Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y Tecnología

Máster en Ingeniería Matemática y Computación

Segmentación semántica de imágenes para la obtención de rutas libres de obstáculos

Trabajo fin de estudio presentado por:	Carrillo Velarde, Cristhyan
Tipo de trabajo:	Tipo 2. Aplicación práctica real
Director/a:	Garrido Sáez, Neus
Fecha:	09 de febrero del 2021

Resumen

La identificación de patrones y objetos dentro de una imagen con el fin de elegir una ruta libre de circulación para el desplazamiento de un sistema móvil es un problema complejo si se tiene un entorno desconocido o con dificultades para la movilización.

En este trabajo se aplica un algoritmo de segmentación semántica basado en aprendizaje profundo que permite categorizar cada píxel dentro de una imagen. Se inicia con la adquisición y consolidado del conjunto de imágenes aéreas. A continuación, se ejecuta un proceso de etiquetado que nos permita diferenciar clases dentro de la imagen y posteriormente, se aplica un entrenamiento al modelo de segmentación semántica basada en una arquitectura SegNet con lo que se logra categorizar los píxeles de la imagen de manera precisa. Con la información extraída de la imagen se logra determinar una ruta libre de movilidad para un sistema móvil.

Los resultados obtenidos en este trabajo aplicando el modelo de segmentación semántica a un conjunto de 100 imágenes aéreas logra obtener una precisión superior al 82%, con lo que se alcanza a discernir una ruta libre de movilidad. La precisión puede ser mejorada si se aumenta el número de imágenes para ser analizadas. A la par, al procesar una gran cantidad de imágenes conlleva un aumento considerable en el tiempo de ejecución y de procesamiento.

Palabras clave: procesamiento de imágenes, segmentación semántica, SegNet, aprendizaje profundo.

Abstract

Identifying patterns and objects within an image to choose a circulation-free path for moving a mobile system is a complex problem if you have an unknown environment or with difficulty mobilizing.

In this work, a semantic segmentation algorithm based on deep learning is applied that allows categorizing each pixel within an image. It begins with the acquisition and consolidating of the aerial image dataset. Then, we perform a labeling process that allows us to differentiate classes within the image. Later, a workout is applied to the semantic segmentation model based on a SegNet architecture so that the pixels of the image can be accurately categorized. With the information extracted from the image it is possible to determine a mobility-free path for a mobile system.

The results obtained in this work applying the semantic segmentation model to a dataset of 100 aerial images achieves accuracy greater than 82%, therefore a mobility-free path is discerned. Accuracy can be improved by increasing the number of images to be analyzed. At the same time, processing a large number of images leads to a significant increase in execution and processing time.

Keywords: image processing, semantic segmentation, SegNet, deep learning.

Tabla de contenido

Índice de figuras

Índice de tablas

1. Introducción

La adquisición de la información a partir de una gran variedad de fuentes de datos es un pilar fundamental para el desarrollo del conocimiento y la toma de decisiones. A partir de esta premisa, concluimos que existen una variedad de fuentes donde se pueden obtener datos, pudiendo ser de diferentes índoles, tales como: fuentes de videos, fuentes de imágenes, fuentes de sonido, etc., que con el pasar de los años se han ido evolucionando y se han desarrollado técnicas y sensores más sofisticados para adquisición, que hace necesaria el uso de sistemas de cómputo más avanzados para el análisis y extracción de información útil para diferentes aplicaciones.

El procesamiento digital de imágenes ha tenido un gran crecimiento en las últimas décadas, convirtiéndose en un área que forma parte de nuestra vida diaria y un gran complemento en un sin número de aplicaciones, entre las que se pueden nombrar: robótica, medicina, telecomunicaciones, control de calidad, etc.

En la literatura asociada al procesamiento de imágenes digitales, se puede enunciar varios problemas coligados a la identificación de objetos o patrones dentro de una imagen. Según el trabajo de Sermanet et al. (2014) y de Toro (2019) se pueden definir 5 tipos de procedimientos:

- Segmentación de imágenes.
- Detección de objetos.
- Reconocimiento de objetos.
- Clasificación de imágenes.
- Anotación semántica.

Uno de los problemas más usuales en el procesamiento digital de imágenes y en el que se basará este trabajo es la de anotación semántica, que consiste en la adquisición de características extraídas de imágenes que permiten etiquetar cada píxel de la imagen con la finalidad de identificar patrones, objetos, entidades similares para clasificarlas y agruparlas en elementos con que comparten una similitud.

Tradicionalmente el problema de anotación semántica se ha limitado a procedimientos de eliminación de ruido, filtrado, y aplicación de segmentación binaria por medio del histograma

que nos da información del umbral óptimo con lo que se puede diferenciar ciertas estructuras y objetos dentro de una imagen. Recientemente se ha introducido al procesamiento digital de imágenes nuevos algoritmos inteligentes basados en el “Machine Learning” y en el “Deep Learning” o aprendizaje profundo con el fin dar solución al problema de anotación semántica.

En este trabajo se aplicará la segmentación semántica implementando un algoritmo de aprendizaje profundo que nos permitirá categorizar cada píxel dentro de una imagen. Tras la discriminación de las distintas categorías o clases, seremos capaces de clasificar el contenido de la imagen, extraer sus características y realizar un análisis para generar rutas de libre circulación.

1.1. Justificación

Nuestro mundo se enfrenta a un crecimiento tecnológico inmensurable donde el principal objetivo se basa en facilitar las operaciones realizadas por los humanos. Una de estas actividades se centra en la de abstracción información a partir de imágenes. Para la vista humana, la identificación de patrones para agruparlos según sus semejanzas y similitudes es un proceso rutinario y sin mayor complicación. Sin embargo, abstraer información para aplicaciones como la robótica, medicina, visión artificial, etc., se vuelve un problema con una complejidad agregada, por lo que se emplea sistemas de cómputo que nos ayuden con el procesamiento de algunos algoritmos para llegar al objetivo de extraer dichas características de forma automática y con una precisión aceptable.

Uno de los problemas actuales que abarca la robótica es el desplazamiento autónomo en entornos que presentan dificultades para la movilización. Este problema se puede encontrar en una gran variedad de aplicaciones tales como: vehículos auto dirigibles, drones, brazos robóticos, robots CNC, etc. Para ello se han investigado una gran variedad de algoritmos y diferentes métodos de obtención de datos de un entorno desconocido que será representado por medio de una imagen o fotografía. Las técnicas aplicadas difieren según las aplicaciones que van enfocadas, el nivel de dificultad presente o inclusive el grado de recursos que se tenga a disposición. Todos estos algoritmos tienen una base de matemáticas aplicadas y juntamente con la ayuda de procesadores computacionales podemos ejecutarlos y generar resultados lo más cercanos posibles a la realidad.

En nuestro trabajo emplearemos un algoritmo de segmentación semántica basado en aprendizaje profundo que nos permitirá categorizar y etiquetar cada píxel dentro de una imagen con el fin de distinguir las distintas categorías. De esta forma, podemos clasificar el contenido de la imagen y realizar el análisis para generar rutas de libre circulación.

A partir de imágenes de varios entornos reales con obstáculos, por medio del procesamiento de imágenes y el algoritmo de segmentación semántica podemos obtener un modelo de entorno capaz de diferenciar una ruta que se encuentre libre para circular y reconocer obstáculos presentes, para posteriormente aplicar un algoritmo para planificar una ruta desde un punto de inicio hasta un punto final evitando los obstáculos del entorno.

Como se mencionó en un inicio, las aplicaciones reales que abarcan esta problemática son varias y que tienen como punto de partida los mismos principios: adquisición de imágenes o videos del entorno, procesamiento de las imágenes, adquisición de información a partir de las imágenes para la toma de decisiones o control de actuadores que permita una aplicación en un ámbito real. Como consecuencia, este trabajo constituye la base para trabajos posteriores.

1.2. Problemática

Lograr determinar una ruta libre de obstáculos para la libre movilidad de un robot, un dron, o cualquier vehículo autónomo en un entorno desconocido nos lleva a plantearnos la siguiente inquietud: ¿Cómo se puede extraer información de cualquier entorno desconocido para identificar rutas libres de obstáculos que se puedan circular?

La identificación de patrones y objetos dentro de una imagen que será nuestro entorno a circular, es un proceso que dependerá de una segmentación adecuada, como indica Toro (2019) en su trabajo en donde se analizan varios algoritmos para la segmentación semántica de imágenes. Realizar una segmentación adecuada puede llegar a ser un problema no muy bien definido y puede llegar a ser un problema de percepción de cada observador.

Como solución a esta problemática proponemos la selección de imágenes aéreas que representarán el entorno desconocido a circular. Tras un determinado procesamiento se implementará un algoritmo de segmentación semántica basado en el aprendizaje profundo, que nos ayude a categorizar los píxeles dentro de la imagen de una manera autónoma, con el menor índice de errores. Definidos las categorías en la imagen, podemos obtener una máscara

que nos indique cuál es la ruta libre de obstáculos que permita la libre circulación de cualquier objeto o móvil.

1.3. Estructura del trabajo

La presente memoria se encuentra distribuida de la siguiente manera:

En el capítulo 1 se aborda la introducción al trabajo, así como la justificación que nos lleva a ejecutarlo y finalmente se plantea la problemática que se va a resolver.

En el capítulo 2 se describe el contexto de nuestro Trabajo Fin de Máster especificando las bases teóricas en que se basará nuestro planteamiento. Adicionalmente, analizaremos varios trabajos y estudios relacionados de diversos investigadores que buscan dar solución a la segmentación de imágenes por medio de algoritmos de segmentación semántica. Finalmente, analizaremos las ventajas de nuestro trabajo y el ámbito de aplicación que le daremos.

En el capítulo 3 se presenta cuáles son los objetivos que nos plantemos alcanzar en este trabajo, así como, la metodología que se emplea para alcanzar dichos objetivos. Además, se puntualiza las herramientas de software utilizadas para poder cumplir con el objetivo planteado.

En el capítulo 4 se describe el proceso llevado a cabo a lo largo del trabajo. Se describe detalladamente cada una de las fases ejecutadas para cumplir los objetivos. Para comenzar, se selecciona las fotografías o imágenes de interés. Luego, se aplica métodos de procesamiento digital de imágenes con el propósito de mejorarlas. A continuación, se etiqueta los píxeles con iguales características en categorías o clases. Después, se crea una arquitectura de redes neuronales para segmentación semántica. Posteriormente, se ejecuta el entrenamiento de la red neuronal para poder segmentar las imágenes. Por último, se aplica el algoritmo a imágenes de prueba que nos permite el análisis de resultados y la evaluación de la precisión obtenida.

En el capítulo 5 se analiza la aportación de nuestro trabajo y grado de cumplimiento de nuestros objetivos. Adicionalmente, se muestran posibles recomendaciones encontradas a lo largo de la ejecución de la propuesta y se describen trabajos futuros que se podrían implementar teniendo como base el presente trabajo.

2. Contexto y estado del arte

Este capítulo se encuentra divido en tres secciones. En la primera sección, se abordan los conceptos básicos y el contexto teórico que empleamos en el procesamiento de las imágenes digitales de entre los métodos empleados para segmentar imágenes. La segunda sección se enfoca en el análisis de diversos algoritmos de segmentación semántica y se definen conceptos básicos de aprendizaje profundo, así como el uso de redes neuronales para llegar al objetivo de obtener información de una imagen. En la tercera sección, se analizan los trabajos similares más influyentes que se encuentran relacionados con el ámbito de este trabajo. También se explican las similitudes, así como las diferencias de cada trabajo con respecto al proyecto que se ha desarrollado en esta memoria.

2.1. Procesado de imágenes

Una imagen es un arreglo bidimensional de píxeles y, por tanto, se puede representar como una matriz numérica. A cada elemento de esta matriz se lo denomina píxel que será el elemento más básico dentro de una imagen.

Según Alonso (2009) podemos distinguir dos tipos de imágenes:

- Imagen monocromática: una matriz de dimensión $m \times n$, donde cada píxel tiene asociada una función que corresponde a la intensidad de luz en cada una de las coordenadas de la imagen (x, y).

$$Im = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,n-1) \\ f(1,0) & f(1,1) & \cdots & f(1,n-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(m-1,0) & f(m-1,1) & \cdots & f(m-1,n-1) \end{bmatrix}$$

- Imagen a color: Combinación de tres colores básicos: rojo, verde y azul representados en un arreglo de tres matrices de tamaño $m \times n$ o mejor conocido como el espacio de colores RGB donde cada componente representa el nivel de intensidad de cada color básico. En el caso de imágenes digitales los valores de las componentes R, G y B están representados por valores números enteros dentro del rango de 0 a 255.

Estos niveles de intensidades con que definimos a las imágenes podemos representarlos mediante un gráfico denominado histograma donde se puede evidenciar la distribución de los distintos tonos. En el eje de las abscisas se incorporan los tonos que va desde el negro puro hasta el blanco puro, mientras que en el eje de las ordenadas se distribuyen la cantidad de píxeles para cada tono.

El histograma será una de las primeras herramientas utilizadas para poder valorar si una imagen es adecuada para su posterior análisis ya que nos permite obtener información como el contraste, el tipo de fondo y si los niveles de intensidad del color están distribuidos de forma homogénea. Adicionalmente, nos puede proporcionar información estadística como:

- Media, representa el valor de intensidad medio.
- Desviación estándar, representa la variación de los valores de intensidad.
- Rango: Valores en los que se extiende la intensidad.

En la Figura 1 se ilustra el histograma de una imagen a color, donde se observa cómo se distribuyen las intensidades correspondientes a cada una de las componentes del espacio de color RGB.

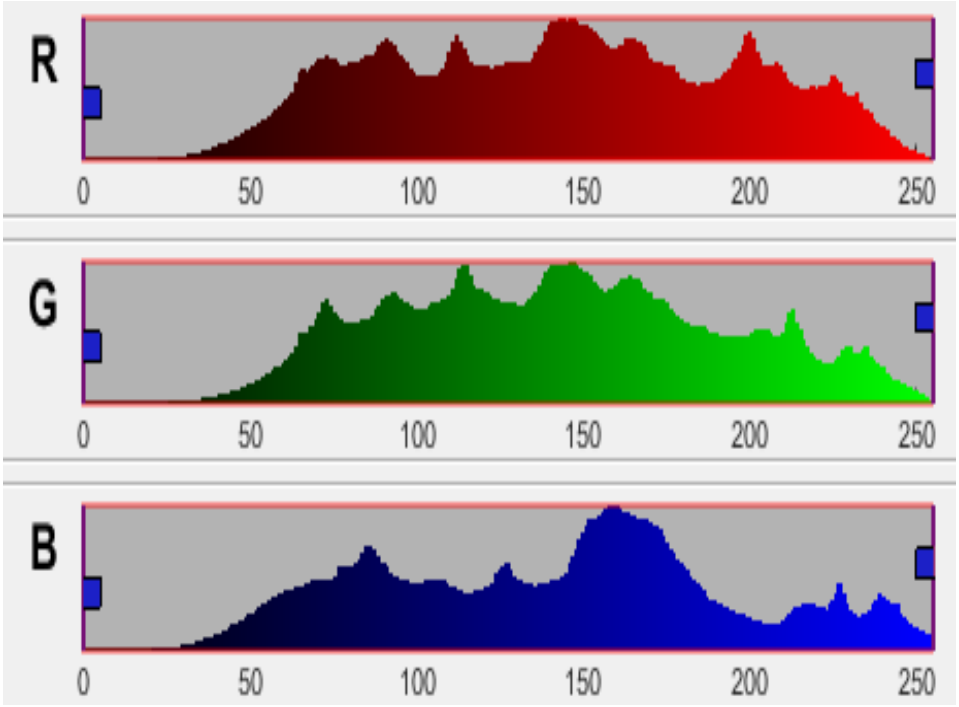


Figura 1. Histograma de una imagen a color. (Elaboración propia)

Para poder llevar acabo la extracción de las características de una imagen y lograr una segmentación semántica, es imprescindible llevar a cabo varias fases como lo plantea Toro (2019) y se puede ver en la Figura 2.

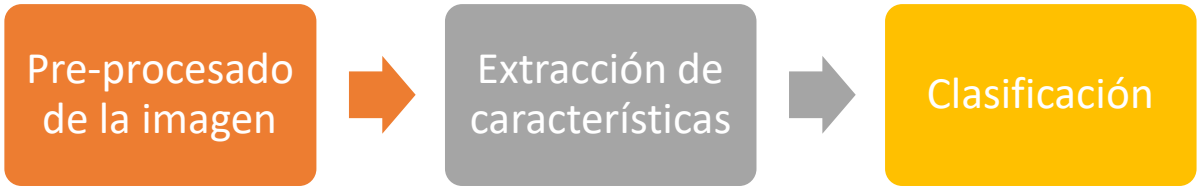


Figura 2. Fases para la segmentación semántica. (Elaboración propia)

Cada una de estas fases lleva asociadas un conjunto de técnicas y algoritmos que depende del problema que se desea resolver, depende de la aplicabilidad para hacer uso de una o de otra o inclusive una combinación de varias técnicas.

A continuación, se describen las principales técnicas aplicadas dentro de cada una de las fases.

2.1.1. Pre-procesado de imágenes

La primera fase consiste en un procesamiento previo que constará de un conjunto de técnicas que se aplican a las imágenes con la finalidad de eliminar imperfecciones, realzar características de interés o minimizar elementos no deseados que se pueden presentar al momento de adquirirla con la finalidad de obtener una imagen mejorada para el posterior análisis.

Una función de procesado de imagen se puede expresar como:

$$s(x,y) = T[f(x,y)]$$

donde $f(x,y)$ representa la imagen adquirida, que por medio de una operación de transformación T, se obtiene una imagen de salida $s(x,y)$.

Estas técnicas aplicadas en el pre-procesamiento se pueden clasificar en tres categorías: reconstrucción de imágenes, restauración de imágenes (eliminar cualquier imperfección al momento de adquirir la imagen) y mejora de la imagen (realce de ciertas características deseadas) (Bharathi & Subashini, 2011). Una vez aplicadas estas técnicas, se pretende facilitar

los pasos posteriores de procesamiento para lograr un alto grado de eficiencia y mejorar el impacto en tiempos de ejecución y uso de recursos computacionales. Dependiendo de la aplicación a la cual está asociada la imagen a ser procesada, es necesario elegir los aspectos para ser mejorados o eliminados.

Generalmente estas técnicas se pueden efectuar en el dominio del espacio (manipulación directa de los píxeles), así como en el dominio de la frecuencia (modificación de la transformada de Fourier). En el trabajo de Enríquez (2016) podemos ver algunas técnicas aplicadas en el pre-procesado:

- **Aumento de contraste:** permite incrementar el rango dinámico de los niveles de grises de una imagen, especialmente en imágenes con una iluminación defectuosa o errores en el momento de la adquisición.
- **Ecualización del histograma:** obtiene una distribución los más uniformemente posible de los niveles de grises, con lo que podemos equilibrar los niveles entre el rango negro al rango blanco sobre todos los niveles disponibles.
- **Filtrado espacial:** por medio de máscaras espaciales denominadas filtros, se puede realizar un filtrado de la imagen. Un filtro tiene como objetivo modificar la contribución de determinados rangos de frecuencias en ciertas áreas de la imagen. Se pueden aplicar filtros pasa bajo, pasa alto, pasa banda o rechaza banda dependiendo de la modificación que se necesite, entre las cuales se pueden nombrar: realce de nitidez, eliminación de ruido, detección de bordes, etc. Matemáticamente podemos definir como $s(x,y) = f(x,y) * h(x,y)$, donde $h(x,y)$ representa la función de transferencia del filtro aplicada a la imagen de entrada $f(x,y)$, lo que genera la imagen de la salida $s(x,y)$.
- **Operaciones aritméticas:** permite corregir imágenes por medio de la aplicación de operaciones como suma, resta, multiplicación y división. Al ser las imágenes representables como una matriz numérica no existe mayor incompatibilidad. Aplicaciones de estas operaciones pueden quitar: el aumento de brillo, la sustracción para aislar objetos específicos, el escalamiento y la racionalización.
- **Operaciones lógicas:** aplicación de operadores AND, OR, NOT a imágenes representados con valores binarios.

- **Transformaciones geométricas**: permiten corregir escenas por medio de la rotación, el desplazamiento y la corrección de perspectivas, modificando las relaciones espaciales entre los píxeles para facilitarnos el análisis y segmentación.
- **Realce de bordes**: consiste en resaltar píxeles con la finalidad de mejorar la nitidez de la imagen.
- **Detección de contornos:** permiten obtener los lugares en la imagen donde la intensidad de un píxel varía rápidamente utilizando criterios de cálculo como derivadas y gradientes. Estos algoritmos son ampliamente utilizados en el reconocimiento de patrones.

Con la aplicación de varias de las técnicas descritas enfatizaremos ciertas características que nos sean de utilidad para la extracción de información y de una u otra forma mejorar la productividad al aplicar el algoritmo de segmentación semántica.

2.1.2. Extracción de características

Una vez aplicadas las técnicas de pre-procesamiento a las imágenes de estudio, seguiremos con la fase de extracción de características cuyo objetivo es encontrar un subconjunto de variables informativas que clasifique adecuadamente el contenido presente en la imagen. Las características extraídas se utilizan para la segmentación o el reconocimiento de objetos.

Toro (2019) menciona que se pueden dividir en cuatro tipos diferentes de caracterizaciones a extraer de una imagen. Éstas son:

- **Espectrales**

Se pueden extraer características basadas en la cuantificación de la intensidad del color, medidas de similitud, colores dominantes, etc. Esta extracción se basa en la tendencia del conjunto de valores que pueden ser representados por una función de densidad probabilística y se adquieren parámetros como la media, la varianza, asimetrías o sesgos y la curtosis de la distribución de estos valores.

- **Texturales**

Las características extraídas se basan en la variación entre los niveles de intensidad en la vecindad de un píxel. Estas variaciones pueden ser observadas en un histograma. Podemos asociar texturas como suaves, rugosas, irregulares, duras, etc.

- **Basadas en la forma**

Una de las técnicas más usuales al momento de extraer características se basa en buscar relaciones de formas, donde se pueden observar dos tipos: basadas en regiones y basadas en contornos.

- **Basadas en el entorno**

Se basan en relaciones espaciales entre objetos relacionados por medio de orientaciones, distancias o topologías.

2.1.3. Clasificación

Uno de los principales problemas que abarca el aprendizaje autónomo es la tarea de clasificación, que consiste en asignar instancias de un dominio determinado a un conjunto de clases (Gu et al., 2020).

En las investigaciones presentadas por Borràs et al. (2017) y Paoletti et al. (2019) se describen los principales algoritmos de análisis de imágenes para la clasificación, para lo cual se emplean imágenes hiperespectrales y se realiza un análisis comparativo entre estos métodos de clasificación automático.

En los últimos años, estos algoritmos de clasificación están basados en métodos de aprendizaje autónomo, es decir, basados en sistemas que implementan algún tipo de inteligencia artificial debido a la gran cantidad de datos que se necesitan analizar y la naturaleza compleja de los mismos. Dependiendo de las entradas de nuestro algoritmo (imágenes etiquetadas), su procesamiento se puede clasificar en dos tipos: clasificación supervisada y clasificación no supervisada.

2.1.3.1. Clasificación supervisada

Método en el cual se inicia con una cantidad de muestras etiquetadas donde se conoce a priori las clases de interés dentro de la imagen. Estas clases serán previamente definidas por el analista con el objetivo de que, por medio de un entrenamiento, calculemos ciertos parámetros y generemos un modelo matemático a partir de la cual se evalúen otras imágenes en donde el algoritmo asignará a cada píxel una determinada clase.

Como menciona Paoletti et al. (2019), la clasificación supervisada es la más empleada debido al gran rendimiento de inferir una correcta clasificación a partir de un conjunto considerable de imágenes etiquetadas previamente, además de poseer un alto grado de exactitud.

Dentro de la clasificación supervisada se puede distinguir dos enfoques: generativos o llamados probabilísticos y discriminativos o métodos no probabilísticos.

- Algoritmos generativos: Se basan en un análisis probabilístico con el fin de encontrar la probabilidad $p(x|c)$, de que un elemento x pertenezca a una clase c usando modelos bayesianos.
- Algoritmos discriminativos: conocidos como condicionales en donde a cada etiqueta se considera por separado y cada una entrena a un modelo específico.

2.1.3.2. Clasificación no supervisada

Algoritmos donde no se necesita la fase previa de entrenamiento con una muestra inicial de imágenes etiquetadas, es decir, no es conocida a priori las clases de interés dentro de la imagen. Dicha clasificación se ejecuta de manera autónoma buscando características como tamaños, texturas y formas.

2.1.4. Métodos de clasificación

Existe una amplia variedad de técnicas para la clasificación de imágenes, y como se mencionó previamente los más utilizados son los basados en una clasificación supervisada. Por este motivo, a continuación, se realiza una revisión de los algoritmos fundamentales.

- Máquina de vectores de soporte: conocido como SVM, desarrollado por Suykens y Vandewalle (1999), es un método de clasificación supervisado que busca realizar una clasificación binaria entre las clases, buscando un hiperplano que divida las muestras en dos zonas distintas maximizando la distancia de las muestras más cercanas a la zona de decisión. Este tipo de algoritmos presenta muy buenos resultados cuando se emplean para grandes conjuntos de datos o cuando se tienen pocas muestras de entrenamiento. En la Figura 3 se observa la aplicación del método SVM.

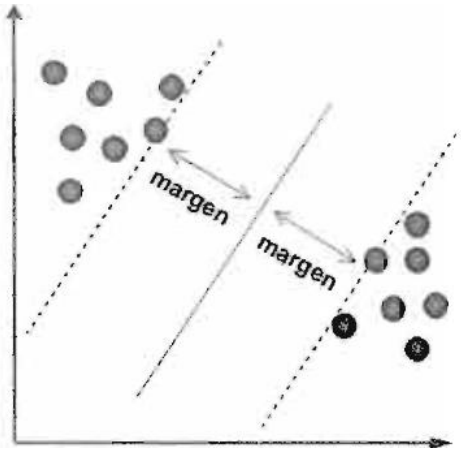


Figura 3. Método SVM. (Rosales, 2019)

- Árboles de decisiones: son métodos de clasificación supervisada paramétricos que organizan los datos de manera jerárquica en forma de árbol que va desde una raíz hasta llegar a un nodo terminal (Breiman, 1998). Cada nodo representa un atributo a ser probado; las ramas representan la salida de la prueba y los nodos finales (hojas) representan la clasificación. Es fácil distinguir la relación entrada-salida ya que se basan en decisiones del tipo *if-then.* Rosales (2019) indica que la razón por la que los árboles de decisión son los más utilizados es que permiten comprender fácilmente a qué clase pertenece un objeto u evento. No pueden existir dos objetos que pertenezcan a dos clases debido a la naturaleza excluyente del método y pueden incluir varios tipos de datos de variables.
- Redes Neuronales: técnicas basadas en clasificación supervisada que tratan de similar la manera en que un ser humano aprende. Se componen de elementos o nodos denominadas neuronas que se encuentran interconectadas en una o varias capas. La exactitud y nivel de complejidad dependen del número de capas y de nodos que se eligen para desarrollar el modelo. Son los métodos más utilizados en la actualidad y en el que se basará este trabajo.

2.1.5. Segmentación

Extraer información de interés dentro de una imagen, identificar patrones o aislar objetos de interés es uno de los principales y más complejos problemas que abarca el procesamiento de imágenes, por lo que se han investigado y desarrollado varias técnicas de segmentación que permiten llegar a estos objetivos.

Como define Toro (2019) en su investigación, el proceso de segmentación consiste en distinguir regiones dentro de una imagen, con el fin de que cada región tenga un criterio de homogeneidad, es decir, se debe buscar características similares como el color, la textura, la luminosidad, etc., para posterior agrupar los píxeles en grupos tal que se distingan un grupo de otro.

Enríquez (2016) ha descrito los principales algoritmos de segmentación que difieren dependiendo de la aplicación que se busca. Entre los principales métodos para segmentar una imagen podemos enumerar:

- **Detección de discontinuidades**

Se aplica sobre imágenes monocromáticas y se basa en los cambios bruscos de nivel de gris. Se logra pasando una máscara a través de la imagen con lo que logramos detectar discontinuidades: puntos, líneas y bordes.

El método más usado para distinguir discontinuidades es la detección de bordes (frontera entre dos regiones que difieren sus niveles de gris). Para lograr la detección y realce de bordes se utilizan herramientas matemáticas como las derivadas, gradientes, laplaciano, que se basan en el análisis de la pendiente de la intensidad luminosa. Existen varias técnicas que difieren en los operadores matemáticos utilizados y la manera en que los aplican a las imágenes. Entre los principales métodos que utilizan la derivada y el gradiente se pueden citar el operador de Robert, el operador de Prewitt, el operador de Kirsh y el operador de Sobel. Uno de los principales métodos que aplica el laplaciano de una función gaussiana recibe el nombre de operador de Marr-Hildreth.

- **Enlazado de bordes**

Una vez que se ha logrado identificar los píxeles que presentan discontinuidades, es necesario reunir estos píxeles que presentan similitudes en límites. Para el enlazado de bordes, los métodos más utilizados son la transformada de Hough, basado en la obtención de un conjunto de puntos del borde por medio de la operación gradiente, y la teoría de grafos, que se sustenta en buscar los caminos de menor coste que serían los bordes significativos en donde cada segmento del borde forma parte del grafo que se desea optimizar.

- **Umbralización (Thresholding)**

Es uno de los métodos más sencillos al momento de aislar una región de interés con regiones que corresponden al fondo de la imagen. Este método se basa en las diferencias de intensidades.

En imágenes monocromáticas, el proceso de umbralización consiste en comparar la intensidad de cada píxel con un valor denominado umbral de intensidad. Dependiendo si el resultado es mayor o menor se discrimina si el píxel corresponde al fondo o a la región de interés. Como resultado se puede generar una imagen binaria. En imágenes de color se utiliza varios valores de umbral para comparar cada píxel con el umbral correspondiente al canal de color, ya sea en R (rojo), G (verde) o B (azul).

Para poder determinar los valores de umbral de intensidad nos podemos remitir al histograma de una imagen, el cual ya hemos visto anteriormente que nos muestra los valores de intensidad vs la cantidad de píxeles que corresponde a cada intensidad. Uno de los aspectos primordiales en este método es la selección del umbral adecuado tal que tenga la capacidad de identificar fidedignamente los picos de un histograma (Enríquez, 2016). Por lo expuesto anteriormente, existe una variedad de métodos para la determinación de un umbral óptico. Según la distribución de intensidad de gris dentro de la imagen, podemos seleccionar varios umbrales para analizar regiones por regiones o usar un umbral adaptativo, es decir, que puede cambiar dinámicamente para todos los píxeles.

- **Segmentación orientada a regiones**

El objetivo de este tipo de segmentación es dividir la imagen en regiones homogéneas. Por un lado, existen métodos que empiecen con regiones iniciales pequeñas, para ir añadiendo píxeles y formando regiones más y más amplias hasta que no cumplan las condiciones de homogeneidad con regiones vecinas. Por otro lado, hay métodos que empiezan en regiones relativamente grandes para posterior empezar a dividirlas hasta cumplir las condiciones de homogeneidad.

2.2. Redes Neuronales artificiales

Flórez y Fernandez (2008) definen a las redes neuronales artificiales como modelos matemáticos artificiales que intentan simular las estructuras del sistema nervioso biológico, basado en la generación de conocimiento a partir de la experiencia. Es un sistema de cómputo desarrollado para el tratamiento de información compuesto por varios elementos computacionales simples entrelazados con otros elementos.

En el estudio realizado por Coello Blanco et al. (2015) se observa cómo se han desarrollado una variedad de modelos de redes y arquitecturas, entre las cuales destacan el modelo neuronal de McCulloch y Pitts, los modelos ADALINE y MADALINE, que constituyen un tipo de red neuronal artificial desarrollada por BernieWidrow y MarcianHoff en la Universidad de Stanford en 1959, las redes MPL (Multi-Layer Perceptron por sus siglas en inglés) aplicado a resolver múltiples problemas.

2.2.1. Elementos básicos

Se puede representar a una red neuronal artificial por medio de un grafo, constituido por elementos simples llamados *nodos* representados por círculos, que se encuentran conectados bidireccionalmente entre los mismos por medio de *conexiones* en modo de flechas. Los *nodos de entrada* son aquellos que no poseen conexiones entrantes, los *nodos de salida* son aquellos sin conexiones salientes, mientras que todos los demás nodos que no se encuentran en la entrada ni la salida se encuentran en *capas ocultas.* En la Figura 4 se puede observar la estructura de una red neuronal.

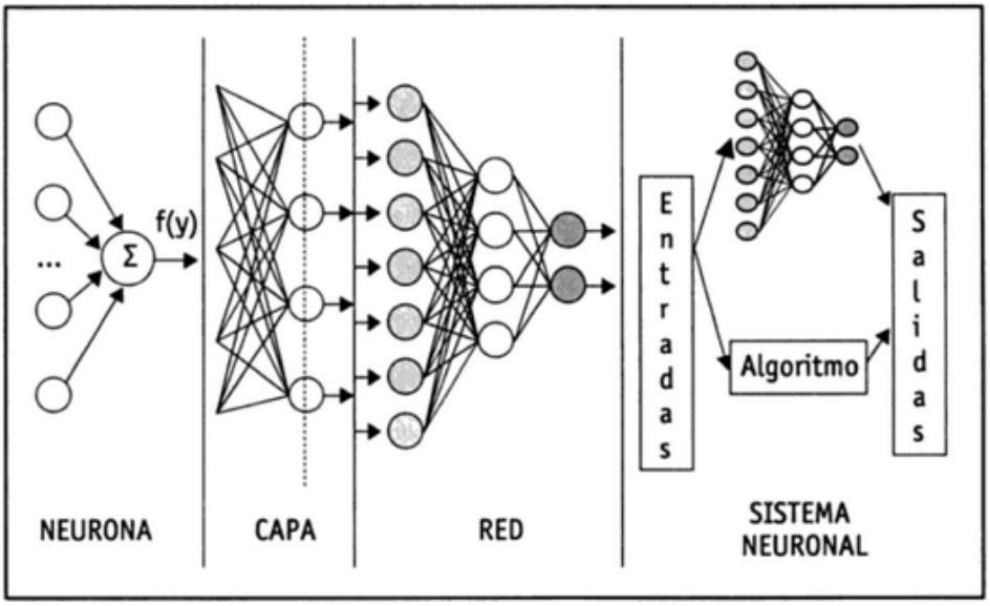


Figura 4. Estructura de una red neuronal artificial. (Flórez & Fernandez, 2008)

A continuación, describimos cómo se ejecuta el método de cómputo presente en cada nodo. En primer lugar, las entradas de la neurona $(x_1, x_2, \dots, x_n)$, poseen sus pesos específicos (w_{ij}). La función de propagación permite, a partir de las entradas y sus pesos, calcular un valor del potencial de la neurona de acuerdo con una determinada función. Usualmente, la función más usada es la lineal, que se basa en la suma ponderada de las entradas con los pesos asociadas a ellas, y no es más que el producto escalar de los vectores de entrada y los pesos de la red. A este producto se le añade un valor (b) denominado *sesgo* o *bias* el cual es otro de los parámetros que debe ser aprendido por el modelo. En la Figura 5 se observa la expresión matemática del cómputo de una neurona, así como la representación gráfica de las entradas, el proceso de cómputo y la salida.

$$z = \sum_i {w_i}^T x_i + b$$

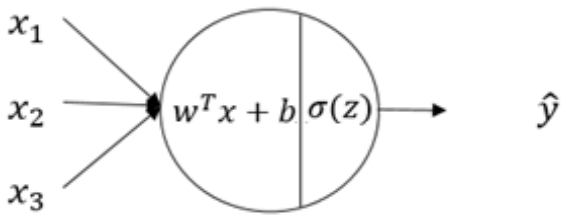
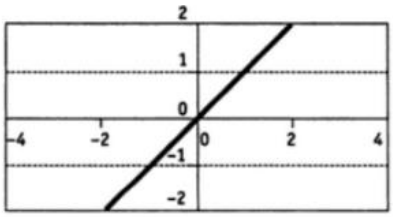
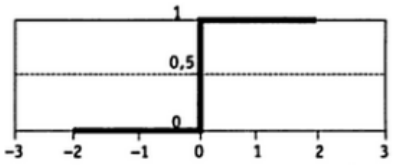
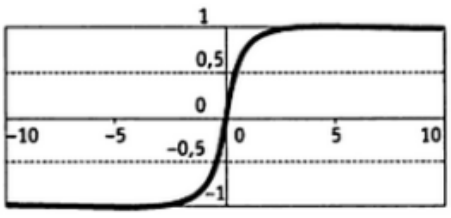
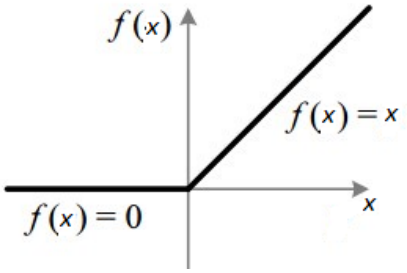
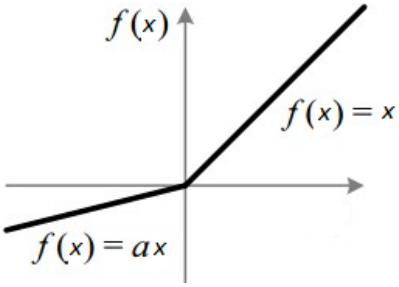


Figura 5. Sistema de cómputo de una neurona artificial. (Elaboración propia)

Al valor obtenido en la función de propagación se le aplica una función de activación, que produce los valores de salida de la neurona artificial, conocido como estado de activación, y cuyo rango normalmente va de 0 a 1 o de –1 a 1. En la Tabla 1 se observan las funciones de activación más comunes junto a una breve descripción y representación gráfica.

Tabla 1. Funciones de activación más comunes

Función	Descripción	Gráfica
Función lineal	Devuelve directamente el valor de activación de la neurona. $f(x) = x$	
Función escalón	Si la activación de la neurona es inferior a un determinado umbral, la salida se asocia con un determinado output, y si es igual o superior al umbral se asocia con el otro output. $f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$	
Función sigmoidea	Genera valores de salida comprendidos dentro de un rango que va de 0 a 1, y caracterizados por tener una derivada siempre positiva e igual a cero en sus límites asintóticos y de valor máximo cuando $x = 0$. $f(x) = \frac{1}{1 + e^{-x}}$	

Función ReLU	Anula a los valores negativos de entrada, para valores positivos devuelve directamente el valor de entrada. $f(x) = max(0, x)$	
Función Leaky ReLU	Similar a la función ReLU. La diferencia radica en que para valores negativos en la entrada, los transforma multiplicando por un coeficiente rectificado. $f(x) = max(0.01x, x)$	

(Flórez & Fernandez, 2008)

2.2.2. Topologías de redes neuronales artificiales

Flórez y Fernandez (2008) mencionan que las arquitecturas de las redes neuronales artificiales o RNAs se basa en cuatro parámetros:

- Número de capas del sistema.
- Número de nodos en cada capa.
- Grado de conectividad entre los nodos.
- Tipo de conexiones neuronales.

Según las estructuras de las capas podemos tener redes monocapa y redes multicapa; según el flujo de datos pueden ser de propagación hacia adelante (feedforward), o redes de propagación hacia atrás (feedback).

2.2.3. Mecanismo de aprendizaje

Obtener un valor de salida para poder tomar una decisión es el objetivo de una red neuronal, por lo que el problema radica en cómo enseñarle a la red a que tome la decisión correcta. El

aprendizaje será la clave para que la red modifique sus valores dependiendo de los valores de entrada.

Según Flórez y Fernandez (2008): "el aprendizaje es el proceso en el que la red neuronal crea, modifica o destruye sus conexiones (pesos) en respuesta de una información de entrada". Entendiendo este concepto se llega a la conclusión de que es necesario entrenar a la red previamente con un conjunto de datos de entrenamiento, con lo que crearemos nuevas conexiones, es decir, modificamos los valores de los pesos. Una vez finalizado el proceso de aprendizaje cuando los pesos no tengan variaciones, la red neuronal extrae conocimiento con lo que podemos llevar a cabo la tarea que nos planteamos resolver.

El objetivo del aprendizaje es minimizar una función de coste o error que viene dado entre el valor predicho por la red neuronal y el valor esperado. La actualización de los pesos es un proceso iterativo que finaliza cuando se alcance un rendimiento óptimo que puede venir dado por un nivel de error máximo aceptable o hasta alcanzar el número de iteraciones establecidas a priori.

Siendo el objetivo que pretende el proceso de aprendizaje minimizar una función de coste, la forma en que actualizan los valores de los pesos permite definir dos tipos de aprendizaje: el aprendizaje supervisado y el aprendizaje no supervisado.

- Aprendizaje supervisado: este tipo de aprendizaje se caracteriza por realizar el aprendizaje mediante un entrenamiento controlado por un agente externo determinando el valor de salida que debe generar a partir de una entrada determinada. Mediante un algoritmo de aproximación buscamos minimizar la función de error. De esta manera el supervisor compara el valor de salida de la red con el valor de salida deseada y procede ajustar los valores de los pesos con el fin de que la salida generada sea lo más similar a la deseada.

  El proceso para aplicar un algoritmo de aprendizaje automático supervisado consiste en obtener los valores de los parámetros (*pesos y sesgo*) examinando una gran cantidad de muestras etiquetadas e intentar determinar unos valores para estos parámetros del modelo que minimicen el error lo que denominamos *función de error.*
- Aprendizaje no supervisado: este tipo de aprendizaje se caracteriza por no requerir de ningún agente externo para aprender. Por medio de un conjunto de datos de entrada se deja que la red encuentre características, regularidades, correlaciones o categorías y los agrupe según crea conveniente, sin que se proporcionar a la red los patrones de salida que

confirmen si la salida es correcta o incorrecta. La salida representa el grado de similitud entre la información en la entrada y las informaciones que se le han mostrado en el pasado.

2.2.4. Deep Learning o aprendizaje profundo

Se conoce como aprendizaje profundo a una rama del machine learning, que se estructura a partir de redes neuronales artificiales compuestas de múltiples capas de procesamiento con muy buenos resultados en la extracción de características aplicadas a imágenes y vídeos. Estas capas ocultas se encuentran apiladas una encima de otra, motivo por el cual adaptan el término "profundo". Cada una de estas capas sirve para diferentes propósitos, y cada parámetro e hiperparámetro afecta en el resultado final (Torres, 2018).

2.2.4.1. Redes neuronales Convolucionales (CNN)

Las redes neuronales convolucionales son un caso particular del aprendizaje profundo que presenta los mejores resultados en problemas de reconocimiento, clasificación, detección de objetos en imágenes y vídeos, basadas en operaciones de convolución.

El propósito de las CNN es extraer todas las características de una imagen y luego usar dichas características para detectar o clasificar los objetos en una imagen (Massiris et al., 2018). Uno de los principales problemas en las CNN es el entrenamiento, ya que consume una gran cantidad de recursos y memoria, además de ser necesaria una gran cantidad de muestras etiquetadas para el correcto entrenamiento.

Quintero et al. (2018) nos muestran la aplicación de una red neuronal convolucional para el reconocimiento automático de imágenes de macroinvertebrados, donde nos muestra como una CNN está compuesta por varias capas, empezando por la fase de extracción de características, siguiendo por una etapa que disminuye las dimensiones para activar características más complejas y al final la etapa de clasificación.

Abordando más en las redes neuronales convolucionales, estas constan de varias capas convolucionales y de pooling (submuestreo) alternadas, finalmente tienen varias capas full-connected. En la investigación presentada por Durán (2017) nos muestra las etapas de una red convolucional, las cuales se observan en la Figura 6.

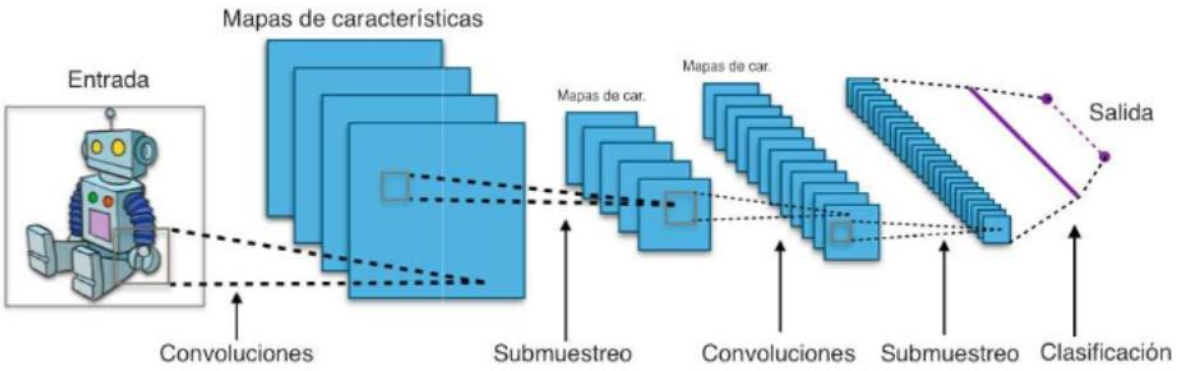


Figura 6. Etapas de una red neuronal convolucional. (Durán, 2017)

- **Capa de entrada**

A la entrada tenemos una imagen de dimensiones: $m\ x\ m\ x\ r$; donde m representa al número de píxeles por filas y columnas; mientras que r representa el número de canales.

- **Capas convolucionales:**

En la capa de convolución se realizan operaciones de sumas y productos entre la imagen de entrada y un número k de filtros (kernels) de dimensiones: $n\ x\ n\ x\ q$; donde n, q son elegidos en el diseño. Mediante la convolución cada filtro genera un mapa de características de dimensiones $(m-n+1)x(m-n+1)\ x\ p$, donde p representa el número de filtros. El principal objetivo de la capa de convolución es reducir el número de conexiones entre neuronas de la capa oculta con los elementos de la imagen de entrada.

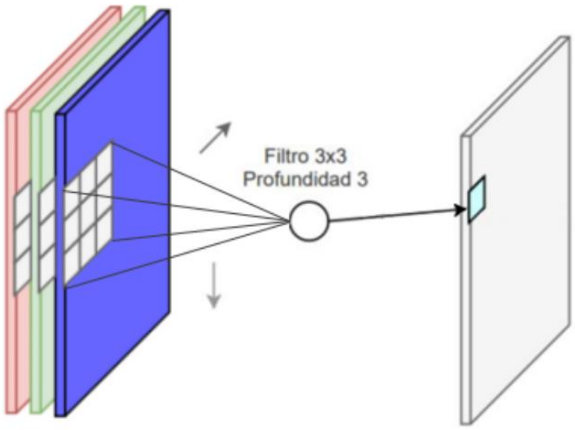


Figura 7. Proceso de convolución (Méndez, 2019)

En la Figura 7 se observa el proceso de convolución del filtro con una imagen con el fin de generar una nueva matriz de características o de activación, es un proceso iterativo que se ejecuta a lo largo de toda la imagen.

- **Capa de pooling**

En esta capa se realiza un sub-muestreado sobre regiones continuas con el objetivo de disminuir las dimensiones de la matriz de características y obtener rasgos predominantes en esta matriz. Para realizar esta labor, se puede calcular la media, que se denomina mean-pooling, o buscar el máximo valor de una característica a lo largo de una región de la imagen, que se conocen como max-pooling.

En la Figura 8 se observa cómo se genera la nueva matriz dependiendo del tipo de pooling. A partir de la matriz de características que es de mayor dimensión, se generan nuevas matrices de dimensiones inferiores.

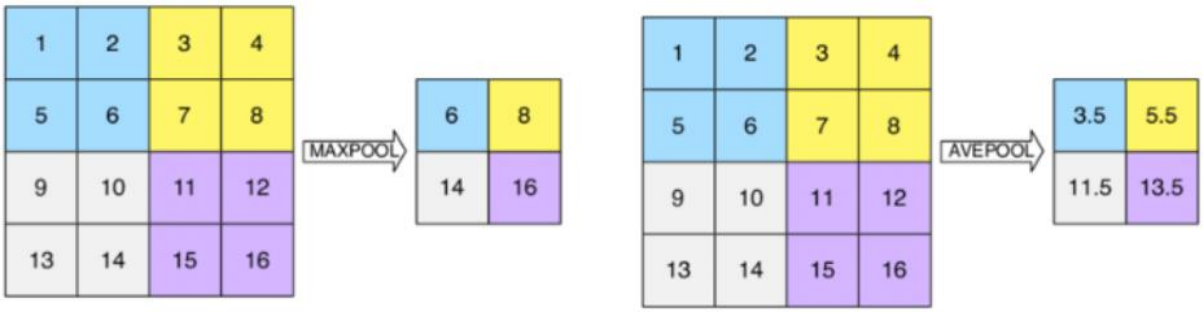


Figura 8. Proceso de pooling. (Durán, 2017)

- **Capa full-connected o de clasificación**

Es la capa final de una CNN. Tiene como fin clasificar y determinar a qué clase pertenece la imagen de entrada. A la salida de esta capa se obtendrá un vector donde cada componente representa la probabilidad que tiene cada píxel de la imagen de entrada de pertenecer a una determinada clase.

2.2.5. Algoritmos de segmentación semántica

La segmentación semántica son algoritmos basados en aprendizaje profundo en el cual a cada píxel de una imagen asocia una clase o etiqueta. Son muy útiles en aplicaciones de detección

de objetos, control de calidad industrial, conducción automática en la identificación de peatones, vías, aceras, segmentación de imágenes satelitales, identificación de estructuras en imágenes médicas, identificación de objetos y áreas aplicadas a sistemas robóticos.

El principal problema presente en la segmentación semántica es clasificar a cada píxel de una imagen dentro de una clase, y no toda la imagen en una categoría. La mayoría de las técnicas para dar solución a este problema están basadas en redes neuronales convolucionales. En el trabajo presentado por Méndez (2019) nos muestra las principales redes neuronales:

1. AlexNet: CNN de la Universidad de Toronto. Ganadora del ImageNet Large Scale Visual Recognition Challenge de 2012 con una precisión del 84,6%.
2. VGG-16: CNN de la Universidad de Oxford, la cual quedó segunda en el concurso ImageNet 2014 con una precisión del 92,7%.
3. GoogLeNet: CNN perteneciente a google también conocida como Inception, ganadora del ImageNet de 2014 con una precisión del 93,3%,
4. ResNet: CNN perteneciente a Microsoft que ganó el concurso ImageNet en 2015 con una precisión del 96,4%.

La arquitectura generalmente utilizada para la segmentación semántica es implementada con una etapa de codificación, seguida de una etapa de decodificación. La parte de codificador es la encargada de extraer las características de la imagen y produce las representaciones en baja resolución. En todas estas arquitecturas se utiliza como codificador a una red convolucional de clasificación. Por otra parte, el decodificador se encarga de recuperar los detalles de los objetos, proyectando las características extraídas para producir características multidimensionales para cada píxel. En la etapa de decodificación varían los tipos de redes convolucionales utilizadas, pero todas ellas buscan llegar al único objetivo de generar como salida una imagen de la misma dimensión que la imagen de entrada. Aquí radica las líneas de investigación y los diferentes estudios que se han realizado. Méndez (2019) muestra las principales arquitecturas estudiadas actualmente como lo son SegNet, FCN, Dilated Convolutions, DeepLabv1, U-Net, FPN, PSPNet, DeepLabv2 y RefineNet.

2.2.6. SegNet

En este trabajo se ha seleccionado e implementado un modelo basado en SegNet (Badrinarayanan et al., 2015) tras ver las ventajas presentadas por sus autores. Entre las principales ventajas que describen se encuentran que es el primer método de aprendizaje profundo que mapea matrices de características de baja resolución a etiquetas semánticas, Generan una precisión de calidad tanto cualitativa como cuantitativa al analizar escenas de interiores y exteriores inclusive sin el uso de post-procesamiento. Por último, han ejecutado pruebas con diversos conjuntos de datos mostrando un coste computacional no tan considerable. En la Figura 9 se observa el estudio comparativo con varios modelos realizado por Badrinarayanan et al. (2015). Muestra una mejora en la precisión de segmentación de clases con un índice elevado de dificultad como en automóviles, peatones y postes. El promedio global es el más elevado con respecto a otros modelos.

Method	Building	Tree	Sky	Car	Sign-Symbol	Road	Pedestrian	Fence	Column-Pole	Sidewalk	Bicyclist	Class avg.	Global avg.
SfM+Appearance [2]	46.2	61.9	89.7	68.6	42.9	89.5	53.6	46.6	0.7	60.5	22.5	53.0	69.1
Boosting [36]	61.9	67.3	91.1	71.1	**58.5**	92.9	49.5	37.6	25.8	77.8	24.7	59.8	76.4
Dense Depth Maps [43]	85.3	57.3	95.4	69.2	46.5	**98.5**	23.8	44.3	22.0	38.1	28.7	55.4	82.1
Structured Random Forests [18]	not available											51.4	72.5
Neural Decision Forests [3]	not available											56.1	82.1
Local Label Descriptors [40]	80.7	61.5	88.8	16.4	n/a	98.0	1.09	0.05	4.13	12.4	0.07	36.3	73.6
Super Parsing [39]	**87.0**	67.1	96.9	62.7	30.1	95.9	14.7	17.9	1.7	70.0	19.4	51.2	83.3
SegNet - 4 layer	75.0	**84.6**	91.2	**82.7**	36.9	93.3	**55.0**	37.5	**44.8**	74.1	16.0	**62.9**	**84.3**
Boosting + pairwise CRF [36]	70.7	70.8	94.7	74.4	55.9	94.1	45.7	37.2	13.0	79.3	23.1	59.9	79.8
Boosting+Higher order [36]	84.5	72.6	**97.5**	72.7	34.1	95.3	34.2	45.7	8.1	77.6	28.5	59.2	83.8
Boosting+Detectors+CRF [20]	81.5	76.6	96.2	78.7	40.2	93.9	43.0	**47.6**	14.3	**81.5**	**33.9**	62.5	83.8

Figura 9. Comparativa de SegNet con otros modelos de segmentación semántica. (Badrinarayanan et al., 2015)

SegNet es una arquitectura de una red neuronal basada en aprendizaje profundo basada en un codificador-decodificador creada por miembros de la Universidad de Cambridge, Reino Unido. Se utiliza especialmente para la clasificación de píxeles dentro de una imagen (Badrinarayanan et al., 2016).

Una red SegNet consta de una red de codificación seguida de una red de decodificación y finalmente una etapa de clasificación de píxeles.

La red de codificador utiliza una arquitectura de una red VGG16 la cual consta de trece capas de convolución. Cada capa de codificador tiene una capa de decodificador correspondiente y, por tanto, la red de decodificador tiene 13 capas. La salida final del decodificador pasa a un clasificador softmax para generar las probabilidades de que un píxel corresponda a una clase determinada. Toda la arquitectura se puede entrenar de un extremo a otro mediante el descenso de gradiente estocástico.

En la Figura 10 se puede ver la estructura de una red SegNet.

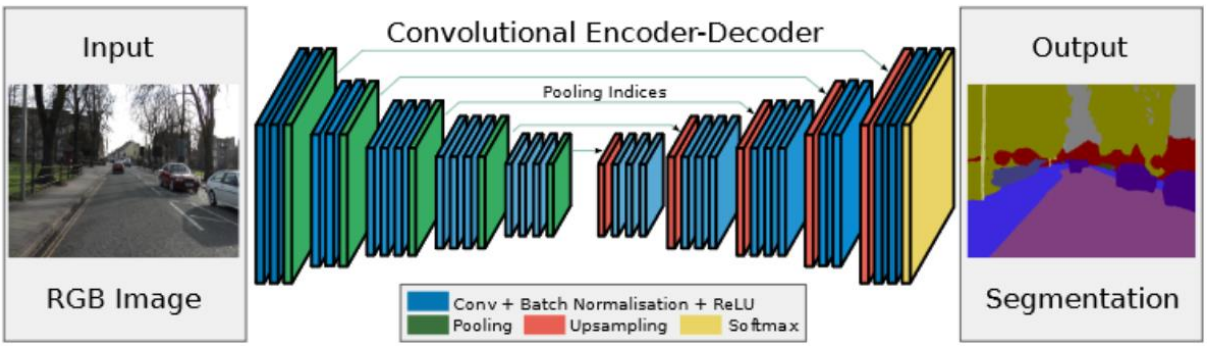


Figura 10. Arquitectura de SegNet. (Badrinarayanan et al., 2016)

La red de codificación se encuentra formada por varias redes de convolución, que por medio de filtros nos generan los mapas de características. A continuación, pasa por una red de normalización para entrar a una función de activación del tipo ReLu. Posteriormente, pasa a una etapa de pooling con una ventana de 2x2 para después, realizar un sub muestreo por un factor de 2.

La red de decodificación tiene como entrada un mapa de características, para luego, aplicar un proceso de convolución con determinados filtros de decodificación para producir mapas de características más densos.

2.3. Estudios similares

El aprendizaje profundo es una de las ramas que se han venido desarrollando en los últimos años y con un gran crecimiento. Gracias a los prometedores resultados que se han obtenido y a la gran cantidad de información que se puede obtener, tienen múltiples aplicaciones en diversas áreas. La segmentación de imágenes utilizada para la extracción de características se ha convertido en una de las áreas más relevantes de investigación por lo que se han estudiado y generado una gran cantidad de algoritmos y técnicas.

A continuación, podemos encontrar varios estudios realizados empleando algoritmos de segmentación semántica aplicados para la clasificación de píxeles, por lo que podemos resaltar los aspectos más importantes y realizar una comparativa con el trabajo que presentamos:

Trabajo 1: Análisis de técnicas de segmentación semántica sobre imágenes aéreas con deeplabv3+ (García, 2019).

Trabajo 2: Agricultura de Precisión: Preprocesamiento y Segmentación de Imágenes para Obtención de una Ruta de Navegación Autónoma Terrestre (Moreano et al., 2020).

Trabajo 3: Algoritmos de segmentación semántica para anotación de imágenes (Toro, 2019).

Trabajo 4: Generación de ruta óptima para robots móviles a partir de segmentación de imágenes (Montiel et al., 2015).

Trabajo 5: Towards semantic segmentation of orthophoto images using graph-based community identification (Moujahid et al., 2019).

El principal objetivo de nuestro trabajo es la aplicación de un algoritmo de segmentación semántica aplicada sobre imágenes aéreas para la detección de objetos para poder generar rutas libres de obstáculos. Realizamos una extracción de las principales características que se encuentran presentes o no dentro de cada uno de los trabajos más relevantes.

García (2019) propone la implementación del algoritmo Deeplabv3+ utilizando el lenguaje de programación Python, basados en los modelos Tensorflow y Keras para segmentar imágenes mediante el uso de varios datasets de imágenes aéreas y no aéreas que contienen la información *groundtruth* con sus respectivas máscaras. Su objetivo es abordar los problemas que conlleva este proceso en imágenes de distintos datasets con distintas resoluciones.

Nuestra propuesta propone la implementación de un algoritmo de Deep learning o aprendizaje profundo en Matlab, generando nuestro propio data set que nos permita tener más control sobre las áreas y objetos de interés que deseemos segmentar, con el fin de obtener una ruta libre de obstáculos.

Al utilizar la segmentación semántica lo que realmente hacemos es etiquetar cada píxel de una imagen en diferentes categorías lo que permite discriminar unas de otras, lo que hace un método más preciso al momento de detectar objetos. Moreano et al. (2020) utiliza un método de segmentación basado en umbrales de color para la obtención de una máscara binaria para el reconocimiento de cultivo con el objetivo de trazar una trayectoria y conseguir la navegación autónoma terrestre de una máquina agrícola. Al aplicar esta técnica es necesario un alto coste computacional ya que al obtener las marcas de navegación en tiempo real es necesario el procesado y segmentado de varios frames por minuto según va avanzando la maquinaria. Dicha técnica mejoraría tanto en coste computacional como en precisión de detección de áreas de cultivo aplicando el algoritmo de segmentación semántica propuesto en nuestro trabajo, aplicado a una imagen aérea de la parcela de cultivo con lo que logramos el reconocimiento del cultivo y generación de la ruta completa que puede ser transmitida a la máquina agrícola para el seguimiento.

La extracción de características de una imagen para la discriminación de objetos de diferente naturaleza implica un problema , por lo que categorizar secciones de una imagen es una tarea compleja como lo indica Toro (2019), por lo que propone el “desarrollo y aplicación de diferentes algoritmos para abordar problemas relacionados con la anotación automática de imágenes, trabajando a un nivel de particionado de la imagen inferior al de una segmentación, una sobre- segmentación”, aplicados a la implementación en imágenes médicas y a imágenes ópticas de teledetección. El análisis de estas técnicas de anotación automática de imágenes nos ayuda como base para nuestro objetivo de segmentar imágenes aéreas urbanas para sacar características en un escenario real y posterior lograr generar una ruta libre de obstáculo.

Montiel et al. (2015) propone la generación de rutas óptimas de navegación para robots en ambientes estáticos con un bajo coste computacional, aplicando algoritmos de segmentación de imágenes a partir de obstáculos geométricos estáticos. El algoritmo de segmentación utilizado se basa en una segmentación por umbral de color y posterior la aplicación de un algoritmo de búsqueda de una ruta libre de obstáculos para la movilidad de un robot móvil.

Para dar una aplicación en ambientes reales se propone en nuestro trabajo la generación de una ruta libre de obstáculos a partir de imágenes aéreas reales de mayor resolución espacial y se implementará un algoritmo de segmentación semántica basado en redes neuronales convolucionales, con lo que se obtendrá una mayor precisión en la detección de obstáculos con lo que obtendremos un camino transitable.

Mediante la implementación de un framework desarrollado por Moujahid et al. (2019), donde permite la detección automática de objetos de interés en imágenes sin un entrenamiento previo, que facilita la cantidad de cálculos y con muy buenos resultados. El procesa es basado en una sobre segmentación de las imágenes, se buscan descriptores y regiones similares para proceder a la implementación de algoritmos de reconstrucción de gráficos y finalmente se realiza la segmentación en clústeres, nuestro proyecto abarcará la segmentación semántica basada en el entrenamiento de una red de segmentación semántica para clasificar imágenes basados en analizar un conjunto de imágenes y etiquetar los píxeles, posterior creamos una red de segmentación semántica para entrenar y clasificar imágenes en categorías o clústeres y finalmente haremos la evaluación, pruebas, medición de la precisión de la segmentación.

En la Tabla 2 se resume el análisis de cada uno de los trabajos relevantes, extrayendo las principales características que sirven para comparación con este trabajo. Entre los parámetros a considerar en cada uno de los trabajos están si se describe o no un método de segmentación semántica, si se implementa en código de programación algún tipo de algoritmo, si se estudia y aplica el método sobre un conjunto de imágenes aéreas y si al final se logra obtener una ruta libre de circulación.

Tabla 2. Comparación con trabajos relevantes

	Método de segmentación semántica	Implementación del algoritmo	Aplicación sobre imágenes aéreas	Obtención de rutas libres
Trabajo 1	Sí	Si	Sí	No
Trabajo 2	No	Si	No	Si
Trabajo 3	Si	Si	No	No
Trabajo 4	No	No	Si	Sí
Trabajo 5	Si	Si	Si	No
Nuestra propuesta	Sí	Si	Sí	Sí

(Elaboración propia)

Como conclusión, nuestro trabajo busca la implementación de un modelo de segmentación semántica sobre un conjunto de imágenes aéreas. Comenzaremos con el análisis del método elegido y la implementación del algoritmo. A continuación, se generarán rutas libres de obstáculos y finalmente, se simulará la circulación de un objeto móvil.

3. Objetivos y metodología del trabajo

3.1. Objetivo general

El objetivo general de este trabajo es implementar un algoritmo de segmentación semántica de imágenes basado en una arquitectura SegNet, con la finalidad de extraer de manera automática las características en imágenes urbanas aéreas y generar rutas libres de circulación para objetos móviles.

3.2. Objetivos específicos

La consecución del objetivo general de nuestro trabajo conlleva a cumplir con los siguientes objetivos específicos:

- Explorar diferentes técnicas de segmentación automática de imágenes, analizando la aplicabilidad de algoritmos de segmentación semántica para la extracción de características de una imagen.
- Generar un banco de imágenes urbanas aéreas y generar un conjunto de imágenes (data set) para la implementación de un algoritmo de segmentación semántica basada en aprendizaje profundo.
- Aplicar técnicas de pre-procesamiento sobre data set de imágenes para generar imágenes con una mejora de calidad, enfoque, distribución de intensidades, eliminación de ruido, etc.
- Desarrollar y aplicar un algoritmo de segmentación semántica basada en el aprendizaje profundo sobre las imágenes pre-procesadas con el fin de categorizar cada uno de los píxeles y poder identificar segmentos de imágenes que no tengan obstáculos y que sean transitables.
- Analizar los resultados obtenidos y probar con nuevas imágenes para validar el modelo de clasificación y segmentación.

3.3. Metodología del trabajo

En la presente sección se detalla la metodología ejecutada en este trabajo para extraer características de una imagen con el objetivo de asociar una determinada clase, etiqueta o categoría a cada píxel presente en una imagen, por consiguiente, establecer las secciones en una imagen aérea que pueden ser áreas libres de obstáculos o vías transitables.

La metodología desarrollada consiste en las siguientes fases:

1. Recopilación, selección e integración de imágenes aéreas.
2. Pre-procesamiento de imágenes.
3. Generación de un conjunto de imágenes originales y un conjunto de imágenes etiquetadas.
4. Aplicación del algoritmo de segmentación semántica.
5. Evaluación de la precisión y análisis de los resultados.

A continuación, se describe cada fase de la metodología.

3.3.1. Recopilación, selección e integración de imágenes aéreas.

En esta fase de recopilación de imágenes de pruebas, se ha efectuado una búsqueda de diferentes datasets que pueden ser aplicados a nuestro propósito. Para fines más didácticos y poder ejecutar nuestro trabajo basados en una aplicación en específico, seleccionaremos estratégicamente un conjunto de imágenes aéreas. Estas imágenes las organizamos e integramos para formar el conjunto de datos de entrada que será el punto de partida.

3.3.2. Pre-procesamiento de imágenes

En esta etapa cubre todos los métodos aplicados al conjunto de datos de entrada, con el fin de mejorar, realzar ciertas características y eliminar falla del conjunto de imágenes. Los métodos aplicados se basan en los vistos previamente en la literatura, entre los que podemos recordar: filtrar, enfocar, aclarar, recortar, redimensionar y redistribuir los niveles de intensidad para mejorar las imágenes.

Como salida de esta fase obtendremos un conjunto de imágenes procesadas con mejores características que nos faciliten la ejecución de las etapas posteriores.

3.3.3. Generación de conjunto de imágenes originales y conjunto de imágenes etiquetadas.

Tras procesar y mejorar las características de las imágenes, podemos pasar a la etapa de etiquetado. Una opción posible es descargar datos etiquetados en Internet ya que se necesita una gran cantidad de muestras para poder ejecutar modelos de aprendizaje profundo.

Como se mencionó anteriormente, el presente proyecto busca ejecutar el modelo en una aplicación en específico por lo que esta etapa lo realizaremos esta etapa de manera manual, con la ayuda de paquetes de software en MATLAB.

La aplicación Image Labeler de MATLAB nos permite realizar la clasificación de cada píxel en diferentes categorías de una manera muy dinámica y fácil con lo que nos permitirá tener el control de las clases que queremos discernir en una imagen.

Una vez generado el conjunto de imágenes etiquetadas procedemos a consolidarlos en una datastore que contendrá la ubicación de cada una de las imágenes. Tendremos dos conjuntos claramente identificados: datastore de imágenes originales y una datastore de imágenes etiquetadas.

3.3.4. Aplicación del algoritmo de segmentación semántica.

En esta fase aplicamos el algoritmo de segmentación semántica, es decir, se seleccionan y aplican las técnicas de segmentación semántica que sean pertinentes y se establecen los parámetros a valores óptimos.

Como revisamos en la literatura, un algoritmo de segmentación semántica está basado en un modelo de red artificial convolucional, por lo que procederemos a cargar una red previamente entrenada, que puede ser una red VGG16, U-Net, etc. Posterior, creamos la arquitectura SegNet basada en codificador-decodificador.

Una vez creado el modelo procedemos al entrenamiento de la red con el conjunto de datos de entrenamiento. Este proceso conlleva tiempo y un uso elevado de recursos computacionales debido a la magnitud elevada de datos que se deben procesar y analizar.

3.3.5. Evaluación de la precisión y análisis de los resultados.

Finalmente, en la última etapa se conoce el rendimiento de la red con cualquier imagen de entrada y podemos medir la precisión de la segmentación. Tendremos un conjunto de datos de validación con los que evaluaremos la fidelidad de nuestro trabajo y verificaremos si se ha alcanzado el objetivo de poder clasificar las áreas de una imagen en diferentes categorías. Una de las categorías de salida será secciones de la imagen donde se puede evidenciar vías transitables sin obstáculos.

3.4. Identificación de requisitos

Con el objetivo de facilitar la calidad de vida de las personas la tecnología avanza a pasos agigantados. Idealmente, por medio de la ciencia, la tecnología busca replicar el comportamiento de los seres humanos por medio de herramientas de software, hardware, sistemas de cómputo avanzados, etc.

Extraer información de una imagen es un problema complejo y con un sin número de aplicaciones en la vida real que utilizan la misma problemática, como pueden ser: movilidad autónoma de sistemas robóticos, seguimientos de áreas de deforestación, identificación de catástrofes naturales, identificación de incendios forestales, etc.

El problema en que nos enfocamos resolver consiste en determinar una ruta libre de obstáculos para la libre movilidad de un robot, un dron, o cualquier vehículo autónomo en un entorno por medio de la identificación de patrones, objetos, dentro de una imagen que será nuestro entorno a circular. Para que este proceso llegue a ser exitoso es necesario aplicar una técnica de segmentación adecuada. Como solución a esta problemática proponemos la implementación de un algoritmo de segmentación semántica basado en el aprendizaje profundo para la clasificación de píxeles de la imagen de una manera autónoma con el menor índice de error.

Existen varias fuentes donde se pueden obtener una gran cantidad de imágenes aéreas. Entre los principales artefactos utilizados para la obtención de estas imágenes, podemos encontrar los satélites y, más recientemente y con un rápido crecimiento aplicativo, los drones. De estos dispositivos se puede obtener una gran cantidad de datos, pero los mismos sin un procesamiento no tendrían sentido. Este es el primer punto donde inicia nuestro trabajo, ya

que, tras adquirir las imágenes de cualquiera de las fuentes, es necesario aplicar varias técnicas que nos ayuden a generar información de valor. Por este motivo, es necesario un proceso de pre-procesamiento de imágenes que no es más que la aplicación de técnicas matemáticas que permitan el realce de las ciertas características de interés.

Al aplicar una segmentación semántica, intrínsecamente es necesario la aplicación de conceptos matemáticos avanzados como convolución de matrices, minimización de funciones de coste o de error, aplicación de filtros y conocimiento de redes neuronales. Se ha optado por la aplicación de una técnica de segmentación semántica basada en una arquitectura codificador-decodificador (SegNet), fundamentada en redes neuronales convolucionales, especialmente, por el alto índice de fiabilidad al momento de generar resultados y ya que son redes adaptadas especialmente para el procesamiento de imágenes debido a las grandes dimensiones espaciales y la gran cantidad de datos que se deben procesar.

4. Descripción del modelo y resultados

En este capítulo se detalla cada una de las fases ejecutadas, detallando los procedimientos y métodos empleados para la implementación del algoritmo de segmentación semántica sobre imágenes aéreas con el fin de generar rutas libres de obstáculos.

4.1. Recopilación, selección e integración de imágenes aéreas

El conjunto de datos de entrada necesarios para la ejecución de este proyecto son las imágenes aéreas, ya sean de fuentes satelitales o de vehículos no tripulados como drones.

Existen varias fuentes de datos o datasets de imágenes satelitales entre las que podemos nombrar: iSAID *(IIAI & Wuhan University, Dec 2019),* SpaceNet 7 *(CosmiQ Works, Planet, Aug 2020),* RarePlanes *(CosmiQ Works, A.I.Reverie, June 2020),* Stanford Drone Data *(Stanford University, Oct 2016).* Estas fuentes ponen a disposición para el público un gran conjunto de imágenes listas para aplicar algoritmos de inteligencia artificial para diferentes aplicaciones.

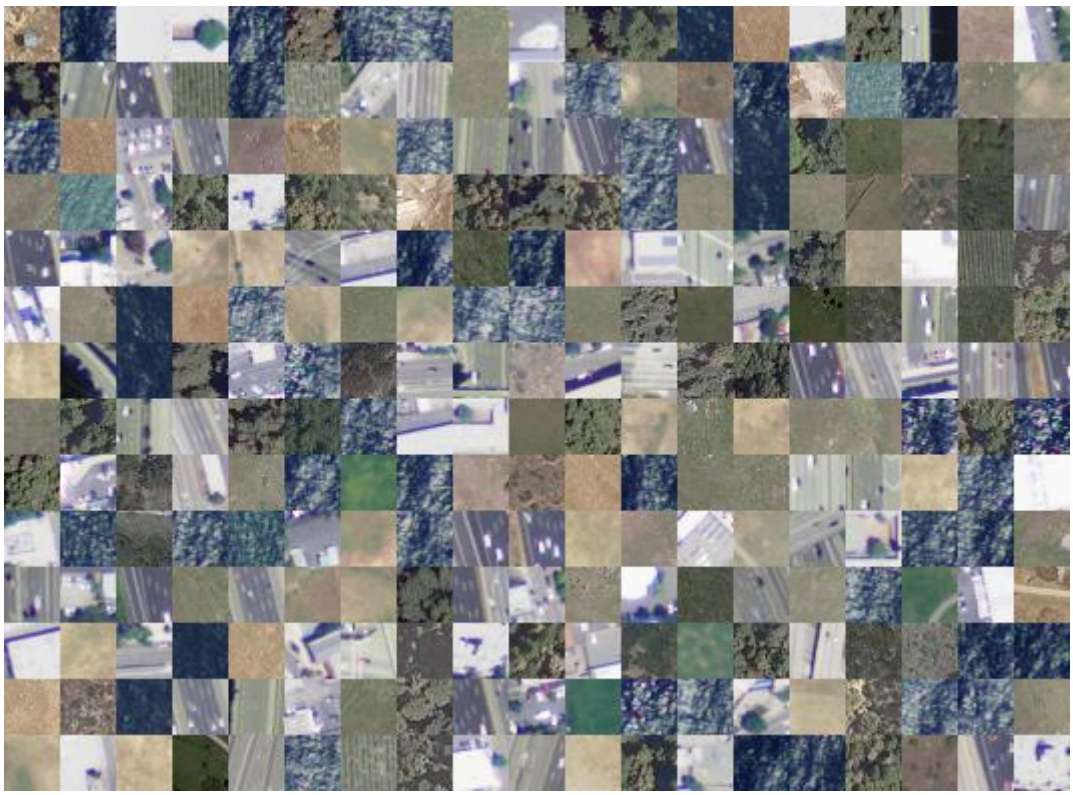


Figura 11. Imágenes de muestra dataset SAT-4 and SAT-6. (Kang et al., 2018)

Nuestro objetivo es diseñar una metodología para generar un modelo para segmentar semánticamente una imagen aérea, motivo por el cual buscamos abordar todas las fases necesarias para lograrlo, desde la obtención de las imágenes, pasando por el etiquetado, el procesamiento digital y finalmente, la aplicación de la técnica de segmentación. Para ello, se ha hecho una búsqueda de varias fuentes de fotografías a partir de la web, especialmente de dos fuentes:

- OpenAerialMap (OAM): conjunto de herramientas para buscar, compartir y usar imágenes de satélites y vehículos aéreos no tripulados (UAV) con licencia abierta.
- Google Earth: herramienta de Google que permite visualizar múltiple cartografía y acceso a imágenes satelitales, mapas, relieves o edificaciones.

Posterior a la búsqueda, se ejecuta una selección de las imágenes más apropiadas. Con este fin, se genera un conjunto de 100 imágenes que posteriormente se dividirá en un conjunto de entrenamiento y un conjunto de validación.

En la siguiente Figura 12 se puede ver ejemplos de los distintos elementos que conforman el conjunto de imágenes aéreas que emplearemos en este trabajo.

Figura 12. Imágenes de muestra. (Elaboración propia)

Como se puede observar, el conjunto de imágenes está conformado por fotografías aéreas de una ciudad. Las categorías que tiene el conjunto de imágenes se resumen en la Tabla 3.

Tabla 3. Parámetros del conjunto de imágenes

Conjunto de datos:	100 imágenes
Dimensiones:	Superior a 2500 x 2500
Espacio de color:	RGB
Formato:	JPG

(Elaboración propia)

4.2. Pre-procesamiento de imágenes

Una vez seleccionadas el conjunto de imágenes que vamos a utilizar, se procede a ejecutar una mejora de las características de cada una de las imágenes para resaltar las propiedades de interés o para corregir características al momento de la adquisición.

Uno de los principales problemas al momento de adquirir una imagen digital es la agregación de ruido indeseado, es decir, que algunos valores de píxeles no reflejan las intensidades reales de la escena real. Para ello se aplican filtros que ayuden a minimizar los efectos de ruido indeseado.

Las imágenes obtenidas en la fase de selección presentan una mínima cantidad de ruido que intentaremos eliminar, por esta razón, se aplicará filtros a las imágenes convertidas a escala de grises.

A cada una de las imágenes se aplica un filtro adaptativo de Wiener, un filtro lineal que emplea mínimos cuadrados. Este filtro es más selectivo que un filtro lineal sobre todo por que conserva los bordes y otras partes de alta frecuencia de una imagen. MATLAB implementa

dicho filtro gracias a la función ***wiener2,*** donde filtra una imagen de entrada en escala de grises utilizando un filtro pasa bajo de píxeles del tipo Wiener**.**

En la Figura 13 se observa en la parte izquierda una porción de la imagen original en la cual se identifica una mínima cantidad de ruido presente. En la parte derecha, se observa la misma porción de la imagen aplicada el filtro, en consecuencia, se evidencia una mejora en la imagen.

Figura 13. Porción de una imagen con ruido y filtrada. (Elaboración propia)

Otra mejora que se aplica al conjunto de imágenes obtenidas es una corrección de la iluminación. Algunas imágenes muestran una poca cantidad de luz, lo que origina una degradación debido a las malas condiciones de iluminación o a artefactos ambientales como nubosidad. Esta falta de iluminación puede afectar el rendimiento del algoritmo de segmentación semántica, por consiguiente, es necesario mejorar la calidad de la imagen con déficit de luz para mejorar la interpretación de las características.

Para reducir la neblina y mejor las imágenes con déficit de luz se aplica métodos de procesamiento digital de imágenes. MATLAB implementa la función ***imreducehaze,*** que reduce la neblina atmosférica dando como parámetro de entrada de la función una imagen a color, además, se especifica la cantidad de neblina que se va a eliminar y el método que se aplicará para la eliminación.

Para ejecutar el método invertimos la imagen original, seguidamente, aplicamos la función de eliminación de ruido y finalmente, invertimos nuevamente a la imagen a escala de color y en consecuencia obtenemos la imagen mejorada. Como se muestra en la Figura 14, tenemos en la parte superior la imagen original seleccionada, en la parte inferior se observa la imagen generada una vez aplicado el procedimiento de mejora de iluminación.

Figura 14. Corrección de iluminación. (Elaboración propia)

A continuación, se realiza un estudio del espacio de color de las imágenes para observar si las imágenes tienen deficiencias en los contrastes o los colores están desequilibrados. Se escoge el espacio de color RGB para el análisis. Como se sabe, una imagen representado en el espacio

de color RGB, está representado por tres componentes: R (rojo), G (verde), B (azul). En las imágenes digitales, cada componente o banda tiene asignado un valor de 0 a 255 que corresponde al nivel de intensidad.

En la Figura 15 se evidencia como un píxel posee tres niveles de intensidades correspondiente a cada componente. En imágenes digitales, los valores de las intensidades se encuentran codificadas en valores de 0 a 255.

Figura 15. Niveles de intensidad de un píxel. (Elaboración propia)

La distribución de intensidades de cada componente de la imagen se puede observar en el histograma de cada banda. En la Figura 16 se observa los histogramas de las tres bandas (R, G, B) y que los niveles se concentran a lo largo del rango dinámico disponible, pero no se encuentran distribuidos para niveles bajos o cercanos a cero. Esto significa que se encuentran casi distribuidos uniformemente todos los niveles de intensidad a lo largo de todo el rango excepto en valores bajos. Este efecto es posible mejorarlo por medio de algoritmos de procesamiento de imágenes.

Figura 16. Histograma de componentes RGB. (Elaboración propia)

De forma adicional, podemos representar cómo se encuentran correlacionadas las bandas. En la Figura 17 se observa la gráfica de dispersión de las tres bandas de color donde se observa que están altamente correlacionadas dentro del espacio de color. A simple vista se observa que no existe una distinción clara de un color dentro de la imagen, es decir, se encuentran muy correlacionadas las tres bandas, con lo que podemos llegar a la conclusión de que no podemos aplicar un algoritmo de segmentación semántica aplicando niveles de umbrales. En otras palabras, si deseamos segmentar solamente las vías por medio de un umbral de intensidad, no lo podemos llevar a cabo, debido a que tienen similitud de intensidades con otras áreas que no corresponden a las vías lo que conllevaría a segmentar secciones erróneas, por lo que se hace necesario aplicar otro tipo de algoritmo más avanzado y acorde al problema que se intenta resolver.

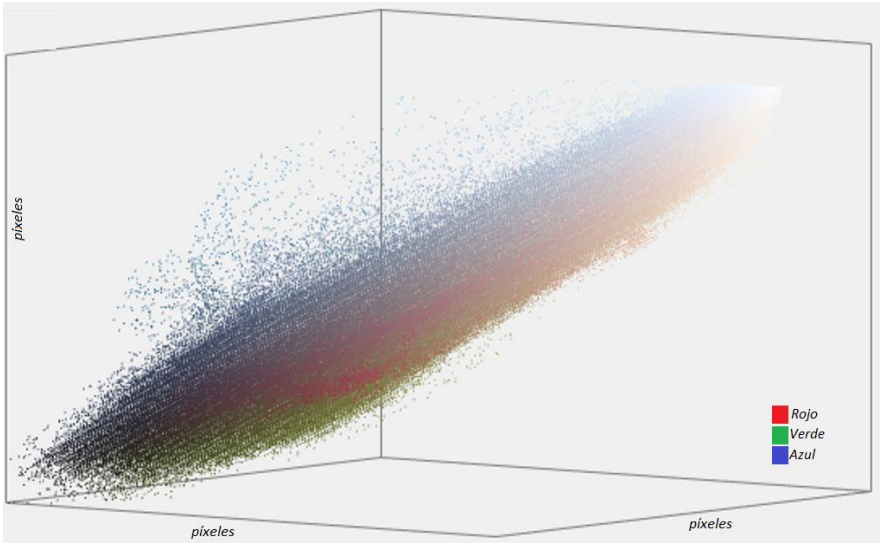


Figura 17. Dispersión de intensidades en el espacio de color. (Elaboración propia)

Otro aspecto que se observa en la Figura 17 es la dispersión de las tres bandas: rojo, verde y azul evidenciando que se encuentran altamente correlacionadas. Por tal motivo, se observa una tendencia de agrupación en el centro de la gráfica. Esta dispersión puede ser mejorada aplicando métodos de ecualización de histogramas, que permiten distribuir las intensidades a lo largo de todo el rango disponible en cada una de las bandas de color.

En la Figura 18 se evidencia la mejora obtenida en la imagen previamente filtrada y corregida la iluminación.

Figura 18. Ecualización de histograma. (Elaboración propia)

La mejor implementada al realizar la ecualización de intensidades se comprueba en el histograma de cada una de las bandas, ya que se observa como los niveles de intensidad se distribuyen en todo el rango dinámico disponible. A continuación, en la Figura 19 se visualiza el histograma de las tres bandas de color de la imagen mejorada en la cual se observa cómo ahora las intensidades van desde el nivel cero hasta el nivel más elevado que es el 255.

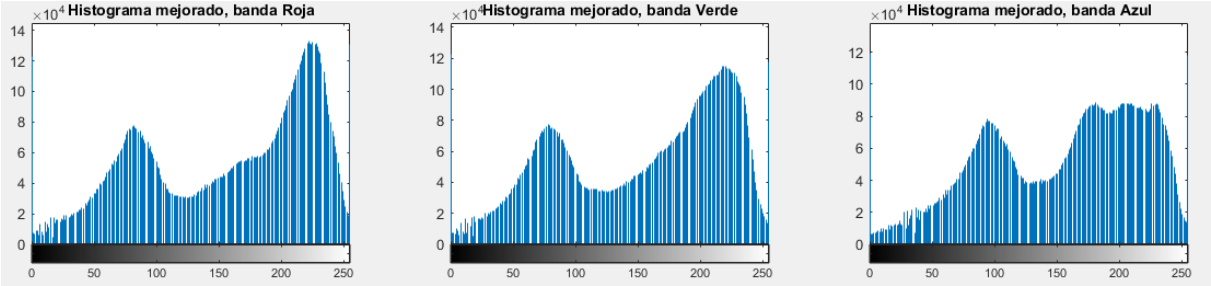


Figura 19. Histogramas de la imagen mejorada. (Elaboración propia)

Una vez establecidas las técnicas empleadas para la mejora de las imágenes, se aplica a todo el conjunto de imágenes seleccionadas en la etapa uno.

4.3. Generación de conjunto de imágenes originales y conjunto de imágenes etiquetadas

Finalizada la etapa de pre-procesamiento del conjunto de imágenes seleccionadas, se procede a agruparlas para conformar un conjunto al que denominaremos set de imágenes originales o dataset de imágenes originales. A este grupo lo dividimos en tres subconjuntos:

1. Set de imágenes de entrenamiento.
2. Set de imágenes de validación.
3. Set de imágenes de prueba.

Al grupo de imágenes de entrenamiento se las debe etiquetar para poder ejecutar el algoritmo de segmentación semántica. Existen datasets con datos etiquetados que pueden ser descargados de Internet. Como en el trabajo se genera nuestro propio conjunto de imágenes aéreas, éstas serán etiquetadas de manera manual.

El proceso de etiquetar datos consiste en etiquetar cada píxel de una imagen en una clase. Este conjunto de muestras etiquetadas pasará a la red neuronal para entrenarse y poder calcular los parámetros de la red.

La aplicación ***Image Labeler*** de MATLAB nos permite realizar este proceso de manera manual. En primer lugar, se procede a cargar todo el conjunto de datos de entrada para realizar el proceso de etiquetado. En nuestro conjunto vamos a definir cuatro clases o categorías mostradas en la Tabla 4 con la respectiva descripción de todos los objetos que serán incluidos en cada clase.

Tabla 4. Clases de etiquetas

CLASES	DESCRIPCIÓN
VÍAS	Área que puede ser transitable, caminos, carreteras
AUTOMÓVILES	Vehículos, carros, camiones
ÁRBOLES	Áreas verdes como parques, arbustos, árboles
CASAS	Toda edificación presente.

(Elaboración propia)

En cada imagen se procede a etiquetar cada píxel dentro de una categoría. Las etiquetas se pueden diferenciar con diferentes colores. En la Figura 20 se importa una imagen del conjunto de entrada y se muestra sobrepuesta con las cuatro etiquetas generadas. Las superficies que no tienen un color, indica que los píxeles no se encuentran etiquetados por lo que no son usados en el algoritmo de entrenamiento.

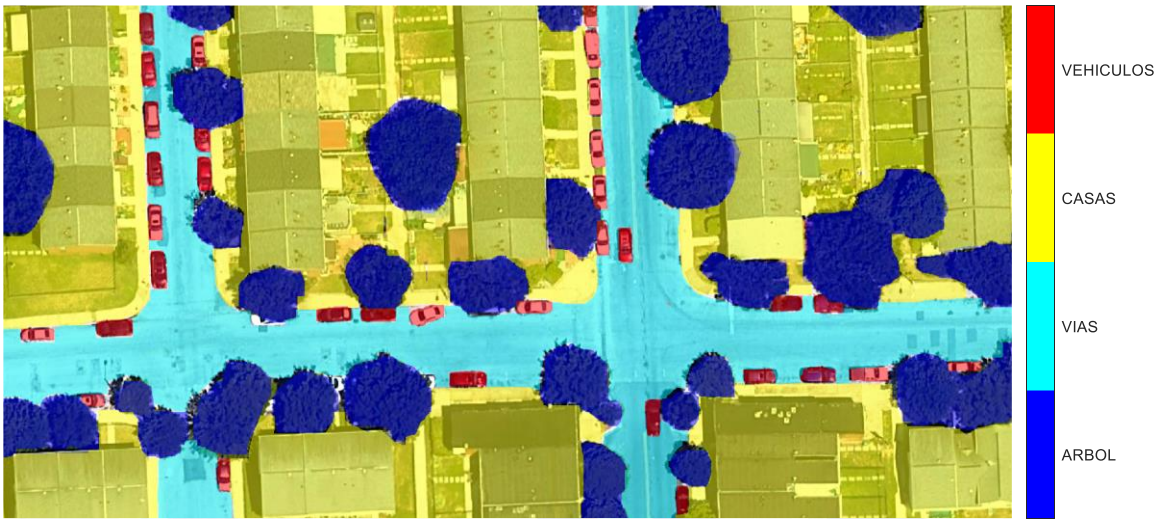


Figura 20. Etiquetado de píxeles. (Elaboración propia)

Como se puede observar, cada uno de los píxeles se encuentran clasificados en una de las categorías: VEHÍCULOS, CASAS, VÍAS, ÁRBOL, distinguidas por un color diferente. De igual manera, se adiciona las etiquetas a cada una de las imágenes seleccionadas. Como se muestra en la Figura 21, todas las muestras del conjunto de imágenes deben ser etiquetadas y clasificadas en las mismas clases establecidas a priori.

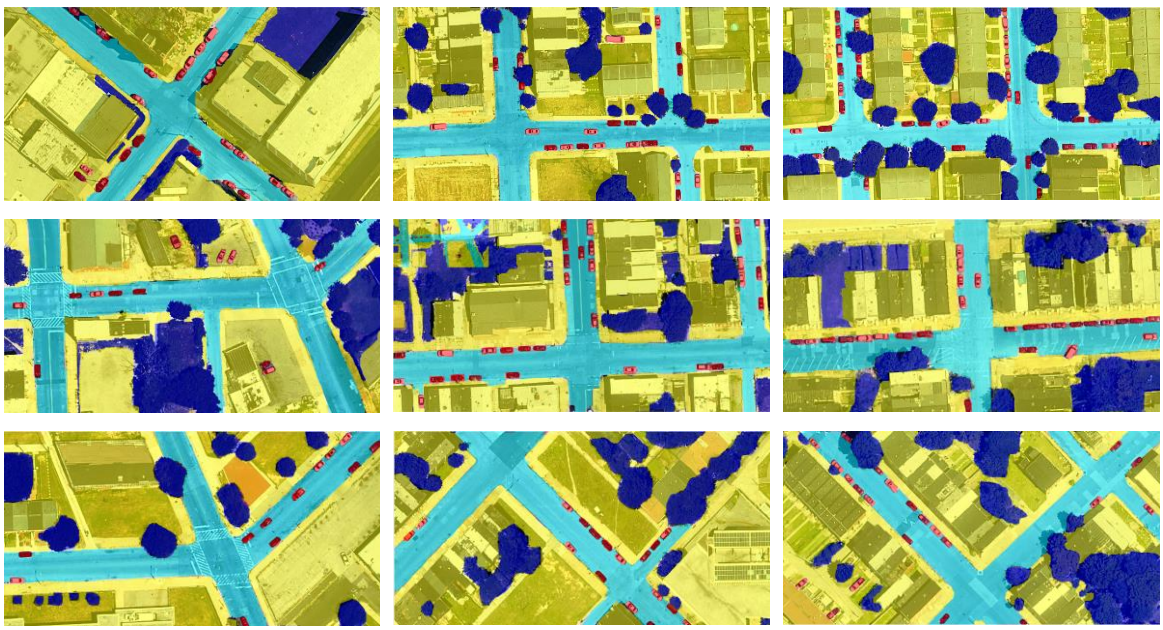


Figura 21. Conjunto de muestra, imágenes etiquetadas. (Elaboración propia)

Finalizando de etiquetar las imágenes, podemos ver como se distribuyen cada una de las clases dentro de cada una de las muestras. A continuación, se observa dentro de las primeras imágenes etiquetas como se distribuyen las 4 clases, siendo la más representativa las CASAS, seguida de las VÍAS y ÁRBOL y finalmente, con menor proporción se encuentran los VEHÍCULOS.

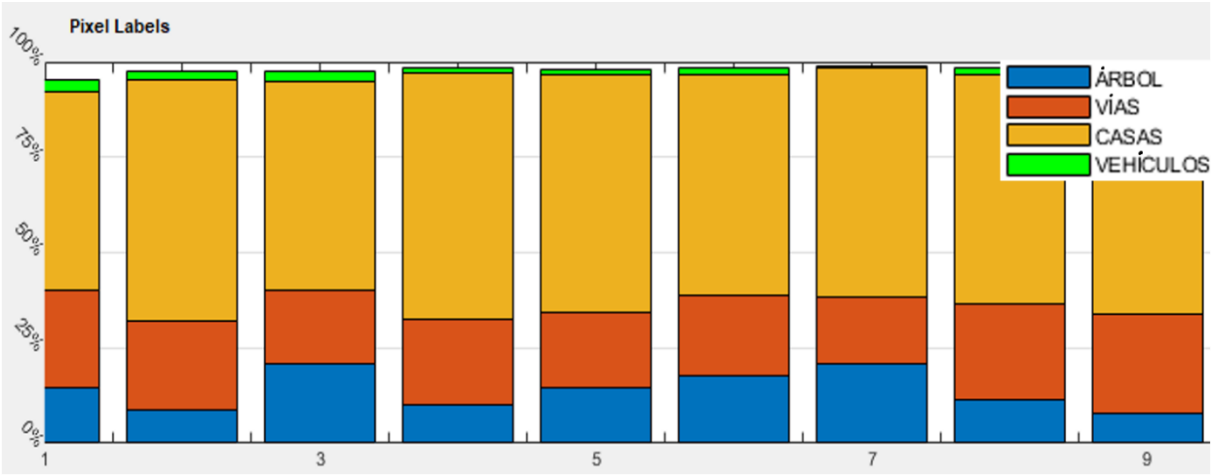


Figura 22. Distribución de las clases dentro de cada imagen. (Elaboración propia)

Idealmente, todas las clases deben tener las mismas proporciones y el mismo número de observaciones en cada una de las imágenes, en otras palabras, se han definido cuatro clases diferentes, entonces, lo recomendable es que aparezcan en todo el conjunto de datos en la misma proporción. Sin embargo, se observa cómo las clases están un poco desequilibradas en la frecuencia en que aparecen. Si no se maneja correctamente, este desequilibrio puede ser perjudicial para el proceso de aprendizaje dado que el aprendizaje estará sesgado a favor de las clases dominantes. Este comportamiento lo observamos en la Figura 23 mediante un histograma de frecuencias.

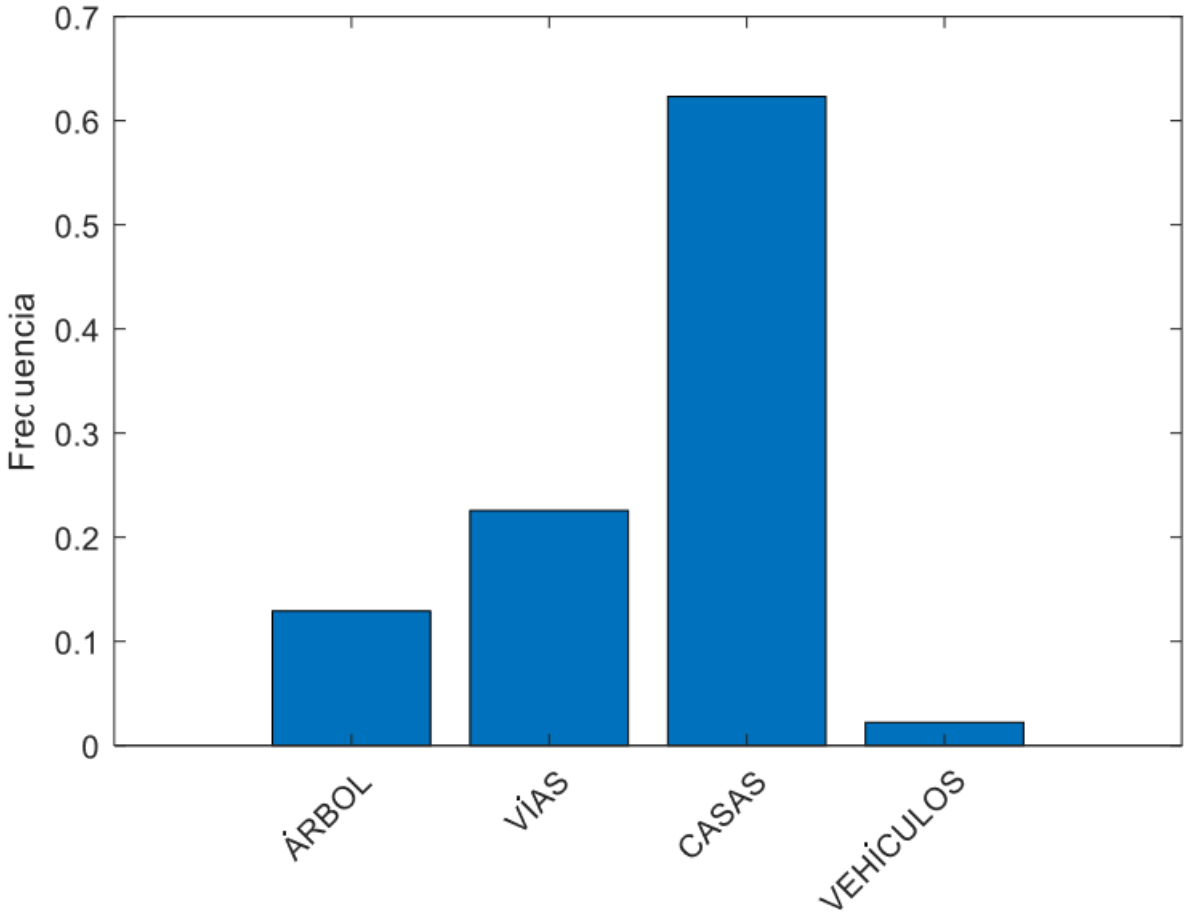


Figura 23. Histograma clases vs frecuencia. (Elaboración propia)

Debido a la gran cantidad de datos generados conjuntamente entre las imágenes originales y las imágenes etiquetadas, es necesario gestionarlos por medio de un almacén de datos o comúnmente llamados *datastore,* el cual contendrá la referencia a la ubicación exacta a cada archivo y solo se carga a memoria cuando se necesite operar sobre este archivo.

Para aplicar el algoritmo de segmentación semántica se debe generar dos datastore:

1. Originales_Datastore: contiene todas las imágenes originales.
2. Etiquetas_Datastore: contiene las imágenes con sus respectivas etiquetas.

En MATLAB generamos un datastore con la función ***imageDatastore,*** en la cual especificamos como parámetro de entrada el directorio donde se encuentren los archivos. Adicionalmente, podemos definir las propiedades que describen los datos y se especifica la manera en que se van a leer los datos.

4.4. Aplicación de la segmentación semántica

En este Trabajo Fin de Máster se aplica una red de segmentación semántica SegNet, basada en una arquitectura codificador-decodificador. Una red SegNet se compone de una etapa de codificadores seguida de otra etapa de decodificadores que pasan a una capa de clasificación de píxeles.

En la etapa de codificación o conocida como *downsampling,* a partir de una imagen de entrada, se genera un mapa de características de baja resolución. Seguida a la etapa de codificación, viene la etapa de decodificación o conocida como *upsampling* que nos aumenta la dimensión espacial del mapa de características generado en el codificador con el fin de obtener una imagen con las mismas dimensiones que la imagen de entrada.

4.4.1. Arquitectura

A partir de una red convolucional previamente entrenada, creamos la arquitectura de codificador-decodificador necesaria para etiquetar cada píxel en su respectiva categoría.

Se selecciona la red neuronal convolucional VGGNet-16 (Simonyan & Zisserman, 2015), basándonos en los buenos resultados obtenidos en los trabajos de Chanampe et al. (2019) y Kang et al. (2018). Esta red muestra buenos resultados en la segmentación de imágenes aéreas ya que, debido a que no es una red tan profunda, no es necesario un gran uso de recursos de cómputo para el entrenamiento lo que se ajusta a nuestra disponibilidad. Adicional, la red VGGNet-16 es considerada actualmente la mejor red para extracción de características en imágenes.

En la Figura 24 podemos observar la estructura de la red VGGNet-16. En ella se observa que la capa de entrada está ajustada para imágenes de dimensión 224x224x3, adicionalmente consta de 13 capas convolucionales, 5 capas de pooling, 3 capas completamente conectadas (FC) y finalmente la última capa softmax para la clasificación de píxeles en sus respectivas categorías.

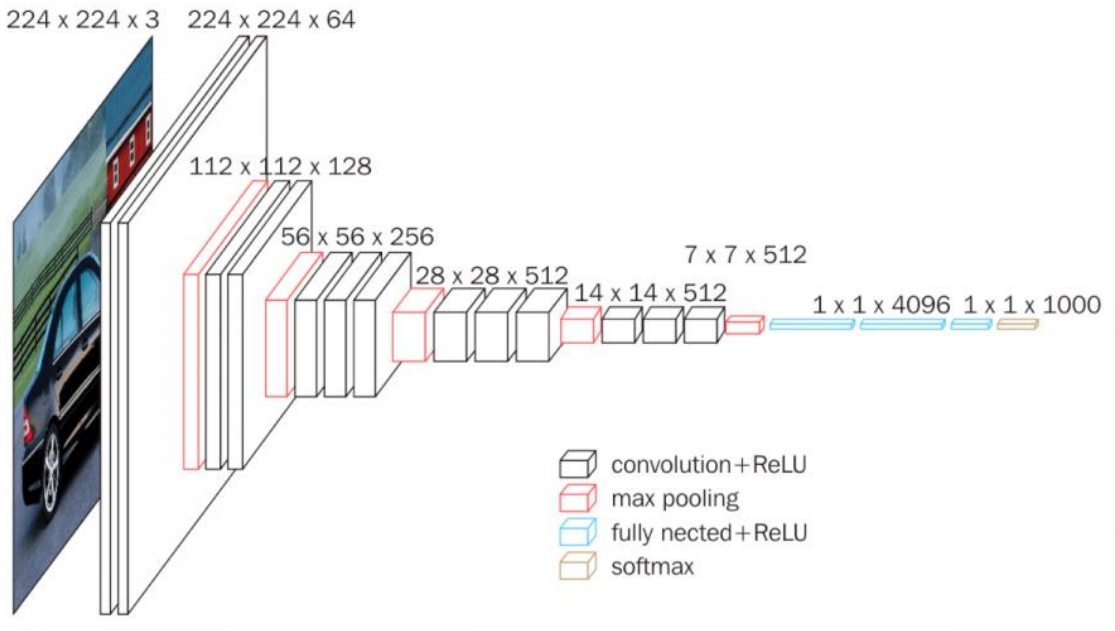


Figura 24. Arquitectura red VGGet 16. (Simonyan & Zisserman, 2015)

Para implementar el algoritmo de segmentación semántica ejecutamos el comando en MATLAB:

SegNetLayers(tamañanoEntrada,numClases,'vgg16')

en donde generamos una arquitectura *SegNet* basada en una etapa de codificación seguida de una etapa de decodificación, en donde especificamos como parámetros de entrada: tamaño de las imágenes de entrada, número de clases y modelo de red neuronal previamente descargada. Para la aplicación al dataset de imágenes de entrada se deben modificar las dimensiones de las imágenes para ajustarlas a una dimensión de 224x224x3 correspondientes a 224x224 píxeles, que corresponde a la capa de entrada de una red neuronal convolucional del tipo VGG16. El número de clases está especificado en cuatro correspondientes a las clases: VÍAS, CASAS, VEHÍCULOS, ÁRBOL.

En la Figura 25 se ilustra la correspondencia de cada etapa de codificación o downsampling con una etapa de decodificación o upsampling. Al finalizar se obtiene una arquitectura con 91 capas combinadas entre convolucionales, de pooling, capas de activación relu, etc.

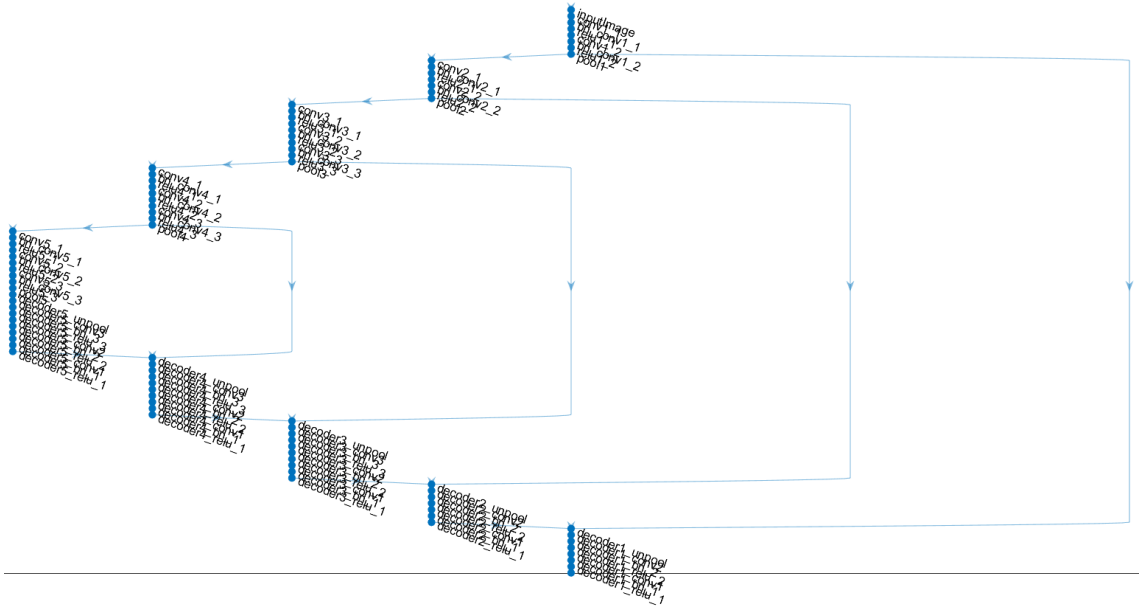


Figura 25. Arquitectura SegNet. (Elaboración propia)

4.4.2. Entrenamiento del modelo de segmentación semántica

Una vez definidos los dos conjuntos de imágenes, Originales_Datastore y Etiquetas_Datastore, podemos pasar a la etapa de entrenamiento del modelo de segmentación semántica previamente definido. El proceso de entrenar el modelo de segmentación semántica basada en una red SegNet reside en calcular y ajustar cada uno de los pesos de las entradas de cada nodo o neurona de la red neuronal, para minimizar una función de coste o pérdida que mide el nivel de exactitud entre la salida del modelo con los datos que conocemos.

El entrenamiento se realiza con todo conjunto de imágenes y al final del entrenamiento del modelo se evalúa la precisión.

Para entrenar el modelo se debe escoger los parámetros número de repeticiones (epoch, número de veces que se recorre todo el conjunto de datos para entrenamiento), tasa de aprendizaje, iteraciones, tamaño del lote (número de muestras que se usan en una iteración). A partir de la correcta selección de los parámetros tendremos mejores resultados y una mejor precisión al final de entrenamiento. Otro factor para tomar en cuanta al momento de la selección de los parámetros es el uso de los recursos computacionales, debido a que, al

trabajar con imágenes de alta resolución y en espacio de color RGB, es necesario el uso de una gran cantidad de procesamiento y memoria.

Para la ejecución del entrenamiento utilizamos los siguientes recursos computacionales mostrados en la Tabla 5.

Tabla 5. Recursos computacionales

CPU:	Intel Core i5 6200 U @2.30 GHz
Memoria:	8 Gb DDR3 1600 MHZ
Gráficos:	Intel HD Graphics 520

(Elaboración propia)

La implementación de algoritmos de aprendizaje profunda requiere una potencia de procesamiento elevada, especialmente en la etapa de entrenamiento. Por ello, es imprescindible enfocarse en dos cuestiones para seleccionar los parámetros, la primera es el uso de los recursos computacionales que tenemos disponibles y la segunda es alcanzar una precisión aceptable para el modelo. Llegar al equilibrio entre estos dos factores es la principal meta al momento de escoger los parámetros de entrenamiento. Para lo cual, se ha efectuado diversos entrenamientos, con varias combinaciones de parámetros que conllevó a diferentes resultados, con diferentes niveles de precisión y con distintos rangos de tiempo empleados.

A continuación, describimos cada uno de los entrenamientos más representativos con sus respectivos resultados.

- **Primer entrenamiento**

Tabla 6. Parámetros del primer entrenamiento

Taza de aprendizaje:	0.001
Tamaño imagen de entrada:	224x224x3
Repeticiones:	20
Iteraciones por repetición:	1

(Elaboración propia)

En la Tabla 6 se muestra los parámetros configurados para la ejecución del primer entrenamiento. En la Figura 26 se examina el proceso de entrenamiento el cual se ejecuta con una duración de 5 minutos y 23 segundos, obteniendo a una exactitud máxima de 66%. En el primer entrenamiento se ejecuta con una tasa de aprendizaje baja, pero con un número de repeticiones mínimo, lo que da como resultado una precisión baja, pero con una potencia computacional también baja.

Figura 26. Exactitud y pérdida del primer entrenamiento. (Elaboración propia)

En consecuencia, al seguir aumentando el número de repeticiones no se observa una disminución en el valor de la función de coste, por consiguiente, el modelo deja de aprender y no llega a ser una buena elección de parámetros y se ejecuta nuevamente otro proceso de entrenamiento con nuevos parámetros.

- **Segundo entrenamiento**

Una vez comprobado la falta de exactitud en el primer entrenamiento y con la intención de aumentar el índice de precisión se aumenta el número de repeticiones, así como el número de iteraciones para que el modelo ejecute más iteraciones con todo el conjunto de datos. De esta manera, le permita al modelo generar más conclusiones, es decir, más aprendizaje y lograr minimizar la función de coste. En la Tabla 7 encontramos los parámetros utilizados para segundo entrenamiento. El principal cambio con respecto al primer entrenamiento está en que se eleva el número de repeticiones a 100.

Tabla 7. Parámetros del segundo entrenamiento

Taza de aprendizaje:	0.001
Tamaño imagen de entrada:	224x224x3
Repeticiones:	100
Iteraciones por repetición:	7

(Elaboración propia)

En la Figura 27 se observa los resultados obtenidos. El entrenamiento se ejecuta con una duración de 8 minutos y 31 segundos, como consecuencia, se logra obtener una exactitud promedio del 60%. Analizando el progreso del entrenamiento se detiene el proceso a las 163 iteraciones, debido a que, no presenta una disminución en la función de coste por lo que no llegamos a un entrenamiento óptimo y se debe buscar otra combinación de parámetros para mejorar los resultados.

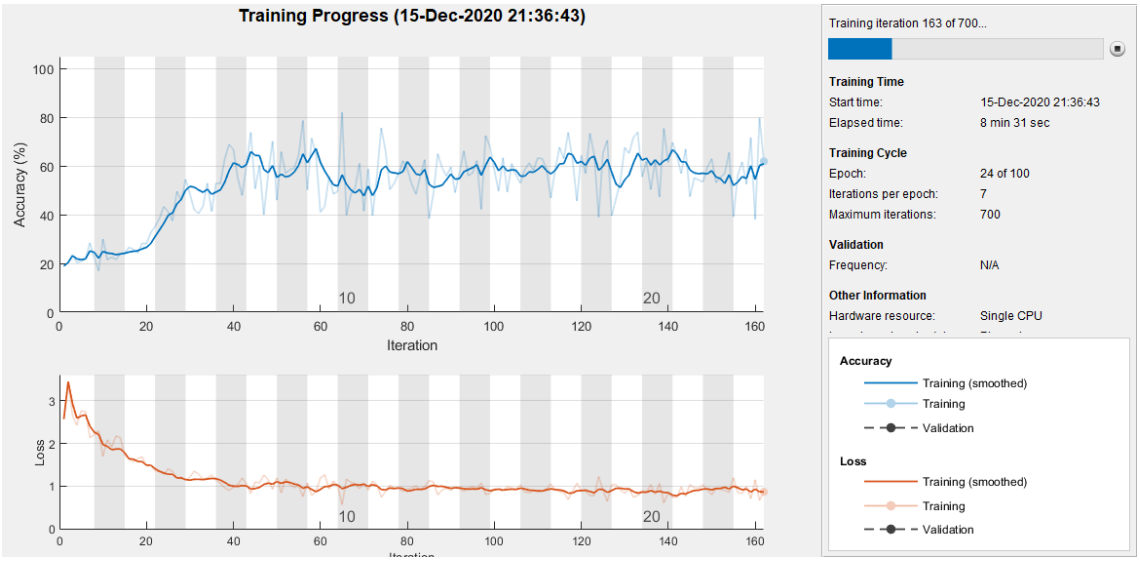


Figura 27. Exactitud y pérdida del segundo entrenamiento. (Elaboración propia)

- **Tercer entrenamiento**

En los entrenamientos previos, los hemos ejecutado con una tasa de aprendizaje baja, con un valor de 0.001. Se ha comprobado con pocas repeticiones, así como, elevando el número de repeticiones. Como resultado, no se ha logrado generar un modelo óptimo con una buena precisión. El tercer entrenamiento está configurado con los parámetros mostrados en la Tabla 8, con una taza de aprendizaje mayor a los entrenamientos previos, con la finalidad de realizar un descenso de gradiente más pronunciado, es decir, buscamos que el modelo aprenda de manera más rápida.

Tabla 8. Parámetros del tercer entrenamiento

Taza de aprendizaje:	0.01
Tamaño imagen de entrada:	224x224x3
Repeticiones:	100
Iteraciones por repetición:	3

(Elaboración propia)

En la Exactitud y pérdida del tercer entrenamiento. (Elaboración propia)Figura 28 se puede apreciar que el entrenamiento se ejecuta con una duración de 23 minutos y 48 segundos, llegando a alcanzar una exactitud promedio de 68%. De la misma manera que el entrenamiento previo, se detiene el proceso a las 233 iteraciones, debido a que, el modelo ya no se encuentra aprendiendo, es decir, no disminuye el valor de la función de pérdida que permanece constante en 0,75. Este efecto se conoce como *underfitting*, donde el modelo se encuentra imposibilitado de obtener nuevo aprendizaje del conjunto de datos analizado.

Por otra parte, en este entrenamiento se observa una leve mejora en la precisión con un pequeño coste adicional en tiempo de procesamiento, lo que se refleja en un menor coste computacional.

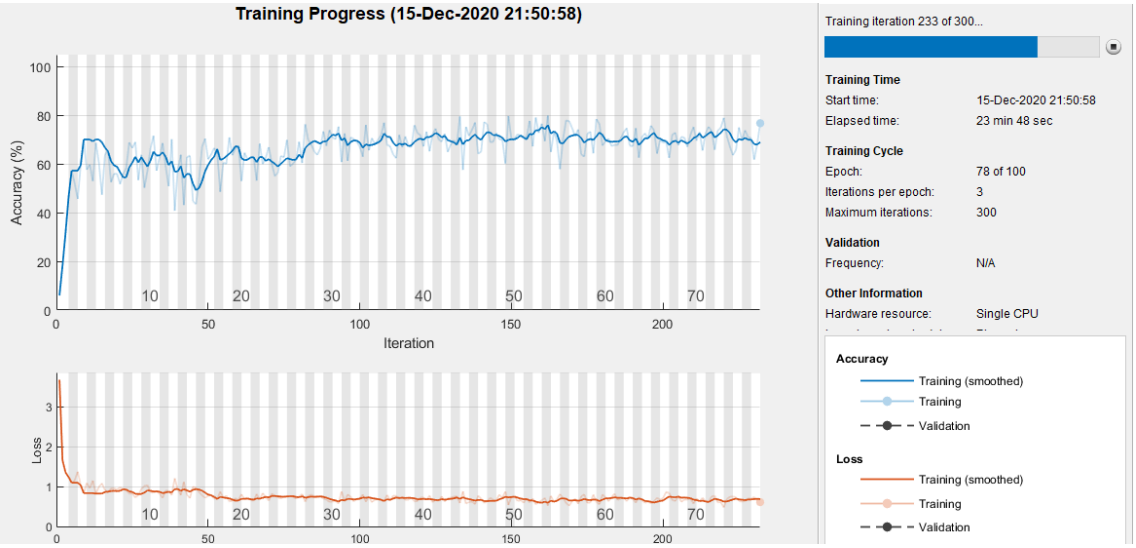


Figura 28. Exactitud y pérdida del tercer entrenamiento. (Elaboración propia)

- **Cuarto entrenamiento**

Para contrarrestar el efecto *underfifting*, debemos ajustar los parámetros experimentando un equilibrio entre las repeticiones, las iteraciones y la tasa de aprendizaje. Como observamos en los entrenamientos previos, aumentando el valor de la taza de aprendizaje se logra obtener una mejor precisión. Como consecuencia, para el cuarto entrenamiento utilizamos los parámetros mostrados en la Tabla 9. Parámetros del cuarto entrenamiento diferencia radica

en que se disminuye la tasa de aprendizaje y el número de repeticiones, para de esta manera, intentar no caer en un problema de *underfifting.*

Tabla 9. Parámetros del cuarto entrenamiento

Taza de aprendizaje:	0.05
Tamaño imagen de entrada:	224x224x3
Repeticiones:	20
Iteraciones por repetición:	4

(Elaboración propia)

Una vez finalizado con el entrenamiento se obtiene muy buenos resultados como se muestra en la Figura 29. Se logra obtener una precisión máxima del 81% y un valor de la función de pérdida de 0,6. La principal desventaja al aumentar la taza de aprendizaje y ejecutar un total de 80 iteraciones radica en el elevado tiempo de cómputo, alcanzando a una duración de 319 minutos y 28 segundos, y un elevado coste computacional.

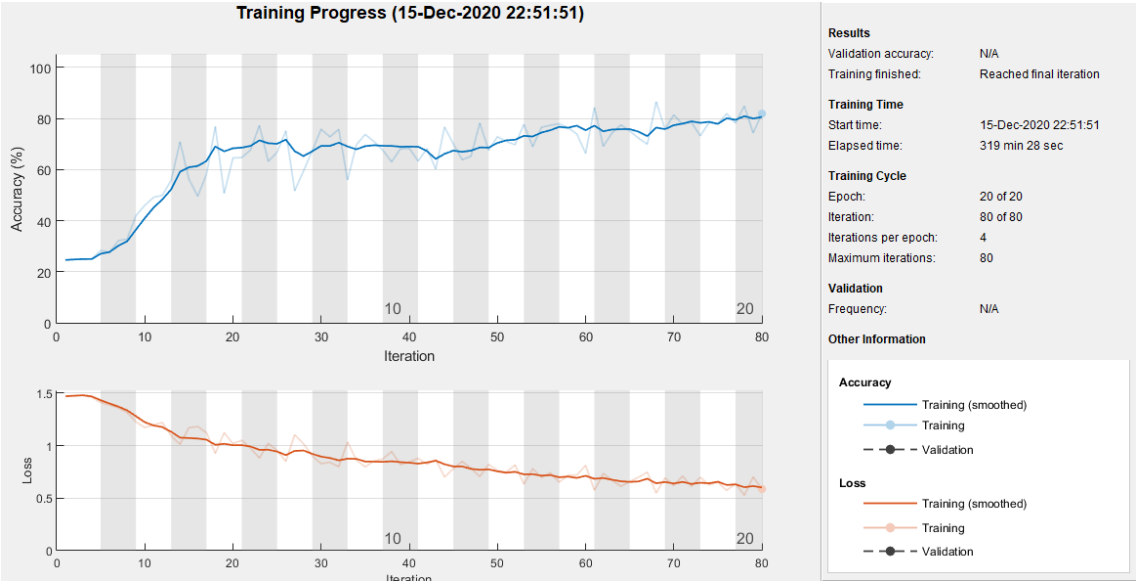


Figura 29. Exactitud y pérdida del cuarto entrenamiento. (Elaboración propia)

- **Quinto entrenamiento**

En el cuarto entrenamiento se ha comprobado los buenos resultados obtenidos con una taza de aprendizaje de 0,05. Razón por la cual, se mantiene este parámetro y se aumenta el número de repeticiones e iteraciones, para llegar a obtener una precisión más elevada. Los parámetros de entrenamiento se encuentran en detalle en la Tabla 10.

Tabla 10. Parámetros del quinto entrenamiento

Taza de aprendizaje:	0.05
Tamaño imagen de entrada:	224x224x3
Repeticiones:	50
Iteraciones por repetición:	4

(Elaboración propia)

Una vez finalizada la ejecución del entrenamiento, como consecuencia se emplea un total de 200 iteraciones, con lo que se obtiene los mejores resultados hasta el momento en todos los entrenamientos ejecutados. Se consigue un modelo con una precisión de alrededor del 83% y un valor de la función de pérdida inferior a 0.5. El principal factor en contra el ejecutar este entrenamiento es el alto cómputo y procesamiento empleado. Al final, llega a superar una duración de 836 minutos y 55 segundos. Por esta razón, emplea una capacidad del 100% de uso de recursos del sistema disponible.

Este modelo entrenado es el elegido para las siguientes fases de evaluación, debido a sus buenos resultados. Todo el proceso de entrenamiento y los parámetros empleados en el que será el modelo final de segmentación semántica se aprecia en la Figura 30.

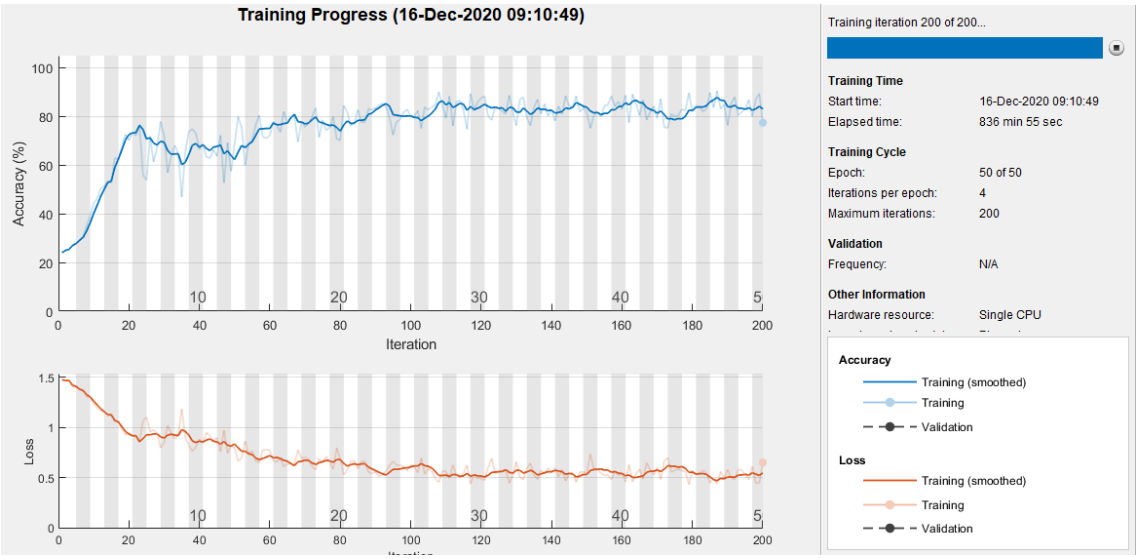


Figura 30. Exactitud y pérdida del quinto entrenamiento. (Elaboración propia)

El modelo de entrenamiento puede llegar a tener mejores resultados si se ejecuta algunas mejoras, entre las principales constan usar un conjunto de datos de entrenamiento con un mayor número de muestras. Adicionalmente, se recomienda un sistema de cómputo con mejores características como un mejor procesador o el uso de un GPU, con lo que se obtiene un mejor rendimiento al momento de procesar imágenes o videos.

4.4.3. Pruebas del modelo de segmentación semántica

Una vez seleccionado el modelo de segmentación semántica entrenado, se lo guardará en memoria para proceder a ejecutar las pruebas. Las pruebas se ejecutan con otro dataset o conjunto de imágenes aéreas "ocultas" para nuestro ordenador. Estas pruebas nos permitirán evaluar la precisión del modelo, la fiabilidad de los resultados y la detección de efectos indeseados como la imposibilidad de analizar nuevos datos de entrada generando clasificaciones erróneas.

4.5. Generación de las rutas libre de obstáculos

Para alcanzar el objetivo de generar una ruta libre de circulación, es necesario aislar del algún modo las secciones de la imagen que corresponda a esta ruta. Para ello, hemos definido previamente cuatro categorías en las que cada píxel se asociará. Las categorías son: ÁRBOL,

VÍAS, CASAS y VEHÍCULOS. Con el algoritmo de segmentación semántica se reconocerá cada una de las cuatro categorías dentro de la imagen, seguidamente, se puede aislar cada una de las clases en máscaras binarias. Con la máscara aislada de VÍAS tendremos la ruta en que se puede circular, razón por la cual nos aseguraremos de no encontrar otros objetos como ÁRBOL, CASAS o VEHÍCULOS.

Con el conjunto de datos de entrenamiento previamente etiquetado, se observa la manera en que podemos aislar cada una de las categorías o clases y representar mediante una imagen binaria llamada máscara. En la Figura 31 se analiza cada una las posibles máscaras tomando como imagen de entrada a un elemente del conjunto de entrenamiento. En conclusión, al momento de generar una máscara binaria se asigna a todos los píxeles correspondientes a la clase un valor de uno, mientras que, a los píxeles que no corresponda a la clase se les asigna un valor de cero.

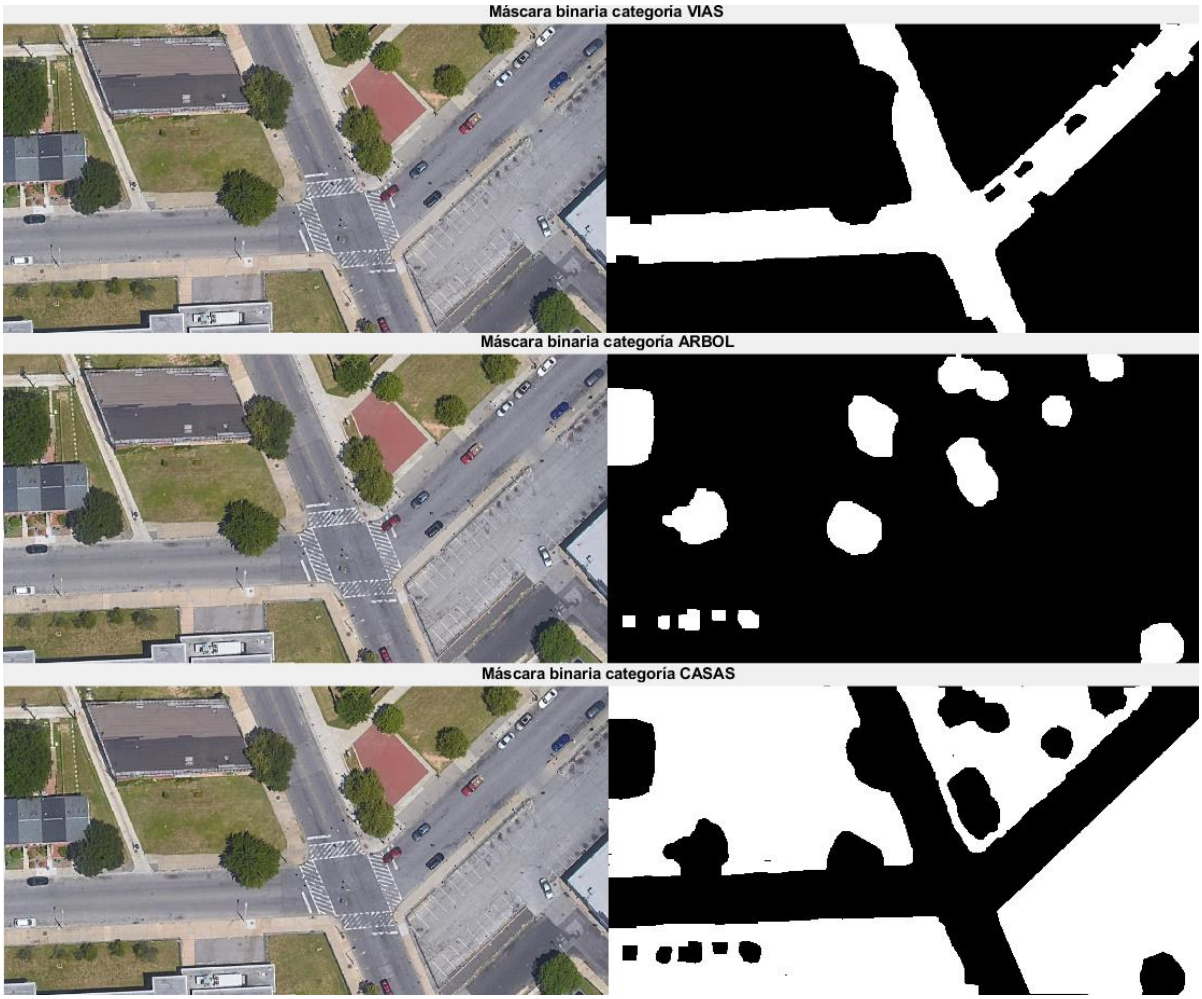


Figura 31. Máscaras binarias de las categorías dentro de la imagen. (Elaboración propia)

En este trabajo se busca generar una ruta libre de circulación, de manera que, una vez obtenida la máscara binaria correspondiente a la categoría VÍAS, se genera un mapa de ocupación que nos mostrará los sectores de la imagen donde está libre de objetos y donde no se pueda circular. Este mapa de ocupación será igualmente una matriz binaria.

4.6. Pruebas y resultados

En esta sección se presentan los resultados obtenidos con el modelo de segmentación semántica basado en una arquitectura SegNet. Para ello, previamente ha sido entrenado, probado y validado el modelo. Se presentarán los resultados sobre un conjunto de imágenes para realizar una valoración de los objetivos a alcanzar. Es decir, a partir de una imagen aérea se busca segmentar un camino libre de obstáculos que permita la circulación de un sistema móvil.

Para evaluar el modelo utilizado se han empleado cuatro métricas:

1. Exactitud global: Indica el porcentaje de píxeles que han sido clasificados correctamente, independientemente de la clase y del número total de píxeles.
2. Exactitud promedio: Indica el porcentaje de píxeles identificados correctamente para cada clase. Para cada clase, la exactitud es la relación entre píxeles clasificados correctamente y el número total de píxeles en esa clase, que se conocen previamente con las imágenes etiquetadas.

3. Índice Jaccard: Conocido como IoU (Intersection over union), mide el porcentaje de píxeles clasificados correctamente con respecto al número total de píxeles reales y predichos en esa clase.
4. Puntuación de coincidencia de contorno: bfscore, calcula la coincidencia de contornos entre la segmentación pronosticada y la segmentación verdadera.

Por medio de un conjunto de datos de prueba agrupados en un *datastore*, se evalúa el modelo de segmentación semántica para generar un conjunto de objetos en igual cantidad que de imágenes de prueba, este nuevo grupo de datos contiene objetos con la clasificación de cada píxel en una de las categorías, es decir se genera un tipo de objeto categórico.

Con las imágenes previamente etiquetadas para el entrenamiento (resultados que esperamos obtener) y con los objetos generados al aplicar el modelo de segmentación (resultados reales), se comparan uno a uno para generar las métricas detalladas en la Tabla 11.

Tabla 11. Métricas de evaluación del modelo de segmentación semántica

Exactitud global	Exactitud promedio	Índice Jaccard	Bfscore
0.8205	0.8048	0.7746	0.5236

(Elaboración propia)

Como se observa en la Tabla 11, ejecutado el modelo de segmentación al conjunto de prueba se logra alcanzar una exactitud de alrededor del 82%, en efecto, similar valor al que se había predicho al momento de realizar el entrenamiento del modelo. Con estos resultados podemos afirmar que se tiene una fiabilidad aceptable, por consiguiente, el modelo puede segmentar de manera autónoma cualquier imagen aérea.

Para validar los resultados obtenidos se realizan un conjunto de procedimientos a diversas imágenes. Primero, aplicamos el modelo entrenado de segmentación semántica a las imágenes de prueba. Segundo, analizamos las métricas para evaluar el resultado en cada imagen. Tercero, generamos una máscara binaria con la finalidad de segmentar la ruta de libre circulación. Seguidamente, se puede aplicar un algoritmo de planeación de ruta en la que por

medio de segmentos de recta nos marcará un camino transitable desde un punto inicial hasta un punto final. Finalmente, podemos segmentar en la imagen original a color la ruta libre de circulación.

- **Primera prueba**

Aplicamos el modelo de segmentación a una imagen aérea de tipo RGB. Como resultado obtenemos la clasificación de cada píxel dentro de una categoría que viene representada en la Figura 32 con un color diferente.

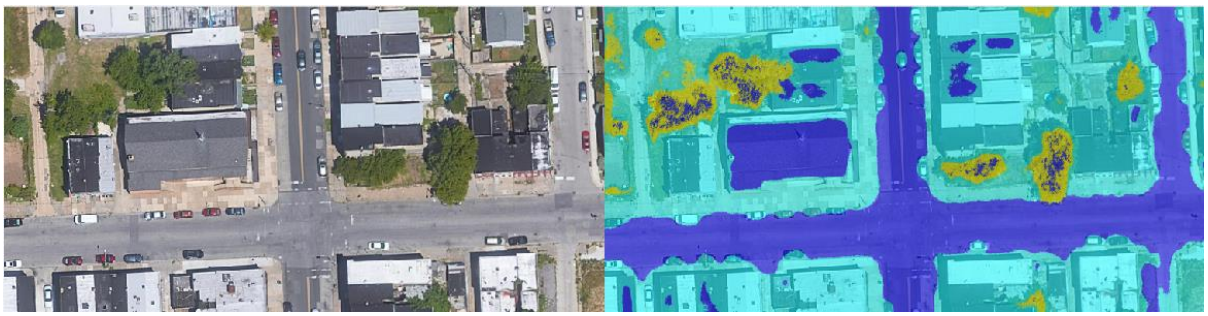


Figura 32. Primera prueba, segmentación semántica. (Elaboración propia)

La Figura 32 nos muestra una idea visual de la manera que se ha clasificado cada píxel en las cuatro categorías establecidas: ÁRBOL, VÍAS, CASAS, VEHÍCULOS. Por un lado, la categoría de interés VÍAS está representado con un color azul, y agrupa toda el área de libre circulación. Por otra parte, se representa a la categoría ÁRBOL en color amarillo, la categoría CASAS en color verde y la categoría VEHÍCULOS es casi imperceptible debido a la pequeña proporción de píxeles asociados con relación a la cantidad de píxeles totales de la imagen. Las categorías ÁRBOL, CASAS, VEHÍCULOS son analizadas como objetos donde no se puede circular para este estudio.

Para tener una idea más objetiva, se procede a ejecutar un análisis individual a cada imagen, analizando los píxeles dentro de cada categoría o clase, de manera que se calculan las métricas de evaluación del modelo. En la Tabla 12 se listan los resultados obtenidos.

Tabla 12. Métricas de evaluación, primera prueba.

Categoría	Exactitud promedio	Índice Jaccard	Bfscore
ÁRBOL	0.6425	0.4734	0.4294
VÍAS	0.8311	0.7109	0.4090
CASAS	0.8729	0.7745	0.5669
VEHÍCULOS	0.0012	0.0010	0.0001

(Elaboración propia)

Para generar la ruta libre para circular, se genera una máscara binaria solamente con las "VÍAS". Como la precisión de la segmentación semántica en la categoría VÍAS alcanza un valor del 83%, se puede mejorar el segmentado por medio de la aplicación de métodos de procesamiento de imágenes como: filtros, técnicas para llenado de huecos, eliminación de píxeles que no correspondan a dicha clase, etc. Con la máscara binaria procesada, se crea un mapa de ocupancia, que muestra un 1 si dicha posición está libre de obstáculos y 0 si la posición está ocupada. Posteriormente, se aplica un algoritmo de planificación de rutas. En este trabajo aplicamos el algoritmo PRM (Probabilistic Roadmap) por medio una simulación de un robot diferencial para el seguimiento de la ruta.

En Figura 33 se representa en la parte superior la máscara binaria procesada con la generación de ruta con el algoritmo PRM. En la parte inferior, se encuentra la simulación de movilidad de un robot móvil diferencial y la máscara aplicada a la imagen original, de esta manera, distinguimos la segmentación de la vía con respecto a los demás objetos de la imagen. Por ende, no se toman en cuenta los vehículos, las edificaciones y la vegetación, si no, únicamente una ruta libre de obstáculos que puede circular cualquier sistema móvil.

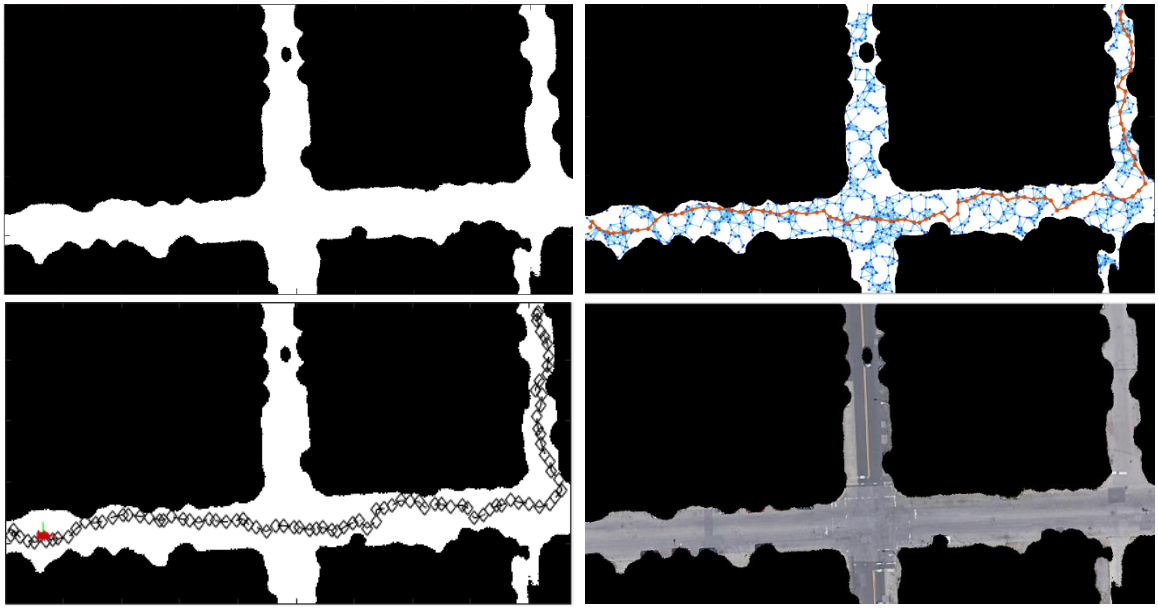


Figura 33. Primera prueba, generación de ruta libre de obstáculos. (Elaboración propia)

- **Segunda prueba**

De la misma manera que las pruebas ejecutadas en el apartado anterior, se realiza a otra imagen del conjunto o dataset de prueba. Al aplicar el modelo de segmentación semántica se obtiene la siguiente segmentación de clases representada en la Figura 34.

Figura 34. Segunda prueba, segmentación semántica. (Elaboración propia)

De igual modo, cada categoría viene representada por un color que se diferencia claramente. Se observa nuevamente como la extracción de las clases VÍAS, ÁRBOL y CASAS la hace con una precisión considerable, por el contrario, no se tiene una buena segmentación de la clase

AUTOMÓVILES. En la Tabla 13. Métricas de evaluación, segunda prueba.Tabla 13 se detalla las métricas obtenidas al aplicar el modelo sobre la segunda imagen de prueba.

Tabla 13. Métricas de evaluación, segunda prueba.

Categoría	Exactitud promedio	Índice Jaccard	Bfscore
ÁRBOL	0.6710	0.5049	0.7113
VÍAS	0.8312	0.7112	0.6112
CASAS	0.9134	0.8405	0.5421
VEHÍCULOS	0.0001	0.0001	0.0000

(Elaboración propia)

Adicionalmente, en la Tabla 13 se observa que al momento de segmentar la clase VÍAS se alcanza una precisión del 83%. De la misma manera, se ejecuta una etapa de procesamiento a la máscara binaria con la finalidad de mejorar las características de la segmentación de la ruta libre de obstáculos.

En la Figura 35 se muestra el proceso llevado a cabo posterior a la segmentación de los píxeles es sus respectivas categorías. Primero se genera el mapa de ocupación correspondiente a la clase VÍAS, luego se planifica una ruta con el algoritmo PRM y finalmente, se simula la movilidad de un robot diferencial a lo largo de la ruta.

Puesto que los objetos automóviles no tienen una buena extracción al momento de segmentar, el modelo los asigna dentro de la clase CASAS. Ciertamente, no es la clase que corresponde, sin embargo, para el objetivo de generar una ruta libre de obstáculos no es un impedimento, debido a que, tanto las CASAS como los AUTOMÓVILES son considerados objetos que se debe evadir con lo que nos aseguramos de no tener obstáculos en la ruta generada.

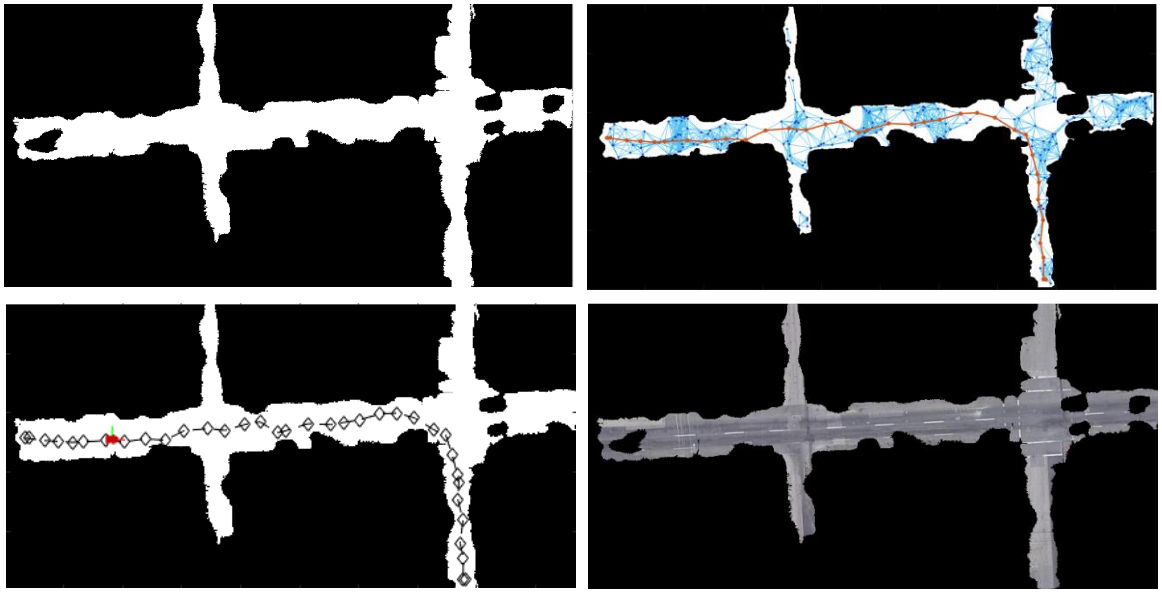


Figura 35. Segunda prueba, generación de ruta libre de obstáculos. (Elaboración propia)

- **Tercera prueba**

Ejecutamos el modelo de segmentación semántica a una tercera imagen del conjunto de prueba. En la Figura 36 se muestra la clasificación de los píxeles en su respectiva categoría.

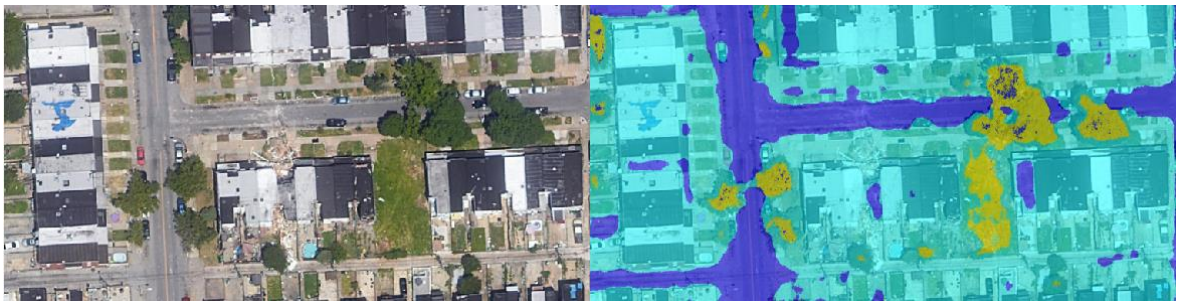


Figura 36. Tercera prueba, segmentación semántica. (Elaboración propia)

Tanto como en las pruebas anteriores como en la tercera prueba, se observa una buena segmentación de las clases VÍAS en color azul, CASAS en color verde y ÁRBOLES en color amarillo. El análisis de las métricas de evaluación del modelo para esta prueba se detalla en la Tabla 14.

Tabla 14. Métricas de evaluación, tercera prueba.

Categoría	Exactitud promedio	Índice Jaccard	Bfscore
ÁRBOL	0.6243	0.4538	0.6110
VÍAS	0.7892	0.6518	0.5073
CASAS	0.9222	0.8556	0.4773
VEHÍCULOS	0.0012	0.0001	0.0000

(Elaboración propia)

La segmentación semántica de la categoría VÍAS alcanza una exactitud del 78.9%. Una pequeña disminución en el valor de la precisión debido a la distribución de los objetos dentro de la imagen original. Como se puede observar en la Figura 36, en la parte izquierda se encuentra la imagen original en la cual se localizan objetos del tipo ÁRBOL sobre objetos del tipo VÍAS. Esta ubicación provoca una dificultada añadida al modelo de aplicar el algoritmo, generando dificulta para discernir estos píxeles.

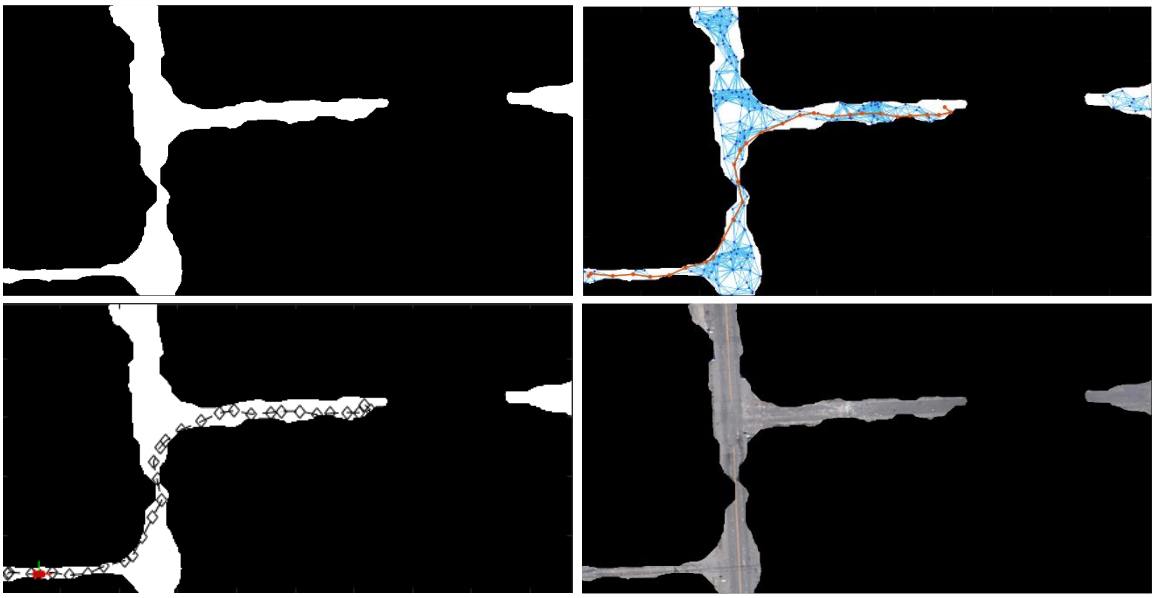


Figura 37. Tercera prueba, generación de ruta libre de obstáculos. (Elaboración propia)

De la igual forma, se genera una máscara binaria de la clase VÍAS a la cual aplicamos técnicas de procesamiento de imágenes para mejorar la segmentación. Adicionalmente, se ejecuta un algoritmo de planificación de ruta y se simula la movilidad de un robot diferencial. En la Figura 37 se observa las imágenes resultantes obtenidas al aplicar este procedimiento.

Como se observa en las pruebas ejecutadas, se logra alcanzar la segmentación de una ruta libre de circulación sin obstáculos con una precisión promedia del 82%. Esta precisión se puede mejorar al aplicar un post procesamiento a la máscara binaria resultante.

Existe un índice de precisión bastante bajo al momento de reconocer automóviles en las imágenes. Esto debido a la pequeña relación de píxeles que corresponde a un auto con respecto a otro objeto. Un factor influyente para la mejora es aumentar la resolución de las imágenes al momento de evaluar al modelo. El incrementar la dimensión de una imagen conlleva un aumento en el consumo computacional que se encuentra fuera del alcance de los recursos empleados para este trabajo.

5. Conclusiones y trabajo futuro

El procesamiento de imágenes es una herramienta de alto impacto que se ha venido implementando en una gran cantidad de aplicaciones y que se ha desarrollado en los últimos años. Tras el surgimiento de nuevas ramas como la Inteligencia artificial, y más específicamente el aprendizaje profundo, ha llevado al procesamiento de imágenes al siguiente nivel, con aplicaciones infinitas como en áreas de la medicina, ingeniería, robótica, etc.

5.1. Conclusiones

En este trabajo Fin de Máster se ha propuesto implementar un algoritmo de segmentación semántica de imágenes para la obtención de rutas libres de obstáculos a partir de imágenes urbanas aéreas, con lo que se puede afirmar que se ha cumplido el objetivo a totalidad. El trabajo ha sido ejecutado implementado un modelo de segmentación semántica basada en la arquitectura SegNet. Tras la recopilación del conjunto de imágenes, el pre-procesado de las mismas, la implementación de la arquitectura, el entrenamiento del modelo, pruebas y validaciones, post procesado de las máscaras binarias y el segmentado de la ruta libre de obstáculos, se han obtenido las siguientes conclusiones:

- Existen varias técnicas de segmentación para extraer características de una imagen. En estos momentos las técnicas más empleadas son desarrolladas bajo conceptos de aprendizaje profundo, especialmente para aplicaciones de reconocimiento de objetos y patrones en imágenes con excelentes resultados.
- Existe una gran cantidad de bases de datos de libre acceso, con un gran número de imágenes que ya se encuentran clasificadas y etiquetadas, hecho que facilita el entrenamiento e implementación de un modelo de aprendizaje profundo para clasificación de imágenes. En el presenta trabajo se ha seleccionado, extraído y agrupado varias imágenes aéreas con el fin de crear nuestro propio conjunto de 100 imágenes, con el fin de implementar desde cero toda la metodología para cumplir con el objetivo propuesto.
- Debido a la naturaleza de las imágenes aéreas fue necesario aplicar técnicas de procesamiento de imágenes sobre el conjunto de datos obtenidos previamente, para

lograr una mejora en las características que se necesita extraer de las imágenes. El procesado facilita la implementación del modelo de segmentación semántico, especialmente en la fase de entrenamiento. La eliminación de todo tipo de ruido, la distribución correcta de las intensidades y la corrección de la iluminación favorecen al aumento el rendimiento de los modelos de clasificación y disminuya el coste computacional.

- Al aplicar un modelo de segmentación semántica con una arquitectura codificador-decodificador se logra segmentar automáticamente imágenes aéreas con una precisión del 83%. Adicionalmente, se obtiene un valor de la función de coste inferior a 0.5. Como resultado, se alcanza un promedio de 82% de precisión sobre imágenes utilizadas como pruebas.
- Si bien se alcanza un valor del 82% de precisión, se puede afirmar que los resultados obtenidos son buenos para el objetivo que se ha planteado. En efecto, el modelo permite segmentar claramente una ruta libre de obstáculos, excluyendo a objetos que puede ser catalogados como entidades que no se puede circular, tal como las edificaciones, los árboles u otros vehículos.
- Se puede mejorar la efectividad del modelo implementando varios cambios. Uno de ellos es aumentar el tamaño del conjunto de entrenamiento, utilizando una base de datos de libre disponibilidad, donde se tiene acceso a cientos de imágenes ya etiquetadas. Por otra parte, otra mejora se logra aumentando la resolución de las imágenes de entrada, es decir, obtener una mayor información por píxel. No obstante, utilizar imágenes en alta resolución conlleva el aumento considerable en el coste computacional.
- Al generar una ruta libre de obstáculos estamos generando una máscara binaria con los píxeles que corresponden al objeto VÍAS. Esta clase igualmente alcanza una precisión del 83%, es decir, que el 83% de los píxeles clasificados como ruta libre de obstáculos son correctos. Este valor puede ser mejorado al implementar un post procesamiento, lo que ayuda a eliminar píxeles aislados, ruido generado al momento de clasificar y discontinuidades que no se puede aceptar dentro de la ruta.
- Posterior a la segmentación del trayecto, se puede aplicar diversos algoritmos de planificación de rutas con la finalidad de generar un conjunto de puntos que nos lleve desde una posición inicial a una posición final, pero con la certeza de que es un camino libre de obstáculos.

5.2. Líneas de trabajo futuro

Una vez cumplido con el objetivo propuesto para este Trabajo Fin de Máster, se pueden definir varias líneas de trabajo futuro para dar continuidad al estudio con la finalidad de perfeccionar el modelo o llevar a otras aplicaciones prácticas:

- Aumentar la cantidad de la base de datos de imágenes aéreas, con el fin de obtener una mejor precisión al momento de entrenar el modelo, así como obtener mejores resultados al momento de segmentar la ruta libre de obstáculos. También se puede utilizar un dataset disponible en la web donde ya se tiene etiquetadas una gran cantidad de imágenes.
- Diseñar un sistema de movilidad para un robot móvil. Con la segmentación de un camino libre de obstáculos y la planificación de la ruta que fueron los resultados obtenidos en el trabajo, implementar un sistema que permita controlar los parámetros de velocidad, orientación, aceleración de un robot móvil que permita circular libremente.
- Con la segmentación y clasificación de los píxeles en varias categorías, también se puede obtener información visual de las áreas verdes dentro de una urbe, se puede estudiar la disminución de la vegetación a lo largo del tiempo. Se puede realizar estudios comparativos respecto a años anteriores.
- Hemos desarrollado una metodología que permite segmentar semánticamente imágenes aéreas, la misma metodología puede ser implementada para realizar este proceso en una fuente de vídeo. Al considerar un vídeo como una secuencia de imágenes con una frecuencia elevada, se puede segmentar cada frame del vídeo. A partir de un vídeo se puede obtener información como control de tráfico, seguimiento vehicular, detección de incendios forestales, etc.

Como se observa, existe una gran cantidad de aplicaciones posibles que se pueden llevar a cabo a partir del trabajo propuesto que conlleva la vinculación de varias áreas como el procesamiento de imágenes, la visión artificial, la robótica y el control automático.

BIBLIOGRAFÍA

Alonso, M. A. (2009). *Espacios de Color RGB, HSI y sus Generalizaciones a n-Dimensiones por*.

Badrinarayanan, V., Handa, A., & Cipolla, R. (2015). *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling*. http://arxiv.org/abs/1505.07293

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2016). *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*. http://mi.eng.cam.ac.uk/projects/segnet/.

Bharathi, P. T., & Subashini, P. (2011). Optimization of image processing techniques using neural networks - A review. In *WSEAS Transactions on Information Science and Applications* (Vol. 8, Issue 8).

Borràs, J., Delegido, J., Pezzola, A., Pereira, M., & Morassi, G. (2017). Land use classification from Sentinel-2 imagery. *Revista de Teledetección*, *48*, 55–66. https://doi.org/10.4995/raet.2017.7133

Breiman, L. (1998). *Classification and regression trees. Leo Breiman [and others].* https://www.routledge.com/Classification-and-Regression-Trees/Breiman-Friedman-Stone-Olshen/p/book/9780412048418%0Ahttp://proxy.library.tamu.edu/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=cat03318a&AN=tamug.3057200&site=eds-live

Chanampe, H., Aciar, S., De La Vega, M., Luis, J., Sotomayor, M., Carrascosa, G., Lorefice, A., Nacional, U., & Juan, S. (2019). *Modelo de Redes Neuronales Convolucionales Profundas para la Clasificación de Lesiones en Ecografías Mamarias*. http://sedici.unlp.edu.ar/handle/10915/77381

Coello Blanco, L., Casas, L., Lidia, O., González, P., Caballero Mota, Y., & De Camagüey, U. (2015). Redes neuronales artificiales en la producción de tecnología educativa para la enseñanza de la diagonalización 1. In *Revista Academia y Virtualidad* (Vol. 8, Issue 1).

Durán, J. (2017). *Redes Neuronales Convolucionales en R Reconocimiento de caracteres escritos a mano Redes Neuronales Convolucionales en R Reconocimiento de caracteres escritos a mano Redes Neuronales Convolucionales en R*. 78. http://bibing.us.es/proyectos/abreproy/91338/fichero/TFG+Jaime+Durán+Suárez.pdf

Enríquez, R. A. (2016). *Técnicas de Realce de Imágenes 3.0.1 Realce de la Imagen*.

Flórez, R. F., & Fernandez, J. M. (2008). *Las Redes Neuronales Artificiales* (Bello Lorena (ed.)). NETBIBLO, S.L. https://books.google.es/books?hl=es&lr=&id=X0uLwi1Ap4QC&oi=fnd&pg=PA9&dq=redes+neuronales&ots=gOHCgsjvYm&sig=W-1uMZNhtKP1knIq6pRdwQdM3xA#v=onepage&q=redes neuronales&f=false

García, R. (2019). *Análisis de técnicas de segmentación semántica sobre imágenes aéreas con deeplabv3+*. https://ddd.uab.cat/record/211437

Gu, T., Wang, X. H., Pung, H. K., & Zhang, D. Q. (2020). *An Ontology-based Context Model in Intelligent Environments*. http://arxiv.org/abs/2003.05055

Kang, J., Körner, M., Wang, Y., Taubenböck, H., & Zhu, X. X. (2018). Building instance classification using street view images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *145*, 44–59. https://doi.org/10.1016/j.isprsjprs.2018.02.006

Massiris, M., Delrieux, C., & Álvaro Fernández, J. (2018). *DETECCIÓN DE EQUIPOS DE PROTECCIÓN PERSONAL MEDIANTE RED NEURONAL CONVOLUCIONAL YOLO*. https://doi.org/10.17979/spudc.9788497497565.1022

Méndez, R. (2019). *Aprendizaje profundo para la segmentación de lesiones pigmentadas de la piel*. https://idus.us.es/handle/11441/92186

Montiel, H., Jacinto, E., & Martínez, F. H. (2015). Generación de ruta óptima para robots móviles a partir de segmentación de imágenes. *Informacion Tecnologica*, *26*(2), 145–152. https://doi.org/10.4067/S0718-07642015000200017

Moreano, G., Cajamarca, J., & Tenicota, A. (2020). Agricultura de Precisión: Preprocesamiento y Segmentación de Imágenes para Obtención de una Ruta de Navegación Autónoma Terrestre. *Revista Politécnica*, *44*(2), 43–50. https://doi.org/10.33333/rp.vol44n2.05

Moujahid, A., Dornaika, F., Ruichek, Y., & Hammoudi, K. (2019). Towards semantic segmentation of orthophoto images using graph-based community identification. *Neural Computing and Applications*, *31*(2), 1155–1163. https://doi.org/10.1007/s00521-017-3056-y

Paoletti, M. E., Haut, J. M., Plaza, J., & Plaza, A. (2019). Estudio Comparativo de Técnicas de Clasificación de Imágenes Hiperespectrales | Paoletti | Revista Iberoamericana de Automática e Informática industrial. *Revista Iberoamericana de Automática e Informática Industrial*, *16*, 129–137. https://158.42.9.104/index.php/RIAI/article/view/11078/11141

Quintero, C., Merchán, F., Cornejo, A., & Galán, J. S. (2018). Uso de Redes Neuronales Convolucionales para el Reconocimiento Automático de Imágenes de Macroinvertebrados para el Biomonitoreo Participativo. *KnE Engineering*, *3*(1), 585. https://doi.org/10.18502/keg.v3i1.1462

Rosales, A. G. (2019). *Universidad Autónoma Metropolitana Unidad Azcapotzalco Metodología para la Comparación de Algoritmos de Aprendizaje Automático Caso de estudio: Clasificación de Eventos Académicos*. Universidad Autónoma Metropolitana (México). Unidad Azcapotzalco. Coordinación de Servicios de Información. http://zaloamati.azc.uam.mx/handle/11191/6066

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & Lecun, Y. (2014). *OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks*.

Simonyan, K., & Zisserman, A. (2015, September 4). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. http://www.robots.ox.ac.uk/

Suykens, J. A. K., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, *9*(3), 293–300. https://doi.org/10.1023/A:1018628609742

Toro, C. A. (2019). *Algoritmos de segmentación semántica para anotación de imágenes* [Universidad Politécnica de Madrid]. http://oa.upm.es/55407/1/CESAR_ANTONIO_ORTIZ_TORO.pdf

Torres, J. (2018). *Deep Learning – Introducción práctica con Keras*. https://torres.ai/deep-learning-inteligencia-artificial-keras/