Obtención de Información Semántica de Fotografías a Partir de Métodos no Supervisados

Juan Camilo Navia

Luis Guillermo Holguin

Facultad de Ingeniería Civil y Geomática, Universidad de Valle

3740: Ingeniería Topográfica

M.Sc. Francisco Luis Hernández Torres

28 de enero de 2025

Dedicatoria

Juan Camilo Navia Castrillón

A mis padres, quienes con sus sueños de juventud sembraron en mí la semilla de la perseverancia y el deseo de superación. Aunque las circunstancias no les permitieron cumplir su anhelo de cursar un pregrado, su esfuerzo, sacrificio y amor incondicional hicieron posible que yo llegara hasta aquí. Este logro no es solo mío, sino también suyo, porque cada paso que di fue impulsado por su ejemplo y su incansable apoyo. Les dedico este trabajo de grado como un homenaje a su lucha y como una forma de honrar aquellos sueños que, a través de mí, hoy se hacen realidad. Gracias por ser mi inspiración y mi motor. Con todo mi amor.

Luis Guillermo Holguin Agudelo

A mi familia, por su apoyo inquebrantable que me ha permitido perseguir un sueño arraigado en mi corazón desde que tengo memoria. A mis compañeros de la universidad, por estar a mi lado en los momentos más desafiantes, brindándome su ayuda y aliento sin reservas. Y a mi dedicado grupo de profesores, cuya generosidad al impartir directrices esenciales y compartir conocimientos inestimables no sólo hizo posible la culminación exitosa de este proyecto de grado, sino que también enriqueció cada paso de mi formación académica y profesional.

Agradecimientos

Juan Camilo Navia Castrillón

Agradezco de corazón a mi familia por el respaldo material que me brindaron durante todo este proceso. Su apoyo económico y logístico fue fundamental para que pudiera dedicarme por completo a mi formación y a la realización de este trabajo de grado. Sin su ayuda, no habría sido posible alcanzar este logro. Su sacrificio y esfuerzo constante han sido un pilar esencial en mi camino, y les estaré eternamente agradecido por creer en mí y en mis sueños. A mis profesores, quiero expresarles mi sincero agradecimiento por compartir sus conocimientos y experiencias, los cuales enriquecieron mi formación profesional. Sus enseñanzas no solo ampliaron mi visión en el campo de la ingeniería, sino que también me inspiraron a pensar de manera crítica y a enfrentar los retos con determinación. Su orientación y consejos fueron clave para el desarrollo de este trabajo, y valoro profundamente el tiempo y la confianza que depositaron en mí. Gracias por ser guías en este proceso y por contribuir a mi crecimiento académico.

Luis Guillermo Holguin Agudelo

Expreso mi profunda gratitud a mi familia por haberme brindado el apoyo inquebrantable que me ha permitido perseguir un sueño que ha estado en mi corazón desde que tengo memoria, a mis compañeros de la universidad que estuvieron a mi lado ofreciéndome su ayuda y aliento en los momentos en que lo necesitaba, y al dedicado profesorado que generosamente nos impartió las directrices esenciales y los conocimientos inestimables que no solo facilitaron la finalización exitosa de este proyecto de investigación, sino también enriqueció nuestra comprensión a lo largo del la totalidad de nuestra trayectoria académica y profesional.

Tabla de Contenido

Listado de Figuras

Listado de Gráficas

Listado de Tablas

Listado de Abreviaturas

CNN: Redes neuronales convolucionales (Convolutional Neural Networks).
SOM: Mapas auto organizados (Self Organized Maps).
SFM: Estructura desde el movimiento (Structure From Motion).
MVS: Multivisión estéreo (Multi Vision Stereo).
MVG: Multivisión geométrica (Multi Vision Geometry).

Resumen

Este estudio propone un enfoque no supervisado para asociar información semántica a nubes de puntos 3D generadas a partir de fotografías, abordando la carencia de etiquetas en métodos tradicionales como SfM y MVS. Se implementaron redes neuronales convolucionales (CNN) combinadas con autoencoders completos y mapas auto-organizados de Kohonen (SOM), comparando su rendimiento con técnicas clásicas de clustering (k-means) y crecimiento de regiones. Los experimentos se realizaron en tres escenarios: banano (16 imágenes), monumento (396 imágenes) y parque (73 imágenes), procesadas mediante flujos de trabajo en Python con librerías como TensorFlow y OpenMVG.

La metodología incluyó cuatro etapas: generación de nubes de puntos, segmentación no supervisada con redes neuronales (extracción de características, SOM y refinamiento con k-means), segmentación con algoritmos tradicionales, y evaluación mediante métricas de precisión e IoU. Las redes neuronales destacaron en la escena del banano, con precisión global del 98% e IoU de 0.97 para el fondo, aunque la clase "Banano" mostró menor rendimiento (IoU: 0.76-0.87). En contraste, el clustering y crecimiento de regiones presentaron precisiones inferiores (70.8% y 67%, respectivamente), con IoU críticos para clases complejas (ej. 0.08 en clustering para "Banano").

En la escena del monumento, las redes neuronales lograron una precisión máxima del 46.4%, con desafíos en la clase "Ambiente" (precisión $\leq$37.14%). El clustering mostró resultados variables (IoU: 0.14-0.58 para "Vegetación"), mientras el crecimiento de regiones fracasó en segmentar "Vegetación" (precisión $\approx$0%). En el parque, las redes neuronales alcanzaron un 88.4% de precisión en "Vegetación baja" (IoU: 0.87), pero la "Zona artificial" tuvo bajo desempeño (IoU $\leq$0.27). Los métodos tradicionales evidenciaron limitaciones similares, con precisiones $\leq$0.64% para "Zona artificial".

Los resultados demuestran que las redes neuronales superan a los métodos clásicos en entornos controlados (ej. banano), mientras que en escenas complejas (monumento, parque) persisten desafíos por la heterogeneidad espacial. El clustering mostró eficacia parcial en "Vegetación baja" (IoU: 0.80), pero el crecimiento de regiones resultó inconsistente, especialmente en clases minoritarias. Las métricas de IoU reflejaron correlación directa entre la complejidad escénica y la dificultad de segmentación.

El enfoque basado en redes neuronales ofrece mayor robustez para la segmentación semántica no supervisada, aunque requiere optimización en contextos multivariados. La integración de técnicas híbridas y el aumento de datos podrían mejorar la generalización, especialmente para clases con alta variabilidad morfológica. Este trabajo contribuye a la automatización de procesos geoespaciales, reduciendo la dependencia de etiquetado manual en aplicaciones de reconstrucción 3D.

Palabras Clave

Segmentación semántica, nubes de puntos, redes neuronales convolucionales, métodos no supervisados, algoritmos.

Introducción

En las últimas décadas, el empleo de tecnologías como Vehículos Aéreos No Tripulados (UAV) y sensores remotos ha revolucionado la captura de datos geoespaciales, permitiendo la generación de nubes de puntos tridimensionales con aplicaciones en cartografía, arqueología y reconstrucción 3D (Torresan et al., 2017; Xue et al., 2021). Estas nubes de puntos, representaciones digitales de coordenadas espaciales, son herramientas fundamentales para el análisis geoespacial, pero su carencia de información semántica limita su utilidad (Poullis, 2013; Mirzaei et al., 2022).

Métodos tradicionales para la generación de nubes de puntos, como la fotogrametría de estructura desde el movimiento (SfM) y multi vista estéreo (MVS) generan nubes de puntos detalladas (Berra & Peppa, 2020), pero sin información semántica asociada.

Un aspecto importante es la información semántica, que refiere al significado de los datos que se están procesando, esto es análogo al tipo de superficie que representa un punto para este caso. Lo anterior representa una carencia de las nubes de puntos, pues la información semántica es necesaria para su procesamiento (Y. Arayici, 2007; V. Patraucean et al., 2015).

En contraste, los enfoques modernos basados en aprendizaje profundo (Deep Learning, DL) ofrecen soluciones precisas, aunque dependen de procesos de etiquetado manual o conjuntos de datos públicos que no siempre se adaptan a nuevos proyectos (Mirzaei et al., 2022; Jung et al., 2014).

En este contexto, esta investigación propone vincular información semántica extraída de fotografías, a nubes de puntos para optimizar su procesamiento, reducir costos computacionales y mejorar la calidad de los resultados. El objetivo principal es asociar información semántica generada a partir de fotografías con nubes de puntos desprovistas de esta información, mediante el uso de técnicas avanzadas de aprendizaje profundo. Además, se busca comparar esta metodología con enfoques existentes para evaluar su eficacia.

Para este estudio, se desarrollaron métodos para la generación de nubes de puntos de forma automática y con recursos libres, paralelamente, se implementaron técnicas no supervisadas basadas en deep learning, como lo fueron las redes neuronales convolucionales y la fusión de las mismas con mapas auto organizados de kohonen, así como técnicas tradicionales basadas en crecimiento de regiones y clustering. Los resultados de estas técnicas vistos en imágenes segmentadas semánticamente, se fusionaron a las nubes de puntos correspondientes a las mismas imágenes, dando como resultado una nube de puntos etiquetada semánticamente.

Objetivos

Principal

Asociar la información semántica generada a partir de un conjunto de fotografías, vincularla con una nube de puntos sin información semántica.

Secundarios

Determinar una metodología para obtener información semántica, utilizando un conjunto de fotografías asociadas a la nube de puntos.

Adaptar la metodología seleccionada para la adquisición de la información semántica de la nube de puntos a partir de fotografías.

Implementar la caracterización de la información semántica con el método adaptado, comparándola con otras técnicas existentes que extraen información semántica de nubes de puntos.

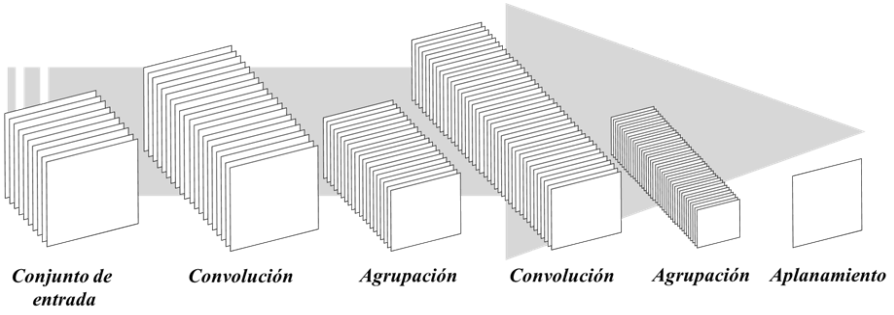
Revisión Literaria

Redes Neuronales

El aprendizaje profundo, especialmente las redes neuronales, es otro enfoque destacado para la segmentación automática. Se describen como sistemas paralelos y jerárquicos que buscan imitar el funcionamiento del sistema nervioso biológico (Kohonen, 1988). También, como sistemas compuestos por numerosos elementos interconectados que procesan información dinámicamente en respuesta a estímulos externos (R.Hecht-Nielsen, 1988). Actualmente, una red neuronal puede entenderse como un sistema computacional que simula al cerebro humano, compuesto por neuronas artificiales organizadas jerárquicamente, capaces de aprender y adaptarse mediante la experiencia para resolver tareas complejas de procesamiento de datos.

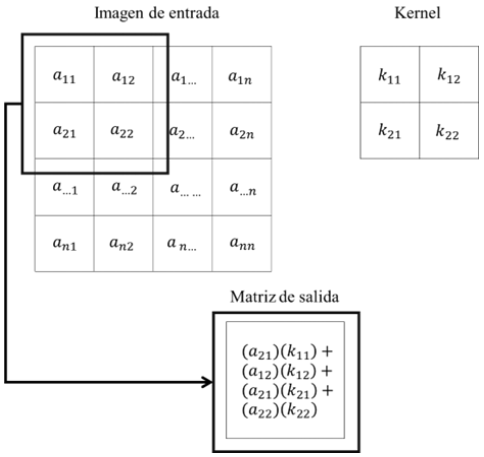
Redes Neuronales Convolucionales

Figura 1.
Estructura estándar de una red neuronal convolucional.



La figura 1, muestra la estructura general de las redes neuronales convolucionales. Estas se definen como un tipo de red neuronal que utiliza una operación matemática llamada convolución. Son especialmente adecuadas para procesar datos con una estructura de cuadrícula, como imágenes. Las CNN utilizan filtros para extraer características importantes de los datos y se caracterizan por su eficiencia en la gestión de parámetros y su capacidad para manejar entradas de gran tamaño, lo que las hace ideales para tareas de visión por computadora y procesamiento de imágenes (Heaton, 2018).

Figura 2.
Operación de convolución para cada píxel.

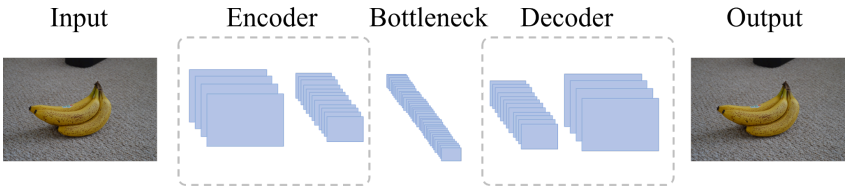


Nota. La figura expone el funcionamiento de la operación de convolución 2D. Adaptado de *Figura 9.1* (p.299), por Bengio et al., (2018).

Los valores del pixel de cada imagen de entrada son vistos como una matriz de dimensión NxM, donde se aplica un filtro o kernel que es otra matriz NxN con valores aleatorios, la matriz de salida se da luego de aplicar la operación de convolución descrita anteriormente.

Full Auto Encoders

Figura 3.
Estructura de un full auto encoder.



La figura 3, retrata la estructura de un full auto encoder tipo. Un autoencoder completo es una red neuronal artificial de aprendizaje profundo que se utiliza para aprender representaciones latentes de datos. Está compuesto por dos partes principales: un codificador y un decodificador. El codificador toma los datos de entrada y los convierte en una representación de menor dimensión, mientras que el decodificador toma la representación de menor dimensión y la reconstruye en los datos de salida. La red se entrena minimizando la función de pérdida entre los datos de entrada y los datos reconstruidos.

Los autoencoders completos se han utilizado con éxito en una amplia variedad de tareas, incluida la reducción de la dimensionalidad, la generación de datos y la clasificación de imágenes. Una de sus principales ventajas es que pueden aprender características de los datos sin supervisión, lo que los hace útiles para tareas donde las etiquetas de datos no están disponibles. Como señalan Goodfellow, Bengio y Courville (2016), "Los autoencoders son modelos que aprenden a copiar su entrada a su salida".

Elementos Importantes de las Redes Neuronales

Funciones de activación

Las funciones de activación son funciones matemáticas esenciales en las redes neuronales artificiales. Introducen no linealidad, permitiendo que las redes aprendan patrones complejos y realicen tareas como clasificación y regresión. Se aplican a la salida de cada neurona, determinando su salida final. Las funciones comunes incluyen sigmoidea, tanh, ReLU, Leaky ReLU y ELU. La elección de la función depende de la tarea específica.

La función de activación ReLU (Rectified Linear Unit) es una función no lineal utilizada comúnmente en las redes neuronales convolucionales (CNN). ReLU es una función simple y eficiente que mapea los valores de entrada negativos a cero y los valores positivos se mantienen sin cambios (Maas et al., 2013).

$$ReLU = max(w^{(i)}x, 0) \rightarrow \begin{cases} w^{(i)}x & w^{(i)}x > 0 \\ 0 & de\ cualquier\ otro\ modo \end{cases} \quad (1)$$

Donde $w^{(i)}$ es la matriz de peso de cada unidad oculta y x es el valor de entrada.

Funciones de pérdida

La función de pérdida en una red neuronal cuantifica la discrepancia entre las predicciones de la red y los valores reales, reflejando su desempeño en el aprendizaje. La red ajusta sus pesos para minimizar esta función. Entre las funciones de pérdida comunes se encuentran el error cuadrático

medio (MSE), la entropía cruzada y la divergencia Kullback-Leibler (KL). La elección de la función depende de la tarea: MSE para regresión y entropía cruzada para clasificación. La función de pérdida es esencial para el entrenamiento de redes neuronales, guiando el proceso de aprendizaje.

La función de entropía cruzada (cross-entropy en inglés) es una función de costo utilizada comúnmente en las redes neuronales convolucionales (CNN) para problemas de clasificación. Esta función de costo se utiliza para medir la discrepancia entre la distribución de probabilidad predicha por el modelo y la distribución de probabilidad real de las etiquetas.

$$Loss = -\sum_{K=1}^{N\ Clases} y_K ln(\hat{y}_K) \tag{2}$$

Donde son las probabilidades para cada clase que el modelo predice (la salida de la función softmax para cada clase) y cuál es la clase correcta a la que pertenece la imagen que estamos procesando (Andreas Eitel et al., 2015).

El error cuadrático medio, es una métrica estadística utilizada para evaluar la precisión de un modelo de predicción, especialmente en tareas de regresión. Se calcula como el promedio de los cuadrados de las diferencias entre los valores observados y los valores predichos por el modelo.

$$MSE = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \tag{3}$$

Donde y_i es el valor observado en la i-ésima observación, $\hat{y}_i$ es el valor predicho por el modelo en la i-ésima observación y n es el número de observaciones.

Métricas de evaluación

Las métricas de evaluación son medidas matemáticas que evalúan el rendimiento de un modelo de red neuronal convolucional (CNN). Incluyen precisión (proporción de predicciones correctas), recall (proporción de objetos verdaderos detectados correctamente), F1-Score (media armónica de precisión y recall) e Intersección sobre Unión (IoU, grado de superposición entre cuadros delimitadores predicho y verdadero). Estas métricas ayudan a evaluar y comparar modelos, y a ajustar hiperparámetros para mejorar el rendimiento.

El índice de precisión es una métrica utilizada para evaluar el rendimiento de un modelo en problemas de clasificación. El índice de precisión calcula el porcentaje de muestras clasificadas correctamente entre todas las muestras evaluadas.

$$Precisión = \frac{TP}{TP+TF} \tag{4}$$

Donde TP son los verdaderos positivos y FP son los falsos positivos (Aldahoul et al., 2021).

La precisión de la clase es una métrica fundamental en la evaluación de modelos de clasificación, calcula el porcentaje de muestras clasificadas satisfactoriamente dentro de una clase.

$$Precisión\ de\ la\ clase = \frac{TP}{TP+FP} \tag{5}$$

Donde como el cociente entre los verdaderos positivos (TP) y la suma de verdaderos positivos y falsos positivos (FP).

El Índice de Intersección sobre Unión (IoU) es una métrica fundamental en visión por computadora para evaluar la precisión de los modelos de detección y segmentación de objetos. Se calcula como la relación entre el área de intersección y el área de unión de las regiones predicha y real de un objeto en una imagen.

$$IoU = \frac{Área\ de\ intersección}{Área\ de\ unión} \quad (6)$$

Donde el área de Intersección, es donde la región predicha por el modelo y la región real (ground truth) del objeto se superponen. Esta área representa la coincidencia exacta entre la predicción y la realidad. Por otro lado, el área de unión es la combinada de las dos regiones, la predicha y la real. Se calcula sumando las áreas de ambas regiones y restando el área de intersección para evitar contar dos veces.

Mapas Auto Organizados de Kohonen

Los mapas auto organizados de Kohonen, también conocidos como redes de Kohonen, son un tipo de red neuronal artificial que se basa en el aprendizaje no supervisado para transformar datos de alta dimensión en un espacio de menor dimensión, generalmente bidimensional, mientras se conservan las relaciones topológicas inherentes en los datos originales.

En esencia, un SOM opera mediante un proceso iterativo en el que los "nodos" o "neuronas" en el mapa se ajustan para asemejarse a los patrones presentes en los datos de entrada. Durante el entrenamiento, cada nodo compite para representar un subconjunto de los datos, y el nodo ganador y sus vecinos se actualizan para acercarse al patrón de entrada. Este proceso se repite para cada dato de entrada, y con el tiempo, los nodos se organizan de manera que los nodos similares están cerca unos de otros en el mapa, formando así una representación visual de la estructura de los datos (Kohonen, T., 1982; Kohonen, T., 1995).

Inicialización

El proceso de inicialización en los SOM consiste en asignar valores iniciales a los vectores de pesos de las neuronas. Una práctica común es inicializar estos pesos con valores pequeños y aleatorios. Alternativamente, se pueden utilizar métodos como la inicialización basada en componentes principales, donde los pesos iniciales se seleccionan del espacio definido por los dos mayores autovectores de los componentes principales de los datos de entrada. Esta técnica puede acelerar la convergencia, ya que los pesos iniciales proporcionan una buena aproximación a la distribución final de los pesos del SOM (Kohonen, T., 2001).

Matriz BMU

Durante el entrenamiento de un SOM, para cada vector de entrada, se identifica la "Unidad de Mejor Correspondencia" (BMU, por sus siglas en inglés), que es la neurona cuyo vector de pesos es más similar al vector de entrada, generalmente determinado mediante la distancia euclidiana. La matriz BMU se refiere a la estructura que almacena las posiciones de estas unidades ganadoras en la cuadrícula del SOM para cada patrón de entrada. Esta matriz es esencial para analizar cómo los datos de alta dimensión se mapean en la representación de menor dimensión del SOM, facilitando la interpretación y visualización de la estructura de los datos (Kohonen, T., 2001).

Generación de Nubes de Puntos a Partir de Fotografías

La técnica de Structure from Motion (SFM) permite reconstruir estructuras tridimensionales a partir de secuencias de imágenes bidimensionales, estimando las posiciones de la cámara y la geometría de la escena mediante la identificación de puntos de interés comunes en múltiples vistas. Al combinar SfM con Multi-View Stereo (MVS), se logra una reconstrucción más detallada y precisa, ya que MVS utiliza múltiples imágenes para densificar la nube de puntos y capturar detalles finos de la

superficie. Esta integración es esencial en aplicaciones como la documentación 3D de huellas de dinosaurios, donde se requiere una representación detallada y precisa de las superficies estudiadas.

Además de la paleontología, la combinación de SFM y MVS se ha aplicado con éxito en la reconstrucción de entornos urbanos y en la documentación del patrimonio cultural. Por ejemplo, (Bódis-Szomorú et al.,2016) propusieron una infusión volumétrica eficiente de datos aéreos y terrestres para la reconstrucción urbana, demostrando la viabilidad de combinar diferentes fuentes de datos para obtener modelos 3D más completos y detallados. Asimismo, la fotogrametría basada en SfM y MVS se ha utilizado para digitalizar documentos históricos de manera no invasiva, proporcionando una metodología flexible y de bajo costo para la preservación digital del patrimonio cultural.

Segmentación a Partir de k-means

Es un método de clustering que consiste en ordenar los valores de intensidad de píxeles en una matriz *X*, donde *n* es el número de píxeles extraídos. A esta matriz se le aplica el algoritmo K-Means para determinar *k* grupos y así abordar la búsqueda de los centroides *mi* que, en efecto, constituyen la matriz *M* = [*m*1, *m*2, ..., *mk*]. Con ello se obtiene la media general μ de los centroides, partir de esta media se construye la matriz de dispersión entre conglomerados *B*.

$$B = \frac{1}{K-1}(M - \mu 1^T)(M - \mu 1^T)^T \tag{7}$$

Donde 1 es un vector columna de unos. De manera similar, para cada cluster i se define la submatriz X_i que contiene los datos asignados y se evalúa la dispersión interna W_i mediante.

$$W_i = \frac{1}{n_i - 1}(X_i - m_i 1_{n_i}^T)(X_i - m_i 1_{n_i}^T)^T \tag{8}$$

Donde n_i representado el número de píxeles del cluster i. La suma total de las dispersion internas se expresa con $W_i = \sum_i^K W_i$. Finalmente, el coeficiente de Fisher, que evalúa la calidad del agrupamiento, se define como.

$$J(k) = \frac{tr(B)}{tr(w)} \tag{9}$$

Donde tr, representa la traza de una matriz, es la suma total de la diagonal principal de abajo hacia arriba de una matriz. Esta cantidad aumenta cuando las demarcaciones son definidas y homogéneas internamente. Esto es vital para una partición exacta, es decir, dividir la información en categorías distintas.

Segmentación a Partir de Crecimiento de Regiones

También conocido como segmentación basada en regiones, consiste en la división de imágenes a partir de la detección de zonas homogéneas junto a propiedades definidas: intensidad o textura (Adams & Bischof, 1994). En primer lugar, se eligen píxeles semilla mediante unos criterios determinados y se evalúa la proximidad con los píxeles vecinos. Un píxel cumple con el criterio de similitud P(p, R) sí cumple con un umbral de diferencia de intensidad o bien está basado en estadísticas locales (González & Woods, 2002), en este caso, se añade a la región y se incorpora al conjunto de píxeles semilla. Este proceso es iterativo y se repite indefinidamente hasta que no se encuentren nuevos píxeles con homogeneidad. La elección de semillas y la definición del criterio adecuado son muy importantes para el éxito de este método (Haralick & Shapiro, 1985).

Figura 4.
Funcionamiento del método de crecimiento de regiones.

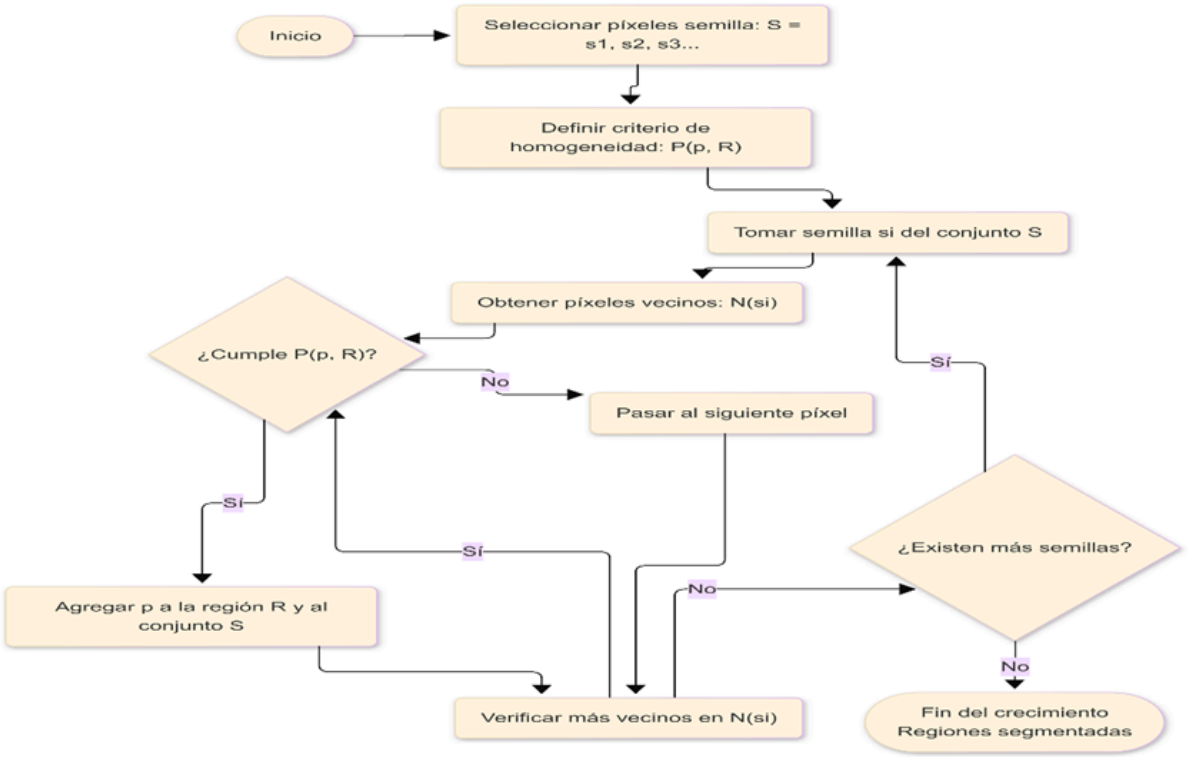
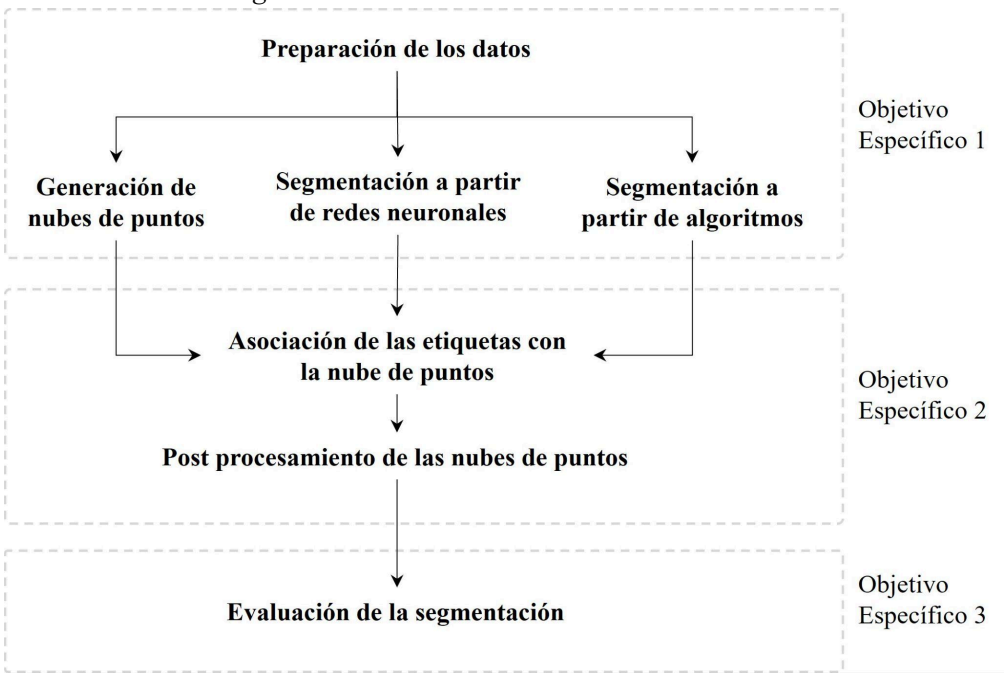


Figura 4, expone el funcionamiento del método de segmentación de crecimiento de regiones. Este proceso de expansión de las regiones se reitera hasta alcanzar las condiciones de parada definidas, las cuales pueden ser que una región crezca hasta cierto tamaño o que no pueda haber más píxeles que cumplan con los criterios de similitud. La metodología utilizada para este algoritmo se resalta la importancia de la selección de las semillas y la de los criterios de parada de las cuales se sirvieron los algoritmos de crecimiento de regiones para la segmentación de imágenes (Zucker, 1976).

Desarrollo Metodológico

Figura 5.
Flujo del desarrollo metodológico.

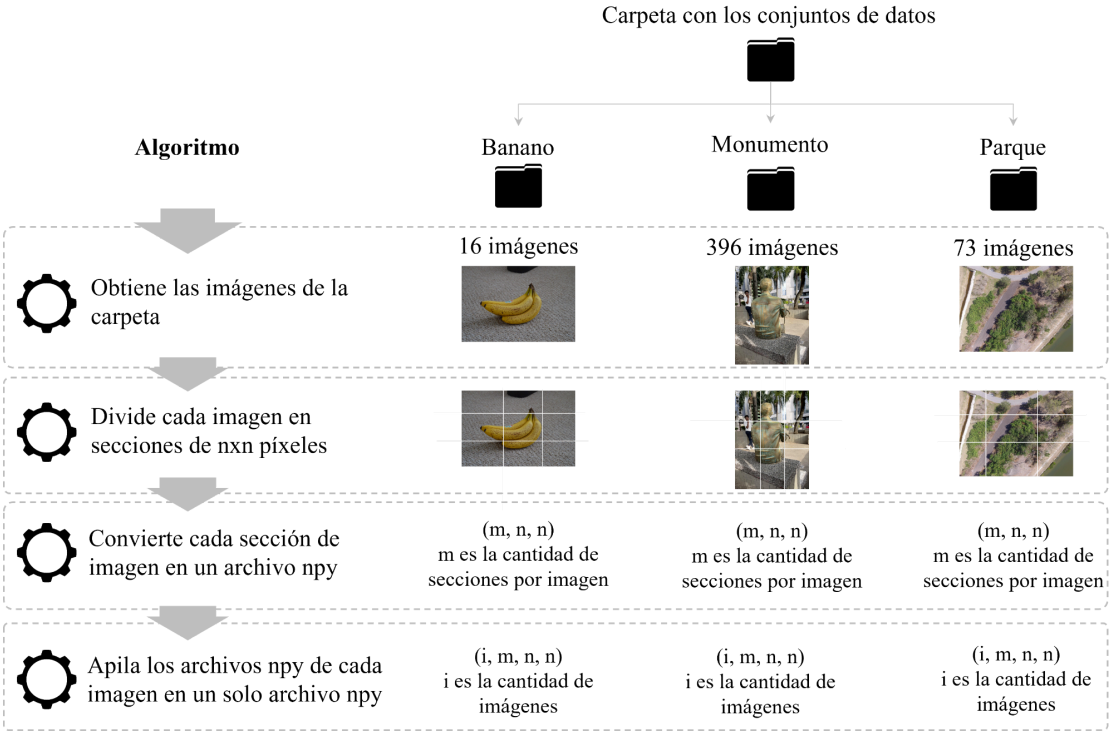


la figura 5, expone la metodología desarrollada en este trabajo se estructuró en cinco etapas principales, con el objetivo de asociar información semántica generada a partir de imágenes con nubes de puntos tridimensionales desprovistas de dicha información. Cabe aclarar, la metodología se desarrolló completamente con código en python, este está disponible en el repositorio accesible haciendo clic en el siguiente enlace, es importante considerar que los códigos empleados funcionan empleando módulos creados para este trabajo, estos se ubican en la carpeta 'resources'. Dentro del repositorio existen dos carpetas, una con el proyecto de segmentación de imágenes y otra con el proyecto de segmentación de nubes de puntos, en ambas se repite la estructura mencionada.

Preparación de los Datos

Se trabajó con tres conjuntos de imágenes en formato jpg correspondientes a diferentes objetos. Un conjunto de 16 imágenes capturando la escena de un banano, un conjunto de 396 imágenes correspondientes a un monumento y un conjunto de 73 imágenes capturando un parque. Las primeras dos fueron escenas tomadas con celulares y la tercera corresponde a fotografías de drone, donde la escena del monumento se obtuvo de autoría propia, mientras que, las otras dos fueron descargadas del sitio DroneDB como conjuntos libres.

Figura 6.
Algoritmo para la creación de datasets.

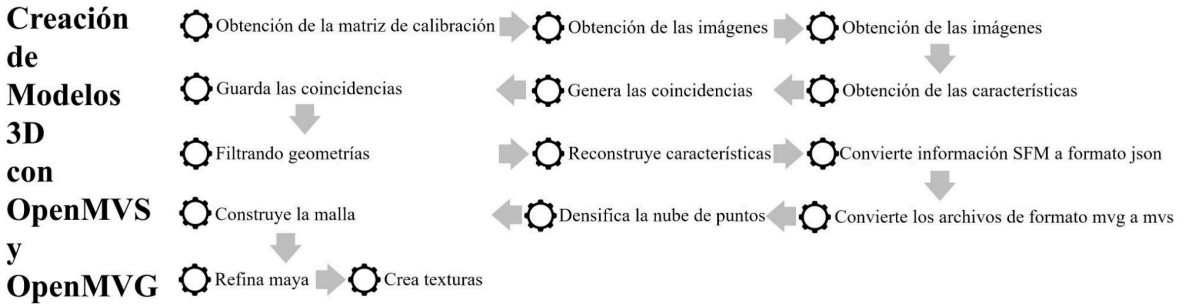


La figura 6, explica el algoritmo utilizado a partir del desarrollo librerías en Python utilizando módulos como cv2, PIL, entre otros, para automatizar el proceso de carga y procesamiento de las imágenes. Usando las librerías mencionadas, se accede a las carpetas de cada conjunto de datos, con lo que se extrae imágen por imágen, dividiéndolas en trozos de nxn píxeles y posteriormente apiladas en archivos numpy, generando los conjuntos de datos necesarios para entrenamiento y prueba. El código en cuestión se puede consultar haciendo clic en el siguiente enlace.

Para este estudio, no se utilizó el total de las imágenes para generar el conjunto de datos de entrenamiento, para la escena del banano se utilizaron 3 imágenes donde cada una se dividió en trozos de 20x20 píxeles, teniendo en cuenta que la resolución de cada imágen es de 3000x2000 píxeles, se obtuvieron 45000 muestras para el entrenamiento; para la escena del monumento se utilizaron 8 imágenes de 3024x4032 píxeles, divididas en secciones de 16x16 píxeles, dando un total de 381024 muestras de entrenamiento; por último, la escena del parque contempló 13 imágenes, divididas en secciones de 20x20 píxeles y teniendo cada una resolución de 4000x3000 píxeles, se obtuvo un conjunto de datos de 390000 muestras.

Generación de Nubes de Puntos

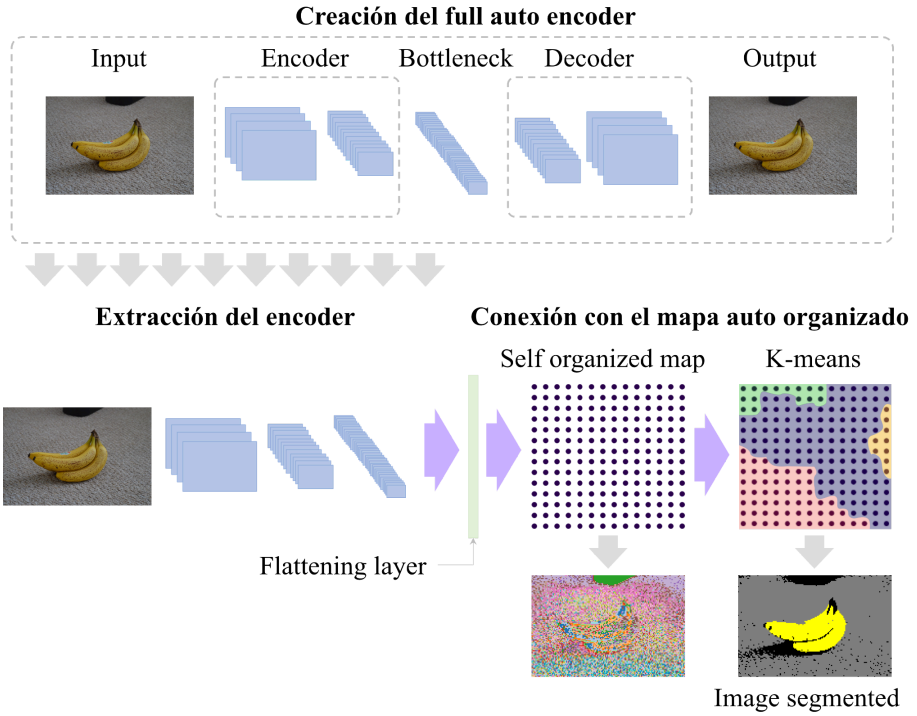
Figura 7.
Flujo de trabajo implementado con OpenMVG y OpenMVS.



La figura 7, expone el flujo de trabajo para la generación de modelos tridimensionales que se llevó a cabo mediante las librerías de C OpenMVG y OpenMVS, integradas en un flujo de trabajo en Python. Este proceso permitió obtener nubes de puntos no densificadas, densificadas, y el modelo 3d, además, un archivo JSON con la asociación entre puntos de la nube no densificada y los píxeles de las imágenes, así como otros datos auxiliares necesarios para las etapas posteriores. El código empleado en este proceso se puede consultar en el siguiente enlace.

Segmentación no Supervisada a Partir de Redes Neuronales

Figura 8.
Flujo de trabajo para obtener la segmentación no supervisada con redes neuronales.



La figura 8, resume la parte de la metodología se implementa la segmentación no supervisada, que consta de cuatro pasos principales, creación de full auto encoders, extracción de los encoder, conexión con el mapa auto organizado de Kohonen y refinamiento de la segmentación con K-means.

Creación del full auto encoder

Utilizando tensorflow se crearon funciones para definir arquitecturas tipo full auto encoder, de forma sencilla y flexible, donde los parámetros de las funciones permiten variar los parámetros e hiper parámetros (número de capas, dimensión de la entrada y salida y número de filtros). Adicionalmente, se empleó la librería optuna para entrenar y encontrar los modelos con mejores métricas de pérdida, variando el número de épocas, la tasa de aprendizaje inicial, el número de épocas de espera para la para temprana, dropout, regularización, número de kernels para la capa inicial, además de las características mencionadas anteriormente referentes a la arquitectura. El código empleado en este proceso se puede consultar en el siguiente enlace.

Extracción del encoder

Una vez encontrado el mejor modelo de full auto encoder para cada conjunto de datos, se extraen únicamente las capas hasta el cuello de botella, esto con la finalidad de obtener las características más representativas de cada imágen, añadido a esto, se agrega una capa de aplanamiento, esta capa toma los datos cuya dimensión contempla el alto y ancho de los mapas de características más el número de mapas de características en esta capa como tercera dimensión, y los organiza en un solo vector unidimensional, haciendo compatibles los datos para el entrenamiento de un mapa auto organizado de Kohonen. El código empleado en este proceso se puede consultar en el siguiente enlace.

Teniendo en cuenta lo anterior, se evalúan todas las imágenes de cada conjunto sobre el encoder, donde las salidas de este son los conjuntos de datos para entrenar los mapas auto organizados de Kohonen. El código empleado en este proceso se puede consultar en el siguiente enlace.

Conexión con el mapa auto organizado de Kohonen

Empleando la librería sompy, y de forma similar a la metodología de creación de full auto encoders, se desarrollan módulos con la finalidad de construir modelos de mapas de forma flexible, cambiando principalmente el método de inicialización, el número de épocas de entrenamiento fino y robusto, utilizando oportuna para ejecutar y guardar los modelos que mejor comportamiento mostraron. Se entrenaron mapas con una dimensión de 50x50 neuronas, buscando el error topográfico más bajo, obteniendo de cada mapa, los pesos asociados a cada neurona ganadora, según la matriz bmu. El código empleado en este proceso se puede consultar en el siguiente enlace.

Refinamiento con K-means

Una vez obtenidos los mapas, se pueden observar un número de clases máximo igual al número de neuronas, en este caso 2500. Si bien, lo anterior no implica que se vayan a observar toda esa cantidad de clases, es importante refinar esta clasificación. Para ello, se entrenaron modelos k-means con los datos de la matriz de pasos asociados a cada neurona ganadora, encontrando un número de clases elegido manualmente a partir de la interpretación de los resultados del mismo proceso. El mapa con los datos para clasificar las imágenes de la escena de bananos se definió para resumirse en 2 clases, 3 para la escena del monumento, mientras que, para el parque se usaron 3. El código empleado en este proceso se puede consultar en el siguiente enlace.

Segmentación no Supervisada a Partir de Algoritmos

Segmentación con Clustering

Respecto al método de clustering, se utilizó algoritmo K-Means y una metodología de optimización eficiente, con base en criterios de dispersión entre y dentro de clústeres (Martin H, et al. 2020). El enfoque tiene dos partes fundamentales, entrenar el modelo y segmentar imágenes. En una primera etapa se toman imágenes de muestra de una carpeta que será específica para el caso y deben

ser preprocesadas, se leen las imágenes, las que deben ser pasadas, si están en formato RGB a escala de grises utilizando la biblioteca skimage a fin de homogeneizar las imágenes frente a los análisis. El código empleado en este proceso se puede consultar en el siguiente enlace.

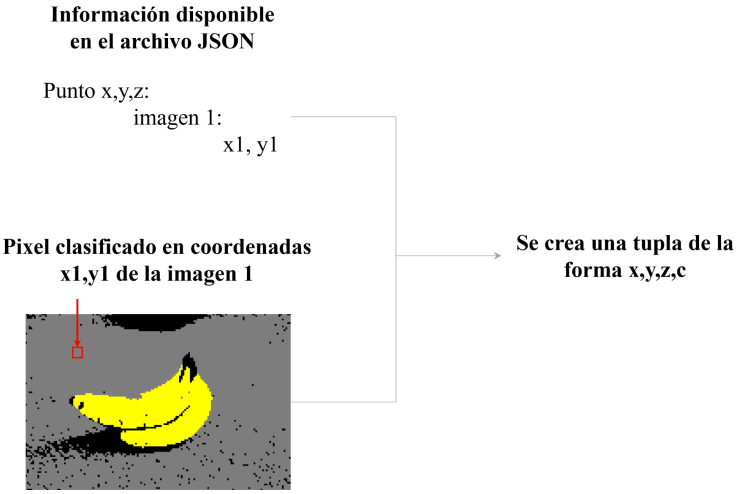
Se entrenaron modelos para cada conjunto de datos, utilizando cinco imágenes de muestra aleatorias, una vez entrenado el modelo, se realizaron las predicciones del modelo con el conjunto completo de imágenes, designando una cantidad de clases de 2 para la escena del banano; 3 para la escena del monumento; y 3 para la escena del parque, finalmente, se exporta cada imágen clasificada en formato npy.

Segmentación con Crecimiento de Regiones

En cuanto al método de segmentación de regiones, se implementó una rutina que no precisa de entrenamiento. Se obtienen las imágenes de cada escena desde las carpetas, sistemáticamente se convierten a escala de grises y se normalizan. Luego, se seleccionan semillas de forma aleatoria y se crea un arreglo con las mismas dimensiones de la imágen de entrada, formado completamente por ceros, posteriormente, se implementa el método de crecimiento de regiones usando la librería skimage, y se reescribe el arreglo mencionado. El método permite seleccionar la cantidad de clases a segmentar y exportar cada imágen segmentada en formato npy; designando una cantidad de clases de 2 para la escena del banano; 3 para la escena del monumento; y 3 para la escena del parque. El código empleado en este proceso se puede consultar en el siguiente enlace.

Asociación de Etiquetas con la Nube de Puntos

Figura 9.
Ejemplo de asignación de clase a un punto de la nube de puntos.



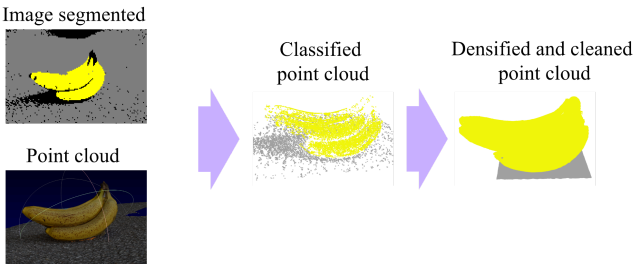
La figura 9, ejemplifica un caso de clasificación para un punto hipotético x,y,z dentro de la nube de puntos sin densificar, donde con la información del proceso de creación de la nube de puntos, se puede saber a qué imágen y pixel dentro de la misma está asociado cada punto, con lo que se consulta la coordenada del pixel en la imágen clasificada y se genera un arreglo de datos con las coordenadas x, y, z de un punto, más su clase asociada. El código empleado en este proceso se puede consultar en el siguiente enlace.

El proceso mencionado se replicó para todos los puntos de cada nube e iterando sobre las imágenes correspondientes usando Python. Cabe destacar, que este procedimiento no es posible realizarse en la nube de puntos densificada, dado que el archivo que contiene dicha información, se genera antes del proceso de densificación.

Post Procesamiento de la Nube de Puntos

Figura 10.

Diferencias entre la nube de puntos clasificada sin post procesar y post procesada.



La figura 10, muestra el proceso de obtener la nube de puntos clasificada y post procesada, al mismo tiempo, la diferencias entre la nube de puntos sin post procesar, donde aún no se ha hecho la densificación ni la limpieza de la misma.

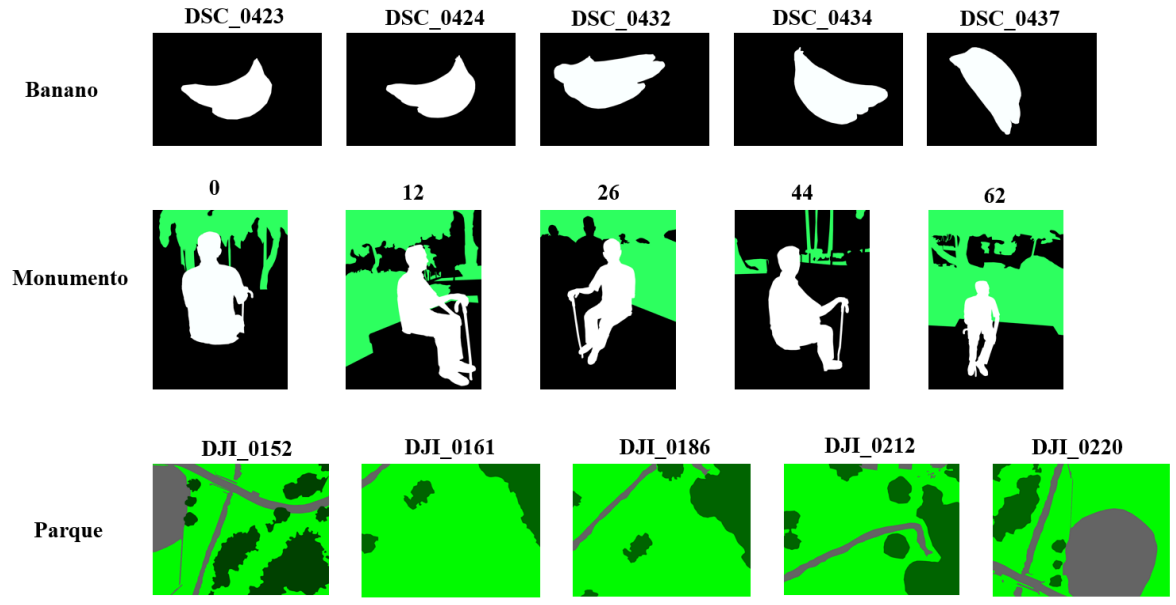
Las nubes de puntos clasificadas se separaron por clases y se almacenaron en archivos independientes, para ello, los arreglos de datos de la forma x, y, z, clase, se dividieron en subconjuntos donde solo existen registros con la misma clase, exportando en archivos numpy separados. Posteriormente, se densifican mediante un algoritmo que observa cada punto dentro de la nube de puntos sin densificar e identifica los n puntos más cercanos dentro de la nube de puntos densa, realizando este proceso para cada nube de puntos, se obtienen nubes de puntos densificada para cada clase. Finalmente, se empleó el software libre Meshlab para cargar y limpiar manualmente las nubes de puntos resultantes, de puntos residuales o que invaden espacio de los objetos de otras clases. Los códigos implementados para este proceso son los correspondientes al 5, 6, 7, 8 7 9 visibles en el siguiente enlace**.**

Evaluación de la Segmentación

La segmentación de las imágenes se evaluó creando etiquetas de forma manual como puntos de verdad terreno, para cada conjunto de datos se tomaron cinco imágenes representativas del conjunto completo, posteriormente, se transformaron en arreglos de numpy con la misma forma de las segmentaciones no supervisadas, generando datos comparables.

Figura 11.

Visualización de las imágenes etiquetadas manualmente.

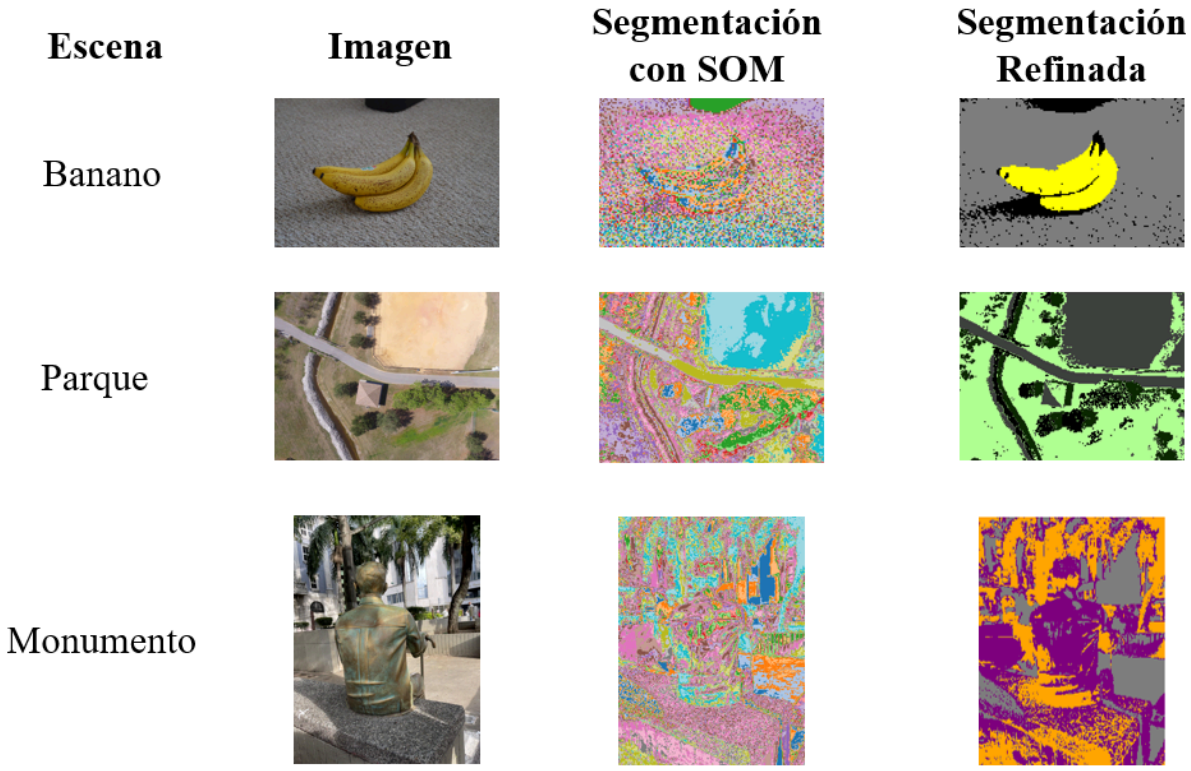


La figura 11, muestra las imagenes tomadas como grupo de control sobre las cuales se creó una segmentación manual, y se migraron a formato npy, donde cada color corresponde a una de las clases homólogas de los métodos implementados con anterioridad, brindando una matriz de clasificación comparable imagen por imagen respecto a los resultados de los métodos anteriores. El código empleado en este proceso se puede consultar en el siguiente enlace.

Se calculó la precisión de los píxeles, el índice de intersección sobre unión y las matrices de confusión de cada par de imágenes, a partir de los registros observados en las matrices de confusión, se calculó la precisión de los píxeles ahora en sobre cada clase en particular.

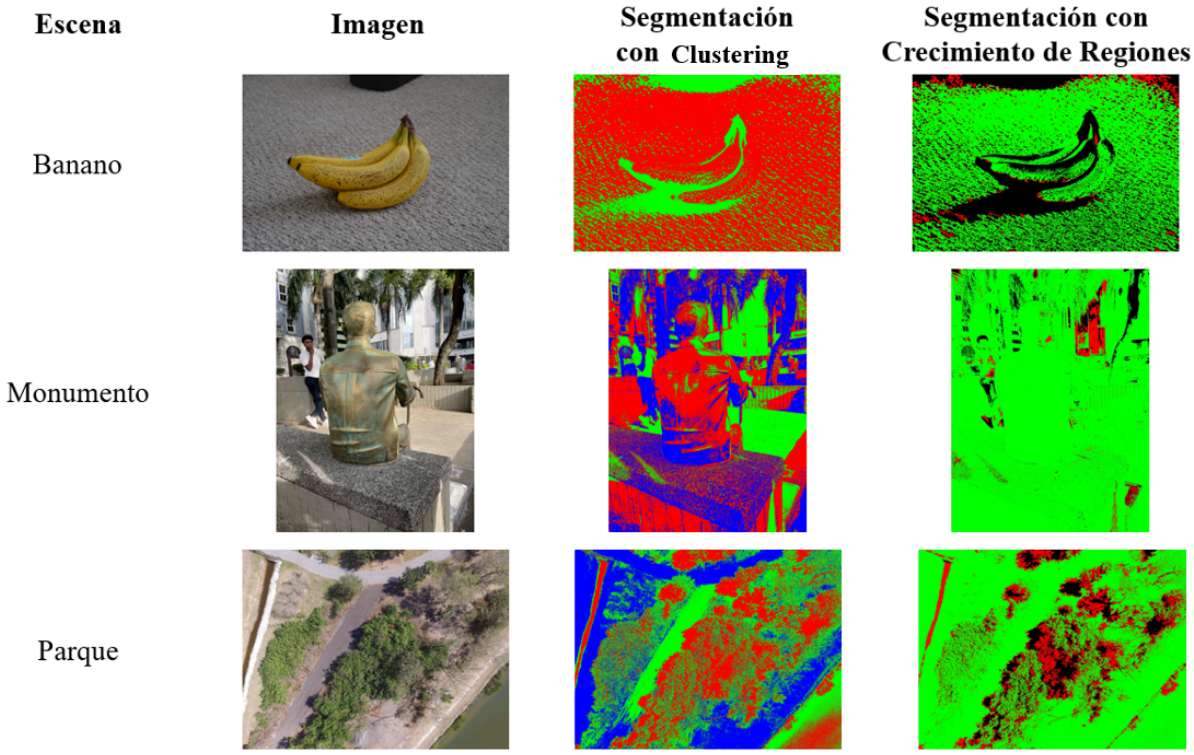
Resultados y Discusión

Segmentación no Supervisada con Redes Neuronales
Figura 12.
Resultados de segmentación a partir de redes neuronales.



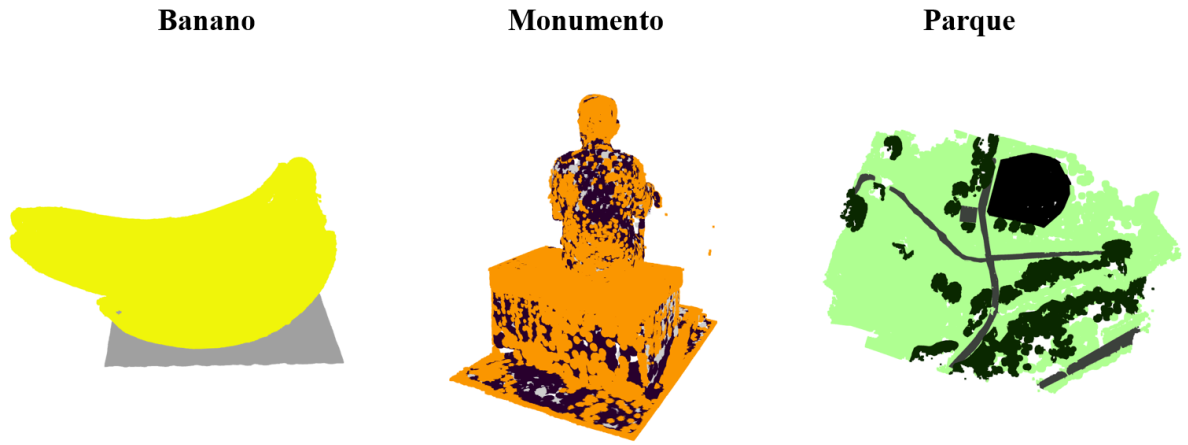
La figura 12, muestra un ejemplo de los resultados obtenidos, a partir de la segmentación semántica no supervisada, donde se contempla la clasificación del mapa auto organizado de Kohonen, la segmentación refinada usando K-means sobre los datos del mapa auto organizado, así como las nubes de puntos con la misma clasificación. A partir de las clasificaciones mencionadas, se identificaron diferentes objetos en las diferentes escenas; de las tres clases utilizadas para refinar la clasificación de la escena del banano, se resumieron en dos categorías mediante el post proceso manual, como lo fueron banano y suelo; para la escena del parque, se clasificaron 4 objetos, vegetación frondosa, vegetación baja, zonas construidas y arena; por último, para la escena del monumento, se usaron 3 clases más no se identificaron objetos completamente abarcado por alguna de estas clases, el monumento, el pavimento u otros objetos comparten regiones clasificadas de la misma manera.

Segmentación no Supervisada con Algoritmos
Figura 13.
Resultados de segmentación a partir de Algoritmos.



La figura 13, exhibe los resultados de la segmentación semántica no supervisada utilizando algoritmos tradicionales, observando un ejemplo de imagen por cada conjunto de datos o escenas, y sus homólogos correspondientes a las técnicas usadas.

Nubes de Puntos Segmentadas
Figura 14.
Nubes de puntos segmentadas.

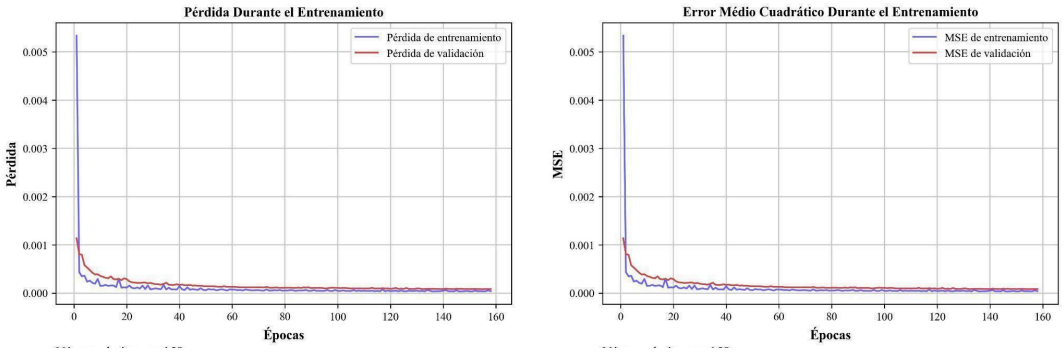


La figura 14, muestra una vista de las nubes de puntos clasificadas, las nubes de puntos se generaron con la segmentación realizada con redes neuronales.

Análisis del Full Auto Encoder

Gráfica 1.

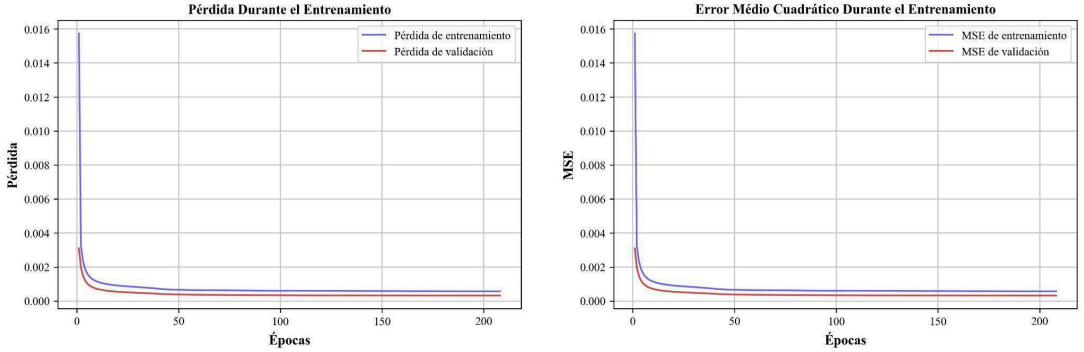
Gráficas del entrenamiento del Full Auto Encoder sobre la escena del banano.



La gráfica 1, muestra el comportamiento de la pérdida y error medio cuadrático a lo largo del entrenamiento del full auto encoder para la escena del banano, tanto en entrenamiento como en validación. El entrenamiento duró 158 épocas, alcanzando una pérdida mínima de 3.31e-05 en entrenamiento y de 8.1e-05 en validación, por otro lado, el error medio cuadrático mostró un valores iguales a los de la pérdida, diferenciándose apenas en el sexto decimal.

Gráfica 2.

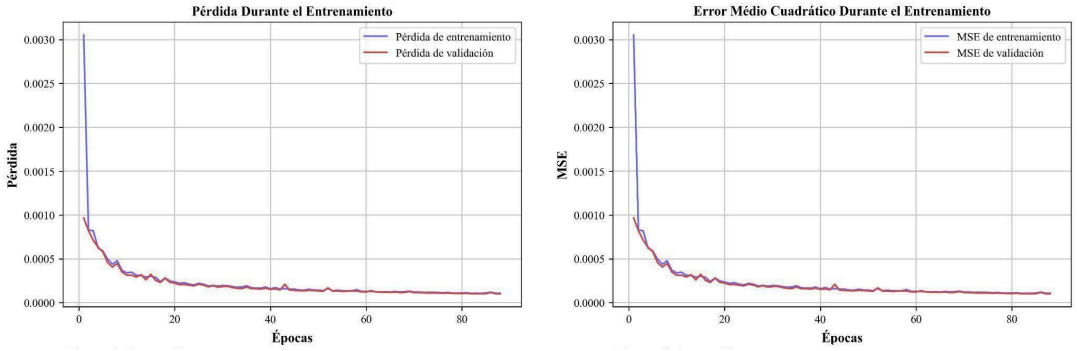
Gráficas del entrenamiento del Full Auto Encoder sobre la escena del monumento.



La gráfica 2, muestra el comportamiento de la pérdida y error medio cuadrático a lo largo del entrenamiento del full auto encoder para la escena del monumento, tanto en entrenamiento como en validación. El entrenamiento duró 208 épocas, alcanzando una pérdida mínima de 0.0005 en entrenamiento y de 0.0003 en validación, por otro lado, el error medio cuadrático mostró un valores iguales a los de la pérdida, diferenciándose apenas en el sexto decimal.

Gráfica 3.

Gráficas del entrenamiento del Full Auto Encoder sobre la escena del parque.



La gráfica 3, muestra el comportamiento de la pérdida y error medio cuadrático a lo largo del entrenamiento del full auto encoder para la escena del monumento, tanto en entrenamiento como en validación. El entrenamiento duró 88 épocas, alcanzando una pérdida mínima de 0.0001 en

entrenamiento y de 0.0001 en validación, por otro lado, el error medio cuadrático mostró un valores iguales a los de la pérdida, diferenciándose apenas en el sexto decimal.

Análisis de las Segmentaciones no Supervisadas a Partir de Redes Neuronales

Escena del banano

Tabla 1.
Métricas de evaluación para las imágenes de prueba en la escena del banano, usando redes neuronales.

Imagen	Precisión general de píxeles (%)	Clases	IoU	Precisión de la clase (%)
DSC_0423	98.0	1-Fondo	0.97	99.9
		2-Banano	0.86	86.7
DSC_0424	98.0	1-Fondo	0.97	99.9
		2-Banano	0.86	86.7
DSC_0432	97.0	1-Fondo	0.96	99.8
		2-Banano	0.85	86.6
DSC_0434	95.4	1-Fondo	0.95	99.9
		2-Banano	0.76	75.9
DSC_0437	97.8	1-Fondo	0.97	99.8
		2-Banano	0.87	87.8

La tabla 1, resume los resultados de la segmentación sobre el conjunto del banano donde se aprecian dos clases, "Fondo" y "Banano". La precisión general de píxeles oscila entre 95.4% y 98.0%, evidenciando una clasificación global robusta. Para la clase "Fondo", se obtienen valores excepcionales de IoU (0.95-0.97) y precisión (99.8%-99.9%), lo que refleja una identificación casi perfecta. En contraste, la clase "Banano" muestra un rendimiento inferior, con IoU entre 0.76-0.87 y precisión de 75.9%-87.8%, destacando DSC_0434 como el caso más desafiante (IoU: 0.76) y DSC_0437 como el mejor (IoU: 0.87). Estos resultados indican que el modelo es altamente efectivo para segmentar el fondo, mientras que la precisión en bananas, aunque sólida, sugiere espacio para optimizaciones, particularmente en casos con variabilidad morfológica o contextos complejos.

Escena del monumento

Tabla 2.
Métricas de evaluación para las imágenes de prueba en la escena del monumento, usando redes neuronales.

Imagen	Precisión general de píxeles (%)	Clases	IoU	Precisión de la clase (%)
0	40.0	0-Monumento	0.25	64.31
		1-Ambiente	0.27	30.26
		2-Vegetación	0.21	47.16
12	46.4	0-Monumento	0.20	44.43
		1-Ambiente	0.25	29.41
		2-Vegetación	0.42	64.39
26	40.9	0-Monumento	0.17	58.87
		1-Ambiente	0.33	37.14
		2-Vegetación	0.23	37.92
44	36.5	0-Monumento	0.20	55.73
		1-Ambiente	0.28	31.52
		2-Vegetación	0.07	41.01
62	23.6	0-Monumento	0.12	56.75
		1-Ambiente	0.12	19.61
		2-Vegetación	0.15	21.67

La tabla 2, resume los resultados de la segmentación sobre el conjunto del monumento donde se aprecian tres clases, "Monumento", "Ambiente" y "Vegetación", evaluado en cinco imágenes. La precisión general de píxeles varía notablemente (23.6%-46.4%), reflejando un rendimiento moderado con fluctuaciones significativas entre imágenes. La clase "Monumento" muestra una precisión de clase relativamente estable (44.43%-64.31%) pero un IoU bajo (0.12-0.25), indicando superposición limitada entre predicciones y anotaciones reales. La clase "Vegetación" destaca en la imagen 12 (IoU: 0.42; precisión: 64.39%), pero presenta resultados críticos en la imagen 44 (IoU: 0.07) y 62 (precisión: 21.67%), sugiriendo dificultades en contextos complejos o escasa representación de datos. La clase "Ambiente" registra el rendimiento más bajo, con precisión ≤37.14% e IoU ≤0.33, evidenciando desafíos en su discriminación. Estos resultados subrayan la necesidad de optimizar el modelo, especialmente para clases minoritarias o escenarios con alta variabilidad contextual.

Escena del parque

Tabla 3.

Métricas de evaluación para las imágenes de prueba en la escena del parque, usando redes neuronales.

Imagen	Precisión general de píxeles (%)	Clases	IoU	Precisión de la clase (%)
DJI_0152	40.7	1-Zona artificial	0.22	38.13
		2-Vegetación baja	0.30	41.27
		3-Vegetación Alta	0.21	43.41
DJI_0161	88.4	1-Zona artificial	0.13	83.96
		2-Vegetación baja	0.87	94.21
		3-Vegetación Alta	0.43	46.08
DJI_0186	77.5	1-Zona artificial	0.15	79.27
		2-Vegetación baja	0.74	82.17
		3-Vegetación Alta	0.55	57.15
DJI_0212	76.0	1-Zona artificial	0.27	41.48
		2-Vegetación baja	0.74	92.42
		3-Vegetación Alta	0.36	41.92
DJI_0220	79.3	1-Zona artificial	0.68	87.55
		2-Vegetación baja	0.69	78.55
		3-Vegetación Alta	0.40	54.57

La tabla 3, resume los resultados de la segmentación sobre el conjunto del parque donde se aprecian tres clases, ("Zona artificial", "Vegetación baja" y "Vegetación Alta") en cinco imágenes aéreas. La precisión general de píxeles varía significativamente entre las imágenes, desde un 40.7% (DJI_0152) hasta un 88.4% (DJI_0161), indicando una alta dependencia del contexto escénico. La clase "Vegetación baja" muestra el rendimiento más consistente, con valores destacados en IOU (0.87 en DJI_0161) y precisión de clase (94.21% en la misma imagen). En contraste, la "Zona artificial" presenta un IOU generalmente bajo (≤0.27 en cuatro imágenes), excepto en DJI_0220 (IOU: 0.68; precisión: 87.55%), lo que sugiere dificultades en su identificación salvo en condiciones específicas. La "Vegetación Alta" exhibe resultados mixtos, con un IOU máximo de 0.55 (DJI_0186) y precisiones moderadas (41.92%-57.15%), reflejando desafíos en su discriminación, posiblemente por variabilidad morfológica o solapamiento con otras clases. Estos resultados subrayan la necesidad de mejorar la generalización del modelo, especialmente para clases con alta heterogeneidad espacial o escasa representación en ciertos escenarios.

Análisis de las Segmentaciones no Supervisadas a Partir de Clustering

Escena del banano

Tabla 4.
Métricas de evaluación para las imágenes de prueba en la escena del banano, usando clustering.

Imagen	Precisión general de píxeles (%)	Clases	IoU	Precisión de la clase (%)
DSC_0423	70.4	1-Fondo	0.68	76.47
		2-Banano	0.14	35.85
DSC_0424	70.2	1-Fondo	0.68	76.47
		2-Banano	0.15	36.08
DSC_0432	70.8	1-Fondo	0.68	79.42
		2-Banano	0.20	37.65
DSC_0434	63.0	1-Fondo	0.60	67.35
		2-Banano	0.20	51.53
DSC_0437	52.5	1-Fondo	0.52	60.00
		2-Banano	0.08	22.69

La tabla 4, muestra los resultados de segmentación en imágenes de banano, evaluando dos clases: "Fondo" y "Banano". La precisión general varía entre 52.5% y 70.8%, con un rendimiento moderado. La clase "Fondo" tiene una precisión de 60.00%-79.42% y un IoU de 0.52-0.68, indicando una buena superposición. En cambio, la clase "Banano" presenta dificultades, con una precisión de 22.69%-51.53% y un IoU de 0.08-0.20, sugiriendo problemas en la segmentación. La imagen DSC_0437 tiene el peor rendimiento (precisión general: 52.5%; precisión "Banano": 22.69%), mientras que DSC_0432 tiene el mejor (precisión general: 70.8%; precisión "Banano": 37.65%). Estos resultados resaltan la necesidad de mejorar el modelo, especialmente para la clase "Banano", posiblemente con más datos de entrenamiento o ajustes en el modelo para manejar mejor la variabilidad en las escenas.

Escena del monumento

Tabla 5.
Métricas de evaluación para las imágenes de prueba en la escena del monumento, usando clustering.

Imagen	Precisión general de píxeles (%)	Clases	IoU	Precisión de la clase (%)
0	42.8	0-Monumento	0.26	65.32
		1-Ambiente	0.28	28.95
		2-Vegetación	0.28	64.66
12	53.9	0-Monumento	0.23	56.15
		1-Ambiente	0.33	41.58
		2-Vegetación	0.58	66.16
26	36.3	0-Monumento	0.17	10.69
		1-Ambiente	0.23	26.09
		2-Vegetación	0.25	47.06
44	0.35	0-Monumento	0.18	51.78
		1-Ambiente	0.26	28.26
		2-Vegetación	0.14	70.62
62	48.4	0-Monumento	0.12	52.13
		1-Ambiente	0.28	33.33
		2-Vegetación	0.47	60.77

La tabla 5, resume los resultados de segmentación en imágenes de monumentos, evaluando tres clases: "Monumento", "Ambiente" y "Vegetación". La precisión general de píxeles varía entre 0.35% y 53.9%, mostrando un rendimiento moderado con fluctuaciones significativas. La clase

"Monumento" tiene una precisión de clase relativamente estable (10.69%-65.32%) pero un IoU bajo (0.12-0.26), indicando superposición limitada. La clase "Vegetación" destaca en la imagen 12 (IoU: 0.58; precisión: 66.16%), pero tiene resultados críticos en la imagen 44 (IoU: 0.14). La clase "Ambiente" presenta el rendimiento más bajo, con precisión ≤41.58% e IoU ≤0.33, evidenciando dificultades en su discriminación. Estos resultados subrayan la necesidad de optimizar el modelo, especialmente para clases minoritarias o escenarios complejos.

Escena del parque

Tabla 6.

Métricas de evaluación para las imágenes de prueba en la escena del parque, usando clustering.

Imagen	Precisión general de píxeles (%)	Clases	IoU	Precisión de la clase (%)
DJI_0152	40.4	1-Zona artificial	0.22	45.73
		2-Vegetación baja	0.32	43.75
		3-Vegetación Alta	0.17	30.62
DJI_0161	81.5	1-Zona artificial	0.07	84.45
		2-Vegetación baja	0.80	84.15
		3-Vegetación Alta	0.44	59.65
DJI_0186	68.9	1-Zona artificial	0.12	80.35
		2-Vegetación baja	0.63	69.50
		3-Vegetación Alta	0.47	62.13
DJI_0212	71.4	1-Zona artificial	0.24	51.50
		2-Vegetación baja	0.65	73.31
		3-Vegetación Alta	0.47	70.35
DJI_0220	72.2	1-Zona artificial	0.64	89.96
		2-Vegetación baja	0.55	58.46
		3-Vegetación Alta	0.40	83.64

La tabla 6, resume los resultados de segmentación en imágenes de parques, evaluando tres clases: "Zona artificial", "Vegetación baja" y "Vegetación alta". La precisión general de píxeles varía entre 40.4% y 81.5%, mostrando un rendimiento variable. La clase "Zona artificial" tiene una precisión de clase entre 45.73% y 89.96%, pero un IoU bajo (0.07-0.64), indicando superposición limitada. La clase "Vegetación baja" destaca en la imagen DJI_0161 (IoU: 0.80; precisión: 84.15%), pero presenta resultados menos consistentes en otras imágenes. La clase "Vegetación alta" muestra una precisión de clase entre 30.62% y 83.64%, con un IoU que varía de 0.17 a 0.47. Estos resultados sugieren la necesidad de mejorar el modelo, especialmente para la clase "Vegetación alta", que presenta mayor variabilidad en su segmentación.

Análisis de las Segmentaciones no Supervisadas a Partir de Crecimiento de Regiones

Escena del banano

Tabla 7.
Métricas de evaluación para las imágenes de prueba en la escena del banano, usando crecimiento de regiones.

Imagen	Precisión general de píxeles (%)	Clases	IoU	Precisión de la clase (%)
DSC_0423	19.0	1-Fondo	0.10	10.20
		2-Banano	0.11	72.11
DSC_0424	67.0	1-Fondo	0.64	68.63
		2-Banano	0.19	55.93
DSC_0432	64.7	1-Fondo	0.58	60.42
		2-Banano	0.31	78.39
DSC_0434	62.2	1-Fondo	0.56	59.18
		2-Banano	0.27	75.15
DSC_0437	45.5	1-Fondo	0.35	36.00
		2-Banano	0.21	90.13

La tabla 7, resume los resultados de segmentación en imágenes de banano, evaluando dos clases: "Fondo" y "Banano". La precisión general de píxeles varía entre 19.0% y 67.0%, mostrando un rendimiento irregular. La clase "Fondo" tiene una precisión de clase entre 10.20% y 68.63%, con un IoU que oscila entre 0.10 y 0.64, indicando superposición limitada en algunas imágenes. La clase "Banano" presenta una precisión de clase más alta (55.93%-90.13%) y un IoU entre 0.11 y 0.31, sugiriendo una mejor identificación de los bananos, aunque con superposición variable. La imagen DSC_0437 destaca con una precisión de clase "Banano" del 90.13%, pero la precisión general más baja se observa en DSC_0423 (19.0%). Estos resultados indican la necesidad de mejorar el modelo, especialmente para la clase "Fondo", que muestra un rendimiento inconsistente.

Escena del monumento

Tabla 8.
Métricas de evaluación para las imágenes de prueba en la escena del monumento, usando crecimiento de regiones.

Imagen	Precisión general de píxeles (%)	Clases	IoU	Precisión de la clase (%)
0	23.2	0-Monumento	0.22	99.21
		1-Ambiente	0.04	4.75
		2-Vegetación	0.00	0.79
12	21.6	0-Monumento	0.22	98.93
		1-Ambiente	0.01	1.92
		2-Vegetación	0.01	2.00
26	26.0	0-Monumento	0.16	94.13
		1-Ambiente	0.19	20.29
		2-Vegetación	0.00	0.00
44	21.6	0-Monumento	0.20	98.67
		1-Ambiente	0.02	2.19
		2-Vegetación	0.15	18.55
62	31.7	0-Monumento	0.10	65.43
		1-Ambiente	0.42	62.75
		2-Vegetación	0.00	0.00

La tabla 8, resume los resultados de segmentación en imágenes de monumentos, evaluando tres clases: "Monumento", "Ambiente" y "Vegetación". La precisión general de píxeles varía entre 21.6% y 31.7%, mostrando un rendimiento bajo y fluctuante. La clase "Monumento" tiene una precisión de clase alta (65.43%-99.21%) pero un IoU bajo (0.10-0.22), indicando superposición limitada. La clase "Ambiente" presenta una precisión de clase muy baja (1.92%-62.75%) y un IoU entre 0.01 y 0.42, evidenciando dificultades en su identificación. La clase "Vegetación" tiene el peor

rendimiento, con una precisión de clase cercana a 0% en la mayoría de las imágenes y un IoU de 0.00 en varios casos, sugiriendo una representación insuficiente o problemas en la segmentación. Estos resultados resaltan la necesidad de mejorar el modelo, especialmente para las clases "Ambiente" y "Vegetación", que muestran un desempeño crítico.

Escena del parque

Tabla 9.
Métricas de evaluación para las imágenes de prueba en la escena del parque, usando crecimiento de regiones.

Imagen	Precisión general de píxeles (%)	Clases	IoU	Precisión de la clase (%)
DJI_0152	42.5	1-Zona artificial	0.06	9.46
		2-Vegetación baja	0.42	70.45
		3-Vegetación Alta	0.08	12.54
DJI_0161	88.4	1-Zona artificial	0.06	0.92
		2-Vegetación baja	0.88	99.09
		3-Vegetación Alta	0.14	14.78
DJI_0186	79.7	1-Zona artificial	0.00	0.16
		2-Vegetación baja	0.80	96.33
		3-Vegetación Alta	0.22	24.60
DJI_0212	63.3	1-Zona artificial	0.05	6.49
		2-Vegetación baja	0.65	92.80
		3-Vegetación Alta	0.00	0.57
DJI_0220	57.2	1-Zona artificial	0.00	0.31
		2-Vegetación baja	0.55	94.45
		3-Vegetación Alta	0.45	58.83

La tabla 9, resume los resultados de segmentación en imágenes de parques, evaluando tres clases: "Zona artificial", "Vegetación baja" y "Vegetación alta". La precisión general de píxeles varía entre 42.5% y 88.4%, mostrando un rendimiento variable. La clase "Vegetación baja" tiene un rendimiento destacado, con una precisión de clase entre 70.45% y 99.09% y un IoU entre 0.42 y 0.88, indicando una buena superposición. En contraste, la clase "Zona artificial" presenta un rendimiento muy bajo, con una precisión de clase entre 0.16% y 9.46% y un IoU entre 0.00 y 0.06, sugiriendo dificultades en su identificación. La clase "Vegetación alta" muestra un rendimiento inconsistente, con una precisión de clase entre 0.57% y 58.83% y un IoU entre 0.00 y 0.45. Estos resultados resaltan la necesidad de mejorar el modelo, especialmente para las clases "Zona artificial" y "Vegetación alta", que presentan un desempeño crítico.

Discusión de resultados

Desempeño de la red neuronal y algoritmo de Segmentación Clustering en la escena del Banano

La tabla 1, presenta los resultados de segmentación del conjunto generado por la red neuronal, evidenciando una robustez del modelo obtenido con una precisión de píxel de ≤ 98%. En otras palabras, el rendimiento de la red neuronal es sólido, sobre todo en la segmentación de la clase “fondo”, que muestra métricas de IoU de dicha clase del ≤ 99.9%, lo que denota una efectividad notable al superponer la región de fondo. En cuanto a la segmentación basada en clustering, reflejada en la (tabla 4), los resultados obtenidos muestran un rendimiento general inferior en comparación con las redes neuronales, ya que la precisión de las imágenes oscila entre 52,5% y 70,8%. Esto indica que el modelo de clustering no logra segmentar correctamente la imagen, especialmente la clase “Banano”, a diferencia de la precisión y el IoU para la clase fondo, que son relativamente buenos, con valores entre 60% y 79.42% en precisión y un IoU de 0.52 a 0.68. Estos resultados son consistentes con la observación de que el clustering puede identificar de manera aceptable, aunque no al nivel de exactitud de las redes neuronales.

Sin embargo, la clase “Banano” enfrenta retos considerables con un IoU notablemente bajo que va de 0.08 a 0.20, lo que sugiere que el modelo tiene dificultades significativas para segmentar la

clase de interés, posiblemente porque el clustering no captura adecuadamente las variaciones en la banana, como forma, color o tamaño, especialmente en entornos más complejos o con sombras cambiantes. En contraste, las redes neuronales son más efectivas en clasificar el fondo y muestran una precisión superior en la clase "Banano", aunque aún con espacio para mejorar para mejorar el rendimiento en la clase "Banano", podría ser útil explorar técnicas de aumento de datos, incorporar más variabilidad en las imágenes de entrenamiento o mejorar la arquitectura del modelo para capturar mejor las características morfológicas de las bananas.

Desempeño de la red neuronal y algoritmo de segmentación de clustering en la escena del monumento

En términos generales, los resultados de la segmentación en la escena del monumento evidencian un rendimiento moderado, con variaciones significativas entre imágenes y clases, tanto al utilizar redes neuronales como métodos de *clustering*. En el caso de las redes neuronales (tabla 2), la exactitud total de píxeles varía entre 23.6% y 46.4%, y se observa que la clase “Monumento” tiene una precisión de clase bastante estable (44.43%-64.31%) aunque con valores de IoU bajos (0.12-0.25). La categoría “Vegetación” alcanza resultados favorables en la imagen 12 (IoU: 0.42; precisión: 64.39%) pero enfrenta retos en contextos más complicados o con representación limitada de datos, como se ve en la imagen 44 (IoU: 0.07) y 62 (precisión: 21.67%). La clase "Ambiente", por su parte, presenta los valores más bajos (precisión ≤37.14%, IoU ≤0.33), lo que muestra retos adicionales para su correcta identificación.

Al analizar las mismas clases utilizando clustering (tabla 5), se aprecia también un amplio rango en la precisión global de píxeles (0.35%-53.9%), lo que indica variaciones significativas en la segmentación. La categoría “Monumento” también presenta una exactitud constante (10,69%-65,32%) pero exhibe un IoU bajo (0,12-0,26). La categoría “Vegetación” resalta principalmente en la imagen 12 (IoU: 0.58; precisión: 66.16%), pero nuevamente se ve perjudicada en la imagen 44 (IoU: 0.14). La categoría “Ambiente” sigue siendo la más complicada de segmentar, con precisión ≤41,58% y IoU ≤0,33. Estos resultados enfatizan la importancia de mejorar los algoritmos, especialmente para las clases menos representadas o en situaciones con alta variabilidad contextual, con el objetivo de aumentar la precisión y la coincidencia efectiva entre las predicciones y las anotaciones reales.

Desempeño de la red neuronal y algoritmo de segmentación clustering en la escena del parque

En el caso de la segmentación en la escena del parque, tanto con redes neuronales (tabla 3) como con *clustering* (tabla 6), los resultados muestran un desempeño variable y dependiente del contexto escénico. Con redes neuronales, la precisión general de píxeles oscila entre el 40.7% (DJI_0152) y el 88.4% (DJI_0161), mientras que con *clustering* fluctúa entre el 40.4% y el 81.5%, evidenciando retos de generalización en ambos métodos. La clase “Vegetación baja” tiende a presentar un rendimiento más sólido, destacando especialmente en la imagen DJI_0161 (IoU: 0.87 y 94.21% de precisión con redes neuronales; IoU: 0.80 y 84.15% de precisión con clustering).

Por el contrario, la “Zona artificial” muestra un IoU generalmente bajo (≤0.27) salvo en casos puntuales (DJI_0220), lo que sugiere que solo en condiciones muy definidas se logra una discriminación adecuada. La clase “Vegetación alta” exhibe resultados más irregulares, con IoU en redes neuronales que alcanza un máximo de 0.55 y precisiones entre 41.92%-57.15%, mientras que con *clustering* su IoU oscila entre 0.17 y 0.47. Este comportamiento apunta a desafíos debidos a la alta heterogeneidad espacial o la similitud con otras clases. En consecuencia, se hace patente la necesidad de optimizar ambas metodologías, sobre todo para aquellas clases con mayor complejidad morfológica o menor representación en determinadas escenas, a fin de mejorar la robustez y estabilidad de la segmentación.

Desempeño de la red neuronal y algoritmo de segmentación crecimiento de regiones en la escena del banano

En la comparación de los resultados de segmentación en la escena del banano, se observa un desempeño claramente diferenciado entre el uso de redes neuronales (tabla 1) y el algoritmo de crecimiento de regiones (tabla 7). Al emplear redes neuronales, la precisión general de píxeles alcanza valores elevados (95.4%-98.0%), con una identificación casi perfecta del "Fondo" (IoU: 0.95-0.97; precisión: 99.8%-99.9%). Sin embargo, la clase "Banano" muestra un rendimiento ligeramente menor (IoU: 0.76-0.87; precisión: 75.9%-87.8%), lo cual sugiere la necesidad de optimizaciones específicas para casos con alta variabilidad morfológica o contextos complejos.

Por otro lado, el enfoque de crecimiento de regiones presenta una precisión global más irregular (19.0%-67.0%) y mayores fluctuaciones en la identificación del "Fondo" (IoU: 0.10-0.64; precisión: 10.20%-68.63%). Aunque la clase "Banano" exhibe una precisión de clase relativamente alta (55.93%-90.13%), los valores de IoU (0.11-0.31) evidencian una superposición limitada entre predicciones y anotaciones. En conjunto, estos hallazgos resaltan la efectividad de las redes neuronales para lograr una segmentación global sólida, al tiempo que ponen de manifiesto la necesidad de mejorar el algoritmo de crecimiento de regiones principalmente en la diferenciación del fondo y de continuar afinando ambos métodos en escenarios con alta variabilidad de forma y textura.

Desempeño de la red neuronal y algoritmo de segmentación crecimiento de regiones en la escena de monumento

En la comparación de los resultados de segmentación en la escena del monumento, se notan diferencias evidentes en el rendimiento entre las redes neuronales (tabla 2) y el crecimiento de regiones (tabla 8). Con redes neuronales, la exactitud general de los píxeles oscila entre 23,6% y 46,4%, mostrando un desempeño moderado con variaciones. A pesar de que la categoría "Monumento" presenta una precisión de clase bastante constante (44.43%-64.31%), el bajo IoU (0.12-0.25) sugiere una intersección limitada entre las predicciones y las anotaciones. La categoría "Vegetación" muestra un rendimiento destacado en la imagen 12 (IoU: 0.42; precisión: 64.39%), pero enfrenta retos en situaciones complejas o con poca representación, como se observa en la imagen 44 (IoU: 0.07). Mientras tanto, la categoría "Ambiente" presenta el rendimiento más bajo (precisión $\leq$37.14%, IoU $\leq$0.33), lo que demuestra la necesidad de mejorar la identificación de áreas más complejas. En contraste, el enfoque de expansión de regiones presenta una precisión global aúnmenor (21,6%-31,7%), con índices de IoU muy bajos en todas las categorías.

No obstante, la categoría "Monumento" logra una alta precisión (65,43%-99,21%), el IoU restringido (0,10-0,22) resalta problemas de sobreposición. La clase "Ambiente" muestra un extenso rango de precisión (1.92%-62.75%) y valores de IoU que indican dificultades en diversas imágenes, mientras que la clase "Vegetación" tiene el menor rendimiento (precisión casi 0% en la mayoría de los casos, IoU 0.00). Estos descubrimientos subrayan la importancia de optimizar ambos enfoques, pero sobre todo se destaca la necesidad urgente de perfeccionar la categorización de "Ambiente" y "Vegetación" al utilizar el crecimiento de regiones, así como de ajustar las redes neuronales para conseguir una mejor coincidencia entre las predicciones y las anotaciones en situaciones complicadas.

Desempeño de la red neuronal y algoritmo de segmentación crecimiento de regiones en la escena de parque

En este análisis sobre la escena del parque, notamos que tanto el modelo basado en redes neuronales (tabla 3) como el método de crecimiento de regiones (tabla 9) ofrecen un desempeño bastante variable, muy influenciado por las particularidades de cada imagen.

Con las redes neuronales, la precisión global de píxeles va desde un 40.7% (DJI_0152) hasta un 88.4% (DJI_0161). Llama la atención que la clase "Vegetación baja" sea la más consistente (por ejemplo, con un IoU de 0.87 en DJI_0161), mientras que la "Zona artificial" presenta más problemas

(IoU ≤0.27) excepto en casos específicos (como DJI_0220). La “Vegetación alta”, por su parte, muestra resultados intermedios, con un IoU máximo de 0.55 y precisiones entre 41.92% y 57.15%, lo que indica que aún hay margen para mejorar la clasificación en escenarios con mayor complejidad o donde las clases se solapan.

En cuanto al método de crecimiento de regiones, la precisión global fluctúa entre el 42.5% y el 88.4%. Al igual que con las redes neuronales, la clase “Vegetación baja” vuelve a destacar, con precisiones de 70.45% hasta 99.09% y un IoU que oscila entre 0.42 y 0.88, reflejando una buena correspondencia entre las segmentaciones propuestas y las anotaciones reales. Sin embargo, la “Zona artificial” presenta un desempeño muy bajo (con precisiones por debajo del 9.46% y un IoU entre 0.00 y 0.06), evidenciando grandes dificultades para identificar correctamente estas áreas. La “Vegetación alta” también ofrece resultados irregulares, con precisiones que van del 0.57% al 58.83% y un IoU que varía entre 0.00 y 0.45, lo cual confirma la complejidad de segmentar este tipo de vegetación.

En conjunto, los resultados evidencian que la “Zona artificial” sigue siendo un reto en determinadas condiciones y que la “Vegetación alta” necesita un tratamiento más especializado ante su alta variabilidad o superposición con otras clases. Por el contrario, la “Vegetación baja” tiende a obtener valores más estables, quizá debido a una apariencia más homogénea o a una mejor representación dentro del conjunto de datos. A la luz de estos hallazgos, es evidente que se requieren ajustes adicionales, ya sea perfeccionando el entrenamiento de las redes neuronales o afinando los parámetros del algoritmo de crecimiento de regiones, para lograr un mejor equilibrio en la segmentación de todas las clases.

En la propuesta de Su et al. (2015), plantea utilizar redes neuronales convolucionales multivista (MVCNN) con el propósito de poder llevar a cabo la tarea de reconocimiento de formas 3D, logrando precisiones de las del 90% en la clasificación de modelos tridimensionales (por ejemplo, de la colección ModelNet40). Si bien el objetivo de Su et al. (2015) está orientado a obtener un modelo de clasificación y no a la segmentación pixel a pixel, el principio clave que termina de decantar la balanza reside en el hecho de poder aprovechar múltiples vistas y combinarlo con un entrenamiento robusto que permita alcanzar la variabilidad del objeto en estudio.

Por otro lado, y en contraposición a nuestros resultados de segmentación 2D (por ejemplo, hasta un 98% de precisión de fondo en la escena del banano) uno podría percibir un comportamiento muy similar, ya que cuando se tienen suficientes datos (y diversidad) para poder afirmar el rendimiento de las redes neuronales, estas pueden lograr un rendimiento muy elevado (diferentes vistas, ángulos o contextos). Eso sugiere que cuando se esté ante una tarea de segmentación en situaciones complejas (ej. de monumentos o de parques altamente heterogéneos), sería interesante poder incluir más información de fondo o utilizar representaciones que sean más completas, tal como proponen Su et al. (2015) en su esquema de multivista.

En cambio, Chen et al. (2021) implementan Mask R-CNN para calcular áreas construidas a partir de imágenes aéreas por dron, obteniendo un IoU cercano al 68% y un F1-score que oscila entre el 80% y el 90%. Este tipo de enfoque de segmentación de instancias busca reconocer cada objeto (edificio) de manera individual, logrando así otorgar importancia a la segmentación a la hora de trabajar con imágenes aéreas, en las que hay una alta densidad de objetos y estos deben distinguirse del fondo. En comparación con los resultados de esta investigación en la escena del parque o el monumento, las redes neuronales han obtenido IoU’s más bajos en algunas de las clases (la clase "Zona artificial" o "Ambiente", por ejemplo), indicando la dificultad de separar este tipo de áreas en fotos que tienen variaciones con respecto al color y a las texturas.

En contraste, la escena del banano muestra un escenario más controlado (fondo homogéneo y objeto principal relativamente bien definido), lo cual explica que aquí las métricas de segmentación (IoU > 0.95 para el fondo, por ejemplo) se superen con relativa facilidad. Sin embargo, cabe destacar que, incluso en los casos en los que el IoU es más bajo, el empleo de arquitecturas avanzadas como

Mask R-CNN puede mejorar la acción de detección y de segmentación de clases más complejas (por ejemplo, con pocas muestras o con una alta variabilidad visual).

Es decir, los altos valores de IoU en los resultados de segmentación del banano contrastan con los resultados dispersos en parques y monumentos, donde el entorno es más heterogéneo. Este aspecto remarca que la complejidad de la escena afecta de manera directa la métrica de segmentación. La literatura especializada da cuenta de la gran efectividad de las redes neuronales profundas siempre que se disponga de un conjunto de datos representativo y de las arquitecturas y parámetros adecuados. Nuestros resultados, en particular en la segmentación de bananos, monumentos y parques, son acordes con la idea de que a mayor complejidad escénica y menor representatividad de la clase, más se hace evidente la necesidad de contar con métodos robustos y con un entrenamiento exhaustivo.

Conclusiones

Los métodos de segmentación no supervisada evaluados muestran desempeños contrastantes según la escena y las clases analizadas. Las redes neuronales destacan como el enfoque más robusto, especialmente en la escena del banano, con una precisión general de píxeles entre 95.4% y 98.0%, y valores excepcionales para la clase "Fondo" (IoU: 0.95-0.97; precisión: 99.8%-99.9%). Aunque la clase "Banano" presenta menor rendimiento (IoU: 0.76-0.87; precisión: 75.9%-87.8%), supera ampliamente a otros métodos. En la escena del monumento, las redes neuronales logran una precisión general moderada (23.6%-46.4%), con mejor desempeño en "Monumento" (precisión: 44.43%-64.31%), pero limitaciones en clases como "Ambiente" (precisión ≤37.14%). En la escena del parque, su precisión varía drásticamente (40.7%-88.4%), destacando en "Vegetación baja" (IoU: 0.87; precisión: 94.21% en DJI_0161), pero con debilidades en "Zona artificial" (IoU ≤0.27 en cuatro imágenes).

Respecto al clustering, presenta resultados mixtos. En la escena del banano, la precisión general (52.5%-70.8%) y la clase "Banano" (IoU: 0.08-0.20; precisión: 22.69%-51.53%) revelan limitaciones significativas. En la escena del monumento, aunque la precisión alcanza 53.9% en la imagen 12, clases como "Ambiente" (precisión ≤41.58%) y "Vegetación" (IoU: 0.14 en imagen 44) evidencian problemas. En el parque, destaca en "Vegetación baja" (IoU: 0.80; precisión: 84.15% en DJI_0161), pero falla en "Zona artificial" (IoU: 0.07 en DJI_0161).

En cuanto al crecimiento de regiones, muestra el rendimiento más inconsistente. En la escena del banano, aunque la clase "Banano" alcanza 90.13% de precisión en DSC_0437, la precisión general es crítica (19.0%-67.0%), con un IoU bajo para "Fondo" (0.10-0.64). En la escena del monumento, la clase "Vegetación" tiene precisiones cercanas a 0% en múltiples imágenes, y "Ambiente" registra IoU ≤0.42. En el parque, mientras "Vegetación baja" logra IoU de 0.88 (DJI_0161), "Zona artificial" y "Vegetación alta" presentan precisiones ≤9.46% y ≤58.83%, respectivamente.

Referencias

Adams, R., & Bischof, L. (1994). Seeded region growing. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(6), 641-647. doi:10.1109/34.295913

Arayici, Y. (2007). An approach for real world data modelling with the 3D terrestrial laser scanner for built environment. Automation in Construction, 16(6), 816–829. https://doi.org/10.1016/j.autcon.2007.02.008

Bala, Anju. (2020). A Review of Challenges in Clustering Techniques for Image Segmentation. International Journal for Research in Applied Science and Engineering Technology. 8. 496-502. 10.22214/ijraset.2020.32534.

Bengio, Y., Goodfellow, I., & Courville, A. (2018). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. Genetic Programming and Evolvable Machines, 19(1–2), 305–307. https://doi.org/10.1007/s10710-017-9314-z

Binbin Xiang, Jian Yao, Xiaohu Lu, Li Li, Renping Xie, and Jie Li (2017). "Segmentation-Based Classification for 3D Point Clouds in the Road Environment.". https://cvrs.whu.edu.cn/media/PCC/papers/3DPointCloudClassification.pdf

Borrmann, Dorit, Elseberg Jan, Lingemann Kai, Nüchter Bullet. (2011). "The 3D Hough Transform for plane detection in point clouds: A review and a new accumulator design." 3D Research. 0202. 10.1007/3DRes.02(2011)3.

Carbonneau, P. E., & Dietrich, J. T. (2017). Cost-effective non-metric photogrammetry from consumer-grade sUAS: implications for direct georeferencing of structure from motion photogrammetry. Earth Surface Processes and Landforms, 42(3), 473–486. https://doi.org/10.1002/esp.4012

Eltner, A., Kaiser, A., Castillo, C., Rock, G., Neugirg, F., & Abellán, A. (2016). Image-based surface reconstruction in geomorphometry-merits, limits and developments. Earth Surface Dynamics, 4(2), 359–389. https://doi.org/10.5194/esurf-4-359-2016

Erlag Berlin Heidelberg GmbH Physics, S.-V. (2001). Springer Series in Information Sciences 30 Editor: Teuvo Kohonen. http://www.springer.de/phys/

Francisco Caicedo Bravo, E., & Alfonso López Sotelo, J. (2017). UNA APROXIMACIÓN PRÁCTICA A LAS REDES NEURONALES ARTIFICIALES.

Gauss, C. F., & Mean, H. (2020). "Investigating the Role of Curvature in Point Cloud Edge Detection." Geometry and Imaging, 45(1), 33-51. doi:10.1016/gi.2020.3301.

Gonzalez, R. C., & Woods, R. E. (2002). Digital Image Processing (2nd ed.). Prentice Hall.

Gonzalez, R. C., & Woods, R. E. (2008). Digital Image Processing. Prentice Hall.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning.

Haralick, R. M., & Shapiro, L. G. (1985). Image segmentation techniques. Computer Vision, Graphics, and Image Processing, 29(1), 100-132. doi:10.1016/S0734-189X(85)90153-7

Hilera González, R., José Martínez Hernando, V., Hilera, J. R., & Martinez Hernando, V. (1995). Redes neuronales artificiales : fundamentos, modelos y aplicaciones / José. https://www.researchgate.net/publication/44343683

Jung, J., Hong, S., Jeong, S., Kim, S., Cho, H., Hong, S., & Heo, J. (2014). Productive modeling for development of as-built BIM of existing indoor structures. Automation in Construction, 42, 68–77. https://doi.org/10.1016/j.autcon.2014.02.021

Kohonen, T. (1982). Self-Organized Formation of Topologically Correct Feature Maps. In *Biol. Cybern* (Vol. 43).

Kohonen, T. (1988). An Introduction to Neural Computing. In Neural Networks (Vol. 1).

Lindeberg, T. (1998). Edge detection and ridge detection with automatic scale selection. International Journal of Computer Vision, 30(2), 117-154. IEEE Xplore

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier Nonlinearities Improve Neural Network Acoustic Models.

Martin H, Jose Antonio & Montero, Javier & Gomez, Daniel. (2010). A Divisive Hierarchical k-means based Algorihtm for Image Segmentation. 300-304. 10.1109/ISKE.2010.5680865.

Martin H, Jose Antonio & Montero, Javier & Gomez, Daniel. (2010). A Divisive Hierarchical k-means based Algorihtm for Image Segmentation. 300-304. 10.1109/ISKE.2010.5680865.

Mirzaei, K., Arashpour, M., Asadi, E., Masoumi, H., Bai, Y., & Behnood, A. (2022). 3D point cloud data processing with machine learning for construction and infrastructure applications: A comprehensive review. In Advanced Engineering Informatics (Vol. 51). Elsevier Ltd. https://doi.org/10.1016/j.aei.2021.101501

Mitra, N. J., & Nguyen, A. (2003). "Estimating Surface Normals in Noisy Point Cloud Data." Proceedings of the Nineteenth Annual Symposium on Computational Geometry, 322-328.

Pətrəucean, V., Armeni, I., Nahangi, M., Yeung, J., Brilakis, I., & Haas, C. (2015). State of research in automatic as-built modelling. Advanced Engineering Informatics, 29(2), 162–171. https://doi.org/10.1016/j.aei.2015.01.001

Pham, D. L., Xu, C., & Prince, J. L. (2000). "Current methods in medical image segmentation." Annual Review of Biomedical Engineering, 2, 315-337.

Poullis, C. (2013). A framework for automatic modeling from point cloud data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(11), 2563–2575. https://doi.org/10.1109/TPAMI.2013.64

R. Pushpavalli & G. Sivarajde, "An intelligent post processing technique for image enhancement," 2013 International Conference on Emerging Trends in VLSI, Embedded System, Nano

Electronics and Telecommunication System (ICEVENT), Tiruvannamalai, India, 2013, pp. 1-5, doi: 10.1109/ICEVENT.2013.6496555.

R.Hecht-Nielsen. (1988). Neurocomputing_picking_the_human_brain.

Rao, Q., Yu, B., He, K., & Feng, B. (2019). Regularization and Iterative Initialization of Softmax for Fast Training of Convolutional Neural Networks. http://www.ieee.org/publications

Remondino, F., Spera, M. G., Nocerino, E., Menna, F., & Nex, F. (2014). State of the art in high density image matching. Photogrammetric Record, 29(146), 144–166. https://doi.org/10.1111/phor.12063

Roberts, L. G. (1965). Machine perception of three-dimensional solids. Optical and Electro-Optical Information Processing, 159-197. IEEE Xplore

Rusu, R. B., & Cousins, S. (2011). "3D is here: Point Cloud Library (PCL)." 2011 IEEE International Conference on Robotics and Automation, 1-4.

Sander, J., Ester, M., Kriegel, H., & Xu, X. (1998). "Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications." Data Mining and Knowledge Discovery, 2, 169-194

Schuler, C. J., Hirsch, M., Harmeling, S., & Scholkopf, B. (2016). Learning to Deblur. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(7), 1439–1451. https://doi.org/10.1109/TPAMI.2015.2481418

Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. (2015). Multi-view Convolutional Neural Networks for 3D Shape Recognition. 2015 IEEE International Conference on Computer Vision (ICCV), 945–953. https://doi.org/10.1109/ICCV.2015.114

Torresan, C., Berton, A., Carotenuto, F., Di Gennaro, S. F., Gioli, B., Matese, A., Miglietta, F., Vagnoli, C., Zaldei, A., & Wallace, L. (2017). Forestry applications of UAVs in Europe: a review. International Journal of Remote Sensing, 38(8–10), 2427–2447. https://doi.org/10.1080/01431161.2016.1252477

Xue, J., Hou, X., & Zeng, Y. (2021). Review of image-based 3d reconstruction of building for automated construction progress monitoring. Applied Sciences (Switzerland), 11(17). https://doi.org/10.3390/app11177840

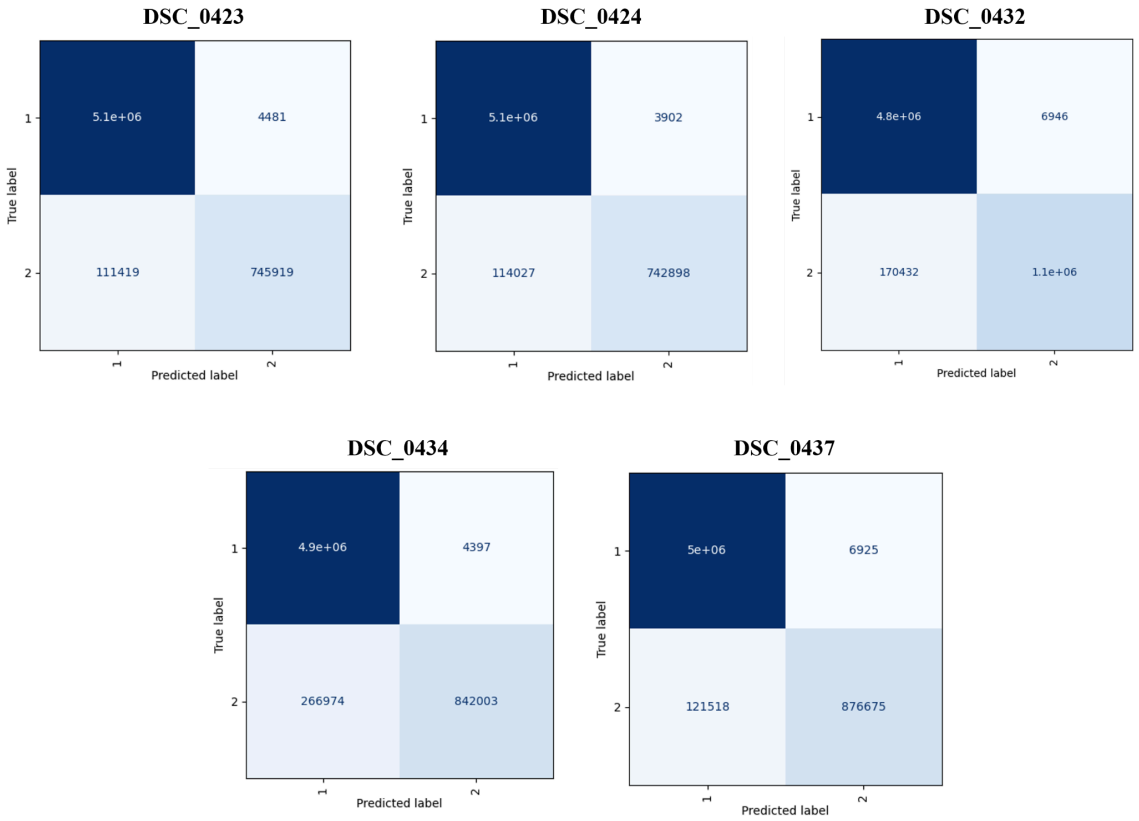
Zhang, Y., Brady, M., & Smith, S. (2016). Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. IEEE Transactions on Medical Imaging, 20(1), 45-57. https://doi.org/10.1109/42.906424

Zhang, Yaqian, Mańdziuk, Jacek, Quek, Chai, & Goh, Boon. (2017). "Curvature-based method for determining the number of clusters." Information Sciences. 415-416. 10.1016/j.ins.2017.05.024.

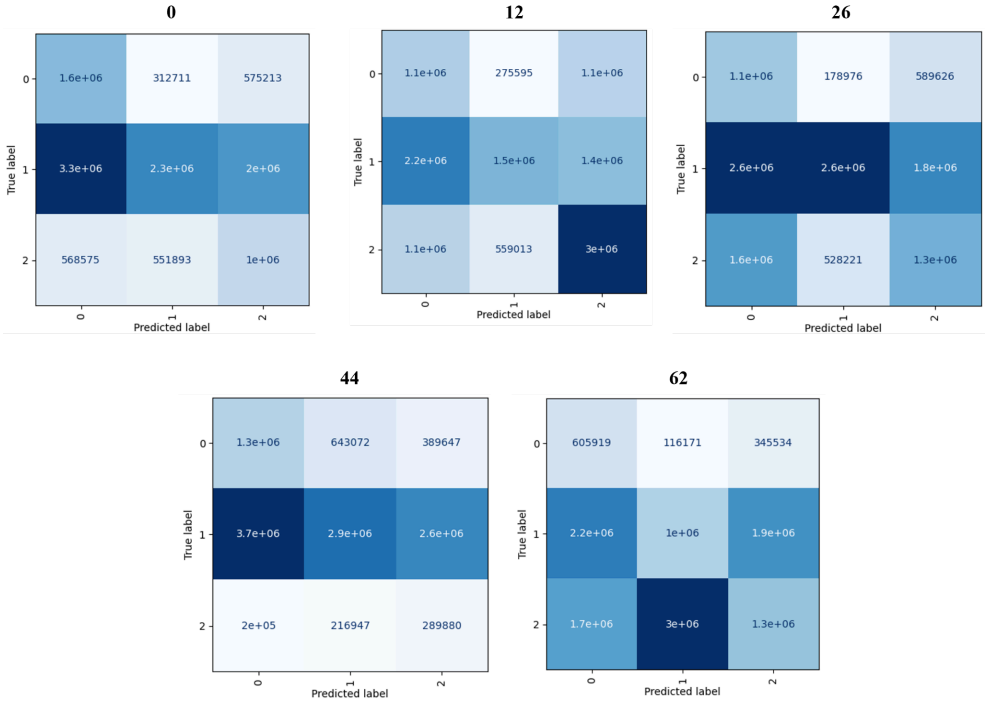
Zucker, S. W. (1976). Region growing: Childhood and adolescence. Computer Graphics and Image Processing, 5(3), 382–399. doi:10.1016/s0146-664x(76)80014-7

Anexos

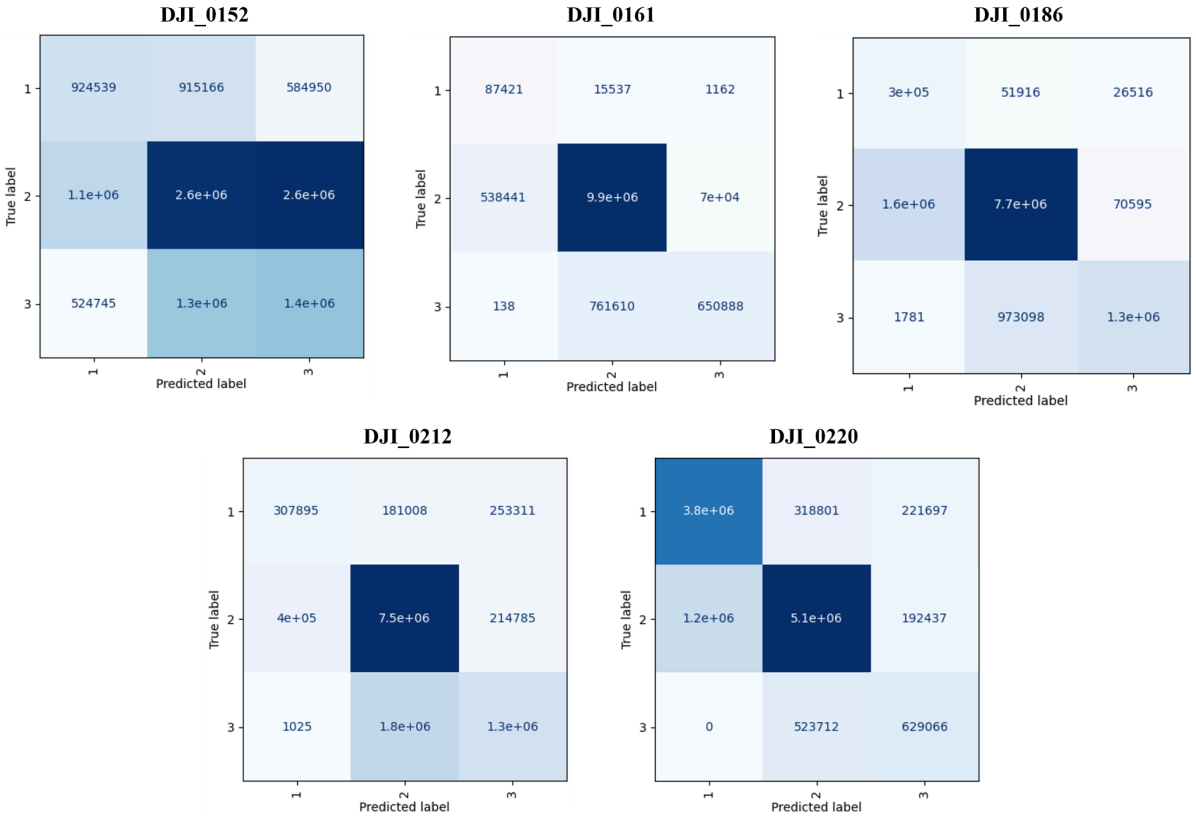
Anexo 1: Matrices de confusión para las imágenes de prueba en la escena del banano, usando redes neuronales.



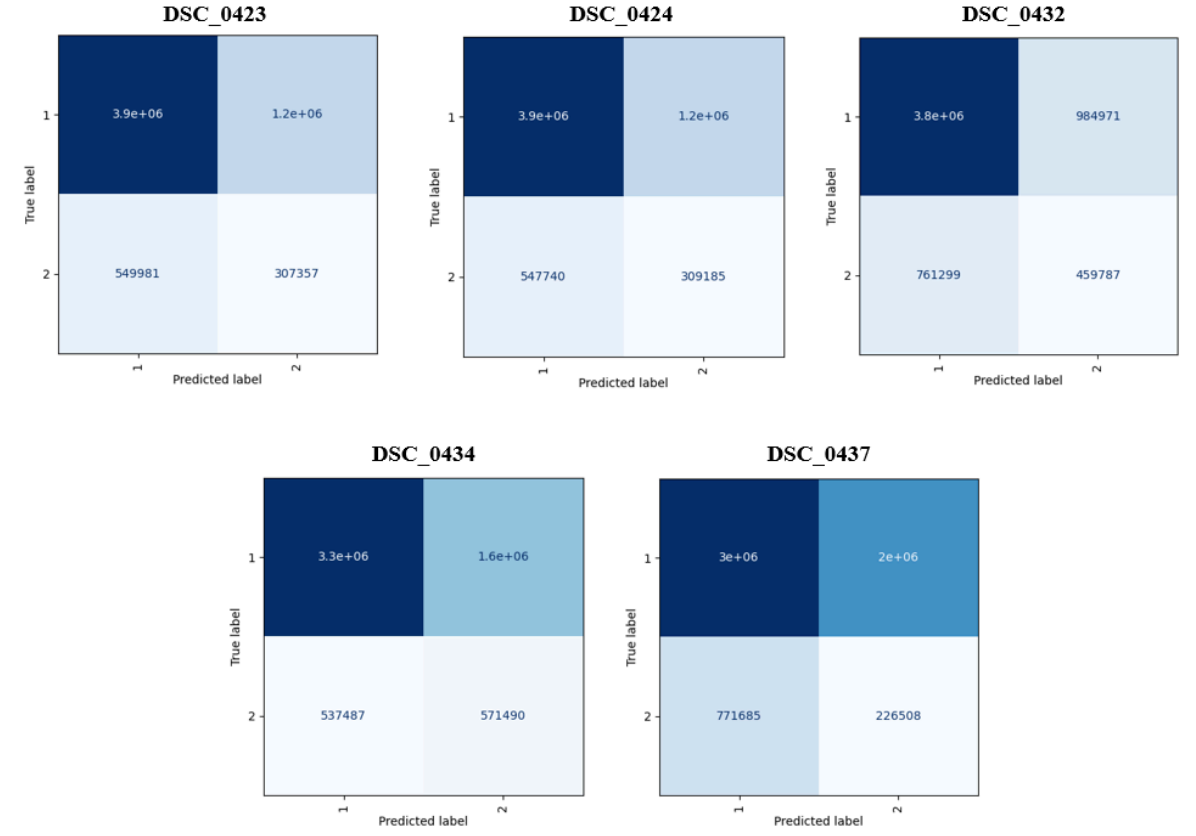
Anexo 2: Matrices de confusión para las imágenes de prueba en la escena del monumento, usando redes neuronales.



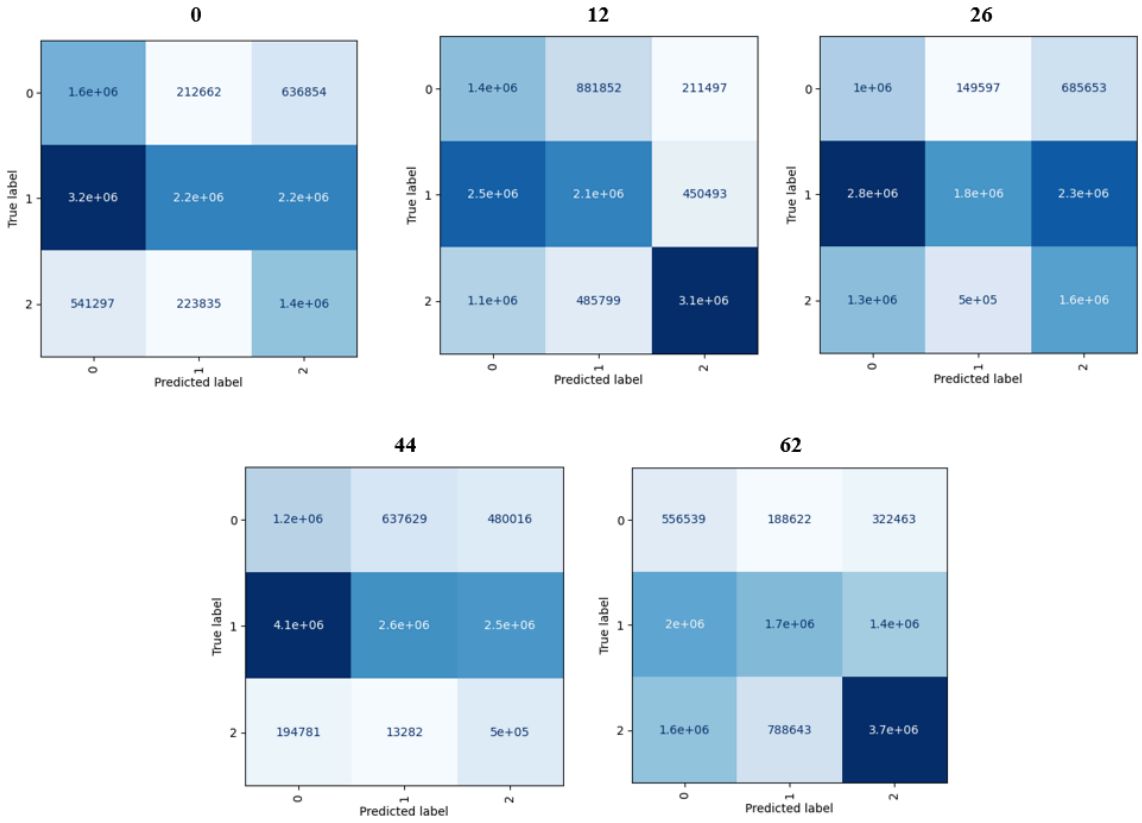
Anexo 3: Matrices de confusión para las imágenes de prueba en la escena del parque, usando redes neuronales.



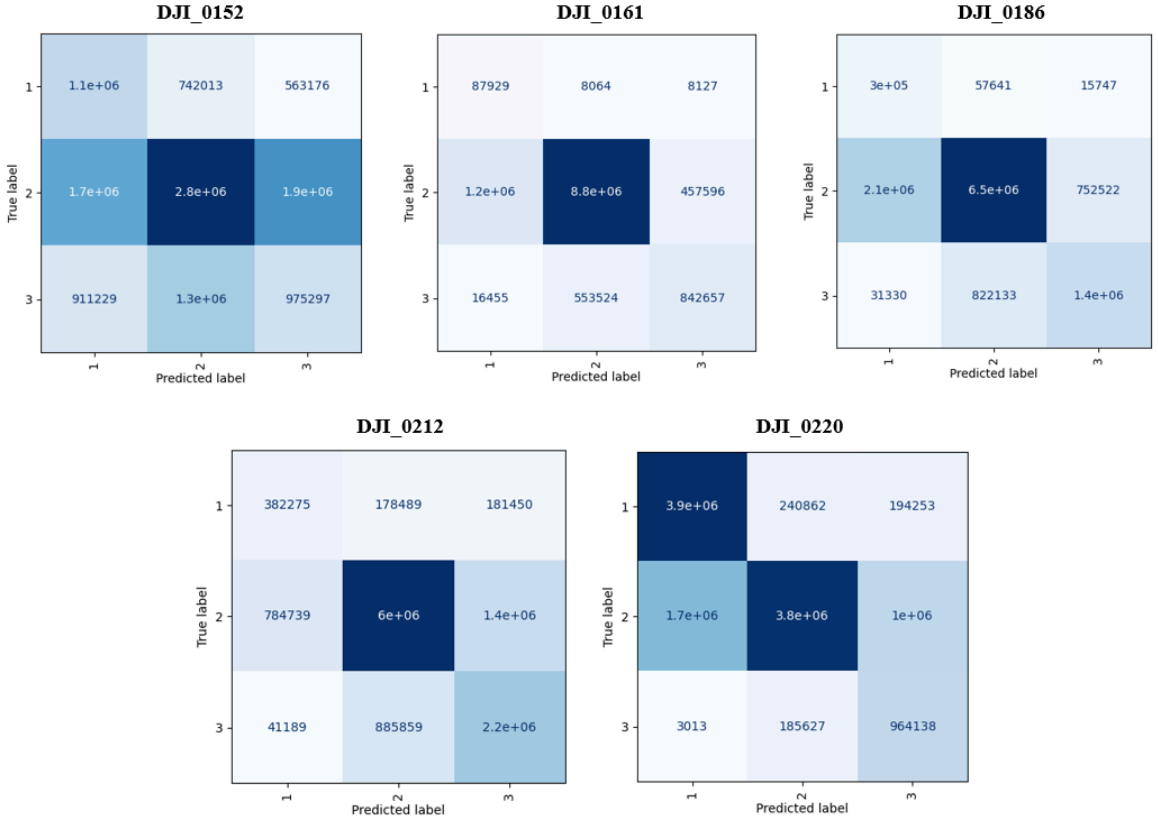
Anexo 4: Matrices de confusión para las imágenes de prueba en la escena del banano, usando clustering.



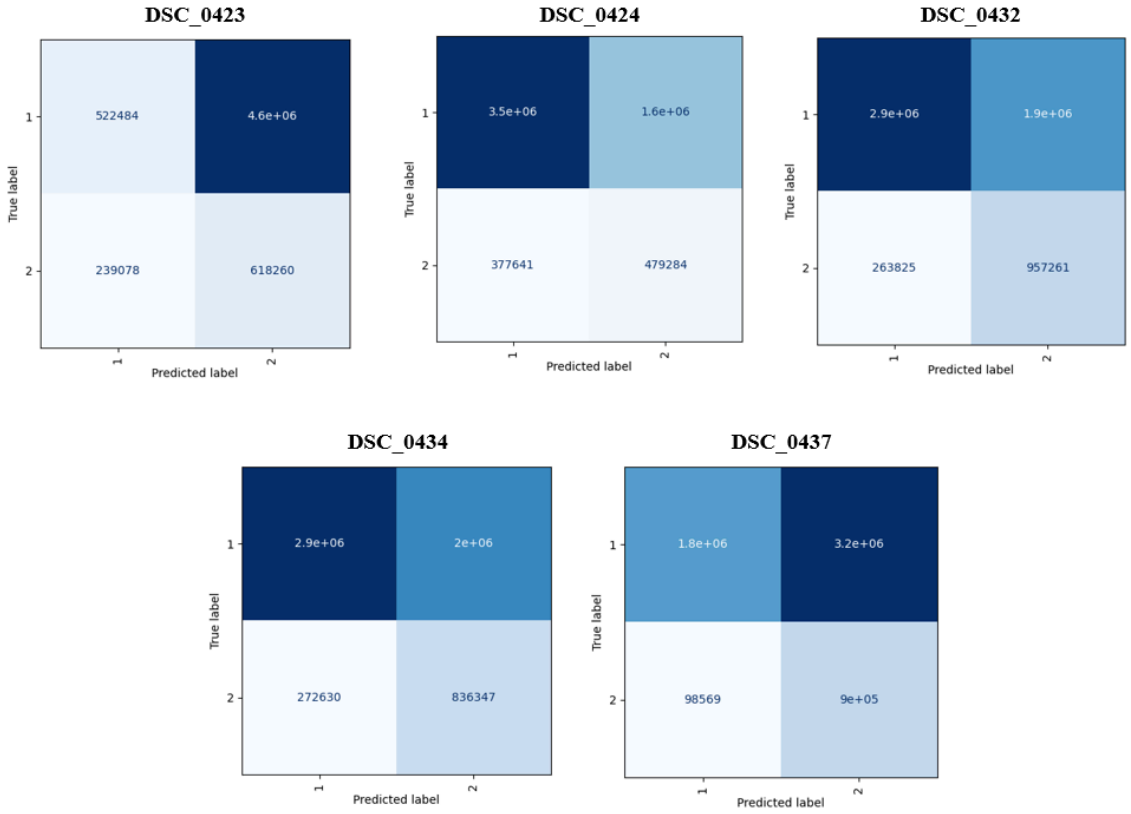
Anexo 5: Matrices de confusión para las imágenes de prueba en la escena del monumento, usando clustering.



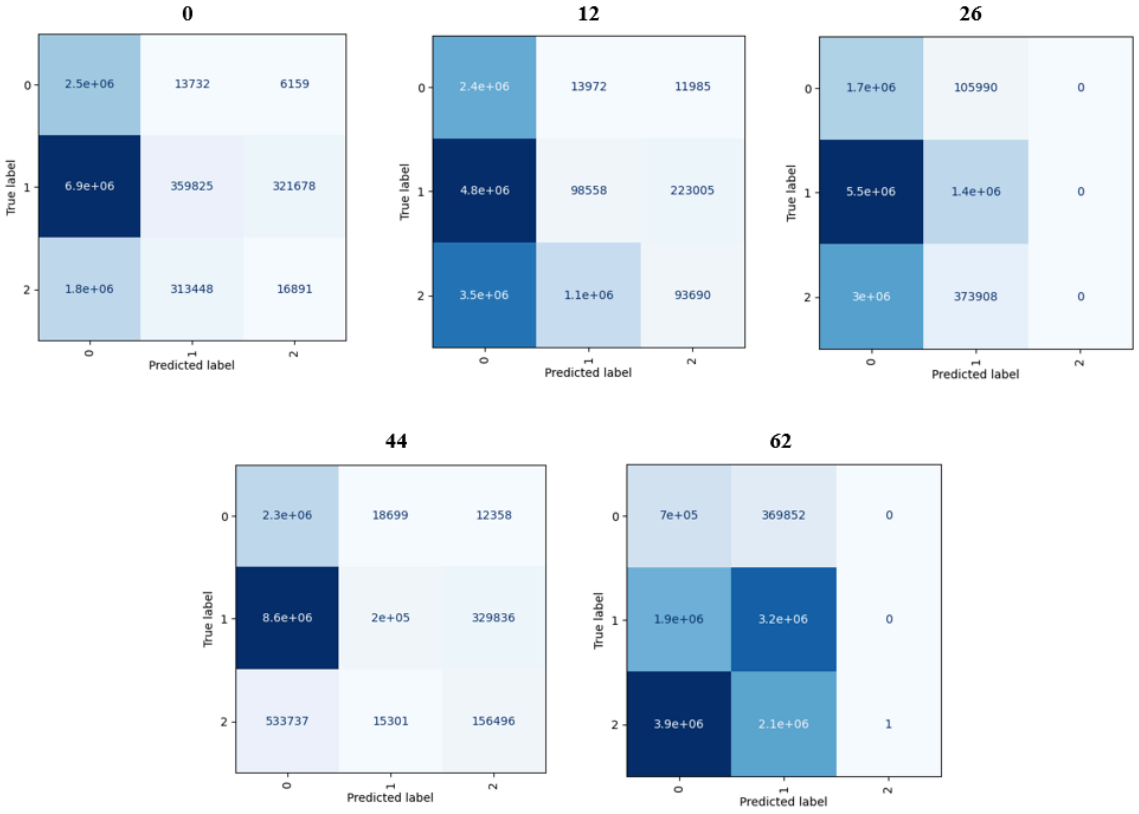
Anexo 6: Matrices de confusión para las imágenes de prueba en la escena del parque, usando clustering.



Anexo 7: Matrices de confusión para las imágenes de prueba en la escena del banano, usando crecimiento de regiones.



Anexo 8: Matrices de confusión para las imágenes de prueba en la escena del monumento, usando crecimiento de regiones.



Anexo 9: Matrices de confusión para las imágenes de prueba en la escena del parque, usando crecimiento de regiones.


DJI_0152
231337
1.9e+06
314456
9e+05
4.5e+06
987583
390715
2.4e+06
4e+05
True label
Predicted label
DJI_0161
8887
95233
0
11204
1e+07
80581
32744
1.2e+06
213855
DJI_0186
590
373739
1291
35157
9e+06
307423
238398
1.5e+06
567038
DJI_0212
48138
353257
340819
124896
7.5e+06
456716
15968
3.1e+06
17996
DJI_0220
13426
4.3e+06
40707
41283
6.2e+06
322915
36581
437997
678200