Segmentación semántica en imágenes aéreas de vegetación

Lou Minxi

Supervisado por:
Adrián Peñate Sánchez
Maria Cristina Benlliure Jiménez

Fecha
11/04/2024

Agradecimientos

A mi familia, que me ha apoyado durante todos estos años de la carrera.

A mis tutores, Adrián y Cristina, que me han guiado y animado durante el desarrollo del TFG.

A todos profesores encontrados en la escuela, que me han dado los conocimientos necesarios para enfrentar a los futuros desafíos.

A mis amigos y compañeros, por acompañarme en esta travesía.

Resumen

En dicho trabajo de fin de grado se tiene como objetivo obtener un modelo de red neuronal en segmentación semántica sobre las imágenes aéreas de vegetación. Donde el modelo será capaz de segmentar las clases relativas de la vegetación y otras clases como edificio, carretera, etc.

Para llegar a dicho objetivo, se divide el proyecto en varios pasos a seguir mediante el método SCRUM. Realizando reuniones con los tutores y miembros del grupo cada una o dos semanas, y coordinando así el progreso y las tareas del proyecto en sprints. Se llevan a cabo unos pasos como: búsqueda y análisis de datasets, entrenamiento del modelo de redes neuronales seleccionado, técnicas de optimización, comparativa de resultados y fusión de varios datasets para aumentar el volumen de datos. Para dicho pasos, se usan diferentes técnicas y herramientas.

Finalmente, se usará transfer learning utilizando los pesos entrenados con los datasets fusionados en un dataset totalmente nuevo y se comparará la efectividad del modelo de la red neuronal.

Abstract

The objective of this work is to obtain a neural network model in semantic segmentation on aerial images of vegetation. Where the model will be able to segment the relative classes of vegetation and other classes like building, road, etc.

To reach this objective, the project is divided into several steps to be followed using the SCRUM method. Meetings with tutors and group members are held every one or two weeks, and the progress and tasks of the project are coordinated in sprints. Some steps are carried out such as: search and analysis of datasets, training of the selected neural network model, optimization techniques, comparison of results and fusion of several datasets to increase the volume of data. For these steps, different techniques and tools are used.

Finally, transfer learning will be used using the weights trained with the fused datasets in a totally new dataset and the effectiveness of the neural network model will be compared.

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

Actualmente, estamos viviendo en una época de gran avance tecnológico, especialmente en el ámbito de la inteligencia artificial. Durante estos años, la inteligencia artificial ha alcanzado objetivos impensables por ser humano.

Por ejemplo, en el año 2016, la victoria de la IA sobre el ser humano en el juego de Go con el triunfo del sistema AlphaGo de Google DeepMind sobre el jugador profesional Lee Sedol. En el año 2023, el lanzamiento de ChatGPT, la IA general que se alimenta con información de internet para su entrenamiento y es capaz de interactuar con los usuarios sobre cualquier tema basándose en la información proporcionada. En solo 5 días, la cantidad de usuarios alcanzó los 10.000.000, y después de 2 meses desde el lanzamiento, la cantidad de usuarios activos mensuales llegó a 100.000.000. Se podrían mencionar numerosos ejemplos sobre el avance de la IA en el día de hoy.

En la ilustración 1.1 se puede observar el número de días tardados en llegar a 10,000,000 de usuarios. Y se puede observar la rapidez con la que ChatGPT alcanzó 1 millón de usuarios.

Casi todas las aplicaciones tardan meses o años en alcanzar 1 millón de usuarios, especialmente Netflix, que tardó casi 3 años. La aplicación más rápida en alcanzar 1 millón de usuarios fue Threads, que tardó aproximadamente 2 horas. En dicho caso, Instagram jugó un papel fundamental para su avance. Con la integración y las promociones con Instagram, sin esta ayuda, Threads no podría haber llegado a esa cifra en tan poco tiempo.

Por lo tanto, con estos casos, se puede reflejar la potencialidad de ChatGPT y su impacto al mundo real.

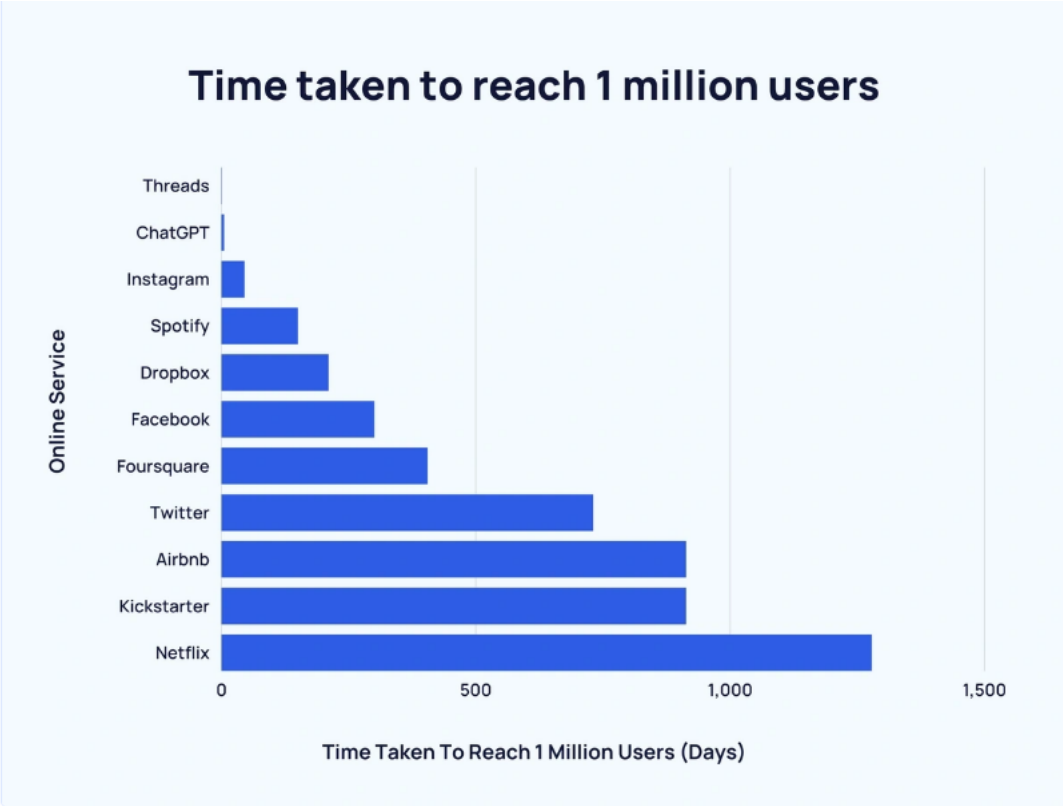

Ilustración 1.1: Tiempo tardado en llegar 1 millón de usuarios[6]

Dentro de dicho proyecto, se va a implementar un modelo de la red neuronal para realizar la segmentación semántica sobre las imágenes aéreas de vegetación.

La vegetación siempre desempeña un papel crucial en el entorno ambiental, siendo responsable de la captación de CO2 y la liberación de oxígeno mediante la fotosíntesis en nuestro planeta. Por otro lado, también juega un papel importante en la cadena alimentaria, la economía, la diversidad biológica y el cambio climático.

Para poder llevar a cabo un control adecuado de la vegetación, es imprescindible realizar un seguimiento constante. Las medidas más modernas de seguimiento son las siguientes:

1. Imagen satelital: presenta cierta dificultad en el análisis y la distinción de las especies de vegetación.
2. Imagen de dron: es la medida más popular para obtener imágenes aéreas de alta calidad y con un bajo coste.
3. Globo aerostático: una medida lenta y costosa.

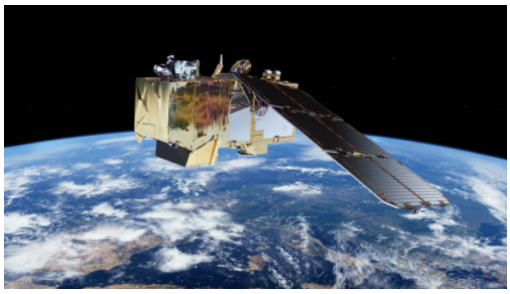

Ilustración 1.2: Satélite[10]

Ilustración 1.3: Drón[8]

Ilustración 1.4: Weather balloon[1]

El control de la vegetación es fundamental en diversas áreas desde una perspectiva medioambiental. Tener la información más reciente sobre la vegetación puede ayudar a tomar decisiones importantes, como tomar medidas para la prevención de incendios forestales, analizar de deforestación o control de plagas de cultivos agrícolas.

Las Islas Canarias están enfrentando un cambio climático grave, donde la subida de temperatura y un descenso de precipitaciones son los dos factores principales que provocan incendios forestales en estos años. Con un control de la vegetación se puede ayudar a expertos en esta área a hacer prevenciones efectivas y realizar políticas medioambientales.Ya que el daño que puede causar un incendio forestal es incalculable.

Por otro lado, la agricultura en islas Canarias también es un sector interesante, ya que necesitan un control sobre sus plantas cultivadas. El principal cultivo de exportación es el tomate, y en segundo lugar está el plátano. Con un control suficiente se puede lograr que las exportaciones de los cultivos sean lo más estables y sostenibles posible.

Por último, el análisis de la vegetación para la estimación del crecimiento y realizar podas preventivas, evitando que la vegetación afecte al tendido eléctrico, lo que puede ocasionar cortes eléctricos e incendios.

1.1. Motivación

La motivación de dicho TFG surge de las asignaturas como Fundamentos de Inteligencia Artificial y Visión por Computador, donde se me mostró el potencial ilimitado que esta tecnología tiene para aplicarse en diversos sectores y mejorar la vida de las personas de manera directa o indirecta. Como detección de los diferentes objetos, clasificación, Chatbot, etc. También me hizo concebir cómo puedo contribuir a este emocionante campo.

Dentro del listado del TFG de la escuela, he conseguido contactar con el tutor de dicho proyecto, D. Adrián Peñate Sánchez, quien me explicó los detalles del proyecto que se va a desarrollar, su contexto en el día de hoy y las tecnologías que se van a utilizar dentro del proyecto.

Tras la explicación detallada, me entusiasmé más que antes. Estos nuevos conocimientos y tecnologías me motivan mucho más, ya que me permiten ampliar mis conocimientos sobre el sector de inteligencia artificial y enfrentar los futuros retos que puedo encontrar en mi vida profesional.

Capítulo 2

Estado del arte

2.1. Metodología

En dicha sección se presentará Scrum como uno de los métodos más populares sobre el desarrollo ágil, y una introducción sobre la técnica que se utilizará durante el desarrollo del TFG.

2.1.1. SCRUM

Durante el desarrollo del TFG, se sigue una metodología ágil de manera iterativa e incremental con referencia al método SCRUM. La idea principal de SCRUM consiste en hacer entregas parciales y regulares hasta que se termine el proyecto.

En la ilustración 2.1 se puede observar el flujo del trabajo sobre SCRUM:

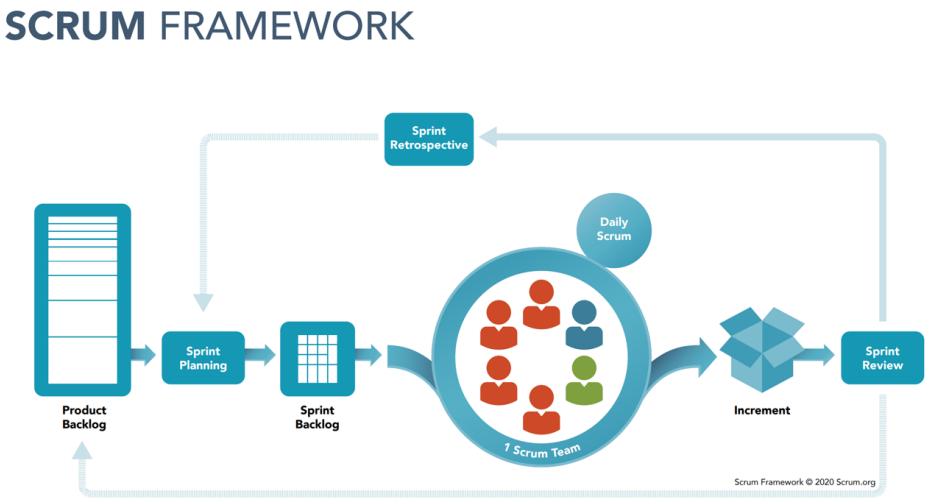

Ilustración 2.1: SCRUM[15]

Durante el desarrollo, se establece una semana o dos semanas como un sprint del SCRUM, y en cada semana se realiza una reunión con los tutores y los compañeros tutorizados para compartir los avances alcanzados durante dicho sprint en comparación con el sprint anterior, y establecer los objetivos que se van a realizar dentro de la semana presente.

Con las presentaciones de los avances de cada sprint, los tutores pueden tener claro el ritmo de los proyectos y ofrecer sugerencias sobre el desarrollo, proporcionar un punto de vista desde otra perspectiva o la solución al posible problema durante el desarrollo.

2.1.2. Segmentación Semántica

En dicho TFG, se realiza la segmentación de la vegetación mediante la técnica llamada ”segmentación semántica”, la cual consiste en la clasificación de las clases a nivel de píxeles, como se puede observar en la siguiente ilustración 2.2 de ejemplo:

Ilustración 2.2: Imagen ejemplar de segmentación semántica multiclase.[21]

En izquierda se puede observar una imagen en RGB como entrada a la red neuronal entrenada, y a su derecha se encuentra una predicción realizada por la red neuronal sobre dicha imagen RGB. Esta predicción clasifica las clases entrenadas y las expone en diferentes colores para facilitar la distinción entre cada clase.

Lo que hace la red neuronal es utilizar las características aprendidas en las capas convolucionales durante la fase de entrenamiento y la información espacial(coordenada de cada píxel, relación entre los píxeles adyacentes, etc.) para hacer la predicción sobre la imagen.

En el ámbito de la segmentación semántica también se puede distinguir basándose en el número de clases que se va a realizar. Cuando se realiza una segmentación de 2 clases, se llama ”segmentación semántica monoclasificada”, como se puede observar en la ilustración 2.3. En caso de realizar múltiples clases, se llama ”segmentación semántica multiclase”, como se ha visto en la ilustración 2.2.

Ilustración 2.3: Imagen ejemplar de segmentación semántica monoclasificada.[21]

2.2. Red Neuronal

En las siguientes secciones se introducirán los conceptos de las redes neuronales para el aprendizaje automático (Machine Learning) y su teoría subyacente.

2.2.1. Aprendizaje Automático

El Aprendizaje profundo (Deep Learning) es "un subconjunto especializado del Machine Learning que, a su vez, es un subconjunto de la Inteligencia Artificial."[34] que consiste en entrenar un equipo informático para que tenga la capacidad de tomar ciertas decisiones sobre las tareas específicas asignadas. Esta capacidad de tomar decisiones se aprende por la red neuronal y sin ser programada por un ser humano para resolver los tres problemas mayores dentro del área de visión por computador que se ilustran en la siguiente ilustración 2.4:

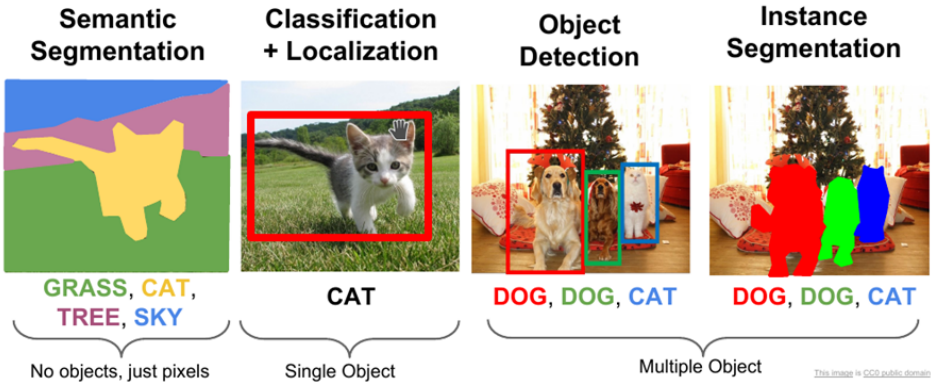

Ilustración 2.4: Tareas de Visión por computador[18]

En dicho sentido, el aprendizaje automático se puede dividir en 3 tipos diferentes basándose en su forma de entrenamiento:

1. **Aprendizaje Supervisado**: En este tipo de aprendizaje, todos los datos destinados al entrenamiento, validación y testeo están etiquetados por un experto en esta área. Posteriormente, se introducen los datos destinados al entrenamiento con las etiquetas correspondientes para entrenar una red neuronal, de manera que aprenda las características de los datos. Luego se realiza la validación y testeo con la red neuronal una vez entrenada, para obtener una red neuronal con una alta capacidad de generalización sobre otros datos que no ha visto durante el entrenamiento.
2. **Aprendizaje no Supervisado**: En este tipo de aprendizaje, la red neuronal no tiene los datos etiquetados, sino que debe ser capaz de agruparlos por características similares. Para este tipo de aprendizaje, se suele utilizar el algoritmo de Clustering. Una vez que el modelo ha sido entrenado con los datos no etiquetados, se valida y testea con datos etiquetados para comprobar la eficiencia del modelo.

3. **Aprendizaje Semi-supervisado**: En este tipo de aprendizaje, se hace uso de la combinación de datos etiquetados y no etiquetados. Normalmente, se utiliza una pequeña cantidad de datos etiquetados junto con una gran cantidad de datos no etiquetados para el entrenamiento. Esta forma intermedia combina las ventajas del aprendizaje supervisado(más preciso) y no supervisado(ahorro del coste del tiempo, recursos, etc.).

Para tener un mejor entendimiento sobre el funcionamiento de una red neuronal, hay que entrar en su unidad básica que compone dicha red: la neurona. Es útil dirigirse a la definición biológica de una neurona.

2.2.2. Neurona Biológica

Las redes neuronales artificiales están inspiradas en las funcionalidades de una neurona biológica(recibir, procesar y transmitir la información) y en la interconexión de millones de neuronas para formar una red neuronal. Una neurona no es más que una célula que se compone de tres partes principales:

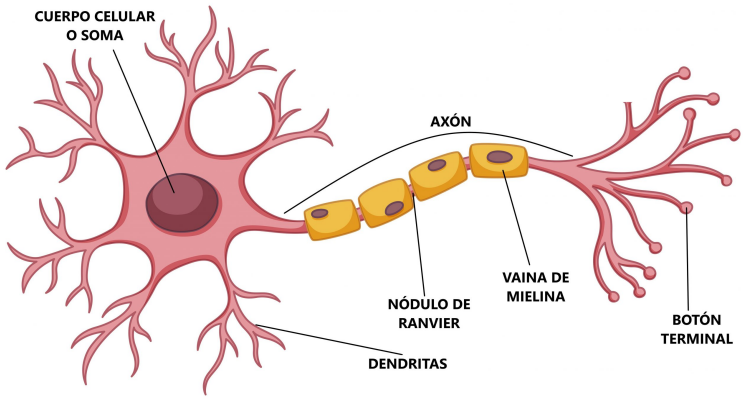

Ilustración 2.5: Neurona Biológica[25]

1. Soma: Parte central y principal de una neurona, donde se encuentran el núcleo, citoplasma, etc. Y es donde se procesan los impulsos nerviosos capturados por las dendritas.
2. Dendritas: Partes encargadas de capturar el impulso nervioso emitido por el axón de otra neurona y transmitirlo hacia la dirección del soma para procesar dicho impulso.
3. Axón: Parte encargada de transmitir los impulsos procesados por el soma hacia otra neurona.

2.2.3. Neurona Artificial

Con el concepto del funcionamiento de una neurona biológica, el concepto de la neurona artificial se comporta de manera similar y también tiene una estructura similar a una neurona biológica como se observa en la ilustración 2.6.

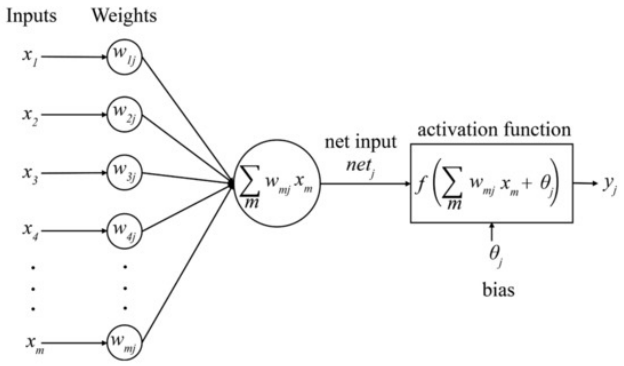

Ilustración 2.6: Neurona Artificial[14]

Dentro de una neurona artificial, los impulsos nerviosos corresponden a la información que la neurona artificial necesita procesar. Esta información puede ser el píxel de una imagen, segmentos de texto, etc.

La entrada de dicha información corresponde a la salida de otra neurona que ha realizado algún procesamiento sobre esa información. Cada entrada tiene un peso correspondiente que indica el nivel de contribución de esa entrada a la actividad de la neurona a la que se dirige. Durante la fase de entrenamiento, estos pesos se modifican constantemente.

Dentro del cuerpo de la neurona artificial, se realiza un sumatorio de la multiplicación de todas las entradas con sus pesos correspondientes, añadiendo el parámetro bias para controlar el nivel de facilidad para activar o desactivar dicha neurona artificial y ajustarla mejor a las entradas. El valor procesado dentro del cuerpo de una neurona artificial se pasa a la función de activación, que se encarga de controlar la activación de la neurona.

La función de activación se puede dividir en 2 tipos:

1. **Función Lineal**: es la función que se basa en regresión lineal. Su representación en el plano cartesiano corresponde a una línea recta. Su expresión matemática se puede observar en la fórmula 2.1:

$$f(x) = mx + b \tag{2.1}$$

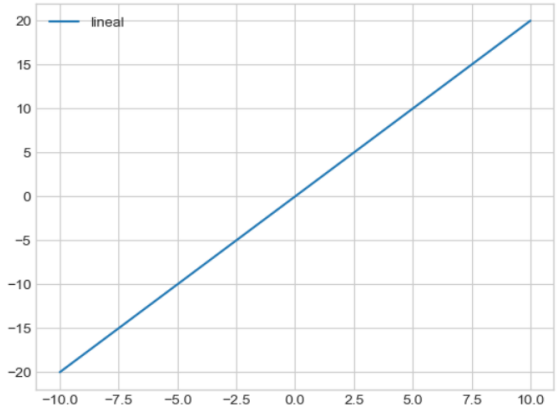

Ilustración 2.7: Función Lineal

2. **Función no Lineal**: es similar a la función lineal, pero su representación en el plano cartesiano es diferente a la función lineal, como se ilustra en los siguientes ejemplos de funciones no lineales.

Función Sigmoid: Es la función matemática donde el rango de los valores de salida está dentro de (0, 1). Por lo tanto, si la entrada de dicha función es un valor negativo o positivo muy grande, la salida de la función siempre estará dentro del rango (0, 1). Los valores dentro de $+\infty$ y $-\infty$ también. Su expresión matemática se puede observar en la fórmula 2.2:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.2}$$

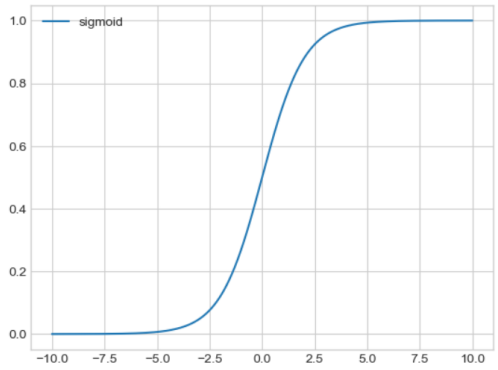

Ilustración 2.8: Función Sigmoid

Función Relu: Es la función matemática si la entrada es un valor negativo, la salida de la función es 0, y en caso de que el valor sea positivo, la salida será el mismo valor sin modificación. Su expresión matemática se puede observar en la fórmula 2.3.

$$f(x) = \text{máx}(0, x) \tag{2.3}$$

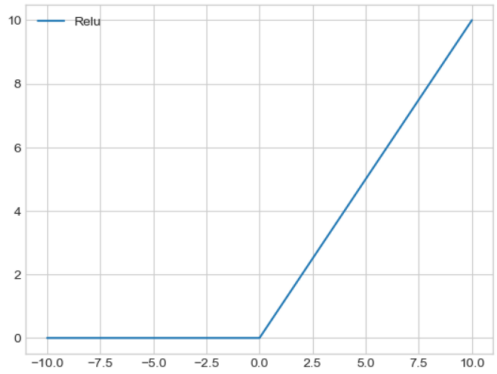

Ilustración 2.9: Función Relu

Función Tanh: Es la función matemática donde se mapea el valor dentro del rango $+\infty$ $-\infty$ al (-1, 1). Su expresión matemática se puede observar en la fórmula 2.4.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.4}$$

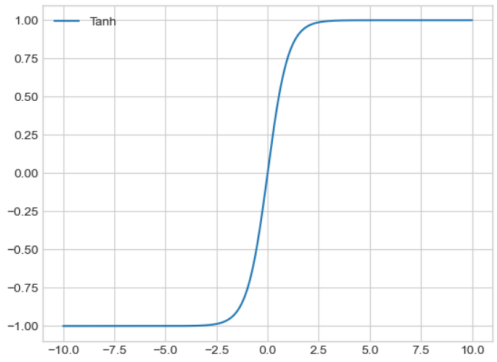

Ilustración 2.10: Función Tanh

Función Softmax: Es la función matemática donde se pueden tomar los valores de entrada dentro del rango desde $-\infty$ hasta $+\infty$ al (0, 1). La entrada se convierte en una salida con valores dentro del rango 0 y 1; se puede entender como la ”probabilidad”. Los valores menores de entrada obtendrán una salida que tiende a 0, mientras que los valores mayores de entrada obtendrán una salida que tiende a 1. Su expresión matemática se puede observar en la fórmula 2.5.

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{2.5}$$

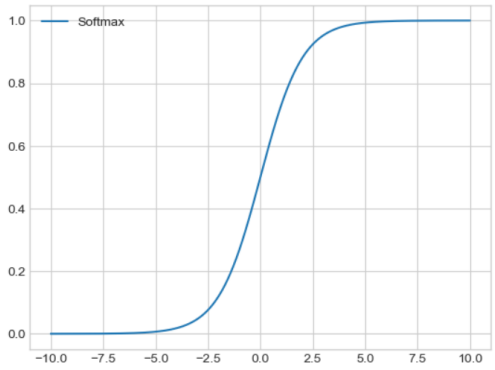

Ilustración 2.11: Función Softmax

2.2.4. Red Neuronal

La interconexión de múltiples neuronas artificiales forma una red neuronal. Una red neuronal se puede dividir en 3 partes, como se puede observar en la ilustración 2.12.

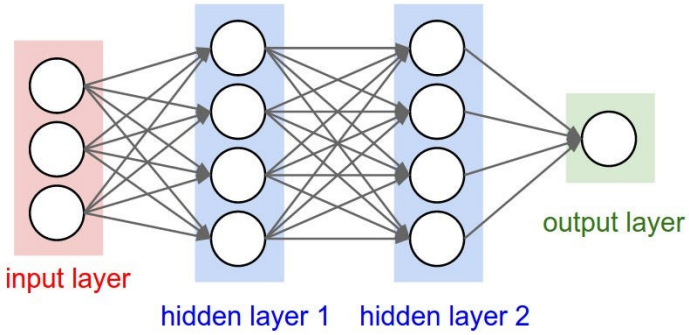

Ilustración 2.12: Estructura de red[9]

En esta imagen, se pueden observar las 3 partes:

- **input layer (capa de entrada)**: es la capa donde se entran los datos, como imágenes aéreas en el caso de este TFG, para que la red neuronal pueda capturar las características y realizar la predicción en base a dichas características capturadas durante la fase de entrenamiento.
- **output layer(capa de salida)**: es la capa donde la red neuronal realiza la predicción sobre una tarea específica. Es aquí donde la red neuronal aplica las características aprendidas de las diferentes clases durante la fase de entrenamiento para hacer su propia predicción. Cuanto más aprenda la red neuronal, más acertada será la predicción.
- **hidden layer (capa oculta)**: son las capas intermedias de la red, podemos encontrar varios tipos, tales como:
  - **Capa convolucional**: es la capa donde se aplica la operación llamada "convolución" mediante un kernel con un tamaño preestablecido para obtener

un nuevo mapa de salida (mapa de características). La convolución consiste en realizar la operación de producto escalar sobre un píxel determinado y sus píxeles adyacentes con el kernel. El kernel se desplaza desde la izquierda hacia la derecha, de arriba hacia abajo, hasta terminar con todos los píxeles de entrada.

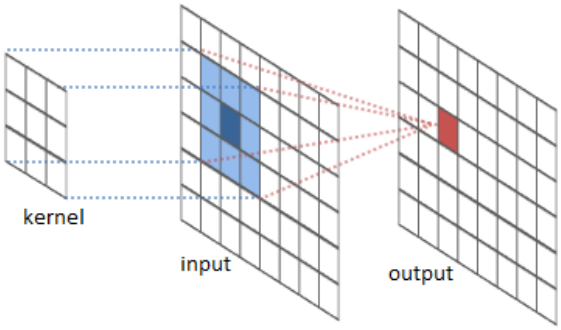

Ilustración 2.13: Proceso de convolución[28]

Mediante la ilustración 2.14 se puede observar que las características extraídas por las capas convolucionales delanteras son características de bajo nivel, que son poseídas por casi todos los objetos, como características de borde, textura, etc.

En las capas convolucionales centrales se presentan las características extraídas del nivel medio. Estas son una combinación de las características del nivel bajo, más cercanas al objeto que la red neuronal intenta predecir.

Por último, en las capas convolucionales finales de la red neuronal se extraen las características más abstractas y profundas, y la red neuronal realiza la predicción basándose en dichas características.

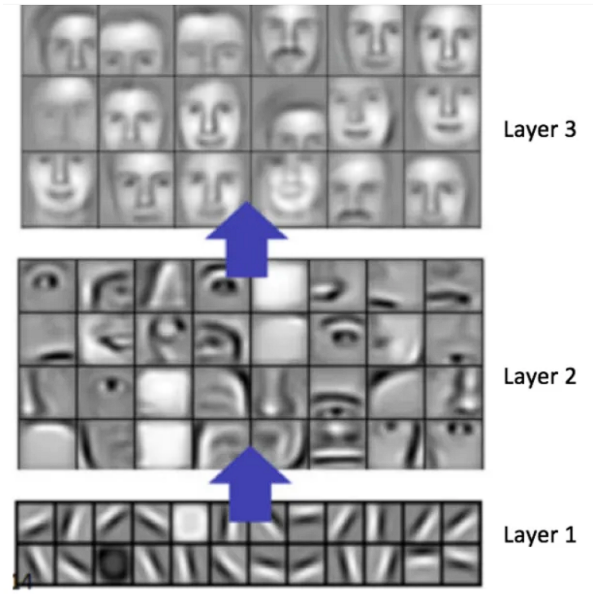

Ilustración 2.14: Características extraídas[19]

- **Capa de Pooling**: Es la capa donde se reduce la magnitud de la salida de una capa convolucional, y también se reduce la dimensionalidad y la cantidad de cálculos requeridos para procesar, manteniendo las características más importantes para un entrenamiento eficiente. En la siguiente ilustración se realiza un max-pooling de 2 x 2, donde solamente se guarda el valor más grande de la matriz de 2 x 2.

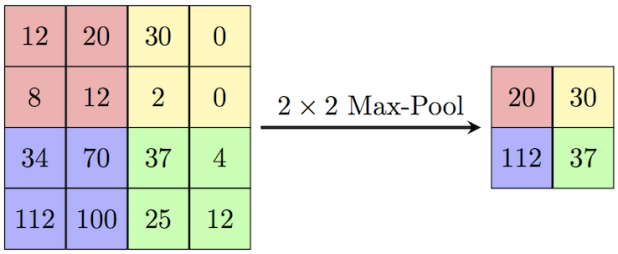

Ilustración 2.15: Proceso del Polling[7]

2.3. División del dataset

Para organizar los datos de los datasets a entrenar, se sigue la siguiente política de división: 70 % de los datos están dedicado al entrenamiento, 20 % a la validación y 10 % al testeo, como se muestra en la ilustración 2.16.

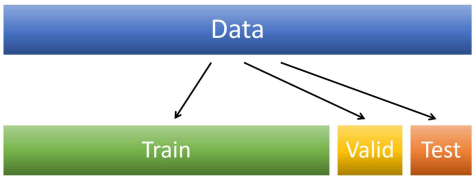

Ilustración 2.16: Train, Validation, Testeo[22]

El uso de cada porción es diferente:

1. Train: son los datos que se entregan a la red neuronal para que "aprenda" las características de las clases especificadas, mediante la actualización de los pesos, bias y otros millones de parámetros dentro de la red neuronal.
2. Validation: son los datos utilizados para evaluar el modelo de la red neuronal en cada época, con el fin de comprobar su capacidad de generalización sobre datos que no ha visto durante el entrenamiento. También ayuda a seleccionar el mejor modelo entrenado. La red no actualiza sus parámetros con estos datos.
3. Test: son los datos que no se ven durante el entrenamiento ni la validación, con el propósito de probar la capacidad del mejor modelo seleccionado por la validación. Es

costumbre en datasets públicos no tener las anotaciones del conjunto de testeo para descargar, teniendo así que subir las predicciones de la red entrenada online, y evaluar los diferentes modelos en igualdad de condiciones.

2.4. Hiperparametros

A la hora de hacer entrenamiento de la red neuronal sobre los diferentes datasets, se preestablecen una serie de hiperparámetros para controlar el proceso de aprendizaje sobre la red neuronal. Los hiperparámetros, como se indica por su nombre, son los parámetros de nivel superior que determinan los valores del modelo mediante el uso del algoritmo de aprendizaje automático, se enumeran las siguientes hiperparámetros usados dentro del TFG:

1. Función de perdida(Loss Function): Es una función que proporciona información sobre la diferencia entre la salida de la red neuronal y el objetivo. Cuando el resultado de la función de pérdida es más grande, significa que la salida de la red neuronal está más desviada del objetivo; en caso contrario, significa que la salida está aproximada al objetivo. En el tema de segmentación semántica, se usa la función de pérdida llamada "CrossEntropyLoss", que es la función apropiada para la tarea de segmentación semántica multiclase.

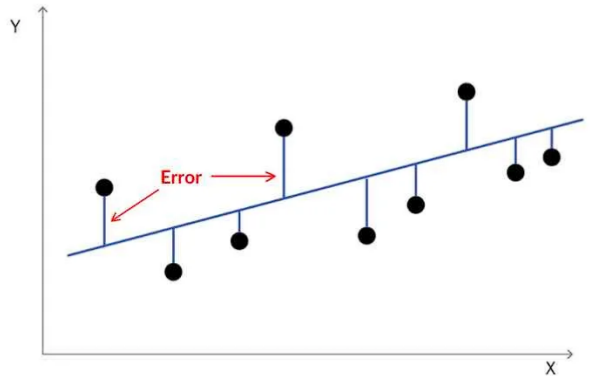

Ilustración 2.17: Función de pérdida[24]

2. **Learning rate** (Tasa de aprendizaje): Es la magnitud del paso que el gradiente da hacia loss minimo proporcionada por la función de pérdida en cada iteración. Normalmente se establece con un valor de 1e-3. Hay dos resultados en función del valor de dicho hiperparámetro. En caso de tener una tasa de aprendizaje alta, la red se entrena rápidamente, pero es difícil de converger; en caso de tener un valor muy pequeño, la velocidad de convergencia de la red es muy lenta. Como se muestra en la ilustración 2.18.

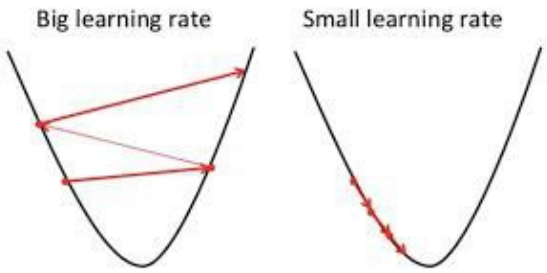

Ilustración 2.18: Corporación de Learning Rate[3]

3. **Optimizador**: Se obtienen los gradientes de los parámetros a ajustar durante el "Back propagation" en el entrenamiento. El optimizador utiliza dichos gradientes para realizar la optimización de los parámetros dentro de la red con el objetivo de obtener un loss mínimo posible.

4. **Batch size**: Es el número de imágenes que se entregan para entrenar en cada iteración de una época. El batch size está restringido por la capacidad de la memoria de GPU, el tamaño del modelo, etc. En el caso de dicho TFG, se utiliza un batch size de 8 para el entrenamiento y 16 para la validación y testeo. El valor del batch size está relacionado la capacidad de generalizar de la red neuronal, por lo que se suele ajustar el learning rate acorde al batch size..

5. **Learning Rate Scheduler**: La función principal de un Learning Rate Scheduler implica una modificación en vivo durante la fase de entrenamiento.

   En dicho TFG, se establece el Learning Rate inicial de entrenamiento con el valor 1e-3, y se utiliza un scheduler llamado "ExponentialLR", cuya característica principal consiste en hacer multiplicación del Learning Rate con el parámetro "gamma" en cada época del entrenamiento. Dado que un Learning Rate mayor ayuda a converger rápidamente al inicio del entrenamiento, acelerando así el proceso de búsqueda del mínimo global y realizando cambios grandes. A medida que avanza el entrenamiento de la red neuronal, el Learning Rate disminuye para lograr una convergencia más estable y suave hacia el mínimo, permitiendo que los cambios en los parámetros sean más precisos y finos con cada iteración. A continuación, se muestra los diferentes schedulers en la ilustración 2.19.

Ilustración 2.19: Diferentes ajustadores de Learning Rate[17]

2.5. Métricas

A la hora de determinar la capacidad de la red neuronal, se establecen una serie de métricas estándares sobre segmentación semántica para proporcionar dicha información. Las métricas no son más que operaciones matemáticas que consisten en hacer los cálculos sobre la predicción que hace la red neuronal y la máscara con los valores reales que representa el Ground Truth.

Para conseguir las métricas, el primer paso siempre consiste en obtener la matriz de confusión. Es una herramienta visual y potente que puede proporcionar la información sobre el comportamiento de la red neuronal en cada clase y permite obtener métricas más precisas.

Ilustración 2.20: matriz de confusión[5]

La tabla de confusión están constituido por 2 partes:

1. En cada fila de la tabla se representa la predicción del modelo sobre una clase especificada. La suma de todos los valores de una misma fila representa el total de píxeles predichos para esa clase.
2. En cada columna de la tabla se representa el Ground Truth sobre una clase especificada. La suma de todos los valores de una misma columna representa el total de píxeles reales que pertenecen a esa clase.

El siguiente ejemplo de la matriz de confusión 2.1 puede ayudar a entender mejor todos los conceptos básicos:

Real (píxeles)	Predicción (píxeles)			
	Acebuche	Drago	Palmera	Sabina
Acebuche	125	20	35	8
Drago	10	259	12	11
Palmera	0	0	58	3
Sabina	100	0	0	300

Cuadro 2.1: Ejemplo de la Matriz de Confusión

En dicha matriz de confusión se incluyen un total de 4 clases correspondientes a diferentes especies de árboles.

A partir de dicha tabla de confusión de ejemplo se puede extraer la información sobre la clase **Drago**:

- 10 píxeles se han predicho como Acebuche, pero realmente pertenecen a la clase Drago.
- 259 píxeles se han predicho como Drago, y realmente sí pertenecen a la clase Drago.
- 12 píxeles se han predicho como Palmera, pero realmente pertenecen a la clase Drago.
- 11 píxeles se han predicho como Sabina, pero realmente pertenecen a la clase Drago.

En la situación ideal, todos los píxeles predichos coinciden con los píxeles reales, pero esta situación rara vez se alcanza, ya que los seres humanos también fallan muchas veces en problemas de distinción. A partir de esta matriz de confusión, se pueden extraer algunas informaciones básicas como:

- **True Positive**: Indica el número de píxeles que se han predicho correctamente para una clase. Por ejemplo, para la clase Drago, cuántos píxeles ha predicho el modelo y realmente eran de la clase Drago?
- **True Negative**: Indica el número de píxeles que no se han predicho como una clase y realmente no eran esa clase. Por ejemplo, para la clase Drago, cuántos píxeles no ha predicho el modelo como Drago y realmente no eran de la clase Drago?

- **False Positive**: Indica el número de píxeles que se han predicho como una clase pero realmente no eran esa clase. Por ejemplo, para la clase Drago, cuántos píxeles ha predicho el modelo que son Drago y realmente no eran de la clase Drago?
- **False Negative**: Indica el número de píxeles que no se han predicho como una clase pero realmente eran esa clase. Por ejemplo, para la clase Drago, cuántos píxeles no ha predicho el modelo que son Drago pero realmente eran de la clase Drago?

Con la información proporcionada por la matriz de confusión, se pueden obtener las métricas para valorar la eficiencia del modelo entrenado. A continuación se presentan las métricas utilizadas durante el desarrollo del proyecto[16]:

- Precision: Esta métrica valora el porcentaje de muestras predichas correctamente (donde coinciden con los valores reales correspondientes) de todas las muestras predichas por el modelo como ejemplos positivos. Tiene su expresión matemática como la siguiente fórmula 2.6:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.6}$$

- Recall: Indica el porcentaje de todas las muestras con etiquetas verdaderas positivas se predicen correctamente. Tiene su expresión matemática como la siguiente fórmula 2.7:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.7}$$

- Dice: Es una combinación de métricas sobre Recall y Precision, lo que puede ofrecer una perspectiva más global sobre el modelo. Tiene su expresión matemática como la siguiente fórmula 2.8:

$$\text{Dice} = \frac{2 * TP}{2 * TP + FN + FP} \tag{2.8}$$

- IOU: Indica el porcentaje de la intersección entre la inferencia hecha por el modelo y su Ground Truth correspondiente. Tiene su expresión matemática como la siguiente fórmula 2.9:

$$\text{IOU} = \frac{TP}{TP + FN + FP} \tag{2.9}$$

Capítulo 3

Objetivos

El objetivo principal de este TFG es obtener un modelo de redes neuronales para tarea de segmentación semántica de imágenes aéreas que contienen clases de vegetación. Para lograr dicho objetivo, se plantean las siguientes pasos a seguir durante el desarrollo del trabajo:

1. **Estudio Previo**: Durante esta etapa del desarrollo, se llevará a cabo un estudio sobre el framework principal del trabajo, PyTorch. Se seguirán los tutoriales disponibles en la página web oficial para obtener una base sólida que permita enfrentar los siguientes desafíos que puedan surgir.

   Además, se realizará un estudio sobre arquitecturas de redes neuronales usadas para la segmentación semántica, especialmente sobre su teoría subyacente.

2. **Búsqueda de datasets**: Durante esta etapa de desarrollo, se realizará una búsqueda exhaustiva de datasets públicos provenientes de diferentes fuentes. Se llevará a cabo un análisis de viabilidad centrándose en datasets que contengan imágenes aéreas y que incluyan clases relevantes de vegetación.

3. **Preparación previa del entrenamiento**: Para poder llevar a cabo el TFG, la preparación antes de hacer los diferentes experimentos es fundamental. Donde incluyen la creación del entorno adecuado y procesamiento de los datos en datasets

4. **Entrenamiento de la red sobre diferentes datasets**: Durante esta etapa del desarrollo, una vez seleccionada la arquitectura del modelo y los datasets, se procederá al desarrollo del código para el entrenamiento de dichos datasets. Posteriormente, con el modelo entrenado, se llevará a cabo un proceso de validación y testeo para evaluar la eficiencia del modelo entrenado.

5. **Análisis de los resultado obtenidos**. Una vez completada la etapa anterior, se toma una porción del dataset que no se haya utilizado durante la fase de entrenamiento para testear el modelo entrenado. Se procederá a realizar un análisis exhaustivo basado en diferentes métricas establecidas y buscar las tecnologías para mejorar la precisión de la red neuronal.

6. **Unión de datasets**: En esta etapa se procederá a combinar los datasets de diferentes

maneras con el objetivo de aumentar los datos de entrenamiento y obtener una mayor variedad de clases de vegetación, para ello se unirán las clases comunes de los diferentes datasets. Luego, se entrenará el modelo seleccionado, sobre dichas variaciones de datos.

7. **Realización de Transfer Learning**: Teniendo los modelos entrenadas con diferentes combinaciones en la etapa anterior, en esta etapa se procederá hacer transfer learning sobre un nuevo dataset. La idea principal del transfer learning consiste en aprovechar los "conocimientos" aprendidos por otro dataset y reutilizar dichos "conocimientos" para una tarea relacionada, en dicho TFG, se utiliza un dataset grande con más de 60.000 imágenes para probar la técnica de transfer learning
8. **Valoración y conclusión de los resultados**: Teniendo las redes neuronales entrenados en la etapa anterior, se procederá el mismo procedimiento que la etapa de "análisis de los resultado obtenidos"para obtener un resultado final y concluir el TFG

Capítulo 4

Herramientas usadas

Para el desarrollo de dicho TFG, se han utilizado diferentes herramientas y librerías para alcanzar el objetivo principal.

4.1. Lenguaje de programación

4.1.1. Python

Python es un lenguaje de programación de alto nivel e interpretado. La meta principal de este lenguaje es la simplificación de la sintaxis y una facilidad de aprendizaje comparada con otros lenguajes de programación.

La selección de este lenguaje se debe a que la mayor parte de las librerías que se van a utilizar están basadas en Python, lo que facilita el desarrollo del proyecto.

Ilustración 4.1: Python[13]

4.2. Herramientas

4.2.1. Anaconda

Anaconda es una herramienta con la funcionalidad principal de gestión del entorno virtual y las dependencias necesarias para el desarrollo y prueba.

Ilustración 4.2: Anaconda[31]

4.2.2. Visual Studio Code

Visual Studio Code es un editor de código fuente desarrollado por Microsoft para diferentes entornos como Windows, Linux, etc. con el soporte para depuración, autocompletado de código y las diferentes herramientas que pueden facilitar la programación

Ilustración 4.3: Visual Studio Code[33]

4.3. Librerías

4.3.1. Pytorch

PyTorch es una librería de código abierto que tiene una influencia mundial en el aprendizaje de la inteligencia artificial. Está basada en Torch, otra librería de código abierto programada en el lenguaje de programación C. Esta librería aborda muchos módulos útiles, como la carga del modelo preentrenado, la creación de DataLoader para cargar los datos y sus máscaras

correspondientes, el seguimiento de las métricas y la visualización a través de SummaryWriter, etc.

Ilustración 4.4: Pytorch[26]

4.3.2. Opencv

Es una librería de código abierto como PyTorch, dirigida a la visión por computador. Dicha librería ofrece módulos como la carga de imágenes desde el disco y las diferentes manipulaciones sobre las imágenes.

Ilustración 4.5: OpenCV[23]

4.3.3. Albumentation

Es una librería de código abierto dirigida a la operación de data augmentation. Consiste en realizar operaciones sobre los datos del dataset con el objetivo de aumentar la cantidad cuando se enfrenta el problema de desbalanceo de clases en la fase de entrenamiento. Ofrece diferentes operaciones como rotación horizontal/vertical, cambio aleatorio de canal, recorte aleatorio, etc.

Ilustración 4.6: Albumentation [2]

Capítulo 5

Competencias específicas cubiertas

Según el documento "Memoria de Plan de Estudio del Grado de Ingeniería Informática", en dicho TFG incluyen las siguientes competencias:

- ✔ **CI15**: «Conocimiento y aplicación de los principios fundamentales y técnicas básicas de los sistemas inteligentes y su aplicación práctica.»
- ✔ **TFG**: «Ejercicio original a realizar individualmente y presentar y defender ante un tribunal universitario, consistente en un proyecto en el ámbito de las tecnologías específicas de la Ingeniería en Informática de naturaleza profesional en el que se sinteticen e integren las competencias adquiridas en las enseñanzas.»

Capítulo 6

Desarrollo

En esta sección se presentan los pasos seguidos durante el desarrollo del proyecto. Se va a presentar el modelo específico utilizado, presentación de diferentes datasets y sus entrenamientos con un análisis exhaustivo sobre los resultados, etc.

6.1. Tarea 1 Estudio sobre arquitectura del modelo usado

En la sección anterior 2.2 se hablaba sobre la teoría de la neurona artificial y la red neuronal, donde se obtiene una base sólida de los conceptos básicos. En esta tarea 1 del estudio previo antes de hacer cualquier entrenamiento u código, se enfoca en el estudio de la red neuronal especificada para la segmentación semántica, FCN (Fully Convolutional Networks). Una FCN tiene una diferencia crucial en la última capa en comparación con las CNN (Convolutional Neural Networks), donde se sustituyen las últimas capas Full Connected de una red CNN por capas de convolución transpuesta para restaurar la dimensión de los mapas de características a la misma dimensión que los datos de entrada.

Con la diferencia comentada, se analizará la arquitectura de FCN seleccionado con buena profundidad. Además, se realizará un estudio sobre el uso de la biblioteca PyTorch para obtener una base sólida, como el manejo de tensores, uso del modelo, dataloader, etc.

Para poder llevar a cabo dicho TFG con éxito, la selección de la arquitectura de la red neuronal va a ser importante. En dicho trabajo se hace uso de la arquitectura de la serie Deeplab, concretamente Deeplabv3. Es una arquitectura reciente y fue publicada por GOOGLE, con características principales derivadas de Deeplabv1 y Deeplabv2:

1. Atrous Convolution: Es la característica más destacada en Deeplabv1. Es un tipo de convolución diferente a la convolución tradicional, donde se realiza la operación escalar sobre los valores adyacentes. En el caso de la Atrous Convolution, los valores que se utilizan para la convolución están separados por el parámetro “dilate rate”. Cuando el dilate rate es igual a 2, el espacio entre dos valores adyacentes del kernel tiene una

distancia de 1 valor(como se muestra en la ilustración 6.1). Con la Atrous Convolution se logra ampliar la "vista" de la red neuronal, lo que permite que las características obtenidas sean más globales.

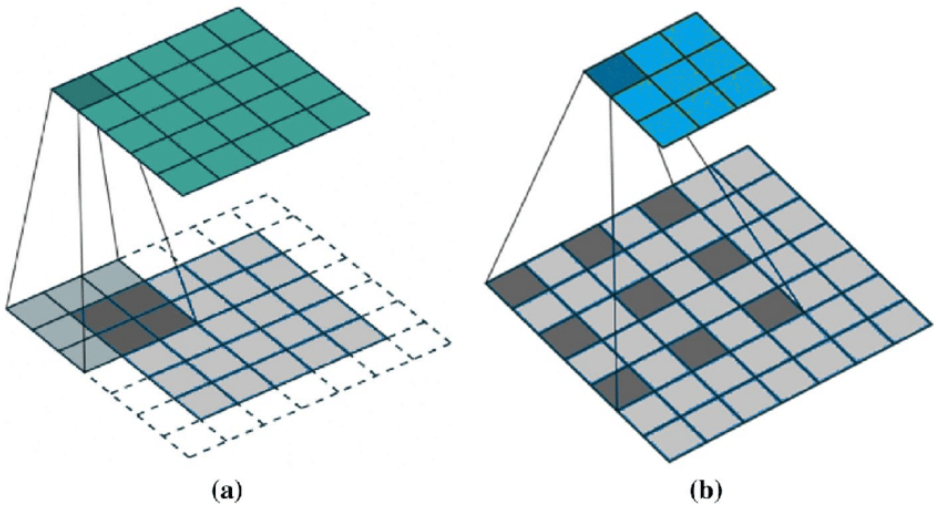

Ilustración 6.1: Atrous Convolution[11]

2. ASPP Layer (Atrous Spatial Pyramid Pooling): Es la característica más destacada en Deeplabv2. Es un módulo inspirado en la idea de SPP (Spatial Pyramid Pooling). En lugar de usar operaciones de pooling de diferentes tamaños como en SPP, en ASPP se utilizan convoluciones atrous con diferentes "dilate rate", y una operación de Global Average Pooling para capturar una característica global promedio en las entradas de los mapas de características. Posteriormente, se concatenan todos los mapas de características obtenidos en el módulo ASPP y se realiza una convolución de 1x1. Con este módulo, la red neuronal puede capturar características desde diferentes perspectivas, desde las más locales hasta las más globales.

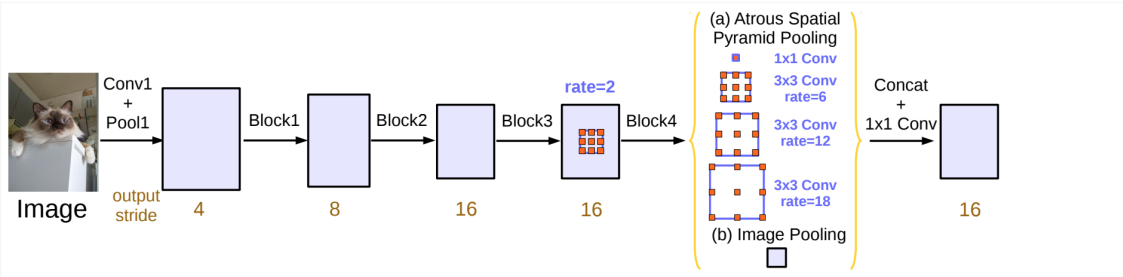

Ilustración 6.2: ASPP Layer[20]

En la siguiente imagen se muestra la arquitectura de Deeplabv3, donde se puede observar que la arquitectura se puede dividir en 2 partes:

- Encoder: es la parte de la red neuronal donde se realiza el "down sampling", extrayendo las características de los datos de entrada. Aquí es donde se aplica el Backbone como

ResNet18, MobileNetV2, etc. para extraer las características, y también se aplica el módulo ASPP para obtener características de diferentes niveles de detalle.

- Decoder: es la parte de la red neuronal donde se realiza el "up sampling" para recuperar la dimensión de la mapa de características pasada por la convolución a la dimensión de los datos de entrada. Las operaciones realizadas en esta parte de la arquitectura consisten en tomar la mapa de características de nivel bajo, que contiene características superficiales del backbone, y concatenarla con la mapa de características obtenidas por el módulo ASPP para realizar convoluciones y up sampling. En este caso, se combinan las mapas de características profundas (que contienen características abstractas) y superficiales (que contienen características básicas, locales, que se pierden dentro de capas sucesivas de convolución) para realizar la segmentación semántica.

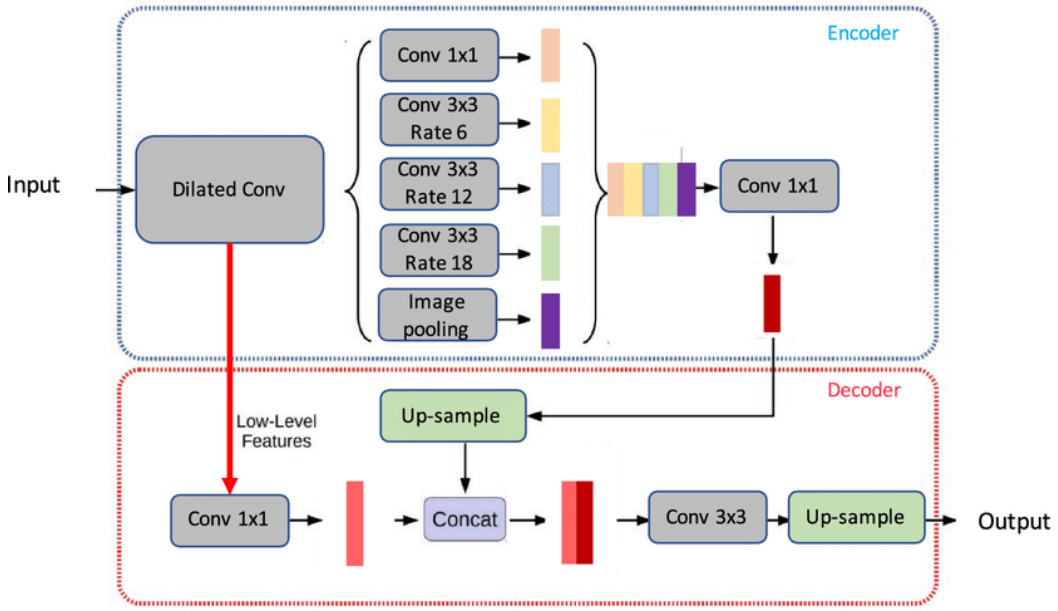

Ilustración 6.3: Arquitectura Deeplabv3[29]

Una vez seleccionado la arquitectura que se va a utilizar durante el desarrollo del TFG, el sigueinte paso consisiten en selecionar el backbone que se a utilizar para capturar las caracteristicas para la red neuronal durante el entrenamietno.

Para Deeplabv3, hay una serie de backbone que se puede elegir, como:

1. Serie Resnet(18, 34, 50, 101)
2. MobileNetV2

En dicho TFG se usa la ResNet para el futuro desarrollo. La arquitectura del backbone de la serie ResNet tiene la característica más distintiva en su bloque residual, como se ilustra en la siguiente imagen. Dicho componente de la red ResNet permite "guardar" los datos de entrada antes de ser procesados por las capas convolucionales, ofreciendo una especie de "garantía" en

caso de que el resultado empeore tras pasar por las capas convolucionales. De esta manera, se conservan los datos de entrada como aseguramiento y permite hacer un entrenamiento profundo evitando el fenómeno del "Problema de desvanecimiento de gradiente".

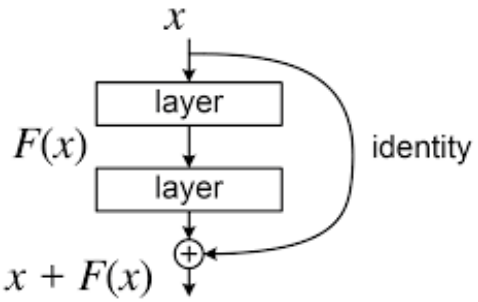

Ilustración 6.4: Bloque Residual[32]

6.2. Tarea 2 Búsqueda y Descripción de los datasets utilizados

Tras el paso anterior sobre el estudio de la arquitectura especificada para el desarrollo del TFG, el siguiente paso consiste en el análisis de los 4 datasets encontrados en Internet con acceso público(GitHub, Kaggle y huggingface), estas paginas webs publicas contienen grande cantidad de datasets con acceso libre y se han contribuido al área de la inteligencia artificial con un nivel de influencia mundial . Se va a analizar la distribución de cada dataset, las clases que componen, cuáles son las clases de interés para el TFG y las razones para elegir dicho dataset.

Para el desarrollo del TFG, es indispensable contar con uno o varios datasets para la etapa de entrenamiento del modelo sobre las clases de interés. Un dataset en el ámbito de la segmentación semántica no es más que un conjunto de datos compuesto por las imágenes RGB(Rojo, Verde y Azul) y sus máscaras correspondientes. En la máscara, cada píxel está etiquetado con el valor de la clase correspondiente.

A continuación, se presenta una lista de los datasets encontrados en Internet y su política de selección.

6.2.1. Dataset FloodNet

En primer lugar, se presenta el dataset FloodNet, dicho dataset consta de un total de 2343 imágenes y sus máscaras correspondientes. Las imágenes tienen una resolución de 3000 x 4000 píxeles, lo que indica una alta resolución, y son capturados por el dron DJI Mavic Pro quadcopter. En FloodNet se incluyen un total de 10 clases, como se puede observar en el cuadro 6.1. La fuente de dicho dataset se puede encontrar en [27]

Clases	COLOR
Background	
Building-Flooded	
Building-non-Flooded	
Road-Flooded	
Road-non-Flooded	
Water	
Tree	
Vehicle	
Pool	
Grass	

Cuadro 6.1: Tabla de clases de FloodNet

La ilustración 6.5 siguiente muestra la visualización de los datos del dataset y sus máscaras correspondientes. En izquierda se encuentra la imagen RGB, y a su derecha se encuentra

su máscara correspondiente con los colores de las clases de dicho dataset y su superposición sobre imagen RGB.

Con estas imágenes se ayuda a entender cómo es el comportamiento de cada clase dentro del dataset y es uno de los métodos más fáciles y visuales para entender el dataset.

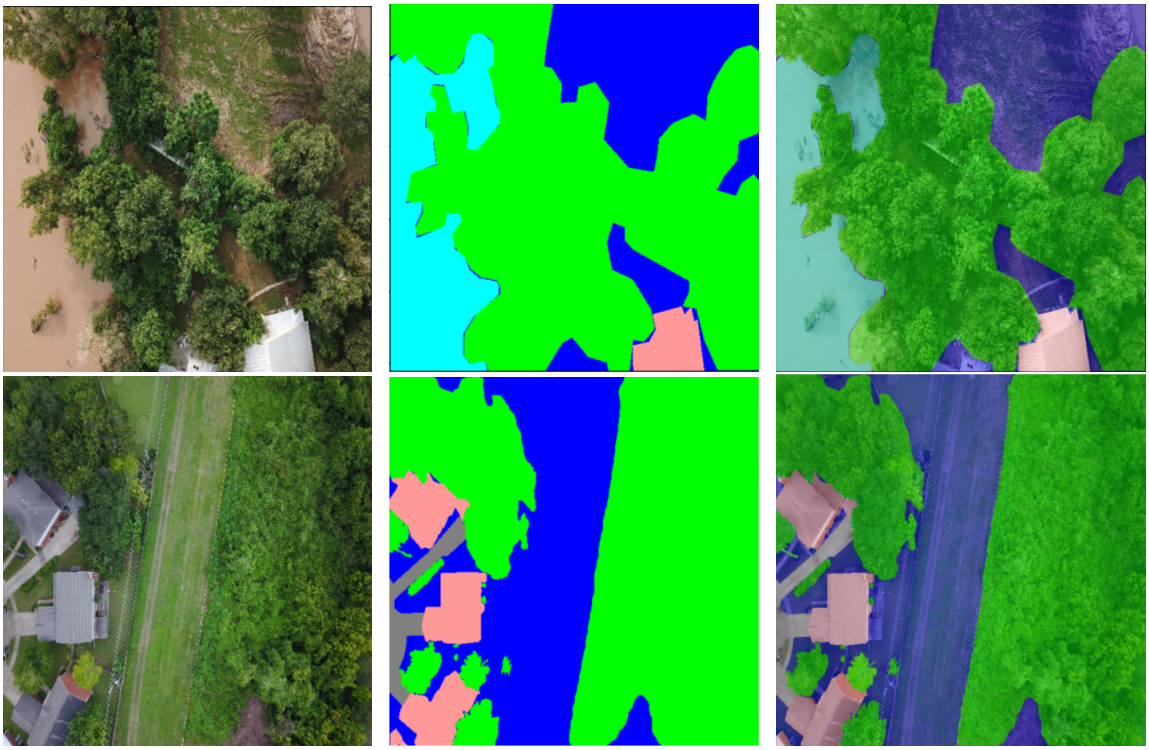

Ilustración 6.5: Datos y mascaras del Floodnet

A continuación, se muestran diferentes gráficas del mencionado dataset, donde se puede observar la distribución de los píxeles de cada clase(Ilustración 6.6) y la cantidad de imágenes en las que aparece cada clase(Ilustración 6.7). Estas gráficas pueden ayudar en el análisis del dataset y en la toma de decisiones adecuadas al respecto.

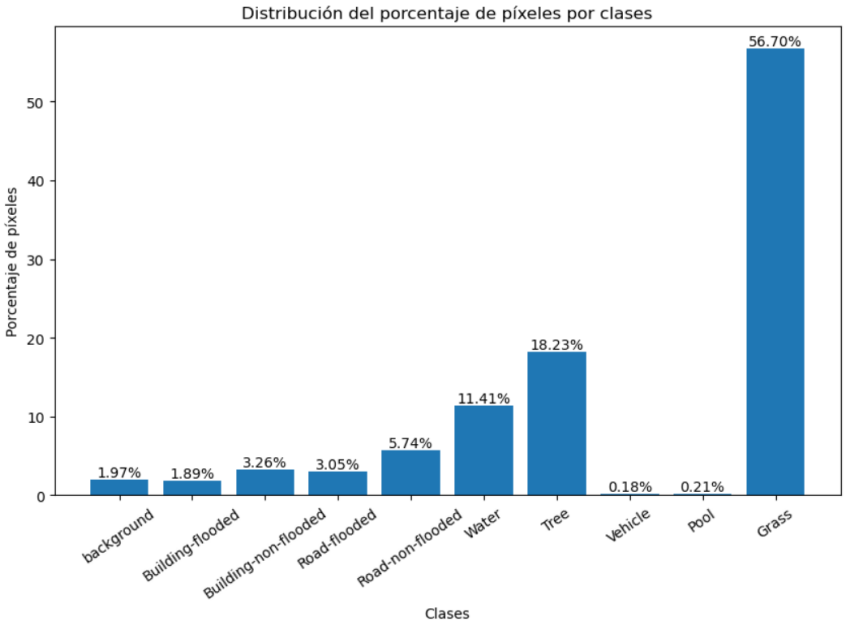

Ilustración 6.6: Distribución de clases en FloodNet

Mediante la ilustración 6.6 se puede observar que en dicho dataset existe un desbalanceo de las clases, donde las clases como "Vehicle" y "Pool" tienen un bajo porcentaje de los datos. Por otro lado, la clase "Grass" tiene un alto porcentaje de datos.

Con dicha distribución se puede saber que es necesario realizar un data augmentation en futuro a las clases minoritarias para que se balanceen adecuadamente su cantidad de datos. Esto es importante porque, con dicha distribución, es probable que el modelo no se entrene muy bien con las clases minoritarias debido a su bajo porcentaje de datos.

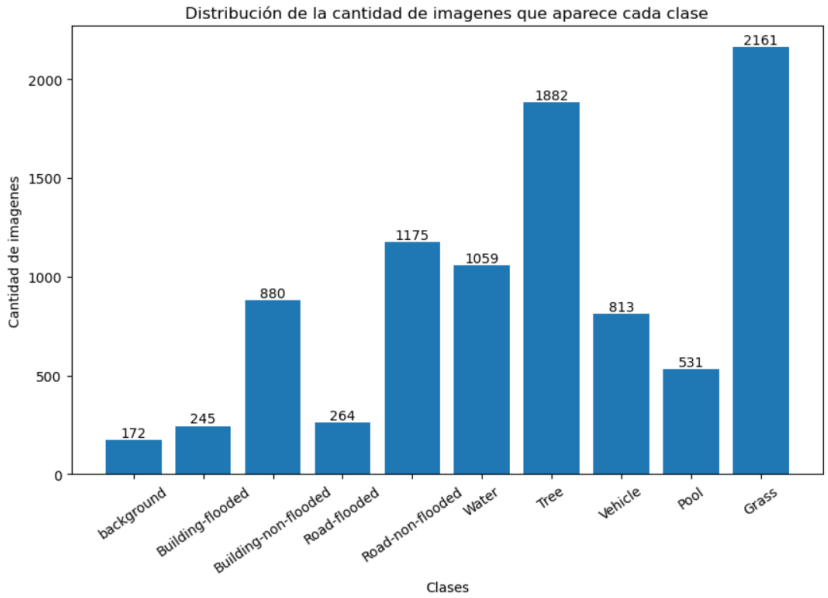

Ilustración 6.7: Cantidad de datos de cada clase en Floodnet

Mediante la ilustración 6.7 se puede observar que las clases “Grass” y “Tree” tienen una alta frecuencia de aparición en cada dato. Esto ayuda al modelo durante el entrenamiento, ya que, al aparecer frecuentemente, se da más oportunidad a la red neuronal de aprovechar dichos datos frecuentes para el entrenamiento y segmentar correctamente dichas clases.

También se puede observar que las clases minoritarias mencionadas en la gráfica anterior tienen una frecuencia no tan baja dentro del dataset, lo cual es una buena señal. Aunque representan un bajo porcentaje del total de datos, una frecuencia moderada puede permitir que la red neuronal aprenda de manera regular estas clases.

El punto de interés de elegir dicho dataset para el TFG son los siguientes:

1. Las clases de vegetación como “Grass” y “Tree” son los objetivos que se quieren detectar durante el desarrollo del proyecto. Este dataset tiene un alto porcentaje y alta frecuencia de dichas clases, lo cual ayuda a que la red neuronal tenga más capacidad para identificar las clases de vegetación.
2. Los datos son recopilados por un dron, por lo tanto, tienen una alta resolución y son similares al dataset que se entrenará en el último paso utilizando la técnica del Transfer Learning.

6.2.2. Dataset DeepGlobe Land Cover

El siguiente dataset que se va a presentar es el dataset DeepGlobe Land Cover, un dataset que contiene clases de diferentes tipos de terreno, como se puede apreciar en el cuadro 6.2. Este consta de un total de 1570 imágenes RGB y sus correspondientes máscaras también en formato RGB. Por lo tanto, para utilizar este dataset será necesario cambiar los valores RGB de cada mascara por el valor correspondiente a cada clase.

Las imágenes tienen una anchura y altura de 2448 x 2448 píxeles, con un total de 7 clases, y la fuente de dicho dataset se puede encontrar en [4].

Clases	COLOR
Urban_land	
Agriculture_land	
Rangeland	
Fores_land	
Water	
Barren_land	
Background	

Cuadro 6.2: Tabla de clases de DeepGlobe

Mediante la visualización de los datos de la ilustración 6.8, se puede observar que este dataset ha sido capturado por un satélite, el cual tiene una resolución espacial más elevada que el dataset recopilado por el dron mencionado anteriormente. A pesar de ser capturado por un

satélite, tampoco alcanza una alta resolución como el dataset anterior. Sin embargo, los datos de este dataset tienen una cobertura global dentro de una sola imagen, lo que es ideal para realizar seguimiento a largo plazo y obtener una perspectiva más global.

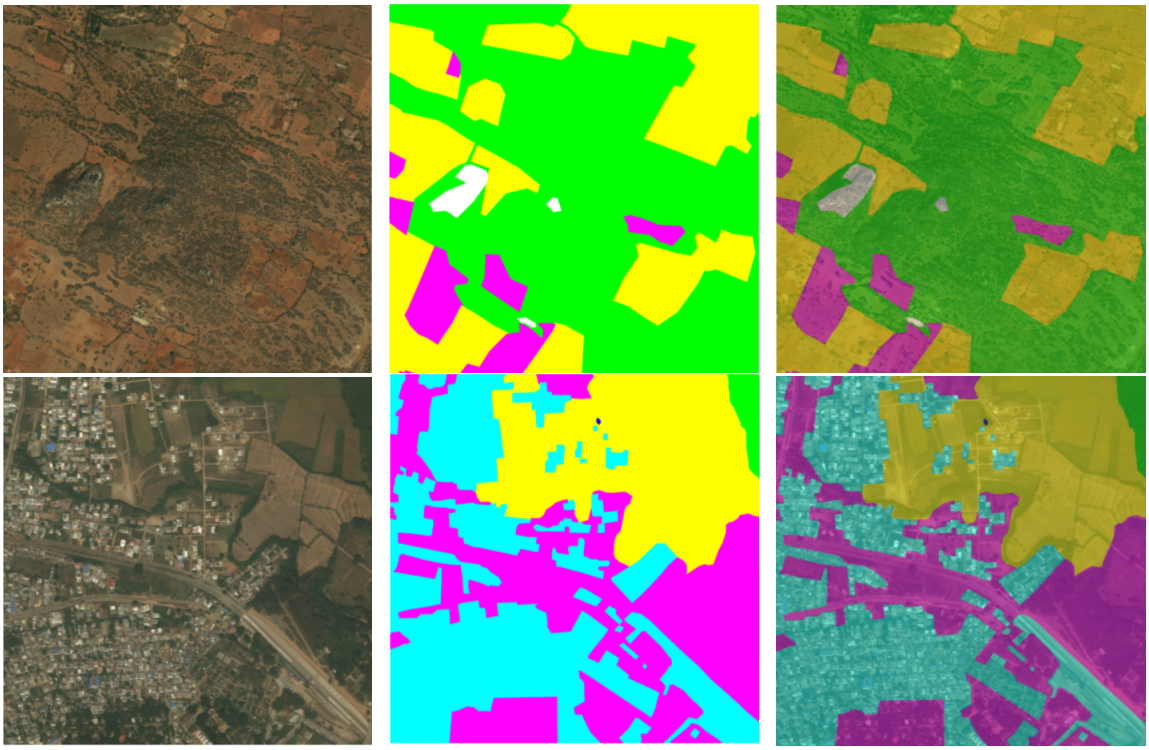

Ilustración 6.8: Datos y mascaras del DeepGlobe

A continuación se muestran las distribuciones de dicho dataset, como en el dataset anterior, el porcentaje de los píxeles por cada clase(Ilustración 6.9) y la cantidad de imágenes que aparece en cada clase(Ilustración 6.10).

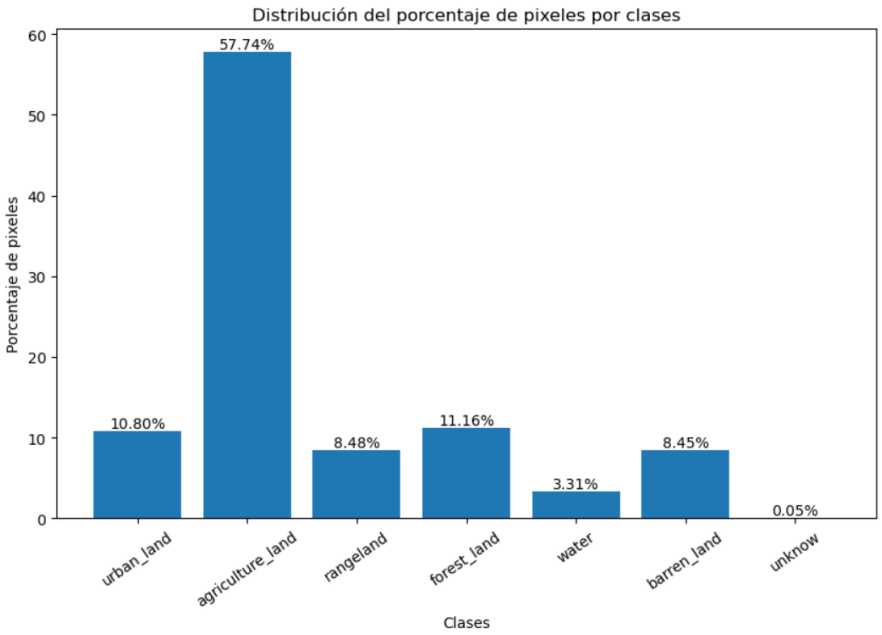

Ilustración 6.9: Distribución de clases en DeepGlobe

Mediante la ilustración 6.9 se puede observar que los datos de la clase "Agriculture Land" tienen un gran porcentaje dentro del dataset, lo cual significa que la red neuronal tiene una alta probabilidad de tender a dicha clase. Sin embargo, afortunadamente, en este dataset no existe un desbalanceo grave como en el dataset anterior, ya que las otras clases tienen un porcentaje equilibrado aproximadamente. En el caso de la clase "Water", aunque tiene un porcentaje del 3 %, pero por sus características destacadas en comparación con otras clases de tipos de terreno, se podría obtener un buen resultado durante el entrenamiento.

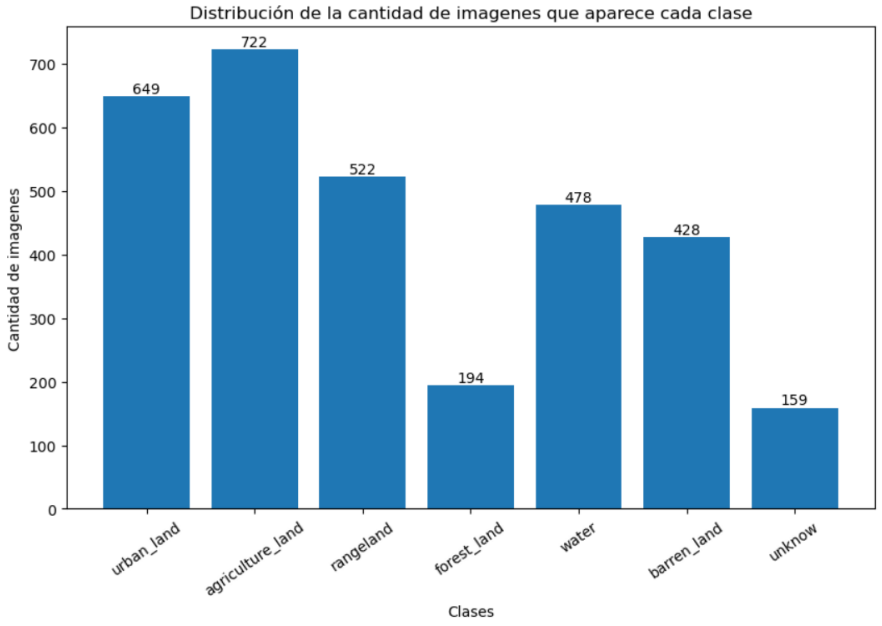

Ilustración 6.10: Cantidad de datos de cada clase en DeepGlobe

Mediante la ilustración 6.10 se puede observar que los datos de la clase “Forest Land” tienen una baja frecuencia, mientras que las otras clases presentan frecuencias similares, excepto la clase “Unknown”.

Los puntos de interés de seleccionar dicho dataset son los siguientes:

1. En dicho dataset se encuentran las clases “Rangeland”, “Agriculture land” y “Forest land” las cuales contienen vegetación y contribuyen al dataset final. Especialmente la clase “Agriculture land” porque en el dataset Flair se encuentra la misma clase también, lo que ayuda mucho a la red neuronal para hacer la predicción sobre dicha clase.
2. Los datos de dicho dataset contribuyen una perspectiva más global que los datos recopilados por el dron, por lo tanto, podrían ser una ayuda a la red neuronal a la hora del entrenamiento.

6.2.3. Dataset Semantic Drone

El penúltimo dataset describe las imágenes capturadas por el dron, utilizando el mismo medio de recopilación de datos que el dataset FloodNet, pero incluye más clases de vegetación y los datos de dicho dataset contienen mayores detalles sobre los datos de la ciudad.

Dicho dataset tiene un total de 400 imágenes, lo cual es más pequeño en comparación con los dos datasets anteriores. Para este dataset, se seleccionan las clases más relevantes y relacionadas con este proyecto, como se puede observar en el cuadro 6.3, donde se cuentan un total de 10 clases. Algunas de estas clases son las mismas que las del FloodNet. Por lo tanto, en el paso posterior de fusionar los datasets, se podría ampliar la diversidad del dataset con el objetivo de ayudar al modelo a obtener una mejor capacidad frente a diferentes circunstancias. La fuente se puede encontrar en: [35]

Clases	COLOR
Background	
Road	
Grass	
Water	
Pool	
Vegetation	
Building	
Person	
Vehicle	
Tree	

Cuadro 6.3: Tabla de clases de Semantic Drone

En la ilustración 6.11, se puede observar que en izquierda se encuentran las imágenes capturadas por el dron, y en derecha se encuentran las máscaras correspondientes y su superposición sobre la imagen RGB.

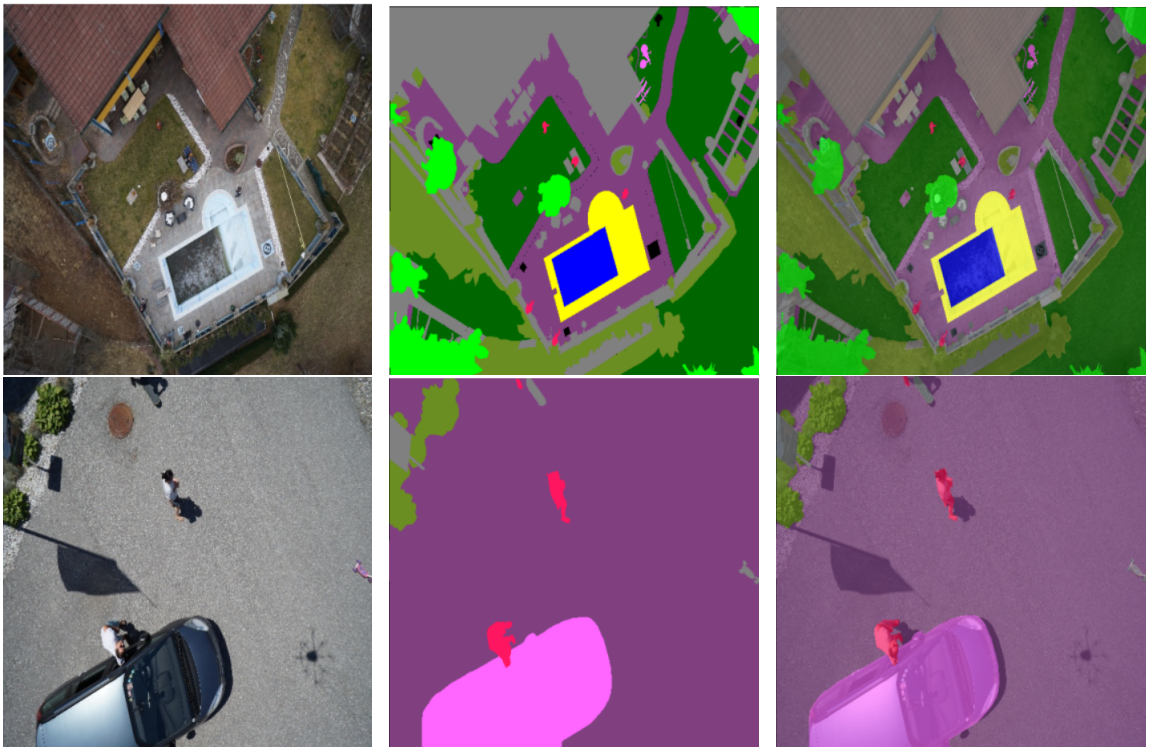

Ilustración 6.11: Datos y mascaras del Semantic Drone

A continuación se muestran las distribuciones de dicho dataset, como en el dataset anterior, el porcentaje de los píxeles por cada clase(Ilustración 6.12) y la cantidad de imágenes que aparece en cada clase(Ilustración 6.13).

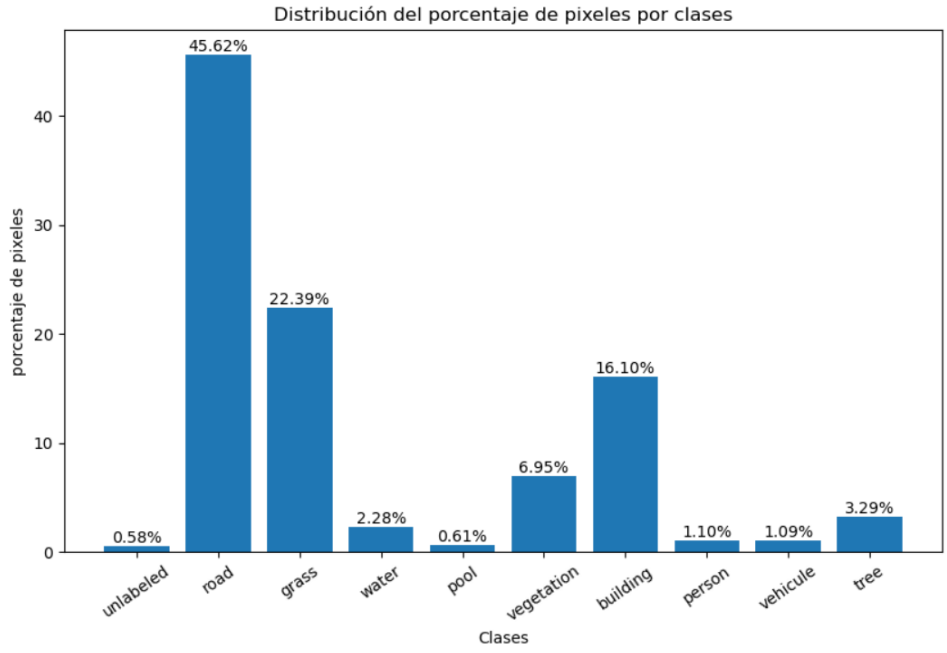

Ilustración 6.12: Distribución de clases en Semantic Drone

Mediante la ilustración 6.12 se puede observar claramente un desbalanceo más grave que en los dos datasets anteriores, donde la clase "Road" tiene un porcentaje del 45 %. Las clases "Grass" y "Building" ocupan el 22 % y el 16 % respectivamente, y el resto de las clases tienen un porcentaje bastante bajo, como el 1 %, 2 % y 3 %, lo cual sería un desafío a la hora de hacer data augmentation adecuadamente por la baja cantidad de datos. Sin embargo, afortunadamente, los datos del dataset FloodNet pueden ayudar a las clases minoritarias de este dataset, porque ambos datasets comparten cierta similitud en las clases.

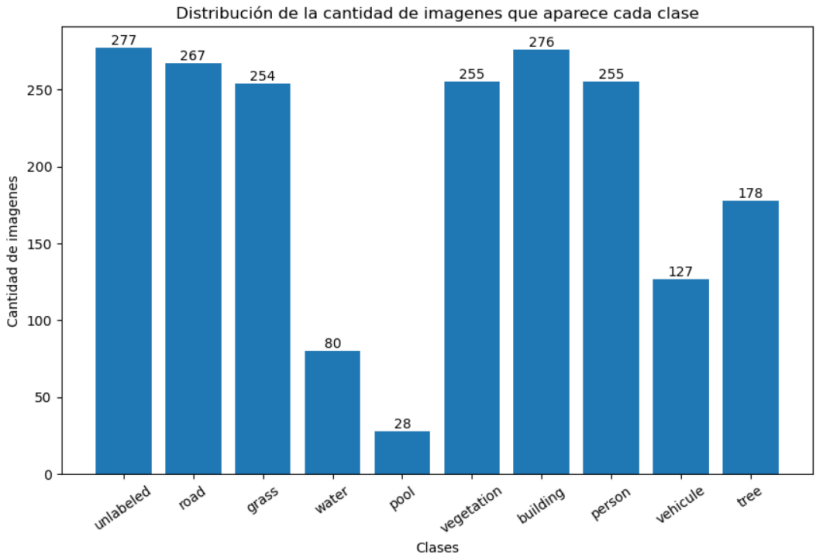

Ilustración 6.13: Cantidad de datos de cada clase en Semantic Drone

El punto de interés al seleccionar dicho dataset se debe a que contienen clases de vegetación y son imágenes aéreas capturadas por el dron, lo cual es similar al dataset "FloodNet". Dicho dataset tienen clases de vegetación como "Vegetation", "Grass", "Tree", y también tienen clases similares al primer dataset, como "Road", "Water", "Pool", "Building" y "Vehicle". Esto será más interesante a la hora de fusionar este dataset con FloodNet.

6.2.4. Dataset Flair

El último datasaet que se va a presentar es uno encontrado en Huggingface, donde tienen una magnitud inmensa de datos. En dicho dataset, hay un total de 61,712 parches destinados al entrenamiento y 15,700 parches para el testeo. Antes de seguir, puede surgir la pregunta: ¿qué es un parche? Un parche no es más que un trozo de una imagen completa. Los investigadores recopilan los datos en imágenes de alta resolución, con un gran ancho y altura. Se hace la anotación de cada imagen y se segmenta en trozos más pequeños. Se pueden obtener ventajas como:

1. Aumentar la cantidad de datos. Al segmentar una imagen de alta dimensión, se puede obtener una mayor cantidad de datos que antes, y la red puede extraer las características locales y aprender con eficiencia.

2. Limitación de memoria. Como es un dataset de un desafío con acceso público, los datos de alta dimensión dificultan a los interesados en ofrecer su idea propia debido a la limitación del equipo físico.

3. Reducción de complejidad. Al reducir los datos en parches, la red neuronal puede concentrarse en áreas más manejables y detalladas.

Dicho dataset está recopilado en el territorio de Francia y está dividido en 2 partes (flair#1, flair#2). La parte flair#1 contiene los datos aéreos con 5 canales (RGB, Infrarrojo y Elevación) y con una resolución espacial de 0.2m, mientras que la otra parte del dataset consiste en datos satelitales con un total de 10 canales. En dicho TFG se procede al procesamiento de los datos del flair#1 con los canales RGB. Los datos de esta parte están divididos por diferentes provincias de Francia, lo que se puede observar en la ilustración 6.14

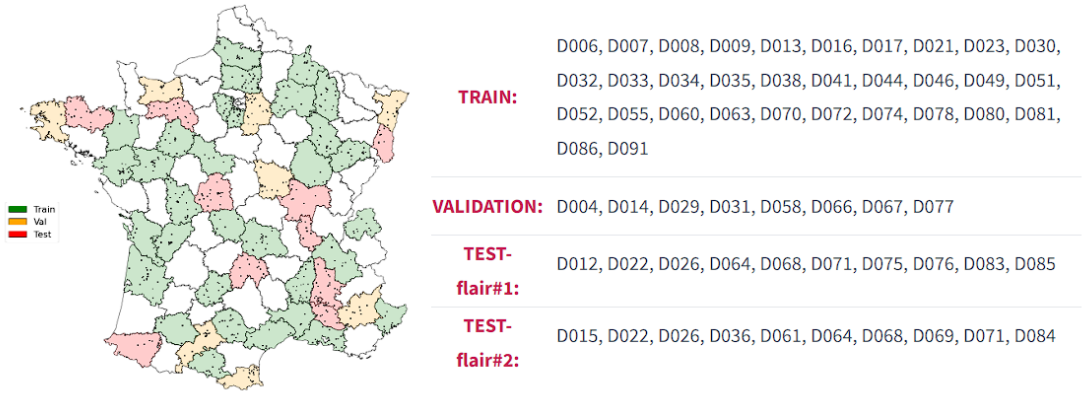

Ilustración 6.14: División del dataset [12]

En la ilustración anterior se puede observar que las provincias en verde están destinadas al entrenamiento, las provincias en amarillo están destinadas a la validación y las provincias en rojo al testeo. Se puede notar que dicho dataset abarca más de la mitad del territorio de Francia para realizar el experimento. Además, la recopilación de los datos destinado al entrenamiento está realizando en manera aleatoria para evitar un porcentaje elevado de la duplicidad de las clases.

En el cuadro 6.4 se puede observar todas las clases que tienen en el dataset Flair

Clases	COLOR
Building	
Pervious surface	
Impervious surface	
Bare soil	
Water	
Coniferous	
Deciduous	
Brushwood	
Vineyard	
Herbaceous vegetation	
Agricultural Land	
Plowed land	
Swimming pool	
Snow	
Clear cut	
Mixed	
Ligneous	
Greenhouse	
Other	

Cuadro 6.4: Tabla de clases del Flair

En la siguiente distribución de los datos de cada clase se puede observar que los datos no están balanceados. Sin embargo, en dicho TFG no se aplica la técnica de data augmentation sobre este dataset, sino que solamente se utiliza el modelo preentrenado en la sección anterior para realizar el experimento de Transfer Learning y probar la eficiencia de los experimentos anteriores.

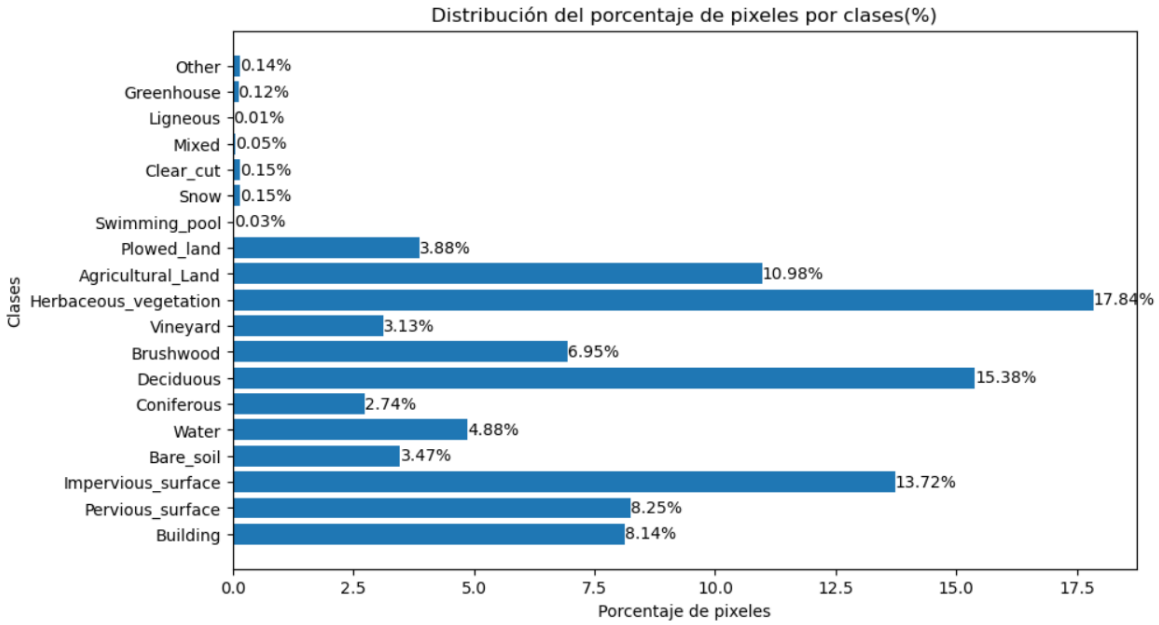

Ilustración 6.15: Distribución de clases en Flair

En dicha distribución se puede observar claramente que las clases como "Herbaceous_vegetation", "Deciduous" e "Impervious_surface" ocupan la mayor cantidad de datos dentro del dataset.

Por otro lado, las clases con menos datos son "Swimming_pool", "Mixed", "Ligneous", "Greenhouse", "Clear_cut" y "Snow", las cuales tienen un porcentaje de datos aproximadamente del 0 %. Esto representa un desafío para el modelo para aprender las características y patrones de estas clases. Sin embargo, entre las clases minoritarias, se encuentra la clase "Swimming_pool", que es la clase que se aparece dentro de los datasets "FloodNet" y "Semantic Drone". Por lo tanto, los modelos entrenado con "FloodNet" y "Semantic Drone" podrían ayudar a capturar las características necesarias para esta clase.

En dicho dataset, la cantidad sobre las clases relativa de la vegetación está bastante rica: "Coniferous", "Deciduous", "Brushwood", "Clear-Cut", "Ligneous", "Mixed" y las clases relacionadas con la vegetación destinada a la agricultura: "Vineyard", "Agricultural Land" y "Plowed Land". Dentro de las clases mencionadas, se diferencia entre el tipo de árbol y el tipo del foreste, donde "Coniferous", "Deciduous" y "Brushwood" son especies de árboles, y "Clear-Cut", "Ligneous", "Mixed" son tipos de terreno forestal. La clase "Clear-Cut" está definida para terrenos forestales donde los árboles ya han sido talados o cosechados; la clase "Ligneous" está definida para terrenos forestales donde las clases "Coniferous" y "Deciduous" están representadas de manera homogénea. Por último, la clase "Mixed" está definida para una representación heterogénea de las clases "Coniferous" y "Deciduous", donde no se puede determinar con suficiente certeza la predominancia de una sobre otra[12].

Por otro lado, las clases relativa de la vegetación en el ámbito de la agricultura son especialmente interesantes. En este caso, lo más destacado son las clases "Vineyard" y "Agricultural Land". la razón de no incluir la clase "Vineyard" dentro del "Agricultural Land" es por su característica distintiva de cobertura terrestre. Y en la clase "Agricultural Land", se incluyen los cultivos principales, pero también abarca los pastizales permanentes y temporales con uso agrícola. Todas las vegetaciones en la clase "Agricultural Land" presentan una cobertura terrestre similar, por lo que se representa como una sola clase. Por otro lado, la clase "Plowed Land" está definida para terrenos sin vegetación visible, como en casos de tierra recién arada o recién cosechada[12].

Por ultimo, la clase relativa a la vegetación es "Herbaceous vegetation", esta definido para el tipo de vegetación sin fin de agrícola, Esta clase incluye céspedes ornamentales (por ejemplo, jardines, parques públicos), campos recreativos (por ejemplo, utilizados para el deporte), zonas herbáceas naturales en zonas boscosas o montañosas, etc.[12]

Las razones de elegir dicho dataset para el último experimento consisten en:

1. Riqueza de las clases de vegetación y el tipo de terreno relativo a la vegetación.
2. Los datos de dicho dataset tienen una magnitud bastante inmensa en comparación con otros datasets encontrados.
3. Una alta similitud con otros datasets encontrados, tanto en el nivel de datos como el nivel de las clases de los datasets, que se permite que los modelos obtenidos por otros experimentos puedan ayudar a obtener un mejor resultado a dicho dataset.

En la siguiente ilustraciones se puede observar los datos de dicho dataset.

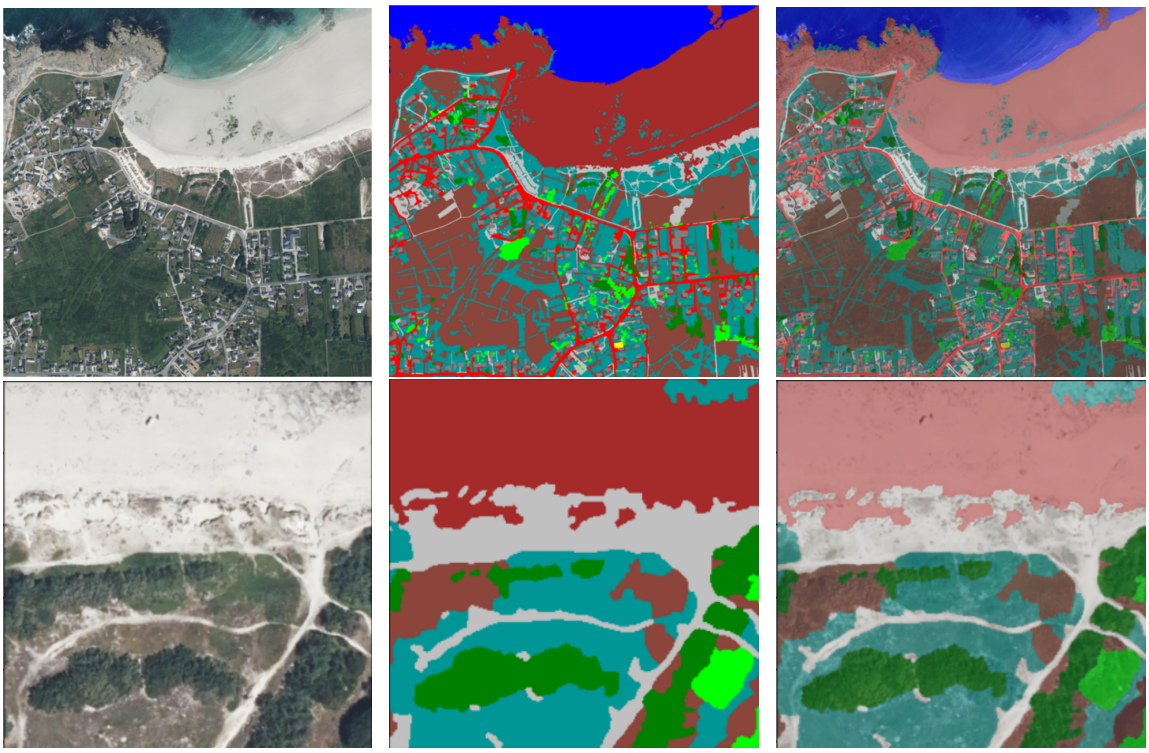

Ilustración 6.16: Datos y mascaras del Flair

En las imágenes de la primera fila está la imagen completada procesada para restaurar la dimensión original. En el caso de las imágenes de la segunda fila, son los datos en el dataset, que es un parche de la imagen original.

6.3. Tarea 3 Preparación previa del entrenamiento

En la tarea anterior se ha buscado unos varios datasets con uso diferentes para llegar al objetivo final del TFG. Pero antes de hacer cualquier experimento, una preparación completa es imprescindible.

6.3.1. Preparación del entorno

En dicha sección se presentará el entorno que se necesita para el desarrollo del proyecto. Un entorno virtual es nada más que un espacio aislado del desarrollo para evitar los conflictos que pueden generarse por diferentes versiones de las bibliotecas que se van a utilizar, y tener un entorno virtual ayuda a hacer testeo sobre el proyecto y aumenta la reproducibilidad.

Para dicho TFG, se realizan todos los trabajos con el framework PyTorch, y se trabaja dentro de un entorno virtual creado por Anaconda. La versión de PyTorch varía según la versión de CUDA instalada en cada dispositivo físico. En el caso de este TFG, se usa la versión 11.7, que es la versión más reciente y estable en este momento. En la ilustración 6.17 se muestra la versión del Driver de GPU, que determina la máxima versión de CUDA soportada.

Ilustración 6.17: Información sobre CUDA Driver

A la ilustración 6.18 se muestra la versión de CUDA en uso:

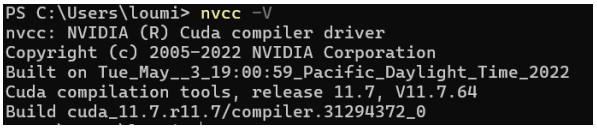

Ilustración 6.18: Información sobre la versión de CUDA

6.3.2. Pre-Porcesamiento de imágenes

Antes del entrenamiento, se realiza un pre-procesamiento sobre los datos de cada dataset. Dado que los datos de cada dataset tienen una magnitud grande, lo que provocaría un uso elevado de la memoria de la tarjeta gráfica, y el equipos físicos actuales es imposible entrenar un dataset con dichas magnitudes y el tiempo de entrenamiento se puede prolongar significativamente.

En función de las razones anteriores, se realiza un redimensionamiento de las imágenes a 512x512 píxeles para todo los datos durante el entrenamiento. Primero, las dimensiones modificada de los datos corresponden a la dimensión del último dataset(Flair). Además, con la dimensión 512x512, se agiliza el uso de la tarjeta gráfica y permite el entrenamiento con un batch size adecuado.

6.4. Tarea 4 Entrenamiento inicial

En esta sección se hace los entrenamientos necesarios sobre los datasets encontrados, excepto el dataset Flair, porque para el dataset Flair, se hace el experimento del Transfer Learning. El motivo de hacer los entrenamientos de diferentes datasets consisten en validar el uso de la técnica de data augmentation en la seccion posterior y tener en claro la capacidad de la arquitectura seleccionada y posibles problemas que se puede encontrar durante el entrenamiento.

Tras hacer todas las configuraciones adecuadas, se procede a realizar todas las pruebas de entrenamiento necesarias para poder llevar a cabo dicho TFG.

Para estandarizar todos los entrenamientos, se establecen los siguientes hiperparámetros:

1. Device: cuda
2. Modelo: deeplabv3_resnet50 con peso preentrenado
3. Funcion de perdida: CrossEntropyLoss
4. Optimizador: Adam
5. Batch size: 8
6. Learning rate: 1e-3 con un scheduler para baja 5 % en cada época

6.4.1. Entrenamiento sobre FloodNet

El primer entrenamiento que se realiza es con el dataset FloodNet. Tras realizar 30 épocas del entrenamiento, se observa que el valor del Loss en la validación ha dejado de disminuir significativamente, lo cual es una señal de que el modelo puede estar experimentando overfitting en el futuro. Esto significa que el modelo ya no está aprendiendo de manera efectiva y está intentando memorizar los datos para obtener métricas aparentemente mejores.

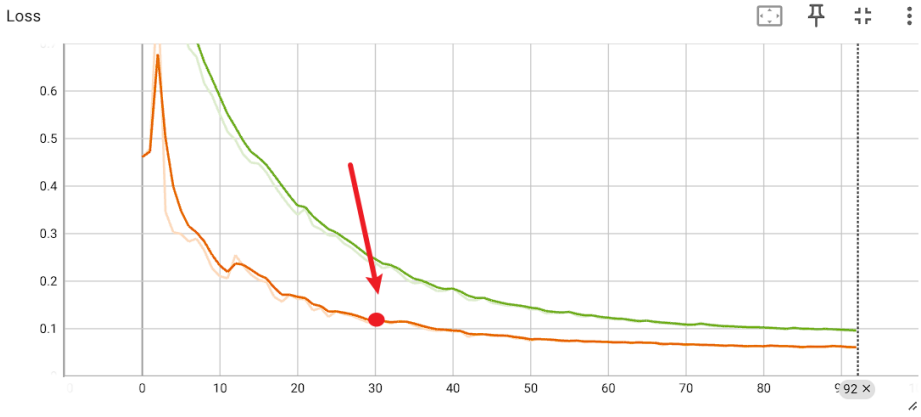

Ilustración 6.19: Evolución de Loss en entrenamiento del FloodNet

Train/Validation	Loss	COLOR
Train	0.2378	
Vlidation	0.1122	

Cuadro 6.5: Tabla de Loss en época 30

Por lo tanto, se seleccionan los pesos de la época 30 con un loss de validación de 0.1122 para el trabajo futuro. Las métricas sobre el testeo(Ilustración 6.7), el mapa de calor de cada clase(Ilustración 6.21) y la evolución de las métricas del modelo(Ilustración 6.20) se pueden observar en las siguientes imágenes:

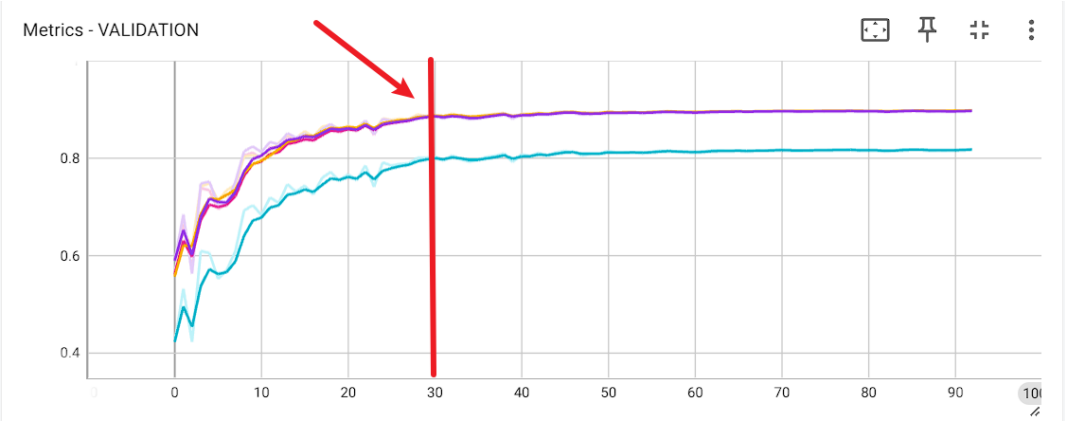

Ilustración 6.20: Evolución de métricas en entrenamiento del FloodNet

Métricas	Valor	COLOR
IOU	0.8	
DICE	0.89	
RECALL	0.89	
PRECISION	0.89	

Cuadro 6.6: Métricas en época 30

Mediante la ilustración 6.20 de la evolución de las métricas se puede observar claramente que las métricas posteriores a la época 30 ya no aumentan significativamente. Aunque siguen aumentando, pero en una magnitud bastante pequeña comparada con la anterior.

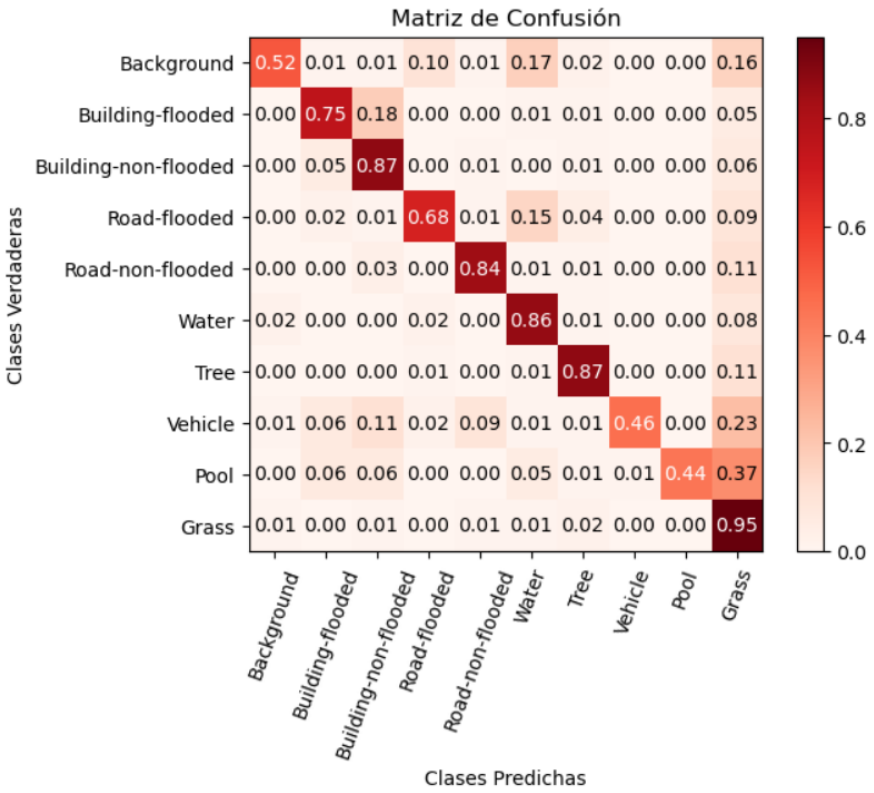

Ilustración 6.21: Mapa de calor en época 30

Mediante el mapa de calor 6.21, se puede observar que en las clases minoritarias como "Vehicle" y "Pool", debido a su baja cantidad de píxeles en el dataset, el modelo tiende a inferir la clase que tiene más datos en el dataset, en este caso, la clase "Grass". En el caso de la clase "Pool", esto es aún más evidente, ya que el modelo infiere que los datos de la clase "Pool" tienen un 44 % de probabilidad de ser clasificados como clase "Pool", pero también tienen un 37 % de probabilidad de ser clasificados como clase "Grass", es casi la mitad de probabilidad.

Clases	IOU(%)	DICE(%)	PRECISION(%)	RECALL(%)
Background	41.15	58.31	65.98	52.24
Building-flooded	58.70	73.97	72.60	75.39
Building-non-flooded	67.47	80.58	75.33	86.61
Road-flooded	51.84	68.28	68.35	68.22
Road-non-flooded	77.42	87.27	91.02	83.82
Water	75.92	86.31	86.45	86.18
Tree	79.84	88.79	91.01	86.67
Vehicle	34.49	51.29	58.64	45.58
Pool	38.06	55.14	74.80	43.66
Grass	87.61	93.39	92.29	94.52
Total	81.23	89.25	89.28	89.34

Cuadro 6.7: Métricas del testeo en época 30

Después de obtener las métricas del test, se puede confirmar lo que se ha dicho previamente, que las clases “Pool” y “Vehicle” tienen métricas de IOU bajas en comparación con otras clases.

En la ilustración 6.22 se puede ver las inferencias hechas por la red neuronal:

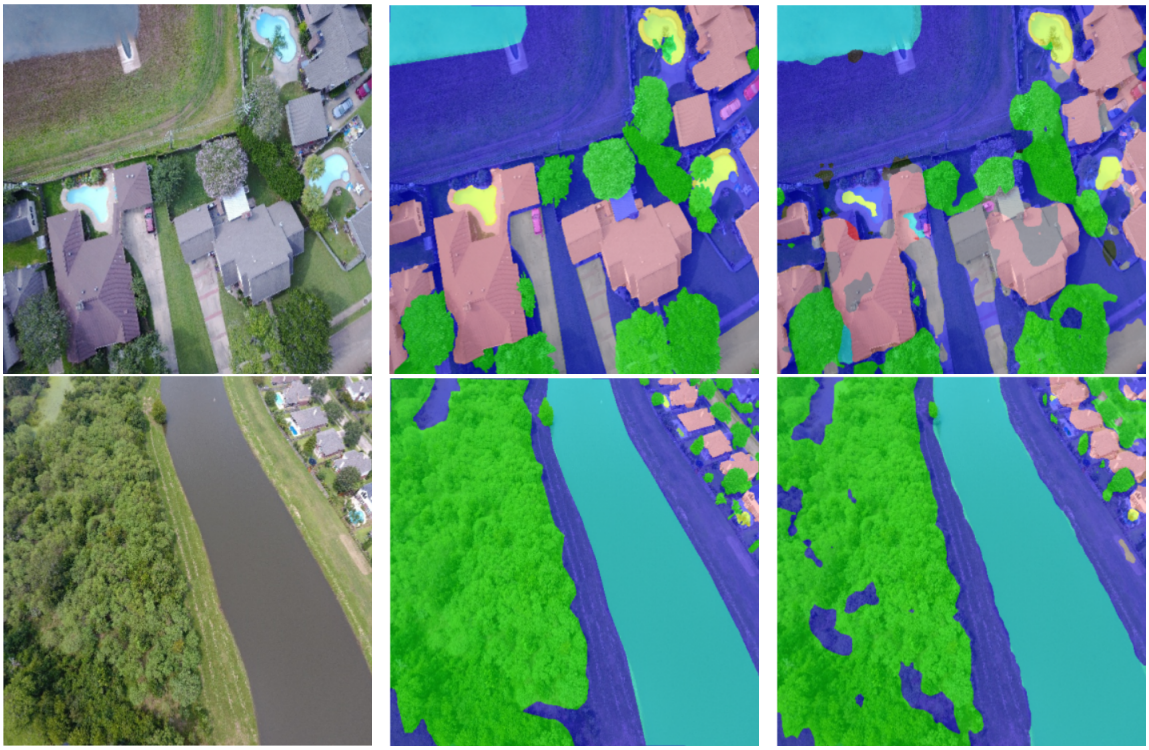

Ilustración 6.22: Datos, máscaras, inferencias del Floodnet

Mediante la visualización de las inferencias hechas por la red neuronal y sus máscaras correspondientes, se puede observar que el modelo tiene una alta precisión sobre las clases mayoritarias. En cambio, entre las clases minoritarias, la clase “Pool” muestra un mejor comportamiento que la clase “Vehicle”.

Después de obtener el modelo entrenado con el dataset FloodNet, se procede a entrenar con los otros datasets restantes. El objetivo final es realizar diferentes combinaciones de los datasets para aprovechar los conocimientos obtenidos y aplicar Transfer Learning sobre el dataset del último apartado.

6.4.2. Entrenamiento sobre DeepGlobe

Se sigue el mismo procedimiento que en el entrenamiento anterior. Se ha entrenado un total de 100 épocas, donde el mínimo loss encontrado en la validación se encuentra en la época 41, con un loss de validación de 0.206, como se puede observar en la siguiente imagen que muestra la evolución del loss. Durante el entrenamiento posterior, se puede observar fácilmente que

el loss de validación se mantiene en niveles aproximadamente iguales. Esto es una señal de que el modelo podría estar experimentando overfitting durante el entrenamiento.

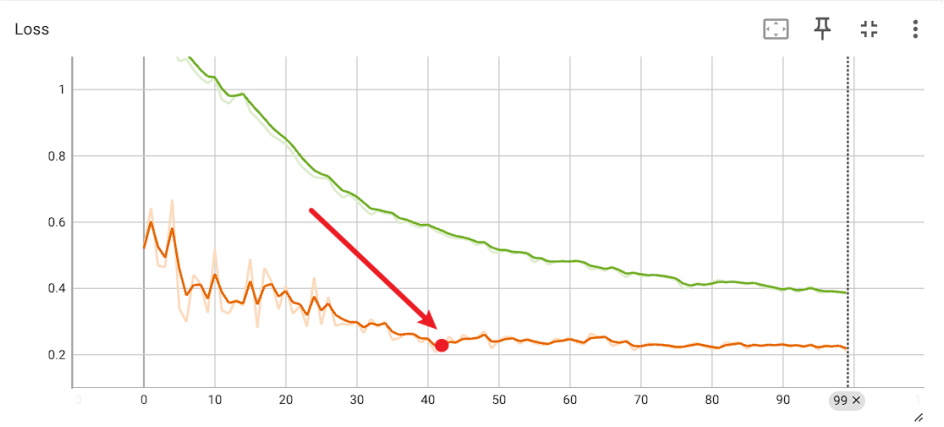

Ilustración 6.23: Evolución de Loss en entrenamiento del DeepGlobe

Train/Validation	Loss	COLOR
Train	0.5727	
Vlidation	0.206	

Cuadro 6.8: Tabla de Loss en época 41

En la ilustración 6.24 se puede observar la evolución de cada métrica durante el entrenamiento, similar al apartado anterior. Después de la época 41, el incremento de cada métrica es cada vez menor que antes. También se puede observar que entre las épocas 10 y 20, hubo una caída significativa de las métricas, pero se recuperaron y mejoraron en pocas épocas. Este fenómeno puede ser causado por:

- Learning Rate Scheduler: En las épocas iniciales, el learning rate es más alto, lo que provoca una exploración más agresiva en las fases tempranas del entrenamiento(causa de la caída de métricas). Posteriormente, mediante el Learning Rate Scheduler, el learning rate disminuye progresivamente, lo que permite ajustes más finos y precisos en las épocas posteriores para obtener las mejores métricas
- Encuentro de un mínimo local: Cuando se entrena la red, se puede encontrar un mínimo local, punto de silla, etc., porque la superficie de optimización de una red neuronal puede ser bastante irregular con muchos valles y colinas. El modelo puede caer fácilmente en un mínimo local. En dicho caso, como tiene un learning rate alto inicialmente, esto puede ayudar a salir de dicho mínimo local y buscar un mínimo global y obtener las mejores métricas.

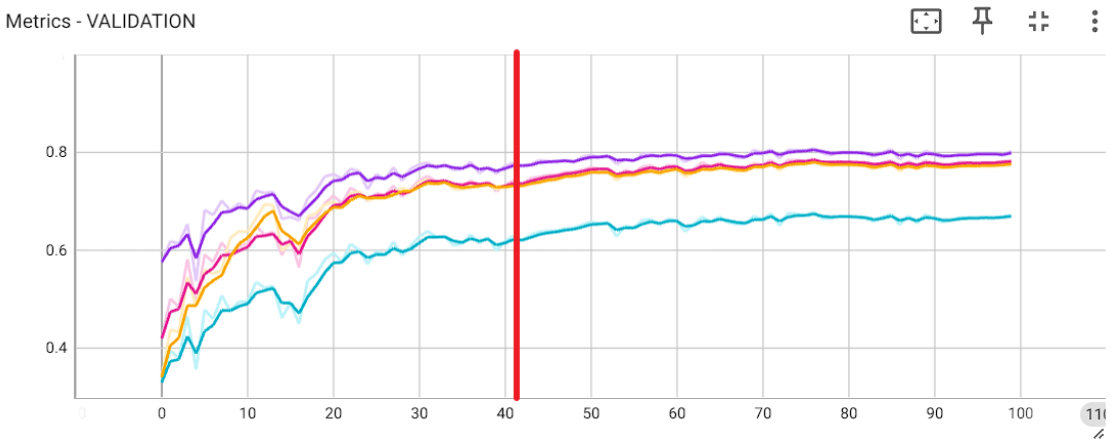

Ilustración 6.24: Evolución métricas en entrenamiento del DeepGlobe

Métricas	Valor	COLOR
IOU	0.74	
DICE	0.89	
RECALL	0.77	
PRECISION	0.73	

Cuadro 6.9: Métricas en época 41

Las métricas sobre el testeo(Cuadro 6.10), el mapa de calor de cada clase(Ilustración 6.25) se pueden observar en las siguientes imágenes:

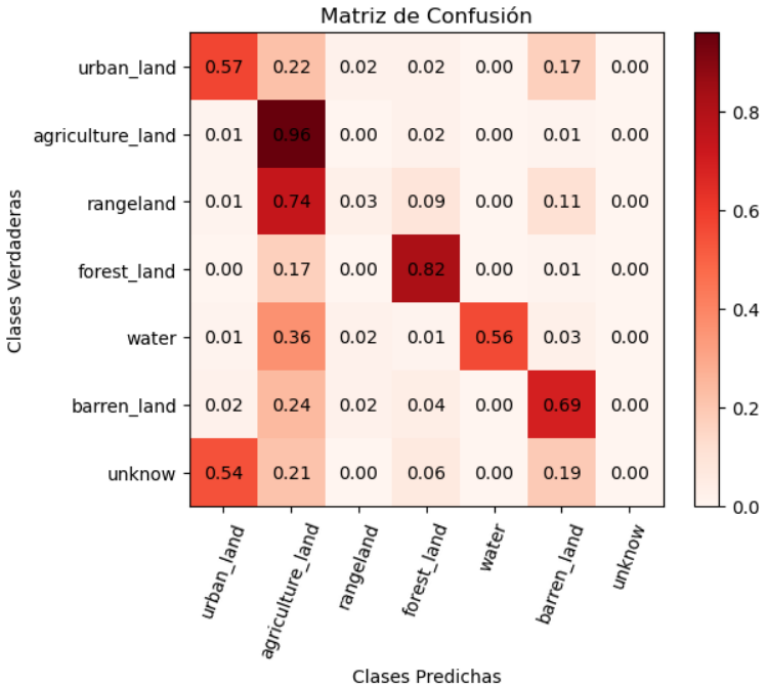

Ilustración 6.25: Mapa de calor en época 41

Clases	IOU(%)	DICE(%)	PRECISION(%)	RECALL(%)
Urban land	53.26	69.50	88.42	57.25
Agriculture land	77.14	87.09	79.65	96.07
Rangeland	3.28	6.36	35.65	3.49
Forest land	67.34	80.48	79.17	81.85
Water	50.51	67.12	82.94	56.37
Barren land	48.61	65.42	62.58	68.54
Total	63.64	74.71	75.24	78.40

Cuadro 6.10: Métricas del testeo en época 41

Se ha observado que las métricas para cada clase tienen un comportamiento diferente debido al desbalanceo de las clases. En este dataset, la clase "Unknown", debido a su baja presencia de píxeles en el dataset total, no se considerará como una clase igual a las demás, ya que tiene casi un 0 % de presencia de píxeles en todo el dataset.

La clase "Rangeland" es la penúltima clase con menos datos en el dataset, que tiene una métrica de IOU de aproximadamente un 3 %, y otras métricas como DICE y RECALL también son muy bajas. Sin embargo, mediante la distribución de clases (Ilustración 6.9) en el apartado anterior, se puede observar que la clase real minoritaria es "Water", que representa un total del 3.31 % de datos en todo el dataset. Pero la clase "Water" es una clase que tiene características claras en comparación con las otras clases, que son tipos de tierra, y esto ayuda a la red neuronal a aprender durante el entrenamiento. Por lo tanto, surgen las siguientes suposiciones:

1. La cantidad de datos sobre la clase "Range Land" no son significativamente altos. Por lo tanto, en el siguiente paso de data augmentation, se realizará especialmente sobre la clase "Range Land" para que el modelo pueda tener más datos sobre dicha clase para aprender mejor sus características propias y diferenciarla con la clase "Agriculture Land".

2. Mediante el mapa de calor se puede apreciar que, en su mayoría, las clases "Agriculture Land" y "Forest Land" muestran mejores métricas en comparación con otras clases, ya que ocupan un mayor porcentaje de datos en el dataset y tienen las características más claras, el modelo tiende a predecirlas. Esto se puede observar en el mapa de calor, donde el modelo tiende a inferir la clase "Agriculture Land" en lugar de "Rangeland" en la segmentación de la clase "Rangeland". Esto puede ser debido a que las características de las clases "Agriculture Land" y "Rangeland" son bastante similares, y como el dataset tiene más datos sobre "Agriculture Land", el modelo tiende a inferir las clases "Agriculture Land" en lugar de "Rangeland".

En la ilustración 6.26 se muestran las inferencias realizadas por el modelo:

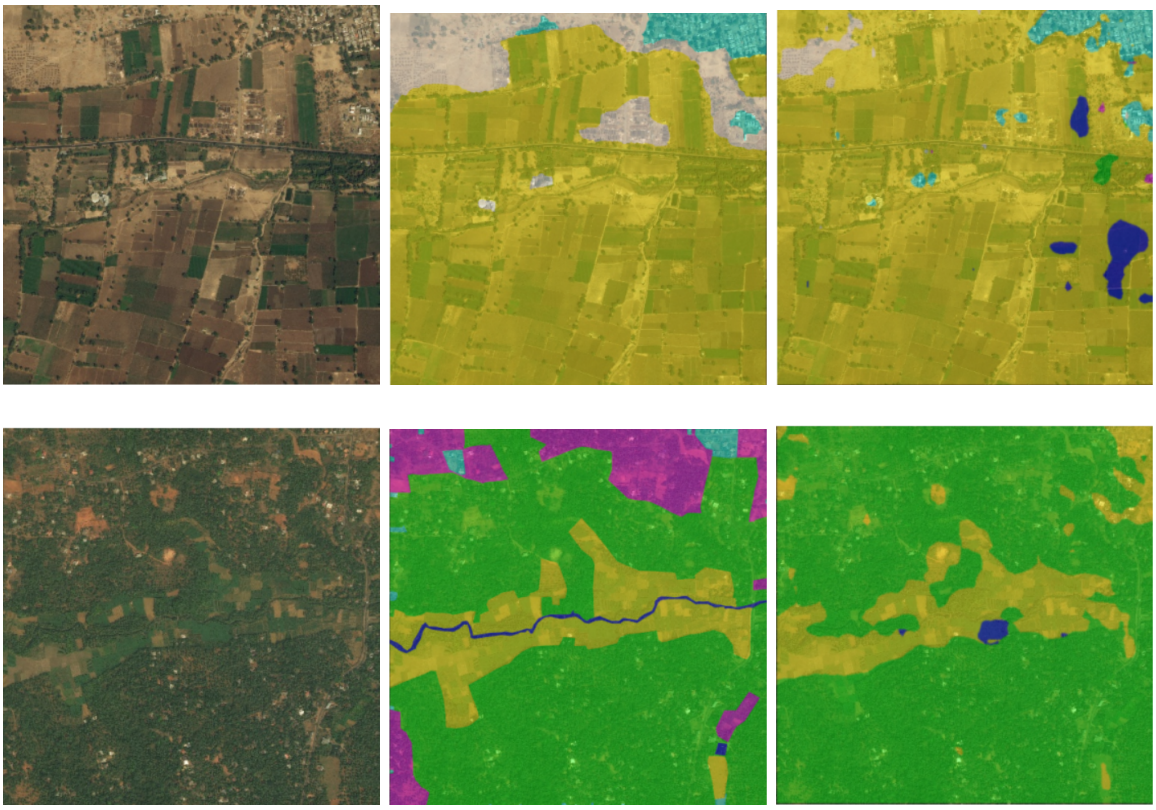

Ilustración 6.26: Datos, máscaras, inferencias del DeepGlobe

6.4.3. Entrenamiento sobre Semanric Drone

Por ultimo se entrena el dataset Semanric Drone con los mismos hiperparámetros que en los entrenamientos anteriores, se ha entrenado un total de 82 épocas, y se ha encontrado el mínimo loss estable de validación, 0.1278, en la época 50, como se puede observar en la ilustración 6.27. En las épocas posteriores, aunque el loss sigue disminuyendo, su progreso es bastante lento en comparación con las épocas anteriores.

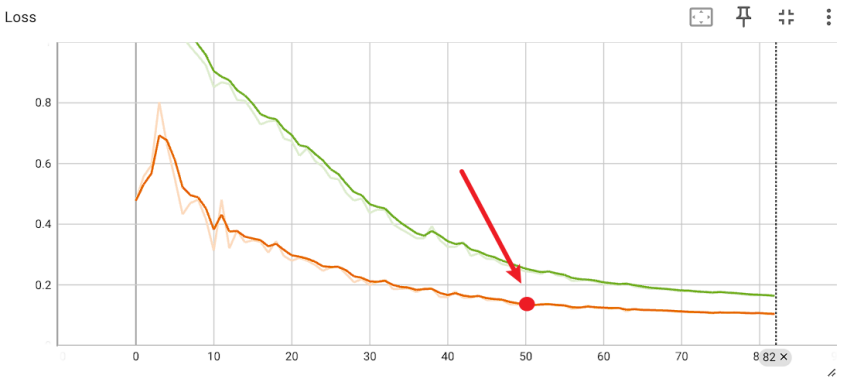

Ilustración 6.27: Evolución de Loss en entrenamiento del Semantic Drone

Train/Validation	Loss	COLOR
Train	0.2431	
Validation	0.1278	

Cuadro 6.11: Tabla de Loss en época 50

En la ilustración 6.28 se puede observar la evolución de cada métrica durante el entrenamiento, similar al apartado anterior. También se puede observar el mismo fenómeno que ocurrió en el entrenamiento anterior: entre las épocas 10 y 15, hay una caída significativa, pero después se recupera y mejora las métricas.Después de la época 50, el incremento de cada métrica es cada vez menor que antes o mantenido en el mismo nivel.

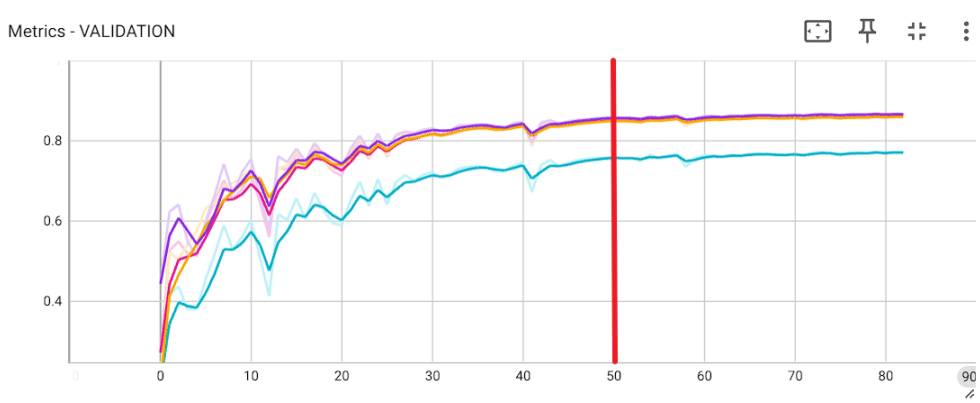

Ilustración 6.28: Evolución de métricas en entrenamiento del Semantic Drone

Métricas	Valor	COLOR
IOU	0.75	
DICE	0.85	
RECALL	0.86	
PRECISION	0.85	

Cuadro 6.12: Métricas en época 50

Las métricas sobre el testeo (Cuadro 6.13), el mapa de calor de cada clase(Ilustración 6.29) se pueden observar en las siguientes imágenes:

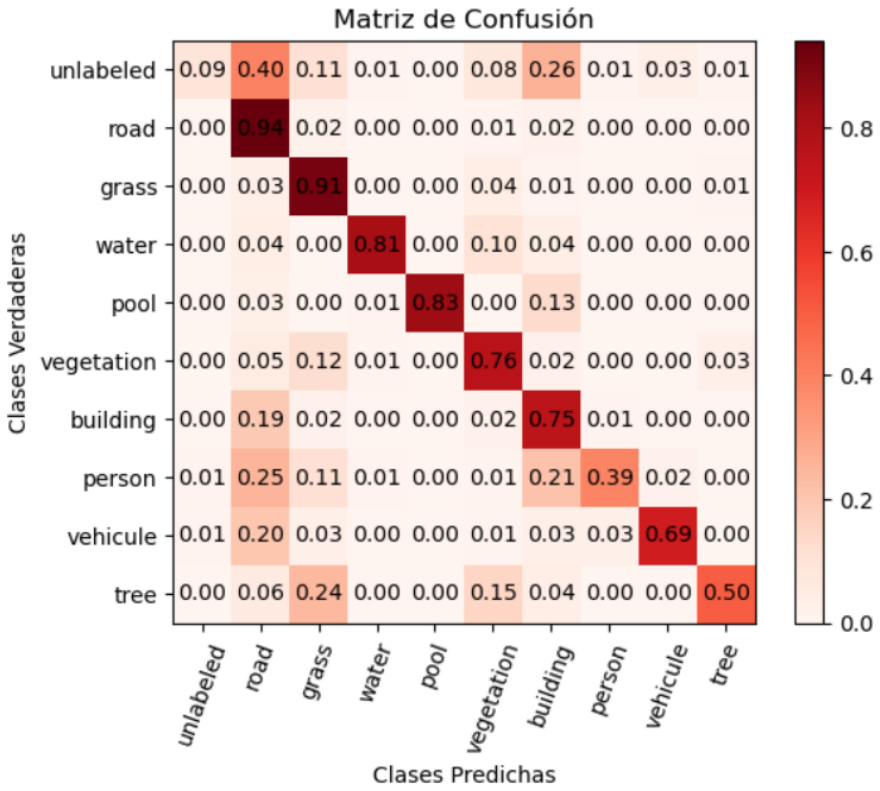

Ilustración 6.29: Mapa de calor en época 50

Clases	IOU(%)	DICE(%)	PRECISION(%)	RECALL(%)
Background	8.71	16.03	52.71	9.45
Road	85.36	92.10	90.64	93.61
Grass	76.93	86.96	83.50	90.72
Water	71.13	83.13	85.45	80.94
Pool	73.67	84.84	87.09	82.71
Vegetation	61.23	75.96	76.06	75.85
Building	65.43	79.10	83.27	75.33
Person	29.83	45.95	56.76	38.60
Vehicle	57.45	72.98	77.59	68.88
Tree	42.74	59.89	74.50	50.07
Total	75.85	85.59	85.63	86.03

Cuadro 6.13: Métricas del testeo en época 50

Mediante las métricas y el mapa de calor se puede observar que las clases, especialmente “Person”, “Vehicle” y “Tree”, tienen métricas bajas, probablemente debido a la baja cantidad de datos en el dataset. Por lo tanto, se realiza data augmentation sobre estas 3 clases especialmente.

Por otro lado, mediante el mapa de calor, también se puede observar que el modelo no

diferencia muy bien la clase “Person” con otras clases como “Road” y “Building”. Se supone que todos los datos de la clase “Person” están siempre relacionado con los datos de las clases “Road” y “Building”, como “una persona caminando en el camino” o “una persona dentro del edificio o cerca de él”. Esta situación también se presenta con la clase “Vehicle”, ya que en los datos de la clase “Vehicle” están relacionadas fuertemente con los datos de la clase “Road”.

En la ilustración 6.30 se muestran las inferencias realizadas por el modelo:

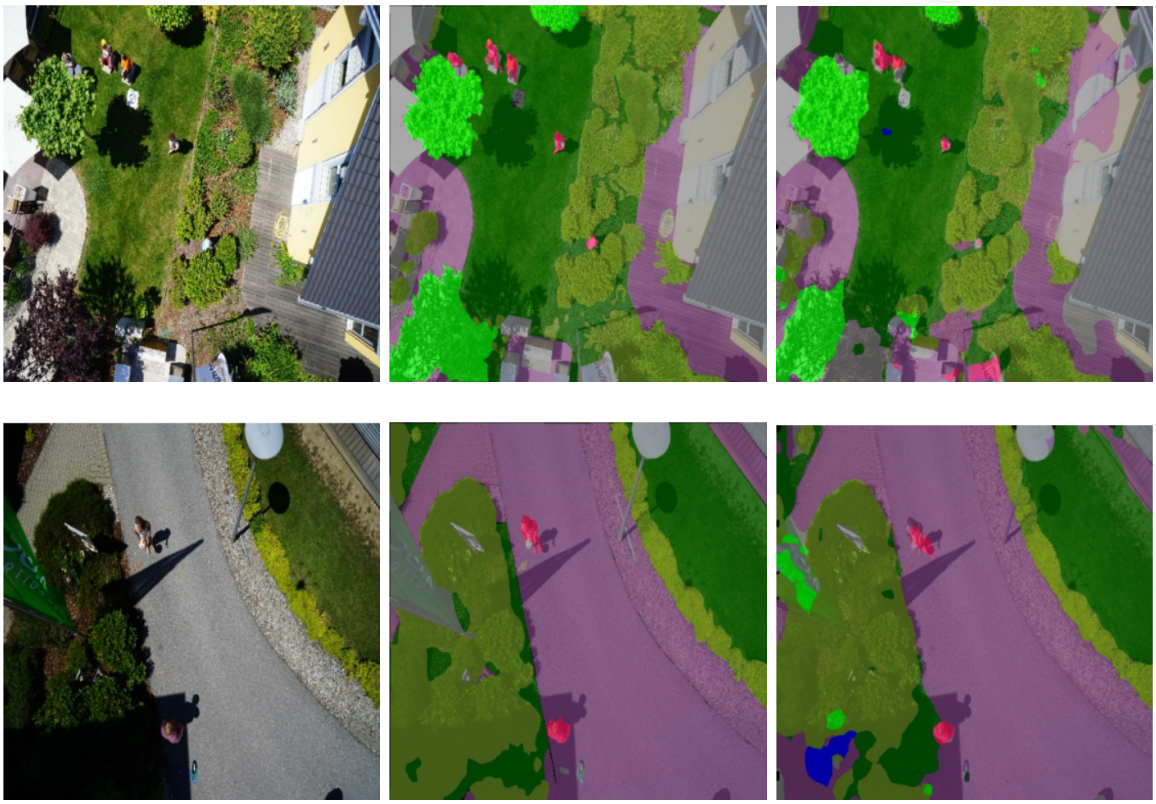

Ilustración 6.30: Datos, máscaras, inferencias del Semantic Drone

Mediante la visualización y comparación sobre las inferencias hechas por el modelo y sus máscaras, se puede observar que las clases “Road”, “Building”, “Tree”, “Grass” y “Vegetation” tienen una precisión bastante alta. La clase “Person” no tiene una precisión alta, pero el modelo puede determinar su ubicación dentro de la imagen, no obstante, el modelo entrenado es débil para segmentar los detalles de la clase “Person”, como el borde.

6.4.4. Análisis de los experimentos

Mediante los experimentos anteriores se puede notar que casi todos los datasets tienen desbalanceo de datos, desde un nivel leve hasta un nivel grave porque la red neuronal no puede aprender adecuadamente las clases minoritarias debido a la carencia de datos de dichas

clases. Donde la red tiende a predecir los resultados a favor de las clases que tienen un mayor porcentaje de datos en el dataset. Por lo tanto, se reduce la precisión de la inferencia sobre las clases que tienen una menor cantidad de datos.

Por lo tanto, en la siguiente sección se centrará en la técnica de Data Augmentation con el objetivo de mejorar las métricas de cada dataset y utilizando los datos obtenidos tras la data augmentation para hacer diferentes combinaciones.

6.5. Tarea 5 Entrenamiento con Data Augmentation

Para aliviar problema de desbalanceo de dataset, en dicha sección se utiliza la técnica de Data Augmentation. El objetivo principal consiste en utilizar diferentes formas para aumentar la cantidad de datos, como recorte, rotación, cambio de color, entre otros, como se muestra en la ilustración 6.31:

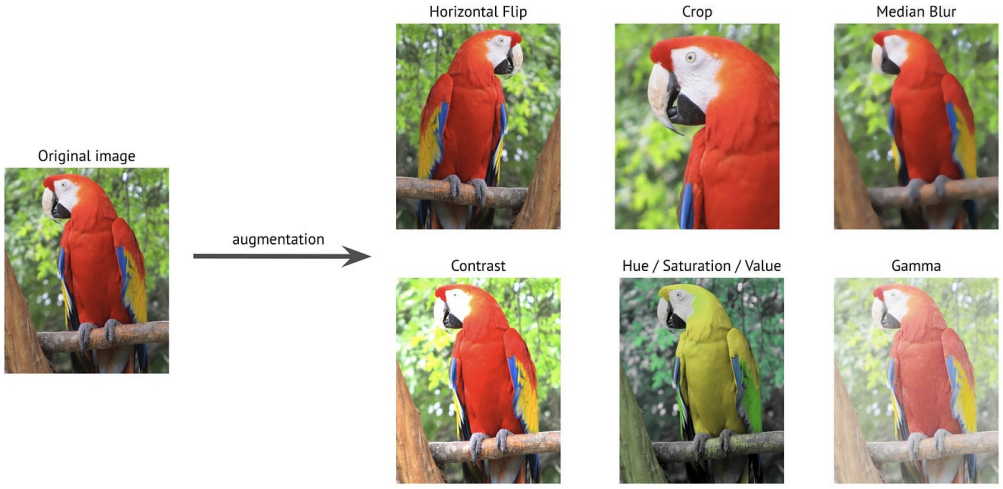

Ilustración 6.31: Data Augmentation

En esta sección se realizan la combinación de transformaciones geométrico y en nivel del píxel. En el ámbito del data augmentation, existen dos niveles de transformación:

- Transformación a nivel del píxel: es una transformación donde se realizan cambios directamente en los valores de los píxeles de la imagen. Estas transformaciones incluyen el cambio de brillo, contraste, entre otros, pero no implican cambios en la posición de cada píxel de la imagen.
- Transformación geométrica: es una transformación que difiere de la transformación a nivel del píxel. En esta transformación, no se modifican los valores de los píxeles de la imagen, sino que se altera su posición, como la rotación horizontal, vertical, volteo, entre otros.

En la siguiente imagen se muestra las diferentes transformaciones se va a aplicar sobre los datos a todos los datasets:

1. Corte de 4 segmentos
2. Volteo Horizontal
3. Volteo Vertical
4. Añadir lluvia
5. Añadir sombra

6. Rotación 90 grado con 1 o más veces

Ilustración 6.32: Imagen Original

Ilustración 6.33: Volteo Horizontal

Ilustración 6.34: Volteo Vertical

Ilustración 6.35: Lluvia

Ilustración 6.36: Sombra

Ilustración 6.37: Rotación

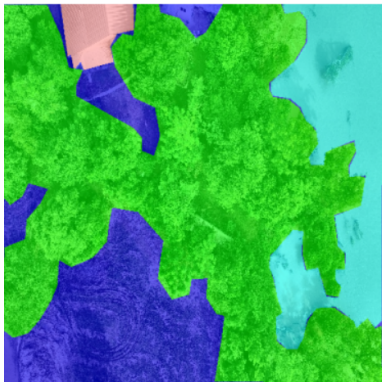

Ilustración 6.38: Máscara Transformada

6.5.1. FloodNet con Data Augmentation

En la ilustración 6.39 se presenta el resultado tras de aplicar data augmentation sobre dicho dataset.

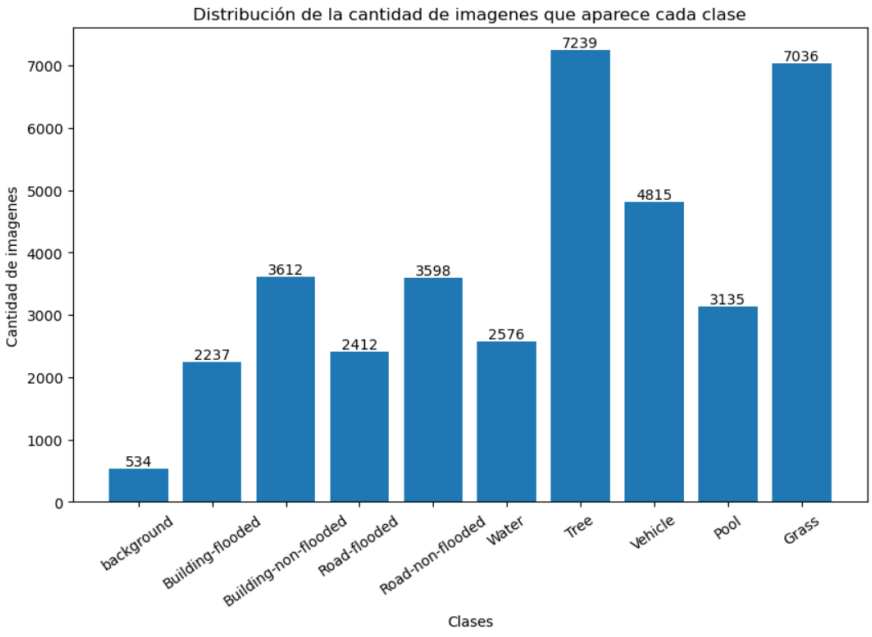

Ilustración 6.39: Cantidad de datos aumentados de cada clase en FloodNet

En comparación con el dataset anterior sin data augmentation, las cantidades aumentadas de cada clase se pueden observar en el cuadro 6.14. Aunque se realiza data augmentation sobre las clases minoritarias, pero los datos de estas clases siempre contienen los datos de las clases mayoritarias, como la clase "Grass". Por lo tanto, cuando se hace data augmentation sobre la clase minoritaria, también se aumenta la cantidad de datos sobre las clases mayoritarias. Hay que considerar dicha relación.

CLASES	RESULTADOS
BACKGROUND	362 ↑
BUILDING-FLOODED	1992 ↑
BUILDING-NON-FLOODED	2732 ↑
ROAD-FLOODED	2148 ↑
ROAD-NON-FLOODED	2423 ↑
WATER	1517 ↑
TREE	5357 ↑
VEHICLE	4002 ↑
POOL	2604 ↑
GRASS	4875 ↑

Cuadro 6.14: Cantidad de imágenes aumentadas en FloodNet

Seleccionado el modelo entrenado con data augmentation en la época 31 con un loss 0.1162 en validación. En la ilustración 6.40 se puede observar las inferencias del modelo tras de data augmentation:

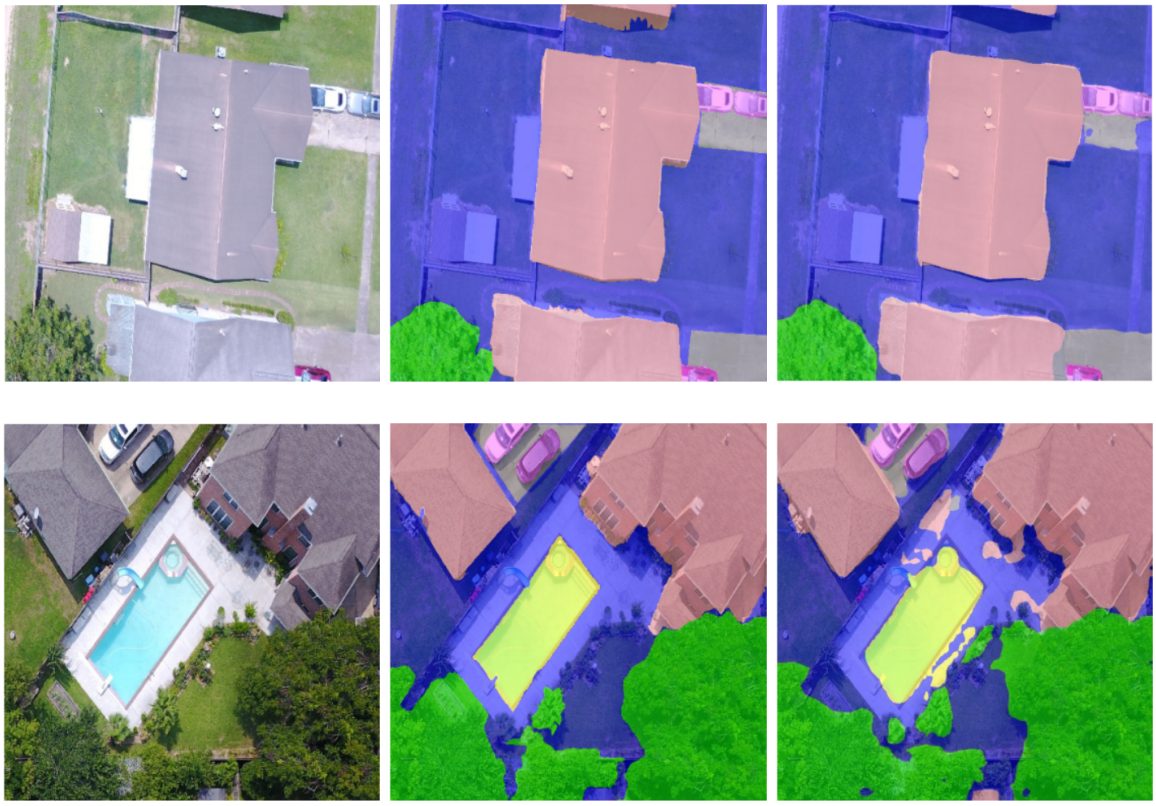

Ilustración 6.40: Datos, máscaras, inferencias tras de Data Augmentation del Floodnet

Se puede notar la gran diferencia en comparación con el dataset sin la aplicación de data augmentation, donde el modelo actual se puede inferir con una precisión bastante alta en relación con el modelo anterior. Además, el modelo puede detectar las clases como “Vehicle” y “Pool”, que tenían una cantidad de datos bastante menor.

A la ilustración 6.41 se muestra el mapa de calor del dataset. Se puede observar que el modelo actual puede realizar inferencias más precisas en comparación con el modelo sin aplicar el data augmentation.

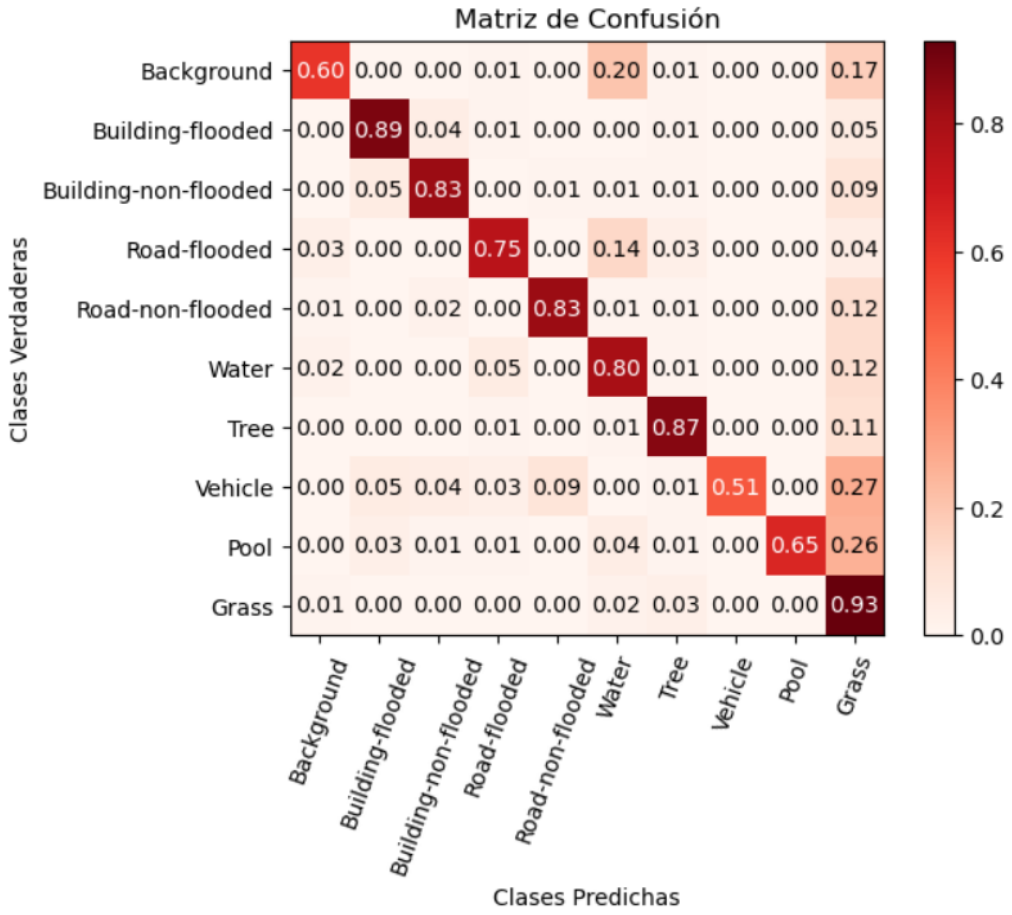

Ilustración 6.41: Mapa del calor en FloodNet tras del uso de Data Augmentation

A continuación se muestra las métricas del modelo en el testeo:

Clases	IOU(%)	DICE(%)	PRECISION(%)	RECALL(%)
Background	46.42	63.40	67.59	59.71
Building-flooded	73.82	84.94	81.04	89.23
Building-non-flooded	73.08	84.45	86.10	82.85
Road-flooded	54.82	70.81	67.04	75.04
Road-non-flooded	78.22	87.78	92.84	83.25
Water	66.77	80.07	80.46	79.69
Tree	79.17	88.38	89.72	87.07
Vehicle	42.38	59.54	72.27	50.62
Pool	58.12	73.51	85.18	64.66
Grass	85.72	92.31	91.27	93.38
Total	79.61	88.32	88.37	88.37

Cuadro 6.15: Métricas del testeo tras del uso de Data Augmentation en FloodNet

Al cuadro 6.16 se muestra una comparativa de las métricas obtenidas en entrenamiento con

y sin data augmentation, manteniendo los mismos hiperparámetros. Con este sentido, se realizará un análisis sobre los resultados obtenidos.

Clases	IOU	DICE	PRECISION	RECALL
Background	+5,47	+5,09	+1.61	+7.47
Building-flooded	+15.12	+10.97	+8.44	+13.84
Building-non-flooded	+5,61	+3.87	+10.77	-3.76
Road-flooded	+2.98	+2.53	-1.31	+6.82
Road-non-flooded	+0.8	+0.51	+1.82	-0.57
Water	-9.15	-6.24	-5.99	-6.49
Tree	-0.67	-0.41	-1.29	+0.4
Vehicle	+7.89	+8.25	+13.63	+5.04
Pool	+20.06	+18.37	+10.38	+21
Grass	-1.89	-1.08	-1.02	-1.14
Total	-1.62	-0.93	-0.91	-0.97

Cuadro 6.16: Métricas con/sin Data Augmentation en FloodNet

Mediante el cuadro 6.16 anterior, donde se observa el color verde como una mejora debido al data augmentation y el rojo como un empeoramiento al usar data augmentation, se puede apreciar que la mayor parte de las clases muestra una mejora significativa, excepto la clase "Water", que experimenta un deterioro del IOU aproximadamente un 10 % al usar data augmentation.

Los factores detrás de dicho fenómeno podrían ser un uso inadecuado de los métodos de data augmentation. En dicho experimento, los métodos más utilizados de data augmentation son la rotación y el volteo. Es posible que al aplicar estos métodos se generen ruidos no deseados en los datos de esta clase, lo que dificulta que la red neuronal aprenda las características importantes de dicha clase o incluso pueda afectar la integridad de los datos de la clase deteriorada.

En las clases minoritarias como "Pool" y "Vehicle", ambas se han beneficiado de aplicar el data augmentation, especialmente la clase "Pool", que ha experimentado una mejora de las métricas aproximadamente un 20 % en IOU, DICE y RECALL. Esto señala un funcionamiento efectivo del data augmentation aplicado a las clases minoritarias. Por otro lado, la clase "Vehicle" también se ha beneficiado del data augmentation, con una mejora aproximada del 8 %.

En las métricas globales, se observa un ligero empeoramiento al aplicar data augmentation, principalmente debido a la clase "Water". Sin embargo, las demás clases muestran un beneficio significativo. A excepción de las clases como "Grass" y "Tree", que experimentan un pequeño deterioro, las métricas de las otras clases muestran una mejora al aplicar data augmentation.

A la ilustración 6.42 se puede observar una comparación más precisa entre el modelo con y sin data augmentation.

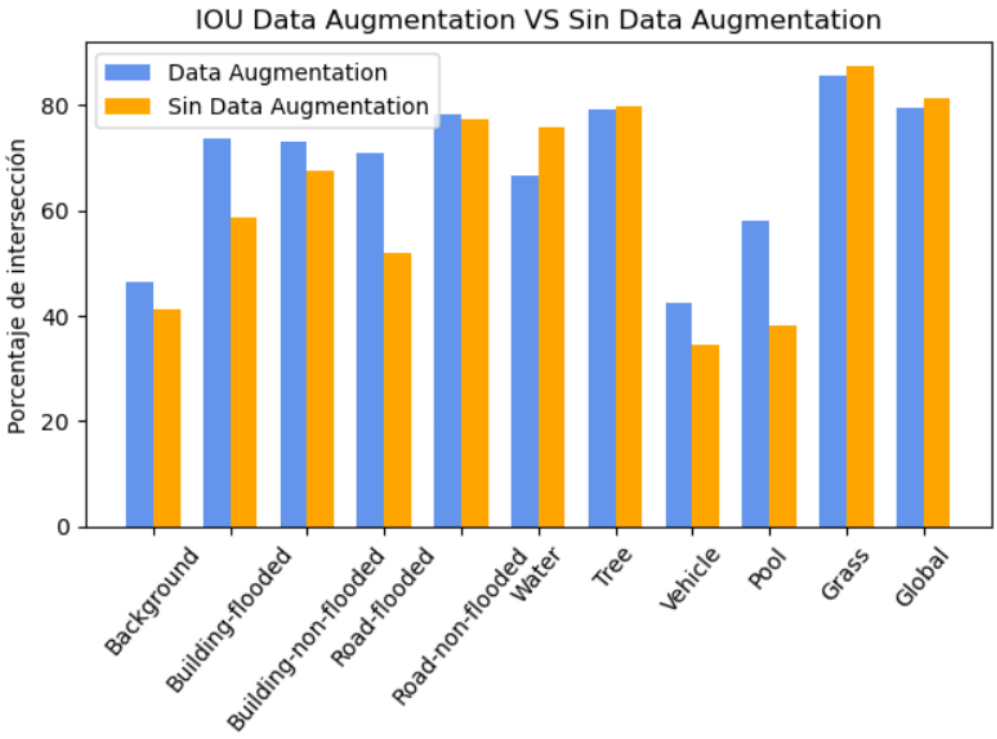

Ilustración 6.42: IOU con/sin Data Augmentation en FloodNet

Mediante dicha comparación se puede observar que casi todas las clases experimentan una mejora. Sin embargo, debido a la propiedad de las métricas ponderadas, los datos de cada clase contribuyen de manera diferente en las métricas globales. Por ejemplo, la clase "Grass" tiene mayor cantidad de datos que otras, por lo tanto, cualquier mejora o empeoramiento puede afectar mucho a las métricas globales que otras clases. En el caso de la clase "Pool", como tiene una cantidad de datos menor, aunque aumente casi un 20 % en las métricas, su contribución a las métricas globales es menor que la clase "Grass".

La causa del empeoramiento de las clases mayoritarias puede ser la introducción de ruido en la red neuronal. El ruido en los datos puede dar la confusión a la red neuronal y degradar los resultados del entrenamiento.

6.5.2. DeepGlobe con Data Augmentation

A continuación se aplica data augmentation al dataset DeepGlobe. En la ilustración 6.43 se presenta el resultado de aplicar data augmentation, similar al apartado anterior.

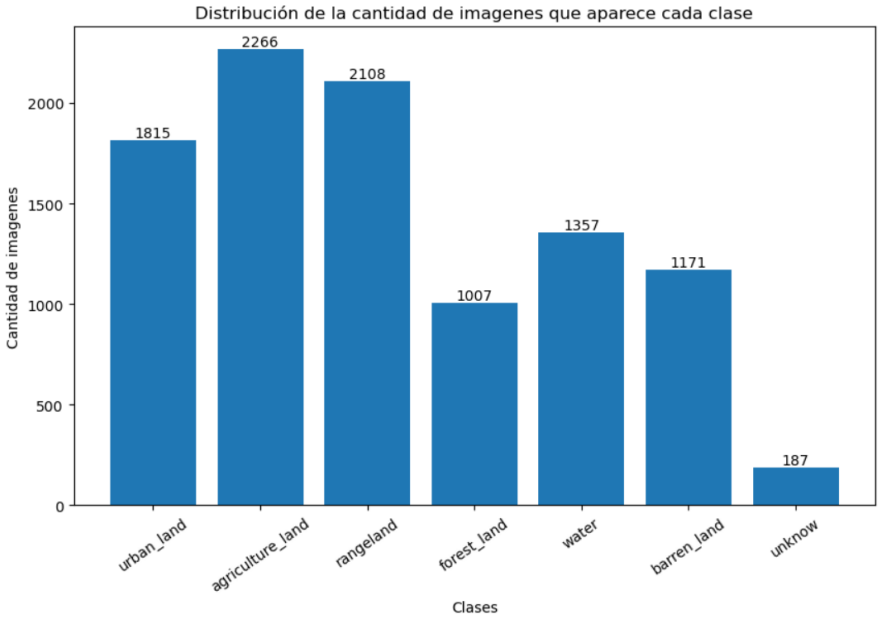

Ilustración 6.43: Cantidad de datos aumentados de cada clase en DeepGlobe

En la cuadro 6.17 se muestra la cantidad aumentada de cada clase. En dicho dataset, se centra en hacer la data augmentation sobre la clase "Range Land", que es la clase que tiene métricas peores que las otras clases.

CLASES	RESULTADOS
URBAN LAND	1162 ↑
AGRICULTURE LAND	1544 ↑
RANGE LAND	1586 ↑
FOREST LAND	813 ↑
WATER	879 ↑
BARREN LAND	743 ↑

Cuadro 6.17: Cantidad de imágenes aumentadas en FloodNet

A la hora del entrenamiento, se ha entrenado un total de 69 épocas, y se ha seleccionado el modelo entrenado en la época 37 como el mejor modelo, debido a su bajo loss de validación de 0.2344. En la ilustración 6.44 se puede observar las inferencias del modelo después de aplicar data augmentation:

Ilustración 6.44: Datos, máscaras, inferencias tras de Data Augmentation del DeepGlobe

Mediante la visualización de las máscaras y las inferencias correspondientes, se puede observar que el modelo ha mejorado mucho gracias a la aplicación de data augmentation. Aunque el modelo todavía no puede hacer una predicción muy precisa sobre los bordes y detalles, pero ya es capaz de capturar la mayor parte de las clases de los datos del testeo.

En el siguientes mapa de calor(Ilustración 6.45) se puede observar el correcto funcionamiento del data augmentation sobre la clase “Rangeland”, donde el modelo ha mejorado su capacidad de inferencia sobre esta clase. Sin embargo, también se observa que el modelo se confunde ocasionalmente con la clase “Agriculture Land”. Esto es una señal de que las características capturadas por el modelo sobre estas dos clases son bastante similares.Por lo tanto, en el siguiente paso del desarrollo, se intenta fusionar estas dos clases en una sola debido a sus características similares.

Y también otras clases como “Urban Land”, “Forest Land”, “Water” y “Barren Land” se benefician del data augmentation, ya que se ha mejorado la precisión del modelo sobre estas clases. Sin embargo, la clase “Agriculture Land” ha experimentado un ligero deterioro.

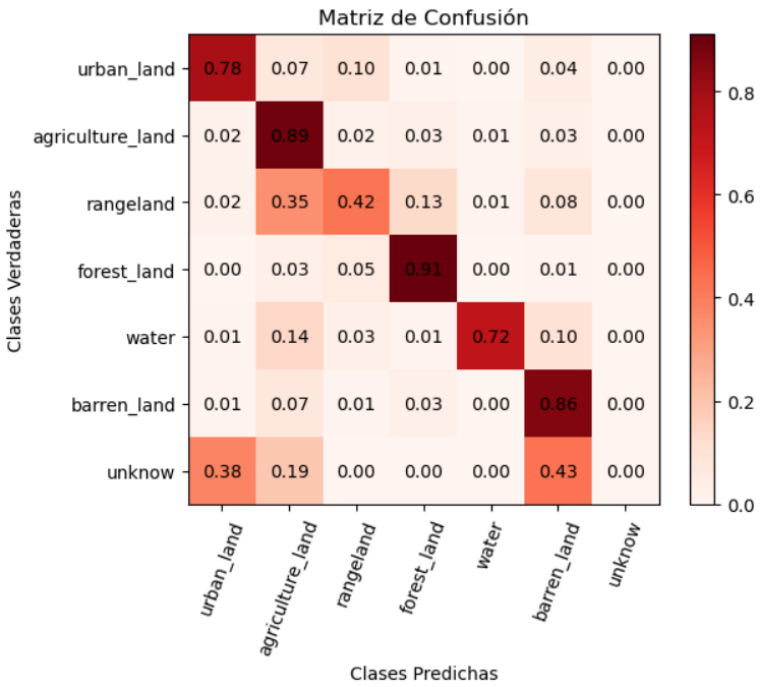

Ilustración 6.45: Mapa del calor en DeepGlobe tras del uso de Data Augmentation

Clases	IOU(%)	DICE(%)	PRECISION(%)	RECALL(%)
Urban land	69.35	81.90	86.33	77.90
Agriculture land	81.60	89.87	90.73	89.02
Rangeland	30.87	47.18	54.18	41.78
Forest land	70.26	82.53	75.25	91.37
Water	58.51	73.83	75.83	71.93
Barren land	62.47	76.90	69.41	86.21
Total	72.12	82.73	82.97	83.03

Cuadro 6.18: Métricas del testeo tras del uso de Data Augmentation en DeepGlobe

A la siguiente tabla(Cuadro 6.19) se muestra una comparativa de las métricas obtenidas en entrenamiento con y sin data augmentation, manteniendo los mismos hiperparámetros. Con este sentido, se realizará un análisis sobre los resultados obtenidos.

Clases	IOU(%)	DICE(%)	PRECISION(%)	RECALL(%)
Urban land	+16.09	+12.4	-2.09	+20.65
Agriculture land	+4.46	+2.78	+11.08	-7.05
Rangeland	+27.59	+40.82	+18.53	+38.29
Forest land	+2.92	+2.05	-3.92	+9.52
Water	+8	+6.71	-7.11	+15.56
Barren land	+13.86	+11.48	+6.83	+17.67
Total	+8.48	+8.02	+7.73	+4.63

Cuadro 6.19: Métricas con/sin Data Augmentation en DeepGlobe

Mediante dicha tabla de comparación entre el modelo con y sin data augmentation, se puede observar que todas las clases se benefician del data augmentation, especialmente la clase "Rangeland", que muestra una mejora en IOU de casi un 30 % y en DICE y RECALL de aproximadamente un 40 %. Además, otras clases también se ven beneficiadas por aplicar data augmentation, mostrando pequeñas mejoras.

En las métricas globales, el rendimiento del modelo ha aumentado aproximadamente un 8 % en las métricas como IOU, DICE y PRECISION, lo cual es una señal bastante buena sobre el éxito que aporta la data augmentation.

A la ilustración 6.46 se puede observar una comparación más precisa entre el modelo con y sin data augmentation.

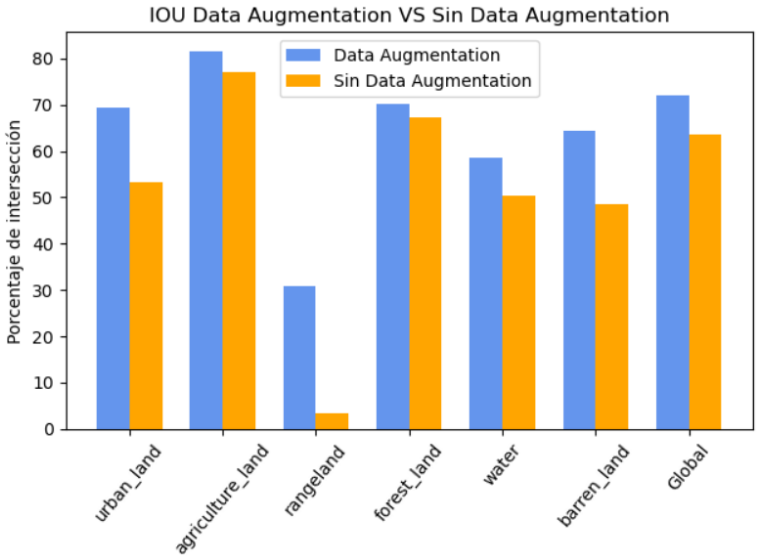

Ilustración 6.46: IOU con/sin Data Augmentation en DeepGlobe

Mediante la ilustración de comparación anterior, se puede observar con facilidad la mejora sobre la capacidad del modelo para detectar la clase "Rangeland", que es la clase que se aumenta principalmente, y todas las otras clases también se benefician del data augmentation.

6.5.3. Semantic Drone con Data Augmentation

Para el último dataset al que se aplica data augmentation, se sigue el mismo procedimiento de los datasets anteriores.

En la ilustración 6.47 se presenta el resultado de aplicar data augmentation.

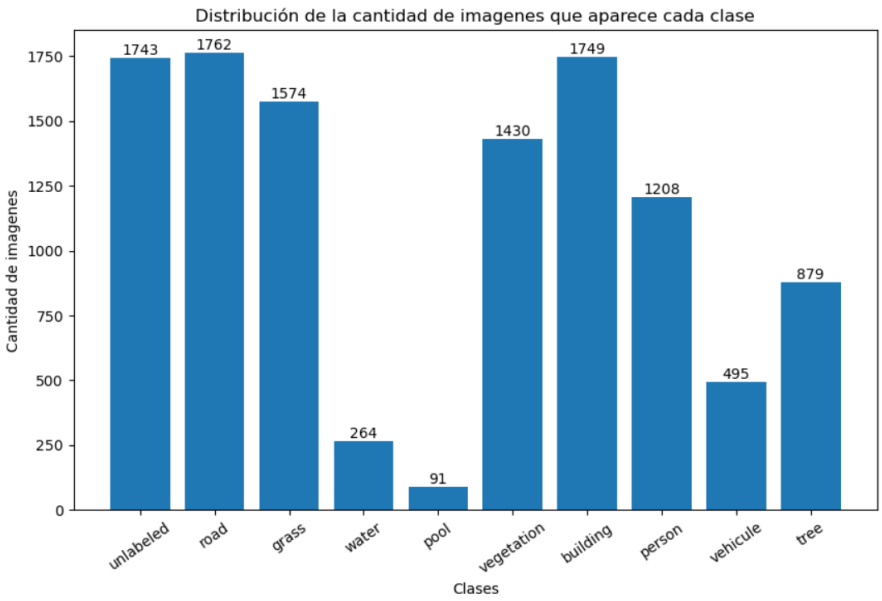

Ilustración 6.47: Cantidad de datos aumentados de cada clase en Semantic Drone

En la siguiente tabla(Cuadro 6.20) se muestra la cantidad aumentadas de cada clases:

CLASES	RESULTADOS
UNLABALED	1466 ↑
ROAD	1495 ↑
GRASS	1320 ↑
WATER	184 ↑
POOL	63 ↑
VEGETATION	1175 ↑
BUILDING	1473 ↑
PERSON	958 ↑
VEHICLE	368 ↑
TREE	701 ↑

Cuadro 6.20: Cantidad de imágenes aumentadas en Semantic Drone

Se ha entrenado un total de 89 épocas, y se ha seleccionado el modelo entrenado en la época 37 como el mejor modelo, debido a su bajo loss de validación de 0.0873. En las siguientes imágenes se pueden observar las inferencias del modelo después de aplicar data augmentation:

Con la visualización de las siguientes inferencias y máscaras, se puede observar que la red tiene una alta capacidad de segmentar las clases objetivas. En las primera inferencia y máscara, la red casi segmenta perfectamente, excepto por unos ruidos que se encuentran al lado de la clase árbol. Además, la línea divisoria entre “Road” y “Grass” está bastante clara. Donde se indica el modelo entrenado tiene muy claro las características cruciales entre estas 2 clases.

En la inferencia hecha por el modelo en la segunda imagen, se puede observar que el modelo confunde un poco sobre la clase “Vegetation” con la clase “Tree”, porque ambas clases son

de vegetación y comparten muchas características relacionadas, como el color, la magnitud, la forma, etc. Esto representa un desafío para el modelo entrenado a la hora de hacer la inferencia futura.

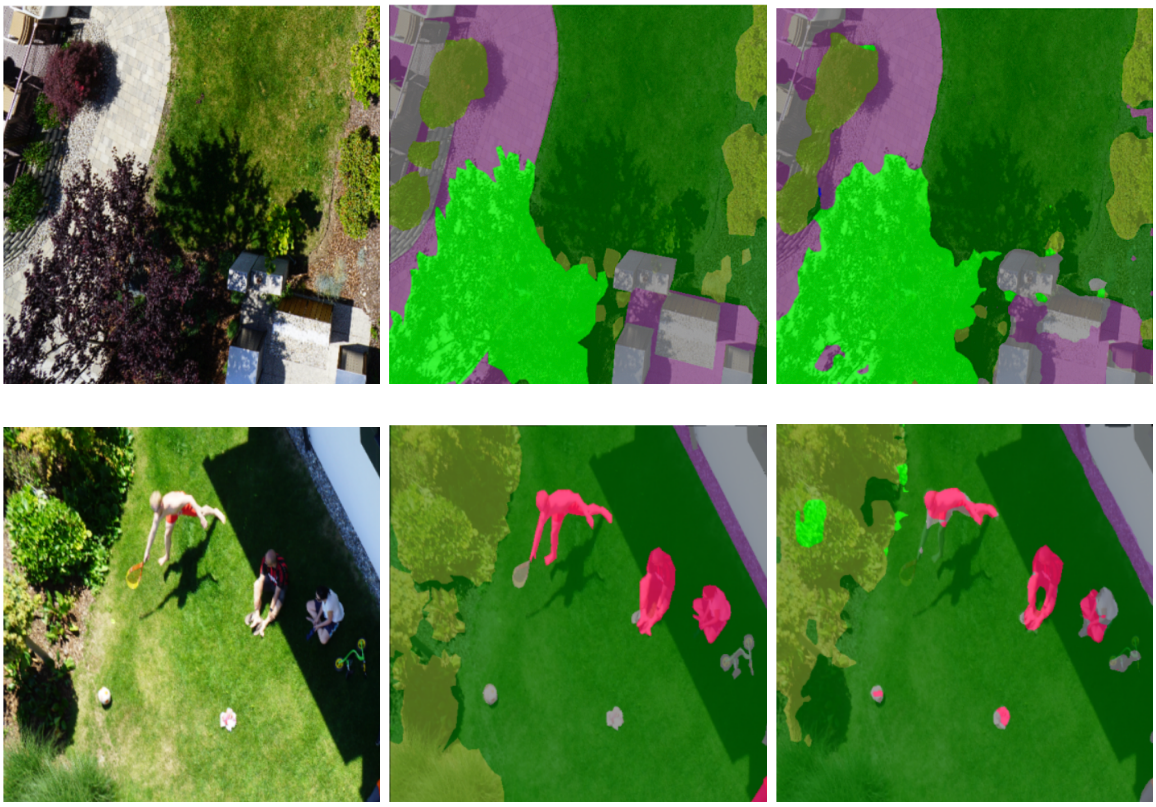

Ilustración 6.48: Datos, máscaras, inferencias tras de Data Augmentation del Semantic Drone

En el siguientes mapa de calor(Ilustración 6.49) se puede observar el correcto funcionamiento del data augmentation sobre las clases “Person”, “Vehicle” y “Tree” donde el modelo ha mejorado su capacidad de inferencia sobre estas clases. Y mediante el mapa de calor, se puede observar que la línea diagonal dentro del mapa se aproxima bastante al valor 1. Este es un buen señal de que el modelo puede diferenciar clase por clase. Aunque en casos ocasionales se confunde una clase con otra, como “Building” y “Road”, pero la probabilidad es bastante baja. Por otro lado, también indica que estas dos clases tienen características comunes a un nivel bajo.

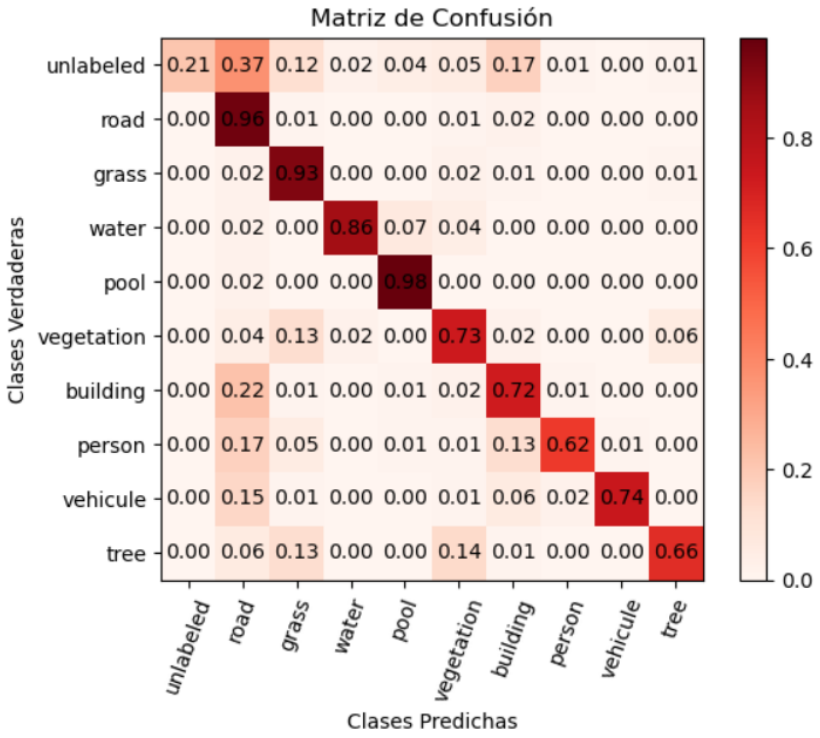

Ilustración 6.49: Mapa del calor en Semantic Drone tras del uso de Data Augmentation

Clases	IOU(%)	DICE(%)	PRECISION(%)	RECALL(%)
UNLABALED	18.52	31.25	80.91	19.37
ROAD	87.21	93.17	90.73	95.74
GRASS	83.10	90.77	88.64	93.01
WATER	72.52	84.07	82.44	85.77
POOL	64.46	78.39	64.65	99.56
VEGETATION	63.69	77.82	82.04	74.01
BUILDING	65.85	79.41	87.81	72.47
PERSON	45.72	62.75	63.54	61.99
VEHICLE	60.22	75.17	75.96	74.40
TREE	51.07	67.61	68.83	66.43
Total	78.71	87.55	87.77	87.88

Cuadro 6.21: Métricas del testeo tras del uso de Data Augmentation en Semantic Drone

Clases	IOU(%)	DICE(%)	PRECISION(%)	RECALL(%)
UNLABALED	+9.81	+15.22	+28.2	+9.92
ROAD	+1.85	+1.07	+0.09	+2.13
GRASS	+6.17	+3.81	+5.14	+2.29
WATER	+1.39	+0.94	-3.01	+4.83
POOL	-9.21	-6.45	-22.44	+16.85
VEGETATION	+2.46	+1.86	+5.98	-1.84
BUILDING	+0.42	+0.31	+4.54	-2.86
PERSON	+15.89	+16.8	+6.78	+23.39
VEHICLE	+2.77	+2.19	-1.63	+5.52
TREE	+8.33	+7.72	-5.67	+16.36
Total	+2.86	+1.96	+2.14	+1.85

Cuadro 6.22: Métricas con/sin Data Augmentation en Semantic Drone

Mediante las tabla anteriores(Cuadro 6.21, Cuadro 6.22), se puede observar que la mayoría de las clases se benefician del data augmentation, especialmente las clases "Person" y "Tree", donde la métrica IOU de ambas clases aumenta en un 15 % y un 8 % respectivamente.

Sin embargo, hay una clase que sufre un deterioro, que es la clase "Pool", la cual empeora aproximadamente un 10 % en la métrica IOU. Se podría pensar que una de las razones podría ser la escasez de datos de dicha clase. Aunque se ha aumentado la cantidad de datos de dicha clase, pero sigue siendo insuficiente en comparación con los aumentos de otras clases. Debido a esto, la red neuronal intenta memorizar por fuerza bruta dichas clases, lo que resulta en un overfitting. Esto se refleja en la métrica de PRECISION, donde ha disminuido casi un 20 %.

Aunque hay un deterioro de la clase especificada durante el entrenamiento, las métricas globales se benefician de la aplicación de data augmentation. Sin embargo, el beneficio no es tan significativo como en el caso del dataset DeepGlobe.

La siguiente comparación(Ilustración 6.50) ofrece una perspectiva más clara y global sobre el uso de la data augmentation.

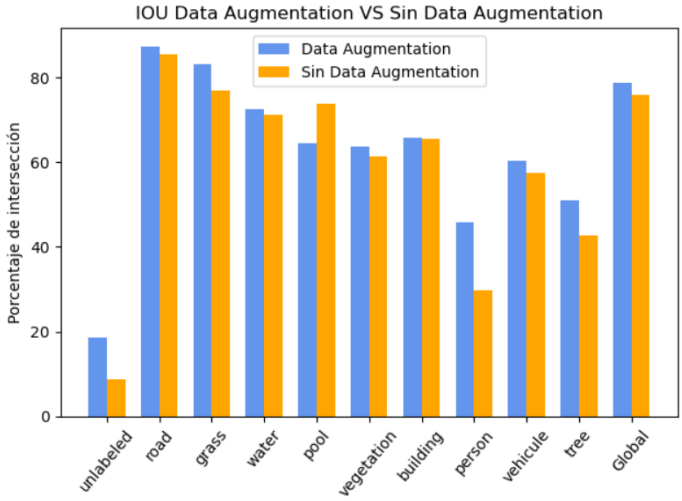

Ilustración 6.50: IOU con/sin Data Augmentation en Semantic Drone

Con dicha comparación se puede observar claramente que las clases que más se benefician son “Person” y “Tree”, mientras que todas las otras clases muestran un beneficio leve. Esto se puede presuponer por un data augmentation excesivo.

Como los datos originales del dataset solamente contienen un total de 400 imágenes, aunque primero se hace el corte en 4 segmentos más pequeños y se utilizan diferentes maneras de data augmentation sobre las clases minoritarias, el uso excesivo puede dar un resultado no deseado. Esto se debe a que la red podría empezar a reconocer los patrones repetidos, sabiendo ya los datos que se van a aparecer en el futuro.

6.5.4. Análisis de los experimentos

Mediante los experimentos anteriores se puede interpretar que la técnica de data augmentation es una herramienta eficaz para mejorar la capacidad de la red neuronal, proporcionando más datos alterados sobre las clases especificadas. Pero siempre hay que tener en cuenta un uso adecuado, ya que el data augmentation sobre el dataset FloodNet no es un ejemplo exitoso. Siempre es importante considerar la relación entre las clases y tratar de no introducir ruidos inesperados a la red neuronal.

En cada dataset, debido a la naturaleza de los datos aéreos obtenidos por dron o satélite, las clases relacionadas con la vegetación siempre presentan las mejores métricas, tanto en IOU, DICE, RECALL como PRECISION. Esto se debe a que en cada dataset, los datos de vegetación siempre tienen un alto porcentaje dentro del mismo y una alta frecuencia de aparición. Estos dos factores son cruciales para el entrenamiento de la red neuronal sobre dichas clases. Una alta frecuencia de aparición de datos ayuda a la red neuronal a identificar el patrón con regularidad, y una alta cantidad de datos brinda más oportunidades para ajustar la red sobre dicha clase.

6.6. Tarea 6 Unión de los datasets

En el siguiente paso del desarrollo, se fusionan los datasets en diferentes combinaciones con los siguientes motivos:

1. Para tener más clases relacionadas con la vegetación.
2. Para asegurar que las clases sean coherentes con el dataset final, lo que facilitará el Transfer Learning con el modelo entrenado usando el dataset fusionados.
3. Comparar qué combinación de la fusión de los datasets es mejor para el Transfer Learning. La combinación que incluye las imágenes satelitales y de drones, que proporcionan perspectivas globales y locales, o la combinación de los dos datasets de dron que se puede proporcionar las características más detalladas.

En los siguientes experimentos, se realizan diferentes combinaciones de los datasets y se analizan los resultados obtenidos. Se selecciona el modelo con el mejor resultado como base para la red que se va a entrenar con el último dataset, aprovechando las características capturadas en los datasets fusionados para obtener el mejor resultado posible.

6.6.1. Floodnet y DeepGlobe

La primera combinación que se realiza es la fusión del dataset FloodNet y DeepGlobe. Los datos del dataset FloodNet proporcionan un nivel de detalles alto, y en caso de los datos del dataset DeepGlobe, proporciona los datos con una cobertura global. Las clases finales del dataset y su relación correspondiente se pueden observar en la siguiente tabla(Cuadro 6.23):

Clases Finales	FloodNet	DeepGlobe	COLOR
Background	Background	Unknow	
Urban_Land	Building-flooded/non-flooded	Urban_Land	
Road	Road-flooded/non-flooded	-	
Water	Water	Water	
Forest_land	Tree	Fores_land	
Vehicle	Vehicle	-	
Pool	Pool	-	
Grass	Grass	-	
Agriculture_land	-	Agriculture_land/Range_land	
Barren_land	-	Barren_land	

Cuadro 6.23: Clases fusionadas entre FloodNet y DeepGlobe

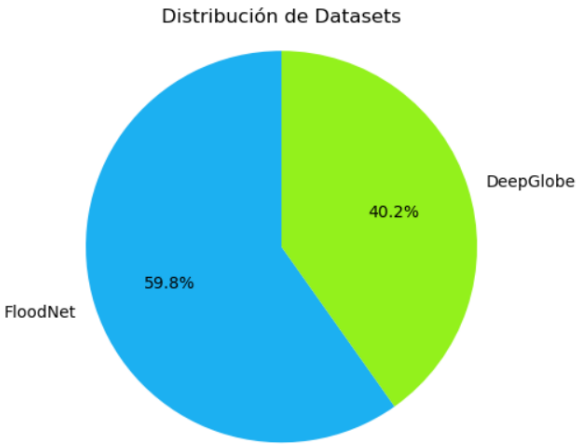

Ilustración 6.51: Porcentaje de datos entre FloodNet y DeepGlobe

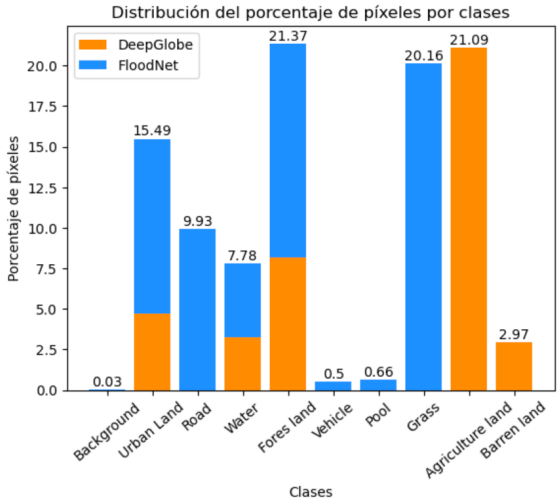

Ilustración 6.52: Porcentaje de datos entre FloodNet y DeepGlobe en clases fusionadas

Mediante las imágenes anteriores(Ilustración 6.51, Ilustración 6.52) se puede observar que los dos datasets tienen un porcentaje aproximado del 40 % y 60 %, respectivamente. En la distribución se puede notar que las clases propias del dataset FloodNet, como “Vehicle” y “Pool” tienen un porcentaje bastante bajo, lo cual puede influir en el entrenamiento de la red neuronal.

En cuanto a las clases de vegetación, se puede observar que las clases “Grass” y “Forest Land” tienen un alto porcentaje de los datos, lo cual puede ayudar a la red neuronal a obtener un mejor entrenamiento.

Las clases fusionadas dependen de sus características representadas en los datos RGB, como en el caso de las clases “Agriculture_land” y “Range_land”. En el experimento anterior, mediante el mapa de calor resultante, se puede observar las características semejantes entre estas dos clases; por lo tanto, se fusiona en una clase llamada “Agriculture_land”. Y las otras clases, como “Water”, “Forest_land”, etc., se fusionan por los datos comunes.

En la ilustración 6.53 se puede observar que se ha entrenado un total de 31 épocas, y con el modelo de la época 15 por tener un loss estable y relativamente bajo.

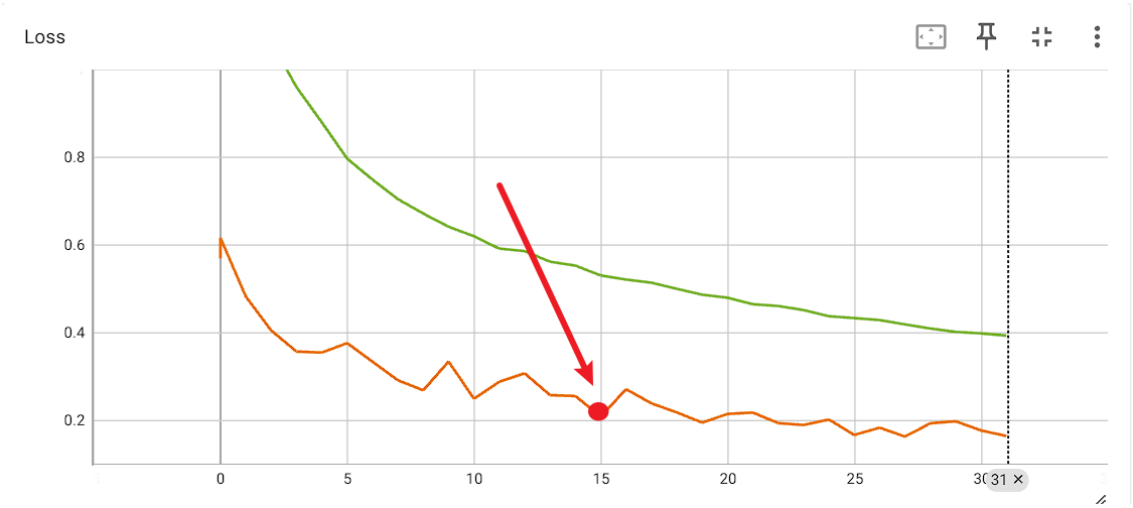

Ilustración 6.53: Evolución de Loss

A continuación, se muestran el mapa de calor(Ilustración 6.55) y las métricas de cada clase(Ilustración 6.25) correspondiente para el análisis.

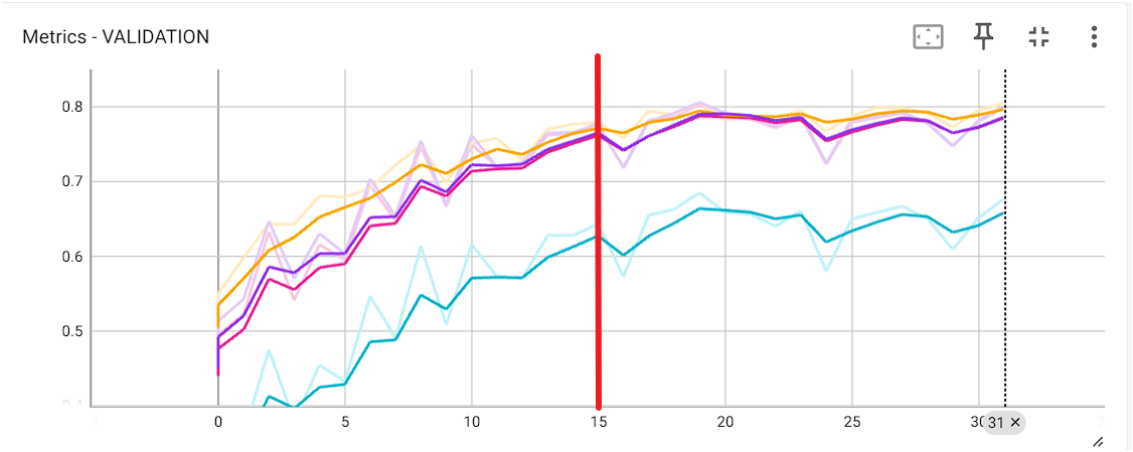

Ilustración 6.54: Evolución de Métricas

Métricas	Valor	COLOR
IOU	0.63	
DICE	0.76	
RECALL	0.76	
PRECISION	0.77	

Cuadro 6.24: Métricas en época 15

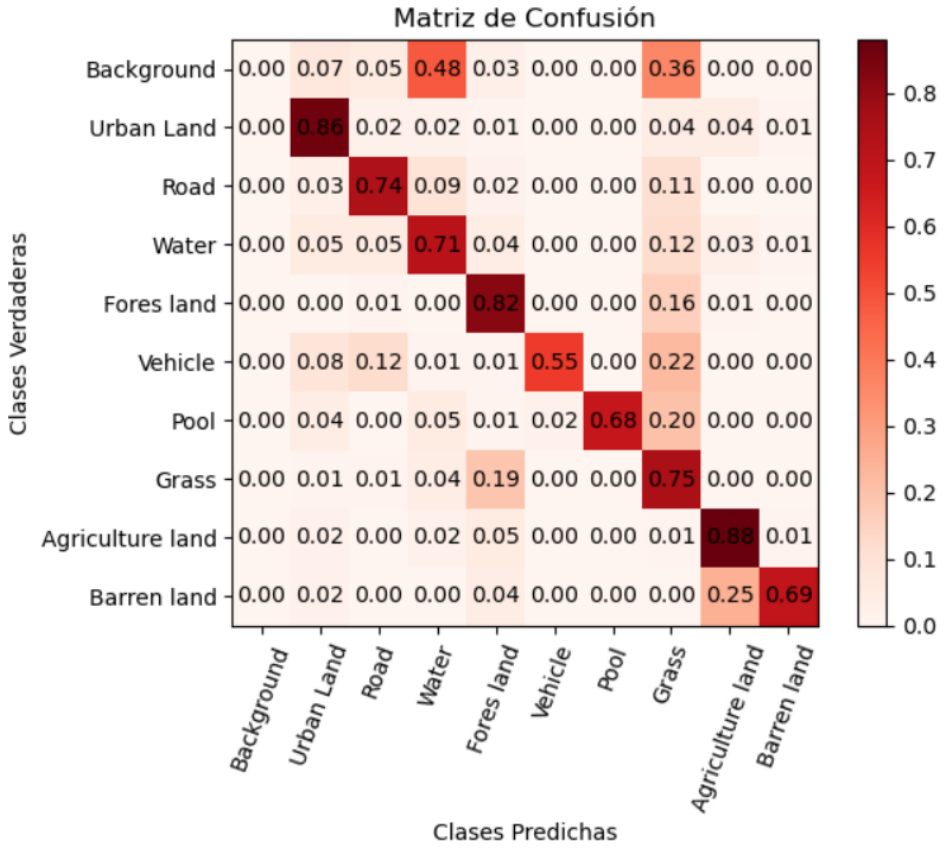

Ilustración 6.55: Mapa de calor en época 15

Clases	IOU(%)	DICE(%)	PRECISION(%)	RECALL(%)
Background	0	0	0	0
Urban_Land	69.15	81.76	77.97	85.93
Road	62.89	77.22	80.82	73.92
Water	51.75	68.21	65.55	71.09
Forest_land	51.06	67.60	57.47	82.06
Vehicle	37.40	54.44	53.64	55.26
Pool	45.48	62.52	57.93	67.90
Grass	66.61	79.96	86.20	74.56
Agriculture_land	80.03	88.91	89.32	88.50
Barren_land	47.58	64.48	60.57	68.93
Total	62.30	75.87	76.87	76.08

Cuadro 6.25: Métricas del testeo en época 15

Mediante el mapa de calor y la tabla de métricas, se puede observar que la clase “Background”, debido a su bajo porcentaje de datos, no se predice en absoluto. Y las clases minoritarias, como “Vehicle” y “Pool”, etc. Alcanzan un 40 % en IOU y entre 60 % y 70 % en otras métricas aproximadamente.

Y las clases mayoritarias, debido a su cantidad de datos, se puede entrenar con suficiente calidad y mejorar la predicción del modelo sobre dicha clase, y normalmente llega a un 65 % aproximadamente en IOU. Especialmente la clase "Agriculture Land", llega a un 80 % en IOU.

La ilustración 6.56 muestran los ejemplos sobre la capacidad de inferencia del modelo entrenado. Aunque las métricas de las clases minoritarias no son tan deseadas, pero en las inferencias también se muestra la capacidad de detección sobre la ubicación de los objetivos.

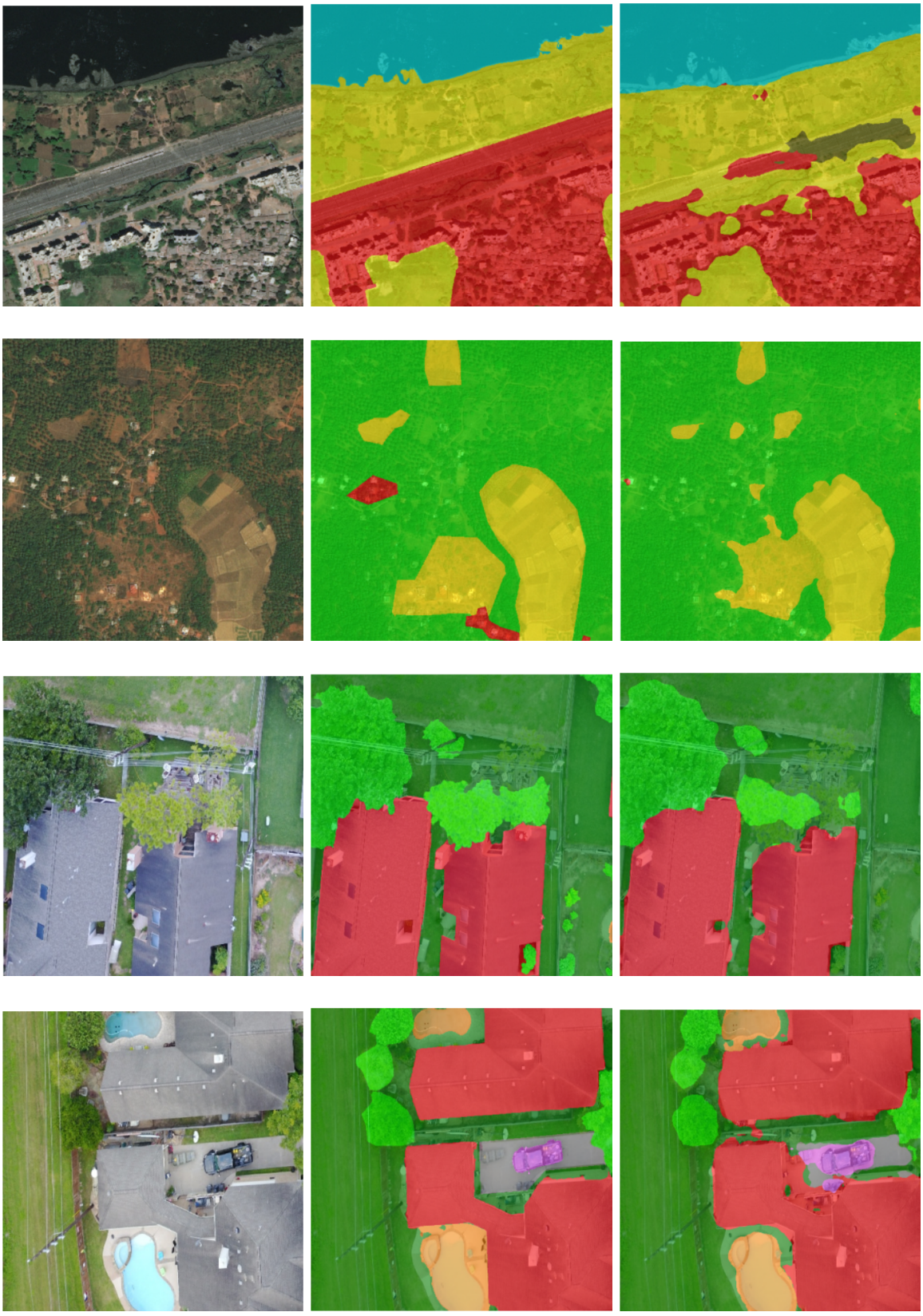

Ilustración 6.56: Datos, máscaras, inferencias tras de fusión sobre FloodNet y DeepGlobe

Mediante la visualización de las inferencias del modelo, se puede descubrir que, aunque la red tiene una métrica no deseada sobre las clases minoritarias como “Pool” y “Vehicle”, pero sigue poseyendo una capacidad de localización de la clase objetiva. Lo que le falta a la red sería la capacidad de procesar los detalles de dichas clases. Además, la clase “Water” tiene una métrica de IOU alrededor del 50 %, no es una cifra satisfecha, pero en las primeras imágenes se puede observar que la red tiene una gran capacidad para distinguir los datos de “Water” de otras clases.

Las clases de vegetación como “Forest Land” y “Grass” en los datos del dataset FloodNet tienen un comportamiento bastante bueno. La red puede inferir con un mayor grado de confianza sobre dichas clases. Especialmente sobre la clase “Grass”, la red puede realizar inferencias y procesamiento detallado. En el caso de la clase “Forest Land”, cuando la red realiza inferencias de esta clase en el dataset DeepGlobe, debido a la influencia de la clase mayoritaria “Agriculture Land”, a veces la red tiende hacia ella. En el caso del dataset FloodNet, la red tiene un comportamiento relativamente mejor y más preciso que en DeepGlobe.

6.6.2. FloodNet y Semantic Drone

La siguiente combinación de datasets que se va a probar es el dataset FloodNet y Semantic Drone. Los datos de ambos datasets son recopilados por drones, por lo tanto, los datos de ambos datasets comportan una similitud alta. Al investigar un poco más, se puede notar que los datos del dataset Semantic Drone son recopilados en un nivel de altura más baja que los datos del dataset FloodNet, lo cual será beneficioso para la red al capturar las características/patrones de las clases más detalladas.

A la siguiente tabla de clases(Cuadro 6.26) se sigue la misma política de selección de clases que en el apartado anterior, enfocándose en las características de las clases adyacentes como la principal política de fusión de datasets.

Clases Finales	FloodNet	Semantic_drone	COLOR
Background	Background	Background	
Building	Building-flooded/non-flooded	Building	
Road	Road-flooded/non-flooded	Road	
Water	Water	Water	
Tree	Tree	Tree	
Vehicle	Vehicle	Vehicle	
Pool	Pool	Pool	
Grass	Grass	Grass	
Vegetation	-	Vegetation	
Person	-	Person	

Cuadro 6.26: Clases fusionadas entre FloodNet y Semantic Drone

A la ilustración 6.57 se puede observar cómo son los porcentajes de cada dataset. Se puede

observar que el dataset FloodNet es el dataset principal de dicha unión, donde ocupa aproximadamente un 85 %, y en caso del dataset Semantic Drone ocupa aproximadamente un 15 %.

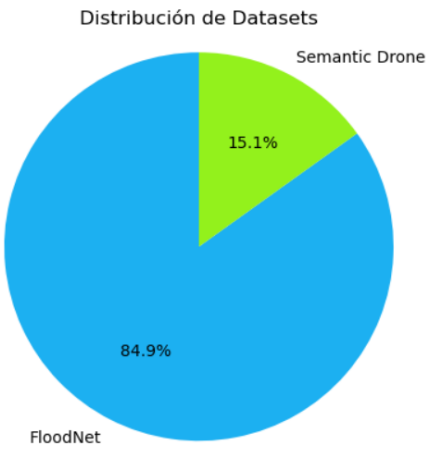

Ilustración 6.57: Porcentaje de datos entre FloodNet y Semantic Drone

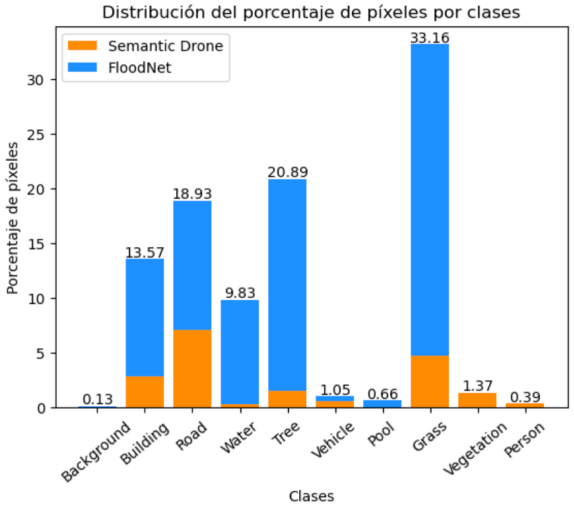

Ilustración 6.58: Porcentaje de datos entre FloodNet y Semantic Drone en clases fusionadas

Mediante dicha distribución(Ilustración 6.58) se puede observar la clase “Grass” sigue siendo la clase que tienen mayor cantidad de datos en el dataset Semantic Drone y en el dataset FloodNet. En dicha fusión, se puede observar las clases propias del dataset Semantic Drone son “Vegetation” y “Person”. Y ambas clases tienen un porcentaje de datos relativa bajo. Y los datos de otras clase del dataset Semantic Drone se puede beneficiar por la fusión de datos con el dataset FloodNet, se permite aumentar la diversidad de datos de las clases en común para obtener un mejor entrenamiento.

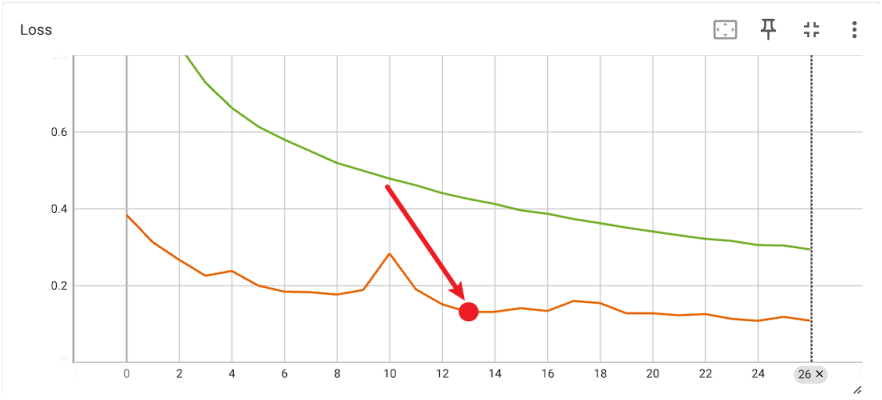

Ilustración 6.59: Evolución de Loss

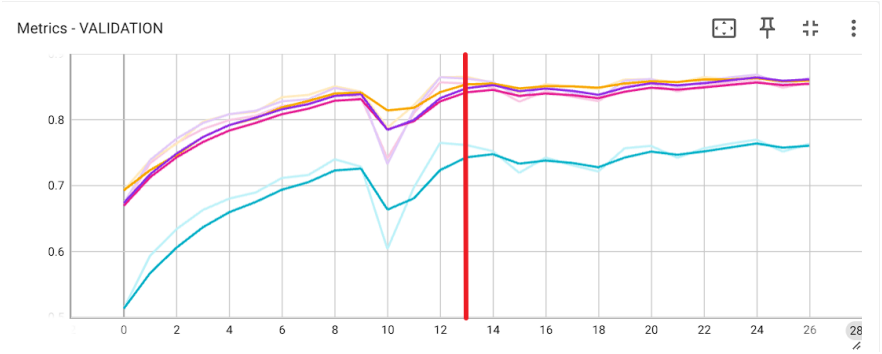

Ilustración 6.60: Evolución de Métricas

Se selecciona el modelo de la época 13 con un loss de validación de 0.1298 para el siguiente desarrollo:

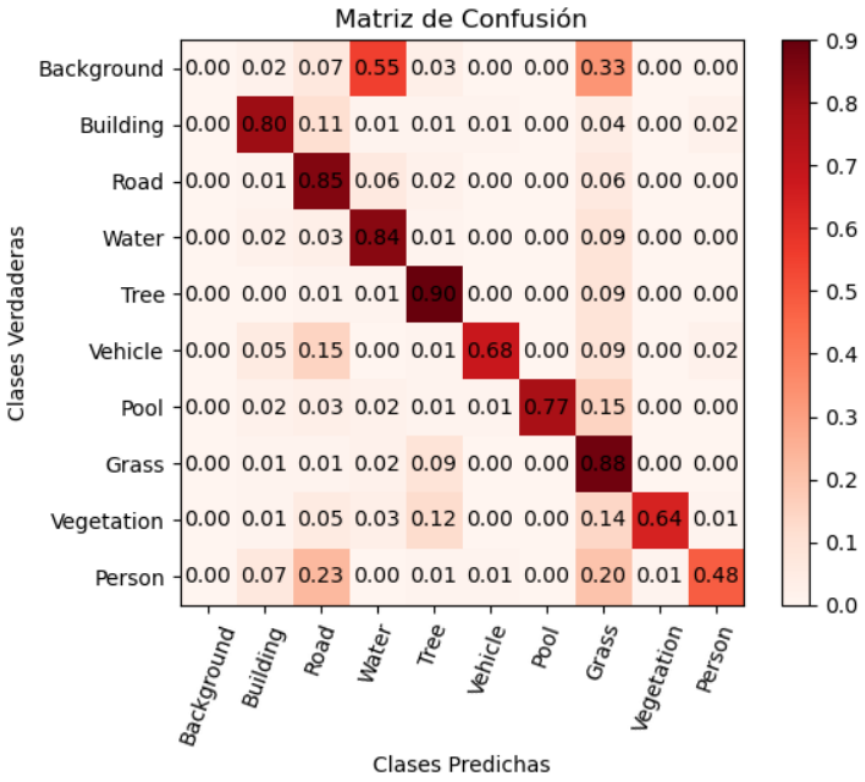

Ilustración 6.61: Mapa de calor en época 13

Mediante dicho mapa de calor(Ilustración 6.61) se puede observar claramente que la línea diagonal está bastante aproximada al 1, lo cual es una buena señal de que la red está diferenciando correctamente entre cada clase del dataset.

Sin embargo, en el caso de la clase “Person”, esta se ve influenciada por las clases mayoritarias como “Road” y “Vegetation”. Esto ocurre porque los datos de “Person” suelen estar

relacionados con los datos de “Road” y “Grass”, y cuando la red hace la inferencia sobre la clase “Person”, las clases mayoritarias pueden influir en la clase minoritaria. En el caso de la clase “Vegetation”, ocurre un fenómeno similar porque, al ser una clase relacionada con la vegetación, cuando la red hace la inferencia sobre ella, puede cometer errores con otras clases de vegetación como “Grass” y “Tree” por tener las características similares.

Clases	IOU(%)	DICE(%)	PRECISION(%)	RECALL(%)
Background	0.23	0.46	88.98	0.23
Building	70.52	82.71	86.18	79.52
Road	73.05	84.42	84.23	84.62
Water	65.30	79.01	74.26	84.42
Tree	68.71	81.45	74.68	89.58
Vehicle	43.94	61.06	55.70	67.56
Pool	68.25	81.13	86.22	76.60
Grass	81.33	89.70	91.77	87.73
Vegetation	53.86	70.01	76.87	64.28
Person	12.07	21.55	13.86	48.42
Total	74.05	84.18	85.67	84.83

Cuadro 6.27: Métricas del testeo en época 13

Con el mapa de calor(Ilustración 6.61) y la tabla de métricas(Cuadro 6.27) se puede observar que la clase “Person” tiene un peor comportamiento reflejado en las métricas. Sin embargo, se puede razonar fácilmente: en la distribución anterior, los datos de la clase “Person” tienen un porcentaje del 0.39 %, y es un desafío para la red aprender las características de dicha clase.

Por otro lado, la clase “Vehicle” tiene una métrica más baja que la clase “Vegetation”, aunque tiene más datos, una posible suposición sería que, con otras clases relacionadas con la vegetación como “Tree” y “Grass”, la red ya ha capturado las características básicas sobre la vegetación. Por lo tanto, como la clase “Vegetation” también es una clase relacionada con la vegetación, la red utiliza las características aprendidas de otras clases d vegetación y aprende fácilmente sobre dicha clase. En el caso de la clase “Vehicle”, como es una clase particular y no está relacionada con ninguna otra clase, la red tiene que aprender sus características desde cero con los pesos preestablecidos.

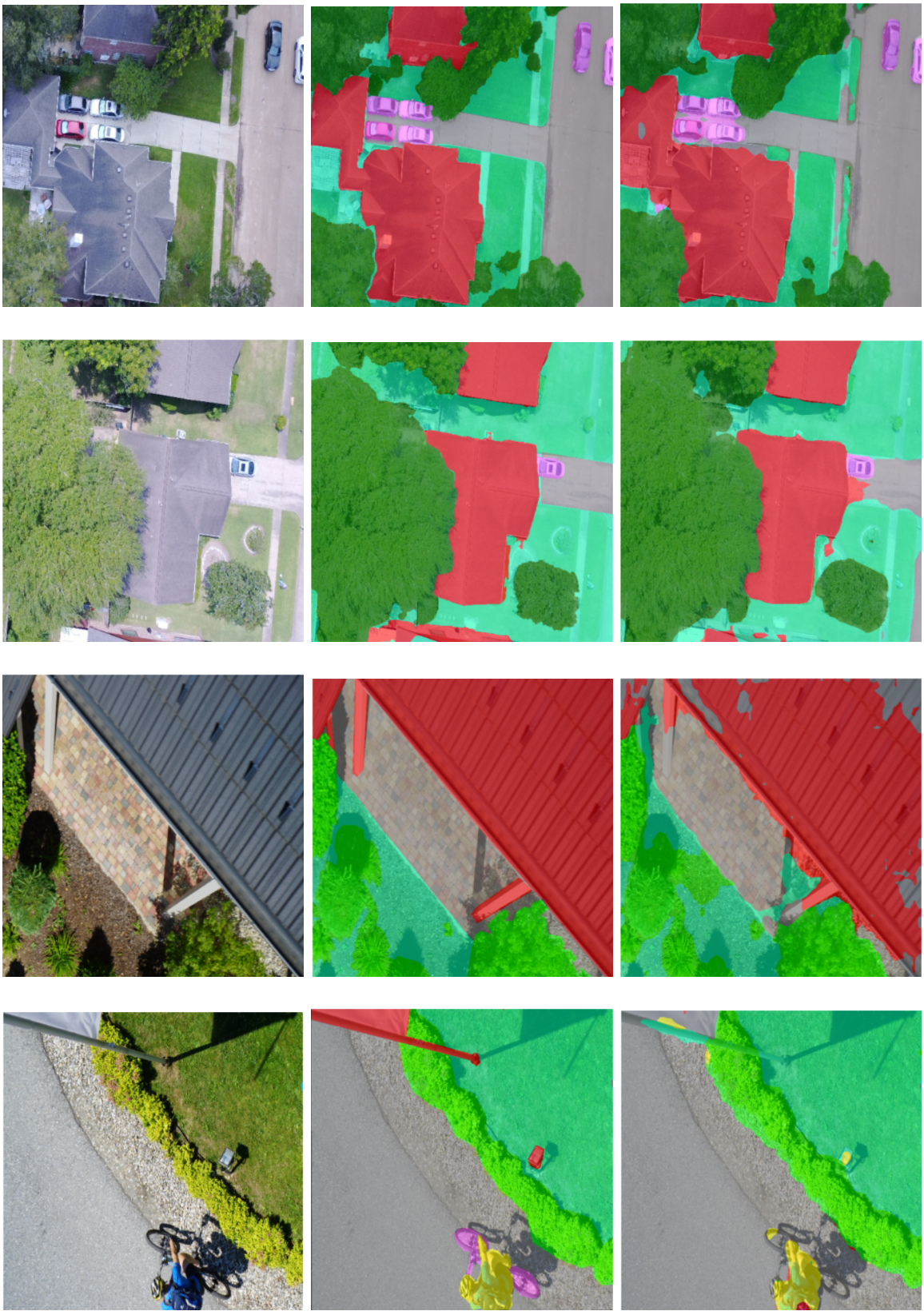

Ilustración 6.62: Datos, máscaras, inferencias tras de fusión sobre FloodNet y Semantic Drone

6.6.3. Análisis de los experimentos

Al final de hacer una comparación de los entrenamientos realizados con diferentes combinaciones de datasets, se puede observar que el entrenamiento de la combinación entre FloodNet y Semantic Drone presenta una métrica IOU superior a la combinación de FloodNet y DeepGlobe por un 10 %, lo cual es una cifra bastante significativa. Las posibles suposiciones para este resultado son las siguientes:

- Al entrenar una red con diferentes perspectivas, a veces no se logra un resultado deseado porque, durante la fase de entrenamiento, la introducción de diferentes perspectivas a veces actúa como ruido, lo que dificulta que la red capture y aprenda las características adecuadamente.
- Relación de diferentes clases dentro de las combinaciones de los datasets: En el caso de la combinación entre FloodNet y Semantic Drone, las clases de ambos datasets tienen una relación más alta, ya que los datos son recopilados por drones y dentro del entorno urbano. Por otro lado, en la combinación de FloodNet con DeepGlobe, los datos de FloodNet se recopilan mayormente en la ciudad, mientras que los datos de DeepGlobe, que tienen una resolución espacial más alta, incluyen una variedad de terrenos y circunstancias que no se limitan solo a entornos urbanos, sino también a terrenos agrícolas, montañas, etc.

6.7. Tarea 7 Realización de Transfer Learning

En la sección anterior se obtuvieron dos modelos diferentes entrenados con sus propios pesos. El siguiente y último paso para el desarrollo de dicho TFG consiste en el uso de los modelos preentrenados en la sección anterior para aplicar la técnica de Transfer Learning sobre el dataset Flair y probar la eficiencia de los pesos entrenados en dos diferentes combinaciones de datasets para determinar en qué combinación se puede obtener el mejor resultado: la combinación de FloodNet y DeepGlobe, que incluye diferentes perspectivas tanto a nivel global como a nivel local, y otra combinación de FloodNet y Semantic Drone, que se enfoca en los detalles de la perspectiva local.

La teoría sobre la técnica de Transfer Learning es muy simple. Su idea principal consiste en "transferir conocimientos de un dominio de origen a un dominio de destino. De este modo, la eficacia del modelo en el dominio de destino puede mejorar significativamente, sobre todo cuando el dominio de destino tiene datos limitados para el entrenamiento" [30].

En el ejemplo siguiente(Ilustración 6.63) se puede observar un ejemplo que se explica más claro sobre el idea de Transfer Learning:

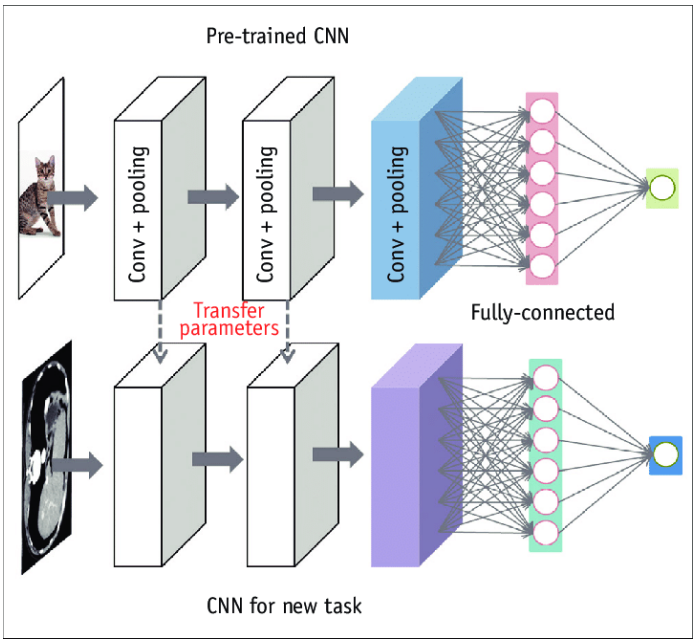

Ilustración 6.63: Transfer Learning

En el ejemplo anterior, se realiza un entrenamiento del primer modelo destinado a la predicción de la clase "Gato", donde dicha red ha aprendido todas las características necesarias para predecir la clase "Gato". En el caso del segundo modelo, su objetivo principal es la predicción de imágenes de tomografía computarizada (CT) en el ámbito de la medicina.

A primera vista, estas dos clases no tienen ninguna relación entre sí. Sin embargo, según la ilustración 2.14, las capas delanteras del modelo solo capturan características de bajo

nivel, como bordes, texturas, etc. Siguiendo esta idea, las dos clases pueden tener las mismas características a nivel bajo, aunque no se aprecie a primera vista. El segundo modelo utiliza las capas delanteras ya entrenadas por el modelo para la clase "Gato" para entrenar el dataset de imágenes CT con el fin de ahorrar coste computacional, recursos necesarios, etc. Lo más importante para un modelo destinado a predecir a qué clase pertenece una imagen son las capas traseras, donde se encuentran las características más abstractas. Por lo tanto, como las capas delanteras extraen las características de nivel bajo, se mantiene la igualdad entre los dos modelos, y solo se entrenan las últimas capas para el modelo de datos CT.

En dicho TFG está utilizando la red FCN con el backbone de Resnet50, que contiene 4 capas (layers) con varias unidades residuales. Por lo tanto, se "congelan" las 3 primeras capas y solo se entrena la última capa en los siguientes experimentos.

6.7.1. Análisis del resultado obtenido entre 2 combinaciones

El siguiente experimento consiste en el entrenamiento sobre la combinación de los datasets FloodNet y DeepGlobe, donde se entrena un total de 11 épocas. El loss mínimo se ha encontrado en la época 11 con un valor de 0.3611. Los valores de las diferentes métricas se pueden observar en el siguiente cuadro (Cuadro 6.28)

Clases	IOU(%)	DICE(%)	PRECISION(%)	RECALL(%)
Building	63.54	77.71	77.23	78.19
Pervious surface	40.25	57.40	51.36	65.04
Impervious surface	59.68	74.75	77.45	72.24
Bare soil	41.84	59.00	71.77	50.09
Water	67.29	80.45	80.38	80.51
Coniferous	18.17	30.76	36.87	26.39
Deciduous	56.00	71.79	72.22	71.37
Brushwood	27.86	43.58	41.04	46.46
Vineyard	74.53	85.41	81.39	89.85
Herbaceous vegetation	39.70	56.84	65.74	50.06
Agricultural Land	34.15	50.92	40.90	67.44
Plowed land	38.38	55.47	48.56	64.69
Swimming pool	43.33	60.47	75.48	50.43
Snow	0.00	0.00	0.00	0.00
Clear cut	0.00	0.00	0.00	0.00
Mixed	0.00	0.00	0.00	0.00
Ligneous	0.00	0.00	0.00	0.00
Greenhouse	10.16	18.44	17.51	19.48
Other	0.00	0.00	0.00	0.00
Total	48.30	63.93	65.41	63.83

Cuadro 6.28: Métricas del testeo en época 11

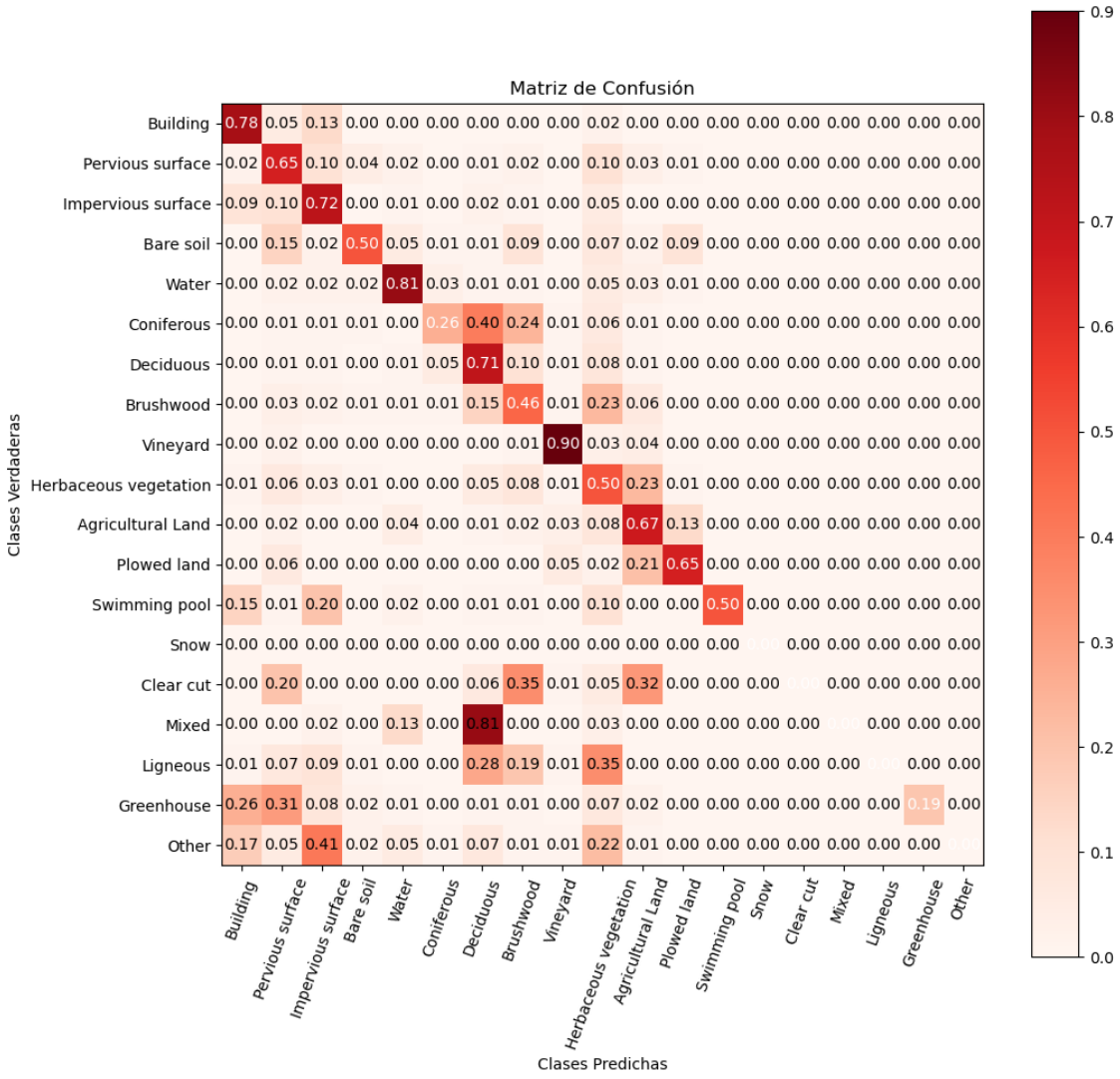

Ilustración 6.64: Mapa de calor para combinación FloodNet y DeepGlobe

y el siguiente cuadro representa las métricas sobre la combinación de los datasets FloodNet y Semantic Drone

Clases	IOU(%)	DICE(%)	PRECISION(%)	RECALL(%)
Building	64.99	78.78	84.19	74.02
Pervious surface	40.17	57.32	58.36	56.31
Impervious surface	60.45	75.35	76.97	73.80
Bare soil	46.08	63.09	69.17	57.98
Water	71.46	83.35	81.60	85.18
Coniferous	17.44	29.69	45.95	21.93
Deciduous	57.75	73.21	70.90	75.68
Brushwood	27.67	43.35	45.26	41.60
Vineyard	77.22	87.14	87.10	87.19
Herbaceous vegetation	43.94	61.05	63.37	59.90
Agricultural Land	38.75	55.85	45.13	73.25
Plowed land	40.23	57.38	51.55	64.70
Swimming pool	37.21	54.24	66.12	45.98
Snow	0.00	0.00	0.00	0.00
Clear cut	0.00	0.00	0.00	0.00
Mixed	0.00	0.00	0.00	0.00
Ligneous	0.00	0.00	0.00	0.00
Greenhouse	10.22	18.55	12.58	35.28
Other	0.00	0.00	0.00	0.00
Total	50.58	66.00	66.96	66.11

Cuadro 6.29: Métricas del testeo en época 9

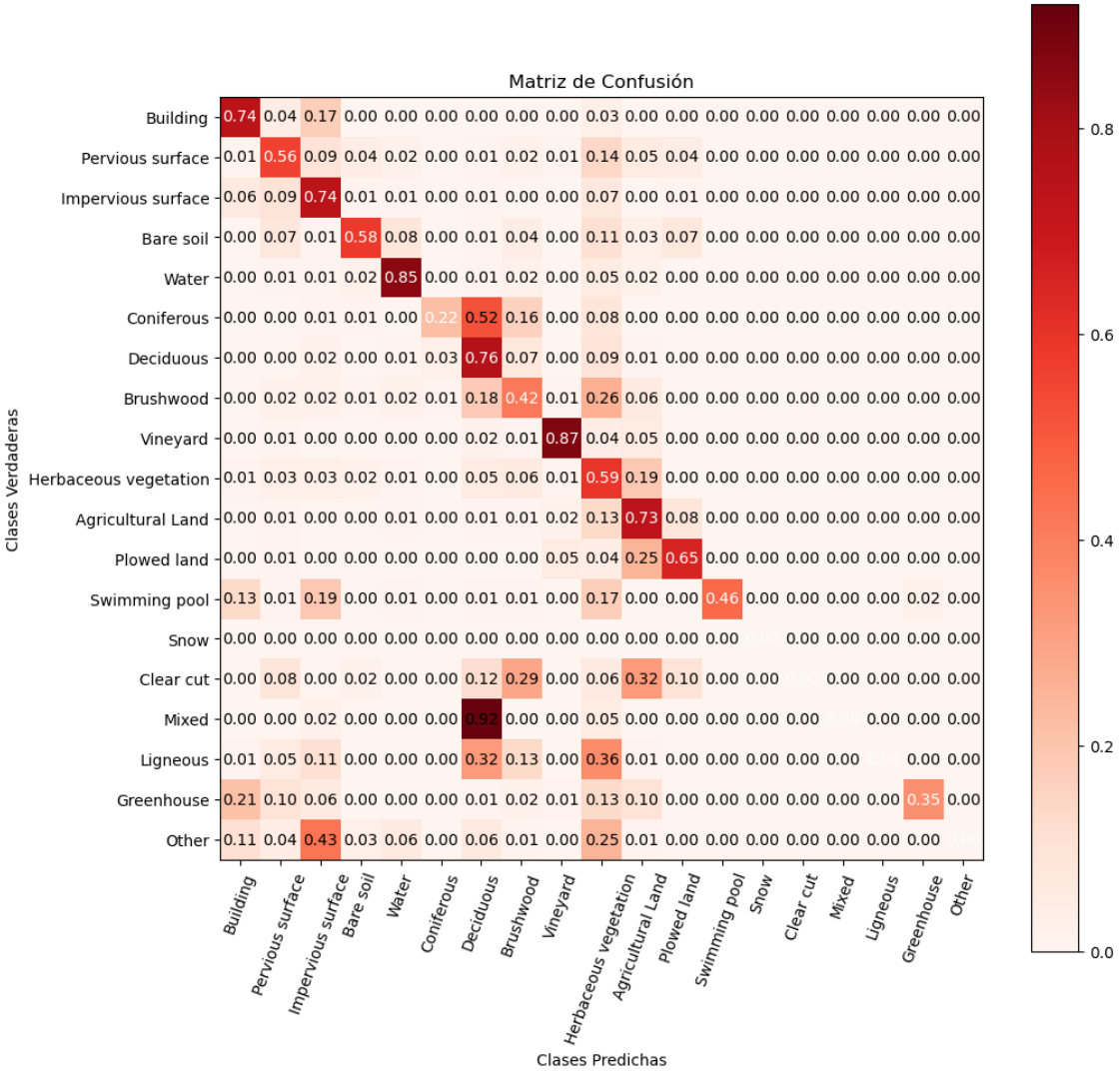

Ilustración 6.65: Mapa de calor para combinación FloodNet y Semantic Drone

En la siguiente ilustración (Ilustración 6.66) se representan las métricas de la combinación de FloodNet con DeepGlobe, que se identifica como “comb1”, y las métricas de la combinación de FloodNet con Semantic Drone, que se identifica como “comb2”. Se va a realizar un análisis exhaustivo sobre los experimentos de dichas combinaciones.

En el siguiente análisis, para simplificar, se llamará a la combinación de FloodNet con DeepGlobe “combinación 1”, y a la combinación de FloodNet con Semantic Drone “combinación 2”.

Clases	IOU(comb1)	IOU(comb2)	DICE(comb1)	DICE(comb2)	PRECISION(comb1)	PRECISION(comb2)	RECALL(comb1)	RECALL(comb2)
Building	63.54	64.99	77.71	78.78	77.23	84.19	78.19	74.02
Pervious surface	40.25	40.17	57.4	57.32	51.36	58.36	65.04	56.31
Impervious surface	59.68	60.45	74.75	75.35	77.45	76.97	72.24	73.8
Bare soil	41.84	46.08	59	63.09	71.77	69.17	50.09	57.98
Water	67.29	71.46	80.45	83.35	80.38	81.6	80.51	85.18
Coniferous	18.17	17.44	30.76	29.69	36.87	45.95	26.39	21.93
Deciduous	56	57.75	71.79	73.21	72.22	70.9	71.37	75.68
Brushwood	27.86	27.67	43.58	43.35	41.04	45.26	46.46	41.6
Vineyard	74.53	77.22	85.41	87.14	81.39	87.1	89.85	87.19
Herbaceous vegetation	39.7	43.94	56.84	61.05	65.74	63.37	50.06	59.9
Agricultural Land	34.15	38.75	50.92	55.85	40.9	45.13	67.44	73.25
Plowed land	38.38	40.23	55.47	57.38	48.56	51.55	64.69	64.7
Swimming pool	43.33	37.21	60.47	54.24	75.48	66.12	50.43	45.98
Snow	0	0	0	0	0	0	0	0
Clear cut	0	0	0	0	0	0	0	0
Mixed	0	0	0	0	0	0	0	0
Ligneous	0	0	0	0	0	0	0	0
Greenhouse	10.16	10.22	18.44	18.55	17.51	12.58	19.48	35.28
Other	0	0	0	0	0	0	0	0
Total	48.3	50.58	63.93	66	65.41	66.96	63.83	66.11

Ilustración 6.66: Comparación de métricas en diferentes combinaciones

Mediante la ilustración anterior se puede observar claramente que el modelo que ha usado los pesos preentrenados de la combinación 2 resulta en una mejor eficiencia que el modelo que ha usado los pesos de la combinación 1. Se comienza con el análisis clase a clase.

En la primera clase que se encuentra es la clase “Building”. Dicha clase ocupa un 8.14 % de los datos dentro del dataset. En la métrica del IOU, la combinación 1 tiene un valor de 63.54 %, y en el caso de la combinación 2, tiene un valor de 64.99 %. Ambas combinaciones representan un valor muy similar, con una diferencia de solo 1.54 %. En la métrica PRECISION, se puede notar una gran diferencia, donde la combinación 2 tiene un 6.96 %, casi un 7 % mayor que la combinación 1. Esto indica que el modelo entrenado con los pesos de la combinación 2 tiene un acierto del 85 % aproximadamente en las inferencias realizadas, mientras que el modelo con la combinación 1 tiene un acierto del 77 % aproximadamente. En la métrica RECALL, la situación es inversa, donde el modelo con la combinación 1 muestra una mejora de 4.17 %. Al final, en la métrica DICE, aunque en la métrica RECALL el modelo con la combinación 1 tiene una mejora leve respecto al modelo con la combinación 2, no es suficiente para compensar la debilidad en la métrica PRECISION. Por lo tanto, en la métrica DICE, la combinación 2 muestra una leve mejora respecto a la combinación 1.

La clase “Pervious surface” ocupa el 8.25 % de los datos dentro del dataset. Es muy similar a la clase anterior, pero no salió un resultado similar a la clase “Building”. En la métrica IOU, el modelo entrenado con los pesos de la combinación 1 tiene una mejora respecto al modelo entrenado con los pesos de la combinación 2, pero dicha mejora es muy pequeña, tan pequeña hasta que se puede omitir. En las métricas PRECISION y RECALL se puede notar una situación interesante, donde en la métrica de PRECISION, la combinación 2 tiene una mejora del 7 % que la combinación 1, pero en la métrica RECALL, la combinación 1 tiene una mejora del 8.73 % que la combinación 2.

La siguiente clase es “Impervious surface”, que ocupa un 14 % de los datos aproximadamente,

casi el doble que la clase "Pervious surface". En dicha clase, la métrica IOU de las 2 combinaciones es muy similar, pero la combinación 2 es mejor que la combinación 1 con un 0.77 %, y también sucede en otras métricas como DICE, PRECISION y RECALL. Todas estas métricas muestran una diferencia leve, similar a la métrica IOU.

Después de la clase "Impervious surface" está la clase "Bare Soil", que ocupa aproximadamente el 3.5 % de los datos en el dataset. Lo curioso es que, aunque tiene un porcentaje de datos inferior al de la clase "Pervious surface", pero presenta métricas mejores. En la métrica IOU, el modelo entrenado con la combinación 2 obtiene un mejor resultado, con una mejora del 4.24 %, lo cual es bastante notable. En otras métricas como PRECISION, aunque la combinación 1 obtiene un valor un poco mejor que la combinación 2, esto indica que las inferencias correctamente predichas como positivas por el modelo entrenado con la combinación 1 tienen una alta proporción sobre todos los predichos positivos. Sin embargo, en la métrica RECALL, la combinación 2 muestra una mejora del 7.89 % sobre la combinación 1, indicando una alta proporción de las inferencias correctamente predichas como positivas sobre todos los datos que realmente son positivos.

En la clase "Water", el modelo entrenado con la combinación 2 también obtiene mejores métricas tanto en IOU como en otras métricas como DICE, PRECISION y RECALL. Y dicha clase tiene una métrica bastante alta en comparación con otras clases debido a sus propias características. Aunque ambas combinaciones tienen la clase "Water", pero la segunda combinación presenta métricas mejores que la primera. Una posible causa de esta situación podría ser que en la combinación 2, la red ha extraído suficientes características más detalladas que en la combinación 1. Es importante recordar que la combinación 1 tiene tanto la perspectiva global como local, mientras que la combinación 2 solo se enfoca en la perspectiva local.

La siguiente clase es "Coniferous", que ocupa el 2.74 % del dataset, pero su métrica es bastante peor. Mediante el mapa de calor de dichos entrenamientos se puede observar que el modelo a veces se confunde con otras clases como "Deciduous" y "Brushwood". Estas dos clases también son de vegetación y tienen más datos en el dataset que la clase "Coniferous". Dicho factor es muy posiblemente la causa de que la métrica sea tan mala.

Después de la clase "Coniferous", se encuentra la clase "Deciduous", que es la segunda clase que ocupa la mayor cantidad de los datos dentro del dataset, pero su métrica no es tan deseada como las clases "Water", "Building", etc. Todas estas clases tienen una menor cantidad de datos, pero obtienen mejores métricas. En IOU, se puede observar que la diferencia entre las dos combinaciones es muy baja, con una diferencia de 1.75 %. En la métrica RECALL, ambas combinaciones tienen un valor del 70 %, que es un valor bastante alto y según dicha métrica, indica que las inferencias hechas por el modelo tienen un alto acierto.

La clase "Brushwood" tiene una métrica no deseada también, con un porcentaje de datos aproximadamente 7 % dentro del dataset. La diferencia entre las dos combinaciones sobre las métricas de IOU y RECALL es bastante leve. El modelo de la combinación 1 tiene una leve mejora respecto a la combinación 2. Según los mapas de calor, la clase "Herbaceous vegetation" tiene una alta influencia en el modelo cuando se hace la inferencia, porque la clase "Herbaceous vegetation" tiene la mayor cantidad de datos en comparación con el resto de

las clases. Este fenómeno también se ha observado en los experimentos anteriores. Cuando el modelo no sabe exactamente a qué clase pertenecen ciertos datos, la clase relativa que presenta un alto nivel de similitud y con mayor cantidad de datos puede influir significativamente en el modelo.

La clase "Vineyard" tiene un porcentaje del 3 % dentro del dataset, pero tiene buenos resultados en las métricas. En la métrica IOU, la combinación 2 tiene una mejora del 2.7 % que la combinación 1, y ambas combinaciones tienen un valor del 75 % y 77 % en la métrica IOU. Otras métricas tienen un valor aproximado del 90 %, lo cual es una señal bastante buena que indica que la red ha capturado las características esenciales de dicha clase y las usa para hacer inferencias posteriores. En la métrica DICE, donde se puede observar que el modelo entrenado con ambas combinaciones tiene una mejor precisión, tanto en las inferencias predichas correctamente como positivas sobre todos los datos realmente positivos, como en la proporción de inferencias correctamente predichas como positivas sobre todos los predichos positivos por el modelo.

La clase que ocupa la mayor cantidad de datos es "Herbaceous vegetation", con aproximadamente el 18 % del dataset. En teoría, al tener la mayor cantidad de datos, el modelo debería ser entrenado con éxito en comparación con otras clases, pero ocurre lo contrario, donde las métricas de dicha clase son bastante peores en relación con su gran cantidad de datos. En la métrica IOU, la combinación 2 tiene una mejora del 4.24 % que la combinación 1. Es una diferencia relativamente grande cuando se habla de un dataset con más de 70,000 datos para el entrenamiento, pero no es un valor deseado. En la métrica DICE, la combinación 2 también muestra una mejora del 4.21 % respecto a la combinación 1, pero en la métrica PRECISION, la combinación 1 tiene una leve mejora respecto a la combinación 2.

La clase "Agriculture Land", es la clase relacionada con "Vineyard" y es la cuarta clase que ocupa la mayor cantidad de datos, con un valor de aproximadamente el 11 %. Tiene casi cuatro veces más datos que la clase "Vineyard". Sin embargo, el valor de sus métricas es la mitad de peor que "Vineyard". En esta clase, la combinación 2 domina todas las métricas. En IOU, la combinación 2 tiene una mejora del 4.16 %; en DICE, la combinación 2 tiene una mejora del 4.93 %; en PRECISION, la combinación 2 tiene una mejora del 4.23 %; y por último, en RECALL, la combinación 2 tiene una mejora del 5.81 %.

La clase "Plowed Land" tiene un porcentaje de datos similar al de "Vineyard". En la métrica IOU, el modelo con la combinación 2 tiene una leve mejora que la combinación 1 y en todas las otras métricas, pero la diferencia no es tan grande.

La clase minoritaria "Swimming Pool" tiene un porcentaje de datos 0.03 %, aproximadamente al 0, es insignificante. Pero el modelo ha capturado las características de dicha clase, mientras que en otras clases minoritarias con un porcentaje de datos un poco mayor, el modelo no ha predicho absolutamente nada. Esto lleva a pensar si el modelo entrenado con las dos combinaciones diferentes ha conseguido las características necesarias para hacer la predicción. En ambas combinaciones tiene la clase "Pool", lo que es una llave importante, ya que en el modelo preentrenado de cada combinación ya se han capturado las características de dicha clase. En esta clase, la combinación 1 domina todas las métricas, aunque la combinación 2

tiene una mayor cantidad de datos de la clase “Pool”.

La última clase es “Greenhouse”, un tipo especial de la clase “Building”. Ambas combinaciones presentan un resultado muy similar, casi indistinguible.

En las métricas globales, se puede observar que el modelo entrenado con la combinación 2 se entrena un poco mejor que el modelo entrenado con la combinación 1. Esto sugiere que una mayor cantidad de datos no siempre ayuda al modelo a mejorar sus características, ya que una mayor cantidad de datos también puede significar un mayor porcentaje de ruido o errores que se introducen en el modelo durante la fase de entrenamiento. Esta puede ser la causa principal de las métricas deficientes. A veces, los datos de alta calidad, con características visibles y destacadas, pueden ayudar al modelo a hacer mejores inferencias.

En las siguientes imágenes(Ilustración 6.67) se puede observar las inferencias hechas por el modelo entrenado con Transfer Learning:

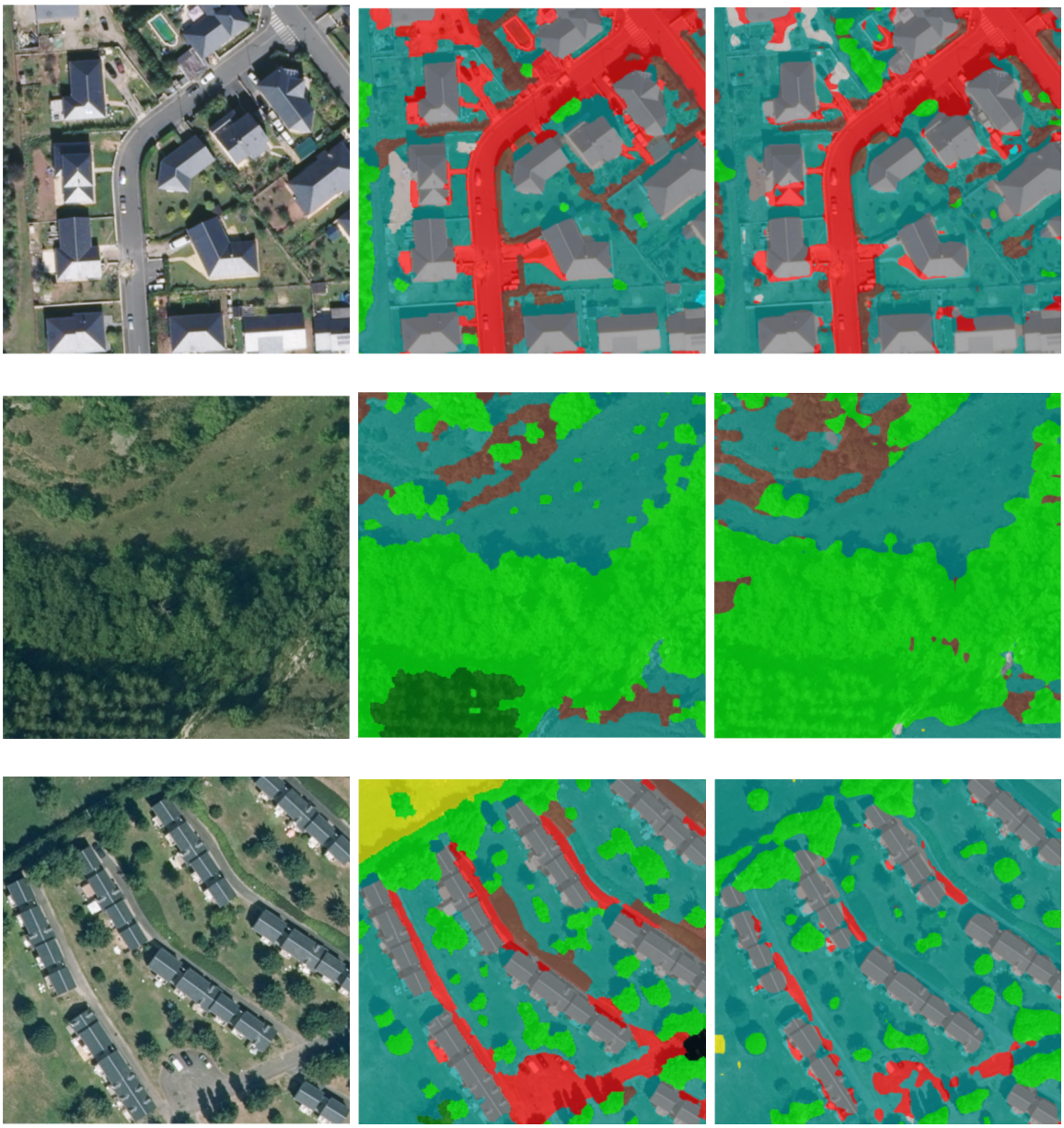

Ilustración 6.67: Datos, máscaras, inferencias sobre Flair

6.8. Validación con imágenes no etiquetadas

En esta última sección se lleva a cabo una segmentación semántica sobre las imágenes aéreas proporcionadas por la empresa Aerolaser System S.L. Los datos proporcionados son similares a otros datasets que se han utilizado durante el desarrollo del TFG. El primer paso consiste en hacer una redimensión a 512x512, porque la dimensión original de las imágenes es de alta resolución con 14204x10652.

Para hacer la predicción, se utiliza el modelo con mejor resultado en el transfer learning. La razón de seleccionar dicho modelo es debido a su mejor comportamiento en comparación con la otra combinación sobre los datos del testeo.

A continuación se muestran unas imágenes aéreas con la predicción hecha por el modelo, y para ello, hay que recordar los colores de cada clase en 6.4. A la izquierda se puede encontrar la imagen proporcionada por la empresa Aerolaser System S.L. A su derecha se encuentra la predicción de segmentación semántica hecha por el modelo entrenado. Para observar mejor la correspondencia entre la imagen y la predicción, en el margen derecho se encuentra la superposición de las anteriores.

En las siguientes imágenes (Ilustración 6.68) se puede observar que el modelo ha detectado principalmente las clases: “Pervious surface”, donde se detecta el territorio con agricultura cosechada o recién sembrada. Esta clase tiene una precisión bastante alta en comparación con otras clases. El modelo encuentra difícil diferenciar entre la clase “Agriculture Land” y “Herbaceous vegetation”, porque ambas clases tienen características similares. Respecto al ruido que está en la parte superior de la imagen, el modelo ha sido entrenado con datos recopilados verticalmente hacia abajo, por lo tanto, el cielo y el fondo no han sido entrenados en el modelo, y la imagen tiene una inclinación considerable que puede influir en la predicción del modelo.

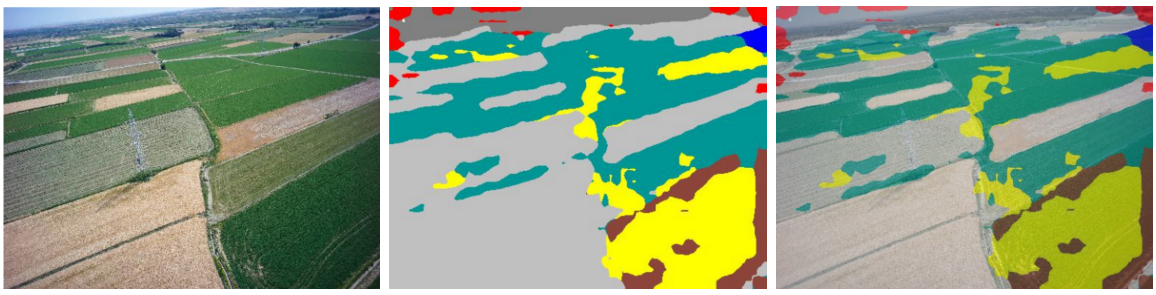

Ilustración 6.68: Ejemplo 1: Imagen RGB e inferencia del modelo

En estas imágenes se puede observar que el modelo ha predicho correctamente la clase “Agriculture Land” pero con ruidos de otras clases como “Plowed land” y “Herbaceous vegetation”. Especialmente la torre eléctrica, que es un edificio que el modelo no ha visto durante su entrenamiento, por lo tanto, es muy difícil de predecir para el modelo. Sin embargo, si el modelo se entrena con una cantidad adecuada de datos sobre la torre eléctrica, será capaz de identificarla. Respecto al área forestal, el modelo la ha detectado satisfactoriamente; la

línea entre “Agriculture Land” y la clase “Deciduous” está bastante clara. Otra clase como “Herbaceous vegetation” también tiene una buena segmentación. Al igual que antes, el modelo es incapaz de predecir lo que hay en el fondo porque no ha sido entrenado con dichos datos, pero las demás predicciones hechas son satisfactorias, aunque acompañadas de un poco de ruido.

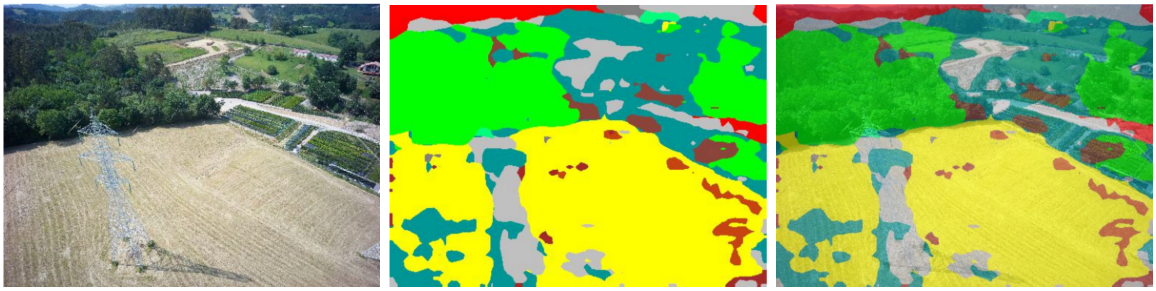

Ilustración 6.69: Ejemplo 2: Imagen RGB e inferencia del modelo

Capítulo 7

Conclusiones y trabajo futuro

Como se puede observar este proyecto tiene el objetivo final de, entrenar un modelo capaz de hacer segmentación semántica sobre imágenes aéreas de vegetación. Para ello, se han ampliado los conocimientos adquiridos en diferentes asignaturas de la carrera sobre inteligencia artificial, explorando nuevas posibilidades en la inteligencia artificial, trabajando con la librería Pytorch y otras librerías como Albumentations, sklearn, etc.

Tras todos los experimentos del proyecto, se puede obtener las siguientes conclusiones como:

1. La cantidad de datos influye en el resultado del aprendizaje de los modelos de redes neuronales, pero la calidad de los datos también forma un papel importante. La diferenciación de características entre clases y diversidad de datos, ayudan a que el modelo sea capaz de abstraer características más robustas durante el entrenamiento. Es decir, una mayor cantidad de datos no siempre significa un mejor resultado. Al igual que un ser humano, cuando se va a conocer un objeto que nunca ha visto antes, siempre intenta buscar las características propias de dicho objeto para diferenciar con otros objetos, y esta ley también se aplica a la red neuronal.

2. Un grado adecuado de Data Augmentation es fundamental. En Data Augmentation hay diferentes funciones para aumentar los datos de las clases minoritarias, pero siempre hay que tener en cuenta la relación entre las clases minoritarias y mayoritarias. Es importante elegir las funciones adecuadas para aumentar los datos, ya que algunas clases no se adaptan bien a la rotación, volteo, etc., mientras que otras clases se benefician de alterar el contraste, cambio de los canales RGB, etc. Siempre hay que considerar estos casos, para no obtener el efecto contrario durante el entrenamiento del modelo.

3. En la última sección con el Transfer Learning, se ha mostrado que enfocarse en los datos con una resolución espacial similar(recopilado por dron) resulta en un mejor resultado que combinar los datos de diferentes resolución espacial(recopilado por dron y satélite). Enfocarse en datos de alto nivel de relación puede ayudar a la red a concentrarse en el objetivo sin los posibles ruidos y es mejor entrenar un modelo con datos similares para hacer transfer learning. En el último experimento, se ha demostrado que el modelo con las características capturadas en otros datasets puede ayudar en las clases minoritarias.

Dentro de las diferentes técnicas utilizadas durante el desarrollo del TFG, la técnica de Data Augmentation sigue siendo una potente herramienta para mejorar las métricas del modelo. Con un data augmentation adecuado, en los experimentos anteriores se ha mostrado que se pueden aumentar aproximadamente un 8 % y un 3 % sobre las métricas usadas. Dicha técnica, combinada con la técnica de Transfer Learning, puede ayudar al modelo a aprender las características necesarias sobre las clases minoritarias y hacer buenas inferencias en futuros experimentos.

Por otra parte, se encuentran posibles mejoras que se pueden llevar a cabo en el futuro, como la prueba de diferentes arquitecturas. Aunque DeepLabV3 es una de las arquitecturas modernas para hacer la segmentación semántica, pero tambien existen otras como U-net, SegNet, etc., y también el modelo de Meta (SAM). Todas estas arquitecturas son muy interesantes para probar cada una de ellas, poner en práctica y absorber las ideas propias sobre dichas arquitecturas.

La segunda posible mejora sería probar diferentes técnicas para mejorar los resultados del modelo, ya que Data Augmentation es una técnica más usada a nivel mundial por su eficiencia y rapidez. Sin embargo, siempre existen otras técnicas como L2/L1 Regularization, Batch Normalization, etc. Estas técnicas se pueden combinar con Data Augmentation para obtener un mejor resultado, y es interesante hacer pruebas sobre ellas y su teoría subyacente.

Lo analizado, entrenado y validado en este TFG es un paso de inicio en dicho área compleja y amplia. Todavía hay infinitas posibilidades que se pueden explotar y descubrir.

Bibliografía

[1] Actualidad Aeroespacial. Imagen globo del tiempo. 2021. URL https://actualidadaeroespacial.com/la-nasa-estudia-con-globos-la-conexion-entre-el-sol-y-la-tierra/.

[2] Pagina Web Oficial Albumentation. Logo del albumentation. 2024. URL https://albumentations.ai/.

[3] Ibrahim Ismael Alnaib and Ahmed Nasser B. Alsammak. Advance artificial neural networks. page 10, 2022. URL https://www.researchgate.net/publication/363833676_Advance_Artificial_Neural_Networks.

[4] Ilke Demir, Krzysztof Koperski, David Lindenbaum, Guan Pang, Jing Huang, Saikat Basu, Forest Hughes, and Ramesh Tuia. Deepglobe 2018: A challenge to parse the earth through satellite images. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. URL https://www.kaggle.com/datasets/balraj98/deepglobe-land-cover-classification-dataset.

[5] CARLOS AUNIÓN DOMÍNGUEZ. Deep learning para la clasificaciÓn de imÁgenes de marcas. page 48, 2021. URL https://titula.universidadeuropea.com/bitstream/handle/20.500.12880/762/Carlos_Aunion_Domingez.pdf?sequence=1&isAllowed=y.

[6] Fabio Duarte. Number of chatgpt users (may 2024). 2024. URL https://explodingtopics.com/blog/chatgpt-users.

[7] Koustav Dutta, Rasmita Lenka, and Md Selim Sarowar. Improvement of denoising in images using generic image denoising network (gid net). page 4, 2022. URL https://www.researchgate.net/publication/358646375_Improvement_of_Denoising_in_Images_Using_Generic_Image_Denoising_Network_GID_Net.

[8] elchesemueve. Imagen del dron. 2023. URL https://elchesemueve.com/captura-fotos-epicas-desde-la-perspectiva-de-un-dron-aereo.

[9] Camilo Ferro, Nathalia Celis, and Alejandro Casallas. Llenado de series de datos de 2014 a 2019 de pm2.5 por medio de una red neuronal y una regresión lineal. page 8, 2020. URL https://www.researchgate.net/publication/343450066_Llenado_de_series_de_datos_de_2014_a_2019_de_PM25_por_medio_de_una_red_neuronal_y_una_regresion_lineal.

[10] ELSA VELASCO MARSELLA (FRANCIA). Imagen del satelite. 2018. URL https://www.lavanguardia.com/ciencia/fisica-espacio/20181211/453512507984/europa-satelites-copernicus-emisiones-co2-cambio-climatico.html.

[11] Rajat Garg, Anil Kumar, Nikunj Bansal, Manish Prateek, and Shashi Kumar. Semantic segmentation of polsar image data using advanced deep learning model. 2021. URL https://www.nature.com/articles/s41598-021-94422-y.

[12] Anatol Garioud, Stéphane Peillet, Eva Bookjans, Sébastien Giordano, and Boris Wattrelos. Flair #1: semantic segmentation and domain adaptation dataset. 2022. doi: 10.13140/RG.2.2.30183.73128/1. URL https://arxiv.org/pdf/2211.12979.pdf.

[13] HoTseChu. Logo del python. 2024. URL https://hotsechu.wordpress.com/2024/01/22/como-crear-un-libro-excel-a-partir-de-otro-en-python-conservando-las-imagenes-y-el-texto-enriquecid

[14] Yuyi Hu, Weizeng Shao, Wei Shen, Yuhang Zhou, and Xingwei Jiang. Machine learning applied to a dual-polarized sentinel-1 image for wind retrieval of tropical cyclones. December 2023. URL https://www.mdpi.com/2072-4292/15/16/3948.

[15] Joel Francia Huambachano. ¿qué es scrum? 2017. URL https://www.scrum.org/resources/blog/que-es-scrum.

[16] Nghi Huynh. Understanding evaluation metrics in medical image segmentation. March 2023. URL https://medium.com/@nghihuynh_37300/understanding-evaluation-metrics-in-medical-image-segmentation-d289a373a3f.

[17] Swathi Jadav and Vishhvak Srinivasan. Network optimization. page 19. URL https://deeplearning.cs.cmu.edu/F22/document/recitation/Recitation2/11785-NetworkOptimization-Fall22.pdf.

[18] Punitha Jaikumar, Remy Vandaele, and Varun Ojha. Transfer learning for instance segmentation of waste bottles using mask r-cnn algorithm. page 3, 2022. URL https://arxiv.org/pdf/2204.07437.

[19] Riley Lazarou. Watch this neural network learn to see. 2019. URL https://towardsdatascience.com/watch-this-neural-network-learn-to-see-545492272440.

[20] Ethan Yanjia Li. Witnessing the progression in semantic segmentation: Deeplab series from v1 to v3+. 2020. URL https://towardsdatascience.com/witnessing-the-progression-in-semantic-segmentation-deeplab-series-from-v1-to-v3-4f1dd0899e6e.

[21] MathWorks. Procesamiento de imágenes y visión artificial. 2024. URL https://es.mathworks.com/solutions/image-video-processing/semantic-segmentation.html.

[22] nattakit pinyorattanakit. Segmentation of datasets for train and test in machine learning. 2023. URL https://nattakit-nice2580.medium.com/segmentation-of-datasets-for-train-and-test-in-machine-learning-976179ae211d.

[23] Pagina Web Oficial OpenCV. Logo del opencv. 2024. URL https://opencv.org/.

[24] Neri Van Otten. Top 9 performance metrics in machine learning & how to use them. 2024. URL https://spotintelligence.com/2024/03/12/performance-metrics-in-machine-learning/.

[25] PSYCOLab. Imagen de neurona. 2021. URL https://www.psycolab.com/diferencia-entre-neurona-y-neuroglia/.

[26] Pagina Web Oficial PyTorch. Logo del pytorch. 2024. URL https://pytorch.org/.

[27] Maryam Rahnemoonfar, Tashnim Chowdhury, Argho Sarkar, Debvrat Varshney, Masoud Yari, and Robin Murphy. Floodnet: A high resolution aerial imagery dataset for post flood scene understanding. *arXiv preprint arXiv:2012.02951*, 2020. URL https://github.com/BinaLab/FloodNet-Supervised_v1.0.

[28] Alberto García Serrano. Visión artificial con redes convolucionales (cnn). 2019. URL https://www.ellaberintodefalken.com/2019/10/vision-artificial-redes-convolucionales-CNN.html.

[29] Saeid Asgari Taghanaki, Kumar Abhishek, Joseph Paul Cohen, Julien Cohen-Adad, and Ghassan Hamarneh. Deep semantic segmentation of natural and medical images: A review. page 9, 2020. URL https://arxiv.org/abs/1910.07655.

[30] Lark Editorial Team. Transfer learning. December 2023. URL https://www.larksuite.com/en_us/topics/ai-glossary/transfer-learning.

[31] totitu. Logo del anaconda. 2020. URL https://txtitx.wordpress.com/2020/01/30/4-3-anaconda-environments/.

[32] Wikipedia. Residual neural network. 2024. URL https://en.wikipedia.org/wiki/Residual_neural_network.

[33] Wikipedia. Logo del visual studio code. 2024. URL https://es.wikipedia.org/wiki/Visual_Studio_Code.

[34] Arne Wolfewicz. Deep learning vs. machine learning – what's the difference? February 2023. URL https://levity.ai/blog/difference-machine-learning-deep-learning.

[35] LALU ERFANDI MAULA YUSNU. Aerial semantic drone dataset. 2021. URL https://www.kaggle.com/datasets/nunenuh/semantic-drone?resource=download.