerando que las redes neuronales artificiales se fundamentan en su ajuste puesto que "la computación neuronal se basa en algoritmos de optimización parametrizados en exceso para aumentar el poder predictivo (generalización) sin modelar explícitamente la estructura generativa subyacente del mundo" (Hasson et al., 2019). El principal problema en este tipo de redes surge en la opacidad de las mismas, es decir, cuando una red se torna más grande, con una distribución lógica en millones de parámetros, su interpretación resulta más complicada en comparación a un modelo simple que asigna un único coeficiente a cada una de las características. Es posible facilitar esta interpretación, por ejemplo, reduciendo las variables o el número de capas y con ello las neuronas, pero esto conlleva a que este tipo de modelos más simples estén menos capacitados para tratar con los datos complejos y desordenados que son los que aparecen en la naturaleza.

### 1.1.2 Generalización de la Inteligencia Artificial

Uno de los principales objetivos de la Inteligencia Artificial (Artificial Intelligence, AI) es el de encontrar un modelo con capacidad de extrapolación, esto es, que frente a los datos esté preparado para extraer las reglas subyacentes fuera del entrenamiento. Por ello es necesario encontrar un modelo con un ajuste ideal capaz de entender la estructura generativa o global latente en los datos, es decir, generalizar, permitiendo que nuevas observaciones que no se han visto durante el entramiento puedan ser predichas con precisión. Cuando el problema que se presenta es simple es posible ajustar una red neuronal profunda con un pequeño grupo de datos de entrenamiento, de este modo sería posible alcanzar el nivel de extrapolación deseado siendo útil en entornos con características limitadas, pero cuando se trabaja con datos no estructurados y desordenados, los enfoques de datos pequeños tienden a fallar, por ejemplo, ante una imagen donde cada píxel se convierte en una variable, por lo que el análisis de un grupo de imágenes de 100x100 píxeles pasa a ser un problema con 10.000 dimensiones, cada una con miles de posibilidades. En este nivel dimensional es donde se centra el presente trabajo, puesto que la segmentación semántica se aborda a nivel de píxel.

Si la cuestión a tratar presenta un muestreo denso del espacio, la predicción de este pasa a ser un problema basado en interpolación en lugar de uno basado en la extrapolación. "La interpolación utiliza cálculos locales para situar las observaciones nuevas dentro del contexto de las observaciones pasadas, no se basa en un modelo explícito de los principios generativos generales. A diferencia de la extrapolación, se pensaba que la interpolación proporcionaba una

1 Método de prueba y error.

forma débil de generalización porque solo puede predecir nuevos puntos de datos dentro del contexto de observaciones pasadas. Por lo tanto, cuando consideramos el cerebro, tradicionalmente hemos asumido que la interpolación no proporciona una forma de generalización suficiente para respaldar un comportamiento complejo, ya que la tarea del cerebro es extrapolar de un pequeño número de ejemplos a un rango casi infinito de posibles observaciones" (Hasson et al., 2019). El problema comentado aparece cuando el alcance del espacio de entrenamiento es escaso, si bien, es posible replantearlo gracias al cambio a *big data* cuyo muestreo se lleva a cabo por millones de parámetros que pueden ajustarse haciendo posible aplicar el enfoque de ajuste directo a dominios complejos que no se pueden representar con pocas muestras y reglas generales. Se conoce como ajuste directo al modelo que, dentro de las redes neuronales artificiales, se basa en la sobreparametrización para que esta corresponda con la estructura de datos, además, regulariza el proceso con el propósito de evitar el sobreajuste al tiempo que optimiza la alineación con la arquitectura de los datos de entrenamiento.

El teorema de "no almuerzo gratis[2]", empleado en la complejidad computacional y optimización, demuestra que la optimización de una tarea necesariamente deteriorará el rendimiento en otra (Wolpert y Macready, 1997). Este teorema afecta tanto a las redes neuronales artificiales como a las biológicas, aunque los modelos computacionales cada vez son más complejos gracias a su entrenamiento en gran cantidad de datos por lo que podrían aproximarse, en cierto modo, a la robustez del cerebro humano respecto a estructuras distribuidas temporales y espaciales. Estos modelos tratan de aprender soluciones ad hoc para un área concreta del espacio del problema (Gomez-Marin y Ghazanfar, 2019). Es evidente que afectará a la interpolación del modelo. Del mismo modo, la evolución de las especies produce organismos optimizados para entornos específicos, pero no encuentra una solución concreta para que ese mismo organismo pueda sobrevivir en cualquier parte.

Cuando la zona de interpolación se amplía, la insuficiencia del modelo para extrapolar pasa a ser una desventaja menor (Feldman, 2019). Los modelos sobreparametrizados poseen una robusta capacidad de ajuste a estructuras de datos complejas con una gran condición predictiva en la zona de interpolación, lo que hace que estos sean singularmente apropiados para escenarios multidimensionales del mundo real donde no existe un modelo ideal simple. De esta forma, es posible determinar que "cuando la estructura de datos es compleja y multidimensional, un modelo de ajuste directo "sin sentido", capaz de realizar predicciones basadas en interpolación dentro de un espacio de parámetros del mundo real, es preferible a un modelo explícito tradicional de ajuste ideal que no explica mucha variación en los datos" (Hasson et al., 2019). En la figura 1.1 se muestra un esquema gráfico relativo a los conceptos de extrapolación (izquierda) en el cual se intentan extraer reglas de *big data* para su posterior aplicación a todo el

2 "Establece que, para ciertos tipos de problemas matemáticos, el coste computacional de encontrar una solución, promediado sobre todos los problemas de la clase, es el mismo para cualquier método de solución" (Wikipedia, 2020).

espacio de problemas y, la interpolación (derecha) basada en un gran muestreo del espacio del problema para calcular los espacios entre muestras.

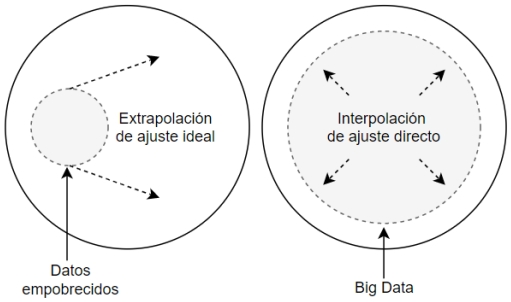



Figura1.1: Representación gráfica de extrapolación e interpolación.

Este enfoque se contradice con el problema de "cola larga[3]", en su perspectiva de la AI. Un ejemplo de ello se observa en la clasificación de imágenes ya que, mediante el uso de bases de datos de entrenamiento populares, como lo es ImageNet (Deng et al., 2009; ImageNet LSVRC, 2014), se obtienen una gran cantidad de imágenes de objetos, pero se presentan en condiciones ideales de iluminación y ángulos convencionales. Esto provoca que las redes neuronales entrenadas en estas bases de datos puedan fallar en la identificación si los objetos se encuentran en alguna posición inusual. El concepto de "cola larga" hace referencia pues, a ejemplos poco comunes o de baja frecuencia, como podría ser el de una silla con forma de un objeto que no tiene relación directa con ella. Si enfocamos esta situación a una red neuronal biológica, se observa que una persona frente a una situación que nunca ha experimentado no es capaz de extrapolar la información necesaria, como sería el caso de la conducción por primera vez por la nieve donde la mayoría de las personas fallan.

La psicofísica visual humana clásica resuelve muchas rarezas de manera trivial simplemente permitiendo que un observador se desplace y muestree el escenario, es decir, la manifestación de "cola larga" se considera en cierta medida una deficiencia de muestreo. La solución no consiste en obtener más muestras, sino que necesitará de un muestreo más avanzado para abordar el problema, tal y como se muestra en los organismos biológicos, por ejemplo, en la búsqueda de novedades.

### 1.1.3 El futuro de la Inteligencia Artificial

En muchas ocasiones se subestima la enorme cantidad de datos que procesa el cerebro. El ser humano está expuesto diariamente a millones de ejemplos visuales que de manera inconsciente va categorizando, de esta forma es posible extraer reglas del entorno con el fin de desarrollar pensamientos y conceptos abstractos que se emplean para el proceso y análisis de nueva información. Esta forma de extrapolar contextos excede a las redes neuronales artificiales

3 Problema difícil de resolver para un caso concreto puesto que hay una larga cola de variaciones (casos extremos) en las cosas más mundanas.

modernas, comprender y aplicar la causalidad sigue siendo una de las características únicas del cerebro humano.

Es posible englobar el pensamiento humano en dos tipos de sistemas, el primero engloba las tareas que se pueden aprender, es decir, la posibilidad de realizar estas tareas de manera subconsciente, el segundo requiere concentración y pensamiento consciente el cual se obtiene mediante una comprensión explica de la estructura subyacente que posee el mundo. De esta idea parten multitud de proyectos de investigación en el campo de la AI, si bien, muchos piensan que, mediante los sistemas puros, basados en redes neuronales, ocasionalmente será posible alcanzar un nivel de AI igual al segundo sistema, no obstante, recientes estudios muestran que las estructuras de redes neuronales avanzadas presentan capacidades de manipulación de símbolos que se creían fuera de los límites del Aprendizaje Profundo. Surge así una visión alternativa que se basa en la creación de sistemas híbridos que incorporan a las redes neuronales, inteligencia artificial simbólica, es decir, basada en reglas. Estos sistemas se complementan fortaleciéndose mediante la suma de sus partes.

## 1.2 Inteligencia artificial y percepción humana

En los últimos años el aprendizaje automático ha sufrido un notable avance provocado por el impulso de algoritmos de AI en diversas materias, tales como detección de objetos, reconocimiento facial, respuesta a preguntas, etc. Aunque se trata de aplicaciones prácticas para determinados sectores que optimizan algunos procesos, solamente se tiene en cuenta el resultado final de la prueba de estos algoritmos en un conjunto limitado de datos lo que puede conllevar un resultado no deseado, pudiendo hacer peligrar los procesos asociados ante situaciones críticas.

En la búsqueda para la reconstrucción de la percepción humana, el campo de la visión por computador, gracias al aprendizaje profundo y mediante las ya mencionadas CNN, ha dado resultados muy favorables. Se obtienen grandes avances en tareas que, mediante el software tradicional, eran muy complicadas de afrontar, sin embargo, comparar estas redes con el entendimiento humano sigue siendo un reto debido a que aún queda mucho por conocer sobre el sistema de visión humano general, además, el complejo funcionamiento de los sistemas de aprendizaje profundo dificulta esta cuestión.

Los métodos actuales de comparación de redes neuronales profundas y el sistema de visión humana son caso de estudio, una gran oportunidad para aprender sobre un sistema mediante el análisis del otro con el objeto de evaluar situaciones del mundo real. Recientemente, investigadores de distintas organizaciones y universidades han llevado a cabo este desafío donde se destaca la problemática en los métodos actuales que se emplean para este fin. Se

proponen tres casos de estudio: detección de contornos cerrados, razonamiento sobre las imágenes y la diferencia de reconocimiento (Funke et al., 2020).

La metodología empleada para alcanzar este objetivo se fundamenta en una base de verificaciones. En primer lugar, es necesario aislar los problemas funcionales o de implementación, esto se debe a que es posible que los sistemas identifican más de un motivo de diferencia, pero lo que realmente se pretende conseguir es que este se centre en una sola razón para una diferencia observada, de este modo se diseña una red artificial restringida que hace hincapié en el aspecto de interés, aunque en muchos casos es prácticamente imposible. Brendel y Bethge (2019) trataron este planteamiento mediante la reducción del tamaño de los campos perceptivos para un procesamiento local puro. Por otro lado, es importante destacar la necesidad de alinear las condiciones experimentales en los sistemas, es decir, realizar un estudio comparativo entre humanos y máquinas u otros organismos que hagan una diferenciación lo más explícita posible. Estas características podrían ser elegidas en base a la descripción del comportamiento de interés como el que sucede en un determinado tiempo tras el inicio del estímulo (DiCarlo et al., 2012).

Cabe destacar que la interpretación creada por el ser humano ante una situación, en ocasiones, tiende a antropomorfizarse, como es conocido en la psicología comparada (Tomasello y Call, 2008; Boesch, 2007; Haun et al., 2010; Koehler, 1943; Romanes, 1883). Este fenómeno puede influir directamente en cualquier experimento, por ejemplo, Dujmovi et al. (2020) explican que la búsqueda de similitudes o diferencias entre humanos y máquinas puede verse afectada por la selección de los estímulos y etiquetas.

### 1.2.1 Detección de contornos cerrados

Una comparación entre humano y máquina puede representarse con la tarea de detección de contornos cerrados. Se pretende obtener un resultado frente al aprendizaje de los algoritmos en cuestiones de formas cerradas y abiertas en distintas condiciones. Esta identificación resulta evidente para cualquier persona donde la percepción de un contorno cerrado flanqueado por muchos abiertos destaca mientras que la detección por una Red Neuronal Profunda (Deep Neural Network, DNN) puede ser complicada debido a que necesita de una integración de contorno de largo alcance.

El estudio planteado por diversos investigadores propone el uso de la red neuronal convolucional ResNet-50 (He et al., 2016). Se utiliza el aprendizaje por transferencia para ajustar este modelo empleando 14.000 imágenes con contornos cerrados y abiertos. Tras verificar su eficiencia en varios ejemplos se obtuvieron resultados destacables por lo que una red neuronal bien entrenada puede asimilar el concepto de contorno cerrado, pero, si se somete a la variación en color, grosor de las líneas o tamaño, el modelo de aprendizaje profundo presenta una caída en su precisión

cosa que no ocurre en la misma medida en la percepción humana. Se demostró que, frente a perturbaciones adversas apenas perceptibles por la visión humana, se interrumpe el comportamiento de los sistemas de aprendizaje.

Con el fin de profundizar más en el proceso de toma de decisiones de los algoritmos se empleó el modelo *Bag-of-Feature*, técnica que trata de localizar los bits de datos que ayudan en la decisión de un modelo de aprendizaje demostrando que "existen características locales como un punto final junto con un borde corto que a menudo puede revelar la etiqueta de clase correcta" (Funke et al., 2020, p.9).

### 1.2.2 Razonamiento sobre las imágenes

Hay cuestiones que un observador humano resolvería con facilidad, por ejemplo, en tareas de iguales-diferentes, identificar dos formas idénticas en una imagen o en tareas espaciales, el tamaño de estas. Se emplea la Prueba de Razonamiento Visual Sintético (Synthetic Visual Reasoning Test, SVRT), figura 1.2, para analizar las habilidades de los algoritmos de aprendizaje profundo en el razonamiento visual abstracto. Mediante la red RedNet-50 se llevó a cabo este cometido entrenando el modelo con 28.000 muestras llegando a la conclusión de que su funcionamiento en ambos tipos de tareas, de iguales y espaciales, es correcto, pero a medida que esta cantidad de muestras se reduce en el entrenamiento, los resultados se degradan, sobre todo en tareas de iguales-diferentes que requieren más muestras de entrenamiento que las tareas de razonamiento espacial. En comparación con el sistema visual humano, donde la abstracción es inherente, una red neuronal requiere de gran cantidad de muestras para aproximarse a este.

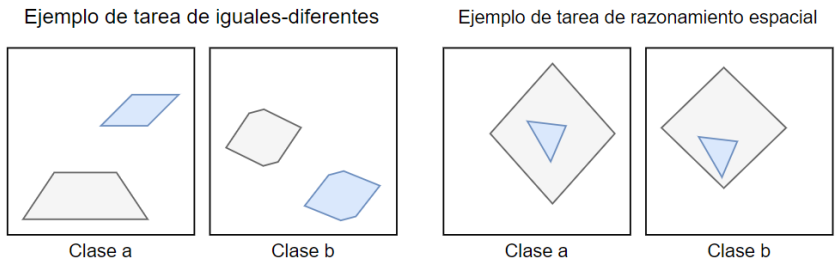



Figura 1.2: Ejemplo de dos problemas de los 23 de la prueba SVRT.

### 1.2.3 Alcance de la visión por computador

La brecha del reconocimiento visual es fundamental para entender el alcance de la visión por computador. Ullman et al., (2016) trataron este tema de estudio donde se analiza qué información mínima es necesaria para el reconocimiento de objetos. Se redujo progresivamente la resolución de una imagen natural, más del 50%, hasta que los participantes humanos no fueron capaces

de identificar el objeto. Fue posible demostrar que el rendimiento del sistema de visión por computador para el reconocimiento también disminuye, como se puede observar en la figura 1.3, aunque hay que destacar que, los sistemas de aprendizaje profundo, a pesar de operar también con las características de las imágenes, su funcionamiento es más sutil pudiendo encontrar características prácticamente imperceptibles para el ojo humano. Se llegó a la conclusión en este estudio de que los algoritmos de visión artificial no son capaces de explicar la sensibilidad humana a configuraciones precisas de características y que el sistema visual humano emplea características y procesos que no usan los modelos actuales siendo críticos para el reconocimiento (Funke et al., 2020, p.15).

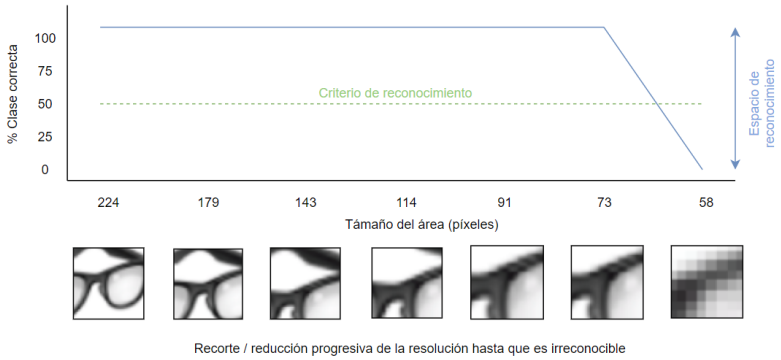



Figura 1.3: Probabilidad de BagNet-33 de clase correcta para muestras con distinto tamaño y resolución decreciente.

Conforme los sistemas de AI se vuelven más complejos es necesario el desarrollo de métodos más complicados para su comprobación. "El desafío general en los estudios de comparación entre humanos y máquinas parece ser el fuerte sesgo interno de interpretación humana". “Las herramientas de análisis adecuadas y las comprobaciones cruzadas exhaustivas, como las variaciones en la arquitectura de la red, la alineación de los procedimientos experimentales, las pruebas de generalización, los ejemplos contradictorios y las pruebas con redes restringidas, ayudan a racionalizar la interpretación de los hallazgos y ponen este sesgo interno en perspectiva. En general, se debe tener cuidado de no imponer nuestro sesgo sistemático humano al comparar la percepción humana y la de las máquinas " (Funke et al., 2020).

## 1.3 El paradigma de los algoritmos en la visión por computador

¿Qué es lo que nos hace tan buenos a los humanos para dar sentido a los datos visuales? Esta es una cuestión que ha inquietado a los científicos del campo de la inteligencia artificial y la visión por computador durante muchos años. Los esfuerzos por imitar las capacidades de la visión humana han dado resultados elogiables, aunque sigue siendo una tarea que necesita seguir madurando. Los algoritmos existentes de AI son capaces de detectar objetos en imágenes con una elevada precisión, pero solamente después de la visualización de numerosos ejemplos o de una imagen que no difiera en exceso de las vistas anteriormente. Actualmente, el sistema de

aprendizaje profundo en la visión por computador posee una estructura frágil y superficial la cual se pretende resolver, pero encontrar la solución adecuada para la corrección de los algoritmos es una tarea confusa debido a que encontrar la solución correcta consiste en formular las cuestiones apropiadas y exponer el problema de una manera acertada.

La definición clásica de visión por computador propuesta por el neurocientífico y fisiólogo británico David Marr (1985), quien integró la psicología, inteligencia artificial y la neurofisiología en modelos nuevos de procesamiento visual, es la que comprende las cuestiones "qué", propia del reconocimiento de objetos que corresponde con la visión de objetos y "dónde", que indica la reconstrucción tridimensional y la localización de objetos siendo esta la visión espacial. Ambas están relacionadas con la hipótesis de las dos corrientes en el cerebro humano, por un lado, la vía ventral, encargada de la identificación y reconocimiento visual de objetos y, por otro lado, la vía dorsal, responsable del procesamiento de la ubicación espacial del objeto, es decir, de la reconstrucción de profundidad, formas, etc. Este paradigma encauzó la perspectiva de estudio de la geometría de la visión por computador entre las décadas de 1980 y 1990 pero a lo largo de los últimos años se ha avanzado en materia de detección y localización de objetos debido al desarrollo de los modelos DNN gracias a la aparición de máquinas dotadas de un hardware potente y la posibilidad de disponer de conjuntos masivos de datos etiquetados.

Zhu et al. (2020), científicos del instituto de Tecnología de Massachusetts y la Universidad de California en Los Ángeles, publicaron recientemente un artículo donde se argumenta la posible clave para dotar a los sistemas de AI de razonamiento sobre datos visuales semejándose a la percepción humana. Esta consiste en abordar la “materia oscura” de la visión por computador, es decir, las cosas que no son visibles en los píxeles. El documento profundiza en cinco elementos fundamentales de la visión: funcionalidad, física, intención, causalidad y utilidad, que permitirán el paso desde la AI de “big data para pequeñas tareas” donde se emplean grandes cantidades de datos para el entrenamiento de un clasificador para una sola tarea limitada a “pequeños datos para grandes tareas”.

La detección de las relaciones de entidades, tanto en imágenes como en videos, es inalcanzable por la mayoría de las perspectivas existentes, las cuales contemplan solamente la apariencia donde los factores latentes no están reproducidos en píxeles, pero estos se encuentran en todas partes, gobernando tanto la ubicación como el movimiento de estas entidades visibles sencillas de detectar mediante los métodos actuales. Tales entidades son fundamentales para abordar de manera completa la visión por computador donde las tareas que se afrontan mayormente se basan en problemas de clasificación impulsados por datos a gran escala y entrenamiento de redes neuronales. Al tratar este asunto es posible aproximarse al razonamiento humano el cuál ampliaría las cuestiones planteadas inicialmente sobre la definición clásica “qué” y “dónde”, completándola con “por qué”, “cómo” y “qué pasaría si”, las cuales integran el razonamiento causal, física intuitiva y el aprendizaje funcional.

Los recientes avances en materia de inteligencia artificial se fundamentan en las redes neuronales profundas. Se trata de algoritmos de aprendizaje automático que se aproximan a las capacidades de coincidencia de patrones del cerebro cuya estructura consiste en capas de funciones matemáticas. Las DNN necesitan ser entrenadas con ejemplos, cuantas más capas tenga un DNN y más datos de calidad posea en su entrenamiento, será capaz de extraer y detectar de una manera más eficiente patrones comunes, situados en los objetos de las imágenes ya que se ajustan los pesos de sus funciones para tal fin. Muchos son los científicos que creen que, si se trata con modelos de aprendizaje profundo más grandes, en determinadas ocasiones será posible evolucionar hacia sistemas de AI próximos a la eficiencia de la visión humana, sin embargo, existe otro convencimiento basado en que una excesiva profundidad puede llegar a ser contraproducente debido a que se podrían producir sobreajustes en los parámetros del modelo, teniendo que llegar a un compromiso entre la eficiencia del modelo y profundidad del mismo.

### 1.3.1 Física intuitiva

Uno de los componentes fundamentales del sistema visual humano es la comprensión del conocimiento funcional del mundo a nivel físico. Esta comprensión física no exige necesariamente la precisión de las leyes de la mecánica de Newton, sino que es posible entender y razonar, en cierta medida, el comportamiento tanto de objetos rígidos como de las propiedades de los líquidos en una determinada situación gracias a la intuición construida con la ayuda de las interacciones con el entorno. El conocimiento implícito de estas actividades se denomina física intuitiva. Este campo se ha explorado durante muchos años en la ciencia cognitiva y actualmente se ha asociado a nuevas técnicas vinculadas a la AI.

Las relaciones físicas en las imágenes son comúnmente visibles, pero sería prácticamente imposible disponer de un modelo de física intuitiva en la visión por computador. Por ejemplo, un robot de búsqueda y rescate, ante una misión de socorro, debe ser capaz de razonar la estabilidad de varios objetos, lo que engloba la física de estos para poder identificar qué elementos soportan a otros con el fin de emplear la información adquirida para poder desplazarse por el terreno con prudencia y de este modo no provocar nuevas situaciones de peligro. Este enfoque se contrapone al actual paradigma de la AI en el cual, estructuras tales como las redes neuronales son las encargadas de recibir una sucesión de imágenes etiquetadas, es decir, con una respectiva descripción, en las que se confía que estas propiedades físicas se añadan a sus pesos.

En la actualidad existen sistemas de AI que agregan motores físicos a su funcionamiento con el fin de realizar un mejor razonamiento respecto a las relaciones existentes entre objetos en comparación con los sistemas puros de redes neuronales, tratándose de un nivel dentro de la AI de rango superior.

### 1.3.2 Causalidad

La causalidad es la principal componente faltante en los sistemas de AI. Esta noción abstracta hace referencia a la causa y efecto resultante del entorno percibido por lo que es posible emplearla como base para la generación de nociones de tiempo y espacio, es decir, posibilita no solamente el razonamiento sobre un suceso en un momento concreto, sino que plantea la cuestión de “qué pasaría si” en un escenario hipotético (Robb, 1911; Malament, 1977; Robb, 2014). La capacidad de comprensión de la causalidad es la responsable, en su mayoría, de la comprensión humana, sin esta capacidad sería imposible predecir una determinada situación con una reacción acorde. En cambio, los algoritmos de aprendizaje automático tienen la facultad de perseguir correlaciones existentes entre distintas variables, pero no son capaces de razonar la causalidad puesto que es necesaria una comprensión del mundo debido a que no siempre son visibles estas conexiones.

Hasta el momento, el aprendizaje profundo se ha basado en la adquisición de grandes cantidades de datos para compensar esta falta de razonamiento causal, se entrenan los sistemas de inteligencia artificial mediante millones de escenarios con la confianza de capturar todas las posibles combinaciones, aunque hoy en día sigue existiendo un gran debate en la literatura respecto a la necesidad o no de la compresión de la relación causal para la inteligencia artificial. Sin embargo, este aprendizaje es esencial en muchas situaciones donde el sistema de AI puede operar en entornos con una variación dinámica común. Un claro ejemplo se encuentra en el aprendizaje por refuerzo donde se exploran los entornos a través de la prueba y error. Si bien, estos no captan las relaciones causales, lo que provoca que su capacidad de transmisión de su funcionalidad a otros problemas se vea limitada.

"Un enfoque para resolver este desafío es aprender una codificación causal del entorno, porque el conocimiento causal codifica inherentemente una representación transferible del mundo". "Suponiendo que la dinámica del mundo sea constante, las relaciones causales seguirán siendo verdaderas independientemente de los cambios observacionales en el medio ambiente" (Zhu et al., 2020).

La causalidad es determinante tanto para la comprensión como para el razonamiento de los humanos en cualquier escena cuya visibilidad puede ser variable respecto del tiempo. El modelo causal And-Or Graph (C-AOG), que representa la relación causa-efecto, fue empleado por Fire et al. (2013) con el fin de hacer frente al problema de la “visibilidad y razonamiento fluido”. Se evalúa la visibilidad de un objeto con arreglo a una variable fluida en la que la interacción con su entorno es la responsable en mayor parte de su cambio. En relación con este modelo se promovió un modelo gráfico probabilístico con el propósito de demostrar que, mediante el razonamiento causal, es posible retomar y detallar trayectorias de los individuos en entornos complejos. La definición de causalidad como un cambio fluido provocado por una acción

relevante fue definido por Xiong et al. (2016) quienes emplearon el modelo C-AOG como método causal de descripción comprensiva. Demostró cómo los robots eran capaces de doblar la ropa exitosamente después de contemplar esta acción realizada por los humanos.

### 1.3.3 Funcionalidad

El sistema de visión humano se basa mayormente en tareas. Se reflexiona sobre el entorno y los objetos vistos, teniendo en cuenta las funciones que pueden realizar, siendo posible clasificarlos según sus funcionalidades. Es factible afirmar que la percepción del medio lleva, de forma inevitable, al curso de una acción, por lo que esta se percibe de forma directa e inmediata, sin necesidad de procesamiento sensorial, en objetos y entornos creados por el hombre (Gibson, 1966), ya que "se identifica un objeto primero como que tiene relaciones funcionales importantes" y "el análisis perceptual se deriva del concepto funcional" (Nelson, 1974). Para la comprensión del cómo referido al uso de un objeto es necesario que se entienda qué cambio de estado derivará tras la interacción con un objeto de una determinada manera. Sin embargo, aunque las prestaciones dependen directamente del actor, la funcionalidad de un objeto es una propiedad permanente la cual no depende de las características del usuario.

La funcionalidad de los objetos está directamente relacionada con la percepción de causalidad. Ambos conceptos entrelazados se vuelven más invariantes respecto a la compresión del objeto y el entorno que sus características de apariencia y geométricas. Resulta importante la incorporación de la funcionalidad y capacidad, desde el nivel de objeto al de escena, en el campo de la inteligencia artificial, concretamente en la visión por computador. En la segunda Conferencia Conjunta Internacional sobre Inteligencia Artificial (IJCAI) en 1971, surgió esta idea, en la cual Freeman y Newell (1971) expusieron que las estructuras disponibles deberían describirse en términos de funciones provistas y funciones desarrolladas. Por otro lado, Winston et al. (1983) argumentaron el uso de descripciones de categorías de objetos basadas en funciones. El primer trabajo que hace empleo de una definición funcional pura de una categoría de objeto sin que esta sea un modelo geométrico o estructural evidente, fue presentado por Stark y Bowyer (1991). Esta visión ha ido evolucionando en la última década. Los algoritmos de aprendizaje profundo son capaces de encontrar consistencia espacial en imágenes del mismo objeto, pero ¿qué ocurre cando es necesario que traten con una clase de objetos muy variada?

Es complejo evaluar la facultad de razonamiento funcional y causal de un sistema de visión por computador ya que no todos los sistemas verán de la misma manera. El sistema tiene que ser capaz, además de imitar los movimientos humanos en una determinada situación, de evaluarla y razonar sobre los mecanismos ocultos que pueda contener debido a que podría encontrar dificultades en una situación en la que un humano no las tiene. Este problema es conocido como "problema de correspondencia" dentro del Aprendizaje a partir de la Demostración (Learning from Demonstration, LfD) (Dautenhahn y Nehaniv, 2002). Para hacer frente a los problemas que

presenta este planteamiento, por ejemplo, el robot necesita de una comprensión más profunda de la comprensión funcional y causal de la manipulación, en consecuencia, el modelado sobre objetos y fuerzas físicas tiene que ser más explícito.

Evidentemente, en cualquiera de los enfoques existentes para abordar el reconocimiento de la funcionalidad en la visión por computador, existen limitaciones. Recientemente Liu et al. (2019) plantearon una perspectiva de espejo y una definición de manipulación funcional que extiende el LfD mediante la simulación basada en la física, de esta forma se pretende abordar el problema de correspondencia anteriormente comentado. El robot, en este caso, buscará acciones con una funcionalidad equivalente, aunque estas sean visualmente diferentes, de esta forma será posible producir el mismo efecto y, por lo tanto, alcanzar el mismo objetivo.

### 1.3.4 Intención

Tanto la percepción como la comprensión de la intención facilita a los humanos el entendimiento del comportamiento de otros de tal forma que permite su participación con estos en actividades cooperativas con unas metas compartidas. La intención ha sido tratada como mecanismo de organización básica que guía la interpretación situada en el funcionamiento cognitivo humano, de modo que debería ser un componente esencial de la Inteligencia Artificial próxima.

El mecanismo subyacente a través del cual el sistema visual humano recibe e interpreta el entorno se basa en el principio de racionalidad. Este determina que los humanos emplean su tiempo y recursos limitados exclusivamente a aquellas acciones que modifican el mundo conforme a sus intenciones y deseos, y, alcanzan sus intenciones racionalmente optimizando su utilidad a la vez que minimizan sus costes (Gergely et al., 1995; Liu et al., 2019). Pese a la complejidad del comportamiento, es la intención implícita lo que tiene importancia cuando se observa el movimiento. La intención está más estrechamente relacionada con la predicción de acciones que con la simple predicción basada en etiquetas en los sistemas actuales de visión por computador.

Uno de los temas más abordados en la investigación de un gran número de aplicaciones del mundo real, como la videovigilancia, la conducción autónoma o la interacción humano-robot se basa en la comprensión y predicción de las intenciones del hombre desde imágenes y videos.

Es esencial aprovechar al máximo tanto señales integrales como la trayectoria del movimiento, la dinámica de la mirada, postura, gestos, etc. con el fin de obtener una mejor predicción de la intención en relación con las entradas de píxeles. Shu et al. (2018) propusieron un espacio psicológico integrado que desvela la distribución entre la percepción de eventos sociales que comprenden las interacciones humanas con otros agentes y la percepción de eventos físicos que abarcan objetos inanimados, orientado a los sistemas de computación. Este espacio se forma

por dos dimensiones, la primera hace referencia al sentido intuitivo de la obediencia o no de las leyes física y la segunda de la intencionalidad deducida a partir de los movimientos de formas simples. De este modo fue posible verificar la correcta división de ambos espacios.

La mirada juega un papel fundamental en este enfoque psicológico debido a que se trata de un estímulo cognitivo ligado al cerebro, por ello, es posible deducir la intencionalidad de otros a partir de las miradas. Los sistemas de visión por computador se apoyan en las miradas como herramienta de predicción de intenciones fundamentada en imágenes y videos. Wei et al. (2018) desarrollaron un sistema jerárquico en el que se deducía simultáneamente la intención, atención y las tareas desarrolladas, este respondía a tres preguntas respecto a las personas: "¿dónde está mirando?", “¿por qué está mirando?” y “¿qué tarea está realizando?”.

### 1.3.5 Utilidad

La utilidad es uno de los principios fundamentales dentro del campo de la teoría de decisiones moderna. Se establece que a partir de una creencia o deseo en alcanzar una utilidad esperada, se toma una elección, es decir, las señales que se observan en el transcurso de la vida están impulsadas por este principio, de modo que, aplicar este estudio en la visión por computador puede aportar una mayor comprensión sobre las observaciones visuales. Es posible representar este concepto mediante la formulación matemática $U(a) > U(b)$ que indica, en este caso, que la opción (a) predomina sobre la (b), lo que significa que en la toma de decisiones de una utilidad concreta es posible clasificar la preferencia del individuo a través de esta (Russell y Norvig, 2016). Cabe destacar que se trata de una teoría positiva que pretende alcanzar el comportamiento observado y la decisión del individuo siendo este un aspecto importante para el diseño de algoritmos e interpretación de señales de observación.

Fueron numerosos estudios los que surgieron a partir de este concepto de aprendizaje, por ejemplo, Shukla et al. (2017) hicieron uso de las demostraciones humanas como utilidad de aprendizaje para instituir una tarea robótica. Fue posible que el robot aprendiera la utilidad y planificara la tarea mediante la clasificación de las imágenes extraídas, bajo el supuesto de que la utilidad de los estados objetivo se considera mayor que la de los estados iniciales. También, gracias a la observación de las personas ante un suceso, concretamente en la elección de una silla, fue posible que el sistema aprendiera las preferencias de estas en términos de utilidad interna humana, es decir, mediante la observación de los intervalos de comodidad de las fuerzas ejercidas en distintas partes del cuerpo cuando se permanece sentado (Zhu et al., 2016).

## 1.4 Evolución de los modelos de segmentación semántica basados en las CNN

Nos encontramos ante un mundo cambiante donde la computación avanza de una manera vertiginosa, pudiendo situar este momento como una era clave de la inteligencia artificial gracias al inmenso progreso del aprendizaje profundo en el que las CNN han causado un gran impacto en el campo de la visión por computador demostrando una extraordinaria eficiencia en las distintas tareas que la engloban. Las CNN partieron del estudio de la corteza visual primaria clasificando en ella neuronas simples y complejas donde el procesamiento visual empieza con estructuras simples (Hubel y Wiesel, 1968). Tomando en cuenta este estudio, Marr (1982) partió de la idea de la jerarquización de la visión. También Fukushima (1980), alentado por esta idea, a través del uso de neuronas simples y complejas, desarrolló una red neural multicapa conocida como Neocognitron permitiendo el reconocimiento de patrones en imágenes. Gracias a esta red, LeCun et al. (1989) concluyeron el estudio inicial con LeNet-5 (1989). Fue a partir de entonces cuando se impulsó el éxito de las CNN para tareas de visión por computador alentando la investigación de las capacidades de tales redes para problemas de clasificación a nivel de píxeles como la segmentación de imágenes, objeto principal de este trabajo.

Los algoritmos de aprendizaje profundo que surgieron a raíz de que las CNN lograron una gran acogida en los distintos campos de la visión por computador, así como en el área de la segmentación de imágenes (Chen et al., 2014; Srinivas et al., 2016). Esta segmentación se refiere a la forma de diferenciar en una imagen digital las múltiples regiones de las que se compone conforme a las propiedades de los píxeles, siendo esta una tarea de visión de bajo nivel o de píxel, debido a que la información espacial que se obtiene a partir de una imagen es fundamental para segmentar semánticamente las distintas regiones. De esta forma es posible obtener información significativa de la imagen donde los píxeles se etiquetan según unas determinadas características comunes (Pal y Pal, 1993; Zaitoun y Aqel, 2015).

Numerosos son los modelos de segmentación semántica basados en CNN que han surgido en los últimos años. Su eficiencia está condicionada a muchos factores tales como la arquitectura interna de la red, el tamaño del conjunto de datos, los datos de entrenamiento, el algoritmo de optimización, la función de pérdida, etc. Entre estos destaca la Red Convolucional Completa (Fully Convolutional Network, FCN) (Long et al., 2014) empleada como base para otros modelos a través de la eliminación de capas completamente conectadas que usan las CNN convencionales. En distintos estudios se han obtenido altos rendimientos gracias al uso de las características contextuales (Szegedy et al., 2014; Mostajabi et al., 2015). En la actualidad se ha llegado a emplear una red convolucional de fusión de dos corrientes con el fin de segmentar las imágenes interactivas (Chen et al., 2016).

Cabe destacar los estudios de Chen et al. (2015) que incorporaron el algoritmo conocido como “atrous” y el campo aleatorio condicional en la segmentación semántica planteando DeepLab para posteriormente implementar Atrous Spatial Pyramid Pooling (ASPP) en DeepLabv2 (Chen et al., 2016). Los autores trataron de reestructurar la arquitectura de DeepLab con el fin de optimizarla, de esta forma surgió DeepLabv3, que es una estructura mejorada respecto a los modelos anteriores ya que hace uso de un módulo ASPP profundo en cascada para posibilitar la incorporación de contextos múltiples (Chen et al., 2017).

Entre otros estudios de interés se encuentra DeconvNet que emplea una red convolucional precedida de una capa deconvolucional, es decir, se compone por capas de deconvolución y desunión para la segmentación semántica (Noh et al., 2015). Ronneberger et al. (2015) utilizaron UNet que consiste en una red en forma de U basada en la arquitectura de codificador – decodificador, donde un codificador mapea los datos de entrada a los datos codificados y un decodificador facilita la restauración de los datos partiendo de su versión codificada. Por otro lado, SegNet, que se basa también en la arquitectura de decodificador – codificador emplea índices de agrupación máximos para realizar una muestra superior, superando en desempeño a FCN, DeepLabv1 y DeconvNet. En la actualidad se ha utilizado UNet en el ámbito de la biomedicina con bloques *multiRes* para la segmentación de imágenes con unos resultados superiores al clásico UNet (Ibtehaz, Rahman, 2019). La intuición arquitectónica básica de UNet, DeconvNet y SegNet se asemejan entre sí a excepción de algunas modificaciones individuales.

Con el fin de mejorar la segmentación surge ParseNet, propuesta por Liu et al. (2015) que combina la técnica para agregar contexto global a redes convolucionales profundas para la segmentación semántica con la capa de normalización en la arquitectura FCN. “Cuando agregamos nuestra función global propuesta y una técnica para aprender los parámetros de normalización, la precisión aumenta constantemente incluso en nuestras versiones mejoradas de las líneas de base” (Liu et al., 2015). Zhao et al. (2016) presentaron la Red de análisis de Escenas Piramidales (Pyramid Scene Parsing Network, PSPNet) donde se usa el módulo *Pyramid Pooling* en el último mapa de características extraído con el fin de insertar información contextual global, lo que conlleva a una mejor segmentación. Por último, Peng et al. (2017) avanzaron hacia la Red Convolucional Global (Global Convolutional Network, GCN), que al igual que en PaseNet y PSPNet, se utiliza la información de contexto global con características locales para obtener una mejor segmentación en este caso, con el uso de un núcleo de gran tamaño.

A partir de las redes anteriores emergen otros nuevos modelos con el mismo propósito: optimizar la segmentación semántica en cualquier área. Entre ellas se encuentra DenseNet (He et al., 2016), DeepNet (Li et al., 2017), ENet (Paszke et al., 2016), ICNet (Zhao et al., 2019), entre otras. Junto a estas surgen números estudios que se centran en el uso de las CNN para abordar este tema como pueden ser los trabajos de combinación de la arquitectura de decodificador – codificador con la convolución dilatada (Chen et al., 2019; Liu et al., 2019; Sun et al., 2019),

aprendizaje por transferencia (Zhang et al., 2019) y el renderizado basado en puntos en DeepLabv3 (Chen et al., 2017).

La finalidad de este capítulo ha sido la de ofrecer una perspectiva global de todos los conceptos que giran alrededor de la segmentación semántica, desde un término más amplio, como es el aprendizaje profundo, la inteligencia artificial, la visión por computador y las redes neuronales hasta uno más concreto y objeto de estudio del presente trabajo: la segmentación semántica. De este modo se ha realizado una breve revisión de todos los estudios de investigación que han ido surgiendo a lo largo de los años hasta la actualidad para contextualizar el tema de estudio, además de adentrarse en el debate entre redes neuronales artificiales y biológicas, inteligencia artificial y percepción humana, fundamental para comprender el propósito final de los algoritmos de visión por computador dentro del paradigma del aprendizaje profundo.

A través del conciso análisis de la evolución de la AI que se ha expuesto en el capítulo se pretende dar a conocer el complejo desarrollo de los modelos de visión por computador, basados en entrenamientos, variedad de datos y análisis de patrones cuyo objetivo es el de imitar el comportamiento del cerebro humano pero que, sin embargo, la complejidad de este y la capacidad de abstracción e inconsciencia del aprendizaje diario y cognitivo del cerebro humano, limitan la evolución de un sistema artificial semejante. No obstante, la evolución y el avance de la ciencia, mediante el análisis de parámetros y el aprendizaje constante de los modelos descriptivos de la AI, han permitido realizar grandes avances en la exploración de este campo.

Los siguientes capítulos tratarán de manera más específica el objeto principal de estudio partiendo desde una perspectiva teórica hasta su aplicación práctica.

## 1.5 Objetivos y metodología

El objetivo principal de este trabajo, tal y como indica su título, es comparar analíticamente los distintos métodos de segmentación semántica mediante ejemplos prácticos, concretamente el tratamiento de imágenes aéreas. Este propósito se consigue a través de los siguientes objetivos específicos:

- Realizar una revisión bibliográfica y estudiar en profundidad la segmentación semántica con el fin de comprender todos los aspectos relacionados con esta seleccionando los más prometedores.
- Elaborar el etiquetado de las diferentes imágenes de entrenamiento que se emplearán comúnmente en todos los métodos seleccionados.
- Ajustar parámetros de los modelos, siempre que sea posible, con el propósito de buscar el ajuste óptimo.

- Analizar y comparar cuantitativa y cualitativamente los resultados obtenidos por los diferentes modelos.

Para la realización del presente trabajo se ha estudiado el campo de la segmentación semántica dentro de la visión por computador, concretamente los modelos DeepLabv3, FCNNet, SegNet y UNet. Mediante el uso de la herramienta MatLab ha sido posible simular el comportamiento de cada uno de ellos, entrenados previamente con las mismas imágenes para posteriormente visualizar el resultado obtenido frente a distintas fotografías aéreas. De este modo se ha llevado a cabo el análisis comparativo buscado.

## 1.6 Organización del trabajo

El contenido del trabajo se enfoca, tal como se ha expuesto con anterioridad, en la segmentación semántica dentro del paradigma del aprendizaje automático. Se ha realizado una estructuración por capítulos siendo estos:

- Capítulo 1: Aproximación a la visión por computador bajo aprendizaje profundo. En este capítulo se pretenden introducir todos los conceptos relacionados con el tema de estudio además de enumerar los objetivos, la metodología empleada y la organización del trabajo.
- Capítulo 2: La segmentación semántica en las CNN. Se analiza con profundad este concepto centrándose de manera teórica en los modelos concretos de estudio: DeepLabv3, FCNNet, SegNet y UNet.
- Capítulo 3: La segmentación semántica como mecanismo de prevención de incendios forestales. Desarrollo práctico. Se explica gráficamente las distintas partes del programa desarrollado con MatLab desde el punto de vista de la ejecución. Se tratará la arquitectura del sistema detallando las imágenes de entrenamiento y la clasificación propuesta.
- Capítulo 4: Análisis comparativo de los distintos modelos de segmentación semántica. Conclusiones y trabajo futuro.

# CAPÍTULO 2

## La segmentación semántica en las CNN

A lo largo de la historio de la visión por computador, la segmentación semántica ha sido uno de los grandes desafíos que afrontar. Mediante ella se pueden separar elementos o estructuras subyacentes de una imagen cualquiera en distintas partes no siendo necesario el reconocimiento o clasificación de objetos para llevarla a cabo. Una clasificación solo clasificará objetos para los que tiene etiquetas específicas mientras que un algoritmo de segmentación de imágenes ideal también segmentará objetos nuevos y desconocidos. Es posible observar un claro ejemplo en una persona, donde se es capaz de fraccionar una imagen sin conocer qué objetos se encuentran en ella, es decir, realizará la segmentación sin saber previamente la identidad exacta de todos los objetos que forman la escena, lo que se trata de una parte importante del proceso de comprensión visual humano que permite facilitarnos un modelo con el que comprender el mundo. En la actualidad, nos encontramos ante un gran número de aplicaciones en las que es posible emplear la división de imágenes para mejorar los algoritmos existentes. Así pues, se define la segmentación semántica de imágenes como una tarea de visión por computador en la que se etiquetan regiones específicas de una imagen a nivel de píxel de acuerdo con lo que se muestra.

Un enfoque hacia la construcción de una arquitectura de red neural consiste en apilar una serie de capas convolucionales con el fin de generar un mapa de segmentación final. Esto es posible gracias al mapeo y segmentación de la imagen de entrada mediante la transformación consecutiva de mapeos de características, si bien, desde la perspectiva computacional, conservar la resolución completa en toda la red resulta bastante costoso. En las redes convolucionales profundas, las capas anteriores son propensas a aprender conceptos de bajo

nivel, mientras que las posteriores son las que llevan a cabo el desarrollo de asignaciones de características especializadas de más alto nivel siendo necesario el aumento de la cantidad de mapas de características a medida que se adentra en la red. Esta cuestión no ha sido un problema para las tareas de clasificación de imágenes debido a que solamente es importante el contenido de la imagen y no el lugar donde se ubica, haciendo posible disminuir de este modo la carga computacional mediante la reducción periódica de los mapas de características a través de convoluciones escalonadas o agrupaciones, es decir, con la comprensión de la resolución espacial.

Un esquema general para los modelos de segmentación semántica de imágenes se basa en la estructura de codificador-decodificador (Badrinarayanan et al., 2017; Chen et al., 2018; Noh et al., 2015) donde se submuestrea la resolución espacial de la entrada por medio del desarrollo de asignaciones de características de menor resolución preparadas para ser altamente eficientes en la discriminación entre clases, y se sobremuestrean las representaciones de características en un mapa de segmentación de resolución completa, generalmente del mismo tamaño que la imagen original.

Los avances recientes en el aprendizaje profundo, especialmente las redes neuronales convolucionales profundas, han llevado a una mejora significativa con respecto a los sistemas de segmentación semántica anteriores. Como ejemplo, Long et al. (2014) propusieron el empleo de una red completamente convolucional (Fully Convolutional Network, FCN) entrenada de extremo a extremo, es decir, píxel por píxel para la tarea de la segmentación de imágenes. Se sugiere la adaptación de las redes de clasificación de imágenes existentes para que sirvan como módulo codificador de la red, agregando un módulo decodificador con capas convolucionales de transposición para muestrear los mapas de características gruesos en un mapa de segmentación de la resolución prevista. “La segmentación semántica enfrenta una tensión inherente entre la semántica y la ubicación: la información global resuelve qué mientras que la información local resuelve dónde. La combinación de capas finas y capas gruesas permite que el modelo haga predicciones locales que respeten la estructura global” (Long et al., 2014).

El presente capítulo trata de abordar las distintas arquitecturas de segmentación semántica de imágenes con CNN que se emplearán en este trabajo comenzando por las estructuras basadas en el modelo FCN y considerando las arquitecturas UNet, Deeplab y SegNet.

## 2.1 Segmentación semántica basada en FCN

La red FCN (Long et al., 2014) se trata de una extensión de la CNN clásica cuyo principal objetivo es conseguir que la CNN obtenga como entrada imágenes de tamaño arbitrario. Como se ha comentado anteriormente, la principal idea en la que se fundamentan los métodos basados en FCN consiste en el aprendizaje de los pesos y parámetros involucrados en la red para conseguir

el mapeo de píxel a píxel sin la necesidad de extraer regiones. Las CNN son restrictivas ya que aceptan solamente etiquetas para entradas de un tamaño específico debido a que las capas conectadas son de tamaño fijo, de forma opuesta, las FCN se forman por capas convolucionales agrupadas que capacitan a la red para poder predecir entradas de tamaño variable, por consiguiente, su tamaño de salida depende del de entrada, no como ocurre con las CNN el cual es siempre fijo. Esta característica hace que las FCN sean ampliamente utilizadas en tareas locales como la segmentación semántica (Long et al., 2014) o la detección de objetos (Dai et al., 2015) y no en globales como la clasificación de objetos (Ren et al., 2015).

Debido a la arquitectura de las FCN, compuestas por capas convolucionales agrupadas con muestreo ascendente, es posible entrenarlas de extremo a extremo. Long et al. (2014) demostraron que era posible obtener mejores modelos de FCN en un menor tiempo de entrenamiento en el aprendizaje. Para ello utilizaron dos esquemas de aprendizaje diferentes empleando una configuración con una salida densa de píxeles con 21 canales y el uso de la función de pérdida, que bien puede ser la entropía cruzada categórica por píxel para definir un clasificador con *n* clases para las que se obtiene un valor de probabilidad a través de la función *softmax*, el primero, con un lote de tamaño de 20 imágenes donde se acumularon degradados en cada una de ellas y, el segundo, con un lote de tamaño 1, conocido como aprendizaje en línea. El problema de las FCN se fundamenta en la propagación a través de las capas convoluciones que se agrupan alternadamente provocando que la resolución de los mapas de características se vea reducida en la salida, por lo tanto, las predicciones que pueden darse de manera directa en este tipo de redes probablemente son de baja resolución por lo que el resultado respecto a los límites del objeto representado por sus píxeles puede ser impreciso.

Para solventar este problema surgen diversos enfoques basados en las FCN. Uno de ellos es el propuesto por Eigen et al. (2015) que desarrollaron una red convolucional formada por múltiples escalas compuesta por subredes con distintos niveles de salida de resolución con el fin de afinar de manera progresiva la predicción. Nuevamente, Long et al. (2014) expusieron el uso de capas deconvolucionales para el muestreo de las salidas de píxeles densas, Noh et al (2015), reemplazaron este procedimiento simple para poder reconstruir con una mayor precisión estructuras de límites de objetos altamente no lineales mediante el uso de una red deconvolucional profunda, consiguieron identificar etiquetas de clase de píxeles y predecir máscaras de segmentación. DeepLab-CRF (Cheng et al., 2015; Cheng et al., 2017) ofreció una alternativa para el aumento de la resolución de salida. En primer lugar, se aplicó la convolución denominada *atrous* para aumentar la resolución de las características y luego se empleó la interpolación bilineal para sobremuestrear el mapa de puntuación con el fin de alcanzar la resolución de la imagen original. Posteriormente, se adoptó el método CRF (Krähenbühl et al., 2011) con el que se consiguió refinar el límite del objeto.

Fueron numerosos los trabajos que trataron de mejorar la precisión de la segmentación por medio de la explotación de la información contextual. Liu et al. (2016) emplearon la combinación de promedios globales con el fin de conseguir el contexto general, agregándolo a las redes completamente convolucionales para la segmentación semántica, esto provocó que la precisión aumentara considerablemente. Las convoluciones dilatadas exponencialmente expandidas para agregar información contextual surgieron como alternativa, también, Lin et al. (2017) exploraron distintos tipos de contexto espacial a fin de mejorar el rendimiento de la segmentación usando el método CRF para modelar explícitamente las relaciones contextuales (Guo et al., 2017).

En conclusión, las FCN pueden aportar una solución a la arquitectura tradicional de las CNN de manera que, mediante el apilamiento de una serie de capas de convolución con un contenido similar es posible mantener la dimensión y producir un mapa de segmentación final lo que supone que el modelo es capaz de aprender la proyección de la imagen de entrada a su mapa de segmentación correspondiente gracias a la transformación sucesiva de mapeos de características. El problema que genera el uso del mismo relleno en todas las capas de convolución, debido a que se pretende obtener la misma dimensión en la salida que la de la imagen de entrada, sería posible afrontarlo con tareas de clasificación ya que el único elemento importante es el objeto de interés, en este caso el píxel, no siendo necesario conservar la información sobre su ubicación en la escena, por esta razón es posible reducir la resolución de las imágenes mediante la agrupación. Sin embargo, esta solución no es válida desde el punto de vista de la segmentación semántica puesto que es necesario conservar otros elementos en la escena, por ejemplo, en la segmentación de una imagen para el entrenamiento de un vehículo autónomo es necesario tener presentes otros objetos que formarán la salida del sistema.

El submuestreo y sobremuestreo surgen como solución ante el problema que presenta la red completamente convolucional donde en la primera mitad del modelo se reduce la resolución espacial de la imagen, desarrollando mapeos de características complejas. Es posible conseguir información más detallada en cada una de las convoluciones a través de la discriminación entre las distintas clases, en consecuencia, se pierde la información relativa a la ubicación. Con el fin de recuperar esta información perdida se aplica el sobremuestreo, es decir, el aumento de muestreo que trata diversas imágenes de resolución menor como entrada y entrega un mapa de segmentación de alta resolución como salida. En la figura 2.1 es posible observar una imagen de resolución *H x W* con convolución *H/2 x W/2* y finalmente *H/4 x W/4* siendo el resultado final un mapa de segmentación de alta resolución (HxW), y por lo tanto de la misma dimensión que la imagen original, con cada píxel clasificado en la clase de probabilidad más alta.

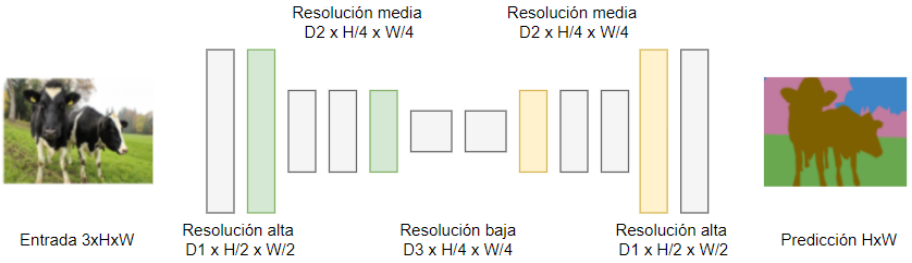



Figura 2.1: Arquitectura FCN: Submuestreo y sobremuestreo.

### 2.1.1 Conceptos

Como se ha indicado previamente, los modelos de redes FCN, figura 2.1, se basan en una fase de submuestreo seguida de otra de sobremuestreo. En la fase de submuestreo se aplican operaciones de convolución y agrupación con el fin de reducir la dimensionalidad de las imágenes de entrada, para posteriormente incrementarla mediante operaciones "opuestas", esto es deconvolución o convolución transpuesta y desagrupamiento, hasta conseguir la misma resolución que la imagen original. En ambas fases se aplican además dos operaciones de gran relevancia: ReLU y Normalización. En cualquier caso, como el objetivo final consiste en asignar cada píxel a una etiqueta dada, en la última capa se calcula una probabilidad de pertenencia de cada píxel a una determinada clase, lo cual se consigue mediante la denominada función *softmax*. A partir de la probabilidad calculada en relación con la probabilidad esperada para cada píxel, dado que el entrenamiento está basado en un conjunto de imágenes etiquetadas, se define lo que se conoce como función de pérdida (loss), con la que se determina el error cometido en cada caso. Este error es el que se propaga hacia atrás en la red (retropropagación), permitiendo actualizar los pesos involucrados en las distintas capas del modelo. En esta actualización interviene también lo que se denomina razón de aprendizaje (learning rate), que puede ser fija a lo largo de todo el proceso o irse decrementando a medida que se avanza en las iteraciones. Dicha actualización se lleva a cabo mediante el correspondiente proceso de optimización, durante el cual las imágenes se procesan por lotes (batch). A continuación, se definen brevemente los conceptos mencionados (Pajares et al., 2021).

- Unidad Lineal Rectificada (ReLU, Rectified Linear Unit): definida como $f(x) = max(0, x)$ y representada en la figura 2.2. Se caracteriza principalmente por:

    - Evitar la saturación del gradiente durante el proceso de optimización por el método del gradiente descendente.
    - Disminuir la complejidad computacional debido a que reduce valores negativos a cero.

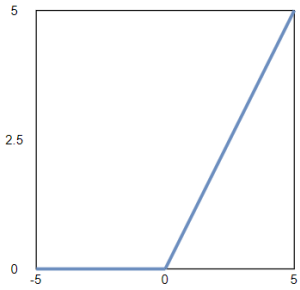



Figura 2.2: Representación de la función ReLU

- Normalización: la operación de normalización es la que se emplea con el objeto de acelerar la convergencia a lo largo del proceso de aprendizaje. La expresión que define esta función por lotes viene dada por la ecuación:

$$b_{x,y}^{i} = \frac{a_{x,y}^{i}}{\left(k + \alpha \sum_{j=max(0,i-n/2)}^{min(N-1,i+n/2)} \left(a_{x,y}^{i}\right)^{2}\right)^{\beta}} \tag{2.1}$$

Donde:

N = Número de filtros de la capa dada

$a_{x,y}^{i}$= actividad de la neurona

k, α, n, β = hiperparámetros. En Krizhevsky (2012) se proponen como más apropiados los siguientes valores para los parámetros k = 2, n = 5, α=10^(-4), β = 0.75.

- *Softmax* y entropía cruzada: en la función *softmax* los valores obtenidos en la capa de salida se procesan de tal forma que se encuentran en el rango [0,1], siendo su suma 1. De esta manera se obtiene una probabilidad de pertenencia a las distintas clases de las que se compone una imagen de entrada dada. La expresión que la representa viene dada por:

$$softmax(x)_i = \frac{exp(x_i)}{\sum_{j=1}^{n} exp(x_j)} \tag{2.2}$$

La *función de pérdida* conocida como *entropía cruzada* (cross entropy) está estrechamente relacionada con la anterior, pero en este caso se consideran dos distribuciones de probabilidad. Por un lado, la relacionada con los valores predichos $q \equiv \{q_1, q_2, \ldots, q_n\}$ de manera que $\sum_i^n q_i = 1$ y, por otro lado, con los verdaderos (ground truth) $p \equiv \{p_1, p_2, \ldots, p_n\}$ *que también cumple* $\sum_i^n q_i = 1$. Así pues, la entropía cruzada viene expresada por:

$$H(p,q) = -\frac{1}{n}\sum_{i}^{n} p_i log\,(q_i) \tag{2.3}$$

- Optimización: el gradiente descendente es un método de optimización numérica cuya finalidad es la de estimar los mejores coeficientes, es decir, se emplea para la ajustar los pesos implicados en las redes durante el proceso de retropropagación. Su finalidad es la de minimizar una función a optimizar, conocida como función objetivo, que tiene la forma:

$$J(w) = \frac{1}{n}\sum_{i=1}^{n} J_i(w) \tag{2.4}$$

Consiste en encontrar qué valor del parámetro *w* minimiza *J(w)* siendo $J_i$ relativo a la i-ésima observación en el conjunto de datos utilizados para el entrenamiento. El gradiente descendente minimiza esta función de forma iterativa, con iteraciones *t*:

$$w(t+1) = w(t) - \varepsilon \frac{1}{n}\sum_{i=1}^{n} \nabla J_i(w) \tag{2.5}$$

El método es capaz de recorrer iterativamente el conjunto de entrenamiento mientras efectúa la actualización anterior para cada una de las muestras de entrenamiento hasta la convergencia del algoritmo. Las muestras seleccionadas forman lo que se denomina *batch*. La técnica conocida como Gradiente Descendente Estocástico (Stochastic Gradient Descent, SGD) (Bishop, 2006; Ruder, 2017) es que se utiliza habitualmente.

Es posible el uso de una razón de aprendizaje adaptativa con el fin de que el algoritmo converja. En pseudocódigo, el SGD sigue las siguientes pautas:

1. Se elije un vector inicial de parámetros w y una razón de aprendizaje $\varepsilon$.
2. Se repite hasta alcanzar un mínimo aproximado.
    2.1 De manera aleatoria se seleccionan ejemplos en el conjunto de entrenamiento.
    2.2 Para *i = 1, 2, …, n*, se realiza $w(t+1) = w(t) - \varepsilon \nabla J_i(w)$

Dentro de la diversidad de variantes del algoritmo, es necesario establecer una razón de aprendizaje adecuada debido a que, si esta se compone por valores elevados, los cambios de *w* serán también muy grandes y en consecuencia se tenderá a la divergencia de datos, en contra, ante valores pequeños, el gradiente descendiente tardará mucho en encontrar la solución adecuada por lo que la convergencia será lenta. Una de las maneras de hacer frente a esta problemática consiste en hacer que la razón sea variable disminuyendo su valor conforme se van incorporando muestras nuevas por lo que será dependiente del número de datos o iteraciones. El método del *momentum* o momento (Rumelhart et al., 1986; Murphy, 2012), surge como extensión a este algoritmo en el que se realiza en cada iteración la

actualización $\Delta w$ estableciendo la actualización siguiente como una combinación lineal del gradiente y la actualización previa (Sutskever et al., 2013):

$$\begin{aligned} \Delta w(t) &= \alpha \Delta w(t-1) - \varepsilon \nabla J_i(w(t-1)) \\ w(t) &= w(t-1) + \Delta w(t-1) \end{aligned} \tag{2.6}$$

De la expresión anterior se llega a:

$$w(t) = w(t-1) + \alpha \Delta w(t-1) - \varepsilon \nabla J_i(w(t-1)) \tag{2.7}$$

Donde:

$\varepsilon > 0$ = razón de aprendizaje

$\alpha \in [0,1]$ = momento constante que controla la velocidad de actualización $\Delta w(t-1)$

w = vector de parámetros que minimiza *J(w)*

Otro optimizador que destacar se conoce como *ADAM* cuyo nombre proviene de *Adaptive Moment Estimation* (Kingma y Ba, 2015). Consigue que elemento a elemento se mantenga una actualización de la media móvil, ya sea de los gradientes de los parámetros, así como de sus valores al cuadrado. Se expresa como:

$$\begin{aligned} m(t) &= \beta_1 m(t-1) + (1-\beta_1) \nabla J_i(w(t-1) \\ v(t) &= \beta_2 v(t-1) + (1-\beta_2)(\nabla J_i(w(t-1))^2 \\ w(t) &= w(t-1) - \frac{\alpha m(t-1)}{\sqrt{v(t-1)} + \varepsilon} \end{aligned} \tag{2.8}$$

- Razón de aprendizaje *(learning rate)*: encargada de establecer la rapidez de actualización de la red además de ajustar los pesos que comprende. Como se ha comentado, la razón de aprendizaje es variable de tal manera que, partiendo de una razón fija, se emplea un factor de reducción cada determinado número de *epochs.*

- *Batch size*: la dimensión del lote es un hiperparámetro del gradiente descendiente que controla la cantidad de muestras de entrenamiento con las que se trabajará antes de que se actualicen los parámetros internos del modelo, concretamente la actualización de los pesos de la red.

- *Epochs* e iteraciones: establece el número de veces que el conjunto total de muestras de entrenamiento son procesadas por el método de optimización donde cada una de ellas ha tenido la opción de actualizar los pesos en cada *epoch*. Para poder detener el proceso de entramiento en el caso de que no se alcance la convergencia y el error no sea aceptable, es posible limitar el número de *epochs* en los que el proceso se detendrá.

Otros de los conceptos generales que caracterizan las FCN son:

- Convolución: Se trata de la primera capa en la que se extraen las características de la imagen de entrada mediante la multiplicación de la matriz que genera esta con una matriz de filtro siendo posible aplicar tantas matrices de filtros como características se quieran obtener. A medida que se profundiza en la red de convolución el nivel de complejidad de las características aumenta. El resultado es una red holística secuencial de la imagen, semejante al procesamiento humano ante una imagen, figura 2.3.

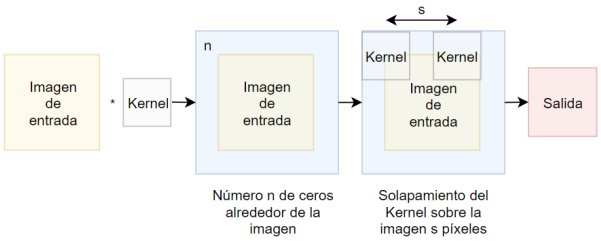



Figura 2.3: Capa de convolución.

- Agrupación (Pooling): La agrupación es la operación que sucede seguidamente de la convolución. Mediante esta se obtiene una mayor reducción de las características complejas al mismo tiempo que es capaz de reducir el ruido y de extraer las características dominantes, es decir, las que no varían ante la posición y rotación. Es posible llevarla a cabo de dos maneras: agrupación máxima que retorna el valor máximo de la imagen y la agrupación promedio la cual devuelve el promedio de los valores de la misma, figura 2.4.

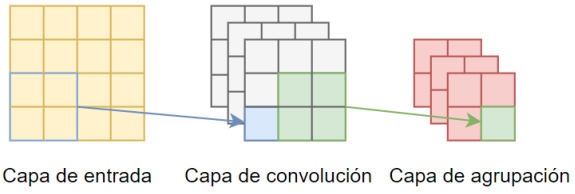



Figura 2.4: Capa de convolución y agrupación

- Desagrupación (unpooling): tras la reducción de la resolución extrayendo características complejas y la agrupación es fundamental, dentro del segmentación semántica, obtener la resolución original de la imagen de entrada a la salida por ello, a través de la desagrupación es posible convertir un único valor en un parche de valores. Existen diversos métodos para aplicar esta función, cabe destacar:

    - El “vecino más próximo” que, tras la elección de un valor, las celdas más cercanas toman ese mismo valor, figura 2.5.

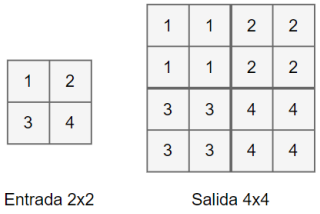



Figura 2.5: Desagrupación “el vecino más próximo”

- La “cama de clavos” se llena el valor en una celda predefinida y el resto con ceros, figura 2.6.

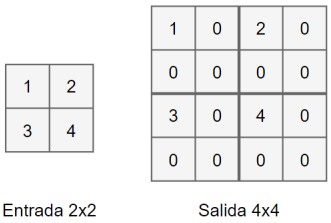



Figura 2.6: Desagrupación “cama de clavos”.

- La desagrupación máxima nace como mejora al método anterior conocido como “cama de clavos”. Emplea la simetría característica de una red de submuestreo-sobremuestreo, es decir, para cada capa de submuestreo existe una similar de sobremuestreo. En este caso, el valor se completa en la celda que posee el valor máximo en la correspondiente capa de submuestreo como se observa en la figura 2.7.

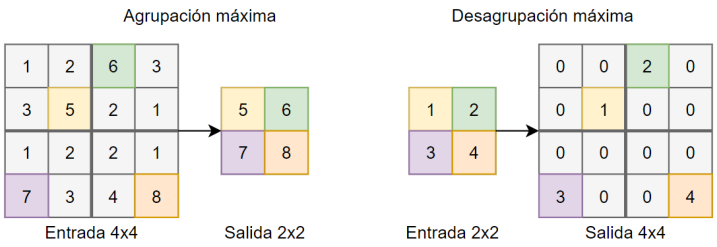



Figura 2.7: Agrupación y desagrupación máxima.

- Deconvolución o convolución transpuesta: se utiliza una capa para sobremuestrear y recuperar la resolución original puesto que mediante el aprendizaje de un conjunto de valores es posible obtener el resultado final de las características de menor resolución. Se puede considerar la convolución transpuesta como una mejor técnica de sobremuestreo que las tratadas anteriormente ya que a diferencia de estas, puede aprender lo que hace que llegue a ser muy eficiente.

- El principal problema al aplicar el submuestreo en una red FCN radica en que la resolución de entrada se ve muy reducida por lo que el sobremuestreo posterior se complica en cuanto a la reproducción de detalles incluso después de la deconvolución provocando que la imagen de salida pueda ser imprecisa. A través del método de omisión de conexiones es posible hacer frente a esta cuestión gracias a la combinación de capas finas y gruesas que conducen a predicciones locales con una estructura global casi precisa.

## 2.2 UNet

UNet nace ante la necesidad de conocer la ubicación de los objetos en las imágenes, concretamente para el procesamiento de imágenes biomédicas que requieren la localización del resultado final. Ciresan et al. (2012) trataron de alcanzar este fin a través del entrenamiento de una red en configuración de “ventana deslizante” con la que fue posible predecir la etiqueta de clase de cada uno de los píxeles mediante un parche vinculado con la región local. Aunque tuvieron un resultado sobresaliente, este estudio presentó diversos inconvenientes, entre ellos destaca la lentitud provocada por los numerosos parches que se han de ejecutar además de la redundancia causada por la superposición de estos, también se requiere precisión en la localización condicionada al contexto, de esta forma el uso de agrupación máxima de capas para los parches de mayor tamaño provocan la reducción de la precisión de la localización a la vez que los pequeños facilitan la visión del contexto a la red. Emergen distintos enfoques con la misma finalidad basados en el uso de una salida de clasificador de múltiples capas donde la localización y el uso del contexto se dan de manera simultánea (Seyedhosseini et al., 2013), en cambio, como comentan los autores del estudio, “UNet emplea las FCN modificando y ampliando su arquitectura para que pueda trabajar con un número pequeño de imágenes de entrenamiento con el objetivo de producir segmentaciones más precisas” (Ronneberger et al., 2015).

Una de las principales diferencias de UNet frente a otras FCN se sitúa en la arquitectura, caracterizada por su forma de U, donde en la parte de sobremuestreo se albergan numerosos canales de características que facilitan la propagación de información de contexto a otras capas de mayor resolución. A razón de esto, la trayectoria expansiva podría considerarse simétrica respecto a la trayectoria de contracción. Las capas no se encuentran completamente conectadas, se sirven de la parte válida de cada una de las convoluciones lo que quiere decir que el contexto completo solamente estará disponible en la imagen de entrada, pero el mapa de segmentación contendrá los píxeles. De esta forma es posible alcanzar una segmentación precisa de grandes imágenes a través de la estrategia del “mosaico superpuesto” donde los píxeles en la región del borde se agregan simétricamente alrededor de la imagen para que las imágenes se puedan segmentar de forma continuada. Esta estrategia se basa en la adición de píxeles, de lo contrario, la resolución estará limitada por la capacidad de la memoria de la GPU.

Las variaciones más comunes en el tejido en una imagen biomédica son las deformaciones, por lo que el estudio de Ciresan et al. (2012) trata de simularlas de forma eficiente para hacerles frente. De esta manera, el proceso de aprendizaje de la red es más exitoso ya que en la fase de entrenamiento se emplean pocas imágenes que se someten a deformaciones elásticas aumentando el tamaño del conjunto de datos. Se consigue que la red sea capaz de aprender la invarianza de estas deformaciones sin que necesariamente se sitúen en la entrada de entrenamiento. Destacan los estudios de Dosovitskiy et al. (2014) en relación con esta propiedad donde se trata el valor del incremento de datos para la invarianza dentro del aprendizaje de funciones no supervisado.

Finalmente, se aborda la problemática de la separación de los límites de los objetos de la misma clase cuando están en contacto proponiendo como solución la pérdida ponderada aplicada a las etiquetas. Para ello, se recomienda utilizar los valores que tengan un gran peso en la función de pérdida, separando primero la información a segmentar del fondo (Ronneberger et al., 2015).

### 2.2.1 Arquitectura

La arquitectura de UNet se compone de tres secciones: la contracción, el cuello de botella y la sección de expansión. La sección de contracción se forma por varios bloques de contracción, cada uno de ellos toma una entrada y aplica dos capas de convolución de 3x3 seguidas de una agrupación máxima de 2x2. El número de núcleos o mapas de características después de cada bloque se ve duplicado con el fin de que la arquitectura tenga la capacidad de aprender las estructuras complejas de forma efectiva. La capa más inferior se sitúa entre la capa de contracción y la capa de expansión utilizando dos capas CNN de 3x3 seguidas de una capa de convolución superior de 2x2.

El corazón de esta arquitectura radica en la sección de expansión, similar a la capa de contracción, pero formada por diversos bloques de expansión. Cada bloque pasa la entrada a dos capas CNN de 3x3 seguidas de una capa de muestreo ascendente de 2x2. Asimismo, después de cada bloque, el número de mapas de características utilizados por la capa convolucional obtiene la mitad para mantener la simetría, no obstante, la entrada también se agrega mediante mapas de características de la capa de contracción correspondiente. Esto garantiza que las características aprendidas en la contracción de la imagen se empleen para reconstruirla, siendo el mismo el número de bloques de expansión que el de contracción. Finalmente, el mapeo resultante pasa a través de otra capa CNN 3x3 con el número de mapas de características igual al número de segmentos deseados. En la figura 2.8 se observa la representación gráfica de esta arquitectura.

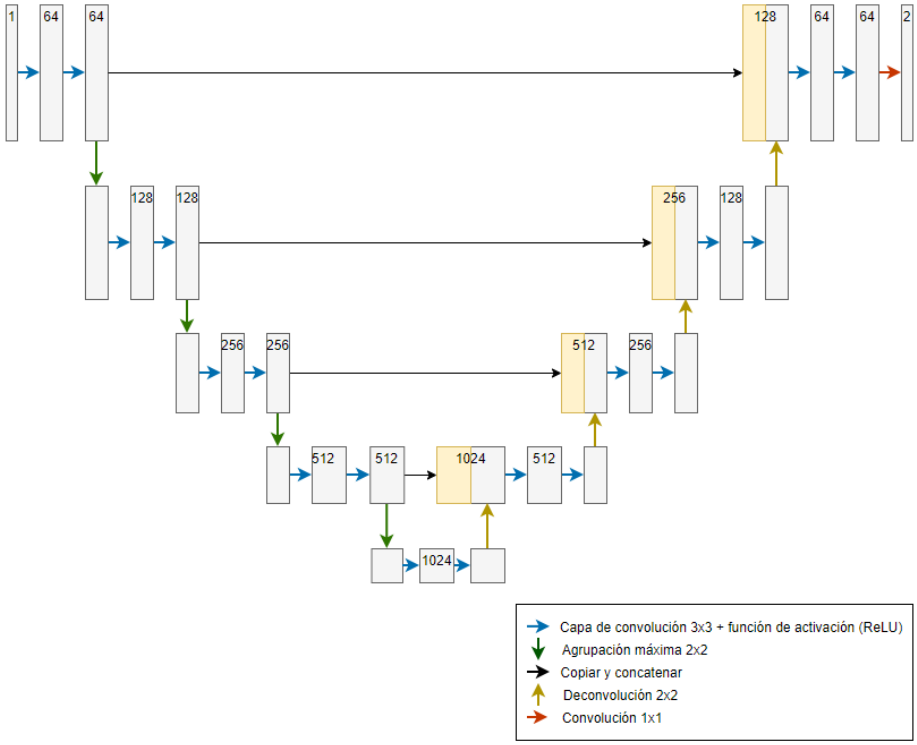



Figura 2.8: Representación y ejemplo de la arquitectura de UNet.

### 2.2.2 Cálculo de pérdidas en UNet

El propio estudio define este concepto: “La función de energía se calcula mediante *softmax* aplicado en píxeles sobre el mapa de características final combinado con la función de pérdida de entropía cruzada” (Ronneberger et al., 2015).

El esquema de ponderación de pérdidas utilizado por UNet es bastante novedoso donde el mayor peso se sitúa en el borde de los objetos segmentados. Para ello, se utiliza la función *softmax* aplicada a los píxeles de la imagen resultante, seguido de la pérdida de entropía cruzada. De esta forma es posible clasificar cada uno de los píxeles en una de las clases cuya idea final reside en que incluso en la segmentación, cada píxel debe pertenecer a alguna categoría y solo hay que asegurarse de que así sea, por ello, se parte de un problema de segmentación para convertirlo en uno de clasificación multiclase.

Por supuesto, la segmentación no solo se usa para imágenes médicas. Las ciencias de la tierra o los sistemas de teledetección a partir de imágenes de satélite también utilizan la segmentación, al igual que los sistemas de vehículos autónomos. Después de todo, hay patrones en todas partes.

## 2.3 DeepLab

DeepLab es un modelo de segmentación semántica de última generación diseñado, de código abierto, por Google en 2016 para la segmentación semántica de imágenes donde su objetivo es asignar etiquetas semánticas, por ejemplo: persona, gato, perro, etc. a cada píxel de la imagen de entrada. Desde entonces, se han realizado múltiples mejoras del modelo, incluidos DeepLabv2, DeepLabv3 y el último DeepLabv3+.

Las principales diferencias entre las distintas versiones se basan en:

1) DeepLabv1: utilizan la convolución *atrous* para controlar explícitamente la resolución a la que se calculan las respuestas de las características dentro de las redes neuronales convolucionales profundas.

2) DeepLabv2: emplea la combinación de pirámides espaciales, conocida como agrupación *atrous* de pirámides espaciales (Atrous Spatial Pyramid Pooling, ASPP), para segmentar objetos de manera robusta a múltiples escalas con filtros a también múltiples tasas de muestreo y campos de visión efectivos.

3) DeepLabv3: aumentan el módulo ASPP con una función de nivel de imagen para capturar información de mayor alcance. También incluye parámetros de normalización por lotes para facilitar el entrenamiento, en particular, aplican la convolución *atrous* con el fin de extraer características de salida durante el entrenamiento y la evaluación.

4) DeepLabv3+: amplían DeepLabv3 para incluir un módulo decodificador simple pero efectivo para refinar los resultados de la segmentación, especialmente a lo largo de los límites de los objetos. Además, en esta estructura de codificador – decodificador, se puede controlar arbitrariamente la resolución de las características del codificador extraídas mediante una convolución *atrous* para compensar la precisión y el tiempo de ejecución.

La revisión v3 de DeepLab es la que se empleará en el presente trabajo. A continuación, se tratará con mayor detalle.

### 2.3.1 DeepLabv3

Chen et al. (2017) trataron de mejorar las versiones anteriores de DeepLab mediante su evolución. Las redes neuronales convolucionales profundas (Deep Convolutional Neural Networks, DCNN) (LeCun et al., 1989) presentan un desafío en su aplicación para las tareas de segmentación semántica. El primero de ellos radica en la resolución reducida de las

características provocada por las distintas operaciones de agrupación consecutivas lo que hace posible que las DCNN sean capaces de aprender representaciones de características que se hacen cada vez más abstractas, aunque esto se trate de una capacidad favorable para el modelo; por otro lado, la invarianza a la transformación de la imagen local se puede oponer a las tareas donde es necesaria una mayor información detallada, siendo estas tareas de predicción densas. Con el fin de abordar esta problemática, Chen et al. (2017) proponen el uso de la convolución *atrous*, también conocida como convolución dilatada (Holschneider et al., 1989; Ciresan et al., 2013; Sermanet et al., 2013; Papandreou et al., 2015) que permite controlar la resolución a la que se calculan las respuestas de las características dentro de las DCNN sin que sea necesario el aprendizaje de parámetros adicionales mediante el uso de redes preentrenadas (Russakovsky et al., 2015) con el fin de extraer los mapas de características más densos a través de la eliminación de las operaciones de reducción de muestreo que se sitúan en las últimas capas.

La existencia de objetos a múltiples escalas presenta otra complejidad que afrontar. Los diferentes métodos que se exponen en el estudio para subsanar esta dificultad se muestran en la figura 2.9: en a) se parte de una pirámide de imágenes a la que se aplica DCNN con el propósito de extraer propiedades para cada entrada de escala (Farabet et al., 2013; Eigen y Fergus, 2014; Pinheiro y Collobert, 2014; Lin et al., 2015; Chen et al., 2016) donde sobresalen los objetos en los distintos mapas de características, en b) se representa la estructura de codificador-decodificador (Badrinarayanan et al., 2015; Ronneberger et al., 2015; Ghiasi y Fowlkes, 2016; Lin et al., 2017; Pohlen et al., 2016; Peng et al., 2017) donde es posible recuperar la resolución espacial de la parte del decodificador además de exprimir las características de múltiples escalas en la parte del codificador, en c) a partir de la red original se sitúan los módulos adicionales en cascada con el propósito de obtener la información de largo alcance gracias al uso de DenseCRF (Krähenbühl and V. Koltun, 2011) el cual codifica similitudes por pares a nivel de píxel al mismo tiempo que se incrementan las capas en cascada con el fin de obtener de manera gradual este contexto (Liu et al., 2015; Yu and V. Koltun, 2016). Finalmente, en d) se combinan las pirámides espaciales (Chen et al., 2016; Zhao et al., 2016) para obtener un mapa de características de entrada que es capaz de capturar objetos a escalas múltiples.

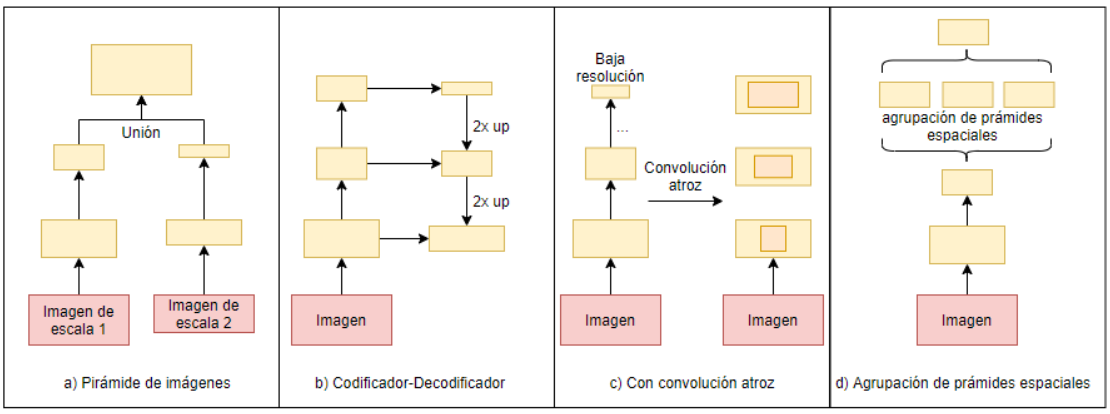



Figura 2.9: Arquitecturas alternativas para capturar contexto multiescala.

El propósito de DeepLabv3 se basa en el empleo de la convolución *atrous* que permite ampliar el campo de visión de los filtros añadiendo contexto de múltiples capas mediante un módulo en cascada y la combinación de pirámides espaciales para lograr una mejora respecto a los trabajos anteriores (Chen et al., 2015; Chen et al., 2016), alcanzando un rendimiento del 85.7% dentro del conjunto de datos PASCAL VOC 2012 sin procesamiento posterior DenseCRF (Chen et al., 2017).

Así pues, DeepLabv3 combina diversos conceptos de visión por computador dentro del aprendizaje profundo:

- Combinación de pirámides espaciales: esta arquitectura ayuda a procesar la información de la imagen a diferentes escalas. Sus redes de agrupación generalmente emplean versiones paralelas de la misma red subyacente para entrenar entradas a diferentes escalas y combinar las características en un paso posterior.

- Arquitecturas de codificador – decodificador: se emplea el codificador para reducir la escala de la imagen a un vector de características que resume la esencia de la imagen y luego se usa un decodificador para expandir el vector de características retornando de nuevo a las dimensiones de la imagen, sin embargo, el decodificador devuelve una imagen con segmentación semántica.

- Convoluciones *atrous*: requieren del parámetro llamado “tasa” que se utiliza para controlar explícitamente el campo de visión efectivo de la convolución. Su beneficio se encuentra en la posibilidad que ofrecen de capturar información desde un campo de visión efectivo más grande mientras usan el mismo número de parámetros y complejidad computacional. Un ejemplo de esta se puede observar en la figura 2.10.

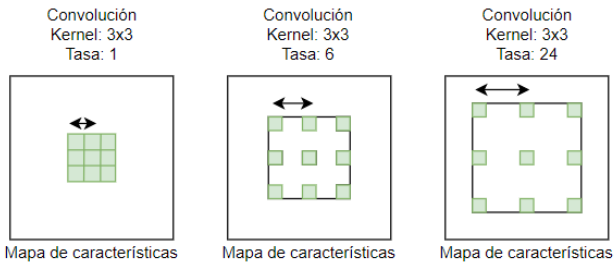



Figura 2.10: Convolución *atrous*

Analíticamente, la convolución *atrous* se representa por la ecuación:

$$y[i] = \sum_{k} x[i + r \cdot k]w[k] \tag{2.9}$$

Donde:

x = entrada

i = ubicación

y = salida

w = filtro

r = tasa *atrous,* que controla la dilatación

Se aplica una convolución *atrous* sobre el mapa de características de entrada *x* siendo la tasa *atrous r* correspondiente a la zancada con la que se muestrea la señal de entrada para cada ubicación *i* en la salida *y* un filtro *w*. Este proceso corresponde con la convolución de la entrada *x* con filtros de muestreo superior que se producen por la inserción de ceros *r-1* entre dos valores de filtro seguidos situados a lo largo de cada una de las dimensiones espaciales. Si se da el caso que *r = 1*, la convolución es estándar. Al tratar de ajustar el parámetro *r* es posible modificar el campo de visión del filtro (Chen et al., 2017).

## 2.4 SegNet

Badrinarayanan et al. (2017), al igual que en los estudios comentados anteriormente, llevaron a cabo un trabajo cuyo principal objetivo es el de encontrar una mejor arquitectura para su implementación en la segmentación semántica. En este caso, SegNet surge con el fin de comprender las escenas de carretera, desde la modelación de la apariencia y la forma hasta la relación espacial existente entre las distintas clases que la forman. En este entorno es necesario el uso de segmentaciones suaves debido a que gran parte de los píxeles forman clases de gran tamaño además de ser indispensable la conservación de la información de los límites de los objetos independientemente de su tamaño. Por ello es esencial que la red neuronal sea eficiente en relación con la memoria o tiempo de cálculo. SegNet nace ante estos requerimientos donde es necesario el entrenamiento de extremo a extremo con el objeto de optimizar todos los pesos de la red. Para ello se emplea la técnica de Descenso de Gradiente Estocástico (Stochastic Gradient Descent, SGD) (Bottou, 2010) siendo su red de codificador topológicamente igual a las capas convolucionales en VGG-16 (Simonyan y Zisserman, 2014) pero eliminando las capas que se encuentran completamente conectadas, consiguiendo un menor tamaño y mayor simplicidad para el entrenamiento.

Por otro lado, SegNet emplea en su red de decodificadores una idea similar a la presentada en la arquitectura trazada para el aprendizaje de funciones sin supervisión (Ranzato et al., 2007) en la cual se desarrolla la reutilización de índices de agrupación máxima que supone mejoras en la delimitación de los límites, reducción de parámetros de tal forma que el entrenamiento de extremo a extremo es posible y el uso de sobremuestreo con una ligera modificación sobre una arquitectura de codificador – decodificador (Long et al., 2014; Zheng et al., 2015). Así pues, se pretende dar un nuevo enfoque al diseño de arquitecturas de segmentación destacando el

análisis de la técnica de decodificación que proporciona SegNet y la FCN involucrada. La figura 2.11 muestra un esquema de codificación-decodificación para el modelo SegNet.

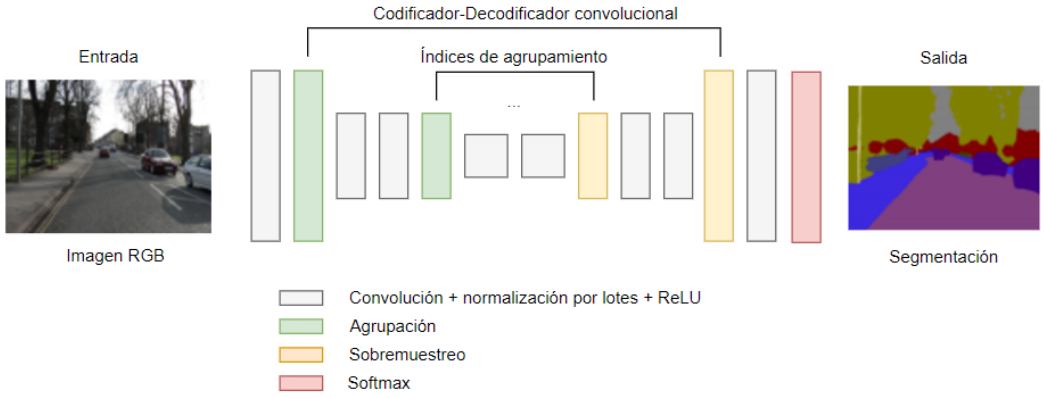



Figura 2.11: SegNet.

### 2.4.1 Arquitectura

Como se ha mencionado previamente, la arquitectura de segmentación semántica genérica viene dada por una red de codificadores seguida de una red de decodificadores donde el codificador suele ser la red de clasificación previamente entrenada y el decodificador trata la proyección semántica de las características discriminatorias que se aprenden en el codificador a nivel de píxel. SegNet sigue este ejemplo en el cual toma las primeras 13 capas convolucionales de la red VGG-16 para emplearlas en su red de codificador formada por el mismo número de capas convolucionales correspondientes con las anteriores. Gracias a su proceso de entrenamiento es posible reducir de manera considerable el número de parámetros de la red de codificadores de SegNet frente a otros modelos, por lo que este valor se sitúa entre los 134 a 147 millones debido a que emplea pesos entrenados para la clasificación en conjuntos de datos de gran tamaño además de descartar las capas que se encuentran plenamente conectadas para conservar los mapas de características de mayor resolución. Después del último decodificador, la salida se envía a la función *softmax* que da la predicción final. Esto permite establecer probabilidades de clase para cada uno de los píxeles derivando en una imagen de *n* canales, por lo que es necesario escribir una función separada para convertirla en una imagen RGB donde sea posible ver los resultados cualitativamente.

Los pasos que siguen los codificares para llevar a cabo su función son:

- Se realiza la convolución con un banco de filtros con el fin de obtener un conjunto de mapas de características.
- Posteriormente estos mapas se normalizarán por lotes.
- Se emplea sobre la entrada *x* la función de activación *f(x) = max(0, x)* donde la neurona utilizada se conoce como Unidad Lineal Rectificada (Rectified Linear Unit, ReLU).

- A continuación, se realiza la agrupación máxima con una ventana de 2×2 y paso 2. Esto indica que solamente se almacenará el valor máximo dentro de dicha ventana que será el más significativo. De esta forma se guardan los índices obtenidos con el fin de no perder información relevante en la imagen y, por lo tanto, mantener la resolución requerida.
- La salida resultante se submuestrea por un factor de 2.

Posteriormente, los decodificadores siguen los siguientes pasos:

- Los índices de agrupación máxima que se guardaron anteriormente se emplean para la interpolación de entrada de los decodificadores mediante una ventana de 2x2. El valor en un determinado punto del mapa se situará en la posición prefijada por el índice de agrupación máxima y el resto se considerarán cero. En la figura 2.12 se muestra cómo en el proceso de decodificación, para sobremuestrear las capas, SegNet utiliza los índices de agrupación máxima en la capa de codificador correspondiente. Esto facilita el proceso de entrenamiento ya que la red no necesita volver a aprender los pesos de muestreo.

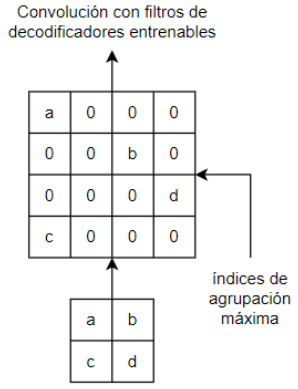



Figura 2.12: Ejemplo de proceso de decodificación en SegNet.

- Seguidamente, se procede con la convolución de los mapas de características anteriores, usando un mapa de filtros modificable donde los parámetros que antes fueron cero toman diferentes valores para generar mapas de características densos.
- Se aplica la normalización por lotes a cada uno de los mapas anteriores.

Se emplea la función *softmax* en la salida obtenida tras el último decodificador. Gracias a esta es posible obtener la probabilidad de cada píxel de manera independiente siendo su resultado una imagen de probabilidades de canal *k*, donde *k* se refiere al número de clases. La clase con máxima probabilidad en cada píxel corresponderá con la segmentación predicha.

En el presente capítulo se han tratado cuatro de las principales arquitecturas de segmentación semántica dentro del aprendizaje profundo cuya finalidad es la de alcanzar un mejor entendimiento de las imágenes ya que la clasificación se trata a nivel de píxel. A diferencia de otros métodos, como la detección de objetos y la clasificación de imágenes, la segmentación semántica es capaz de proporcionar no solamente la categoría, tamaño o número del objeto, sino que permite conocer de manera precisa el límite y la posición de este. La principal contribución de este trabajo consiste en abordar una investigación comparativa de los distintos métodos propuestos: FCNNet, DeepLabv3, UNet y SegNet. Para alcanzar dicho fin, en el siguiente capítulo se discutirá de forma práctica su aplicación para cada uno de ellos sobre el mismo conjunto de imágenes de entrenamiento por lo que se contrastarán de manera objetiva.

# CAPÍTULO 3

## La segmentación semántica como mecanismo de prevención de incendios forestales

Es necesario tomar consciencia sobre el entorno que nos rodea y de las repercusiones que pueden desencadenar determinados actos o hechos como los incendios provocados o no. Estos llevan consigo desgracias humanas, pérdidas de vegetación, animales, etc. que quizás sean difíciles de recuperar. Por ello, es un deber de todos proteger los pulmones de la Tierra.

El fuego es un proceso importante del ecosistema que afecta de forma significativa a los medios terrestres, acuáticos y atmosféricos en todo el planeta. A escala local puede quemar la vegetación lo que supone una alteración de la estructura y composición de los suelos (Wyant et al., 1986; McHugh y Kolb 2003) además de la pérdida de los animales de la zona, a escala regional afectan a la calidad y cantidad del rendimiento de agua (Minshall et al., 2001) provocan la aceleración de la erosión de igual modo que la sedimentación desembocando en una miríada de consecuencias para los sistemas acuáticos (Gresswell 1999; Vieira et al., 2010). En ambas escalas, ante esta situación, se localizan contaminantes que impactan con la calidad del aire, tales como el monóxido de carbono y los óxidos de nitrógeno que suponen un riesgo para la salud humana. Estas emisiones, a nivel global, repercuten directa y considerablemente sobre los ciclos atmosféricos, biogeoquímicos y el balance energético del mundo.

La prevención, detección, monitoreo y supresión de incendios forestales son preocupaciones ecológicas, económicas y de seguridad pública clave en muchas partes del mundo. Mediante estas acciones es posible mitigar el impacto económico además de evitar la invasión del fuego en áreas habitadas. Con el fin de alcanzar este propósito es necesario un buen seguimiento y una extinción exitosa siendo de gran importancia la detección temprana del incendio que permite que las actividades de supresión empiecen mientras el fuego aún es pequeño y manejable. La inaccesibilidad de muchos de estos incendios forestales hace que los datos de detección remota sean un recurso importante y ampliamente aplicado para la ciencia y el manejo del fuego (Hardy et al. 1999). Los algoritmos de visión por computador pueden llegar a ser un elemento clave en esta problemática debido a que, análogamente al creciente avance tecnológico referente a nuevas plataformas de sensores, consiguen clasificar el entorno con el propósito de prestar mayor atención a las zonas que así lo requieran.

La aplicación práctica propuesta para los distintos métodos de segmentación semántica, presentados en el capítulo anterior, surge con la intención de hacer frente a la prevención de incendios forestales identificando de forma temprana las áreas más propensas a que estos ocurran de tal forma que es posible producir mapas para los equipos de respuesta a emergencias.

## 3.1 Materiales y Recursos utilizados

### 3.1.1 Descripción de las imágenes

A fin de desarrollar el contenido práctico del presente proyecto se han elegido diversas fotografías aéreas procedentes de la región de Abadín en Lugo (España), cuya área alcanza los 196 $km^2$. Se trata de un entorno con abundante vegetación y frondosos bosques, caracterizado por laderas con grandes robledales, pinares, eucaliptales, acebos, bosques de ribera y extensas praderas. Uno de los bosques más grandes, compuesto por Eucaliptales, tiene una extensión de 31.61 $km^2$ además de la presencia de otros 9 bosques con superficies que rondan los 5 – 8 $km^2$ (Naturaspain, 2021). Su composición hace de este un territorio en el que hay que prestar especial atención con el fin de localizar los puntos más propensos a sufrir un incendio, así como vigilar las zonas verdes próximas a poblaciones o aldeas. De hecho, las imágenes utilizadas en este trabajo proceden de los proyectos que la empresa Dimap desarrolló para la Junta de Galicia en la mencionada zona de Abadín con el fin de determinar zonas de riesgo de cara a la prevención temprana y actuación rápida, en su caso, durante el periodo estival.

Las imágenes tratadas son fotografías tomadas con sensores aerotransportados, es decir, mediante un vuelo fotogramétrico donde un aeroplano toma repetidas capturas con el fin de abarcar toda la superficie. Estas reflejan la representación cónica de la realidad que se ve alterada por las deformaciones del terreno, la perspectiva y la falta de verticalidad de las

fotografías además de las distorsiones características del objetivo de la cámara utilizada. Desde la década de los ochenta y mediados de la siguiente, la fotografía aérea se transforma en ortofotografías, esto es, la corrección de las deformaciones anteriormente comentadas mediante el tratamiento digital de las imágenes aéreas atribuyendo una rigurosidad geométrica equivalente a un mapa cartográfico. La información completa de las instantáneas tomadas del territorio se conserva en este tipo de mapas en contra de la visión selectiva y parcial que ofrece la cartografía, por ello, para el presente estudio, se parte de ortofotografías debido a sus múltiples aplicaciones técnicas. En la figura 3.1 se muestra un ejemplo de este tipo de fotografía.



Figura 3.1: Fotografía aérea de la región de Abadín

Se tratan 20 imágenes a color seleccionadas para el entrenamiento y 18 para la clasificación con una resolución de 501x501x3 píxeles. Cada imagen consiste en un cuboide de tres elementos en la forma [altura, ancho, profundidad] siendo la profundidad el número de canales de la imagen, en el caso presentado su valor equivale a 3 ya que se trata de una imagen RGB. El tamaño de las imágenes a tratar en todo el proceso de entrenamiento y clasificación ha de ser el mismo y este está condicionado con la profundidad del codificador, teniendo que ser múltiplo de $2^{EncoderDepth}$ para cada uno de los métodos propuestos.

La complejidad de los entornos naturales, extensos, variados y dinámicos, en el que se combinan zonas urbanas con densas cubiertas boscosas, hace que este proyecto suponga un desafío para la visión por computador porque, aunque la segmentación semántica trate las imágenes a nivel de píxel, prácticamente no se encontrarán patrones idénticos en el paisaje. Para hacer frente a esta situación se han propuesto tres etiquetas distinguiendo tres tipos de texturas mayormente diferenciadas: zonas con presencia de bosque o grandes extensiones de arboledas (SotoBosque), zonas verdes o campos con cultivos (CamposVerdes) y zonas sin vegetación (CamposSinVegetacion). Este aspecto se tratará en el apartado referente al entrenamiento.

### 3.1.2 Recursos Hardware/Software

Nos encontramos ante un momento en el que la AI ha alcanzado un punto de inflexión crítico, las implicaciones que ello supone repercuten en gran cantidad de aplicaciones existentes. Mediante esta confluencia de factores se consigue que la AI sea más práctica y efectiva siendo

el rendimiento de las redes neuronales en multitud de aplicaciones muy superior a los enfoques logarítmicos tradicionales. Los modelos previamente entrenados y los grandes conjuntos de datos reducen el esfuerzo de desarrollo necesario para la construcción de un modelo de AI, asimismo, los nuevos hardware de cómputo disminuyen en gran medida el impacto de la enorme computación demandada para el funcionamiento de las redes neuronales.

MatLab es una de las herramientas empelada para ello y, debido a su uso a lo largo de la trayectoria universitaria de cualquier ingeniero, resulta familiar e intuitivo, por tal razón, se ha desarrollado el aspecto práctico del trabajo íntegramente con la versión de estudiante de MatLab R2021a. A raíz de la versión lanzada en el 2017 (R2017b), MathWorks incluyó una serie de capacidades en MatLab Y Simulink que permitía el aprovechamiento del aprendizaje profundo para determinadas aplicaciones. En su versión R2018a completó esta iniciativa en la que se reforzaron las competencias. Actualmente, MatLab engloba un flujo de aprendizaje profundo completo partiendo de la recopilación y etiquetado de datos hasta la creación y acceso a modelos, pruebas, implementación e inferencias.

Las imágenes tratadas requieren de un proceso de etiquetado que se facilita dentro de MatLab con la aplicación Image Labeler incluida en Automated Driving Toolbox™, utilizado en este proyecto, que permite el etiquetado de píxeles y regiones de interés para la segmentación semántica ya sea con etiquetas rectangulares, polilíneas, de píxeles y poligonales etiquetando una escena en una imagen o secuencia de imágenes. Debido a que se han empleado imágenes de un entorno natural, se ha decidido usar etiquetas poligonales con el fin de adaptarse libremente a cada región. La figura 3.2 ilustra la interfaz de esta aplicación.

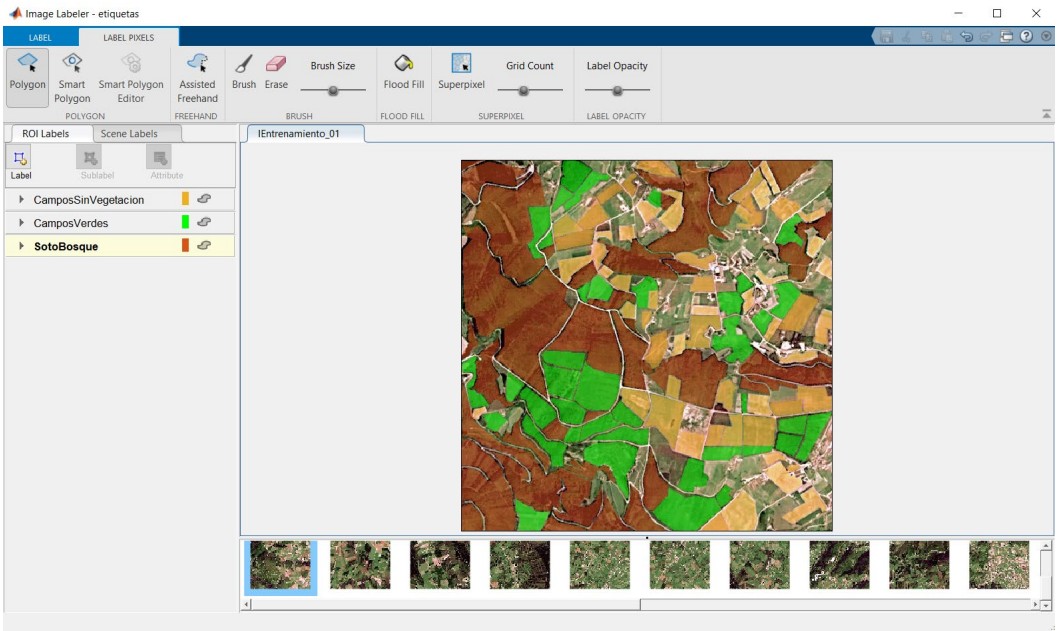



Figura 3.2: Image Labeler

En muchas ocasiones es necesario acceder a modelos previamente entrenados que ya han formado parte de otros trabajos siendo innecesario realizar esta tarea de nuevo, por ello, MatLab permite su importación, como los modelos de CNN AlexNet, VCG-16 y VCG-19 que han sido perfeccionados a lo largo de los años resultando en una clasificación de imágenes extraordinaria. Además de Deep Learning Toolbox™, se han importado los siguientes *toolbox* con el objeto de tratar los modelos presentados en el trabajo:

- Deep Learning Toolbox Model for VGG-16 Network.
- Deep Learning Toolbox Model for Inception-ResNet-v2 Network
- Deep Learning ToolboxTM Model for Xception Network
- Deep Learning Toolbox Model for MobileNet-v2 Network
- Deep Learning Toolbox Model for ResNet-50 Network
- Deep Learning Toolbox Model for ResNet-18 Network

El entrenamiento puede ser muy costoso computacionalmente, debido a esto, MatLab permite apuntar a la GPU o incluso a múltiples clústeres de GPU para su aceleración. De esta manera se optimiza el proceso siendo inherentemente altamente paralelizable. Gracias a esta característica, con la instalación de Parallel Computing Toolbox™ ha sido posible emplear la GPU NVIDIA GeForce GTX 1050 que incorpora el equipo utilizado para la simulación. Por otro lado, en la fase de inferencia también se requieren requisitos computaciones que afectan tanto al rendimiento como a la latencia, aquí MatLab también facilita esta función posibilitando la conversión de los modelos al código NVIDIA CUDA, que beneficiará el rendimiento de estos.

Con el fin de poder ejecutar los distintos métodos de segmentación semántica con MatLab, según las observaciones anteriores, se ha empleado un equipo con las siguientes características:

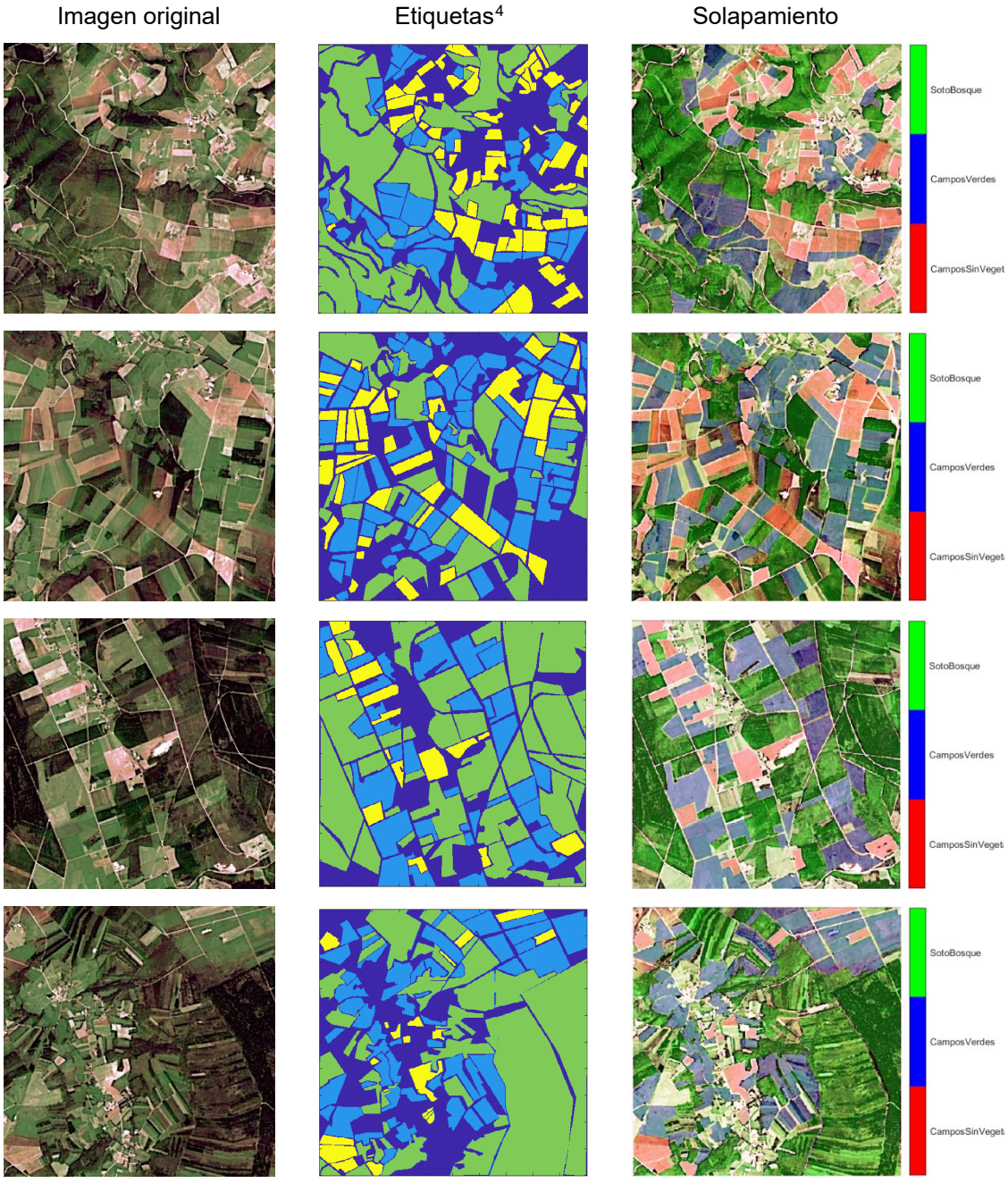
- CPU: Intel Core i7-7700HQ a 2.80Ghz
- GPU: NVIDIA GeForce GTX 1050 GDRR5 4Gb
- RAM: 16Gb

## 3.2 Entrenamiento

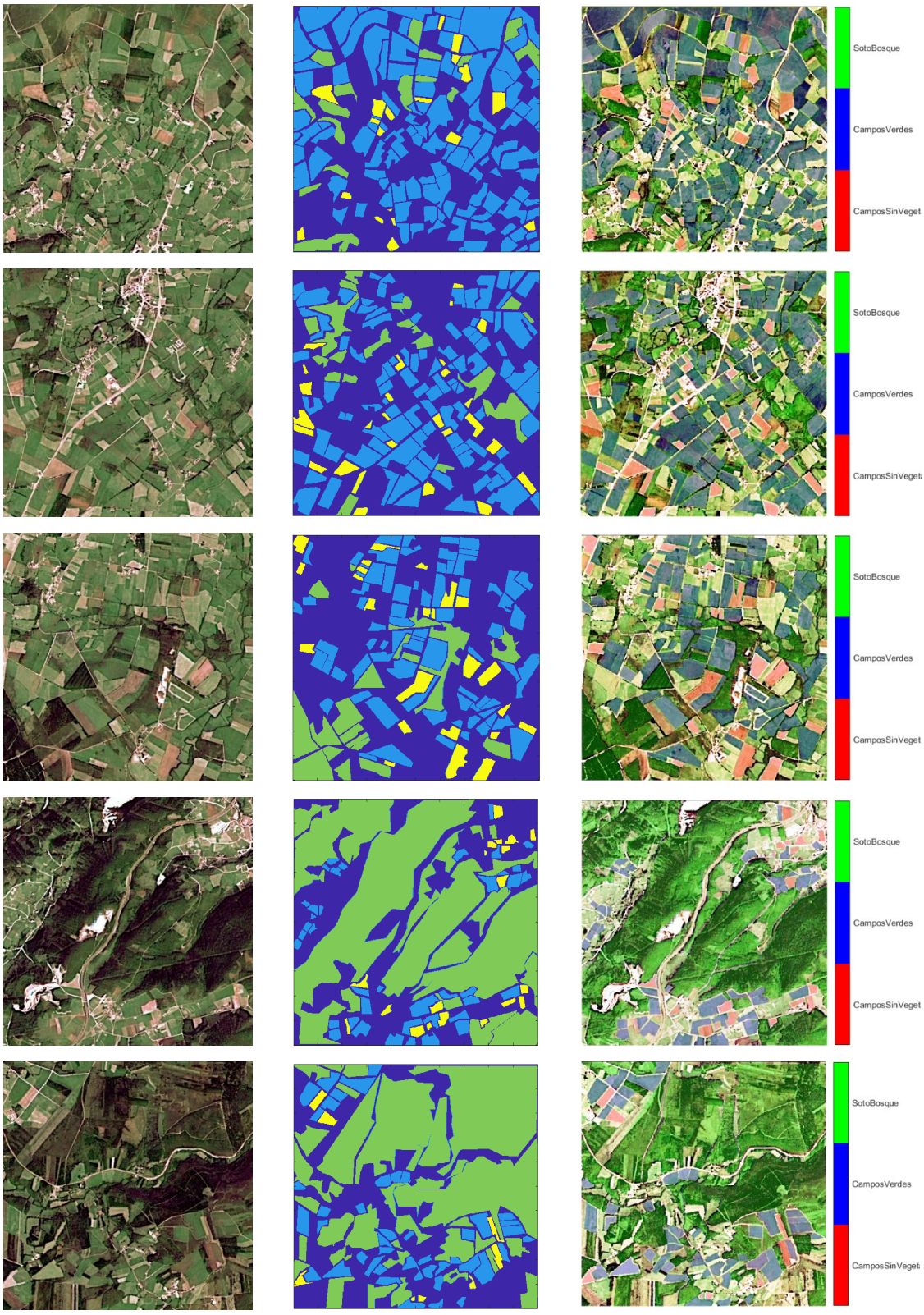
Tal y como se ha comentado, para esta fase se han empleado 20 imágenes. Estas han sido etiquetadas previamente con la aplicación Image Labeler con el fin de establecer un criterio de clasificación a nivel de píxel para cada una de las etiquetas propuestas:

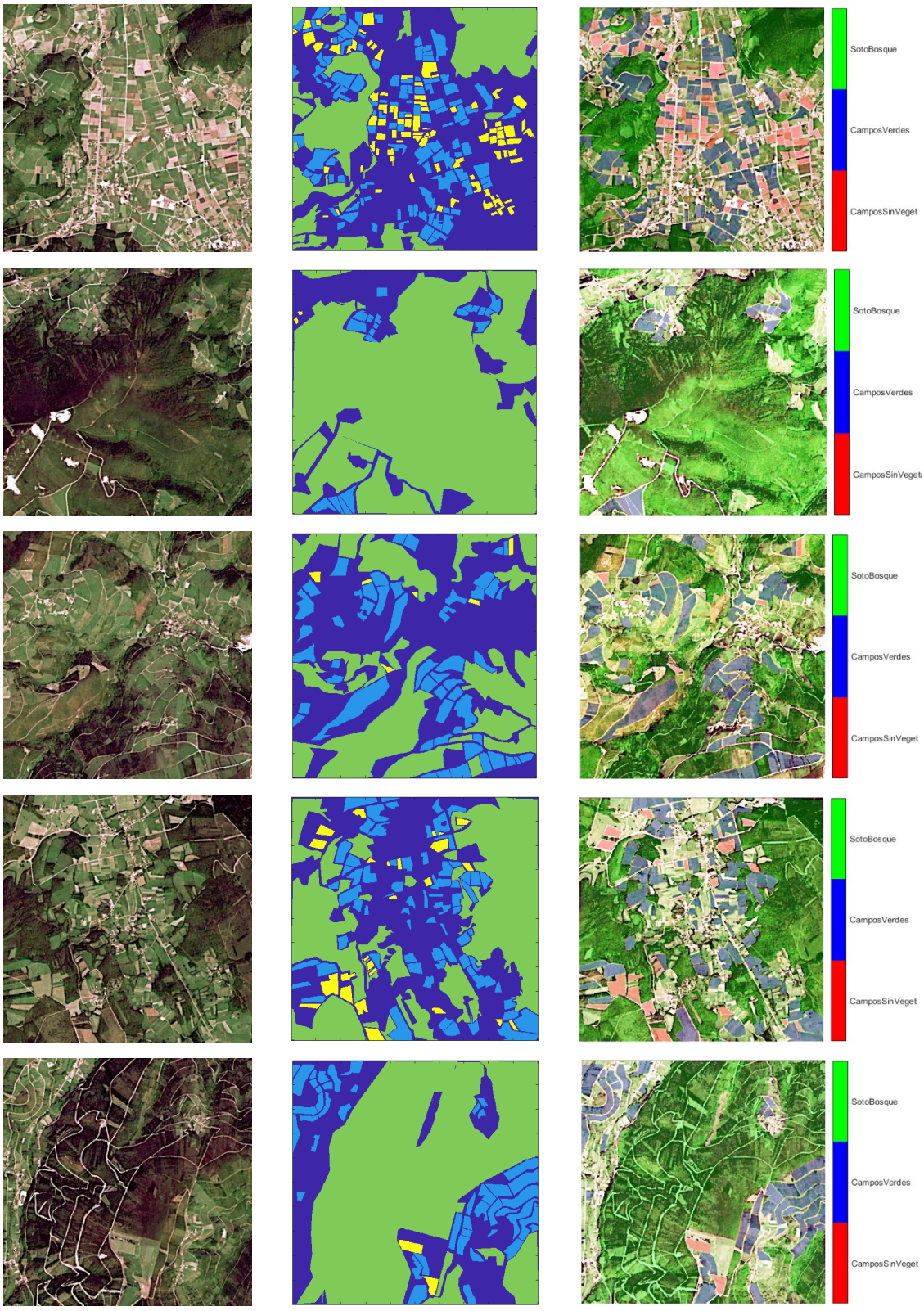
- SotoBosque
- CamposVerdes
- CamposSinVegetación

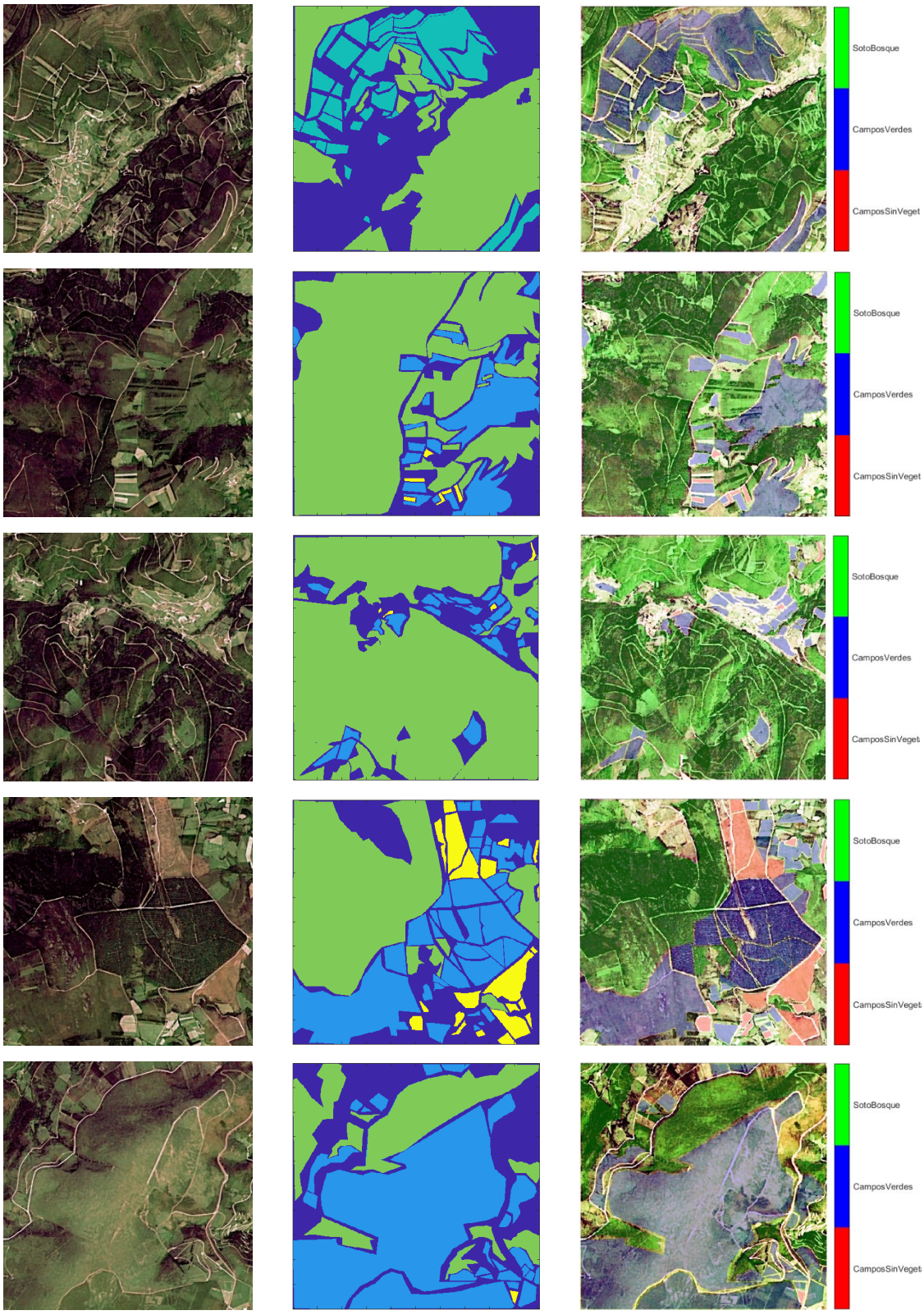
A partir de esta aplicación se exportan las etiquetas en un fichero gTruth.mat que permitirá crear los *datastore* necesarios para proceder con el entrenamiento. Las capturas, mostradas a lo largo de la figura 3.3, son fruto de esta etiquetación donde se observa la imagen original, las etiquetas creadas para ella (gTruth) y el solapamiento de estas con la fotografía:




[4] El azul oscuro hace referencia al fondo no etiquetado.


SotoBosque
CamposVerdes
CamposSinVeget
SotoBosque
CamposVerdes
CamposSinVeget
SotoBosque
CamposVerdes
CamposSinVeget
SotoBosque
CamposVerdes
CamposSinVeget
SotoBosque
CamposVerdes
CamposSinVeget


SotoBosque
CamposVerdes
CamposSinVeget


SotoBosque
CamposVerdes
CamposSinVeget
SotoBosque
CamposVerdes
CamposSinVeget
SotoBosque
CamposVerdes
CamposSinVeget
SotoBosque
CamposVerdes
CamposSinVeget
SotoBosque
CamposVerdes
CamposSinVeget

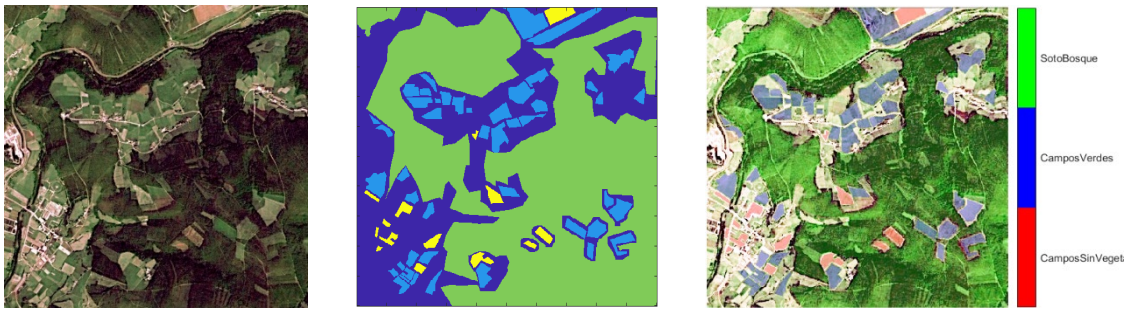



Figura 3.3: Etiquetado de las fotografías

Tras la creación del *datastore* se preparan los conjuntos de entrenamiento, validación y prueba sobre una arquitectura CNN predefinida. Es posible ajustar determinados parámetros globales en cada uno de los entrenamientos gracias a la función *trainingOptions*. Para el presente estudio se han variado los argumentos *solverName*, *InitialLearnRate*, *LearnRateSchedule*, *L2Regularization*, *MaxEpochs*, *MiniBatchSize* en cada uno de los modelos ya que se considera que pueden repercutir notablemente en el proceso de entrenamiento.

Cada optimizador funciona mejor según qué contexto, por ello cabe destacar la especificación del algoritmo del optimizador mediante *solverName*, empleando ADAM o SGDM. Como se comentó en el capítulo anterior, ambos pretenden minimizar la pérdida según su actualización característica de los parámetros de la red de tal forma que el entrenamiento llegue a mejorar. En esta actualización de parámetros se emplea un subconjunto de los datos en cada uno de los pasos correspondiendo a lo que se conoce como mini-lote (MiniBatch) cuyo tamaño es posible concretarlo utilizando el argumento de *trainingOptions*, *MiniBatchSize,* aunque, como se verá, su asignación estará condicionada a la memoria de la GPU. Se conoce como iteración a cada actualización del parámetro y como *epoch* a la pasada completa a través de todo el conjunto de datos. Para limitar el número máximo de *epoch* se utiliza el argumento *MaxEpochs* que por defecto se establece en 30 aunque para este estudio se ha valorado duplicarlo debido a que el número de imágenes para el entrenamiento es muy bajo. De esta forma se pretende afinar los parámetros para mejorar el aprendizaje.

Otra modificación a tener en cuenta se trata de la especificación de la tasa de aprendizaje global a través del argumento *InitialLearnRate* que por defecto se usará durante todo el proceso. Es posible reducirla gradualmente durante el entrenamiento estableciendo *piecewise* dentro de *LearnRateSchedule*, de tal manera que, en algunos experimentos de este estudio, se multiplica la tasa de aprendizaje inicial por un factor de 0.3 cada 10 *epoch*. Estos valores, *LearnRateDropFactor* y *LearnRateDropPeriod,* se han mantenido fijos, pero sí se ha valorado tanto la disminución gradual de la tasa de aprendizaje inicial como mantenerla fija.

En los siguientes subapartados se analizarán los resultados obtenidos en relación con los métodos de segmentación semántica estudiados para este trabajo.

### 3.2.1 DeepLabv3

Se emplean diversas CNN preentrenadas con el objeto de comparar el resultado. Estas son:

- ResNet18
- ResNet50
- MobileNetv2
- Xception
- InceptionResnetV2

La siguiente tabla muestra los valores obtenidos en cuanto al porcentaje de validación y los valores de pérdidas en función de los parámetros de aprendizaje dados y según los modelos de redes indicados para cada una de las situaciones de estudio especificados:

| | solverName | InitialLearnRate | LearnRateSchedule | L2Regularization | MaxEpochs | MiniBatchSize | LearnRateDropPeriod | LearnRateDropFactor | Validación (%) | Pérdidas |
|---|---|---|---|---|---|---|---|---|---|---|
| ResNet18 | SGDM | 1e-1 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 84.29 | 0.4634 |
| ResNet18 | SGDM | 1e-3 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 82.96 | 0.7203 |
| ResNet18 | SGDM | 1e-5 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 41.48 | 1.1855 |
| ResNet18 | SGDM | 1e-1 | none | 0.1 | 30 | 4 | 10 | 0.3 | 77.34 | 0.8053 |
| ResNet18 | SGDM | 1e-3 | none | 0.1 | 30 | 4 | 10 | 0.3 | 84.03 | 0.6386 |
| ResNet18 | SGDM | 1e-5 | none | 0.1 | 30 | 4 | 10 | 0.3 | 49.75 | 1.0969 |
| ResNet18 | SGDM | 1e-1 | piecewise | 0.1 | 60 | 4 | 10 | 0.3 | 80.54 | 0.5633 |
| ResNet18 | SGDM | 1e-3 | piecewise | 0.1 | 60 | 4 | 10 | 0.3 | 82.89 | 0.7136 |
| ResNet18 | SGDM | 1e-5 | piecewise | 0.1 | 60 | 4 | 10 | 0.3 | 41.91 | 1.1797 |
| ResNet18 | SGDM | 1e-1 | none | 0.1 | 60 | 4 | 10 | 0.3 | 60.07 | 0.9961 |
| ResNet18 | SGDM | 1e-3 | none | 0.1 | 60 | 4 | 10 | 0.3 | 84.13 | 0.6508 |
| ResNet18 | SGDM | 1e-5 | none | 0.1 | 60 | 4 | 10 | 0.3 | 60.33 | 1.0129 |
| ResNet18 | SGDM | 1e-1 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 81.81 | 0.8105 |
| ResNet18 | SGDM | 1e-3 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 82.36 | 0.8368 |
| ResNet18 | SGDM | 1e-5 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 41.46 | 1.1869 |
| ResNet18 | SGDM | 1e-1 | none | 0.005 | 30 | 4 | 10 | 0.3 | 83.52 | 0.4235 |
| ResNet18 | SGDM | 1e-3 | none | 0.005 | 30 | 4 | 10 | 0.3 | 83.11 | 0.8128 |
| ResNet18 | SGDM | 1e-5 | none | 0.005 | 30 | 4 | 10 | 0.3 | 49.69 | 1.0990 |
| ResNet18 | SGDM | 1e-1 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 84.53 | 0.5016 |
| ResNet18 | SGDM | 1e-3 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 83.85 | 0.6758 |
| ResNet18 | SGDM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 84.27 | 0.7541 |
| ResNet18 | ADAM | 1e-1 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 62.57 | 0.6788 |
| ResNet18 | ADAM | 1e-3 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 85.80 | 0.5450 |
| ResNet18 | ADAM | 1e-5 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 76.31 | 0.7692 |
| ResNet18 | ADAM | 1e-1 | none | 0.005 | 30 | 4 | 10 | 0.3 | 60.77 | 0.7351 |
| ResNet18 | ADAM | 1e-3 | none | 0.005 | 30 | 4 | 10 | 0.3 | 86.24 | 0.5873 |
| ResNet18 | ADAM | 1e-5 | none | 0.005 | 30 | 4 | 10 | 0.3 | 80.84 | 0.7644 |
| ResNet50 | SGDM | 1e-1 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 84.68 | 0.5075 |
| ResNet50 | SGDM | 1e-3 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 83.22 | 0.6933 |
| ResNet50 | SGDM | 1e-5 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 41.34 | 1.1056 |
| ResNet50 | SGDM | 1e-1 | none | 0.1 | 30 | 4 | 10 | 0.3 | 56.24 | 0.9999 |
| ResNet50 | SGDM | 1e-3 | none | 0.1 | 30 | 4 | 10 | 0.3 | 83.76 | 0.6725 |
| ResNet50 | SGDM | 1e-5 | none | 0.1 | 30 | 4 | 10 | 0.3 | 52.99 | 0.9250 |
| ResNet50 | SGDM | 1e-1 | piecewise | 0.1 | 60 | 4 | 10 | 0.3 | 85.02 | 0.4884 |
| ResNet50 | SGDM | 1e-3 | piecewise | 0.1 | 60 | 4 | 10 | 0.3 | 83.08 | 0.6892 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| ResNet50 | SGDM | 1e-5 | piecewise | 0.1 | 60 | 4 | 10 | 0.3 | 41.82 | 1.0936 |
| ResNet50 | SGDM | 1e-1 | none | 0.1 | 60 | 4 | 10 | 0.3 | 62.65 | 0.8351 |
| ResNet50 | SGDM | 1e-3 | none | 0.1 | 60 | 4 | 10 | 0.3 | 84.07 | 0.6843 |
| ResNet50 | SGDM | 1e-5 | none | 0.1 | 60 | 4 | 10 | 0.3 | 68.74 | 0.7788 |
| ResNet50 | SGDM | 1e-1 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 81.10 | 0.5349 |
| ResNet50 | SGDM | 1e-3 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 82.70 | 0.8046 |
| ResNet50 | SGDM | 1e-5 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 41.32 | 1.1072 |
| ResNet50 | SGDM | 1e-1 | none | 0.005 | 30 | 4 | 10 | 0.3 | 81.12 | 0.6595 |
| ResNet50 | SGDM | 1e-3 | none | 0.005 | 30 | 4 | 10 | 0.3 | 82.80 | 0.8287 |
| ResNet50 | SGDM | 1e-5 | none | 0.005 | 30 | 4 | 10 | 0.3 | 52.89 | 0.9266 |
| ResNet50 | ADAM | 1e-1 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 74.93 | 0.5759 |
| ResNet50 | ADAM | 1e-3 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 82.46 | 0.8586 |
| ResNet50 | ADAM | 1e-5 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 78.31 | 0.7594 |
| ResNet50 | ADAM | 1e-1 | none | 0.005 | 30 | 4 | 10 | 0.3 | 61.22 | 0.8324 |
| ResNet50 | ADAM | 1e-3 | none | 0.005 | 30 | 4 | 10 | 0.3 | 85.07 | 0.6588 |
| ResNet50 | ADAM | 1e-5 | none | 0.005 | 30 | 4 | 10 | 0.3 | 80.79 | 0.7675 |
| MobileNetv2 | SGDM | 1e-1 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 84.86 | 0.5833 |
| MobileNetv2 | SGDM | 1e-3 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 73.85 | 1.0148 |
| MobileNetv2 | SGDM | 1e-5 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 25.93 | 1.1520 |
| MobileNetv2 | SGDM | 1e-1 | none | 0.1 | 30 | 4 | 10 | 0.3 | 38.06 | 0.6639 |
| MobileNetv2 | SGDM | 1e-3 | none | 0.1 | 30 | 4 | 10 | 0.3 | 80.00 | 0.8545 |
| MobileNetv2 | SGDM | 1e-5 | none | 0.1 | 30 | 4 | 10 | 0.3 | 33.18 | 1.1580 |
| MobileNetv2 | SGDM | 1e-1 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 82.05 | 0.6471 |
| MobileNetv2 | SGDM | 1e-3 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 72.24 | 1.1622 |
| MobileNetv2 | SGDM | 1e-5 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 25.91 | 1.1530 |
| MobileNetv2 | SGDM | 1e-1 | none | 0.005 | 30 | 4 | 10 | 0.3 | 80.15 | 0.8705 |
| MobileNetv2 | SGDM | 1e-3 | none | 0.005 | 30 | 4 | 10 | 0.3 | 76.99 | 1.0395 |
| MobileNetv2 | SGDM | 1e-5 | none | 0.005 | 30 | 4 | 10 | 0.3 | 33.13 | 1.1597 |
| MobileNetv2 | ADAM | 1e-1 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 33.61 | 0.6426 |
| MobileNetv2 | ADAM | 1e-3 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 83.26 | 0.8544 |
| MobileNetv2 | ADAM | 1e-5 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 46.74 | 0.9557 |
| MobileNetv2 | ADAM | 1e-1 | none | 0.005 | 30 | 4 | 10 | 0.3 | 26.64 | 1.0865 |
| MobileNetv2 | ADAM | 1e-3 | none | 0.005 | 30 | 4 | 10 | 0.3 | 83.26 | 0.9387 |
| MobileNetv2 | ADAM | 1e-5 | none | 0.005 | 30 | 4 | 10 | 0.3 | 66.21 | 0.9314 |
| Xception | SGDM | 1e-1 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 82.45 | 0.6703 |
| Xception | SGDM | 1e-3 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 84.47 | 0.6021 |
| Xception | SGDM | 1e-5 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 46.13 | 1.1933 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Xception | SGDM | 1e-1 | none | 0.1 | 30 | 4 | 10 | 0.3 | 73.11 | 0.9639 |
| Xception | SGDM | 1e-3 | none | 0.1 | 30 | 4 | 10 | 0.3 | 84.72 | 0.6053 |
| Xception | SGDM | 1e-5 | none | 0.1 | 30 | 4 | 10 | 0.3 | 51.01 | 1.1398 |
| Xception | SGDM | 1e-1 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 76.36 | 0.5964 |
| Xception | SGDM | 1e-3 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 83.66 | 0.6610 |
| Xception | SGDM | 1e-5 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 46.12 | 1.1946 |
| Xception | SGDM | 1e-1 | none | 0.005 | 30 | 4 | 10 | 0.3 | 81.57 | 0.8579 |
| Xception | SGDM | 1e-3 | none | 0.005 | 30 | 4 | 10 | 0.3 | 84.20 | 0.6816 |
| Xception | SGDM | 1e-5 | none | 0.005 | 30 | 4 | 10 | 0.3 | 50.96 | 1.1419 |
| Xception | ADAM | 1e-1 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 84.21 | 0.5898 |
| Xception | ADAM | 1e-3 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 85.84 | 0.4414 |
| Xception | ADAM | 1e-5 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 75.84 | 0.6902 |
| Xception | ADAM | 1e-1 | none | 0.005 | 30 | 4 | 10 | 0.3 | 53.94 | 1.1050 |
| Xception | ADAM | 1e-3 | none | 0.005 | 30 | 4 | 10 | 0.3 | 86.03 | 0.4129 |
| Xception | ADAM | 1e-5 | none | 0.005 | 30 | 4 | 10 | 0.3 | 77.14 | 0.6878 |
| InceptionResnetV2 | SGDM | 1e-1 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | *Out of memory* | - |
| InceptionResnetV2 | SGDM | 1e-1 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 63.24 | 0.7343 |
| InceptionResnetV2 | SGDM | 1e-3 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 81.45 | 0.8678 |
| InceptionResnetV2 | SGDM | 1e-5 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 62.46 | 0.8926 |
| InceptionResnetV2 | SGDM | 1e-1 | none | 0.1 | 30 | 1 | 10 | 0.3 | 53.67 | 1.1135 |
| InceptionResnetV2 | SGDM | 1e-3 | none | 0.1 | 30 | 1 | 10 | 0.3 | 82.87 | 0.5421 |
| InceptionResnetV2 | SGDM | 1e-5 | none | 0.1 | 30 | 1 | 10 | 0.3 | 72.25 | 0.7263 |
| InceptionResnetV2 | SGDM | 1e-1 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 77.53 | 0.6388 |
| InceptionResnetV2 | SGDM | 1e-3 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 82.63 | 0.6996 |
| InceptionResnetV2 | SGDM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 62.56 | 0.8866 |
| InceptionResnetV2 | SGDM | 1e-1 | none | 0.005 | 30 | 1 | 10 | 0.3 | 81.89 | 0.6097 |
| InceptionResnetV2 | SGDM | 1e-3 | none | 0.005 | 30 | 1 | 10 | 0.3 | 81.41 | 0.8679 |
| InceptionResnetV2 | SGDM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 71.38 | 0.7596 |
| InceptionResnetV2 | ADAM | 1e-1 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 11.51 | 0.8033 |
| InceptionResnetV2 | ADAM | 1e-3 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 86.34 | 0.4590 |
| InceptionResnetV2 | ADAM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 82.31 | 0.5871 |
| InceptionResnetV2 | ADAM | 1e-1 | none | 0.005 | 30 | 1 | 10 | 0.3 | 10.91 | 1.0325 |
| InceptionResnetV2 | ADAM | 1e-3 | none | 0.005 | 30 | 1 | 10 | 0.3 | 84.76 | 0.4873 |
| InceptionResnetV2 | ADAM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 83.07 | 0.6075 |

Tabla 3.1: Resultados del entrenamiento con DeepLabv3

DeepLabv3 ha ocupado la mayor parte de investigación del trabajo debido a que se trata del modelo en el que se han obtenido mejores resultados.

Como se observa, la tasa inicial de aprendizaje se ha variado entre los valores 0.1, 0.001 y 0.00001 combinándola con una reducción gradual o no. En cualquier caso, se demuestra que, ante una tasa inicial menor que 0.001, el porcentaje obtenido en la validación decae y los resultados no son buenos lo que conlleva a una clasificación posterior errónea. Si la tasa inicial aumenta de 0.01 a 0.1 el resultado alcanzado no es destacable ante esta diferencia.

Aunque era posible limitar los datos de validación mediante el argumento *ValidationPatience* se ha querido optar por un proceso completo repitiendo esta característica en todos los entrenamientos ya que, en alguna ocasión el porcentaje de validación ha mejorado, aunque aparentemente al inicio tuviera una respuesta plana. Otra respuesta negativa durante el procedimiento se ha dado con la variación del número máximo de *epoch* duplicando el valor por defecto de 30 a 60. Aunque no ha afectado a todas las combinaciones de estudio, mayormente se ha dado la situación en que la red se ha visto sobreajustada, es decir, en lugar de que la red aprenda características generales que posibilitan predicciones precisas para nuevos datos, esta solamente se limita a memorizar los datos de entrenamiento. Para reducir este sobreajuste se ha variado el valor del argumento de regularización *L2Regularization* sin éxito. En la figura 3.4 se muestra una captura del gráfico obtenido durante el entrenamiento de una situación en las peores condiciones donde la precisión (accuracy) se sitúa en torno al 40%, muy lejos del 100% deseable, y donde la pérdida (loss) alrededor de 1 y también muy lejos del valor deseado, en este caso 0.

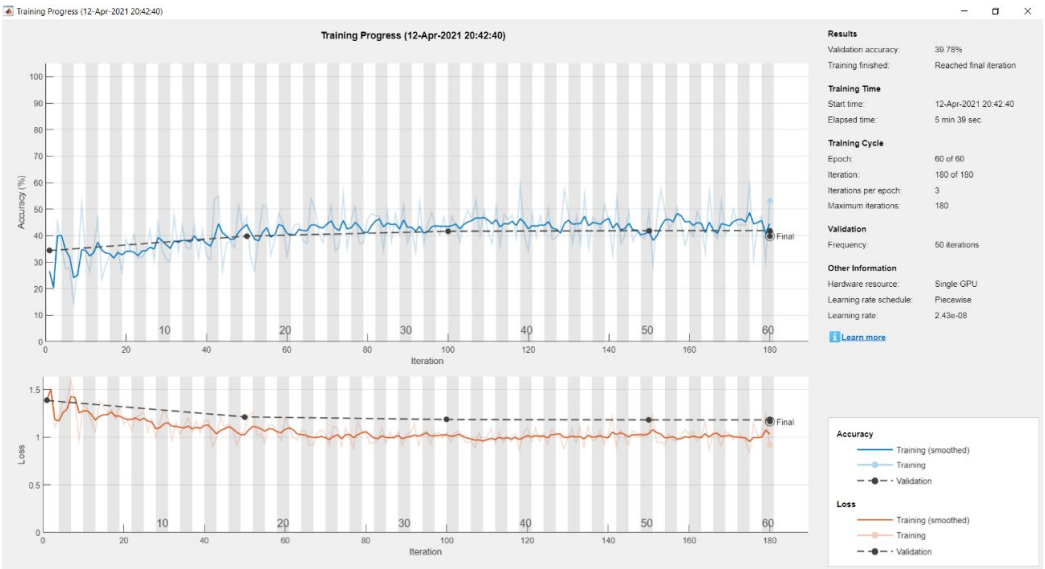



Figura 3.4: Entrenamiento desfavorable en DeepLabv3

Con ambos optimizadores, ADAM y SGDM se han obtenido resultados similares, validación superior al 85% y las pérdidas inferiores al 0.5, pero ADAM es el que mejor se ha comportado

ante cualquier situación consiguiendo un número mayor de resultados favorables. La figura 3.5 representa un ejemplo de un proceso de entrenamiento con un resultado favorable dentro de las limitaciones de la cantidad de imágenes utilizadas.

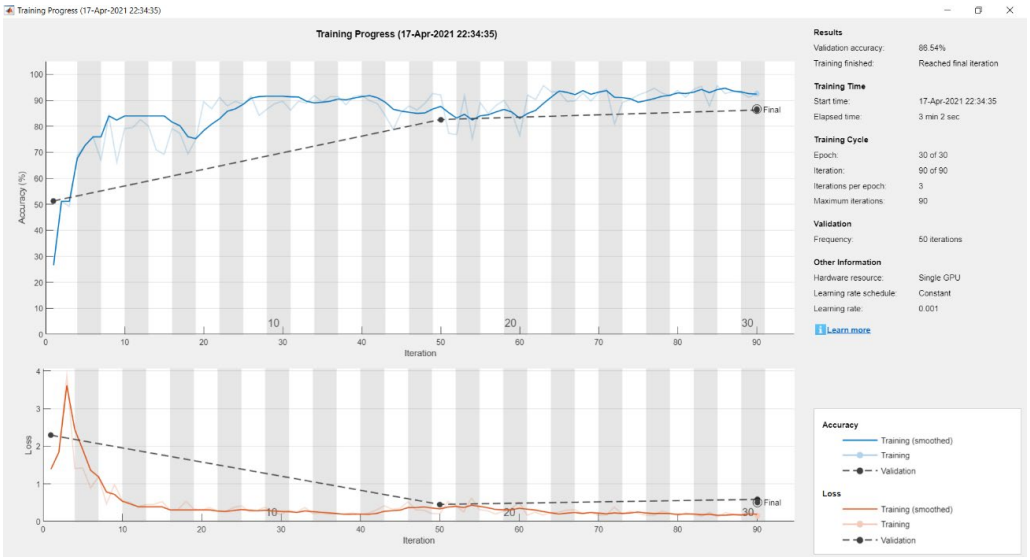



Figura 3.5: Entrenamiento favorable en DeepLabv3

Como se ha comentado a lo largo de este capítulo, MatLab permite seleccionar el recurso de hardware para el entrenamiento si hay una GPU compatible disponible, en el caso de que no se disponga de una, trabajará con la CPU y el proceso será mucho más lento. El argumento *MiniBatchSize* se ha ajustado según la GPU empleada teniendo que disminuirlo a 4 o 1 porque se obtenía el error “fuera de memoria”, esto se debe a que MatLab retiene la memoria de la GPU una vez asignada siendo un gran cuello de botella para el rendimiento. No se observa una diferencia notable en los resultados.

De los 106 entrenamientos que se han llevado a cabo para el estudio de DeepLabv3, destacan los mostrados en la tabla 3.2:

| | Solver Name | InitialLearn Rate | InitialLearn Rate | LearnRate Schedule | Max Epochs | MiniBatch Size | Validación (%) | Pérdidas |
|---|---|---|---|---|---|---|---|---|
| ResNet18 | ADAM | piecewise | 1e-3 | 0.005 | 30 | 4 | 85.80 | 0.5450 |
| ResNet18 | ADAM | none | 1e-3 | 0.005 | 30 | 4 | 86.24 | 0.5873 |
| ResNet50 | SGDM | piecewise | 1e-1 | 0.1 | 60 | 4 | 85.02 | 0.4884 |
| Xception | ADAM | piecewise | 1e-3 | 0.005 | 30 | 4 | 85.84 | 0.4414 |
| Xception | ADAM | none | 1e-3 | 0.005 | 30 | 4 | 86.03 | 0.4129 |
| Inception-resnetv2 | ADAM | piecewise | 1e-3 | 0.005 | 30 | 1 | 86.34 | 0.4590 |

Tabla 3.2: Mejores resultados del entrenamiento con DeepLabv3

### 3.2.2 SegNet

Del mismo modo que con el modelo anterior, se han llevado a cabo varios entrenamientos para el modelo de segmentación semántica SegNet. En este caso no se emplean CNN preentrenadas como podría ser VGG-16 o VGG-19, directamente se crea la red con una profundidad de codificador igual a 4, por lo que se establecen las capas de SegNet según la medida de la imagen, el número de clases y la profundidad del codificador que define este modelo. Esta profundidad se refiere al número de veces que la imagen de entrada se submuestrea mientras se realiza el procesamiento, figura 2.11.

En la tabla 3.4. se muestran todas las soluciones alcanzadas. A diferencia de DeepLabv3, no se ha podido ejecutar ningún entrenamiento con un *MiniBatchSize* superior a 1 debido a que consume más memoria, aunque, como se ha comentado antes, la variación de este parámetro no supone un gran cambio en el resultado. Sin embargo, es destacable la variación del argumento *L2Regularization* de 0.1 a 0.005. Para cualquier caso en el que este sea 0.1 el entrenamiento no es propicio mientras que se consiguen porcentajes mayormente aceptables cuando se establece en 0.005 por lo que la red queda mejor ajustada. La figura 3.6 muestra el proceso para un valor de 0.1 de *L2Regularization.* Se observa cómo la precisión media se sitúa en torno al 60% y la pérdida alrededor del valor 1, no siendo, en ningún caso, valores óptimos.

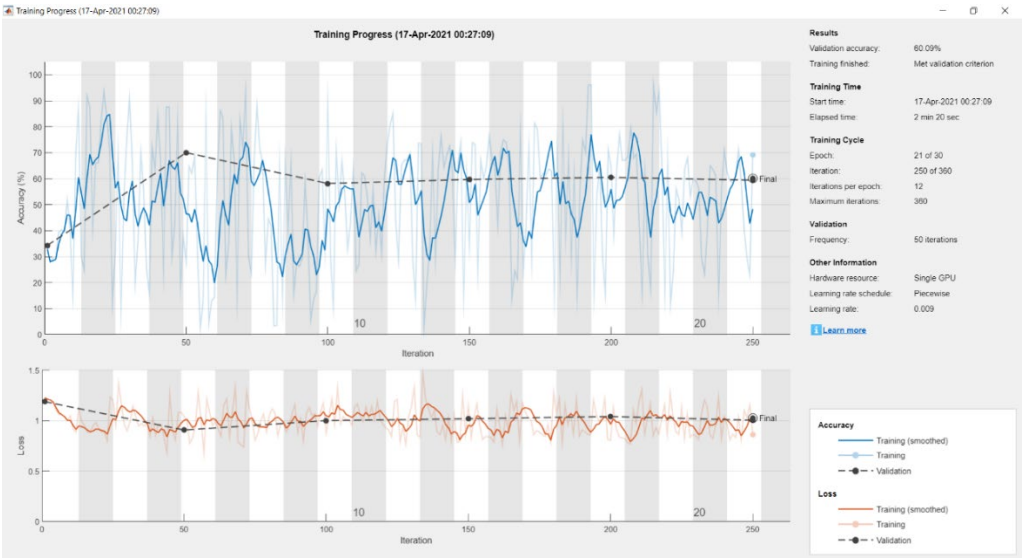



Figura 3.6: Resultado desfavorable con L2Regularization igual a 0.1

A continuación, en la figura 3.7 se presenta el resultado con los mismos parámetros que en el entrenamiento de la figura 3.6 a diferencia de *L2Regularization* igual a 0.005. Obsérvese, cómo la precisión ha conseguido un valor promedio aceptable, superando el 85%, mientras la pérdida queda un poco lejos del valor óptimo.

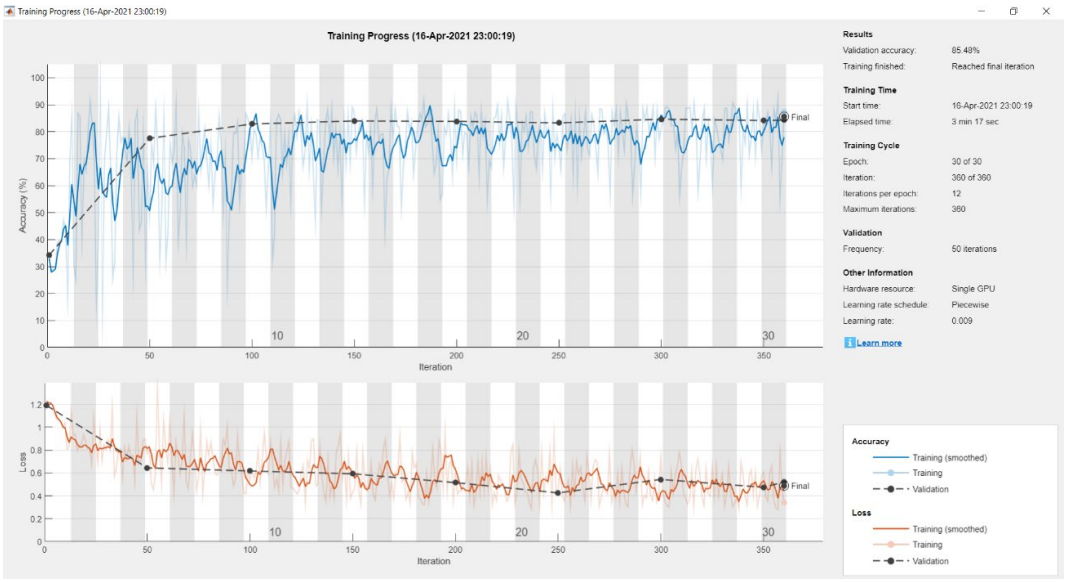



Figura 3.7: Resultado favorable con L2Regularization igual a 0.005

Por otro lado, también cabe mencionar que, ante la variación de la tasa inicial entre los valores 0.1, 0.001 y 0.00001 combinándola con una reducción gradual o manteniéndolos fijos, se comprueba que frente al valor 0.00001 ningún resultado es adecuado mientras que, para 0.1 y 0.001 se consiguen algunos porcentajes por encima del 80% en la validación y unas pérdidas aceptables. Respecto a los optimizadores, ambos, ADAM y SGDM responden de la misma forma, pero, con un *InitialLearnRate* de 0.1 SGDM ofrece una mejor respuesta en cambio para un *InitialLearnRate* de 0.001 es con ADAM con el que se consiguen resultados más relevantes.

En la siguiente tabla, 3.3, se extraen los entrenamientos con mejor respuesta:

| | Solver Name | LearnRate Schedule | InitialLearn Rate | L2Regulari zation | Max Epochs | MiniBatch Size | Validación (%) | Pérdidas |
|---|---|---|---|---|---|---|---|---|
| SegNet | SGDM | piecewise | 1e-1 | 0.005 | 30 | 1 | 84.26 | 0.5209 |
| SegNet | SGDM | none | 1e-1 | 0.005 | 30 | 1 | 83.04 | 0.5286 |
| SegNet | ADAM | piecewise | 1e-3 | 0.005 | 30 | 1 | 84.91 | 0.5003 |
| SegNet | ADAM | none | 1e-3 | 0.005 | 30 | 1 | 84.46 | 0.5480 |

Tabla 3.3: Mejores resultados del entrenamiento con SegNet

A continuación, en la tabla 3.4, se exponen todos los entrenamientos realizados para este modelo de segmentación semántica donde se observa, sobre todo, la diferencia comentada respecto a la variación de la regularización y la tasa inicial.

| | solverName | InitialLearnRate | LearnRateSchedule | L2Regularization | MaxEpochs | MiniBatchSize | LearnRateDropPeriod | LearnRateDropFactor | Validación (%) | Pérdidas |
|---|---|---|---|---|---|---|---|---|---|---|
| SegNet | SGDM | 1e-1 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | *Out of memory* | - |
| SegNet | SGDM | 1e-1 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 59.40 | 1.0034 |
| SegNet | SGDM | 1e-3 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 57.59 | 0.8196 |
| SegNet | SGDM | 1e-5 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 33.71 | 1.2171 |
| SegNet | SGDM | 1e-1 | none | 0.1 | 30 | 1 | 10 | 0.3 | 53.67 | 1.1031 |
| SegNet | SGDM | 1e-3 | none | 0.1 | 30 | 1 | 10 | 0.3 | 63.30 | 0.7742 |
| SegNet | SGDM | 1e-5 | none | 0.1 | 30 | 1 | 10 | 0.3 | 33.99 | 1.2123 |
| SegNet | SGDM | 1e-1 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 84.26 | 0.5209 |
| SegNet | SGDM | 1e-3 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 56.58 | 0.8128 |
| SegNet | SGDM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 33.71 | 1.2175 |
| SegNet | SGDM | 1e-1 | none | 0.005 | 30 | 1 | 10 | 0.3 | 83.04 | 0.5286 |
| SegNet | SGDM | 1e-3 | none | 0.005 | 30 | 1 | 10 | 0.3 | 62.42 | 0.7492 |
| SegNet | SGDM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 33.98 | 1.2131 |
| SegNet | ADAM | 1e-1 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 75.96 | 1.0396 |
| SegNet | ADAM | 1e-3 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 84.91 | 0.5003 |
| SegNet | ADAM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 51.62 | 1.0186 |
| SegNet | ADAM | 1e-1 | none | 0.005 | 30 | 1 | 10 | 0.3 | 32.11 | 1.2133 |
| SegNet | ADAM | 1e-3 | none | 0.005 | 30 | 1 | 10 | 0.3 | 84.46 | 0.5480 |
| SegNet | ADAM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 57.91 | 0.8160 |

Tabla 3.4: Resultados del entrenamiento con SegNet

### 3.2.3 UNet

Con UNet nos encontramos ante la misma situación que con SegNet. No se parte de una CNN preentrenada sino que se emplea una profundidad de codificador, *encoderDepth*, igual a 3 que se tendrá en cuenta para establecer la arquitectura de la red además del tamaño de la imagen y las clases.

En este caso ante cualquier variación de cualquier argumento, el resultado es contraproducente conduciendo a una clasificación errónea. Dado el mal rendimiento de este modelo para las imágenes de estudio, en algunos de los entrenamientos se obtienen valores NaN, figura 3.8. Esto se debe a la pérdida de mini-lotes durante el proceso causada probablemente por una elevada razón de aprendizaje. Aunque se ha disminuido dicho valor, el resultado sigue siendo negativo, figura 3.9, en el sentido de la obtención de malos resultados tanto en lo que respecta a la precisión como a la pérdida.

| Epoch | Iteration | Time Elapsed (hh:mm:ss) | Mini-batch Accuracy | Validation Accuracy | Mini-batch Loss | Validation Loss | Base Learning Rate |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 00:00:05 | 29.62% | 32.14% | 11.2731 | 10.6621 | 0.1000 |
| 1 | 2 | 00:00:07 | 15.14% | | 11.9813 | | 0.1000 |
| 1 | 4 | 00:00:09 | 2.73% | | NaN | | 0.1000 |
| 1 | 6 | 00:00:11 | 5.07% | | NaN | | 0.1000 |
| 1 | 8 | 00:00:13 | 20.43% | | NaN | | 0.1000 |
| 1 | 10 | 00:00:15 | 0.03% | | NaN | | 0.1000 |
| 1 | 12 | 00:00:16 | 5.61% | | NaN | | 0.1000 |

Figura 3.8: Obtención de NaN durante el entrenamiento

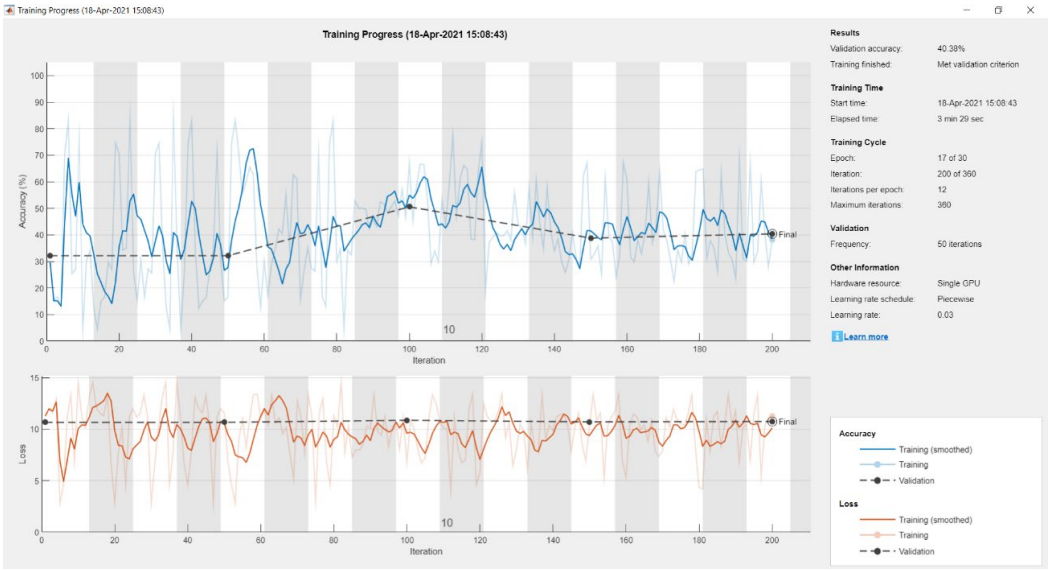



*Figura 3.9: Resultado desfavorable obtenido con UNet*

Observando la tabla 3.5, es posible concluir que, con una cantidad de imágenes pequeña para el entrenamiento y con los mismos parámetros en las opciones de entrenamiento que los empleados para los dos modelos anteriores, no es viable su uso para alcanzar el fin de este apartado práctico: realizar una clasificación exitosa con el objeto de poder elaborar mapas para los equipos de respuesta a emergencias ante incendios.

| | solverName | InitialLearnRate | LearnRateSchedule | L2Regularization | MaxEpochs | MiniBatchSize | LearnRateDropPeriod | LearnRateDropFactor | Validación (%) | Pérdidas |
|---|---|---|---|---|---|---|---|---|---|---|
| UNet | sqgm | 1e-1 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | *Out of memory* | - |
| UNet | SGDM | 1e-1 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 6.33 | NaN |
| UNet | SGDM | 1e-3 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 60.59 | 1.1067 |
| UNet | SGDM | 1e-5 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 40.35 | 3.4046 |
| UNet | SGDM | 1e-1 | none | 0.1 | 30 | 1 | 10 | 0.3 | 6.33 | NaN |
| UNet | SGDM | 1e-3 | none | 0.1 | 30 | 1 | 10 | 0.3 | 46.67 | 1.1348 |
| UNet | SGDM | 1e-5 | none | 0.1 | 30 | 1 | 10 | 0.3 | 40.87 | 2.5699 |
| UNet | SGDM | 1e-1 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 61.53 | 1.1460 |
| UNet | SGDM | 1e-3 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 47.91 | 1.1407 |
| UNet | SGDM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 40.35 | 3.4240 |
| UNet | SGDM | 1e-1 | none | 0.005 | 30 | 1 | 10 | 0.3 | 61.53 | 1.1519 |
| UNet | SGDM | 1e-3 | none | 0.005 | 30 | 1 | 10 | 0.3 | 60.99 | 1.1117 |
| UNet | SGDM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 40.87 | 2.5936 |
| UNet | ADAM | 1e-1 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 40.38 | 10.7453 |
| UNet | ADAM | 1e-3 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 42.77 | 1.1651 |
| UNet | ADAM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 36.48 | 5.9305 |
| UNet | ADAM | 1e-1 | none | 0.005 | 30 | 1 | 10 | 0.3 | 61.51 | 1.1569 |
| UNet | ADAM | 1e-3 | none | 0.005 | 30 | 1 | 10 | 0.3 | 37.67 | 1.1420 |
| UNet | ADAM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 39.51 | 5.2203 |
| UNet | ADAM | 1e-1 | none | 0.1 | 30 | 1 | 10 | 0.3 | 61.53 | 1.1576 |

Tabla 3.5: Resultados del entrenamiento con UNet

### 3.2.4 FCNNet

Finalmente, se evalúa el rendimiento de FCNNet ante las escenas del entorno de Abadín propuestas. En este caso, la FCN se preinicializa utilizando capas y pesos del modelo VGG-16. Posteriormente, se elige el modelo de FCN para llevar a cabo el proceso de entrenamiento. En este estudio se han empleado tres: FCN-8s, FCN-16s y FCN-32s. Estos actuarán sobre el mapa de características final debido a que en ocasiones es necesario obtener características más profundas. En el modelo FCN-32s este mapa de características se ve multiplicado por un factor de 32 proporcionando una segmentación aproximada lo que conlleva a un costo computacional bajo, FCN-16s tras su fusión con el mapa de características ubicado en la cuarta capa de agrupación, muestra el mapa final por un factor de 16. Al obtener información complementaria respecto a las capas anteriores se requieren cálculos adicionales. Finalmente, FCN-8s fusiona los mapas de características situados en la tercera y cuarta capa de agrupación máxima. Al igual que en el caso anterior, la información adicional que se consigue es a costa de un mayor número de cálculos (MathWorks, 2015).

Del mismo modo que en los entrenamientos con otros modelos, no es posible emplear un *MiniBatchSize* superior a 1 ya que se requiere de una mayor memoria en la GPU. Independientemente de la variación de cualquier parámetro, los resultados obtenidos en cada uno de los procesos de entrenamiento no han sido favorables, tabla 3.6. Al igual que con UNet se ha tratado de corregir la obtención de NaN en algún caso modificando el valor de *L2Regularization* entre otros, pero la respuesta de la red ha continuado siendo inapropiada para este tipo de fotografías, figura 3.10. Al tratar de emplear el optimizador ADAM, el tiempo de entrenamiento ha superado la hora, en cambio, para el optimizador SGDM se ha completado en 10 minutos por lo que el resultado con ADAM presenta unas pérdidas inadmisibles.

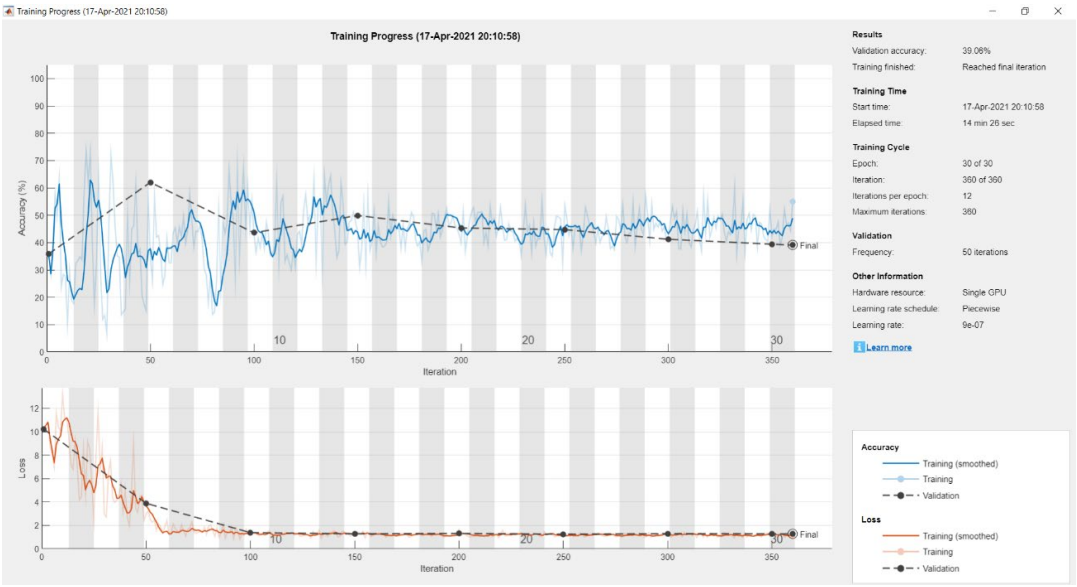



Figura 3.10: Resultado desfavorable con FCNNet

| | solverName | InitialLearnRate | LearnRateSchedule | L2Regularization | MaxEpochs | MiniBatchSize | LearnRateDropPeriod | LearnRateDropFactor | Validación (%) | Pérdidas |
|---|---|---|---|---|---|---|---|---|---|---|
| 8s / 16s / 32s | - | - | - | - | - | 4 | - | - | *Out of memory* | - |
| 8s | SGDM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.1 | 60.58 | 1.2503 |
| 8s | SGDM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 46.83 | 1.2672 |
| 8s | SGDM | 1e-3 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 6.91 | NaN |
| 8s | SGDM | 1e-5 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 54.33 | 1.2697 |
| 8s | SGDM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.1 | 52.26 | 1.2811 |
| 8s | SGDM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 54.39 | 1.2762 |
| 8s | SGDM | 1e-30 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 23.75 | 11.0105 |
| 16s | SGDM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 37.60 | 1.2621 |
| 16s | SGDM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 39.06 | 1.2689 |
| 32s | SGDM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 61.82 | 1.2513 |
| 32s | SGDM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 61.05 | 1.2190 |
| 8s | ADAM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 42.79 | 10.3831 |

Tabla 3.6: Resultados del entrenamiento con FCNNet

## 3.3 Clasificación

Tras el proceso de entrenamiento, se lleva a cabo la clasificación de las 18 imágenes de prueba seleccionadas partiendo de los modelos ya entrenados de tal forma que se predecirán las etiquetas de cada una de ellas. Estas estarán condicionadas a los resultados de validación y pérdidas obtenidos en el punto anterior por lo que, cabe esperar que ante un mayor porcentaje de validación y menor pérdida deberían obtenerse mejores resultados de clasificación, pero esto no es siempre así.

En la clasificación puede surgir el problema de desequilibrio de clases. Cuando las clases están extremadamente desequilibradas, significa que una o varias clases dominan en la imagen, a la vez que otras forman solamente una pequeña parte de esta. Desafortunadamente, el desequilibrio de clases prevalece en muchos conjuntos de datos del mundo real, por lo que no se puede ignorar. Ante tal respuesta es posible que se realice una clasificación en base a una clase mayoritaria dejando de lado la de menor representación, en consecuencia, se devolverá una predicción completamente inútil. Esto pretende ilustrar que una alta precisión de píxeles no siempre implica una capacidad de segmentación superior de mayor calidad y precisión.

Existen métricas para evaluar el modelo de segmentación semántica por lo que es posible conocer si el entrenamiento fue exitoso y por tanto, funcionará bien en la clasificación. Este aspecto también se trata dentro de la clasificación de tal manera que es posible llegar a una determinada conclusión ante una imagen ya etiquetada. Se empleará la Intersección Sobre Unión (Intersection Over Union, IoU), también conocido como el índice Jaccard, introducido por este autor como una medida de similitud en el ámbito de la botánica, rama de su especialidad (Jaccard, 1908). Es una de las métricas mayormente usadas en la segmentación semántica debido a que se trata de una métrica muy sencilla y extremadamente eficaz. IoU consiste en el área solapada entre la segmentación predicha y el *ground truth* dividido por el área de la unión entre la segmentación predicha y el *ground truth*. En MatLab se emplea la función *jaccard()* para tal fin. El resultado de esta métrica varia de 0 a 1 donde 0 significa que no hay superposición y 1 significa una segmentación perfectamente superpuesta. Esto es mínima y máxima similitud o coincidencia respectivamente.

Finalmente, se representarán las matrices de confusión, mecanismo de análisis predictivo que permite visualizar el rendimiento de un modelo de clasificación en un conjunto de datos de prueba para los que se conocen los valores verdaderos. Se genera una matriz donde en cada columna se sitúa el número de predicciones de cada clase y cada fila representa a las instancias en la clase real, es decir, muestra y compara los valores reales con los predichos del modelo. En el caso del presente trabajo nos encontramos ante tres clases predichas: CamposSinVegetación, CamposVerdes y SotoBosque. El clasificador realizará un determinado número de predicciones

de las cuales se pueden obtener distintos resultados, todo ello a nivel de píxel, que es la base de la segmentación semántica:

- Verdaderos positivos (TP)
- Verdaderos negativos (TN)
- Falsos positivos (FP)
- Falsos negativos (FN)

El motivo por el que la matriz de confusión es particularmente útil radica en que frente a otros tipos de métricas es capaz de producir una representación más completa del desempeño del modelo. Las métricas de precisión obtenidas durante los entrenamientos pueden llevar a una situación en la que los modelos identifican erróneamente una clase de manera constante pero esta pasa desapercibida porque el rendimiento promedio es bueno mientras que la matriz de confusión presenta una comparación de distintos valores.

A continuación, se expondrán los resultados de clasificación para cada uno de los métodos, así como las métricas de evaluación de los modelos.

### 3.3.1 DeepLabv3

En la siguiente tabla 3.7, se muestran las métricas de evaluación de segmentación semántica (IoU) de cada una de las clases para los entrenamientos realizados:

| | solverName | InitialLearnRate | LearnRateSchedule | L2Regularization | MaxEpochs | MiniBatchSize | LearnRateDrop Period | LearnRateDrop Factor | CamposSinVegetacion (IoU) | CamposVerdes (IoU) | SotoBosque (IoU) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ResNet18 | SGDM | 1e-1 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 0.17801 | 0.28239 | 0.73262 |
| ResNet18 | SGDM | 1e-3 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 0.23919 | 0.32931 | 0.61903 |
| ResNet18 | SGDM | 1e-5 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 0.025821 | 0.10626 | 0.20309 |
| ResNet18 | SGDM | 1e-1 | none | 0.1 | 30 | 4 | 10 | 0.3 | 0.17045 | 0.21599 | 0.71753 |
| ResNet18 | SGDM | 1e-3 | none | 0.1 | 30 | 4 | 10 | 0.3 | 0.22866 | 0.32522 | 0.61697 |
| ResNet18 | SGDM | 1e-5 | none | 0.1 | 30 | 4 | 10 | 0.3 | 0.035331 | 0.11494 | 0.30906 |
| ResNet18 | SGDM | 1e-1 | piecewise | 0.1 | 60 | 4 | 10 | 0.3 | 0.20897 | 0.29605 | 0.63942 |
| ResNet18 | SGDM | 1e-3 | piecewise | 0.1 | 60 | 4 | 10 | 0.3 | 0.26535 | 0.33203 | 0.61894 |
| ResNet18 | SGDM | 1e-5 | piecewise | 0.1 | 60 | 4 | 10 | 0.3 | 0.020974 | 0.10632 | 0.21777 |
| ResNet18 | SGDM | 1e-1 | none | 0.1 | 60 | 4 | 10 | 0.3 | 0.20099 | 0 | 0.54417 |
| ResNet18 | SGDM | 1e-3 | none | 0.1 | 60 | 4 | 10 | 0.3 | 0.19483 | 0.34674 | 0.62988 |
| ResNet18 | SGDM | 1e-5 | none | 0.1 | 60 | 4 | 10 | 0.3 | 0.04767 | 0.14351 | 0.41656 |
| ResNet18 | SGDM | 1e-1 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.13715 | 0.28914 | 0.68426 |
| ResNet18 | SGDM | 1e-3 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.24478 | 0.30915 | 0.60437 |
| ResNet18 | SGDM | 1e-5 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.025775 | 0.10625 | 0.20286 |
| ResNet18 | SGDM | 1e-1 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.17713 | 0.25251 | 0.68859 |
| ResNet18 | SGDM | 1e-3 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.23222 | 0.31608 | 0.61398 |
| ResNet18 | SGDM | 1e-5 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.035241 | 0.11485 | 0.30843 |
| ResNet18 | SGDM | 1e-1 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.15933 | 0.25943 | 0.63335 |
| ResNet18 | SGDM | 1e-3 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.25697 | 0.38199 | 0.63294 |
| ResNet18 | SGDM | 1e-3 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.29683 | 0.38629 | 0.63432 |
| ResNet18 | ADAM | 1e-1 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.03905 | 0 | 0.68613 |
| ResNet18 | ADAM | 1e-3 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.21235 | 0.34769 | 0.64291 |
| ResNet18 | ADAM | 1e-5 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.13628 | 0.26328 | 0.57767 |
| ResNet18 | ADAM | 1e-1 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.052091 | 0.0011158 | 0.61814 |
| ResNet18 | ADAM | 1e-3 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.21861 | 0.32585 | 0.70954 |
| ResNet18 | ADAM | 1e-5 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.18359 | 0.3016 | 0.60547 |
| ResNet50 | SGDM | 1e-1 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 0.23299 | 0.3049 | 0.70088 |
| ResNet50 | SGDM | 1e-3 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 0.2426 | 0.32103 | 0.66053 |
| ResNet50 | SGDM | 1e-5 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 0.055022 | 0.10657 | 0.08004 |
| ResNet50 | SGDM | 1e-1 | none | 0.1 | 30 | 4 | 10 | 0.3 | 0.024789 | 0.11962 | 0.38192 |
| ResNet50 | SGDM | 1e-3 | none | 0.1 | 30 | 4 | 10 | 0.3 | 0.23255 | 0.33393 | 0.66042 |
| ResNet50 | SGDM | 1e-5 | none | 0.1 | 30 | 4 | 10 | 0.3 | 0.053658 | 0.11312 | 0.24287 |
| ResNet50 | SGDM | 1e-1 | piecewise | 0.1 | 60 | 4 | 10 | 0.3 | 0.20764 | 0.31581 | 0.703 |
| ResNet50 | SGDM | 1e-3 | piecewise | 0.1 | 60 | 4 | 10 | 0.3 | 0.23662 | 0.32872 | 0.65889 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ResNet50 | SGDM | 1e-5 | piecewise | 0.1 | 60 | 4 | 10 | 0.3 | 0.048225 | 0.10326 | 0.090108 |
| ResNet50 | SGDM | 1e-1 | none | 0.1 | 60 | 4 | 10 | 0.3 | 0.13703 | 0 | 0.56266 |
| ResNet50 | SGDM | 1e-3 | none | 0.1 | 60 | 4 | 10 | 0.3 | 0.17578 | 0.39761 | 0.68876 |
| ResNet50 | SGDM | 1e-5 | none | 0.1 | 60 | 4 | 10 | 0.3 | 0.059543 | 0.1479 | 0.45096 |
| ResNet50 | SGDM | 1e-1 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.222 | 0.2392 | 0.68649 |
| ResNet50 | SGDM | 1e-3 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.23329 | 0.30972 | 0.66104 |
| ResNet50 | SGDM | 1e-5 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.055098 | 0.10657 | 0.079734 |
| ResNet50 | SGDM | 1e-1 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.19122 | 0.34903 | 0.65024 |
| ResNet50 | SGDM | 1e-3 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.25811 | 0.31203 | 0.65542 |
| ResNet50 | SGDM | 1e-5 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.053647 | 0.11299 | 0.24145 |
| ResNet50 | ADAM | 1e-1 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.22064 | 0.16772 | 0.56704 |
| ResNet50 | ADAM | 1e-3 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.21251 | 0.39805 | 0.65884 |
| ResNet50 | ADAM | 1e-5 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.18763 | 0.27575 | 0.63807 |
| ResNet50 | ADAM | 1e-1 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.17131 | 0 | 0.54802 |
| ResNet50 | ADAM | 1e-3 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.21118 | 0.39079 | 0.66077 |
| ResNet50 | ADAM | 1e-5 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.2083 | 0.30178 | 0.6299 |
| MobileNetv2 | SGDM | 1e-1 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 0.22039 | 0.30309 | 0.72512 |
| MobileNetv2 | SGDM | 1e-3 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 0.16905 | 0.27725 | 0.59992 |
| MobileNetv2 | SGDM | 1e-5 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 0.0088149 | 0.10927 | 0.15562 |
| MobileNetv2 | SGDM | 1e-1 | none | 0.1 | 30 | 4 | 10 | 0.3 | 0.066009 | 0.098173 | 0 |
| MobileNetv2 | SGDM | 1e-3 | none | 0.1 | 30 | 4 | 10 | 0.3 | 0.182 | 0.36021 | 0.64181 |
| MobileNetv2 | SGDM | 1e-5 | none | 0.1 | 30 | 4 | 10 | 0.3 | 0.0094232 | 0.11862 | 0.23572 |
| MobileNetv2 | SGDM | 1e-1 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.20495 | 0.25805 | 0.71927 |
| MobileNetv2 | SGDM | 1e-3 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.13494 | 0.24907 | 0.58627 |
| MobileNetv2 | SGDM | 1e-5 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.0088103 | 0.10932 | 0.15543 |
| MobileNetv2 | SGDM | 1e-1 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.15834 | 0.23057 | 0.69585 |
| MobileNetv2 | SGDM | 1e-3 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.19128 | 0.31713 | 0.62448 |
| MobileNetv2 | SGDM | 1e-5 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.0094206 | 0.11877 | 0.23534 |
| MobileNetv2 | ADAM | 1e-1 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.018956 | 0.27988 | 0 |
| MobileNetv2 | ADAM | 1e-3 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.22305 | 0.36346 | 0.63156 |
| MobileNetv2 | ADAM | 1e-5 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.019541 | 0.13152 | 0.34855 |
| MobileNetv2 | ADAM | 1e-1 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.013398 | 0.14083 | 0.00052066 |
| MobileNetv2 | ADAM | 1e-3 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.14861 | 0.41297 | 0.65761 |
| MobileNetv2 | ADAM | 1e-5 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.051496 | 0.20587 | 0.58084 |
| Xception | SGDM | 1e-1 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 0.18842 | 0.23877 | 0.77013 |
| Xception | SGDM | 1e-3 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 0.17951 | 0.31543 | 0.6813 |
| Xception | SGDM | 1e-5 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | 0.030333 | 0.078079 | 0.44533 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Xception | SGDM | 1e-1 | none | 0.1 | 30 | 4 | 10 | 0.3 | 0 | 0.23564 | 0.73091 |
| Xception | SGDM | 1e-3 | none | 0.1 | 30 | 4 | 10 | 0.3 | 0.177 | 0.32893 | 0.67272 |
| Xception | SGDM | 1e-5 | none | 0.1 | 30 | 4 | 10 | 0.3 | 0.042416 | 0.096488 | 0.46972 |
| Xception | SGDM | 1e-1 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.19506 | 0.19863 | 0.71253 |
| Xception | SGDM | 1e-3 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.18733 | 0.29955 | 0.67427 |
| Xception | SGDM | 1e-5 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.030303 | 0.078031 | 0.4453 |
| Xception | SGDM | 1e-1 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.21769 | 0.26464 | 0.7128 |
| Xception | SGDM | 1e-3 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.20004 | 0.31257 | 0.67203 |
| Xception | SGDM | 1e-5 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.042282 | 0.096375 | 0.46944 |
| Xception | ADAM | 1e-1 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.25296 | 0.2326 | 0.67785 |
| Xception | ADAM | 1e-3 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.16185 | 0.38579 | 0.67596 |
| Xception | ADAM | 1e-5 | piecewise | 0.005 | 30 | 4 | 10 | 0.3 | 0.10934 | 0.28381 | 0.63987 |
| Xception | ADAM | 1e-1 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.0060077 | 0.13659 | 0.48035 |
| Xception | ADAM | 1e-3 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.1327 | 0.37168 | 0.66833 |
| Xception | ADAM | 1e-5 | none | 0.005 | 30 | 4 | 10 | 0.3 | 0.16573 | 0.33024 | 0.64458 |
| InceptionResnetV2 | SGDM | 1e-1 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | - | - | - |
| InceptionResnetV2 | SGDM | 1e-1 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 0.14278 | 0.019636 | 0.56847 |
| InceptionResnetV2 | SGDM | 1e-3 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 0.2388 | 0.41137 | 0.64010 |
| InceptionResnetV2 | SGDM | 1e-5 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 0.067224 | 0.17033 | 0.43482 |
| InceptionResnetV2 | SGDM | 1e-1 | none | 0.1 | 30 | 1 | 10 | 0.3 | 0 | 0 | 0.52621 |
| InceptionResnetV2 | SGDM | 1e-3 | none | 0.1 | 30 | 1 | 10 | 0.3 | 0.19773 | 0.37825 | 0.63565 |
| InceptionResnetV2 | SGDM | 1e-5 | none | 0.1 | 30 | 1 | 10 | 0.3 | 0.10003 | 0.25273 | 0.5889 |
| InceptionResnetV2 | SGDM | 1e-1 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.15225 | 0.19677 | 0.55404 |
| InceptionResnetV2 | SGDM | 1e-3 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.21385 | 0.40463 | 0.6505 |
| InceptionResnetV2 | SGDM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.067019 | 0.17227 | 0.45326 |
| InceptionResnetV2 | SGDM | 1e-1 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.19746 | 0.24329 | 0.6205 |
| InceptionResnetV2 | SGDM | 1e-3 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.2398 | 0.41136 | 0.64015 |
| InceptionResnetV2 | SGDM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.089372 | 0.22756 | 0.54235 |
| InceptionResnetV2 | ADAM | 1e-1 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.012296 | 0.00056649 | 0 |
| InceptionResnetV2 | ADAM | 1e-3 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.20838 | 0.39425 | 0.65151 |
| InceptionResnetV2 | ADAM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.13147 | 0.38897 | 0.64762 |
| InceptionResnetV2 | ADAM | 1e-1 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.012169 | 0.013306 | 9.0785e-05 |
| InceptionResnetV2 | ADAM | 1e-3 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.24217 | 0.39561 | 0.65255 |
| InceptionResnetV2 | ADAM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.15079 | 0.36762 | 0.64584 |

Tabla 3.7: Métricas de evaluación IoU para DeepLabv3

A partir de las soluciones más favorables obtenidas en el entrenamiento, se muestran los resultados de la clasificación además de las métricas IoU correspondientes según los distintos modelos que se indican a continuación:

- ResNet18

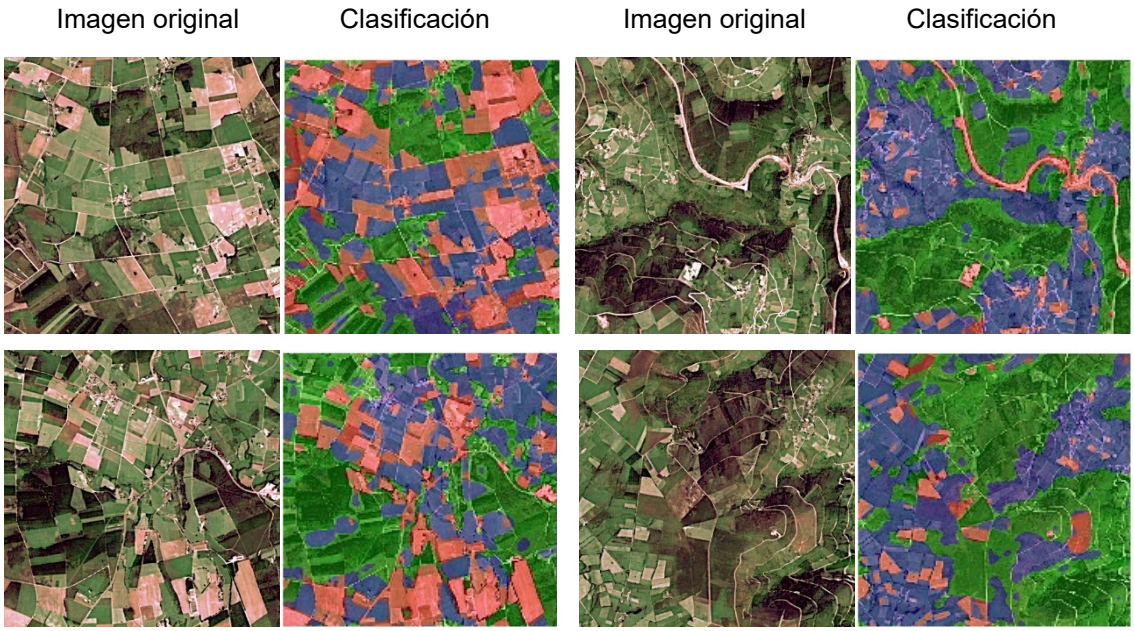
| Solver Name | LearnRate Schedule | InitialLearn Rate | L2regulari zation | Max Epochs | MiniBatch Size | Validación (%) | Pérdidas |
|---|---|---|---|---|---|---|---|
| ADAM | piecewise | 1e-3 | 0.005 | 30 | 4 | 85.80 | 0.5450 |

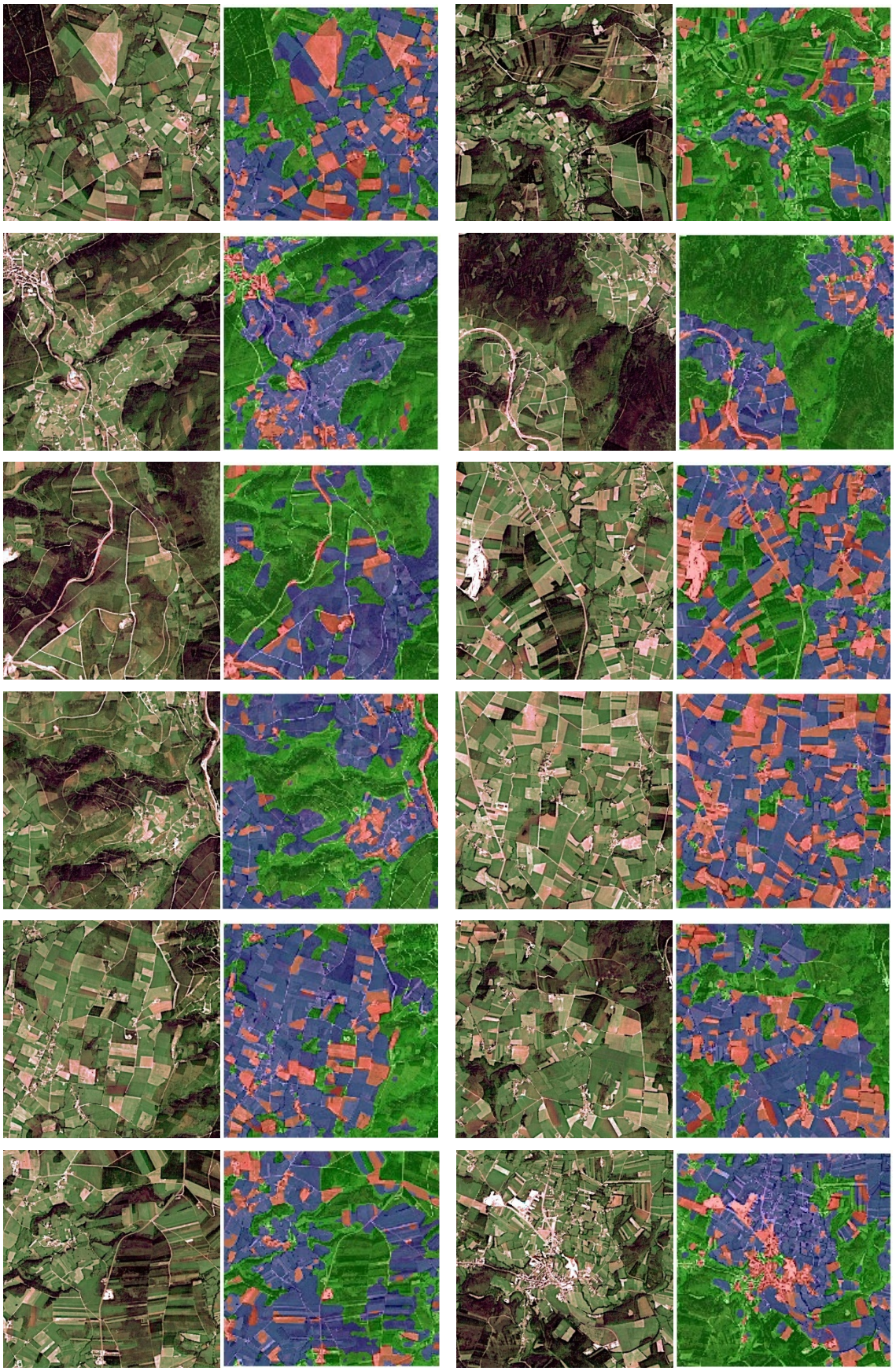
Tabla 3.8: Parámetros favorables empleados para la clasificación con ResNet18

| CamposSinVegetación (IoU) | CamposVerdes (IoU) | SotoBosque (IoU) |
|---|---|---|
| 0.21235 | 0.34769 | 0.64291 |

Tabla 3.9: Métricas IoU obtenidas para el ejemplo ResNet18

El resultado de la clasificación es el que se muestra en la secuencia de imágenes en la figura 3.11, donde aparece la imagen original y a su derecha la imagen clasificada:





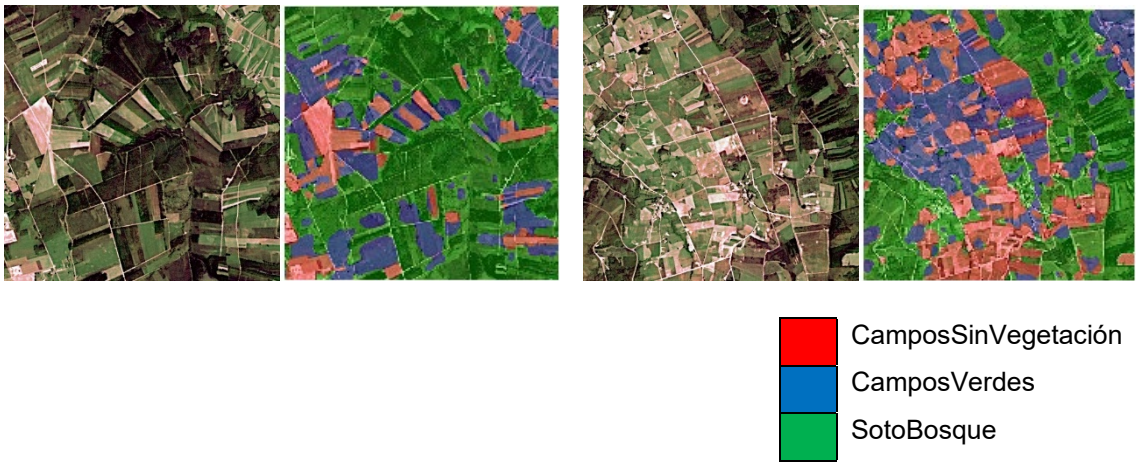



Figura 3.11: Clasificación favorable de imágenes con ResNet18

Como se contempla en la tabla 3.9, las métricas IoU obtenidas para cada una de las clases tienen un valor razonable ya que SotoBosque es la que predomina, después CamposVerdes y CamposSinVegetación se localiza en menor grado. En este ejemplo, la predicción, considerando la limitación respecto al número de fotografías procesadas en el entrenamiento, es favorable. Se diferencian claramente las tres etiquetas creadas que corresponden en buena parte con la realidad.

La matriz de confusión asociada a este experimento es la mostrada en la tabla 3.10 donde se reflejan las tres clases de segmentación semántica siendo cada uno de los elementos de la tabla el recuento de píxeles que se sabe que pertenecen a una clase determinada pero que se prevé que pertenecen a otra, dividido por el número total de píxeles previsto en cada clase. Al tratarse de un valor normalizado se obtienen resultados dentro del rango [0, 1] (Mathworks, 2017).

| | | Clase verdadera | | |
|---|---|---|---|---|
| | | CamposSinVegetación | Campos Verdes | Soto Bosque |
| Clase predicha | CamposSinVegetación | 0.87092 | 0.084791 | 0.044292 |
| | CamposVerdes | 0.036982 | 0.79814 | 0.16488 |
| | SotoBosque | 0.0017984 | 0.087333 | 0.91087 |

Tabla 3.10: Matriz de confusión con SegNet18

Las celdas diagonales corresponden a observaciones que están correctamente clasificadas. Las celdas fuera de la diagonal corresponden a observaciones clasificadas incorrectamente. A partir de los resultados mostrados en la tabla 3.10 se observa que el mejor desempeño se obtiene para la clase Sotobosque, llegando al 90% de aciertos y en menor medida en las otras dos clases.

- InceptionResnetV2

Con InceptionResNetV2 se ha obtenido el mejor resultado de los entrenamientos de DeepLabv3 con los siguientes parámetros, tabla 3.11, tabla 3.12:

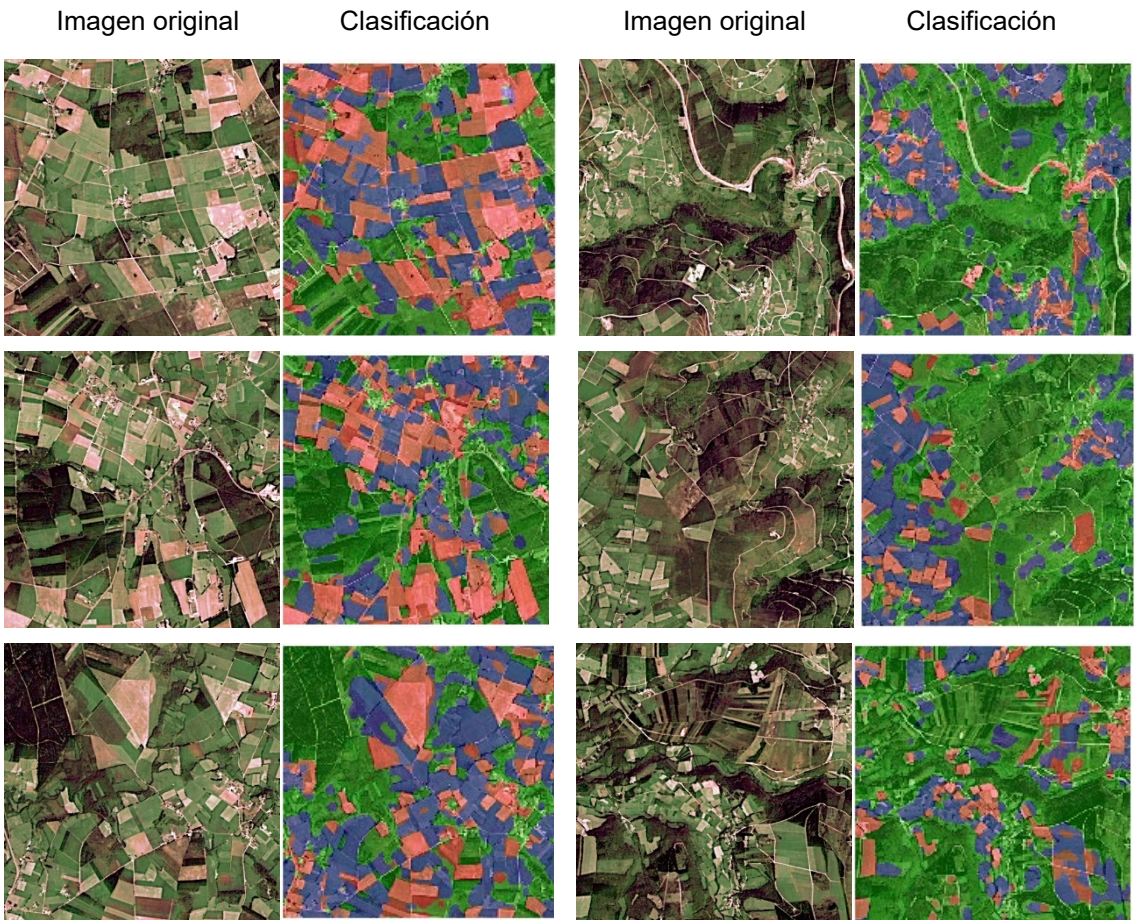
| Solver Name | LearnRate Schedule | InitialLearn Rate | L2regulari zation | Max Epochs | MiniBatch Size | Validación (%) | Pérdidas |
|---|---|---|---|---|---|---|---|
| ADAM | piecewise | 1e-3 | 0.005 | 30 | 1 | 86.34 | 0.4590 |

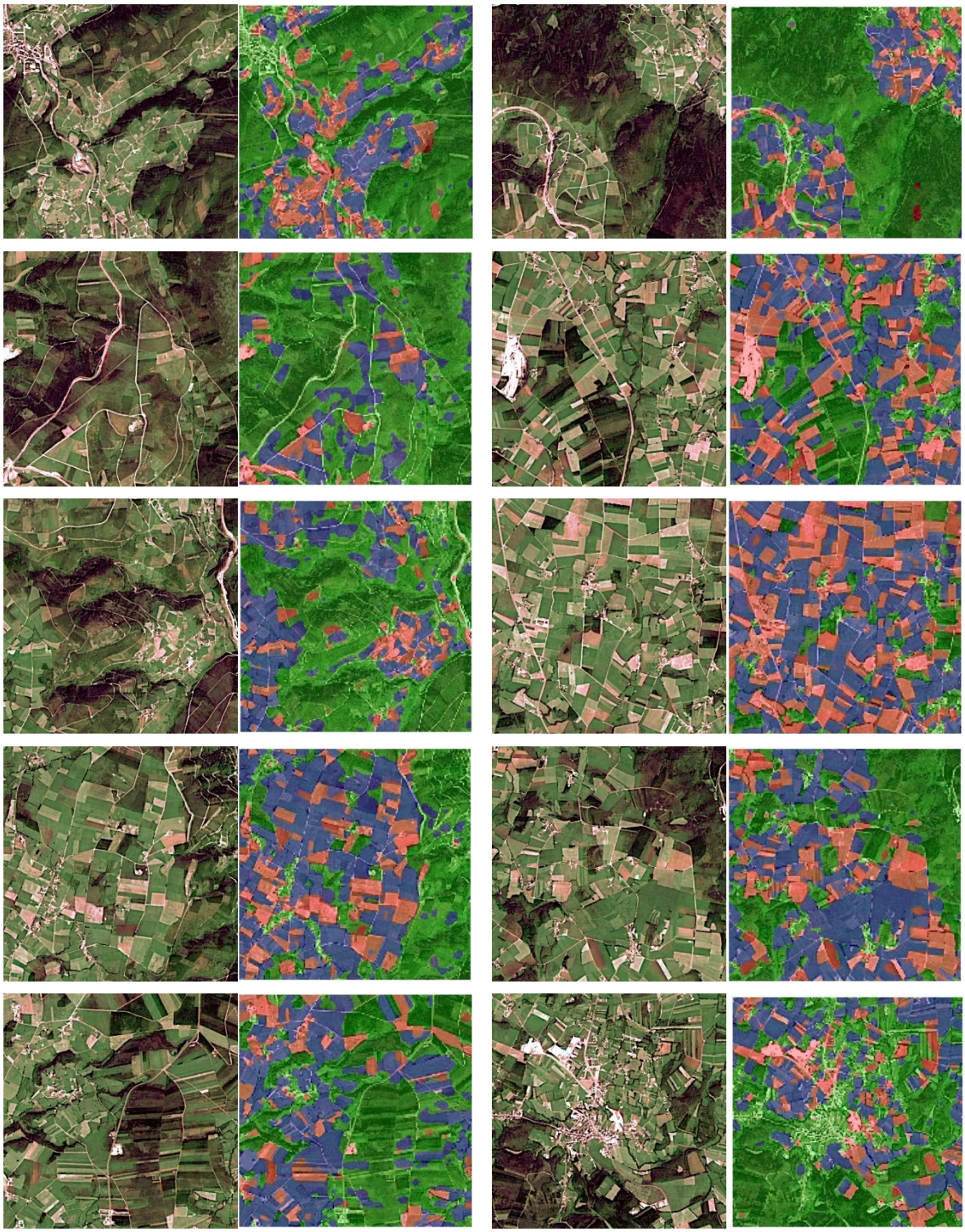
Tabla 3.11: Parámetros favorables empleados para la clasificación con InceptionResNetV2

| CamposSinVegetación (IoU) | CamposVerdes (IoU) | SotoBosque (IoU) |
|---|---|---|
| 0.20838 | 0.39425 | 0.65151 |

Tabla 3.12: Métricas IoU obtenidas para el ejemplo InceptionResNetV2

El resultado de la clasificación es el mostrado en la secuencia de imágenes de la figura 3.12:





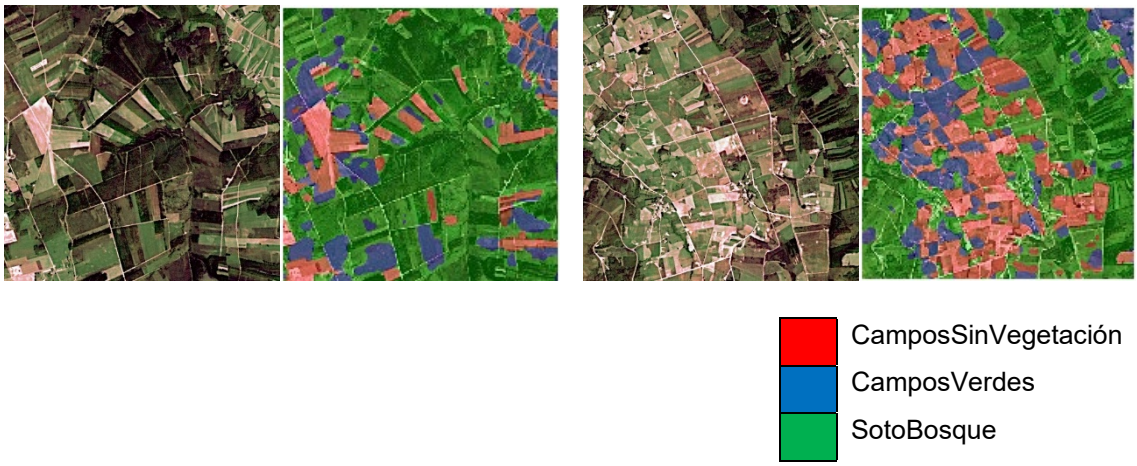



Figura 3.12: Clasificación favorable de imágenes con InceptionResnetV2

De igual forma que para el ejemplo con ResNet18, las métricas IoU de InceptionResnetv2 son coherentes, tabla 3.12, pero, aunque se haya obtenido una validación mayor y una menor pérdida, como los datos obtenidos son muy próximos, no se aprecia una gran mejoría con respecto a la segmentación final.

Por último, se obtiene la matriz de confusión según este mismo experimento, representada en la tabla 3.13:

| | | Clase verdadera | | |
|---|---|---|---|---|
| | | CamposSinVegetación | Campos Verdes | Soto Bosque |
| Clase predicha | CamposSinVegetación | 0.97303 | 0.0043828 | 0.022588 |
| | CamposVerdes | 0.089827 | 0.78973 | 0.12045 |
| | SotoBosque | 0.0021649 | 0.058999 | 0.93884 |

Tabla 3.13: Matriz de confusión con InceptionResnetV2

Las celdas diagonales corresponden a observaciones que están correctamente clasificadas. Las celdas fuera de la diagonal corresponden a observaciones clasificadas incorrectamente. De nuevo es en el caso de la categoría SotoBosque donde se obtienen los mejores resultados en términos de desempeño del método.

### 3.3.2 SegNet

Al igual que con el modelo anterior, se extrae el mejor resultado procedente de la fase de entrenamiento, tabla 3.14 con los parámetros IoU expuestos en la tabla 3.15:

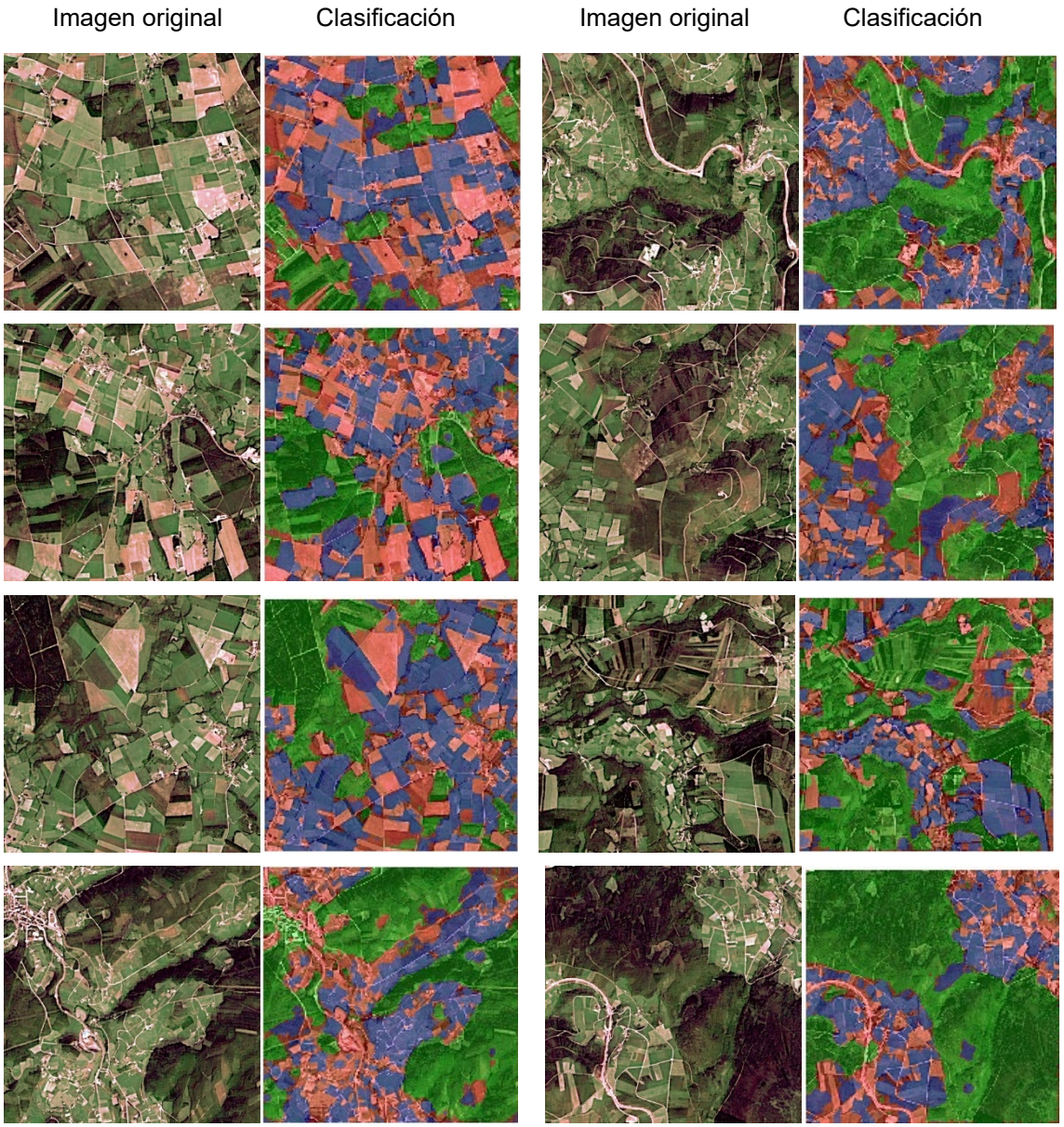
| Solver Name | LearnRate Schedule | InitialLearn Rate | L2regulari zation | Max Epochs | MiniBatch Size | Validación (%) | Pérdidas |
|---|---|---|---|---|---|---|---|
| ADAM | piecewise | 1e-3 | 0.005 | 30 | 1 | 84.91 | 0.5480 |

Tabla 3.14: Parámetros favorables empleados para la clasificación con SegNet

| CamposSinVegetación (IoU) | CamposVerdes (IoU) | SotoBosque (IoU) |
|---|---|---|
| 0.071451 | 0.3953 | 0.70156 |

Tabla 3.15: Métricas IoU obtenidas para el ejemplo SegNet

El resultado de la clasificación es el expuesto en la secuencia de imágenes originales y clasificadas mostradas en la figura 3.13:



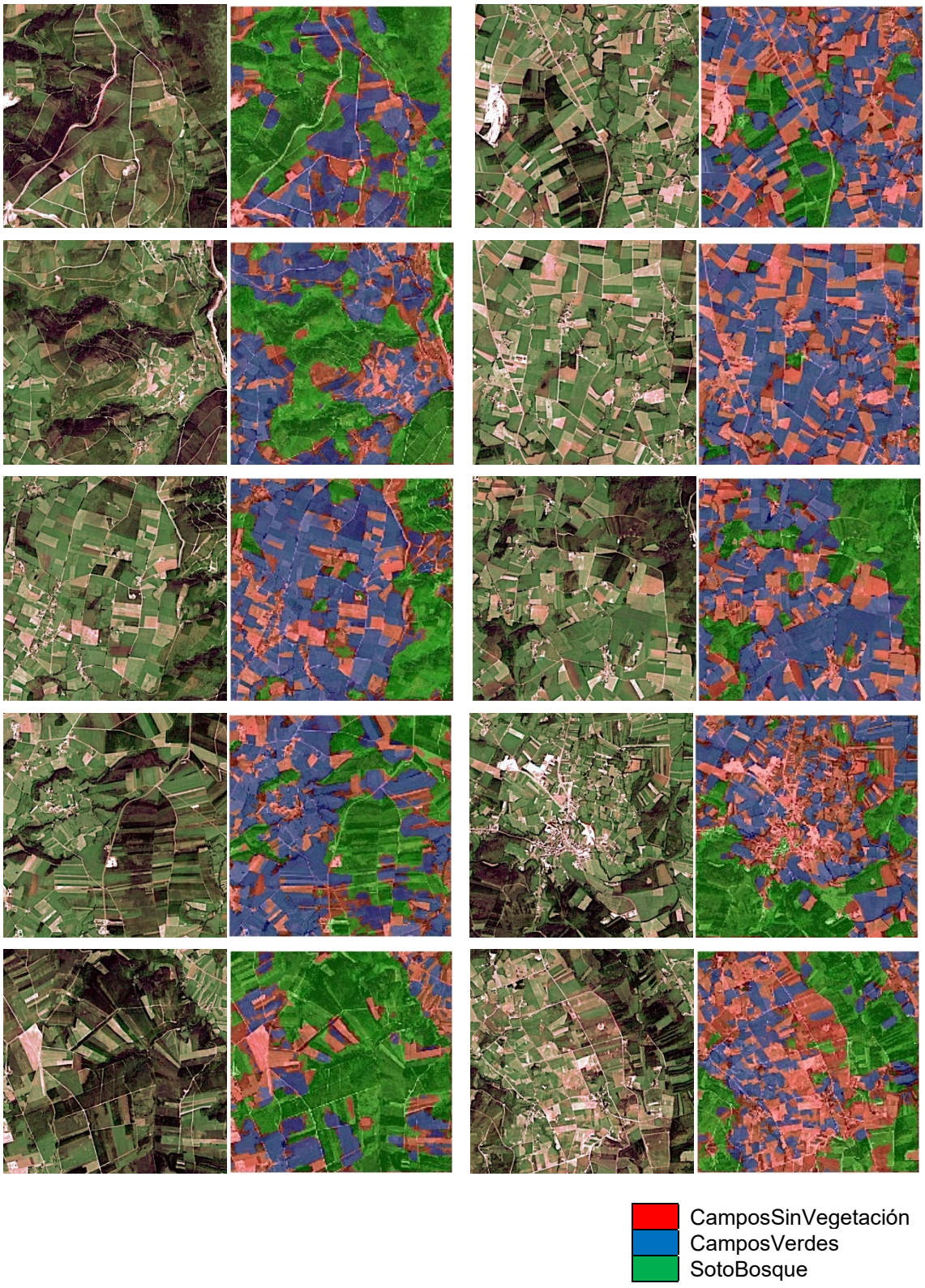



Figura 3.13: Clasificación favorable de imágenes con SegNet

Su correspondiente matriz de confusión en la representada en la tabla 3.16:

| | | Clase verdadera | | |
|---|---|---|---|---|
| | | CamposSinVegetación | Campos Verdes | Soto Bosque |
| Clase predicha | CamposSinVegetación | 0.97632 | 0.0013064 | 0.022378 |
| | CamposVerdes | 0.13265 | 0.78917 | 0.078177 |
| | SotoBosque | 0.086829 | 0.059273 | 0.8539 |

Tabla 3.16: Matriz de confusión con SegNet

Ante estos resultados, aunque las soluciones teóricas obtenidas al finalizar el entrenamiento eran aceptables, con una validación del 84% y pérdidas entorno al 0.5, es posible observar a simple vista que el modelo entrenado no ha funcionado como se esperaba. Esto se ve reflejado en las métricas de evaluación IoU. La clase CamposSinVegetación obtiene un valor muy bajo por lo que surge un problema de desequilibrio de clases, tabla 3.15, aunque este se ve acentuado en otros entrenamientos para el mismo modelo. Por ejemplo, para los datos de la tabla 3.17, se obtiene una métrica IoU igual a cero en CamposSinVegetación como se puede ser en la tabla 3.18.

| Solver Name | LearnRate Schedule | InitialLearn Rate | L2regulari zation | Max Epochs | MiniBatch Size | Validación (%) | Pérdidas |
|---|---|---|---|---|---|---|---|
| ADAM | piecewise | 1e-1 | 0.005 | 30 | 1 | 75.96 | 1.0396 |

Tabla 3.17: Parámetros desfavorables empleados para la clasificación con SegNet

| CamposSinVegetación (IoU) | CamposVerdes (IoU) | SotoBosque (IoU) |
|---|---|---|
| 0 | 0.24586 | 0.68194 |

Tabla 3.18: Métricas IoU obtenidas para el ejemplo SegNet

Esto provoca nuevamente un desequilibrio de clases, pero esta vez una de ellas no se tendrá en cuenta para la clasificación resultando como sigue, figura 3.14:

Imagen original Clasificación Imagen original Clasificación






Figura 3.14: Clasificación desfavorable de imágenes con SegNet

A continuación, en la tabla 3.19, se exponen todos los resultados de las métricas de clasificación IoU para todos los entrenamientos llevados a cabo con el modelo Segnet:

| | solverName | InitialLearnRate | LearnRateSchedule | L2Regularization | MaxEpochs | MiniBatchSize | LearnRateDrop Period | LearnRateDrop Factor | CamposSinVegetacion (IoU) | CamposVerdes (IoU) | SotoBosque (IoU) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SegNet | sqdm | 1e-1 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | - | - | - |
| SegNet | SGDM | 1e-1 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 0.028035 | 0 | 0.66787 |
| SegNet | SGDM | 1e-3 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 0.084867 | 0.088591 | 0.50136 |
| SegNet | SGDM | 1e-5 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 0.012409 | 0.091005 | 0.29656 |
| SegNet | SGDM | 1e-1 | none | 0.1 | 30 | 1 | 10 | 0.3 | 0 | 0 | 0.52621 |
| SegNet | SGDM | 1e-3 | none | 0.1 | 30 | 1 | 10 | 0.3 | 0.096922 | 0.066645 | 0.57293 |
| SegNet | SGDM | 1e-5 | none | 0.1 | 30 | 1 | 10 | 0.3 | 0.012454 | 0.090969 | 0.2993 |
| SegNet | SGDM | 1e-1 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.26774 | 0.37133 | 0.62651 |
| SegNet | SGDM | 1e-3 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.075974 | 0.089919 | 0.48988 |
| SegNet | SGDM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.012377 | 0.091136 | 0.29659 |
| SegNet | SGDM | 1e-1 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.16713 | 0.27168 | 0.68805 |
| SegNet | SGDM | 1e-3 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.098216 | 0.074775 | 0.56041 |
| SegNet | SGDM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.012358 | 0.091048 | 0.29871 |
| SegNet | ADAM | 1e-1 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0 | 0.24586 | 0.68194 |
| SegNet | ADAM | 1e-3 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.071451 | 0.3953 | 0.70156 |
| SegNet | ADAM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.036073 | 0.094011 | 0.48088 |
| SegNet | ADAM | 1e-1 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.016489 | 0.10359 | 0.11422 |
| SegNet | ADAM | 1e-3 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.14657 | 0.36227 | 0.67232 |
| SegNet | ADAM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.081807 | 0.10158 | 0.49426 |

Tabla 3.19: Métricas de evaluación IoU para SegNet

### 3.3.3 UNet

Dados los resultados del entrenamiento comentados anteriormente, es evidente que cualquier clasificación realizada con este modelo entrenado no será correcta. Se toma como ejemplo uno de los casos en los que no se ha obtenido un valor de cero para ninguna clase en las métricas de validación IoU, tabla 3.20 y tabla 3.21.

| Solver Name | LearnRate Schedule | InitialLearn Rate | L2regulari zation | Max Epochs | MiniBatch Size | Validación (%) | Pérdidas |
|---|---|---|---|---|---|---|---|
| SGDM | piecewise | 1e-5 | 0.1 | 30 | 1 | 40.35 | 3.4046 |

Tabla 3.20: Parámetros desfavorables empleados para la clasificación con UNet

| CamposSinVegetación (IoU) | CamposVerdes (IoU) | SotoBosque (IoU) |
|---|---|---|
| 0.0014696 | 0.05289 | 0.39355 |

Tabla 3.21: Métricas IoU obtenidas para el ejemplo UNet

Los resultados de clasificación son los que se muestran en las imágenes de la figura 3.15:

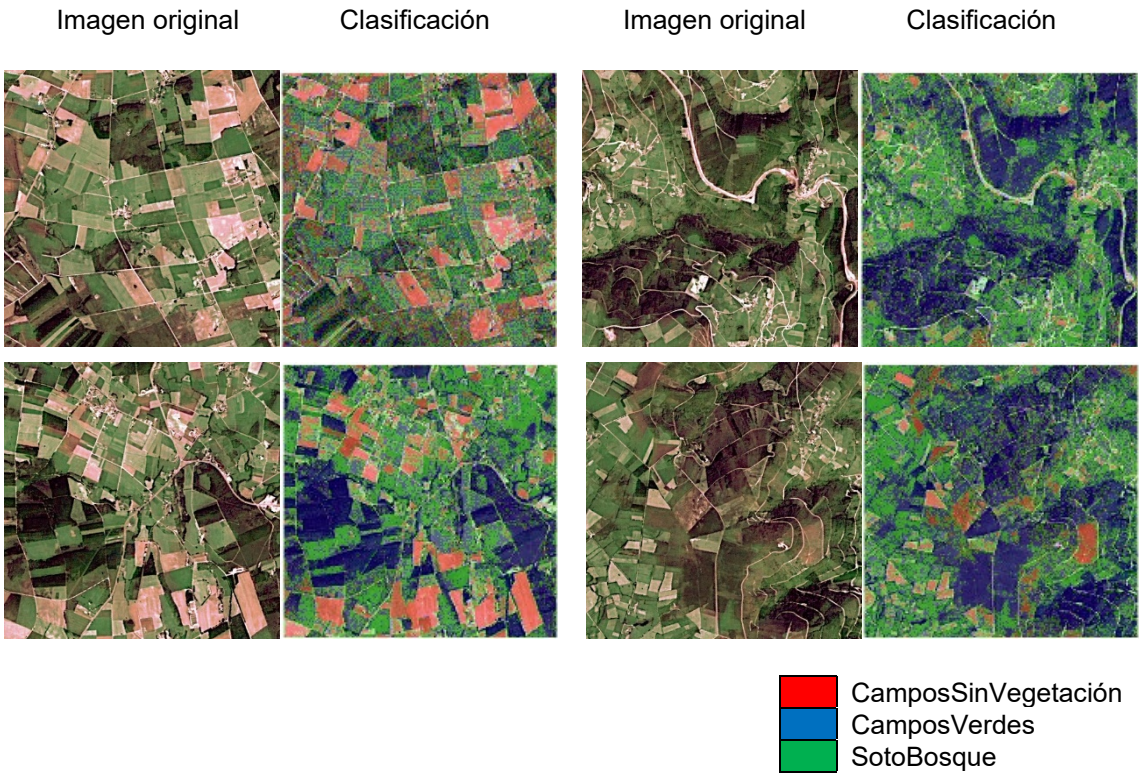



Figura 3.15: Clasificación desfavorable de imágenes con UNet

Frente a estos resultados la matriz de confusión no aportaría nada relevante, por lo que no es necesaria su obtención y análisis.

En la tabla 3.22 se muestran los resultados de todos los entrenamientos realizados y su correspondiente métrica IoU. La mayoría presentan problemas de desequilibrio de clases.

| | solverName | InitialLearnRate | LearnRateSchedule | L2Regularization | MaxEpochs | MiniBatchSize | LearnRateDrop Period | LearnRateDrop Factor | CamposSinVegetacion (IoU) | CamposVerdes (IoU) | SotoBosque (IoU) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| UNet | sqgm | 1e-1 | piecewise | 0.1 | 30 | 4 | 10 | 0.3 | - | - | - |
| UNet | SGDM | 1e-1 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 0 | 0 | 0 |
| UNet | SGDM | 1e-3 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 0 | 0.018179 | 0.75441 |
| UNet | SGDM | 1e-5 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 0.0014696 | 0.05289 | 0.39355 |
| UNet | SGDM | 1e-1 | none | 0.1 | 30 | 1 | 10 | 0.3 | 0 | 0 | 0 |
| UNet | SGDM | 1e-3 | none | 0.1 | 30 | 1 | 10 | 0.3 | 0 | 0.058404 | 0.37265 |
| UNet | SGDM | 1e-5 | none | 0.1 | 30 | 1 | 10 | 0.3 | 0.001381 | 0.053131 | 0.39939 |
| UNet | SGDM | 1e-1 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0 | 0 | 0 |
| UNet | SGDM | 1e-3 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0 | 0.057266 | 0.38167 |
| UNet | SGDM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.0014944 | 0.052814 | 0.39349 |
| UNet | SGDM | 1e-1 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0 | 0 | 0 |
| UNet | SGDM | 1e-3 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0 | 0.010363 | 0.76636 |
| UNet | SGDM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.0014041 | 0.053124 | 0.39951 |
| UNet | ADAM | 1e-1 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0 | 0.055238 | 0.258 |
| UNet | ADAM | 1e-3 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.00023739 | 0.061237 | 0.33883 |
| UNet | ADAM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.0019193 | 0.05185 | 0.35567 |
| UNet | ADAM | 1e-1 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0 | 0.00013777 | 0.77567 |
| UNet | ADAM | 1e-3 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.0016313 | 0.059485 | 0.18203 |
| UNet | ADAM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.0019953 | 0.051122 | 0.39723 |
| UNet | ADAM | 1e-1 | none | 0.1 | 30 | 1 | 10 | 0.3 | 0 | 0 | 0.77505 |

3.22: Métricas de evaluación IoU para UNet

### 3.3.4 FCNNet

Del mismo modo que con el modelo anterior, al encontrarnos ante unos resultados de entrenamiento desfavorables lo esperado es obtener métricas negativas lo que conlleva a una mala clasificación. En este caso, de todos los entrenamientos mostrados en la tabla 3.27, que se muestra después, solamente en dos se ha obtenido un valor mayor a cero para IoU en cada una de las etiquetas, pero las pérdidas se sitúan con un valor excesivo mayor que 10, tabla 3.23 y tabla 3.24.

- FCN-8s

| Solver Name | LearnRate Schedule | InitialLearn Rate | L2regulari zation | Max Epochs | MiniBatch Size | Validación (%) | Pérdidas |
|---|---|---|---|---|---|---|---|
| SGDM | piecewise | 1e-30 | 0.005 | 30 | 1 | 23.75 | 11.0105 |

Tabla 3.23: Parámetros desfavorables empleados para la clasificación con FCN-8s

| CamposSinVegetación (IoU) | CamposVerdes (IoU) | SotoBosque (IoU) |
|---|---|---|
| 0.0014696 | 0.05289 | 0.39355 |

Tabla 3.24: Métricas IoU obtenidas para el ejemplo FCN-8s

Los resultados de clasificación son los mostrados en la figura 3.16:

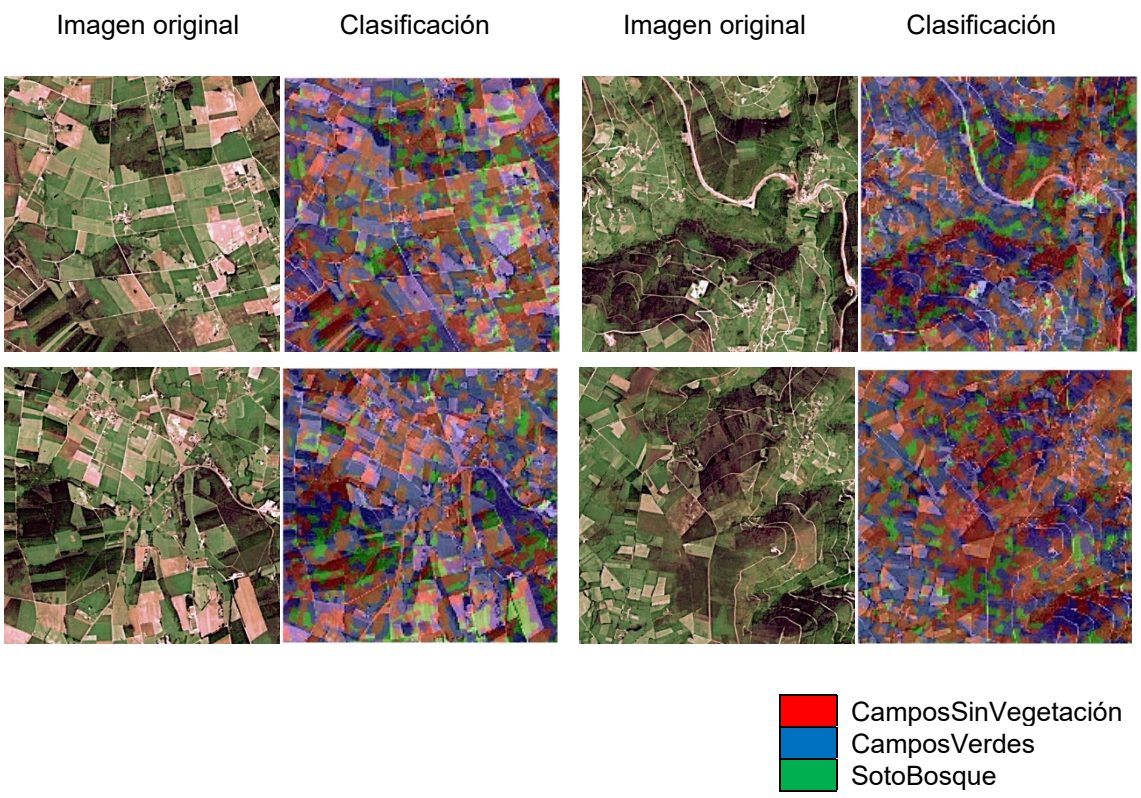



Figura 3.16: Clasificación desfavorable de imágenes con FCN-8s

A continuación se muestra otro caso de ejemplo en el que las pérdidas son menores, aunque se manifieste el problema de discriminación de clases, tabla 3.25 y tabla 3.26.

- FCN-32s

| Solver Name | LearnRate Schedule | InitialLearn Rate | L2regulari zation | Max Epochs | MiniBatch Size | Validación (%) | Pérdidas |
|---|---|---|---|---|---|---|---|
| SGDM | none | 1e-5 | 0.005 | 30 | 1 | 61.82 | 1.2513 |

Tabla 3.25: Parámetros desfavorables empleados para la clasificación con FCN-32s

| CamposSinVegetación (IoU) | CamposVerdes (IoU) | SotoBosque (IoU) |
|---|---|---|
| 0 | 0.0035509 | 0.52544 |

Tabla 3.26: Métricas IoU obtenidas para el ejemplo FCN-32s

Los resultados de clasificación son los mostrados en la figura 3.17:

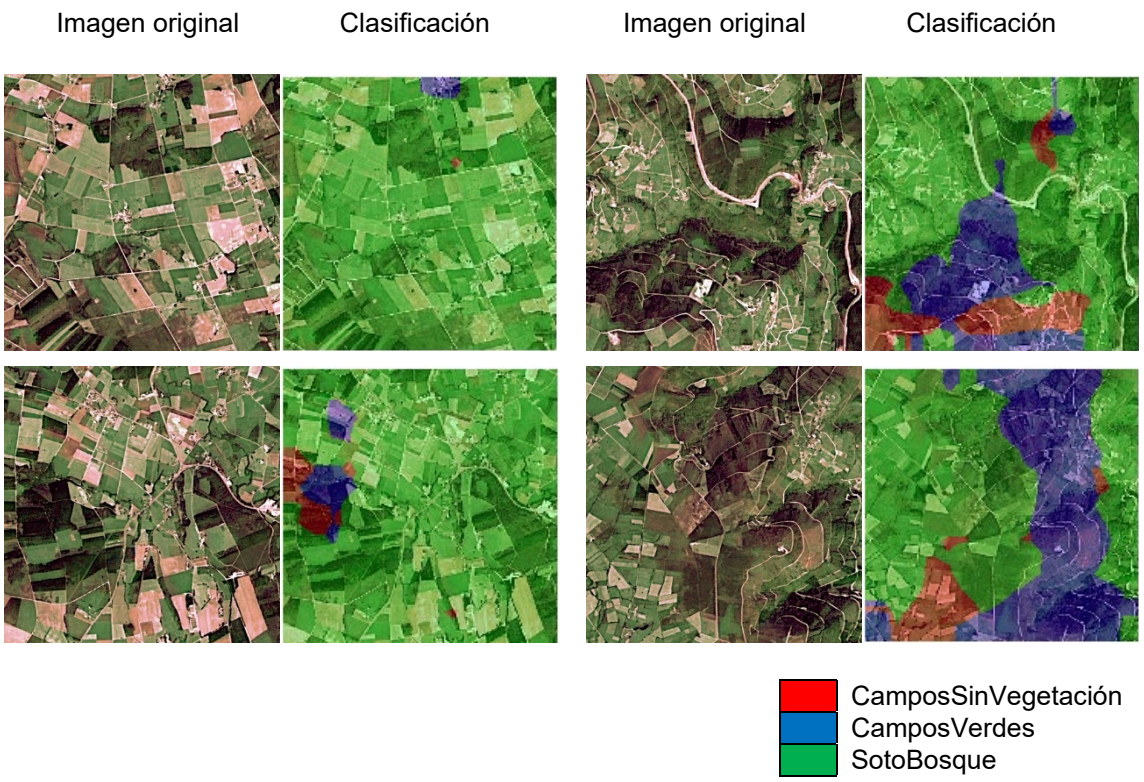



Figura 3.17: Clasificación desfavorable de imágenes con FCN-32s

De nuevo, dada la baja calidad de los resultados, no es necesario el cómputo y análisis de la matriz de confusión.

| | solverName | InitialLearnRate | LearnRateSchedule | L2Regularization | MaxEpochs | MiniBatchSize | LearnRateDrop Period | LearnRateDrop Factor | CamposSinVegetacion (IoU) | CamposVerdes (IoU) | SotoBosque (IoU) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 8s / 16s / 32s | - | - | - | - | - | 4 | - | - | - | - | - |
| 8s | SGDM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.1 | 0 | 0.015115 | 0.7447 |
| 8s | SGDM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0 | 0.05356 | 0.50544 |
| 8s | SGDM | 1e-3 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0 | 0 | 0 |
| 8s | SGDM | 1e-5 | piecewise | 0.1 | 30 | 1 | 10 | 0.3 | 0 | 0.034253 | 0.62605 |
| 8s | SGDM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.1 | 0 | 0.044591 | 0.59767 |
| 8s | SGDM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0 | 0.032483 | 0.62217 |
| 8s | SGDM | 1e-30 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0.0030369 | 0.04328 | 0.15651 |
| 16s | SGDM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0 | 0.052281 | 0.16483 |
| 16s | SGDM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0 | 0.045442 | 0.21099 |
| 32s | SGDM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0 | 0.0035509 | 0.52544 |
| 32s | SGDM | 1e-5 | piecewise | 0.005 | 30 | 1 | 10 | 0.3 | 0 | 0.0051027 | 0.48451 |
| 8s | ADAM | 1e-5 | none | 0.005 | 30 | 1 | 10 | 0.3 | 0.00016565 | 0.055342 | 0.36848 |

3.27: Métricas de evaluación IoU para FCNNet

Las imágenes aéreas nos proporcionan una capacidad única de mirar hacia abajo y observar la tierra desde arriba. Se están empleando para la localización de áreas dañadas después de desastres naturales, medir la deforestación, detectar lugares con ruinas arqueológicas saqueadas, mapear zonas de difícil acceso con riesgo de incendio además de otras muchas situaciones de uso actual aún sin explotar. Es fundamental comprender el potencial impacto que este tipo de imágenes pueden tener en nuestra comprensión del mundo. Su inmensa y creciente cantidad presenta un gran desafío debido a que es necesario obtener la información necesaria de todos estos datos por ello es necesario crear una herramienta efectiva para tal fin: permitir que la tecnología vea lo que nosotros no podemos.

En el siguiente capítulo se evaluarán de manera conjunta los distintos experimentos realizados para las imágenes aéreas propuestas con el fin de realizar una comparativa con la cual determinar el más efectivo y, posiblemente el método con el que partir para el desarrollo futuro de una herramienta que permita mejorar los estudios actuales en este ámbito.

# CAPÍTULO 4

## Análisis comparativo de los modelos de segmentación semántica: conclusiones y trabajo futuro

A diario se capturan enormes volúmenes de imágenes aéreas. Esta cantidad descomunal hace que su interpretación manual sea inadmisible por lo que es necesario utilizar métodos de visión artificial si se quiere abordar esta problemática de tal forma que sea posible hacer uso de los datos disponibles. Como se ha tratado en los capítulos anteriores, mediante la asignación de una clase semántica a cada píxel, es decir, transformar los datos sin procesar en un mapa ráster semánticamente significativo, es posible establecer un mapeo automático para dichas imágenes. Esta ha sido la herramienta de aprendizaje supervisado del que se ha ocupado este trabajo mediante la cual es posible crear una función capaz de predecir el valor que le corresponde a un determinado objeto de entrada tras la visualización de unos ejemplos dados, en otras palabras, determinar, en cada píxel de la imagen, la etiqueta de clase más probable de un conjunto finito de etiquetas posibles, correspondientes a las categorías de objetos deseadas en el mapa. Dicha supervisión con datos de entrenamiento anotados por personas especialistas es esencial para inyectar las definiciones de clases concretas pero este proceso es costoso ya que debe repetirse ante cualquier cambio de ubicación geográfica, características del sensor o las condiciones temporales de la toma de la imagen, resultando en un proyecto dificultosamente escalable.

Partiendo del ejemplo propuesto en el capítulo anterior, este estudio utiliza el aprendizaje profundo para contrastar la eficacia de los distintos métodos de segmentación semántica con el propósito de comprender las imágenes aéreas de forma rápida, repetible y a cualquier escala. A continuación se tratará esta cuestión además de presentar diversos estudios de investigación hasta la fecha donde se parte de la segmentación semántica para el tratamiento de imágenes aéreas con distintos fines y, finalmente exponer las conclusiones y trabajo futuro.

## 4.1 Comparativa de los métodos de segmentación semántica para la clasificación del entorno de Abadín

Con el paso del tiempo, las CNN han demostrado un rendimiento impresionante y se han convertido rápidamente en un referente para la segmentación semántica. Sin embargo, una de las principales desventajas es que requieren una gran cantidad de datos, lo que agrava el eterno cuello de botella de la clasificación supervisada para obtener suficientes datos de entrenamiento anotados. En los distintos experimentos realizados se ha trabajado con un número reducido de imágenes de entrada que ha limitado el rendimiento de algunos modelos de segmentación semántica, como el caso de FCNNet, ampliamente utilizado en proyectos de investigación para abordar fines similares como el que se presenta en este trabajo. Estudiar la respuesta de los distintos modelos ante esta situación también resulta de interés ya que, se han obtenido resultados con una precisión del 85% en algunos de ellos solamente con 20 fotografías de entrada empleadas para la fase de entrenamiento.

Como se ha tratado en el capítulo 3, se prueban varias arquitecturas de redes neuronales profundas para deducir las etiquetas del conjunto de experimentos. El primer modelo empleado es DeepLabv3, cuya arquitectura toma la forma de codificador – decodificador. El codificador consta de una CNN entrenada para codificar mapas de características de la imagen de entrada. El segundo modelo, SegNet, basa su arquitectura en las primeras 13 capas convolucionales de la red VGG-16 para emplearlas en su red de codificador además de la inserción de deconvoluciones en el decodificador. El tercero, UNet, con arquitectura de codificador – decodificador formado en su codificador de una ruta de contracción que captura el contexto en la imagen y, en su decodificador, una ruta de expansión simétrica que permitir obtener una ubicación precisa. Finalmente, el cuarto modelo, FCNNet se preinicializa utilizando capas y pesos de la red VGG-16. Se tratan sus tres variantes, FCN-8s, FCN-16s y FCN-32s.

El resultado más interesante de todos los experimentos realizados se centra en el modelo DeepLabv3, con su arquitectura basada en ResNet18, Xception e InceptionResnetV2 con una validación de la precisión del entrenamiento superior al 85% y unas pérdidas alrededor de 0.4 correspondiendo estos resultados en una notable clasificación. En la figura 4.1 se muestra este modelo frente a los otros tres de estudio. Esta imagen es representativa del conjunto de

imágenes utilizadas, por lo que todo lo que se comenta sobre ella sirve, en términos generales, para el resto.

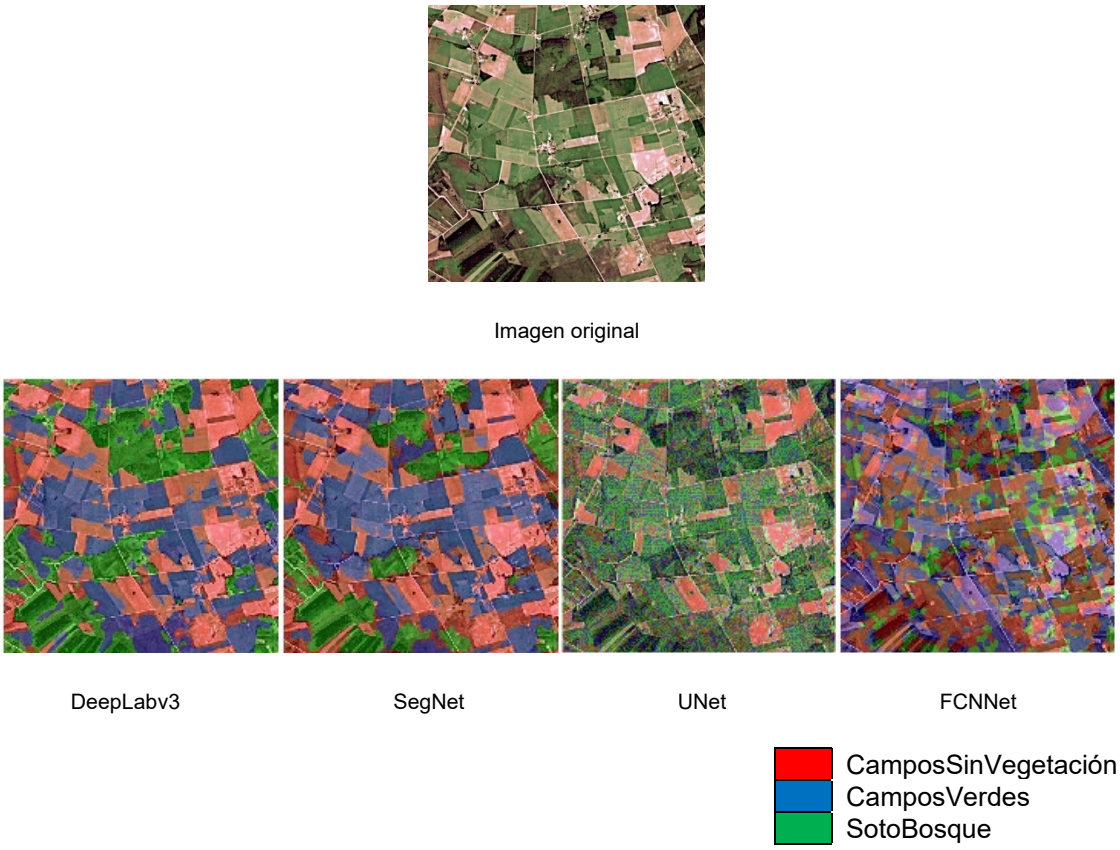



Figura 4.1: Comparativa de los resultados de clasificación

Como se observa en la imagen 4.1, aunque con SegNet también se han obtenido soluciones favorables estas han resultado inferiores a las de DeepLabv3, la validación de la precisión en torno al 84% y las pérdidas en 0,5 con unos índices IoU cercanos a los del anterior, pero desembocando en una peor segmentación. La etiqueta *SotoBosque* no se ha clasificado eficientemente, se observan zonas con abundante vegetación en la imagen original, correspondiente a zonas de *SotoBosque* clasificadas como *CamposSinVegetacion*. Esto se debe a una mala predicción de límites semánticos entre clases. Por otro lado, en ambos modelos, la clasificación de *CamposVerdes* y *CamposSinVegetación* resulta confusa en determinadas zonas provocado por el reducido número de imágenes empleadas para el entrenamiento con un *dataset* escaso.

Cabe destacar que con DeepLabv3 se crea una red con pesos inicializados desde una red de clasificación de imágenes ya capacitada para la extracción de determinadas características con el fin de emplearla como punto de partida para el aprendizaje en una nueva tarea. Esto facilita, en cierta medida, el desarrollo de los experimentos propuestos para este tipo de modelo debido

a que es posible adaptarse a la situación, por ejemplo, ResNet18 es una red que se ajusta bien a aplicaciones con recursos de procesamiento limitados. Esto se ha visto reflejado en los numerosos entrenamientos donde el tiempo de iteración y predicción dependen directamente del hardware empleado y del tamaño del *miniBatch* que se utilice, pero este último también se ha visto limitado por la GPU pudiendo repercutir en la precisión de la clasificación. DeepLabv3 ha sido el único modelo con el que su pudo llevar a cabo algún entrenamiento con un *miniBatchSize* de 4, en cambio, SegNet, UNet y FCNNet en sus distintas variantes no lo permitieron. Este ha sido un factor clave para la elección de DeepLabv3 como el modelo óptimo frente al caso propuesto con unas características condicionadas.

Por otro lado, FCNNet ha sido la red más exigente en materia de hardware, imposibilitando la realización de muchos entrenamientos ya que, incluso con una gráfica dedicada de 4Gb, la memoria disponible no ha sido suficiente. Este factor junto con las pocas imágenes empleadas para el entrenamiento ha causado que este tipo de red no sea válida para afrontar el caso práctico. Con UNet la respuesta tampoco ha sido favorable aunque se han podido clasificar algunas zonas correspondientes a *CamposSinVegetación*. SegNet, por el contrario, empleando un valor de *miniBatchSize* igual a 1 ha sido efectivo en determinadas situaciones. Es necesario mencionar que estas redes no están pensadas para abordar la tarea presentada sino que su creación partió en base del uso de la segmentación semántica en otros campos, por ejemplo, SegNet surge con el fin de comprender las escenas de carretera, desde la modelación de la apariencia y la forma hasta la relación espacial existente entre las distintas clases que la forman por lo que será más efectiva en el reconocimiento de objetos predefinidos y, UNet nace ante la necesidad de conocer la ubicación de los objetos en las en las imágenes biomédicas, alejándose también de la fotografía aérea del entorno natural.

## 4.2 Estudios en el campo de la segmentación semántica para el tratamiento de imágenes aéreas

### 4.2.1 Segmentación semántica de imágenes aéreas con un conjunto de CNN

Marmanis et al (2016) presentaron un enfoque de aprendizaje profundo para la segmentación semántica de imágenes aéreas de muy alta resolución. Estas muestran una problemática adicional debido a que, por un lado, su resolución espectral es intrínsecamente más baja, por otro lado, los objetos pequeños y la textura de la superficie a pequeña escala se vuelven visibles. En conjunto, esto conduce a una alta variabilidad dentro de la clase de las intensidades de imagen y, al mismo tiempo, a bajas diferencias entre clases. “Diseñamos un FCN que toma como entrada los datos de intensidad y rango y, con la ayuda de una deconvolución y reciclaje agresivo de las primeras capas de la red, los convierte en una clasificación de píxeles a resolución completa” (Marmanis et al., 2016). En la red propuesta la parte más complicada del aprendizaje se centra en las capas de deconvolución. Los autores inicializan los pesos de muestreo

ascendente con coeficientes de interpolación bilineal empleando la supervisión profunda pero el uso de estas capas provoca que se ralentice la propagación hacia atrás, lo que conlleva a aumentar el número de iteraciones de entrenamiento. Esto se debe a que las no linealidades de ReLU no están delimitadas desde arriba, por lo que no hay una saturación incorporada que detenga la formación de picos. Se insertaron capas de normalización de la respuesta local (Local Response Normalisation, LRN) en las etapas donde ocurría este efecto resolviendo de tal forma el problema en cuestión. “En nuestra experiencia, una buena práctica con CNN grandes y previamente entrenados es el entrenamiento gradual, comenzando desde los niveles más profundos. Se puede suponer que las características de bajo nivel, aunque no están completamente optimizadas para la tarea en cuestión, ya son razonables, por lo que primero las sujetamos y solo actualizamos las capas profundas de la red cerca de la salida, que inicialmente están sintonizadas con la nomenclatura de clase completamente diferente de la tarea de preentrenamiento. Cuando la pérdida se aplana, o después de un número fijo de iteraciones, se agregan más capas, hasta que finalmente se optimiza la red completa. Esto acelera enormemente el entrenamiento” (Marmanis et al., 2016). De esta forma se demuestra que un conjunto de varias redes alcanza resultados excelentes en datos desafiantes, como el punto de referencia de etiquetado semántico ISPRS (International Society for Photogrammetry and Remote Sensing), utilizando solo los datos sin procesar como entrada.

### 4.2.2 Segmentación semántica de imágenes aéreas con redes neuronales convolucionales barajadas

Chen et al. (2018) proponen el empleo de redes neuronales convolucionales barajadas (shuffling convolutional neural networks, SCNN) para llevar a cabo la segmentación semántica de imágenes aéreas periódicamente. Este enfoque complementa los métodos de segmentación semántica de imágenes aéreas que existen en la actualidad. “Proponemos una versión sencilla y una versión más profunda de este método, ambas expertas en detectar objetos pequeños. Además, proponemos un método llamado “mejora del campo de visión” (Field of View Enhancement, FoV-Enhancement*)* que puede mejorar las predicciones posibilitando su aplicación a varias redes. Los resultados finales se mejoran aún más mediante un método de conjunto que promedia los mapas de puntuación generados por los modelos en diferentes puntos de control de la misma red” (Chen et al., 2018).

El campo de visión es fundamental para una buena predicción debido a que ante un campo demasiado pequeño los resultados pueden ser deficientes debido a la falta de información contextual lo que conlleva a una información insuficiente que imposibilita distinguir objetos con características de textura similares. La predicción de píxeles marginales, cercanos a los bordes de los parches, en comparación con los del centro, pueden resultar en predicciones inexactas. En base a este supuesto, los autores proponen el método llamado FoV-Enhancement que trata de mejorar esta cuestión. “Primero, los tamaños de los parches introducidos en las redes se

amplían al hacer inferencias. En segundo lugar, los parches se recortan de las imágenes de manera superpuesta. La predicción final de un píxel en un área superpuesta está determinada por el parche más cercano. Aunque es simple, es efectivo" (Chen et al., 2018).

### 4.2.3 Aprendizaje profundo de múltiples tareas para la segmentación semántica de imágenes aéreas regularizadas geográficamente

Volpi y Tuia (2018) exponen un método para la segmentación semántica de imágenes aéreas cuya finalidad consiste en el aprendizaje de verosimilitudes de clase semántica, límites semánticos entre clases y características visuales superficiales a profundas, cada una modelada por una arquitectura de red neuronal convolucional multitarea. "Combinamos esta información de abajo hacia arriba con la regularización espacial de arriba hacia abajo codificada por un modelo de campo aleatorio condicional que optimiza el espacio de etiquetas en una jerarquía de segmentos con restricciones relacionadas con relaciones de pares estructurales, espaciales y dependientes de datos entre regiones. La estrategia propuesta ofrece un marco flexible y basado en principios para incluir varias fuentes de información visual y estructural, al tiempo que permite diferentes grados de regularización espacial que tienen en cuenta los antecedentes sobre las estructuras de salida esperadas" (Vopi y Tuia (2018). De esta forma, se modelan dos aspectos clave:

- Extracción de información sobre las relaciones de entrada locales mediante la combinación de CNN con el propósito de predecir límites semánticos entre las diferentes clases. Para este fin se construye un árbol de segmentación utilizando los límites semánticos predichos por la CNN. Dicho árbol representa la jerarquía de regiones que van desde el nivel más bajo definido por grupos de píxeles hasta el nivel más alto que engloba toda la escena.
- Las relaciones de salida entre regiones, en la que combinan la información dentro de cada región de manera jerarquizada. Se basa en los campos aleatorios condicionales (Conditional Random Field, CRF), modelo estocástico empleado comúnmente para etiquetar y segmentar secuencias de datos o extraer información de documentos. De esta forma se pretende reducir la complejidad de los mapas de píxeles mediante el análisis semántico y espacial de las regiones de las que se compone la imagen.

### 4.2.4 Red híbrida de atención múltiple para la segmentación semántica en imágenes aéreas

Niu et al. ( 2020) trataron el desafío que supone la comprensión de imágenes de teledetección mediante la segmentación semántica en imágenes aéreas. Los autores parten de trabajos de investigación anteriores donde se demuestra que con el uso de métodos basados en la atención es posible capturar dependencias de largo alcance y reconstruir aún más los mapas de características, a pesar de ello, estos se ven limitados por la perspectiva de la atención espacial

debido a la gran complejidad computacional del mecanismo de autoatención que supone por lo que es poco probable que se modelen efectivamente las interdependencias semánticas entre cada par de píxeles de datos de teledetección de espectros complejos. “En este trabajo, proponemos un marco novedoso basado en la atención llamado Red de Atención Múltiple Híbrida (Hybrid Multiple Attention Network, HMANet) para capturar de manera adaptativa las correlaciones globales desde la perspectiva del espacio, canal y categoría de una manera más efectiva y eficiente. Concretamente, un módulo de atención aumentada de clase (class augmented attention, CAA) integrado con un módulo de atención de canal de clase (class channel attention, CCA) se puede utilizar para calcular la correlación basada en categorías y recalibrar la información a nivel de clase. Además, presentamos un módulo de atención aleatoria de regiones (region shuffle attention, RSA) simple pero eficaz para reducir la redundancia de funciones y mejorar la eficiencia del mecanismo de auto atención a través de representaciones regionales” (Niu et al., 2020).

### 4.2.5 Mejora de la segmentación semántica de imágenes aéreas con neuronas inhibidoras

En las CNN, cada neurona ubicada en el mapa de características de salida recibe información de las neuronas en su campo receptivo. Ullah et al. (2020) parten de un concepto inspirado en la investigación neurobiológica por el que se propone agregar neuronas inhibidoras fuera del campo receptivo, lo que pretende mejorar el rendimiento de los distintos modelos. “Comenzamos con arquitecturas de redes profundas como VGG y ResNet, y proponemos un enfoque para agregar inhibición lateral en cada neurona de salida para reducir su impacto en sus vecinas, tanto en el ajuste fino de modelos preentrenados como en el entrenamiento desde cero” (Ullah et al., 2020). La inhibición lateral es la capacidad que posee una neurona excitada para reducir la actividad de sus vecinas (Grigorescu et al., 2003; Tjøstheim y Balkenius, 2019). Estas se denominan neuronas inhibidoras, localizadas inmediatamente fuera del campo receptivo de una neurona. El resultado es un campo inhibidor que rodea al campo receptivo. Al ignorar los contornos de la textura interna, es posible mejorar el análisis de esta, importante para el área de aplicación de la segmentación semántica. Los autores proponen este enfoque para mejorar los modelos preentrenados o entrenar un modelo nuevo. Presentan un trabajo inspirado en base a otros estudios (Fernandes et al., 2008; Fernandes et al., 2013) sobre neuronas inhibidoras en redes superficiales, restando la suma ponderada de neuronas inhibidoras de la suma ponderada de neuronas de campo receptivo. “Al adoptar este enfoque, podemos mejorar un modelo de aprendizaje profundo (previamente entrenado o nuevo), concretamente VGG16 sin cambios importantes en el modelo, que funciona tanto en implementaciones de CPU como de GPU. Inicialmente, trabajamos con un conjunto de datos pequeño para encontrar un enfoque óptimo y luego lo usamos como una herramienta para ajustar los modelos profundos existentes que se pueden emplear para problemas más complejos con datos de entrenamiento limitados, y demostrarlo en la tarea de segmentación semántica de imágenes aéreas. Esto se hace mediante

el uso de pesos modelo preentrenados en nuestros bloques convolucionales inhibitorios propuestos" (Ullah et al., 2020).

## 4.3 Conclusiones

A pesar de décadas de investigación, el tratamiento de la segmentación semántica de fotografías aéreas para la creación de mapas de cualquier índole sigue siendo un desafío. La cartografía precisa de la cobertura y el uso del suelo es un campo de investigación activo que avanza en paralelo a los desarrollos en sensores, sistemas de adquisición y los algoritmos de procesamiento de datos.

La segmentación semántica para este tipo de imágenes asigna una categoría de cobertura terrestre a cada píxel resultando en una tarea compleja debido a las grandes diferencias en la apariencia del entorno terrestre. Las CNN supusieron un gran avance en esta materia, ofreciendo un aprendizaje automático de extremo a extremo por el que se obtienen empíricamente resultados excelentes, pero su ajuste para una tarea concreta requiere cierto cuidado. De hecho, los diferentes trabajos reseñados en la sección 4.2, así lo ponen de manifiesto. Queda patente la necesidad de diseñar estrategias, que complementen la mera segmentación semántica píxel a píxel, añadiendo mecanismos de mejora, que entre otras cosas tienen en cuenta las vecindades de los píxeles más allá de la exclusiva información a nivel individual. Para una segmentación exitosa en el tipo de imágenes tratadas en este trabajo son necesarios modelos visuales que permitan suprimir la ambigüedad de la apariencia local al comprender la organización espacial de las clases semánticas. Se emplean CNN para proporcionar probabilidades aproximadas de clases además de predecir límites semánticos entre las diferentes clases que coinciden con los bordes naturales de la imagen además de corresponder a cambios de etiquetas.

Tras emplear distintas arquitecturas de segmentación semántica sobre un mismo conjunto de imágenes con el fin de analizar su comportamiento, es posible concluir que el presente estudio se ve limitado por el uso de un *dataset* pequeño y un hardware reducido. Aunque son destacables los resultados obtenidos con DeepLabv3 ante esta cantidad de imágenes, hay mucho margen de mejora. En general se observan límites entre distintas zonas clasificadas poco nítidos, especialmente en la clasificación obtenida con SegNet, esto se debe en parte a la mayor incertidumbre de ubicación de las características después de la agrupación y convolución. Otra razón puede darse por la inexactitud inherente de los datos de entrenamiento anotados ya que se trata de una etiquetación manual con un posible margen de error dado que los límites son particularmente difíciles de definir con precisión. Una posible solución para hacer frente a esta problemática sería la de desplazar estos límites unos píxeles, de forma aleatoria, hacia dentro o fuera con el objetivo de hacer que el sistema aprenda esa incertidumbre en su localización.

Por otro lado, para realizar un estudio más exhaustivo sobre los distintos modelos de segmentación semántica frente a imágenes aéreas, es necesario que se proporcionen suficientes datos de entrenamiento para que redes como FCNNet puedan resultar efectivas. Gracias a una de las características de las CNN esta tarea sería fácilmente afrontable ya que, por un lado explotan la invariancia de desplazamiento de las señales de imagen y, por otro lado, son capaces de paralelizar fácilmente y ejecutar el proceso en la GPU, lo que permite entrenar a partir de millones de imágenes en una sola máquina, por ello, se propone crear un *dataset* con miles de imágenes para la fase de entrenamiento. Ante esta situación los resultados del presente trabajo cambiarían considerablemente.

Establecer un modelo de segmentación semántico óptimo para el fin presentado queda fuera del alcance del máster ya que corresponde a un proceso de investigación posterior que precisa posiblemente de la creación de una nueva arquitectura de CNN para afrontar de manera óptima el tipo de imágenes en tratadas. Y ello sin tener la certeza absoluta de que un nuevo modelo podría ser suficiente, máxime teniendo en cuenta que los modelos analizados en el presente trabajo han sido y están siendo ampliamente utilizados.

## 4.4 Trabajo futuro

El presente trabajo plantea el punto de partida para un proceso de investigación posterior más completo ya que el tipo de imágenes tratadas sigue siendo un reto para la visión por computador. Como se ha comentado, es posible ampliarlo empleando un mayor número de imágenes para el entrenamiento además de probar más combinaciones de parámetros de aprendizaje lo que conlleva a un mejor ajuste de la red y, por consiguiente, una mejor clasificación. Por otro lado, sería factible distinguir un mayor número de etiquetas para el proceso de entrenamiento, tales como: caminos, zonas urbanas, ríos, etc. Además, sería importante entrenar los modelos según la época del año debido a que el matiz del color de los cultivos o el entorno puede cambiar drásticamente.

Desde el punto de vista técnico, y en lo que afecta a los modelos de redes de segmentación semántica utilizados conviene señalar de cara al futuro la necesidad de experimentar bajo la combinación entre sí de diferentes parámetros de los modelos, concretamente en lo que respecta a los siguientes aspectos:

a) Diferentes métodos de optimización
b) Número de iteraciones y tamaños de los diferentes lotes (batches)
c) Definición de nuevas funciones de pérdida
d) Y tal vez, modificar la arquitectura de los modelos utilizados en el sentido de añadir capas intermedias adicionales, aunque bien es cierto que en este sentido no es posible determinar el alcance de la propuesta en cuanto al número de capas a introducir.

# Bibliografía


Azevedo, F. A., Carvalho, L. R., Grinberg, L. T., Farfel, J. M., Ferretti, R. E., Leite, R. E., Jacob Filho, W., Lent, R., y Herculano-Houzel, S. (2009). *Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain*. J. Comp. Neurol. 513, 532–541. doi: 10.1002/cne.21974

Badrinarayanan, V., Kendall, A., y Cipolla, R. (2017). *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(12), 2481–2495. doi:10.1109/tpami.2016.2644615

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*, Springer, NY, USA.

Bottou, L. (2010). *Large-Scale Machine Learning with Stochastic Gradient Descent*. COMPSTAT'2010, 177–186. doi:10.1007/978-3-7908-2604-3_16

Breiman, L. (2001). *Statistical modeling: the two cultures (with comments and a rejoinder by the author)*. Stat. Sci. 16, 199–231. doi:10.1214/ss/1009213726

Brendel, W., & Bethge, M. (2019). *Approximating cnns with bag-of-local-features models works surprisingly well on imagenet*. arXiv preprint arXiv:1904.00760

Chen, K., Fu, K., Yan, M., Gao, X., Sun, X., & Wei, X. (2018). *Semantic Segmentation of Aerial Images With Shuffling Convolutional Neural Networks*. IEEE Geoscience and Remote Sensing Letters, 15(2), 173–177. doi:10.1109/lgrs.2017.2778181

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., y Yuille, A. L. (2016). *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. doi:10.1109/tpami.2017.2699184

Chen, L.-C., Papandreou, G., Schroff, F., y Adam, H. (2017). *Rethinking Atrous Convolution for Semantic Image Segmentation*. arXiv:1706.05587

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., y Adam, H. (2018). *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. ArXiv.org. arXiv:1802.02611

Chollet, F. (2017). *Xception: Deep learning with depthwise separable convolutions*. (CVPR'17). arXiv:1610.02357v3

Ciresan, D. C., Gambardella, L. M., Giusti, A., Schmidhuber, J. (2012). *Deep neural networks segment neuronal membranes in electron microscopy images*. Proceedings of the 25th

International Conference on Neural Information Processing Systems - Volume 2. 2843–2851. Recuperado de https://www.researchgate.net/publication/267709055_Deep_Neural_Networks_Segment_Neuronal_Membranes_in_Electron_Microscopy_Images

Ciresan, D. C., Giusti, A., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2013). *Fast image scanning with deep max-pooling convolutional neural networks*. 2013 IEEE International Conference on Image Processing. doi:10.1109/icip.2013.6738831

Dai, J., He, K., & Sun, J. (2015). *Convolutional feature masking for joint object and stuff segmentation*. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2015.7299025

Dautenhahn, K. y Nehaniv, C. L. (2002). *Imitation in Animals and Artifacts*. MIT Press Cambridge, MA.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., y Fei-Fei, L. (2009) *ImageNet: A large-scale hierarchical image database*. 2009 IEEE Conference on Computer Vision and Pattern Recognition (ICCV), 248–255, doi:10.1109/CVPR.2009.5206848.

DiCarlo, J. J., Zoccolan, D., y Rust, N. C. (2012). *How does the brain solve visual object recognition?* Neuron, 73 (3), 415–434. Doi:10.1016/j.neuron.2012.01.010

Dosovitskiy, A., Fischer, P., Springenberg, Jost Tobias, Riedmiller, M., & Brox, T. (2014). *Discriminative Unsupervised Feature Learning with Exemplar Convolutional Neural Networks*. arXiv:1406.6909

Dujmovic, M., Malhotra, G., y Bowers, J. S. (2020*). What do adversarial images tell us about human vision?* eLife, 9. doi:10.7554/eLife.55978

Eigen, D., & Fergus, R. (2014). *Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-scale Convolutional Architecture*. 2015 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2015.304

Farabet, C., Couprie, C., Najman, L., & LeCun, Y. (2013). *Learning Hierarchical Features for Scene Labeling.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1915–1929. doi:10.1109/tpami.2012.231

Feldman, V. (2019). *Does learning require memorization? A short tale about a long tail*. arXiv, arXiv:1906.05271

Fernandes, B. J. T., Cavalcanti, G. D. C., & Ren, T. I. (2008). *Classification and Segmentation of Visual Patterns Based on Receptive and Inhibitory Fields*. *2008 Eighth International Conference on Hybrid Intelligent Systems*. doi:10.1109/his.2008.42

Fernandes, B. J. T., Cavalcanti, G. D. C., & Ren, T. I. (2013). *Lateral Inhibition Pyramidal Neural Network for Image Classification*. *IEEE Transactions on Cybernetics*, *43*(6), 2082–2092. doi:10.1109/tcyb.2013.2240295

Fire, A., y Zhu, S. (2013). *Using Causal Induction in Humans to Learn and Infer Causality from Video*. Annual Meeting of the Cognitive Science Society (CogSci). 35. Recuperado de https://escholarship.org/uc/item/4ng247kx

Freeman, P., Newell, A. (1971). *A model for functional reasoning in design*. 2nd international joint conference on Artificial intelligence. doi:10.5555/1622876.1622940

Fukushima, K. (1980). *Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position*, Biological Cybernetics 36 (4) (1980) 193–202. doi:10. 1007/BF00344251

Funke, M. C., Borowski, J., Stosio, K., Brendel, W., Wallis, T. y Bethge. M. *Five Points to Check when Comparing Visual Perception in Humans and Machines*. arXiv:2004.09406

Gergely, G., Nádasdy, Z., Csibra, G., y Bíró, S. (1995). *Taking the intentional stance at 12 months of age*. Cognition, 56(2), 165–193. doi:10.1016/0010-0277(95)00661-h

Géron. A., (2017), *Hands-On Machine Learning with Scikit-Learn and TensorFlow* (Chapter 15: Autoencoders), O'Reilly, CA, USA.

Ghiasi, G., & Fowlkes, C. C. (2016). *Laplacian Pyramid Reconstruction and Refinement for Semantic Segmentation*. Computer Vision – ECCV 2016, 519–534. doi:10.1007/978-3-319-46487-9_32

Gibson, J. J. (1966). *The senses considered as perceptual systems*. Houghton Mifflin.

Gomez-Marin, A., and Ghazanfar, A. A. (2019). *The life of behavior*. Neuron 104, 25–36. doi:10.1016/j.neuron.2019.09.017

Gresswell, R. E. (1999). *Fire and Aquatic Ecosystems in Forested Biomes of North America*. Transactions of the American Fisheries Society, 128(2), 193–221. doi:10.1577/1548-8659(1999)128<0193:FAAEIF>2.0.CO;2

Grigorescu, C., Petkov, N., y Westenberg, M. A. (2003). *Contour detection based on nonclassical receptive field inhibition*. ResearchGate; Institute of Electrical and Electronics Engineers. doi:10.1109/TIP.2003.814250

Guo, Y., Liu, Y., Georgiou, T., & Lew, M. S. (2017). *A review of semantic segmentation using deep neural networks*. International Journal of Multimedia Information Retrieval, 7(2), 87–93. doi:10.1007/s13735-017-0141-z

Hasson, U., Nastase, S.A. y Goldstein, A. (2019). *Direct Fit to Nature: An Evolutionary Perspective on Biological and Artificial Neural Networks.* doi:10.1016/j.neuron.2019.12.002

Haun, D. B., Jordan, F. M., Vallortigara, G., y Clayton, N. S. (2010). *Origins of spatial, temporal and numerical cognition: Insights from comparative psychology*. In (Vol. 14, 552–560). doi:10.1016/j.tics.2010.09.006

He, K., Zhang, X., Ren, S., y Sun, J. (2016). *Deep residual learning for image recognition*. In 2016 ieee conference on computer vision and pattern recognition (CVPR) 770–778. doi:10.1109/CVPR.2016.90

He, K., Zhang, X., Ren, S., y Sun, J. (2016). *Deep Residual Learning for Image Recognition*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.90

Holschneider, M., Kronland-Martinet, R., Morlet, J., & Tchamitchian, Ph. (1989). *A Real-Time Algorithm for Signal Analysis with the Help of the Wavelet Transform.* Inverse Problems and Theoretical Imaging, 286–297. doi:10.1007/978-3-642-75988-8_28

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H. (2017). *MobileNets: Efficient convolutional neural networks for mobile vision applications*. arXiv:1704.04861.

Hubel, D. H., y Wiesel, T. N. (1968). *Receptive fields and functional architecture of monkey striate cortex*. The Journal of Physiology, 195(1), 215–243. doi:10.1113/jphysiol.1968.sp008455

Ibtehaz, N., Rahman, M. S. (2019). *Multiresunet : Rethinking the u-net architecture for multimodal biomedical image segmentation*, Neural Networks 121, 74–87. doi:10.1016/j.neunet. 2019.08.025.

Jaccard, P. (1908). *Nouvelles recherches sur la distribution florale*. Bull. Soc. Vaudoise Sci. Nat. 44, 223-270. doi:10.5169/seals-268384

Jaderberg, M., Czarnecki, W. M., Dunning, I., Marris, L., Lever, G., Castañeda, A.G., Beattie, C., Rabinowitz, N. C., Morcos, A.S., Ruderman, A. (2019). *Human-level performance in 3D multiplayer games with population-based reinforcement learning*. Science 364, 859–865. doi:10.1126/science.aau6249

Kandel, E. R., Schwartz, J. H., Jessell, T.M., Siegelbaum, S., and Hudspeth, A. J. (2012). *Principles of Neural Science*, Fifth Edition (McGraw-Hill).

Kingma, D. P, & Ba, J. (2014). *Adam: A Method for Stochastic Optimization*. arXiv:1412.6980

Kingma, D.P., Ba, J.L. (2014). *Adam: A method for stochastic optimization*. 3rd International Conference on Learning Representations (ICLR 2015), pp. 1-15. arXiv:1412.6980v9.

Koehler, O. (1943). *Zaehl-versuche an einem kolkraben und vergleichsversuche an menschen. Zeitschrift für Tierpsychologie*, 5 (3), 575–712. doi:10.1111/j.1439-0310.1943.tb00665.x

Krähenbühl, P., & Koltun, V. (2012). *Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials*. arXiv:1210.5644

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L. D. (1989). *Backpropagation applied to handwritten zip code recognition*, Neural Comput. 1 (4) 541–551. doi:10.1162/neco.1989.1.4.541.

*Li, R., Liu, W., Yang, L., Sun, S., Hu, W., Zhang, F., y Li, W. (2017). DeepUNet: A Deep Fully Convolutional Network for Pixel-Level Sea-Land Segmentation.* IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, *11(11), 3954–3962. doi:10.1109/jstars.2018.2833382*

Lin, G., Shen, C., Anton, & Reid, I. (2015). *Efficient piecewise training of deep structured models for semantic segmentation*. arXiv:1504.01013

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). *Feature Pyramid Networks for Object Detection*. IEEE Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2017.106

Liu, C., Chen, L.-C., Schroff, F., Adam, H., Hua, W., Yuille, A., & Fei-Fei, L. (2019). *Auto-DeepLab: Hierarchical Neural Architecture Search for Semantic Image Segmentation*. arXiv:1901.02985

Liu, H., Zhang, C., Zhu, Y., Jiang, C., y Zhu, S.-C. (2019). *Mirroring without Overimitation: Learning Functionally Equivalent Manipulation Actions*. AAAI Conference on Artificial Intelligence, 33, 8025–8033. doi:10.1609/aaai.v33i01.33018025

Liu, W., Rabinovich, A., Berg, A., C. (2015). *Parsenet: Looking wider to see better*. arXiv:1506.04579.

Liu, Z., Li, X., Luo, P., Loy, Chen Change, & Tang, X. (2015*). Semantic Image Segmentation via Deep Parsing Network*. arXiv:1509.02634

Long, J., Shelhamer, E., Darrell, T. (2014) *Fully convolutional networks for semantic segmentation*, IEEE Trans. Pattern Anal. Mach. Intell. 39 (4) 640–651. doi:10.1109/TPAMI.2016.2572683.

Malament, D. B. (1977). *The class of continuous timelike curves determines the topology of spacetime*, Journal of mathematical physics 18 (7), 1399–1404. doi:10.1063/1.523436

Marcus, G. (2018). *Deep learning: a critical appraisal*. arXiv, arXiv:1801.00631

Marmanis, D., Wegner, J., Silvano Galliani, & Uwe Stilla. (2016). *Semantic segmentation of aerial images with an ensemble of CNNS*. doi:10.5194/isprs-annals-III-3-473-2016

Marr, D. (1982). *Vision : a computational investigation into the human representation and processing of visual information*. Henry Holt and Co., Inc., New York, NY, USA, 1982.

Marr, D. (1985). *El paradigma de los algoritmos en la visión por computador.* Alianza Editorial.

Mathworks (2017). *Evaluate Semantic Segmentation Results*. Recuperado de https://es.mathworks.com/help/vision/ref/semanticsegmentationmetrics.html

McClelland, J. L., and Rogers, T. T. (2003). *The parallel distributed processing approach to semantic cognition*. Nat. Rev. Neurosci. 4, 310–322. doi:10.1038/nrn1076

McHugh, C. W., & Kolb, T. E. (2003). *Ponderosa pine mortality following fire in northern Arizona*. doi:10.1071/WF02054

Minshall, G. W., Brock, J., Andrews, D. A., & Robinson, C. T. (2001). *Water quality, substratum and biotic responses of five central Idaho (USA) streams during the first year following the Mortar Creek fire*. June 2001International Journal of Wildland Fire 10(2):185-199. doi:10.1071/WF01017

Mostajabi, M., Yadollahpour, P., & Shakhnarovich, G. (2014). *Feedforward semantic segmentation with zoom-out features*. arXiv:1412.0774

Murphy, Kevin P. (2012). *Machine Learning: A Probabilistic Perspective*. Cambridge: MIT Press.

Naturaspain. (2021). *Que ver en Abadín, naturaleza de Abadín, Lugo, flora y fauna en Abadín, espacios protegidos*. Recuperado de Naturaspain. http://www.naturaspain.com/naturaleza-flora-y-fauna-en-el-municipio-de-abadin.html

Nelson, K. (1974). *Concept, word, and sentence: Interrelations in acquisition and development*. Psychological Review, 81(4), 267–285. doi:10.1037/h0036592

Niu, R., Sun, X., Tian, Y., Diao, W., Chen, K., & Fu, K. (2020). *Hybrid Multiple Attention Network for Semantic Segmentation in Aerial Images*. arXiv:2001.02870

Noh, H., Hong, S., & Han, B. (2015). *Learning Deconvolution Network for Semantic Segmentation*. arXiv:1505.04366

Noh, H., Hong, S., & Han, B. (2015). *Learning Deconvolution Network for Semantic Segmentation*. arXiv:1505.04366

Pajares, G., Herrera, P.J., Besada, E. (2021). *Aprendizaje Profundo*. RC-Libros, Madrid.

Pal, N. R., Pal, S. K. (1993) *A review on image segmentation techniques*, Pattern Recognition 26 (9), 1277 – 1294. doi:10.1016/0031-3203(93)90135-J

Papandreou, G., Chen, L.-C., Murphy, K., & Yuille, A. L. (2015). *Weakly- and Semi-Supervised Learning of a DCNN for Semantic Image Segmentation*. arXiv:1502.02734

*Paszke, A., Chaurasia, A., Kim, S., y Culurciello, E. (2016). ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. arXiv:1606.02147*

Pedro, & Collobert, R. (2014). *Recurrent Convolutional Neural Networks for Scene Parsing*. arXiv:1306.2795

Peng, C., Zhang, X., Yu, G., Luo, G., & Sun, J. (2017). *Large Kernel Matters — Improve Semantic Segmentation by Global Convolutional Network*. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi: 10.1109/cvpr.2017.189

Pohlen, T., Hermans, A., Mathias, M., & Leibe, B. (2016). *Full-Resolution Residual Networks for Semantic Segmentation in Street Scenes*. arXiv:1611.08323

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., y Sutskever, I. (2019*). Language models are unsupervised multitask learners*. Recuperado de https://openai.com/blog/better-language-models

Ranzato, M., Huang, F. J., Boureau, Y-Lan., & LeCun, Y. (2007). *Unsupervised Learning of Invariant Feature Hierarchies with Applications to Object Recognition*. 2007 IEEE Conference on Computer Vision and Pattern Recognition. doi:10.1109/cvpr.2007.383157

Ren, S., He, K., Girshick, R., & Sun, J. (2015). *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*. arXiv:1506.01497

Richards, B. A., Lillicrap, T. P., Beaudoin, P., Bengio, Y., Bogacz, R., Christensen, A., Clopath, C., Costa, R. P., de Berker, A. y Ganguli, S. (2019). *A deep learning framework for neuroscience. Nat. Neurosci*. 22, 1761–1770. doi:10.1038/s41593-019-0520-2

Robb A. A. (2014). *Geometry of time and space*, Cambridge University Press.

Robb, A. A. (1911) *Optical geometry of motion: A new view of the theory of relativity*. W. Heffer.

Romanes, G. J. (1883). *Animal intelligence*. D. Appleton.

Ronneberger, O., Fischer, P., y Brox, T. (2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv:1505.04597

Ruder, S. (2016). *An overview of gradient descent optimization algorithms*. arXiv:1609.04747

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). *Learning representations by back-propagating errors*. Nature, 323(6088), 533–536. doi:10.1038/323533a0

Russell, J. S., Norvig, P. (2016) *Artificial intelligence: a modern approach*, Malaysia; Pearson Education Limited.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). *OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks*. arXiv:1312.6229

Seyedhosseini, M., Sajjadi, M., & Tasdizen, T. (2013). *Image Segmentation with Cascaded Hierarchical Models and Logistic Disjunctive Normal Networks*. 2013 IEEE International Conference on Computer Vision. doi:10.1109/iccv.2013.269

Shmueli, G. (2010). *To explain or to predict?* Stat. Sci. 25, 289–310. doi:10.1214/10-STS330

Shu, T., Peng, Y., Fan, L., Lu, H., y Zhu, S.-C. (2017). *Perception of Human Interaction Based on Motion Trajectories: From Aerial Videos to Decontextualized Animations*. Topics in Cognitive Science, 10(1), 225–241. doi:10.1111/tops.12313

Shukla, N., He, Y., Chen, F., y Zhu, S-C, (2017). *Learning Human Utility from Video Demonstrations for Deductive Planning in Robotics*. PMLR, 448–457. Recuperado de http://proceedings.mlr.press/v78/shukla17a.html

Simonyan, K., & Zisserman, A. (2014). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv:1409.1556

Srinivas, S., Sarvadevabhatla, R. K., Mopuri, K. R., Prabhu, N., Kruthiventi, S. S. S., & Babu, R. V. (2016). *A Taxonomy of Deep Convolutional Neural Nets for Computer Vision*. Frontiers in Robotics and AI, 2. doi:10.3389/frobt.2015.00036

Stark, L., y Bowyer, K. (1991). *Achieving generalized object recognition through reasoning about association of function to structure*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 13, no. 10, 1097-1104. doi:10.1109/34.99242

Sun, K., Zhao, Y., Jiang, B., Cheng, T., Xiao, B., Liu, D., Mu, Y., Wang, X., Liu, W., & Wang, J. (2019). *High-Resolution Representations for Labeling Pixels and Regions*. arXiv:1904.04514

Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (n.d.). *On the importance of initialization and momentum in deep learning*. Recuperado de

Szegedy, C., Reed, S., Erhan, D., Anguelov, D., & Ioffe, S. (2014). *Scalable, High-Quality Object Detection*. arXiv:1412.1441

Taigman Y., Yang M., Ranzato M. y Wolf L. (2014). *DeepFace: Closing the Gap to Human-Level Performance in Face Verification*, IEEE Conference on Computer Vision and Pattern Recognition, pp. 1701-1708, doi:10.1109/CVPR.2014.220

Tjøstheim, T. A., & Balkenius, C. (2018). *Cumulative inhibition in neural networks*. Cognitive Processing, 20(1), 87–102. doi:10.1007/s10339-018-0888-z

Tomasello, M., y Call, J. (2008). *Assessing the validity of ape-human comparisons: A reply to boesch (2007)*. Journal of Comparative Psychology, 122 (4), 449-452. doi:10.1037/0735-7036.122.4.449

Ullah, I., Reilly, S., & Madden, M. G. (2020). *Enhancing Semantic Segmentation of Aerial Images with Inhibitory Neurons*. 2020 25th International Conference on Pattern Recognition (ICPR). doi:10.1109/icpr48806.2021.9413021

Vieira, S. A., Alves, L. F., Scaranello, M. A., Camargo, P. B., Santos, F. A. M., Joly, C. A., & Martinelli, L. A. (2010). *Forest structure and live aboveground biomass variation along an elevational gradient of tropical Atlantic moist forest (Brazil).* Forest Ecology and Management, 260(5), 679–691. doi:10.1016/j.foreco.2010.05.023

Volpi, M., & Tuia, D. (2018). *Deep multi-task learning for a geographically-regularized semantic segmentation of aerial images*. ISPRS Journal of Photogrammetry and Remote Sensing, 144, 48–60. doi:10.1016/j.isprsjprs.2018.06.007

von Neumann, J. (1955). *Method in the physical sciences*. In The Unity of Knowledge, L.G. Leary, ed. (Doubleday). Recuperado de http://www.worldcat.org/oclc/1346478

Wei, P., Liu, Y., Shu, T., y Song Chun Zhu. (2018). *Where and Why are They Looking?* Jointly Inferring Human Attention and Intentions in Complex Tasks. doi:10.1109/CVPR.2018.00711

Wolpert, D. H., and Macready, W. G. (1997). *No free lunch theorems for optimization*. IEEE Trans. Evol. Comput. 1, 67–82. doi:10.1109/4235.585893

Wyant, J. G., Omi, P. N., & Laven, R. D. (1986). *Fire Induced Tree Mortality in a Colorado Ponderosa Pine/Douglas-fir Stand*. Forest Science, 32(1), 49–59. doi:10.1093/forestscience/32.1.49

Xiong, C., Shukla, N., Xiong, W., Zhu, S-C. (2016). *Robot learning with a spatial, temporal, and causal and-or graph*. International Conference on Robotics and Automation (ICRA). doi:10.1109/ICRA.2016.7487364

Yarkoni, T., y Westfall, J. (2017). *Choosing prediction over explanation in psychology: lessons from machine learning*. Perspect. Psychol. Sci. 12, 1100–1122. doi:10.1177/1745691617693393

Yu, F., & Koltun, V. (2016). *Multi-Scale Context Aggregation by Dilated Convolutions*. arXiv:1511.07122

Zhang, Y., Ye, M., Gan, Y., Zhang, W. (2019). *Knowledge based domain adaptation for semantic segmentation, Knowledge-Based Systems*. doi:10.1016/j.knosys.2019.105444.

Zhao, H., Qi, X., Shen, X., Shi, J., & Jia, J. (2018). *ICNet for Real-Time Semantic Segmentation on High-Resolution Images*. arXiv:1704.08545

Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J. (2016) *Pyramid scene parsing Network*. arXiv:1612.01105.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., & Torr, P. H. S. (2015). *Conditional Random Fields as Recurrent Neural Networks*. 2015 IEEE International Conference on Computer Vision (ICCV). doi:10.1109/iccv.2015.179

Zhu, Y., Gao, T., Fan, L., Huang, S., Edmonds, M., Liu, H., Gao, F., Zhang, C., Qi, S., Wu, Y. N., Tenenbaum, J. B., y Zhu, S.-C. (2020). *Dark, Beyond Deep: A Paradigm Shift to Cognitive AI with Humanlike Common Sense. Engineering*, *6* (3), 310–345. doi:10.1016/j.eng.2020.01.011

Zhu, Y., Jiang, C., Zhao, Y., Terzopoulos, D., y Zhu, S.-C. (2016). *Inferring Forces and Learning Human Utilities from Videos*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). doi:10.1109/cvpr.2016.415