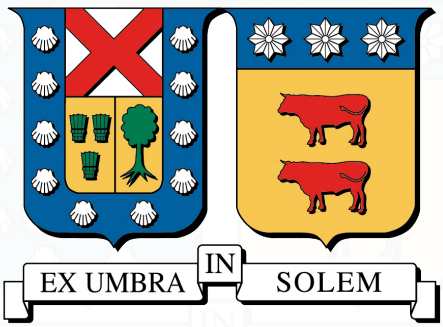
UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INDUSTRIAS
SANTIAGO - CHILE



DESARROLLO DE UN MODELO DE PROCESAMIENTO DE IMÁGENES CAPTURADAS POR DRONES PARA AUTOMATIZAR LA INTERPRETACIÓN DEL DISEÑO GEOMÉTRICO DE CAMINOS MINEROS.

SEBASTIÁN NICOLÁS PADILLA LOBOS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INDUSTRIAL

PROFESOR GUÍA	:	SR. DAVID GODOY R.
PROFESOR CORREFERENTE	:	SR. HANS MILLÁN R.

OCTUBRE 2023

MIS AGRADECIMIENTOS

A cada una de las personas que me ayudaron a llegar hasta acá, las llevo en mi corazón mas que en el papel.
Arigato gozaimas!!!

ABSTRACT

- **Contexto:** En la minería, los costos de transporte de material, los cuales se pueden listar en: la inversión en activos tales como camiones y/o caminos mineros; la mantención y el consumo de insumos para el uso de estos mismos, representan en promedio entre el 40 % y 50 % del costo total en minas profundas de rajo abierto. Por la significancia de estos costos, las empresas tienen principal preocupación en el estudio de la optimización de la productividad y disminución de los costos asociados al transporte. Dentro de los costos listados, la condición de un camino minero puede afectar significativamente el desempeño de la operación de una mina a rajo abierto. Es por lo anterior, que es sumamente importante contar con la claridad suficiente para mantenerlos en su mejor estado posible. Sin embargo, determinar la frecuencia de mantención de un camino minero y/o cuándo sacar uno de estos de servicio para reconstruirlos, puede ser difícil. Esto, principalmente por la presencia de diversos factores que afectan su desempeño, como por ejemplo, la resistencia a la rodadura, variable que representa una resistencia al movimiento de los camiones respecto al piso, y que al aumentar, lo que esta asociado a un deterioro del estado del camino, provoca una disminución de la velocidad máxima que pueden alcanzar los camiones y aumenta considerablemente el consumo de combustible. Lamentablemente, hasta hace poco, las formas de controlar y cuantificar esta variable son escasas, costosas y difíciles de realizar, además de que pueden ser completamente arbitrarias.
- **Necesidades:** Con la finalidad de contar con una alternativa innovadora, en apoyo a la definición de planes óptimos de mantención, actualmente se trabaja en el **Protocolo para obtener diseño geométrico y funcional de una carpeta de rodado** para identificar el diseño geométrico y funcional de caminos mineros mediante el análisis de fotografías captadas por drones con cámaras de alta resolución. Dentro del desarrollo de este protocolo, se cuenta con que el proceso de identificación del diseño geométrico se efectúa de forma manual mediante el uso de la herramienta de recorte de fotografías en los programas Qgis y/o Argis.
- **Tareas:** Por lo cual se busca generar una herramienta automatizada y optimizada en la interpretación geométrica de los caminos que permita un futuro análisis eficaz del diseño funcional de estos.
- **Objetivos:** El objetivo principal, es diseñar un modelo de programación que permita el análisis de fotografías tomadas por drones de la vista superior de una mina a rajo abierto para identificar de forma inmediata qué porción de cada fotografía corresponde a un camino minero, con el fin de optimizar el proceso para obtener el diseño geométrico y funcional de estos, apoyando el desarrollo de una nueva alternativa para definir la resistencia a la rodadura.
- **Metodología:** El plan de trabajo se resumen en: definir el tipo de herramientas y metodologías de procesamiento de imágenes e identificar cuáles de las características definidas anteriormente son posibles de obtener mediante estas; Recopilar, clasificar y limpiar los datos (imágenes) para trabajar con el modelo de programación; Crear un modelo que permita obtener los resultados esperados y por último la Validación del modelo y sus resultados.
- **Resultados Esperados:** Se obtiene un modelo capaz de procesar imágenes e identificar en primera instancia si corresponde a un camino minero o no.

- **Conclusiones:** Modelo el cual permite pasar de un tiempo de recorte de una fotografía de entre 5 y 10 minutos a un tiempo de 2,5 segundos en promedio. Destacando de igual manera que no es totalmente limpio en el proceso de recorte de cada fotografía.

- **Trabajos Futuros:** Por lo mismo es un modelo que requiere de una mayor optimización en cada parte de este, pero que a su vez abre camino al uso de otras herramientas y/o tecnologías como el uso de redes neuronales.

Índice de Contenidos

1. Contexto **1**

2. Problema de Investigación **3**

3. Objetivos **4**
3.1. General . . . 4
3.2. Específicos . . . 4

4. Marco Teórico **5**
4.1. Diseño de Caminos mineros . . . 5
4.1.1. Diseño Geométrico . . . 5
4.1.2. Diseño Funcional . . . 6
4.2. Visión Artificial . . . 7
4.3. Etapas de un sistema de visión artificial . . . 7
4.4. Imágenes . . . 8
4.5. Píxeles . . . 9
4.6. Kernel . . . 10
4.7. OpenCV . . . 11
4.8. Suavizamiento . . . 12
4.8.1. Desenfoque Promedio (Average Blur): . . . 12
4.8.2. Desenfoque Medio (Median Blur): . . . 12
4.8.3. Desenfoque Gaussiano (Gaussian Blur): . . . 13
4.9. Binarización . . . 14
4.9.1. Binarización binaria . . . 14
4.9.2. Binarización truncada . . . 15
4.9.3. Binarización a cero . . . 15
4.10. Binarización adaptativa . . . 16
4.10.1. Umbral adaptativo promedio . . . 16
4.10.2. Umbral adaptativo gaussiano . . . 17
4.10.3. Binarización de Otsu . . . 18
4.11. Operaciones morfológicas: . . . 19
4.11.1. Erosión: . . . 19
4.11.2. Dilatación: . . . 20
4.11.3. Apertura: . . . 20
4.11.4. Clausura: . . . 21
4.11.5. Gradiente morfológico: . . . 21
4.11.6. Top Hat: . . . 22
4.11.7. Black Hat: . . . 22
4.12. Contornos . . . 23
4.13. Intersecciones Bit a Bit . . . 23

5. Metodología **24**
5.1. Procesamiento de imágenes del vuelo de drones sobre minas de rajo abierto 24
5.2. Flujo Propuesto 25

6. Desarrollo **26**
6.1. Conversión a escala de grises 26
6.1.1. Muestras en escalada de grises 26
6.2. Suavizamiento 27
6.3. Binarización 28
6.4. Operaciones Morfológicas 30
6.5. Segmentación 31
6.6. Clasificación y filtrado de los objetos 32

7. Conclusiones **36**
7.1. Identificar una metodología de procesamiento de imágenes que permita identificar caminos mineros. 36
7.2. Desarrollar un modelo de programación en Python que permita aplicar la metodología definida procesando imágenes de minas de rajo abierto. 37
7.3. Evaluar el algoritmo mas apto para cada parte del proceso según el estado del arte. 38
7.4. Obtener un dataset con cada camino minero mediante la aplicación del modelo desarrollado. 39

8. Trabajos Futuros **40**
8.1. Refinar parámetros algoritmo 40
8.2. Análisis de datos obtenidos 40
8.3. Uso del dataset obtenido en un algoritmo de Redes Neuronales 40

9. Referencias **42**

A. ANEXOS **44**

1 | Contexto

En Chile, en la actualidad y en gran parte de su historia, la minería se ha presentado como un actor principal en el desarrollo económico del país. De hecho, en el año 2020, la porción del PIB representada exclusivamente por esta actividad rondeaba el 11 % (Consejo Minero,2020), y un 13 % en el año 2022. Vonsiderando toda la actividad económica relacionada a la minería, se puede llegar a considerar que representó un 25 % del PIB el año recién pasado (Consejo Minero, 2023).

Es preciso identificar que una gran porción de este aporte se define por la rentabilidad operacional de las empresas del rubro. La palanca más considerable en este sentido es la diferencia entre los costos de producción y el precio del cobre.

Dentro de los factores que afectan la productividad, cabe destacar la disminución en las leyes de mineral. Lo que para el año 2018, provocó una caída de 289 mil toneladas de menor producción de cobre en comparación al año anterior en la industria minera Chilena (Reporte Minero, 2019). Se identifican ademas, factores relacionados a la velocidad, la cantidad de material trasladado, los tiempos de operación y logísticos. Estos factores tienen una mayor incidencia en los costos operacionales, especialmente en el caso de las minas a rajo abierto. Esto se debe a que los rajos se hacen más largos y profundos, lo que tiene un impacto directo en el consumo de insumos como energía, agua, combustible, entre otros.

Este hecho ha resultado en un descenso promedio anual de la productividad del 4,7 % entre los años 1993 y 2015, según datos de la OCDE. Esto se traduce en que Chile necesita más del doble del tiempo que los países desarrollados para realizar la misma labor minera. Un ejemplo concreto de esto es que para transportar mil toneladas de material, países como Canadá, Suecia, Australia o Estados Unidos necesitan aproximadamente 30 horas en promedio, mientras que en Chile se requieren 67 horas.

En el mismo informe, la OCDE indica que las principales claves para mejorar la productividad son la innovación y la investigación y desarrollo (I+D), áreas en las cuales Chile también está rezagado. El país cuenta con un valor agregado respecto a la minería del 0,15 %, mientras que Australia, Estados Unidos y Suecia cuentan con un 2 %, 1,5 % y 1 % respectivamente.(Reporte Minero,2018).

Dentro de los principales costos de esta se industria se encuentra el costo de transporte de material, el cual se puede representar mayoritariamente en:

1. Inversión en activos tales como camiones y caminos mineros
2. Consumo de insumos para los activo
3. La mantención de los activos

Este costo oscila entre el 40 % y 50 % del costo total para minas profundas a rajo abierto en promedio(Equipo Minero, 2009).

Al cuantificar la magnitud del impacto de estos costos, es que las empresas mineras han comenzado a buscar enfoques de estudio relacionados a la optimización de la productividad y la minimización de los costos, en donde la capacidad de movilizar grandes volúmenes de material de forma eficaz pasa a ser esencial.

En este contexto, es primordial considerar a los caminos mineros como un activo medular en la operación de extracción del mineral al mismo nivel que camiones y palas, debido a que una buena construcción de estos y correctas prácticas de mantención, son claves para operar con una flota de camiones a un costo eficiente (Tannant & Regensburg,2001).

Las condiciones de un camino minero afectan el desempeño de una mina desde los costos asociados, pero también desde el punto de vista de la productividad. Esto se ve reflejado principalmente en la forma en que las condiciones de éstos afectan a factores relevantes, tales como la velocidad de los camiones, el consumo de combustible o la generación de gases de invernadero (Soofastaei et al. 2015).

El principal parámetro que permite reflejar el impacto del estado de los caminos sobre la operación es la resistencia a la rodadura, la que corresponde a la resistencia que ejerce el camino sobre la rotación de las ruedas de los vehículos, el cual presenta un comportamiento inverso respecto a la velocidad máxima de los camiones y directo respecto al gasto de combustible, es decir, a medida que aumenta la resistencia a la rodadura los camiones son incapaces de alcanzar su velocidad máxima, generando una menor productividad. Así mismo, el gasto de combustible aumenta significativamente provocando mayores costos. En un caso de estudio se determina que un aumento del 2 % en la resistencia a la rodadura puede generar una disminución del 26 % de la velocidad máxima de un camión con una carga límite de 400 toneladas.(R. Thompson, 2019)

Para evitar el aumento en la resistencia a la rodadura se definen ciclos de mantención y/o cambios de estas carreteras. Estas son acciones preventivas, las cuales significan un costo que debe hacer frente a la reducción en la velocidad y el gasto en combustible mencionados anteriormente.

El problema que se presenta actualmente es la dificultad en la determinación del porcentaje óptimo real de la resistencia a la rodadura para aplicar un ciclo de mantención definido que minimice los costos de mantención y transporte de los camiones, debido a que este índice depende de diversas variables geométricas como elevación, ancho y largo del camino, como también de su estado funcional, es decir, si cuenta con: desprendimientos, hoyos, gravilla suelta en las vías, entre otros. Es por lo anterior, que es sumamente difícil de obtener y actualmente cuenta con pocas alternativas para su medición.

Hoy existen dos métodos para cuantificar este parámetro, los cuales se consideran altamente arbitrarios o poco contundentes en la literatura.

- El primer metodo consiste en un análisis y evaluación cualitativa de los defectos presentes en los caminos, midiendo aquellos que son más significativos y comparalandolos con una referencia teórica.
- El segundo es un testeo cuantitativo mediante la toma de una muestra de 20 kg de la carpeta del camino. La cual se envía a un laboratorio que provee reportes de su composición, los que cruzándolos con la velocidad y peso de los camiones permite modelar la resistencia a la rodadura en el tiempo.

El problema respecto a cada método, es que el primero es considerablemente arbitrario y el segundo corresponde a una muestra poco representativa de la totalidad del camino.

2 | Problema de Investigación

Para abordar esta problemática, en el año 2020 en Chile se comenzaron a desarrollar investigaciones que incluyen el uso de drones para fotografiar las faenas mineras. Estas fotografías, al trabajadas a través de herramientas de análisis geográfico, permiten identificar cuáles de ellas contienen caminos mineros y, de esta manera, capturar características específicas de éstos, como su geometría y funcionalidad. Todo lo anterior tiene como objetivo proporcionar una manera mas efectiva de medir el índice de resistencia a la rodadura, con la finalidad de mantener los caminos de mejor forma y hacer el transporte más eficiente.

En la actualidad existe el **Protocolo para obtener diseño geométrico y funcional de una carpeta de rodado**. Este protocolo, mediante el uso de nubes de puntos, softwares para la lectura y edición de estas (ej: Cloudcompare) y el software de manejo de información geográfica Qgis, permite obtener el diseño geométrico y funcional de un camino minero. Cabe destacar que, dentro de este protocolo, la obtención del diseño geométrico se realiza de forma manual utilizando herramientas de recorte. Esto significa que, para cada camino que se quiera analizar en la actualidad, es necesario recurrir a esta herramienta y recortar de forma meticulosa los segmentos de la nube de puntos que correspondan a un camino. Este proceso es lento e ineficiente. Por ejemplo, considerando un universo de 1900 imágenes y un tiempo de recorte aproximado de 6 minutos cada una, significaría un tiempo aproximado de 8 días obtener el diseño geométrico de todos los caminos a estudiar.

Para abordar esta problemática y optimizar el proceso de recorte de estas fotografías, en la siguiente memoria se propone desarrollar un modelo de procesamiento de imágenes que permita identificar de forma automática la sección de cada fotografía que corresponda a un camino minero.

3 | Objetivos

3.1. General

Desarrollar un modelo de procesamiento de imágenes para automatizar la identificación de caminos mineros en fotografías captadas por drones, con el fin de optimizar la interpretación del diseño geométrico de estos.

3.2. Específicos

1. Estudiar una metodología de procesamiento de imágenes que permita identificar objetos dentro de de estas.
2. Desarrollar un modelo de programación en Python que permita aplicar la metodología definida, procesando imágenes de minas de rajo abierto.
3. Evaluar el algoritmo más apto para cada parte del proceso.
4. Obtener un dataset con cada camino minero mediante la aplicación del modelo desarrollado, con miras a un futuro análisis funcional de estos.

4 | Marco Teórico

4.1. Diseño de Caminos mineros

El rendimiento de los caminos mineros es principalmente definido en su etapa de diseño, el cual comprende un enfoque geométrico y funcional (Thompson et al. 2019).

4.1.1. Diseño Geométrico

El diseño geométrico es la base para el desarrollo de un camino minero; este representa principalmente el dibujo y la alineación del camino, tanto en un plano vertical como horizontal. La principal finalidad del diseño geométrico es que sea óptimamente eficiente y confiable geométricamente. En lo que respecta al estado del arte, actualmente se cuenta con un vasto desarrollo de buenas prácticas y recopilación de datos en cuanto a la realización de un diseño geométrico que asegure un buen funcionamiento. Considerando esto, para un buen diseño geométrico es importante aplicar, en conjunto al análisis geométrico, principios de diseño de mantención funcional y estructural.

Específicamente, el diseño geométrico esta orientado a determinar las características de un camino minero, teniendo en cuenta la topografía dónde se emplazaría este, así como el nivel de tráfico y velocidades planificadas sobre el mismo: La finalidad es que se pueda transitar sobre este de forma segura y óptima. Desde el punto de vista técnico, el diseño geométrico se configura mediante tres figuras bidimensionales que se forman individualmente, pero que son interdependientes entre si. De tal forma que, en su unión, generan una figura tridimensional que representa el camino propiamente dicho. Las figuras son:

- Alineación vertical: Plano vertical de abscisas y cotas, generado por distancias horizontales del camino y las pendientes sobre este.
- Alineación horizontal: Plano horizontal de coordenadas norte/este generado por ángulos y distancias.
- Diseño transversal: Planos transversales de distancias y cotas generados por distancias verticales y horizontales(J.Agudelo, 2002)

En resumen, el diseño geométrico es el que definirá cómo sera la forma que tome el camino, adaptándose al medio en que se encuentra y propiciando una circulación segura y eficiente.

4.1.2. Diseño Funcional

El diseño funcional apunta a la selección óptima de los materiales que componen la carpeta de rodado ubicada sobre la figura obtenida en el diseño geométrico. Esto se hace con el propósito de contar con una carpeta que proporcione el adecuado nivel de tracción para los vehículos que transiten por ella y, al mismo tiempo, evite el desarrollo de defectos a lo largo del tiempo sobre en sí misma.

Debido a esto, el diseño funcional comprende principalmente la elección del tipo de material con el objetivo de crear un entorno seguro, económico y respetuoso con el vehículo. El mantenimiento está estrechamente relacionado con el material y es un aspecto importante a considerar durante el diseño de una carretera. El diseño estructural y funcional de una carretera influye en la frecuencia y el tipo de mantenimiento requerido(D. Hugo, 2005)

Por ejemplo, considerando las condiciones climáticas de la ubicación de la mina y el tipo de vehículos que circulen sobre ella, puede ser necesario cubrir la carpeta con arenisca, gravilla u otro material que se adapte mejor a su entorno y permita planificar de manera estratégica las frecuencias de mantenimiento.

4.2. Visión Artificial

Entendiendo el contexto alrededor de los caminos mineros, es importante comprender como el procesamiento de imágenes puede ser útil para su análisis y, en este caso, para obtener de manera precisa y rápida la interpretación geométrica de estos.

La visión artificial forma parte de lo que se conoce como inteligencia artificial, que mediante la utilización de técnicas adecuadas, principalmente matemáticas, permite la obtención, procesamiento y análisis de todo tipo de información especial contenida en una imagen digital.

Esta disciplina comprende procesos que son capaces de otorgar la capacidad de "mirar u observar.[a] una máquina. Respecto a esto es posible afirmar que la visión artificial o comprensión de imágenes corresponde a la estimación automática de la estructura y propiedades de un lugar en tres dimensiones dinámico, a partir de imágenes obtenidas del mundo en dos dimensiones. Las estructuras y propiedades de ese lugar que se quieren deducir en visión artificial incluyen no solo sus propiedades geométricas, sino también sus propiedades materiales. Ejemplo de estas propiedades son su color, iluminación, textura y composición. Si el mundo se modifica en el proceso de formación de la imagen, se necesitará inferir también la naturaleza del cambio e incluso predecir el futuro(Ordieres-Meré et al., 2006).

4.3. Etapas de un sistema de visión artificial

Al trabajar en el análisis de imágenes, generalmente se divide el proceso en 4 etapas principales.

- **El proceso sensorial:**Esta etapa implica la captura o adquisición de la imagen del mundo real en formato digital a través de algún sistema de captura.
- **El pre-pocesamiento:** En esta fase se realiza un tratamiento digital de las imágenes obtenidas con el objetivo de simplificar las etapas posteriores. En esta etapa, se eliminan partes no deseadas de la imagen y se realzan las partes de interés mediante el uso de filtros y transformaciones geométricas, entre otros métodos.
- **Segmentación:**Esta etapa consiste en separar las zonas o elementos relevantes en una escena para poder comprenderla. En la literatura, existen diversos métodos de segmentación, como se muestra en la siguiente figura:

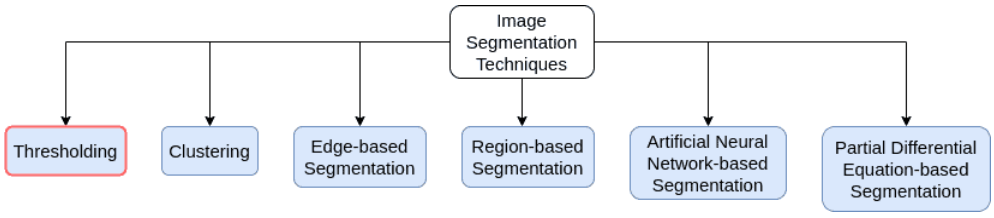


Figura 4.1: Métodos de segmentación tomado de Otsu's thresholding technique | LearnOpenCV (Murzova, 2021)

En este trabajo, se revisa el primer recuadro, que corresponde al método de binarización.

- **Reconocimiento o clasificación:** En esta etapa, se logra la distinción de los objetos segmentados en la escena. Esto se logra mediante el análisis de ciertas características que se establecen previamente para diferenciarlos (S. yoon, 2014).

En este caso, dado que ya disponemos de un conjunto de datos de imágenes digitales de una mina a rajo abierto, el proceso de análisis y desarrollo comienza en la etapa de preprocesamiento.

4.4. Imágenes

El proceso de obtención de una imagen digital comienza con sensores que registran la intensidad de la luz, utilizando diversos filtros de color. Luego, los dispositivos de memoria digital se encargan de almacenar la imagen en formato RGB o como datos crudos para su posterior análisis, como se ilustra en la siguiente imagen.

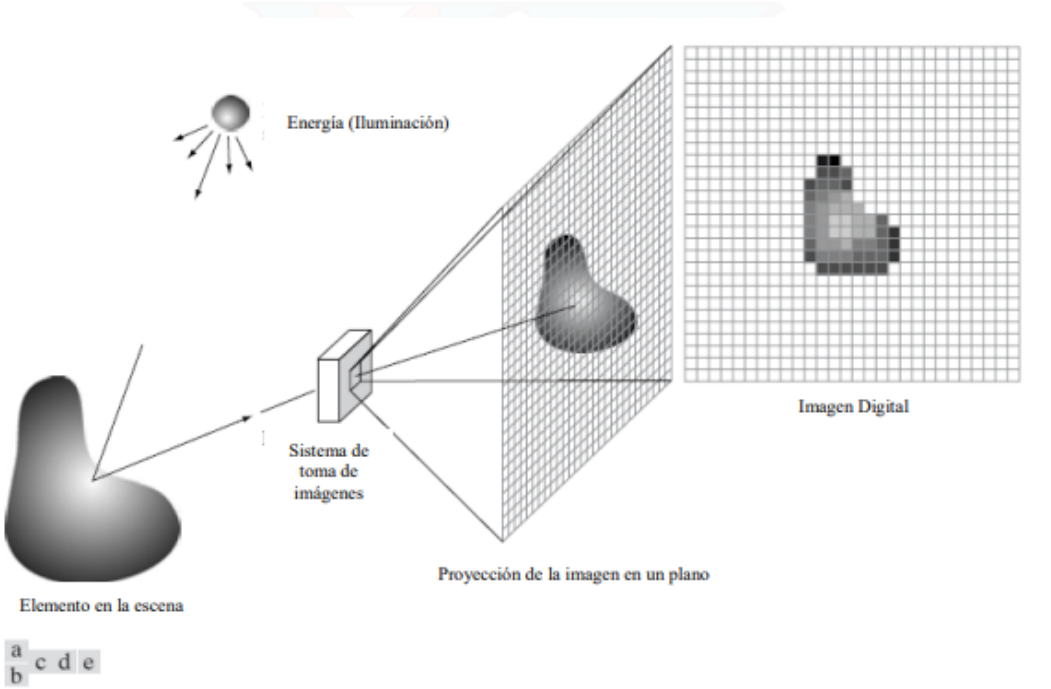


Figura 4.2: Digitalización de una imagen real tomado de Digital Image Processing Second Edition.Gonzalez (Rafael & Faisal, Zahraa. 2019).

4.5. Píxeles

La representación digital de la información en una imagen se basa en el píxel, que es la unidad básica que representa el nivel de intensidad de luz de cada sección de la imagen, formando así una matriz de MxN píxeles. El valor de estos píxeles varía en una escala de colores que va desde 0, que representa el negro absoluto, hasta 255, que representa el blanco absoluto. Es importante destacar que debido a esta definición de la escala de color, la representación más básica de una imagen en su versión digital siempre será en escala de grises.

Para reproducir una imagen de manera precisa, existen diversas formas según la combinación de píxeles y matrices que se generen. La configuración más conocida y utilizada es la R.G.B., que se compone de la convolución de tres matrices con píxeles en la escala de 0 a 255, pero una matriz está en escala de rojo, otra en verde y la última en azul. Esta combinación permite simular cualquier color del espectro conocido por el ojo humano (G. Jaramillo, 2009). A continuación, se muestra un ejemplo de la descomposición de una imagen en las matrices mencionadas.

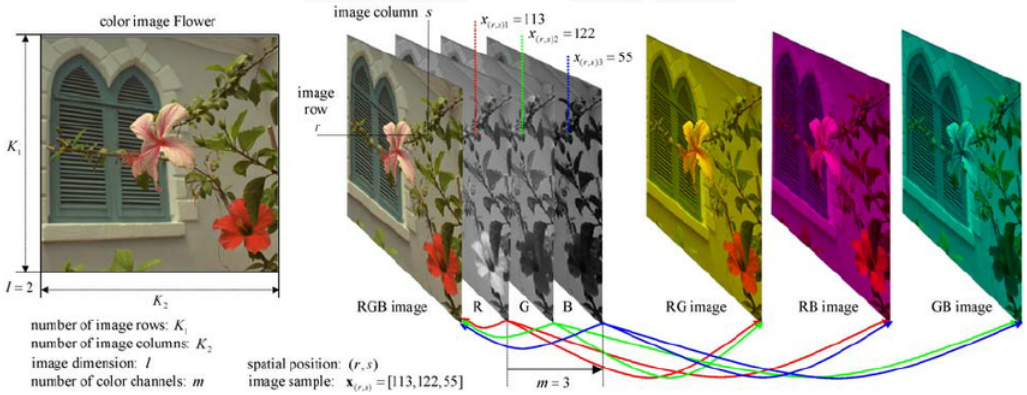
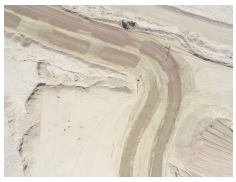


Figura 4.3: Descomposición visual RGB tomado de Single-Sensor Camera Image Processing.(LUKAC, RASTISLAV & Plataniotis, Konstantinos. 2006).

y 6 imágenes de la mina a rajo abierto como muestras a analizar a lo largo del documento para ejemplificar los posibles métodos de procesamiento a desarrollar.



(a) Muestra n°1



(b) Muestra n°2



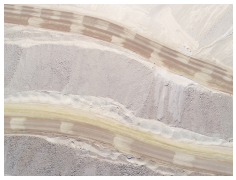
(c) Muestra n°3



(d) Muestra n°4



(e) Muestra n°5



(f) Muestra n°6

Figura 4.4: Muestra de las fotos obtenidas de 4864x3648 píxeles

4.6. Kernel

Las imágenes, al ser equivalentes digitalmente a una matriz matemática, utilizan técnicas para aplicar filtros y análisis basados en máscaras de convolución, comúnmente conocidas como "kernels", y filtros no lineales.

Un kernel es una matriz de coeficientes que, cuando se aplica a un píxel objetivo (considerando el píxel como un punto (x, y) en la matriz), produce una transformación tanto en el píxel objetivo como en sus vecinos. Esta transformación se refleja en la imagen g(x, y) en todos estos píxeles. Aunque la forma y el tamaño del kernel pueden variar, generalmente se trabaja con máscaras de convolución cuadradas, como se muestra en la siguiente figura:

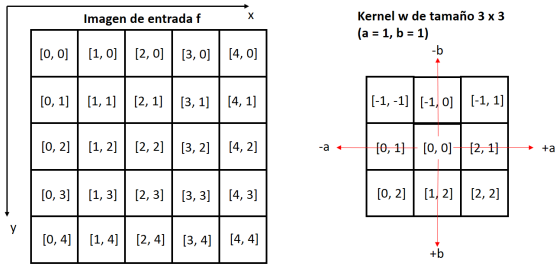


Figura 4.5: Representación Kernel tomado de El concepto de la convolución en gráficos, para comprender las convolutional neural networks (CNN) o redes convolucionadas. (Cuartas, J. 2021).

Para llevar acabo el procesamiento mediante Kernels se sigue la lógica del Teorema de Convolución en el espacio:

$$g(x, y) = h(x, y) * f(x, y) \tag{4.1}$$

En el cual, cada píxel de la nueva imagen se obtiene mediante la sumatoria de la multiplicación del kernel por los píxeles contiguos:

$$W(x, y) = \sum_{i=1}^{m} \sum_{i=1}^{n} w_i z_i \tag{4.2}$$

Siendo W la máscara de mxn, w_i los coeficientes del kernel y z_i los valores de la imagen que corresponden a dichos coeficientes en escala de grises.(P. Querejeta, 2015), lo que se explica gráficamente en la siguiente figura.

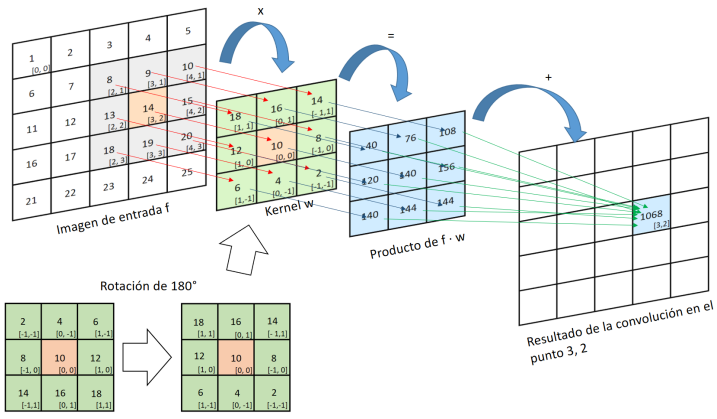


Figura 4.6: Proceso de Convolución tomado de El concepto de la convolución en gráficos, para comprender las convolutional neural networks (CNN) o redes convolucionadas. (Cuartas, J. 2021).

4.7. OpenCV

Para el desarrollo de sistemas y aplicaciones de visión artificial, existen diversas herramientas, programas y librerías disponibles, entre las cuales se destaca la librería OpenCV (Open Source Computer Vision Library) (OpenCV, 2020).

OpenCV es una librería de software de código abierto diseñada para aplicaciones de visión por computadora y aprendizaje automático. Consiste en una colección de archivos, programas, códigos y funciones que pueden ser utilizados a través de varios lenguajes de programación. Fue creada con el propósito de proporcionar una infraestructura común para aplicaciones de visión artificial y acelerar el desarrollo de sistemas de percepción automática en productos comerciales.

Esta librería cuenta con más de 2500 algoritmos optimizados, que incluyen tanto algoritmos clásicos como de última generación en el campo de la visión artificial y el aprendizaje automático. Sus capacidades abarcan desde la detección y reconocimiento de rostros, el seguimiento de movimientos en cámaras, la generación de modelos 3D hasta la identificación de objetos, entre otras aplicaciones.

En este caso, se hace uso de OpenCV y su capacidad para identificar objetos y figuras en imágenes, aprovechando sus funciones integradas. Estas funciones se enumeran en la siguiente tabla y se explicarán a medida que se utilicen a lo largo del desarrollo de este texto.

Al comprender la lógica detrás del uso de kernels y conocer las herramientas proporcionadas por OpenCV, es posible explorar los diversos tipos de análisis, filtros y procesamientos que se pueden llevar a cabo en una imagen utilizando estos elementos.

Parametro	Descripición	Proceso	Funciones	
a	Imagen	imagen	imread(a,IMREAD_GRAYSCALE)	imwrite(nombre,a)
t	umbral		bitwise_and(a,a, mask)	
l	pixel limite	Suavizamiento	blur(a,k)	medianBlur(image, ksize)
mu	metodo de binarización		GaussianBlur(image, k, d)	
mua	metod adaptativo de bin.	Binarización	cv.threshold(a, t, l, mu)	THRESH_BINARY
c	Constante de normalzacion		THRESH_TRUNC	THRESH_TOZERO
k	kernel size	Binarización adaptativa	cv.adaptiveThreshold(a, l, mua, mu, k, c)	ADAPTIVE_THRESH_MEAN_C
d	Desviación estandar		ADAPTIVE_THRESH_GAUSSIAN_C	THRESH_OTSU
cc	Identificador de contorno	Operaciones morfologicas	erode(a,k,i)	dilate(a,k,i)
cr	recuperador de contorno		morphologyEx(i, MORPH_OPEN, k)	morphologyEx(i, MORPH_CLOSE, k)
cnt	contorno		morphologyEx(i, MORPH_GRADIENT, k)	morphologyEx(i, MORPH_BLACKHAT, k)
mask	mascara de contornos		morphologyEx(i, MORPH_TOPHAT, k)	
		Segmentación por bordes	Canny(a,MinVal,MaxVal)	
		Contornos	findContours(a, cc, cr)	RETR_EXTERNAL
			CHAIN_APPROX_SIMPLE	drawContours(a, cnt, -1, 255, -1)
		Geometria contornos	contourArea(cnt)	arcLength(cnt,True)
			moments(cnt)	minAreaRect(cnt)
			meanStdDev(a,mask)	minMaxLoc(a,mask)

Figura 4.7: Listado de funciones a utilizar obtenido en base a OpenCV: Image Processing. (s. f.).

Al comprender la lógica detrás del uso de kernels y conocer las herramientas proporcionadas por OpenCV, es posible explorar los diversos tipos de análisis, filtros y procesamientos que se pueden llevar a cabo en una imagen utilizando estos elementos.

4.8. Suavizamiento

El suavizado o desenfoque es una de las operaciones de procesamiento de imágenes mas simples y frecuentemente utilizadas. La razón principal de su uso es eliminar el ruido presente en una imagen. Este proceso implica transformar secciones de la fotografía que, debido a su forma, podrían considerarse como fondo u o objeto cuando en realidad son lo contrario. Estas operaciones cuentan con el siguiente fundamento matemático(Castellano Carvajale et all, 2023).

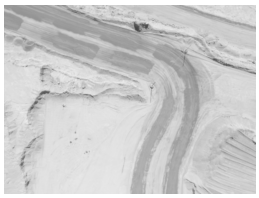
$$g(i, j) = \sum_{k,l} f(i + k, j + l)h(k, l) \tag{4.3}$$

Donde h(k,l) es el Kernel o los coeficientes de cada filtro y g(i,j) es el pixel resultante por la operación. Existen diversos tipos de filtro de suavizamiento y en este trabajo se hará uso de los 3 mas utilizados en la literatura

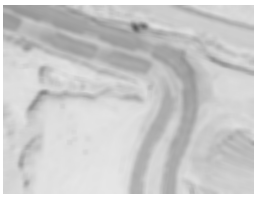
4.8.1. Desenfoque Promedio (Average Blur):

Filtro en que cada píxel de salida es el promedio de todos los píxeles presentes en la vecindad definida por el kernel aplicado. La representación del Kernel aplicado es la siguiente:

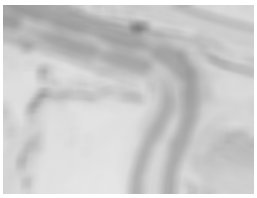
$$K = \frac{1}{K_{\text{width}} \cdot K_{\text{height}}} * \begin{bmatrix} 1 & .. & 1 \\ .. & .. & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{4.4}$$



(a) muestra n°1 kernel de 21X21



(b) muestra n°1 kernel de 101X101

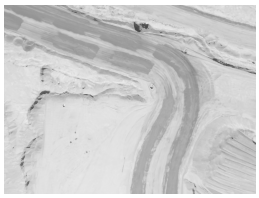


(c) muestra n°1 kernel de 201X201

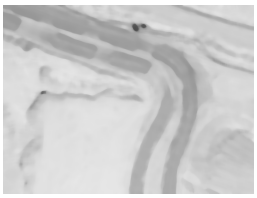
Figura 4.8: Imágenes obtenidas mediante el uso de la función cv.blur(a,k) con a = muestra ; k=21,101,201

4.8.2. Desenfoque Medio (Median Blur):

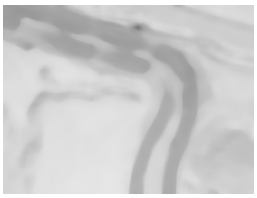
Desenfoque en el cual se calcula la mediana en relación a todos los píxeles que se encuentran dentro del kernel definido, y se reemplaza el valor del píxel central por esta mediana. es especialmente efectivo para eliminar el conocido ruido de sal y pimienta, que se manifiesta como los típicos puntillos que aparecen en una fotografía y que no son parte de la imagen original. Esto se debe al hecho de que tiende a despreciar los valores mas extremos dentro de una vecindad definida, dando mayor peso a los valores que se repiten con mas frecuencia.



(a) muestra n°1 kernel de 21X21



(b) muestra n°1 kernel de 101X101



(c) muestra n°1 kernel de 201X201

Figura 4.9: Imágenes obtenidas mediante el uso de la función cv.medianBlur(a,k) con a = muestra ; k=21,101,201

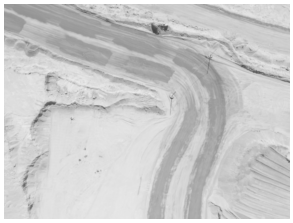
4.8.3. Desenfoque Gaussiano (Gaussian Blur):

El desenfoque mas útil según la literatura, este depende de la convolución sobre un kernel de distribución gaussiana, es decir que los píxeles que se encuentren al centro de la vecindad siempre contarán con un mayor peso respecto a los que se encuentren en los extremos de esta, tal como dicta la ecuación:

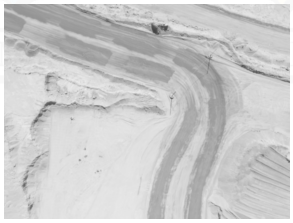
$$G_0(x,y) = A * e^{(\frac{-(x-\mu_x)^2}{2*\sigma_x^2} + \frac{-(y-\mu_y)^2}{2*\sigma_y^2})} \tag{4.5}$$

como por ejemplo en la siguiente matriz de 3x3:

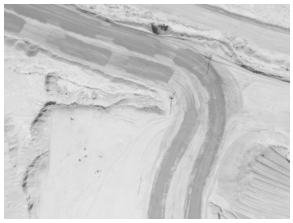
$$K_{3x3} = \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} \tag{4.6}$$



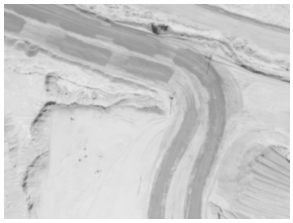
(a) Media=21; Desviación= 10



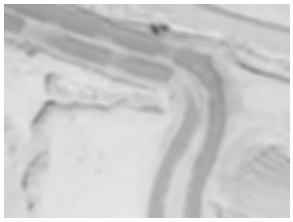
(b) Media=21; Desviación= 100



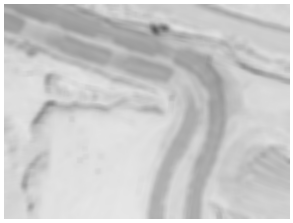
(c) Media=21; Desviación= 500



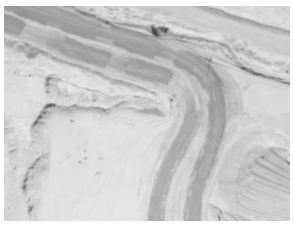
(d) Media=101; Desviación= 10



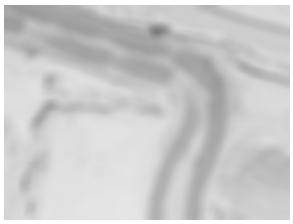
(e) Media=101; Desviación= 100



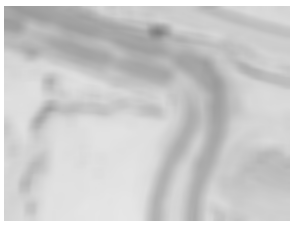
(f) Media=101; Desviación= 500



(g) Media=201; Desviación= 10



(h) Media=201; Desviación= 100



(i) Media=201; Desviación= 500

Figura 4.10: Imágenes obtenidas mediante el uso de la función cv.GaussianBlur(image, k, d) con a = muestra; k=21,101,201; d = 10,100,500

4.9. Binarización

La binarización es el proceso de segmentación más sencillo, que busca principalmente separar una imagen en dos partes: el fondo y la forma (objetos). Esta separación se basa en la variación de intensidad de los píxeles definidos como fondo y los píxeles definidos como objetos. Para determinar si un píxel pertenece al fondo o a la forma, se compara con un umbral y se le asigna un valor específico según su relación con ese umbral. Se llama "binarización"porque se definen dos posibles valores de píxeles: generalmente 0 (negro) para representar el fondo y 255 (blanco) para representar los objetos. En este documento se presentan y analizan los 8 métodos de binarización más utilizados en el estado del arte. (Triana-Guzman et al. 2016).

*A partir de aquí, se utilizarán las muestras suavizadas con un suavizado promedio de kernel = 101 como base para las siguientes imágenes del marco teórico.

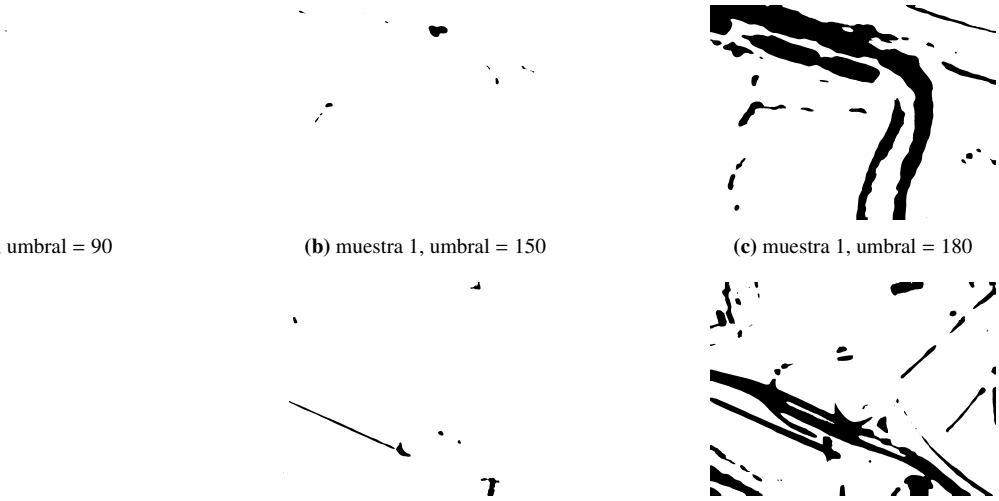
4.9.1. Binarización binaria

Método en el cual, si la intensidad del píxel a evaluar, src(x, y), es mayor que el umbral definido, el píxel se establecerá en su valor máximo posible, que en la mayoría de los casos es 255. Por otro lado, si el píxel es menor al umbral, se establece el píxel en 0. Cabe destacar que este método cuenta con su versión inversa.

Esta operación se expresa como:

$$dst(x,y) = \begin{cases} maxval & \text{si } src(x,y) > t \\ 0 & \text{En otro caso} \end{cases} \tag{4.7}$$

Es decir s



(a) muestra 1, umbral = 90 **(b)** muestra 1, umbral = 150 **(c)** muestra 1, umbral = 180

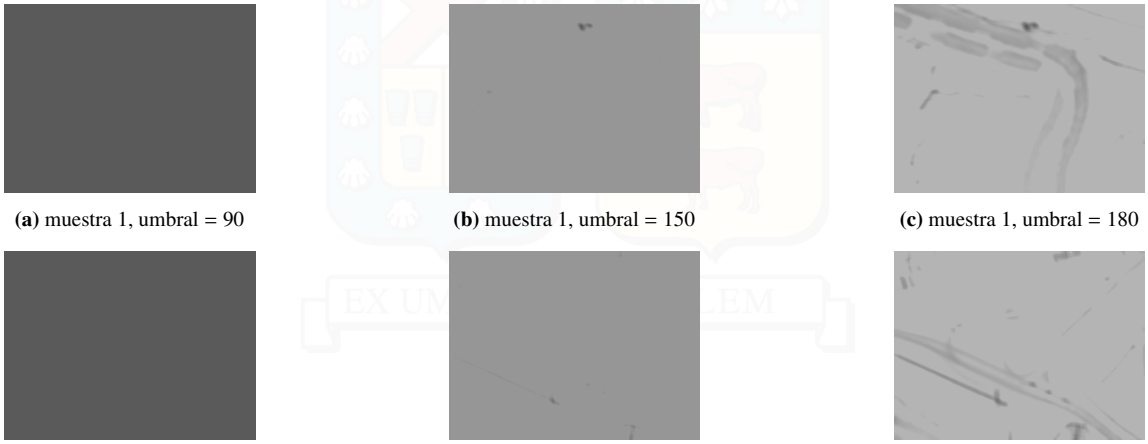
(d) muestra 2, umbral = 90 **(e)** muestra 2, umbral = 150 **(f)** muestra 2, umbral = 180

Figura 4.11: Imágenes obtenidas mediante el uso de la función cv.threshold(a, t, l, THRESH_BINARY) con a = muestra; t=90,150,180; l=255

4.9.2. Binarización truncada

Método que no busca representar la imagen en dos secciones, sino eliminar todo lo que esté por encima de cierto nivel de intensidad:

$$dst(x,y) = \begin{cases} t & \text{si } src(x,y) > t \\ src(x,y) & \text{En otro caso} \end{cases} \tag{4.8}$$



(a) muestra 1, umbral = 90 **(b)** muestra 1, umbral = 150 **(c)** muestra 1, umbral = 180

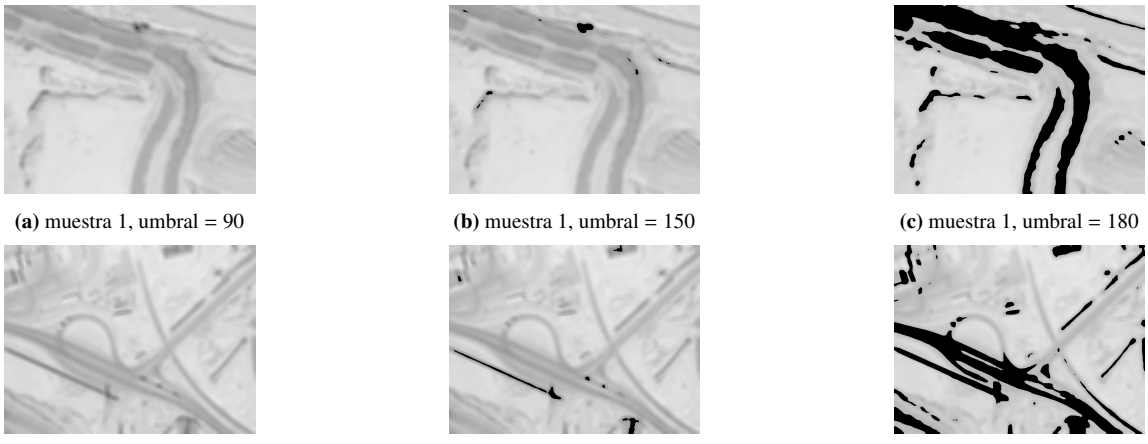
(d) muestra 2, umbral = 90 **(e)** muestra 2, umbral = 150 **(f)** muestra 2, umbral = 180

Figura 4.12: Imágenes obtenidas mediante el uso de la función cv.threshold(a, t, l, THRESH_TRUNC) con a = muestra; t=90,150,180; l=255

4.9.3. Binarización a cero

lleva los píxeles de menor intensidad que el umbral a cero. También cuenta con su versión inversa:

$$dst(x,y) = \begin{cases} src(x,y) & \text{si } src(x,y) > t \\ 0 & \text{En otro caso} \end{cases} \tag{4.9}$$



(a) muestra 1, umbral = 90 **(b)** muestra 1, umbral = 150 **(c)** muestra 1, umbral = 180

(d) muestra 2, umbral = 90 **(e)** muestra 2, umbral = 150 **(f)** muestra 2, umbral = 180

Figura 4.13: Imágenes obtenidas mediante el uso de la función cv.threshold(a, t, l, THRESH_TOZERO) con a = muestra; t=90,150,180; l=255

4.10. Binarización adaptativa

Hasta el método anterior, se recurre a un valor global como umbral, el cual es definido de forma arbitraria o teniendo un conocimiento previo del rango de intensidad de los píxeles en la imagen que se está analizando. Esto puede no ser adecuado en todos los casos, por ejemplo, si una imagen cuenta con diversas condiciones de iluminación en distintas áreas de la imagen. Para abordar estos casos, se han desarrollado métodos de binarización con umbrales adaptativos. En esta ocasión, se presentan tres métodos adaptativos.

4.10.1. Umbral adaptativo promedio

Se hace uso del mismo kernel que el suavizamiento promedio para una vecindad definida. Sin embargo, en vez de definir el valor de cada píxel escaneado, el valor determinado funciona como el umbral local para cada vecindad dentro de la imagen. Se efectúa una binarización binaria en cada una de estas vecindades según el umbral que corresponda.



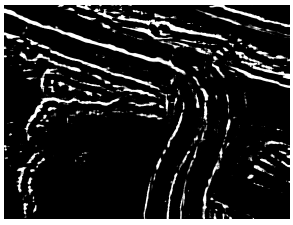
(a) muestra 1, Kernel= 51, C = -2



(b) muestra 1, Kernel= 51, C = 0



(c) muestra 1, Kernel= 51, C = 2



(d) muestra 1, Kernel= 101, C = -2



(e) muestra 1, Kernel= 101, C = 0



(f) muestra 1, Kernel= 101, C = 2



(g) muestra 1, Kernel= 201, C = -2



(h) muestra 1, Kernel= 201, C = 0

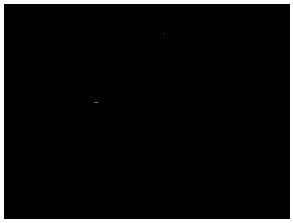


(i) muestra 1, Kernel= 201, C = 2

Figura 4.14: Imágenes obtenidas mediante el uso de la función cv.adaptiveThreshold(a, l, ADAPTIVE_THRESH_MEAN_C,THRESH_BINARY, k, c) con a = muestra; K=51,101,201; l=255; c = -2,0,2

4.10.2. Umbral adaptativo gaussiano

Equivalente al Umbral adaptativo promedio, pero utilizando un Kernel Gaussiano.



(a) muestra 1, Kernel= 51, C = -2



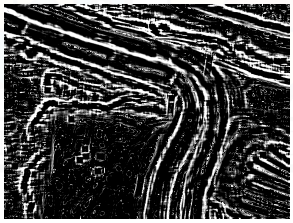
(b) muestra 1, Kernel= 51, C = 0



(c) muestra 1, Kernel= 51, C = 2



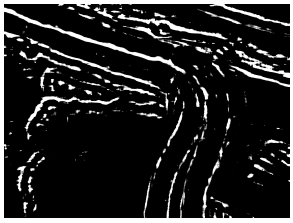
(d) muestra 1, Kernel= 101, C = -2



(e) muestra 1, Kernel= 101, C = 0



(f) muestra 1, Kernel= 101, C = 2



(g) muestra 1, Kernel= 201, C = -2



(h) muestra 1, Kernel= 201, C = 0



(i) muestra 1, Kernel= 201, C = 2

Figura 4.15: Imágenes obtenidas mediante el uso de la función cv.adaptiveThreshold(a, l,ADAPTIVE_THRESH_GAUSSIAN_C,THRESH_BINARY, k, c) con a = muestra; K=51,101,201; l=255; c = -2,0,2

4.10.3. Binarización de Otsu

Además de los métodos mencionados anteriormente, que determinan diversos umbrales locales, se cuenta con el método de Otsu. Este método determina un umbral global óptimo mediante el análisis del histograma de la imagen.

Mediante un análisis de varianzas, se encuentra el valor específico del umbral que se adapta al procesamiento según los niveles de luminosidad y tamaño de objetos presentes en el cuadro principal de la imagen (P. Huamaní, 2015). Además, minimiza la varianza entre las clases de los píxeles blancos y negros (Maldonado et al., 2012). Dicho proceso se realiza según el siguiente procedimiento matemático, que trabaja suponiendo que la imagen a analizar corresponde a una imagen bimodal, es decir, una imagen que cuenta con dos picos de intensidad distintos en la representación en forma de histograma de la luminosidad presente en la imagen. Por lo tanto, la imagen debe ser separada en dos clases o clusters.

La varianza global de estas dos clases se representa como:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \tag{4.10}$$

Donde

$$q_1(t) = \sum_{i=1}^{t} P(i) \quad \& \quad q_2(t) = \sum_{i=t+1}^{I} P(i) \tag{4.11}$$

Corresponden al 'peso' de cada clase según las probabilidades de que un píxel de la imagen corresponda a una de estas dos clases.

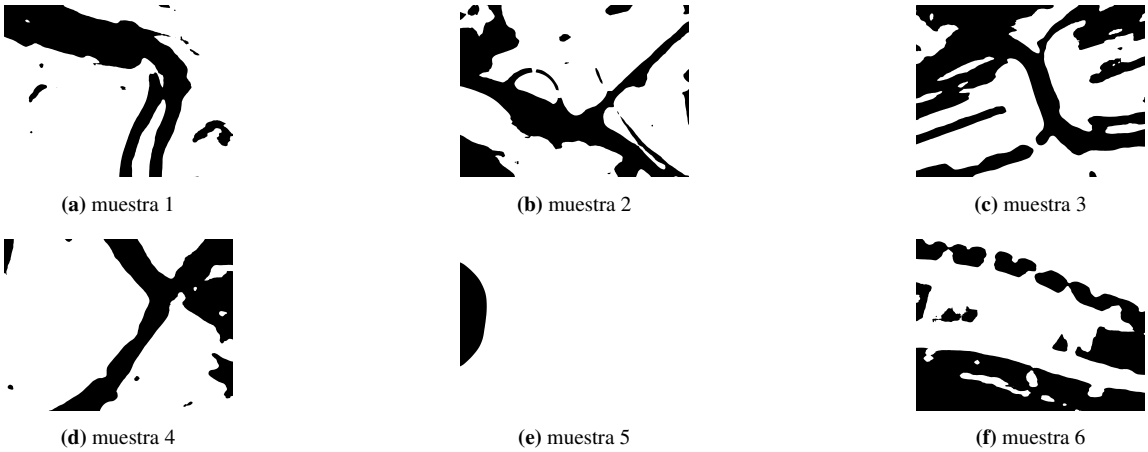
$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \quad \& \quad \mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i}{q_2(t)} \tag{4.12}$$

A las medias de estos 'pesos' y

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \quad \& \quad \sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)} \tag{4.13}$$

a las varianzas de cada uno.

Esta optimización encuentra el valor de "t"que se sitúa en el punto medio de los dos picos de intensidad presentes en la imagen. De esta manera, las varianzas de intensidad con respecto a cada uno de estos picos son mínimas, lo que permite obtener una segmentación mucho más precisa y definida (Murzova, A., 2021).



(a) muestra 1 **(b)** muestra 2 **(c)** muestra 3

(d) muestra 4 **(e)** muestra 5 **(f)** muestra 6

Figura 4.16: Imágenes obtenidas mediante el uso de la función cv.threshold(a, 0, l, cv.THRESH_BINARY + cv.THRESH_OTSU) con a = muestra; l=255

4.11. Operaciones morfológicas:

Al igual que el suavizado, existen un conjunto de filtros conocidos como operaciones morfológicas que también buscan eliminar el ruido de las imágenes. Sin embargo, estas operaciones tienen objetivos más específicos, como:

1. Separar elementos individuales.
2. Unir elementos dispares.
3. Identificar y rellenar agujeros dentro de una imagen.
4. Identificar y eliminar protuberancias de intensidad dentro de una imagen. (Jesús, 2021)

*Es importante destacar que las transformaciones mencionadas se aplican a muestras que previamente han pasado por un suavizado promedio con un kernel de 101x101. Luego, se realiza una binarización de Otsu, seguida de una binarización inversa para intercambiar lo que se identifica como fondo y forma en la imagen.

4.11.1. Erosión:

Esta operación reduce el tamaño de los objetos blancos en la imagen, eliminando los píxeles frontera del objeto.

El operador calcula el mínimo local en la vecindad definida por el kernel de convolución para cada píxel escaneado. En la imagen, el objeto se representa en blanco, y a medida que los píxeles se vuelven más pequeños, se acercan al negro. Al tomar el mínimo en cada vecindad, se produce una erosión en el objeto, ya que varios píxeles del borde del objeto se convierten en fondo al ser transformados a negro.

Por lo tanto el operador de erosión es dst(x,y) = min(x′,y′).



(a) muestra 1, kernel = 21x21 **(b)** muestra 1, kernel = 101x101 **(c)** muestra 1, kernel = 201x201

(d) muestra 2, kernel = 21x21 **(e)** muestra 2, kernel = 101x101 **(f)** muestra 2, kernel = 201x201

Figura 4.17: Imágenes obtenidas mediante el uso de la función cv.erode(a,k,i) con a = muestra; k=21,101,201; i=1

4.11.2. Dilatación:

a operación de dilatación es la hermana de la erosión ya que sigue el mismo proceso, pero a diferencia de esta, el píxel que se mantiene es el máximo dentro de la vecindad definida por el kernel. Como resultado, se produce el efecto opuesto a la erosión: en las secciones del borde, una gran cantidad de píxeles negros se convierten en blancos, incorporándose al objeto y generando un efecto de dilatación en el objeto. el operador es dst(x,y)=max(x′,y′)



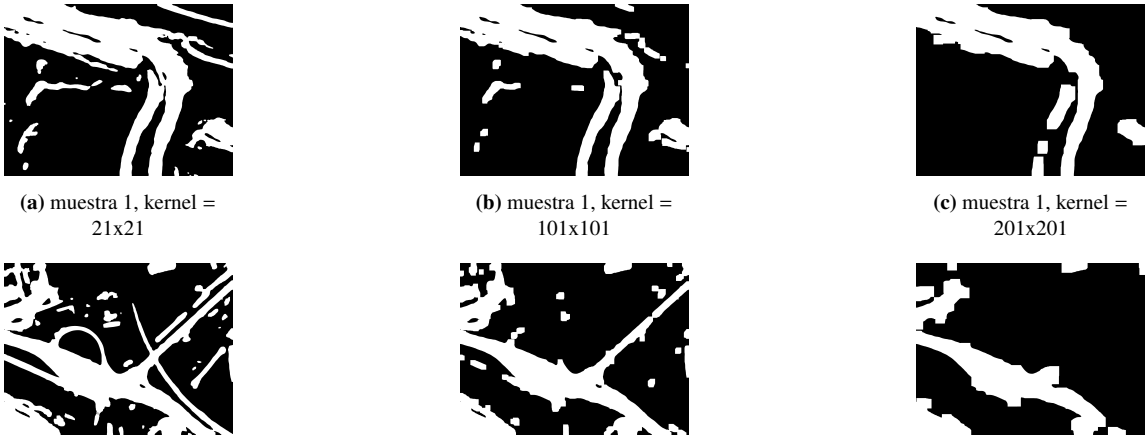
(a) muestra 1, kernel = 21x21 **(b)** muestra 1, kernel = 101x101 **(c)** muestra 1, kernel = 201x201

(d) muestra 2, kernel = 21x21 **(e)** muestra 2, kernel = 101x101 **(f)** muestra 2, kernel = 201x201

Figura 4.18: Imágenes obtenidas mediante el uso de la función cv.erode(a,k,i) con a = muestra; k=21,101,201; i=1

4.11.3. Apertura:

La apertura es una operación que combina una erosión de la imagen seguida de una dilatación. Se utiliza para eliminar ruido y objetos pequeños del fondo, ya que si estos son pequeños en relación a la imagen en general y al kernel definido, serán eliminados al pasar por un proceso de erosión, ya que probablemente cada vecindad de esta porción cuente con un píxel negro. Posteriormente, el objeto que sea rescatado puede ser mejor definido y delimitado gracias a pasar por un proceso de dilatación.



(a) muestra 1, kernel = 21x21 **(b)** muestra 1, kernel = 101x101 **(c)** muestra 1, kernel = 201x201

(d) muestra 2, kernel = 21x21 **(e)** muestra 2, kernel = 101x101 **(f)** muestra 2, kernel = 201x201

Figura 4.19: Imágenes obtenidas mediante el uso de la función cv.morphologyEx(i, MORPH_OPEN, k) con a = muestra; k=21,101,201; i=1

4.11.4. Clausura:

Siguiendo la misma lógica que el proceso de apertura, la operación de clausura es la dilatación de la imagen seguida por su erosión. Esto permite eliminar pequeños hoyos presentes en lo que se ha definido como objeto y, posteriormente, mediante la erosión, volver a "pulir"los bordes de este.



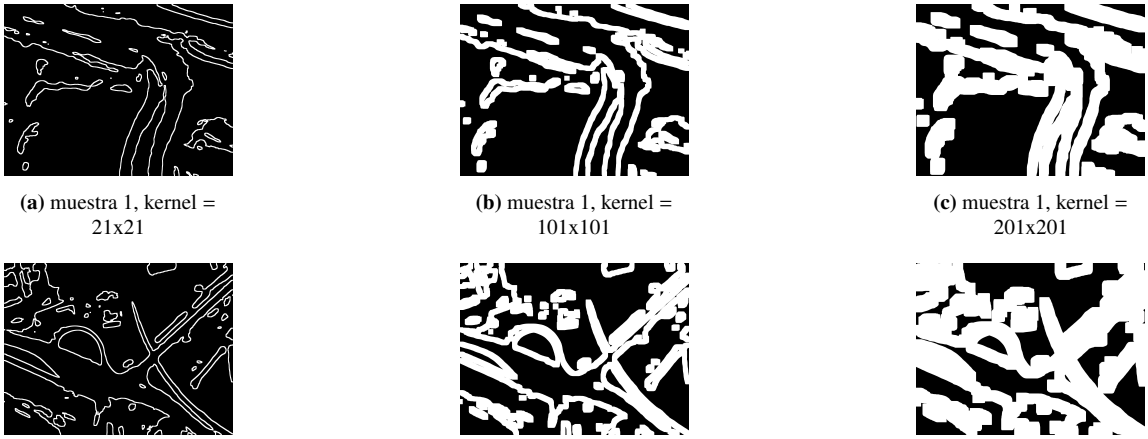
(a) muestra 1, kernel = 21x21 **(b)** muestra 1, kernel = 101x101 **(c)** muestra 1, kernel = 201x201

(d) muestra 2, kernel = 21x21 **(e)** muestra 2, kernel = 101x101 **(f)** muestra 2, kernel = 201x201

Figura 4.20: Imágenes obtenidas mediante el uso de la función cv.morphologyEx(i, MORPH_CLOSE, k) con a = muestra; k=21,101,201; i=1

4.11.5. Gradiente morfológico:

El gradiente morfológico es la diferencia entre la dilatación y la erosión de la imagen. Se utiliza para obtener bordes de objetos.



(a) muestra 1, kernel = 21x21 **(b)** muestra 1, kernel = 101x101 **(c)** muestra 1, kernel = 201x201

(d) muestra 2, kernel = 21x21 **(e)** muestra 2, kernel = 101x101 **(f)** muestra 2, kernel = 201x201

Figura 4.21: Imágenes obtenidas mediante el uso de la función cv.morphologyEx(i, MORPH_GRADIENT, k) con a = muestra; k=21,101,201; i=1

4.11.6. Top Hat:

Es la diferencia entre la imagen original y su apertura. Se utiliza para resaltar pequeños detalles u obtener pequeños objetos de la imagen.



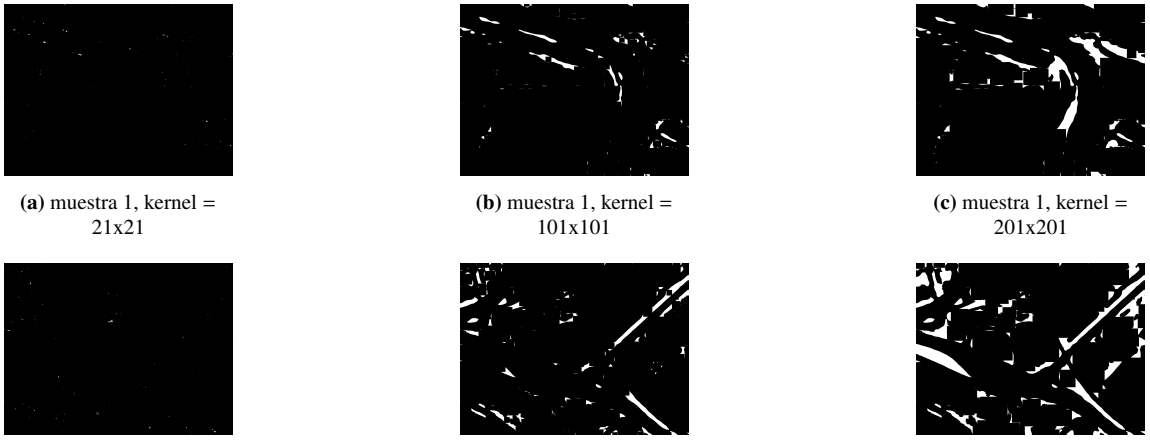
(a) muestra 1, kernel = 21x21 **(b)** muestra 1, kernel = 101x101 **(c)** muestra 1, kernel = 201x201

(d) muestra 2, kernel = 21x21 **(e)** muestra 2, kernel = 101x101 **(f)** muestra 2, kernel = 201x201

Figura 4.22: Imágenes obtenidas mediante el uso de la función cv.morphologyEx(i, MORPH_TOPHAT, k) con a = muestra; k=21,101,201; i=1

4.11.7. Black Hat:

Es la diferencia entre la clausura de la imagen y la imagen original. Se utiliza para resaltar regiones oscuras en fondos claros o, en especifico, para mantener los hoyos eliminados de los objetos.



(a) muestra 1, kernel = 21x21 **(b)** muestra 1, kernel = 101x101 **(c)** muestra 1, kernel = 201x201

(d) muestra 2, kernel = 21x21 **(e)** muestra 2, kernel = 101x101 **(f)** muestra 2, kernel = 201x201

Figura 4.23: Imágenes obtenidas mediante el uso de la función cv.morphologyEx(i, MORPH_BLACKHAT, k) con a = muestra; k=21,101,201; i=1

4.12. Contornos

Al entender el proceso de segmentación, solo queda contextualizar cómo se puede pasar de fotografías segmentadas a la clasificación de objetos dentro de estas, para así identificar sus características y propiedades. Esto es posible mediante el reconocimiento de los contornos del objeto. Hasta el apartado anterior, se había hablado de bordes, que únicamente se identifican como frontera para la segmentación de la imagen. Ahora, al hablar de contornos, nos referimos a todos los bordes que puedan representar el límite de una figura que pueda tener cierre, es decir, que matemáticamente se pueda identificar como un objeto definido.

Los contornos pueden ser explicados como una curva que une todos los puntos continuos sobre un límite, que cuentan con el mismo color o nivel de intensidad. Gracias a la definición de estos contornos, es posible realizar análisis geométricos de las formas, así como detectar y reconocer objetos. (OpenCV: Contours: Getting started, s. f.)

Al identificar figuras mediante sus contornos, es posible generar diversos análisis sobre:

- Su geometría: momentos, centroide, área, perímetro; rectángulo de área mínima que cubra al objeto, lo que permite identificar altura, ancho y ángulo de inclinación en la imagen. (OpenCV: Contour features, s. f.)
- Ratios sobre esta geometría: Aspecto, Extensión, Solidez.
- Características lumínicas: píxel mínimo y su ubicación en la figura, píxel máximo y su ubicación en la figura, intensidad media y desviación estándar de la intensidad. (OpenCV: Contour Properties, s. f.)

4.13. Intersecciones Bit a Bit

Dentro de la computación, el lenguaje básico para la generación de todo lo relacionado con el procesamiento de imágenes es la operación bit a bit, que implica realizar operaciones lógicas sobre combinaciones de ceros y unos para obtener un resultado esperado. En el caso del procesamiento de imágenes, se puede utilizar este mecanismo para preservar únicamente ciertas partes de una imagen en base a una plantilla que sirva como molde. Por ejemplo, en todo el proceso de segmentación al binarizar las imágenes, se pierde la información de la imagen original, manteniendo únicamente una plantilla que separa en blanco y negro qué es figura y qué es fondo en esta. Entonces, ¿qué se puede hacer para mantener esta clasificación pero ahora visualizándola en la imagen original? Para esto, se recurre a un operador lógico de intersección(Som, s. f.).

En este caso la intersección se genera entre la imagen original, donde todo píxel se considera como un uno al ser una imagen coloreada, y por otro lado, la imagen plantilla obtenida a través de todo el proceso de clasificación, donde el fondo o lo negro es un cero y las figuras son un uno, entregando como imagen final las secciones de la imagen original que se encuentren demarcadas por las figuras identificadas.



(a) Imagen original **(b)** Mascara **(c)** Intersección

Figura 4.24: Muestra de las fotos obtenidas de 4864x3648 píxeles

5 | Metodología

5.1. Procesamiento de imágenes del vuelo de drones sobre minas de rajo abierto

El uso de diversas matrices de convolución es fundamental en la siguiente metodología propuesta. Se sigue un flujo lógico para generar los siguientes estados en las imágenes que se analizan, con el objetivo final de detectar los caminos dentro de ellas.

1. **Reducir el peso de la imagen :** la conversión de una imagen en base a tres dimensiones de intensidad, R.G.B., a una imagen de una sola dimensión, lo que conlleva una significativa reducción en la necesidad de procesamiento y permite identificar contrastes dentro de la imagen de manera más simple.
2. **Suavizar la imagen:** Se reduce las variaciones de intensidad entre píxeles vecinos.
3. **Eliminar el ruido:** Se eliminan aquellos píxeles cuyo nivel de intensidad es muy diferente al de sus vecinos.
4. **Segmentar:** Se identifican las coordenadas de los píxeles donde se produce un cambio brusco en la función intensidad, realzando y detectando bordes entre la figura y el fondo,
5. **Clasificar los objetos identificados:** Identificar si los contornos de los objetos segmentados son capaces de definir figuras y obtener toda la información relevante a estas, para evaluar la efectividad de cada método estudiado.

Con toda la información expuesta en el marco teórico, se procede a generar el siguiente flujo sobre una muestra de seis fotografías seleccionadas buscando abordar las distintas formas en que se pueda representar un camino dentro de las imágenes que componen todo el dataset a analizar y utilizarlas para pivotar y definir los parámetros a aplicar en cada fase. Posteriormente, se replicará el proceso en las 1906 imágenes de vuelos sobre la mina a rajo abierto para evaluar la efectividad de este método en la creación de un dataset de caminos en la mina. Cabe destacar que este es un proceso que involucra una cantidad considerable de variables, como el método en sí para cada fase del proceso o el tamaño de los kernel a utilizar, el umbral a usar en una binarización, entre otros.

Debido a esto, se cuentan con dos enfoques para identificar los parámetros a utilizar:

1. Valerse de lo aprendido en el estado del arte y descartar métodos que no apliquen directamente a la problemática en específico o que, por su forma de trabajar, se adapten mejor al análisis de esta.
2. En el caso de la definición del tamaño del kernel de un filtro de suavizado o del umbral a utilizar en una binarización, se procede a generar iteraciones con una selección aleatoria y arbitraria de parámetros hasta que una de estas logre identificar un camino en la mayoría de las fotografías de las muestras a analizar.

5.2. Flujo Propuesto

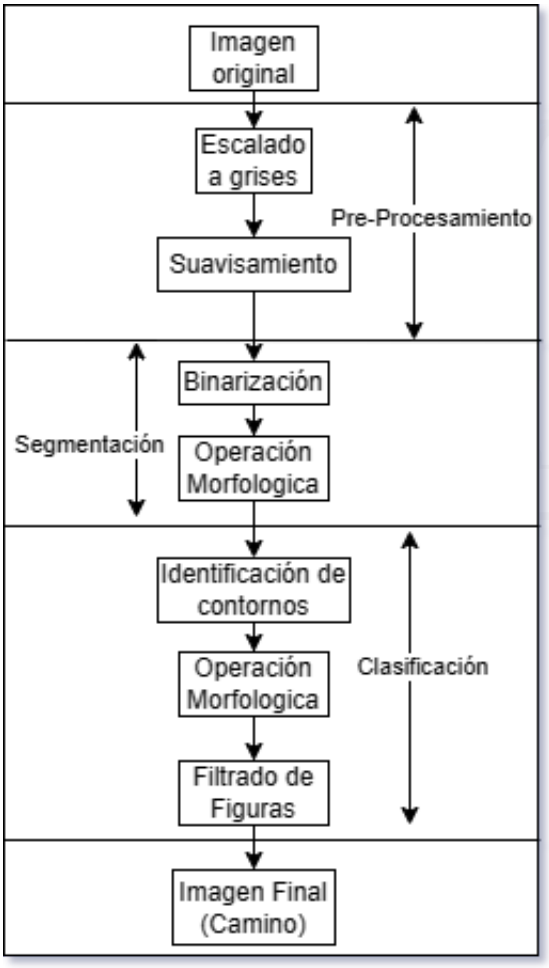


Figura 5.1: Metodología Propuesta(Paso n°1), Elaboración propia

Concibiendo el Flujo anterior como el Paso n°1 también se presenta un diagrama para la creación del dataset final con los caminos obtenidos.

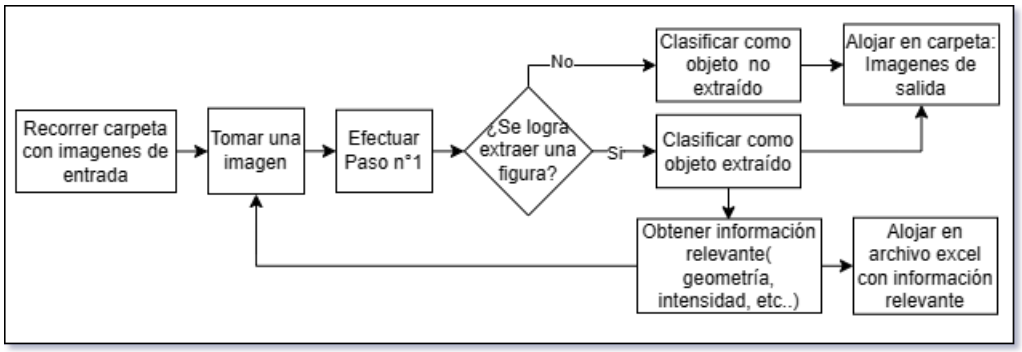


Figura 5.2: Metodología Propuesta(Paso n°2), Elaboración propia

6 | Desarrollo

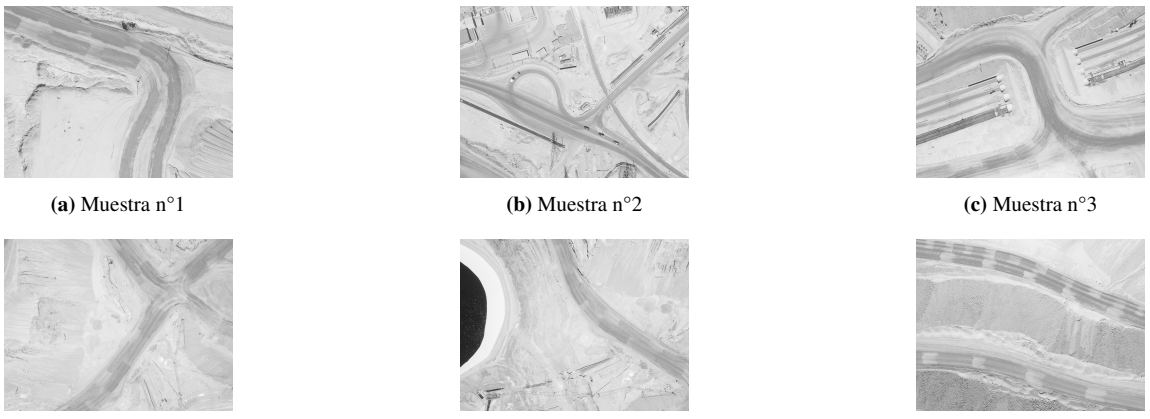
6.1. Conversión a escala de grises

Primero que todo, es importante buscar la forma más simple y eficiente de trabajar el procesamiento de las imágenes. Para esto, lo más conveniente es convertir las imágenes a escala de grises, transformándolas desde su configuración RGB a una versión en la que la intensidad de sus píxeles se mueva de 0 a 255, derivando únicamente en tonalidades grises. Esto ofrece las siguientes ventajas:

- **Reducción de dimensión:** Las imágenes RGB cuentan con tres canales de color definidos en matrices que, al intersectarse, logran la representación de cualquier color. Por otro lado, una imagen en escala de grises cuenta con un solo canal de color, lo que resulta en la representación de la imagen en base a una sola matriz.
- **Reducción de la complejidad de los modelos**: Por ejemplo, para el caso de análisis mediante redes neuronales, una imagen de 10x10 píxeles, al tener 3 canales, cuenta con 300 datos de entrada distintos, mientras que la misma imagen en escala de grises solo cuenta con 100 datos de entrada diferentes.
- **Uso de algoritmos:** gran variedad de algoritmos en el procesamiento de imágenes son funcionales únicamente cuando se trabaja con escala de grises, como por ejemplo la detección de bordes.(Geeks for Geeks, 2022)

Por lo tanto se hace la conversión para las seis imágenes de muestra. Con esta trasformación, solo para la muestra, se pasa de trabajar con 137MB de información a 50,3MB:

6.1.1. Muestras en escalada de grises



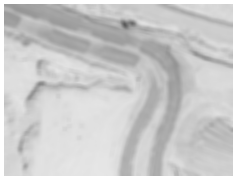
(a) Muestra n°1 **(b)** Muestra n°2 **(c)** Muestra n°3

(d) Muestra n°4 **(e)** Muestra n°5 **(f)** Muestra n°6

Figura 6.1: Muestra de las fotos obtenidas de 4864x3648 píxeles en escala de grises

6.2. Suavizamiento

Posterior a la conversión a grises, se continúa con el suavizamiento de las imágenes para eliminar el ruido en ellas y facilitar la diferenciación entre los bordes reales y los bordes que pueden surgir debido a la imagen en sí. El desafío en esta etapa es determinar cuál de los tres algoritmos es el más adecuado para el proceso y qué tamaño de kernel debería utilizarse, además de la desviación estándar en el caso del suavizado gaussiano. Como se puede ver en las imágenes presentadas en el marco teórico, en esta primera etapa, la diferencia entre alguno de los tres métodos es apenas perceptible en la imagen resultante, pero el tamaño del kernel sí tiene un impacto en la nitidez de la foto. Dado que el objetivo del suavizado es eliminar el ruido y lograr una mejor separación de los bordes, antes de analizar la magnitud del kernel y el método más adecuado, se decide arbitrariamente utilizar el método de suavizado de mediana con un kernel de 101x101. A continuación, se analizará qué método de binarización es capaz de generar una segmentación adecuada del fondo y los objetos, manteniendo la figura de los caminos, para luego determinar si la diferencia entre estos métodos tiene un gran impacto en el resultado. Las imágenes resultantes con el método elegido provisionalmente son las siguientes.



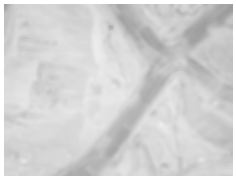
(a) Muestra n°1



(b) Muestra n°2



(c) Muestra n°3



(d) Muestra n°4



(e) Muestra n°5



(f) Muestra n°6

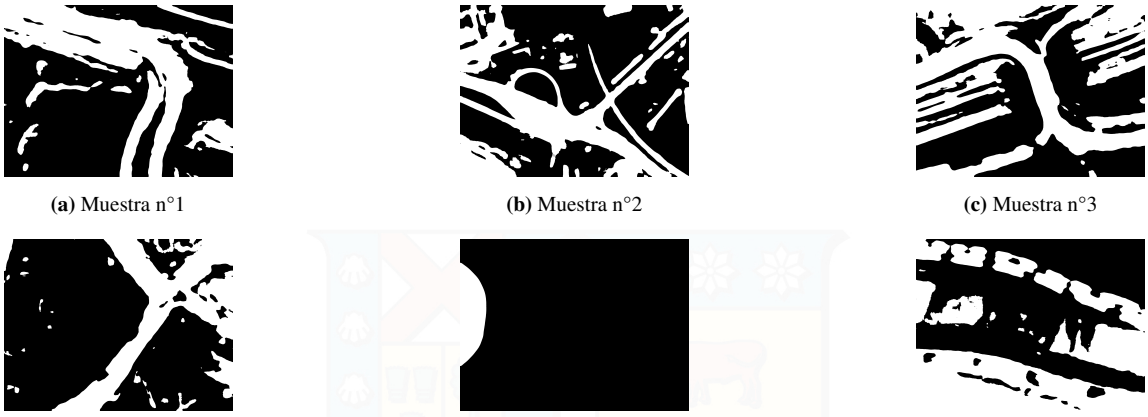
Figura 6.2: Muestras con aplicación de suavizamiento promedio de kernel 101x101

6.3. Binarización

Sabiendo que el principal objetivo de la binarización es convertir la imagen en un conjunto dual de Fondo/Objeto, surge nuevamente el desafío de identificar cuál de los seis métodos de binarización revisados, junto con sus variables, es el más adecuado para realizar esta conversión. En este caso, se tienen dos enfoques principales: los métodos adaptativos y no adaptativos.

De los ejemplos presentados en el marco teórico, se reconoce que para lograr una binarización de calidad en los métodos no adaptativos, se requiere un conocimiento previo de las características específicas de la imagen. En este caso, se identifica que los caminos son mucho más oscuros que el resto de la imagen y que la imagen en general tiene una alta incidencia de luz. Esto significa que el umbral necesario para identificar las partes de la imagen como fondo debe ser muy alto. Además, se observa que la imagen tiene múltiples niveles de intensidad alrededor de los objetos, lo que indica que se necesitaría un umbral muy alto para tratar de considerar estas áreas como fondo. Por ejemplo, al utilizar un umbral de 180 en las muestras, gran parte del camino se considera fondo, pero no completamente, lo que puede deberse a cómo incide la luz en función de la elevación del camino.

Debido a estas observaciones, se concluye que es necesario utilizar un método de umbralización adaptativo que pueda capturar los subconjuntos de intensidad en diferentes secciones de la imagen. Además, el uso de un método adaptativo reduce la dependencia de una variable específica en el análisis. Por su fundamento matemático y por lo que se visualiza en los ejemplos presentados en el marco teórico, se elige el método de Otsu para realizar la binarización en el modelo. Es importante destacar que debido a las intensidades y colores presentes en la imagen, el método de Otsu identifica la sección del camino como fondo y el resto como figura, siguiendo la convención de que el negro representa el fondo y el blanco la figura. Para resolver esto, después de la binarización de Otsu, se aplica una binarización inversa, donde el umbral es despreciable, ya que solo hay píxeles con valores de 0 y 255. Esto permite cambiar todos los píxeles de valor 0 a 255 y viceversa, invirtiendo así el fondo y la forma en la imagen binarizada.



(a) Muestra n°1 **(b)** Muestra n°2 **(c)** Muestra n°3

(d) Muestra n°4 **(e)** Muestra n°5 **(f)** Muestra n°6

Figura 6.3: Muestras con aplicación de binarización adaptativa de otsu

De los resultados obtenidos se observa que en cinco de las seis muestras se mantiene el camino y se clasifica correctamente como objeto, mientras que en la muestra número 5 esto no sucede debido a la presencia de un objeto con un nivel de intensidad significativamente diferente al resto de la imagen. En este caso, se puede identificar que este objeto es la piscina de agua en el sector izquierdo de la fotografía, que tiene un nivel de intensidad muy bajo en comparación con el camino y el entorno circundante.

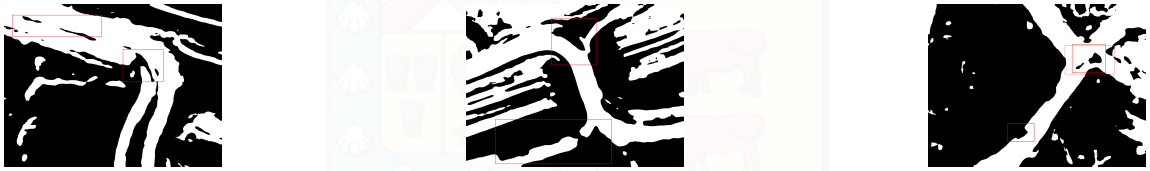
Para abordar este problema y mejorar la segmentación, se podría aplicar una binarización inversa antes de utilizar el método de Otsu. Esto convertiría el objeto muy oscuro (la piscina) en un objeto muy brillante, de modo que, siguiendo el flujo normal de procesamiento, se clasificaría como fondo en lugar de objeto.

Sin embargo, antes de implementar esta solución, es importante identificar el nivel de intensidad específico que corresponde a la piscina y determinar un umbral adecuado para aplicar la binarización inversa.

En resumen, el método actualmente tiene una tasa de efectividad del 80 % en la clasificación de caminos en las muestras analizadas, y se podría mejorar aún más mediante ajustes específicos para abordar casos como la presencia de objetos con niveles de intensidad muy diferentes.

6.4. Operaciones Morfológicas

El objetivo de usar operaciones morfológicas en las imágenes es rellenar hoyos en los objetos o eliminar protuberancias de estos, como también unir elementos que se separan en el proceso de binarización o separar elementos que se unen cuando no debería ser así. En este caso, los dos extremos toman importancia: poder rescatar el camino en su totalidad. En este caso, cada vez que el camino cuenta con una imperfección o tonalidades muy distintas, estas no formaron parte de este en su binarización, como en los siguientes ejemplos.



(a) Muestra n°1 **(b)** Muestra n°3 **(c)** Muestra n°4

Figura 6.4: Ejemplo de fallas en muestras

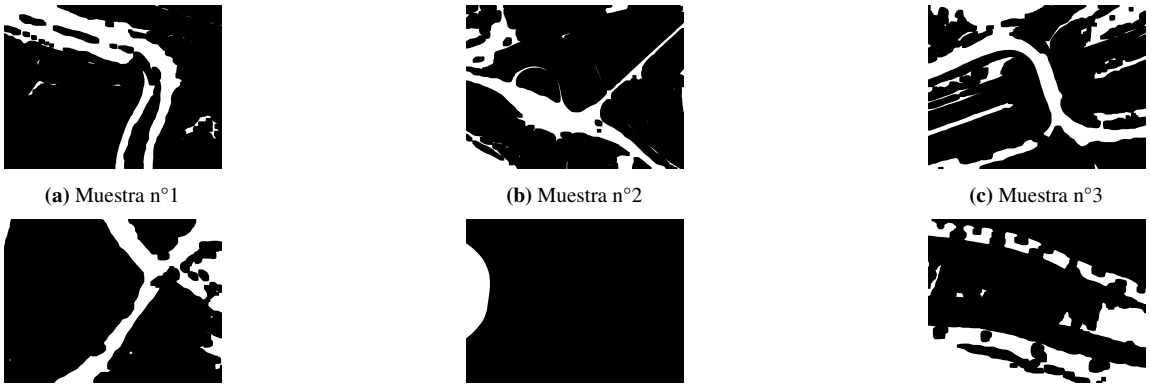
Como también, separar bien los caminos del resto de objetos que puedan estar conectándose a ellos, como las orillas de estos o los cerros en los casos donde el camino se encuentre muy inserto en estos.

Considerando estas dos ideas, se presenta la interrogante de qué método es más idóneo de usar. Desde el estado del arte se tiene que el método de erosión es útil para disminuir los objetos, al igual que la dilatación es para expandirlos. Seguidos de estos, se tiene la apertura, que es útil para eliminar pequeños elementos de una imagen, recuperando la forma original de esta, como también la clausura, que es útil para rellenar hoyos, nuevamente recuperando la forma original de la imagen.

Tanto con el método de dilatación como con el de clausura es posible atacar la primera problemática, pero no así la segunda, ya que al expandir la imagen sin antes clasificar los contornos de esta, filtrando algunos y manteniendo otros, lleva a que al únicamente expandir (Dilatación) como al erosionarse e inmediatamente expandirse (Clausura) en muchos casos, los caminos se fundan con otros objetos de la imagen.

Por otro lado, el método de erosión es capaz de resolver la segunda problemática, sin resolver la primera claramente, pero permitiendo que esta se solucione de forma posterior al aislar y trabajar cada contorno de forma separada y no como un todo como es en este caso. Por lo cual, se procede a aplicar una Erosión de un Kernel de tamaño 31x31.

Con esto, es posible asegurarse de que se está extrayendo únicamente el camino, lo cual es el principal objetivo de esta investigación, sin llevarse otras partes de la imagen con este.



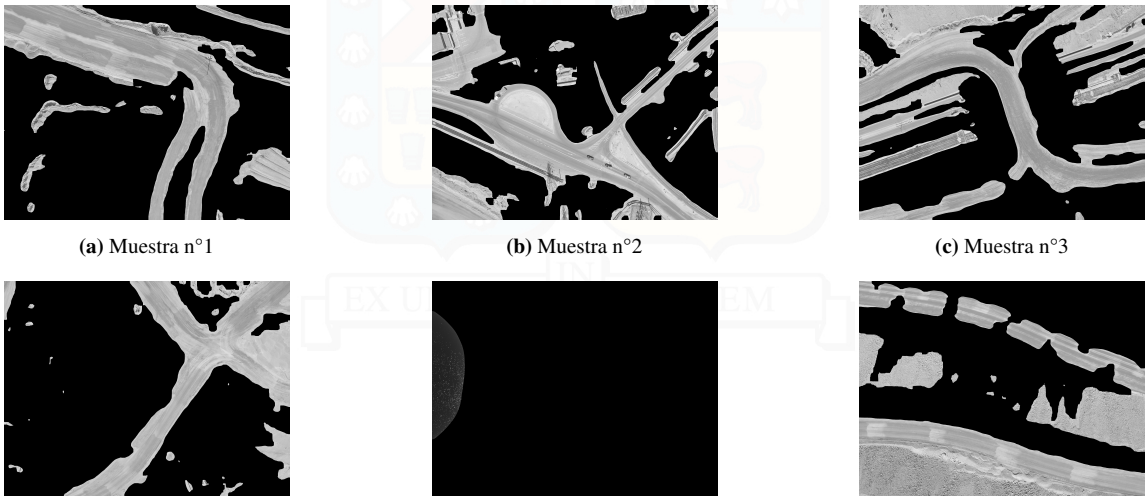
(a) Muestra n°1 **(b)** Muestra n°2 **(c)** Muestra n°3

(d) Muestra n°4 **(e)** Muestra n°5 **(f)** Muestra n°6

Figura 6.5: Muestras con operación de erosión de kernel 31x31

6.5. Segmentación

Analizando las imágenes obtenidas, solo queda evaluar si mediante las imágenes segmentadas obtenidas es posible rescatar los caminos. En este caso, la efectividad de los métodos no depende de la capacidad para reconocer bordes o curvas en general, sino específicamente en la capacidad de identificar curvas cerradas que puedan representar objetos o caminos, para ser específico en este caso. Respecto a esto, se obtienen los siguientes resultados para las seis imágenes de muestra.



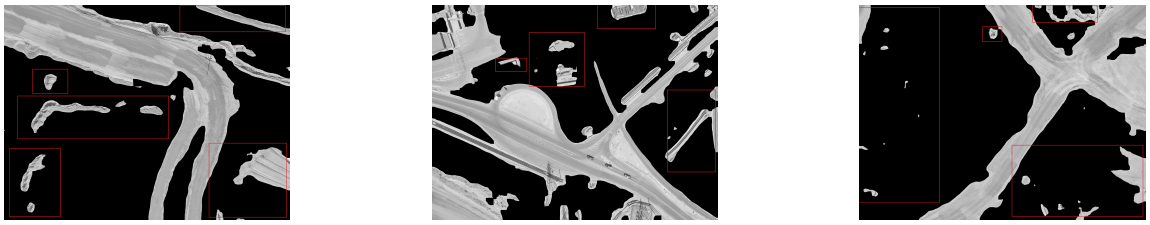
(a) Muestra n°1 **(b)** Muestra n°2 **(c)** Muestra n°3

(d) Muestra n°4 **(e)** Muestra n°5 **(f)** Muestra n°6

Figura 6.6: Diversos casos de operaciones morfológicas sobre las muestras con mayor cantidad de hoyos en la identificación del camino

Mediante estas imágenes se identifica que las operaciones utilizadas, considerando principalmente la binarización mediante el método de Otsu, son capaces de generar una segmentación de los caminos en el flujo propuesto, pero hasta esta instancia no únicamente de estos, sino también de elementos no deseados que deben ser purgados para que la segmentación sea únicamente de los caminos.

Para esto, queda analizar a fondo las figuras obtenidas dentro de cada imagen. Si se analiza por ejemplo las muestras n°1, n°2 y n°4 en las cuales se identifican las siguientes figuras en la imagen.



(a) Muestra n°1 **(b)** Muestra n°2 **(c)** Muestra n°4

Figura 6.7: Resaltado de secciones a eliminar en cada fotografía

6.6. Clasificación y filtrado de los objetos

Como se comentó en el apartado anterior, con la finalidad de separar los casos donde se haya segmentado únicamente el camino o se haya segmentado este y más figuras dentro de la imagen (como es posible evidenciar en algunos de los ejemplos obtenidos y mostrados previamente), se procede a generar cinco datasets con diversos indicadores de cada una de las figuras identificadas dentro de cada imagen de las seis imágenes de muestra presentadas previamente, además de treinta imágenes más elegidas aleatoriamente. Estas imágenes se separan en seis clusters que representan cada uno de los datasets.

Para cada figura de cada imagen de cada cluster, se calculan todos los indicadores asociados a contornos listados en el marco teórico. De estos cálculos se obtiene que en la mayoría de los casos, en primera instancia, no es posible encontrar un patrón que indique si la figura corresponde a un camino o no. Esto se ejemplifica con el Área de cada figura:

Cluster	Camino	Promedio	Desvest.	Máx.	Mín.
1	No	717427	447897	1727038	164304
	Si	2150701	2257524	6591615	215422
2	No	1192553	1500561	6242725	170237
	Si	3464096	1840491	6848476	898029
3	No	1311151	1783937	6542137	224039
	Si	2734263	2417552	7265974	375892
4	No	600355	548863	2208782	155784
	Si	3427503	3309800	9490573	324636
5	No	876374	1059947	3794313	152921
	Si	1683077	1867676	6475099	305756
6	No	657366	495673	1660762	109505
	Si	2350468	2039094	6505022	331404

Tabla 6.1: Área promedio de cada figura en cada cluster

Donde es posible identificar que en la mayoría de los casos, los caminos cuentan con una mayor envergadura que las figuras que no corresponden a caminos. Sin embargo, esto no es suficiente para encontrar un patrón que permita clasificar a través del área de la figura si esta corresponde a un camino o no, como se visualiza en la tabla. En cada uno de los clusters, existen caminos más pequeños que otras figuras identificadas en la imagen que no lo son, valiéndose en la figura de mayor envergadura que no es identificada como camino y el camino de menor envergadura en cada caso.

Por otro lado, se determina que es posible encontrar patrones en base a los siguientes conceptos.

Cluster	Camino	Promedio	Desvest.	Máx.	Mín.
1	No	193	18	211	111
	si	193	7	201	176
2	No	189	16	212	153
	si	188	11	203	175
3	No	186	18	209	134
	si	185	10	201	170
4	No	193	7	201	178
	si	194	8	205	179
5	No	184	15	206	138
	si	186	12	206	166
6	No	187	11	207	167
	si	188	12	210	169

Tabla 6.2: Intensidad promedio de cada figura en cada cluster

Indicador del cual es posible identificar que, considerando cada cluster, la intensidad promedio tiende a representar precisamente la luminosidad general de la imagen más allá del objeto que se esté analizando. Por lo tanto, no es preciso utilizar este indicador para un proceso de umbralización o clasificación de figuras. Sin embargo, por otro lado, se visualiza que para cuatro de los seis casos, la desviación de este promedio es menor en el caso de los caminos.

Cluster	Camino	Promedio	Desvest.	Máx.	Mín.
1	No	29	16	94	11
	si	21	6	32	11
2	No	29	9	44	11
	si	25	6	37	18
3	No	34	8	44	20
	si	24	7	36	15
4	No	23	13	46	7
	si	23	4	35	16
5	No	32	11	49	10
	si	23	6	33	13
6	No	29	13	58	14
	si	24	5	29	12

Tabla 6.3: Desviación estándar de la Intensidad de cada figura en cada cluster

Esto se complementa con el análisis de la desviación de intensidad de cada figura por separado. Se encuentra que en todos los casos, la desviación de la intensidad de los caminos es considerablemente menor que la de otras figuras extraídas. Esto concuerda directamente con la teoría, que indica que uno de los mejores métodos para diferenciar objetos en una imagen es estudiar la varianza de su intensidad, ya que la varianza de los objetos tiende a ser mucho más homogénea que la varianza del fondo. Concluyendo esto, se define como filtro de clasificación que únicamente se consideren figuras que cuenten con una desviación de su intensidad que oscile de la siguiente forma.

$$10[px] \leq \lambda \leq 40[px] \tag{6.1}$$

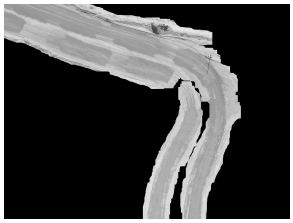
y un promedio de intensidad que se encuentre entre

$$160[px] \leq \mu \leq 210[px] \tag{6.2}$$

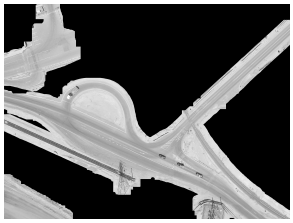
Destacando además que se cuenta con un filtro previo al análisis de cluster, con la finalidad de eliminar todo punto de pequeña envergadura que haya sido segmentado y que visualmente no es capaz de ser considerado como figura.

$$20000[px^2] \leq Area(figura) \tag{6.3}$$

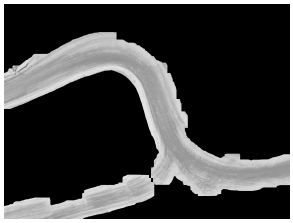
Contando con estos filtros y teniendo en cuenta que se han identificado caminos en la extracción de figuras de las cuarenta imágenes analizadas, como se logra visualizar en las muestra principales.



(a) Muestra n°1



(b) Muestra n°2



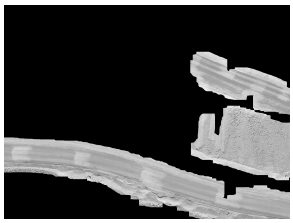
(c) Muestra n°3



(d) Muestra n°4



(e) Muestra n°5



(f) Muestra n°6

Figura 6.8: Resultado final para cada muestra del cluster inicial, cabe destacar que en la muestra n°5 no se reconoce un camino

Repasando todo el flujo se determina la siguiente metodología

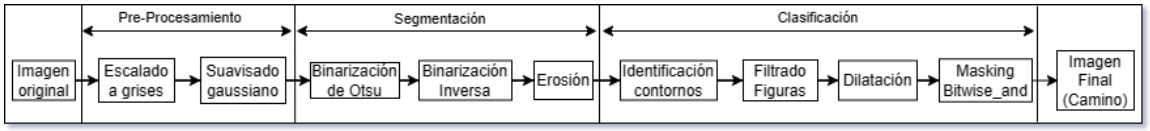


Figura 6.9: Flujo final

La cual es posible aplicar al análisis de las 1900 fotografías presentes en el dataset inicial.

Efectuando la iteración sobre las imágenes, se obtiene un proceso que demora 32,4 minutos en el cual se identificaron caminos en 1537 fotografías. A continuación, se presentan los siguientes indicadores obtenidos.

	Promedio	Desviación	Max.	Min.
Tiempo procesamiento [s]	1,05	0,21	2,42	0,86
Área figura extraida[px^2]	1720017	1493537	9348938	300455
Intensidad Promedio figura extraida [px]	188	9	210	160
Desviación de la intensidad figura extraida [px]	22	6	38	10

Tabla 6.4: Indicadores relevantes respecto a cada fotografía analizada y el camino extraído

7 | Conclusiones

En base a todo lo estudiado, procedemos a concluir respecto al cumplimiento de cada uno de los objetivos planteados.

7.1. Identificar una metodología de procesamiento de imágenes que permita identificar caminos mineros.

Gracias al estudio del estado del arte respecto a la visión artificial, logramos comprender que esta es una herramienta útil en el estudio de imágenes debido a todas las funciones y análisis posibles que esta ciencia ofrece, así como diversas metodologías dependiendo de la tarea que se espera lograr dentro de cada imagen. En este caso, como buscamos identificar cada camino dentro de una imagen, necesitábamos una metodología que pudiera lograr esta función.

Para esto, utilizamos un flujo común de segmentación y clasificación que depende de los procesos de captura, pre-procesamiento, segmentación y clasificación, haciendo uso de la binarización como herramienta de segmentación. Durante este estudio, identificamos que esta metodología propuesta tiene el potencial para ser una herramienta útil en la interpretación de los caminos, y su capacidad y efectividad dependen de una selección adecuada de funciones y algoritmos para cada parte del proceso.

7.2. Desarrollar un modelo de programación en Python que permita aplicar la metodología definida procesando imágenes de minas de rajo abierto.

Para llevar esta metodología a la realidad, se utiliza el lenguaje de programación Python en su versión 3.9. desde el editor Spyder en un computador portátil de 12 gb de Ram utilizables. El algoritmo se divide en tres partes principales:

- **Abrir, cerrar y crear archivos:** En esta primera parte, el desafío era encontrar una forma de abrir la carpeta donde se encuentran las fotografías a analizar, seleccionar cada una de forma automatizada y, al finalizar, guardar la nueva versión de esta en otra carpeta. Para esto, se hace uso de la librería os, la cual permite interactuar con las funcionalidades del sistema operativo. En el código presente en el anexo, las líneas 4 y 5 permiten posicionarse en la carpeta donde se encuentran las imágenes y generar una lista para posteriormente ser trabajadas una por una en el código.
- **Aplicar la metodología en base a visión artificial propuesta:** Para aplicar la metodología definida, se requiere un conjunto de funciones capaces de implementar la teoría estudiada. Para esto, se utiliza la biblioteca OpenCV. De la tabla presentada en el marco teórico, se recurre a las siguientes funciones:
  - **imread:** Para leer la imagen como matriz
  - **Gaussian Blur:** Para aplicar el suavizamiento Gaussiano
  - **threshold(THRESH_OTSU) y threshold(THRESH_BINARY_INV):** Para aplicar la binarización de Otsu e inversa respectivamente.
  - **erode y dilate:** Para erosionar y dilatar cada imagen.
  - **findcontours y drawcontours**: Para encontrar y dibujar contornos respectivamente
  - **bitwise_and**: Para intersectar la imagen original y la mascara con la figura de cada camino para obtener la imagen final
  - **imwrite**: Para guardar la nueva imagen como un archivo en el sistema.

  Además de las funciones de análisis de contornos, como identificar área o perímetro, y el uso de la librería **NumPy** para generar matrices numéricas, esta parte del código puede ser revisada entre las líneas 14 y 62.
- **Rescatar la información obtenida y almacenarla en un archivo:** Por último, mediante el uso de la librería **pandas**, es posible convertir la data recopilada de cada fotografía en una tabla con columnas rotuladas que puede ser almacenada en un archivo Excel, como se indica entre las líneas 66 y 72 del código presentado en los anexos.

7.3. Evaluar el algoritmo mas apto para cada parte del proceso según el estado del arte.

En este primer acercamiento se logra identificar las herramientas que, en su conjunto, logran generar la extracción de los caminos, pero no así los parámetros óptimos dentro de cada una de estas que puedan generar una extracción fina que únicamente identifique lo que realmente corresponde a un camino minero. En los siguientes párrafos se entrega la justificación del uso de cada herramienta:

- Suavizamiento Gaussiano:El suavizado cumple el rol de normalizar la imagen, ya que al ser de gran resolución, la variación entre cada píxel es considerable, generando que sin un filtro los pasos posteriores segmentasen muchas figuras además de las esperadas. En base a esto, se decide hacer uso del suavizado Gaussiano, ya que al considerar una media y una varianza en los valores de su kernel, es posible darle mucho peso a los píxeles centrales de un kernel y poco peso a los más extremos. Lo que, en el caso de los objetos, provoca un resaltado de la parte central de estos y, a medida que el escáner del kernel se aproxima a los extremos de la vecindad, empiezan a tomar menor peso los píxeles fuera de la figura, generando un mejor contraste en los bordes de esta.
- Binarización de Otsu: La binarización se identifica como el elemento más importante de este proceso, ya que en esta se rescata la porción de camino dentro de la fotografía ya sea identificándose como figura o como fondo, dependiendo del comportamiento de la intensidad en la imagen. Gracias al análisis efectuado en el documento, se identifica que la intensidad se define principalmente por dos cosas: los colores en la fotografía y la iluminación que incide en esta. Si previamente se supiese el nivel de intensidad que representan los colores de cada camino, se podría efectuar una binarización parametrizada por un umbral, pero lamentablemente, la iluminación de cada imagen es sumamente variable entre ellas. Por lo cual, si se define un umbral fijo, este se adaptará únicamente a cierta porción de todo el paquete de imágenes a analizar. Para solucionar esto, se trabaja con un método de binarización adaptativa que estudie la variabilidad de esta intensidad en cada imagen y, sobre esta, defina cuál es el umbral ideal para separar la imagen en figura y fondo para cada imagen. Este método es la binarización de Otsu, ya que esta efectúa un análisis de varianza para binarizar cada imagen.
- Binarización Inversa: Posterior a la binarización de Otsu, en cada imagen fue posible identificar de forma general que resaltaba el camino o parte de este, pero no como figura, sino como fondo. Esto no es descabellado, ya que la binarización define lo más oscuro como fondo por un tema de profundidad, en este caso, por el nivel de resolución y luminosidad de la fotografía, además de que el camino en sí es más oscuro que su entorno. Se esperaba que el camino retornase como fondo. Para tratar cada camino como figura, se aplicó una binarización inversa convirtiendo el fondo en figura y la figura en fondo.
- Erosión: Cerrando el proceso de binarización, el desafío es rescatar únicamente lo que sea camino eliminando las orillas de estos u otros puntos en la imagen que hayan sido rescatados también como figura. Para esto, se aplicó una operación de erosión reduciendo cada figura identificada, lo que en general no debiese eliminar los caminos, ya que estos son las figuras de mayor tamaño dentro de la imagen, sino eliminar las pequeñas figuras extraídas o separar los caminos de otros cuerpos.
- Clasificación de Figuras: Tomando la imagen ya con las figuras erosionadas dentro de esta, se procedió a escanearlas en busca de figuras que se puedan definir en base a contornos y, sobre estas, intersectarlas con la imagen original para obtener únicamente las propiedades de esa porción de fotografía para analizarla. Identificando que propiedades puedan ser únicas de las figuras correspondientes a los caminos y cuáles no, se obtuvo en este primer acercamiento que mediante una combinación entre el área de las figuras, su intensidad promedio y la desviación estándar de esta intensidad, en general, es posible clasificar las figuras que representen un camino.
- Dilatación: Teniendo las figuras clasificadas, se procedió dilatar para recuperar partes de estas que se hayan perdido en el proceso de erosión y así contar con cada camino en su totalidad.
- Masking y Bitwise: Por último, se generó una máscara que contiene los caminos identificados en cada imagen para intersectarlos con la imagen original y así obtener el resultado final.

7.4. Obtener un dataset con cada camino minero mediante la aplicación del modelo desarrollado.

Gracias a las características del código generado en base a la metodología propuesta y analizada, basta con ejecutarlo una vez para analizar cada fotografía presente en el dataset inicial. Luego de aplicar todas las transformaciones a cada imagen, se guardan en un dataset final que contiene todos los indicadores propuestos para cada figura extraída. Esto proporciona información relevante para identificar patrones futuros considerando una mayor cantidad de datos. También se calcula el tiempo de procesamiento de cada fotografía para determinar si este método realmente permite optimizar el proceso actual utilizado para la interpretación geométrica de los caminos.

El modelo iterativo fue desarrollado en una computadora utilizando el lenguaje de programación Python y las bibliotecas: OpenCV, os, math, numpy, matplotlib y pandas.

Analizando los indicadores obtenidos se visualiza la similitud del comportamiento del promedio y desviación de la intensidad de cada camino en el dataset total y cada cluster estudiado, como también la determinación de que en promedio el modelo demora 1[s] en obtener el camino de la imagen y, por lo menos en este caso, a lo más 2,5[s], concluyéndose que, perfeccionando la elección de parámetros dentro de cada parte del modelo, este cuenta con el potencial y las capacidades para ser un método útil y eficiente en la interpretación del diseño geométrico de caminos mineros para el futuro análisis del diseño funcional de estos.

8 | Trabajos Futuros

8.1. Refinar parámetros algoritmo

El principal objetivo de este trabajo es desarrollar una metodología eficaz y eficiente para identificar y extraer caminos mineros de imágenes. Los resultados obtenidos muestran que es posible crear un flujo de trabajo que puede extraer caminos en un tiempo promedio de X segundos por imagen, lo que lo convierte en un método eficiente. Sin embargo, en una primera etapa, no es completamente efectivo, ya que no logra extraer caminos en todas las fotografías como debería, y a veces extrae elementos adicionales además de los caminos.

Esta limitación se debe principalmente a la gran cantidad de parámetros en el flujo de trabajo, como el tamaño del kernel en las etapas de suavizado, erosión y dilatación, así como los filtros de contorno para identificar la figura correcta en cada caso.

Se propone continuar trabajando en la iteración y análisis, centrándose especialmente en la etapa de suavizado, que tiene un gran impacto en la segmentación de la imagen al normalizar las figuras dentro de ellas. Además, se debe investigar y encontrar la combinación precisa de tamaños de kernel en cada paso en lugar de utilizar valores arbitrarios basados en iteraciones.

En resumen, se reconoce que la metodología tiene potencial pero requiere una optimización significativa para mejorar su efectividad y poder extraer caminos de manera más precisa en una variedad de imágenes.

8.2. Análisis de datos obtenidos

Además, se sugiere llevar a cabo un análisis exhaustivo de todos los indicadores calculados para cada figura, con el objetivo de identificar patrones y filtros más precisos que puedan mejorar la metodología actual. Este análisis puede proporcionar una comprensión más profunda de las características de los caminos mineros en las imágenes y ayudar a afinar los parámetros y algoritmos utilizados en el flujo de trabajo.

El estudio de patrones y la búsqueda de características distintivas en las figuras identificadas son pasos cruciales para perfeccionar la metodología y hacer que sea más efectiva y robusta en la identificación de caminos mineros en diversas condiciones y escenarios.

8.3. Uso del dataset obtenido en un algoritmo de Redes Neuronales

Por último, es importante reconocer la utilidad del aprendizaje supervisado mediante redes neuronales en este contexto. Este enfoque de aprendizaje, que se basa en entrenar modelos para reproducir resultados deseados a partir de datos etiquetados, puede ser extremadamente valioso para tareas de regresión y clasificación, como la identificación de caminos mineros en imágenes. Sin embargo, es fundamental tener datos etiquetados que indiquen claramente cuáles son los datos de entrada y cuáles son los datos de salida esperados.

En el caso de este estudio, si se desea desarrollar un modelo de redes neuronales para identificar caminos mineros en imágenes de vuelos de drones sobre la mina, sería necesario contar con un amplio conjunto de imágenes de vuelos de drones como datos de entrada y, lo que es más crucial, un conjunto igualmente grande de imágenes con los caminos ya identificados y etiquetados como datos de salida. El trabajo realizado en esta memoria puede servir como punto de partida para la generación de un conjunto de datos etiquetados que luego podrían utilizarse para entrenar un modelo de redes neuronales.

El uso de redes neuronales podría representar una mejora significativa en la eficiencia y precisión de la identificación de caminos mineros en imágenes, y es un paso natural a seguir en el desarrollo de esta metodología.

9 | Referencias

1. Consejo Minero. (2020). La Industria Minera 2011 - 2020. Consejo Minero.

2. Consejo Minero. (2023). Cifras actualizadas de la minería. Consejo Minero.

3. Reporte Minero. (2019, 5 junio). Costo operacional de las grandes empresas productoras de cobre se mantuvo estable el 2018. Reporte Minero | El portal de minería en Chile.

4. Reporte Minero. (2018, 10 enero). OCDE alerta sobre baja productividad de la minería en Chile. Reporte Minero | El portal de minería en Chile.

5. Equipo Minero. (2009, septiembre). Consideraciones para el Diseño de Caminos de Transporte Equipo Minero.

6. Tannant, D. & Regensburg, B. (2001). Guidelines for mine haul road design.
7. Soofastaei, Ali & Aminossadati, Saiied & Kizil, M. & Knights, P.. (2015). The Effect of Rolling Resistance on Fuel Consumption and Greenhouse Gas Emissions by Haul Trucks in Surface Mines. Tribology International
8. Roger Thompson, (2019). Where's the Value? Evaluating the costs of haul road maintenance and rebuilds
9. Thompson, Roger & Peroni, Rodrigo & T., Visser. (2019). Mining Haul Roads: Theory and Practice.
10. Agudelo J. (2002) Diseño geométrico de vías Ajustado al Manual Colombiano.
11. Hugo D. (2005) Haul road defect identification and condition assessment using measured truck response
12. Ordieres-Meré, Joaquín & Limas, Manuel & Ascacibar, Francisco Javier & Alba-Elías, Fernando & González-Marcos, Ana & Pernía-Espinoza, Alpha & Vergara, Eliseo. (2006). Técnicas y algoritmos básicos de visión artificial
13. Murzova, A. (2021). Otsu's thresholding technique | LearnOpenCV. LearnOpenCV – Learn OpenCV, PyTorch, Keras, Tensorflow with examples and tutorials.

14. San-ho yoon. (2014). Algoritmos de procesamiento de imagen aplicados a la detección de figuras geometricas y sus propiedades espaciales.
15. Gonzalez, Rafael & Faisal, Zahraa. (2019). Digital Image Processing Second Edition.
16. Jaramillo Gabriel. (2009). redes neuronales aplicadas al análisis de imágenes para el desarrollo de un prototipo de un sistema de seguridad.

17. LUKAC, RASTISLAV & Plataniotis, Konstantinos. (2006). Single-Sensor Camera Image Processing. 10.1201/9781420009781.ch16.

18. Cuartas, J. (2021, 29 diciembre). El concepto de la convolución en gráficos, para comprender las convolutional neural networks (CNN) o redes convolucionadas. Medium.

19. Querejeta Simbeni, Pedro. (2015). procesamiento digital de imágenes

20. OpenCV. (2020). About. Recuperado 15 de octubre de 2022, de https://opencv.org/about//

21. Castellano Carvajal, Diego & Niño Rondón, Carlos & Medina Delgado, Byron & Guevara, Dinael & Haddad, Miguel. (2023). INCIDENCIA DE LOS FILTROS DE SUAVIZADO EN LAS HERRAMIENTAS DE DIAGNÓSTICO ASISTIDO POR COMPUTADOR. REVISTA COLOMBIANA DE TECNOLOGIAS DE AVANZADA (RCTA). 1. 59-65. 10.24054/rcta.v1i39.1376.

22. Triana-Guzman, Nayid & Jaramillo, Andrés & Gutiérrez, R. & Rodríguez, César. (2016). Técnicas de umbralización para el procesamiento digital de imágenes de GEM-Foils. Scientia et technica.

23. Huamaní Navarrete, Pedro. (2015). umbralización múltiple utilizando el método de otsu para reconocer la luz roja en semáforos.

24. Maldonado, Luis Carlos; Peña, César Augusto; Gualdrón, Oscar (2012). Identificación Automática de Cilindros de Almacenamiento de Gas Utilizando Redes Neuronales Tipo Hopfield

25. Jesús. (2021). Introducción a las transformaciones morfológicas en OpenCV. DataSmarts.

26. OpenCV: Contours: Getting started. (s. f.).

27. OpenCV: Contour features. (s. f.).

28. OpenCV: Contour Properties. (s. f.).

29. Som, G. (s. f.). Operar a nivel de bits (operaciones lógicas a nivel de bit).

30. GeeksforGeeks. (2022, 12 septiembre). Python | Grayscaling of Images using OpenCV.

A | ANEXOS

Listing A.1: Código en Python para procesamiento de imágenes

```
import time,os,cv2 as cv,numpy as np, pandas as pd # importar librerias a utilizar
carpeta="C:/Users/sebos/Desktop/Fotos Final Final/caminos_raw" #carpeta donde se
    almacenan las imagenes a analizar#
carpeta_final="C:/Users/sebos/Desktop/Datos_final/CONTORNOS6/" #carpeta donde se
    almacenan los caminos extraidos
os.chdir(carpeta) #posicionarse en la carpeta a analizar
L = os.listdir('.') #Lista con los nombres de cada archivo dentro de la carpeta
info = {} #Diccionario donde se almacena la informacion de cada figura de cada fotografia
l = 0 #Contador de cada iteracion del algoritmo
j = 0 #Contador de cada foto analizada
m= 0 #Contador de cada figura obtenida en cada fotografia
for i in range(len(L)): #Ciclo para iterar tantas veces como la cantidad de fotografias a
     analizar
    j+=1
    inicio=time.time() #identificar tiempo del inicio de procesamiento de la fotografia
    i = L[i] #guardar el nombre de la fotografia
    a = cv.imread(i,cv.IMREAD_GRAYSCALE) #Leer la fotografia en escala de grises
    b =  a #Crear una copia de la fotografia
    a = cv.GaussianBlur(a, (31, 31), 10) #Aplicar suavisamiento Gaussiano de media 31 y
        desviacion 10
    _,a = cv.threshold(a, 0, 255, cv.THRESH_BINARY + cv.THRESH_OTSU) #Aplicar
        binarizacion de Otsu
    _,a = cv.threshold(a, 125, 255, cv.THRESH_BINARY_INV) #Aplicar binarizacion inversa
        para intercambiar forma por fondo
    a= cv.erode(a, np.ones((31,31)), iterations=1) #Aplicar Erosion con kernel de tam[U+FFFD]
         31
    contours, _ = cv.findContours(a, cv.RETR_EXTERNAL, cv.CHAIN_APPROX_SIMPLE) #
        identificar todos los contornos posibles en la imagen
    CONTORNOS =[] #Lista donde se almacenan los contornos filtrados
    if len(contours) != 0: #Condicion para analizar unicamente imagenes donde se
        identifiquen contornos
        m = 0
        for cnt in reversed(contours): #Ciclo para iterar sobre todos los contornos
            identificados en la imagen
          if cv.contourArea(cnt) >= 300000: #Condicion para tomar unicamente las figuras
               que tengan un area mayor a 300000px cuadrados
            contour_image = np.zeros(a.shape[:2], np.uint8) #Crear imagen vacia donde ir
                generando una mascara para obtener unicamente el contorno filtrado.
            cv.drawContours(contour_image, [cnt], -1, 255, -1) #Dibujar contorno en la
                imagen
            contour_image = cv.dilate(contour_image,np.ones((311,311)), iterations=1) #
                Dilatar el contorno
            MEAN,DESVT = cv.meanStdDev(b,mask = contour_image) #Intersectar la imagen
                original con la mascara para obtener la media de intensidad y la
                desviacion estandar de la figura extraida
            if 160 <= MEAN[0][0] <= 210 and 10 <= DESVT[0][0] <= 40: # Condicion para
                filtrar una figura como camino o no
                contours,_ = cv.findContours(contour_image, cv.RETR_EXTERNAL, cv.
                    CHAIN_APPROX_SIMPLE) #Extraer contorno de la figura dilatada
                m+=1
```

```
                l+=1
                CONTORNOS.append(contours[0]) #Agregar contorno a la mascara final sobre
                    la imagen inicial
                #Determinar indicadores para preservar informacion de cada figura
                A = int(cv.contourArea(contours[0])) #Area
                P = int(cv.arcLength(contours[0],True)) # Perimetro
                M= cv.moments(contours[0]) #Momentos
                Cx = int(M['m10']/M['m00']) # Coordenada X del momento
                Cy = int(M['m01']/M['m00']) # Coordenada Y del momento
                (Px,Py),(W,H),O = cv.minAreaRect(contours[0]) # W= ancho, H= altura,
                    Angulo = O
                ASP = float(W)/H # Ratio de aspecto
                EXT = float(A)/(W*H) # Ratio de extension
                SOL = float(A)/ cv.contourArea(cv.convexHull(contours[0])) # Ratio de
                    solides
                MIN_PIX,MAX_PIX,MIN_LOC,MAX_LOC = cv.minMaxLoc(b,mask = contour_image) #
                    Pixel minimo y maximo y sus coordenadas
                MEAN_PIX,DESVT_PIX = cv.meanStdDev(b,mask = contour_image) #Nuevamente
                    Media y promedio de intensidad
                if  m == 1:
                    fin=time.time()
                    T = fin-inicio
                else:
                    fin2 = time.time()
                    T = fin2-fin
                    fin = time.time()
                info[l] = [j,i,1,m,T,A,P,Cx,Cy,Px,Py,W,H,O,ASP,EXT,SOL,MIN_PIX,MAX_PIX,
                    MIN_LOC,MAX_LOC,MEAN_PIX[0][0],DESVT_PIX[0][0]] #Almacenar
                    informacion de cada figura en el diccionario propuesto
        if len(CONTORNOS) == 0:#Condicion si no se identifica ningun contorno que cumpla
            las condiciones para rotular la fotografia como imagen sin camino extraido
          l+=1
          info[l] = [j,i,0,0,T,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
        else: #Caso en que si se obtienen figuras
            contour_image = np.zeros(a.shape[:2], np.uint8) #Se genera una mascara para
                todas las figuras extraidas de la imagen
            cv.drawContours(contour_image, CONTORNOS, -1, 255, -1) #Se dibujan las
                figuras en la mascara
            img_final = cv.bitwise_and(b,b,mask=contour_image) #Intersectar mascara con
                la imagen final
            cv.imwrite(carpeta_final+'Foto_'+str(i)+'_nro_'+str(j)+'.jpg',img_final) #
                Guardar la imagen final en la carpeta de salida con su nombre y su nro
    else: #Caso en que no se identifican contornos en la primera etapa
        l+=1
        info[l] = [j,i,0,0,T,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
df = pd.DataFrame.from_dict(info, orient='index',
columns=['FOTO','NOMBRE','CAMINO','CONTORNO','TIEMPO PROCESO', 'AREA', 'PERIMETRO',
         'MOMENTOx','MOMENTOy','RECx','RECy','ANCHO','ALTURA','INCLINACION',
         'RATIO_ASPECTO','RATIO_EXTENCIO','RATIO_SOLIDEZ','MINPIX','MAXPIX',
         'POSminpix','POSmaxpix','PIXPROM','PIXDESVT']) #Crear un DataFrame a partir del
             diccionario info
nombre_archivo = 'C:/Users/sebos/Desktop/Datos_final/pruebaa.xlsx' #Definir un nuevo
    archivo Excel
df.to_excel(nombre_archivo, index_label='ITERACION') #Guardar el DataFrame en un archivo
    Excel
```