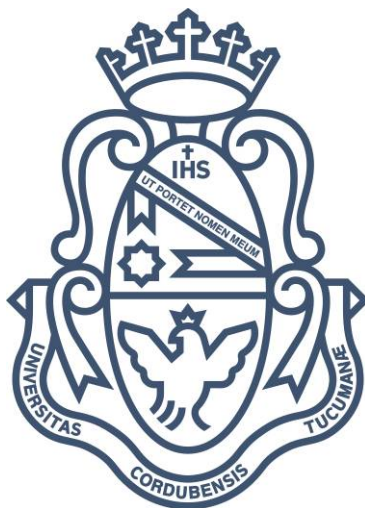


FACULTAD DE MATEMÁTICA, ASTRONOMÍA, FÍSICA Y  
COMPUTACIÓN

UNIVERSIDAD NACIONAL DE CÓRDOBA



## Detección de líneas de cultivo no supervisada a partir de imágenes tomadas por un vehículo aéreo no tripulado

TESIS PARA OBTENER EL TÍTULO DE  
LICENCIADO EN CIENCIAS DE LA COMPUTACIÓN

AUTOR: IGNACIO JOSÉ BALDONCINI

DIRECTORA: MARÍA VICTORIA MARINELLI

PROFESOR REPRESENTANTE: LUIS BIEDMA

CÓRDOBA, ARGENTINA 2023



Esta obra está bajo una [Licencia Creative Commons Atribución - No Comercial 4.0 Internacional](https://creativecommons.org/licenses/by-nc/4.0/).



# Agradecimientos

Quiero agradecer a mi familia y amigos por acompañarme en estos años de cursado, a las personas que me asistieron y dedicaron su propio tiempo en ayudarme en el desarrollo de esta tesis, y en particular a mis amigos y compañeros, Esteban y Nazareno, ya que no hubiera sido posible sin ellos.



# Resumen

En esta tesis se propone una metodología de procesamiento de imágenes de alta resolución espacial tomadas por un vehículo aéreo no tripulado (VANT) a fin de detectar líneas de cultivo, aplicado a campos hortícolas periurbanos. El procesamiento a dichas imágenes es de carácter no supervisado, que permite la generalización del procedimiento desarrollado y la fácil adaptabilidad a otros conjuntos de datos de características similares. Se describen la fuente y propiedades de los datos utilizados, además de los algoritmos y conceptos aplicados para el procesamiento de los mismos hasta llegar al producto final, y se realiza un análisis de los resultados en dónde se detallan las principales fallas o fortalezas de las técnicas empleadas. Finalmente se llega a una conclusión sobre lo explorado durante este trabajo y se dejan asentados posibles líneas de trabajo futuro.



# Índice general

<b>1. Introducción y motivación</b>	<b>1</b>
1.1. Introducción . . . . .	1
1.2. Motivación . . . . .	1
1.3. Esquema de la tesis . . . . .	2
<b>2. Objetivo y datos</b>	<b>5</b>
2.1. Objetivo . . . . .	5
2.2. Datos . . . . .	5
<b>3. Metodología</b>	<b>9</b>
3.1. Herramientas . . . . .	9
3.2. Preparación . . . . .	9
3.3. Segmentación . . . . .	11
3.3.1. Máscara de bordes . . . . .	11
3.3.2. Matriz de distancias . . . . .	14
3.3.3. Algoritmo de Watershed . . . . .	15
3.3.4. Regiones finales . . . . .	15
3.4. Rectificación y recorte . . . . .	15
3.4.1. Rotación . . . . .	15
3.4.2. Recorte . . . . .	16
3.5. Detección de líneas de cultivo . . . . .	16
<b>4. Experimentos</b>	<b>21</b>
4.1. Métricas . . . . .	21
4.2. Búsqueda de parámetros . . . . .	22
<b>5. Análisis de resultados</b>	<b>23</b>
5.1. Segmentación . . . . .	23
5.2. Rotación y recorte . . . . .	23
5.3. Detección de líneas de cultivo . . . . .	26
<b>6. Conclusión y Trabajo Futuro</b>	<b>31</b>
6.1. Conclusión . . . . .	31
6.2. Trabajo Futuro . . . . .	31





# Capítulo 1

## Introducción y motivación

### 1.1. Introducción

En esta tesis, se exploraron diferentes técnicas y métodos de procesamiento de imágenes con el objetivo de obtener, a partir de una imagen multiespectral de un campo tomada mediante un vehículo aéreo no tripulado (VANT), las diferentes regiones o segmentos relevantes dentro de la propia imagen, determinar si es un sector cultivado o no, y si lo es, detectar y marcar líneas de cultivo (surcos) de cada región. Gracias al rango de longitud de onda que la cámara es capaz de captar, se tuvo acceso a bandas que permitieron el uso de índices (álgebras de bandas); en particular el índice de vegetación de diferencia normalizada, más comúnmente llamado índice verde o NDVI, con el cuál se lograron resultados consistentes. El proceso consiste de 3 etapas, segmentación, rectificación y detección, y tiene las ventajas de ser en su mayoría no supervisado, por lo que no requiere de conjuntos de datos, los cuales son difíciles de conseguir, y se podrían modificar sus parámetros para adaptarlo a otras regiones con otros tipos de vegetaciones, o incluso a imágenes tomadas con otras cámaras.

### 1.2. Motivación

Los cultivos hortícolas desempeñan un papel importante en la mejora de la productividad de la tierra, la generación de empleo, la mejora de las condiciones económicas de los agricultores y empresarios, y la seguridad alimentaria y nutricional de la población [1, 8]. Para una mejor gestión de los cultivos existentes y para aumentar la superficie dedicada a cultivos hortícolas, se necesita una base de datos actualizada y precisa para la planificación y la toma de decisiones sistemáticas [9]. La creación y mantenimiento de esta base de datos requiere muchos recursos, tanto económicos como humanos, la automatización de estos procesos puede ser de gran ayuda, más aún para el sector hortícola, el cual no cuenta con el mismo nivel de acceso que el sector agrícola de cultivos masivos. Estas limitaciones dan lugar a la posibilidad de proponer mejoras y probar técnicas de diversa índole, entre ellas, el aprendizaje automático [11].

La geociencia y la teledetección (GRS) son herramientas avanzadas que ayudan a recopilar y actualizar información para desarrollar ciencia para los planes de gestión y manejo. Recientemente, la mejora en términos de resoluciones espaciales, temporales y radiométricas

tricas; y la accesibilidad de los sensores aerotransportados o de los vehículos aéreos no tripulados (VANT) ha cambiado el desempeño y el enfoque de la teledetección hortícola. Debido a que estos datos e imágenes son digitales, pueden ser fácilmente cuantificados y manipulados desarrollando procesos computacionales avanzados y software libres.

La teledetección de la actividad hortícola desde plataformas remotas es un desafío para los investigadores debido a las características fenológicas, diversidad, tamaño de las unidades productivas, ubicación, y manejo agrícola de la horticultura. Algunas de sus características particulares son el contexto y la fragmentación [12]. Además de eso, el tamaño o escala de los lotes, su disposición espacial y la diversidad de cultivos que componen los huertos, interfieren en la recepción espectral del sensor y hacen aún más importante que éste tenga una alta resolución espacial, espectral y radiométrica. Además, se requiere el desarrollo y la accesibilidad al procesamiento de esa complejidad para la identificación de cobertura. Finalmente, y aún más complejo, es tratar de cuantificar la producción de cada cultivo ya que interfiere el tipo de manejo y la interacción o efecto suelo.

En este trabajo se procesan datos de sensado remoto de alta resolución espacial a los fines de responder a la pregunta de investigación ¿Qué y Cuánto se produce? Se usan datos de un vehículo aéreo no tripulado (VANT) con una cámara multispectral de resolución submétrica (Parrot Sequoia+), también llamadas VHRI por su nombre en inglés *very high resolution images*. El procesamiento de las imágenes adquiridas representa un nuevo desafío técnico-científico que será abordado en este trabajo. El aporte de detección de líneas de cultivos específicos que se encuentran rodeadas de otros cultivos hortícolas, no se registra en la bibliografía. Actualizando la búsqueda de [11], concerniente a procesamiento GEOBIA para imágenes VHRI (capturadas por un sensor en un VANT) con fines hortícolas<sup>1</sup>, *Science Direct* arrojó 23 resultados, 15 de los cuales fueron publicados después de 2020, indicando un futuro promisorio de esta línea de trabajo.

El avance en segmentación y detección de líneas de cultivo y plantas individuales de especies hortícolas en contextos de diversidad de variedades y manejo, junto con modelos de estimación de biomasa y la detección de stress de los cultivos, son las bases para la construcción de mapas de rendimiento hortícola y del servicio de abastecimiento de alimentos frescos [13]. Una identificación precisa y de detalle que indique cultivos hortícolas en producción, adquiere carácter de importancia nacional y se incorpora al proyecto marco de estimación de producción y abastecimiento hortícola como lo es el proyecto Monitoreo Espacial de la Producción Hortícola en Periurbanos de la República Argentina (MEPHPRA)[17] del Programa PROSAT II para el Desarrollo de Tecnologías Satelitales de CONAE[2].

### 1.3. Esquema de la tesis

Esta tesis consta de la siguiente estructura, en el Capítulo 2, se introduce al objetivo y resultado final esperado de este trabajo. En el Capítulo 3, se describen los datos utilizados para el desarrollo y los instrumentos que fueron usados para la obtención de los mismos. En el Capítulo 4, se describe paso a paso el procedimiento que atraviesan los datos hasta

---

<sup>1</sup>Búsqueda: (UAV OR DRONE) AND (GEOBIA OR OBIA) AND (horticultural OR horticulture OR vegetables)

llegar al producto final, y experimentos realizados para su optimización. En el Capítulo 5 se exploran los resultados de aplicar el mismo procedimiento sobre un conjunto de datos distinto del utilizado para el desarrollo. Finalmente, en el capítulo 6 se abordan las conclusiones finales sobre el proyecto y sus resultados, y se sientan algunas bases para posibles líneas de trabajo futuro.



# Capítulo 2

## Objetivo y datos

### 2.1. Objetivo

El objetivo de esta tesis es generar una metodología escalable y robusta de detección de líneas de cultivo, en base a imágenes de alta resolución espacial. Dichas imágenes pueden ser tomadas por un sensor sobre un VANT o sobre plataformas satelitales, que posean sensibilidad a la respuesta radiométrica de la vegetación (i.e. NDVI). Escalable a modo de poder ser usado en otras condiciones, y robusto, no dependiendo de un conjunto de datos particular[4, 23, 26]. Este resultado aporta la base para diversas aplicaciones en el sector agropecuario, algunas de ellas, son la estimación de producción o rendimiento de cierta región o lote, así como el seguimiento del ciclo de cultivo.

### 2.2. Datos

Al momento de trabajar con imágenes, ya sean satelitales, aéreas o de cualquier tipo, se debe tener en cuenta cómo son interpretadas por las computadoras, a diferencia los seres humanos. El espectro de luz visible es el segmento del espectro electromagnético que los ojos humanos pueden percibir, a este rango de longitudes de onda se le llama “luz visible”. Típicamente, el ojo humano puede detectar longitudes de onda desde 380 a 700 nanómetros. Cuando un rayo de luz impacta con algún objeto, este absorbe ciertos rangos de longitud de onda y rechaza otros, dependiendo de estos últimos, es el color del objeto que es percibido por las personas, sin embargo, es posible que el espectro rechazado esté fuera del rango visible humano, pero no significa que no pueda ser captado por una cámara. Es por eso que existen ciertas cámaras especiales capaces de ver estos rangos no visibles y adquirir información a la cual de otra forma no se tendría acceso, que podría ser útil para resaltar o identificar cierto tipo de objetos, como por ejemplo, la vegetación.

Para el desarrollo de esta tesis se trabajó con imágenes tomadas por un vehículo aéreo no tripulado (VANT) con una cámara de 5 sensores, infrarrojo cercano, borde rojo, rojo y verde, más un sensor RGB. Con ello puede capturar imágenes multispectrales y RGB en un solo vuelo en forma simultánea.

El VANT utilizado es un eBee SQ, que es una plataforma de vuelo de ala fija creado

para portar el sensor multiespectral Parrot Sequoia +. La plataforma mide 110cm y pesa 1.1kg. La propulsión es una hélice eléctrica, que activa un motor sin escobillas. Tiene un radio de enlace de 3 km y las alas se desmontan para su transporte. La velocidad de crucero nominal es de 40 a 90 km/h, presenta una resistencia al viento de hasta 45 km/h, con una autonomía de vuelo de 55 minutos, lo que permite cubrir más de 200 hectáreas en un único vuelo, a una altura superior a los 120 metros. Realiza multitud de actividades en vuelo, basadas en un esquema de decisiones e inteligencia artificial (IA) a bordo, analiza los datos mediante la unidad de medición inercial y el GPS de a bordo, que optimiza todos los aspectos del vuelo del eBee SQ.

La cámara Parrot Sequoia+ consta de cuatro sensores: infrarrojo cercano, borde rojo, rojo y verde), más un sensor RGB. Con ello puede capturar imágenes multiespectrales y RGB en un solo vuelo en forma simultánea. Cada sensor monocromático tiene una resolución de 1,2 megapíxel (Mpx) en banda estrecha y sincronizada. Cada imagen multiespectral tiene  $1280 \times 960$  píxeles. Además, la cámara RGB captura fotos de 16 Mpx con una definición de  $4608 \times 3456$  píxeles. Un resumen de sus características radiométricas se presenta en la tabla a continuación.

Banda	Verde	Rojo	Borde Rojo	Infrarrojo Cercano
Longitud de onda	550nm	660nm	735nm	790nm
Ancho de banda	40nm	40nm	10nm	40nm
Definición	1,2 Mpx	1,2 Mpx	1,2 Mpx	1,2 Mpx



Figura 2.1: La escena tomada por el VANT, ubicada en Córdoba capital, el 9/11/2019. En rojo, el sector utilizado durante la etapa de desarrollo



Figura 2.2: RGB de la porción de la escena utilizada para el desarrollo



# Capítulo 3

## Metodología

El proceso completo puede dividirse en 5 etapas claras. **Preparación**, que corresponde a cargar los archivos Python y aplicar ciertas normalizaciones y operaciones para mejorar su calidad en preparación del resto del proceso. **Segmentación** para dividir nuestra imagen en regiones morfológicas más pequeñas para facilitar el trabajo con estas más adelante. **Rectificación y recorte**, donde se rota cada una de las regiones anteriores para que los surcos (i.e. espacio no productivo entre líneas) de los cultivos queden alineados con el eje horizontal de la imagen y además se realiza un recorte para eliminar geometrías irregulares y mantener solo la parte más relevante de la región. **Clasificación** para determinar cuáles regiones se consideran de cultivo y cuáles no, para solo aplicar el siguiente paso a aquellas que poseen cultivo en pie (i.e. pertinentes). Finalmente, **Detección**, en donde se utiliza principalmente el índice NDVI para identificar las líneas de cultivo y generar una máscara con los surcos resultantes de cada región.

### 3.1. Herramientas

Se eligió utilizar el lenguaje Python por múltiples razones, una de ellas es la facilidad y rapidez que otorga al momento de programar y realizar experimentos, dado que gran parte del desarrollo fue del estilo prueba y error, se priorizó la velocidad de implementación por sobre performance en términos de tiempo de computo y uso de memoria. De todas formas, la principal razón fue el gran ecosistema de librerías con las que cuenta Python para facilitar todo tipo de tareas, tanto genéricas de programación y manejo de datos, como específicas del procesamiento de imágenes. Una de estas librerías, y la más importante para el desarrollo de este proyecto, es **Scikit-Image**[21], una extensa colección de los algoritmos más utilizados y recurrentes para el procesamiento de imágenes.

### 3.2. Preparación

Para poder trabajar en procesamiento de imágenes en Python, es necesario llevarlas a un formato que el lenguaje de programación pueda entender, para ello es importante conocer como funcionan las imágenes digitales. Toda imagen que vemos en una pantalla tiene una resolución  $M \times N$ , que puede interpretarse como una matriz, en donde cada elemento es un píxel. A su vez, cada píxel es un vector de 3 coordenadas, que representan

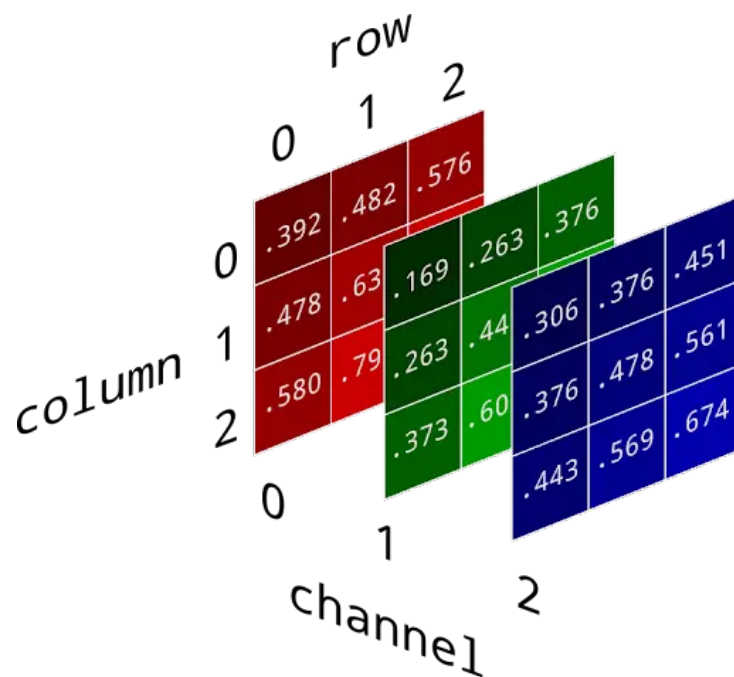


Figura 3.1: Ilustración de la matriz de una imagen 3 x 3 x 3

la intensidad de los colores rojo, verde y azul respectivamente. Si bien para visualizar la imagen, solo se tienen en cuenta 3 canales, el archivo de la imagen puede contener aún más, por lo que podría interpretarse como una matriz  $M \times N \times B$ , donde  $M$  y  $N$  representan la cantidad de píxeles verticales y horizontales, y  $B$  representa la cantidad de bandas, es decir, la cantidad de coordenadas que poseen los vectores de cada píxel. Además, podemos visualizar una imagen usando canales, también llamados bandas, que pueden no corresponder a los típicos colores RGB, lo que nos permite ver características que nuestro ojo humano es incapaz de percibir, como puede ser el espectro infrarrojo o ultravioleta.

En Python, se quiere acceder y trabajar con los datos crudos de la imagen, o sea la matriz  $M \times N \times B$  de la que se habló previamente, para ello, se utilizaron 2 librerías, **Numpy**[16] y **Rasterio**[18]. Numpy es una librería de uso común en el procesamiento de datos con python que otorga una interfaz simple y ágil para el manejo de arreglos y matrices, además de contar con funciones básicas para operar sobre ellas como sumas y multiplicaciones, hasta transposición y pivoteo. Otra ventaja de utilizar numpy es que tras bambalinas está implementado en C, por lo que obtiene una ventaja de velocidad y uso de memoria por sobre Python nativo. Rasterio, por otra parte, es una librería específicamente dedicada al procesamiento de imágenes geoespaciales, y si bien posee varias funciones útiles, la única utilizada en este trabajo fue la de llevar los datos de su formato original a un arreglo de Numpy con las dimensiones correspondientes ( $M \times N \times B$ ).

El primer paso en la cadena de procesamiento es el curado o preparación de los datos, originalmente, la imagen tiene 4 bandas, las cuales pueden tener cualquier rango de valores. Es recomendable, como es típico en cualquier situación donde se trabaje con imágenes, realizar algún tipo de realce sobre los valores de cada banda, para aprovechar

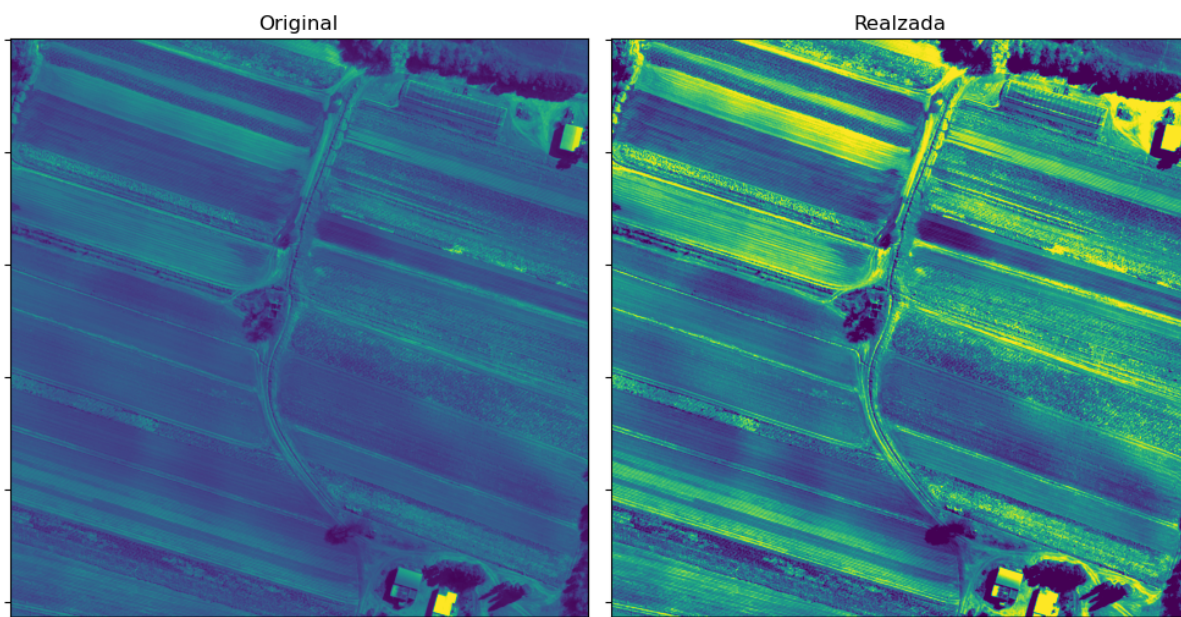


Figura 3.2: Resultado de realzar una de las bandas

al máximo la información. Para lograr esto se aplica una normalización para llevar los valores más altos a valores cercanos al 1 y los menores a valores cercanos a 0, ignorando los casos extremos al 4%, es decir, sin tener en cuenta los 4% mayores y menores valores. Luego se obtiene el índice NDVI[20] que se calcula mediante la siguiente formula:

$$NDVI = \frac{(NIR - Red)}{(NIR + Red)}$$

Y se le agrega como una banda extra a la imagen, resultando al final con una matriz de 5 dimensiones, en donde sus valores están en el intervalo [0, 1].

### 3.3. Segmentación

La etapa de segmentación, a su vez, puede sub-dividirse en 3 partes. Obtención de la **Máscara de bordes**, para la cual se usa el algoritmo de Canny. Creación de la **Matriz de distancias**, que se obtiene mediante la función `ndimage.distance_transform_edt` de la librería `scipy` y luego generación de los segmentos usando el algoritmo de **Watershed**

#### 3.3.1. Máscara de bordes

Una máscara de bordes es una matriz binaria que tiene las mismas filas y columnas que la imagen original, en donde un píxel con valor *True* corresponde a un píxel borde. Para obtener estos bordes se usó el algoritmo de Canny[3], uno de los más comunes en el área procesamiento de imágenes. Para entender el funcionamiento del algoritmo, es necesario comprender primero dos operaciones comúnmente usadas en el procesamiento de imágenes, los **filtros de Sobel**[24] y el **filtro Gaussiano**, también conocido como **Gaussian blur**. Ambos filtros resultan de aplicar una convolución a una matriz, que

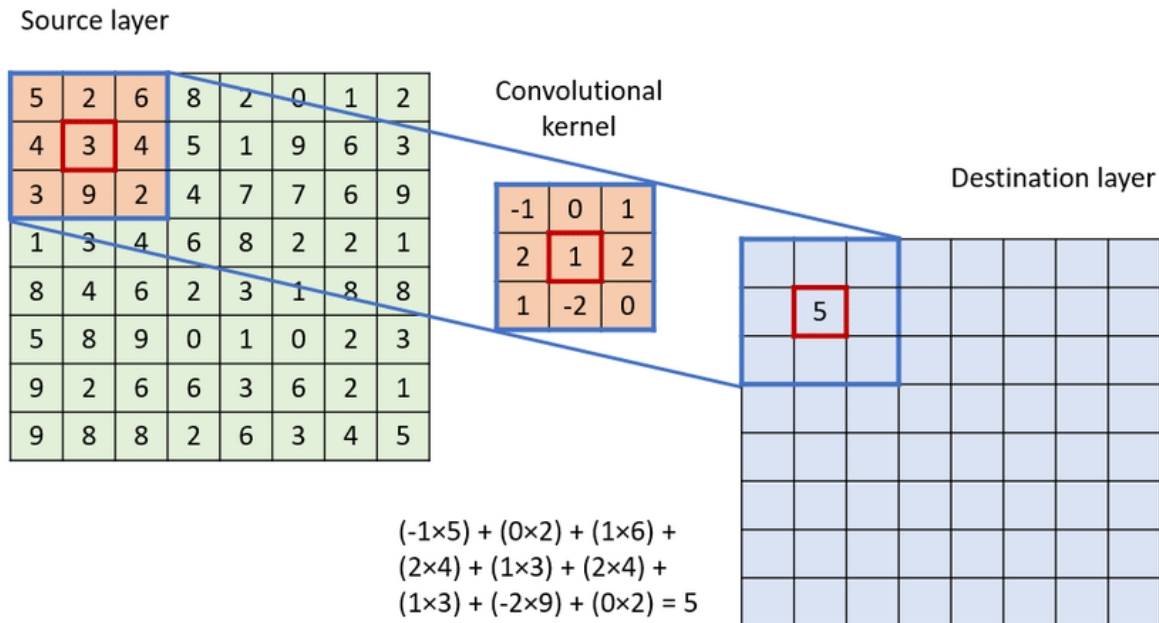


Figura 3.3: Visualización de una convolución aplicada a una matriz

en este caso será nuestra imagen en escala de grises, ya que estos algoritmos toman únicamente como input matrices de 2 dimensiones.

En el procesamiento de imágenes, un kernel o matriz de convolución, es una matriz usada para varios tipos de procesamientos, como *blurring*, detección de bordes, *sharpening*, entre otros. Esto se logra aplicando una convolución entre el kernel y la matriz de la imagen original. Simplemente una convolución mapea el valor de cada elemento en la matriz original, a un nuevo valor en función de los elementos que lo rodean (incluido el mismo), esa función es el kernel[25].

Para el caso del Gaussian blur, este se podría pensar simplemente como ponerle a cada píxel, el valor promedio ponderado entre si mismo y los píxeles que lo rodean. Esta distancia para considerar un píxel como vecino comúnmente se llama sigma, ya mayor sea sigma, mayor el efecto del filtro.

Ejemplo kernel para Gaussian blur:

$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$
$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$
$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{16}$

El filtro de Sobel, resulta de aplicar una convolución tanto vertical como horizontalmente. En ambos casos se busca determinar el valor de cada píxel según la variación entre el píxel anterior y el siguiente sobre cada eje. Luego de algunos cálculos trigonométricos, se obtiene para cada píxel un gradiente que representa la intensidad de la variación, y



Figura 3.4: Efectos del parámetro sigma para el filtro gaussiano

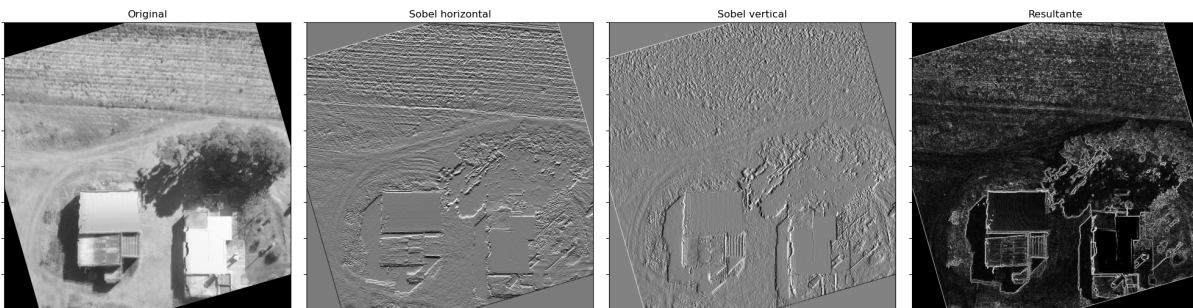


Figura 3.5: Aplicación del filtro de Sobel

también el ángulo del gradiente, que corresponde a la dirección de la variación.

Kernels para el filtro de Sobel:

Kernel horizontal	1	2	1		1	0	-1	Kernel vertical
	0	0	0		2	0	-2	
	-1	-2	-1		1	0	-1	

Luego Canny funciona de la siguiente manera, el input será una imagen en escala de grises, es decir una matriz de 2 dimensiones, el primer paso será aplicar un Gaussian Blur a la matriz para suavizarla y así eliminar algo de ruido, luego, se corre un filtro de Sobel para detectar el gradiente de cada píxel. Ahora, para determinar que píxeles se consideran bordes, primero es necesario reducir todos los potenciales bordes a líneas de 1 píxel de ancho, luego se aplica un doble umbral para determinar tres casos posibles.

Puede que la intensidad del gradiente de un píxel sea menor que el umbral inferior, por lo que es descartado como borde. Si el gradiente es mayor que el umbral superior, luego es considerado borde. Si el gradiente se encuentra dentro de este intervalo determinado por los umbrales, se considera potencial borde, y procede a una etapa extra de procesamiento llamada *hysteresis*, en donde solo se mantendrán estos potenciales bordes, si son continuación de algún borde ya detectado en el paso anterior.

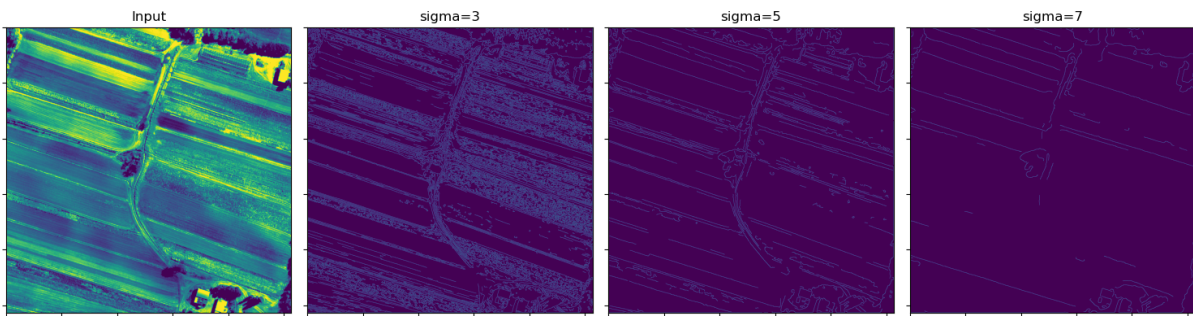


Figura 3.6: Efectos del parámetro sigma sobre los bordes detectados por Canny

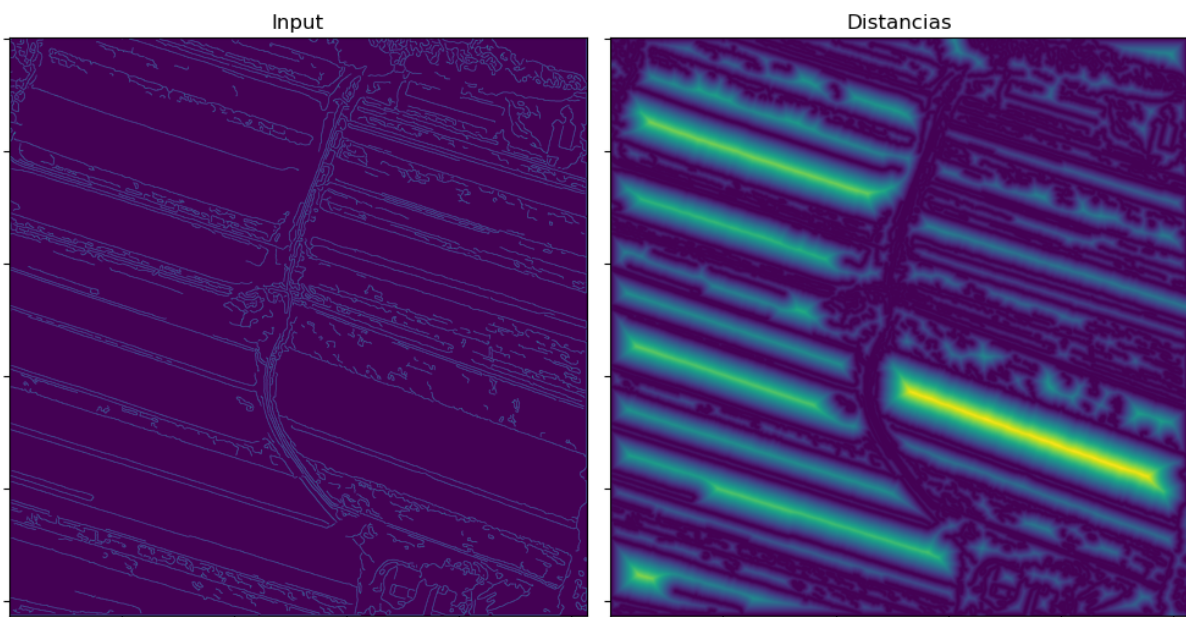


Figura 3.7: Matriz de distancias resultante

Afortunadamente, **Scikit-image**[21] cuenta con una implementación completa del algoritmo de Canny en `skimage.feature.Canny`, por lo que no es necesario implementar cada uno de los pasos intermedios.

### 3.3.2. Matriz de distancias

Una vez obtenida la máscara binaria de bordes, se construye la matriz de distancias. Para ello, se utiliza a la función `distance_transform_edt`, del paquete **ndi** de la librería **Scipy**[22], que usa como input la máscara de bordes y a cada píxel se le asigna como valor, la distancia euclidiana que tenga al píxel borde más cercano. Luego esta matriz se multiplica por -1 para que mientras más lejos se encuentre un píxel de algún borde, menor sea su valor resultante.

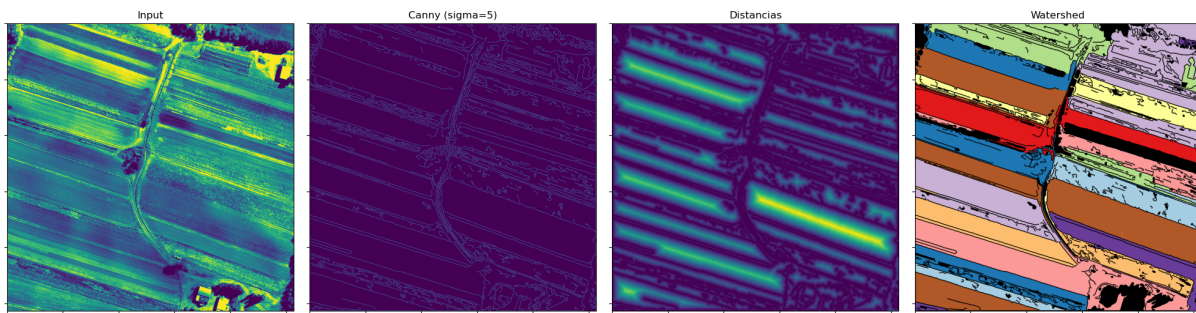


Figura 3.8: Todos los pasos de la etapa de segmentación

### 3.3.3. Algoritmo de Watershed

El algoritmo de Watershed[5, 19] es un algoritmo clásico de segmentación, a partir de una cantidad de marcadores determinada por el usuario, estos marcadores se ubican en los mínimos locales de la matriz de distancias, en este caso particular, en los píxeles mas lejanos a los bordes. Luego el algoritmo interpreta la matriz como un mapa topográfico, en donde el valor de cada píxel representa su altura, y comenzando desde los marcadores, se "inunda" la matriz, de esta forma, cada píxel se asocia a su mínimo local mas cercano y de esta forma se determinan los segmentos finales. Podemos acceder a la implementación de **Scikit-image** de este algoritmo a través del método `skimage.segmentation.watershed`

### 3.3.4. Regiones finales

Es muy probable que debido al funcionamiento del proceso de segmentación, resulten de este algunas regiones demasiado pequeñas para ser utilizadas posteriormente para la detección de líneas de cultivo y extraer información útil de ellas, por lo que se aplica además un paso extra para eliminarlas del conjunto final de regiones sobre el cuál continuará el procedimiento. Para ello, se calcula el área promedio de las regiones y se descartan aquellas que no posean un área mayor al 20% de la media regional.

## 3.4. Rectificación y recorte

El objetivo de esta etapa es el de procesar cada una de las regiones individualmente, para lograr que cada una tenga una geometría más regular y así facilitar los procesos de los pasos posteriores. Primero se busca rotar la imagen para orientar las potenciales líneas de cultivo para que queden horizontales, este proceso hace uso de la **Transformada de Hough**[6], un método para encontrar líneas rectas en imágenes. Luego, a cada región se le realiza un recorte tanto en el eje vertical como horizontal, para poder enfocarse más en áreas de interés y eliminar imperfecciones del paso de Segmentación, para esto es usada una heurística basada en el porcentaje de píxeles inactivos relativamente simple.

### 3.4.1. Rotación

Para poder rotar cada región y alinear las potenciales líneas de cultivo con el eje horizontal de la imagen es necesario determinar primero el ángulo de rotación, para ello,

primero se obtiene una máscara de bordes correspondiente a cada región usando las técnicas descritas en la etapa de segmentación, luego se usa una implementación probabilística basada en la Transformada de Hough[14] para encontrar líneas rectas sobre la imagen. A partir de esas líneas rectas, es posible determinar sus ángulos, y para obtener un único ángulo, se usa el algoritmo de KMeans, donde se toman los ángulos del grupo más poblado y se promedia su valor. Para este trabajo se utilizaron la implementación de KMeans de **Scikit-learn** y la transformada de Hough probabilística de **Scikit-image**.

En esencia, la transformada de Hough genera una matriz que representa el espacio de parámetros que construyen cada recta posible, sin embargo, no se utiliza la típica fórmula

$$ax + b$$

ya que  $x$  sería infinito para el caso de las líneas verticales. En cambio, cada recta es representada por dos parámetros,  $r$ , que corresponde a la longitud del segmento y  $\theta$ , el ángulo respecto al eje  $x$ . Para la versión probabilística del algoritmo, no se utiliza el espacio completo de rectas, sino una porción aleatoria de estas. Luego se calculan los píxeles que se encuentran cerca de cada recta, y sobre estos se buscan seguidillas de píxeles que no sean 0 (Recordar que como es una máscara de bordes, cada pixel es 0 o 1) que cumplan ciertas condiciones que podemos pasar por parámetros, como la longitud mínima de las rectas o la distancia máxima para la cual se conectan rectas.

Una vez obtenidas las líneas, se obtiene una lista con los ángulos de todas ellas y se realiza una clusterización mediante KMeans para separar en tres grupos, idealmente, uno de esos grupos tendrá una población mucho mayor al resto. A partir de los ángulos que pertenecen a este grupo mayoritario, se aplica un paso más de eliminación de extremos, para eso se ordenan, y se eliminan los 10% mayores y menores valores. Por último, sea  $\bar{a}$  el promedio de los ángulos restantes, se rota la región en  $\bar{a} + 180^\circ$  grados.

### 3.4.2. Recorte

Para el recorte, se realiza un proceso basado en la relación entre los píxeles activos y los píxeles totales. Primero se hace sobre el eje horizontal, entonces, para cada fila de la matriz  $M \times N$  correspondiente al índice NDVI de la región ya rectificadas, se le calcula la cantidad de píxeles activos, es decir, cantidad de píxeles con valor  $v > 0$ , luego se obtiene la proporción de píxeles activos respecto a la cantidad total, es decir,

$$\frac{\# \text{ píxeles activos}}{N}$$

Dado un umbral  $U$  determinado por el usuario, sea  $P_i$  la proporción de píxeles activos de la fila  $i$ , para  $i = 0, 1, \dots, M$ . Sea  $X$  el máximo  $i$  tal que  $P_X > U$  y  $x$  el mínimo respectivamente, la región de interés respecto al eje horizontal son las filas  $j$  tal que  $x \leq j \leq X$ . Para el eje vertical se aplica el mismo proceso, pero con un umbral distinto.

## 3.5. Detección de líneas de cultivo

En la etapa de detección de líneas de cultivo, el objetivo es a partir de cada región, orientada y recortada, lograr obtener algún marcador sobre cada una de los surcos reales.



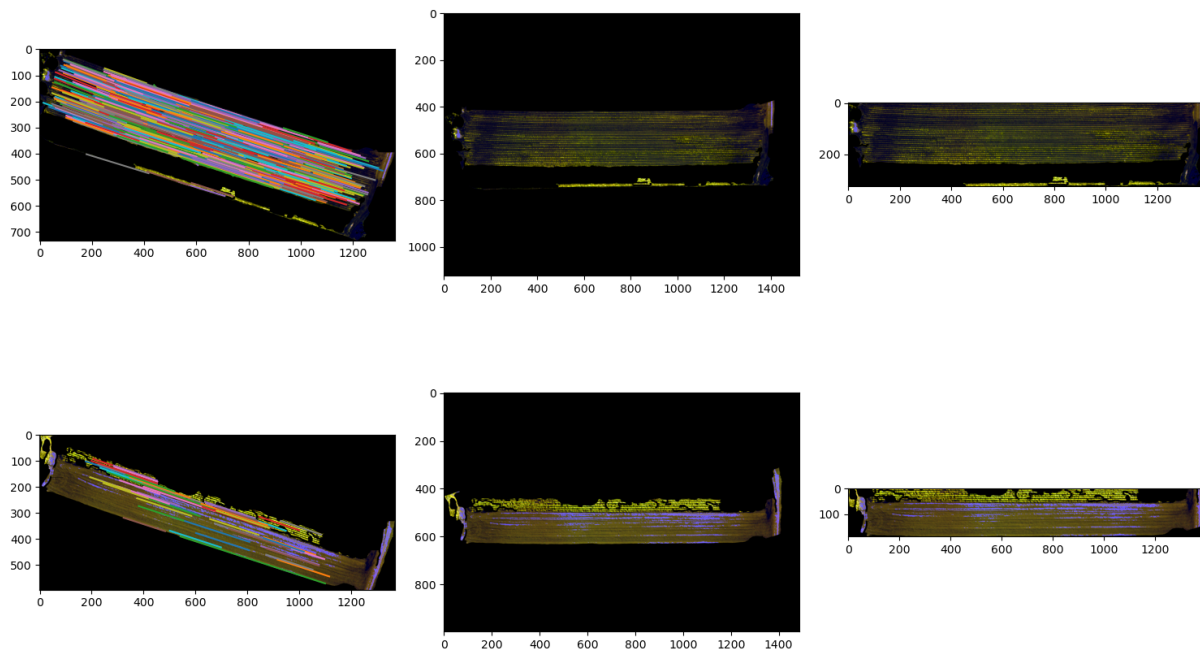


Figura 3.9: Líneas detectadas, rotación de la región y recorte a área de interés

Para esto pueden tomarse muchos acercamientos, como por ejemplo con una red neuronal [4] [26]. Sin embargo muchos de ellos involucran procesos supervisados, en donde es necesario tener de antemano acceso a estos datos, y expertos del dominio para su etiquetado. Además esos procesos son muy dependientes del conjunto de datos con los que se entrenan y podría ser muy costoso aplicarlo a un conjunto de datos de un contexto diferente, como pueden ser imágenes tomadas con una cámara en otras condiciones o de otras características técnicas. Es por eso que se decidió utilizar un algoritmo no supervisado basado en el índice NDVI, que permite adaptar fácilmente el algoritmo a cada situación, la única exigencia es que los surcos deben estar orientados con el eje horizontal de la imagen.

Se comienza por la matriz  $M \times N$  correspondiente al índice NDVI y para cada fila calcularemos la suma de todos sus píxeles (por supuesto, sobre el eje horizontal), luego obtenemos como resultado un arreglo de  $N$  elementos, que podemos imaginarlos como la función intensidad del índice NDVI respecto de la posición sobre el eje vertical. Ahora, se hace uso del método `find_peaks` de la librería `scipy.signal` para encontrar máximos y mínimos locales en la función, luego las filas en donde se encuentran los mínimos corresponden al espacio no cultivado entre líneas, y los máximos a la filas específicas de cada surco donde su valor NDVI es mayor.

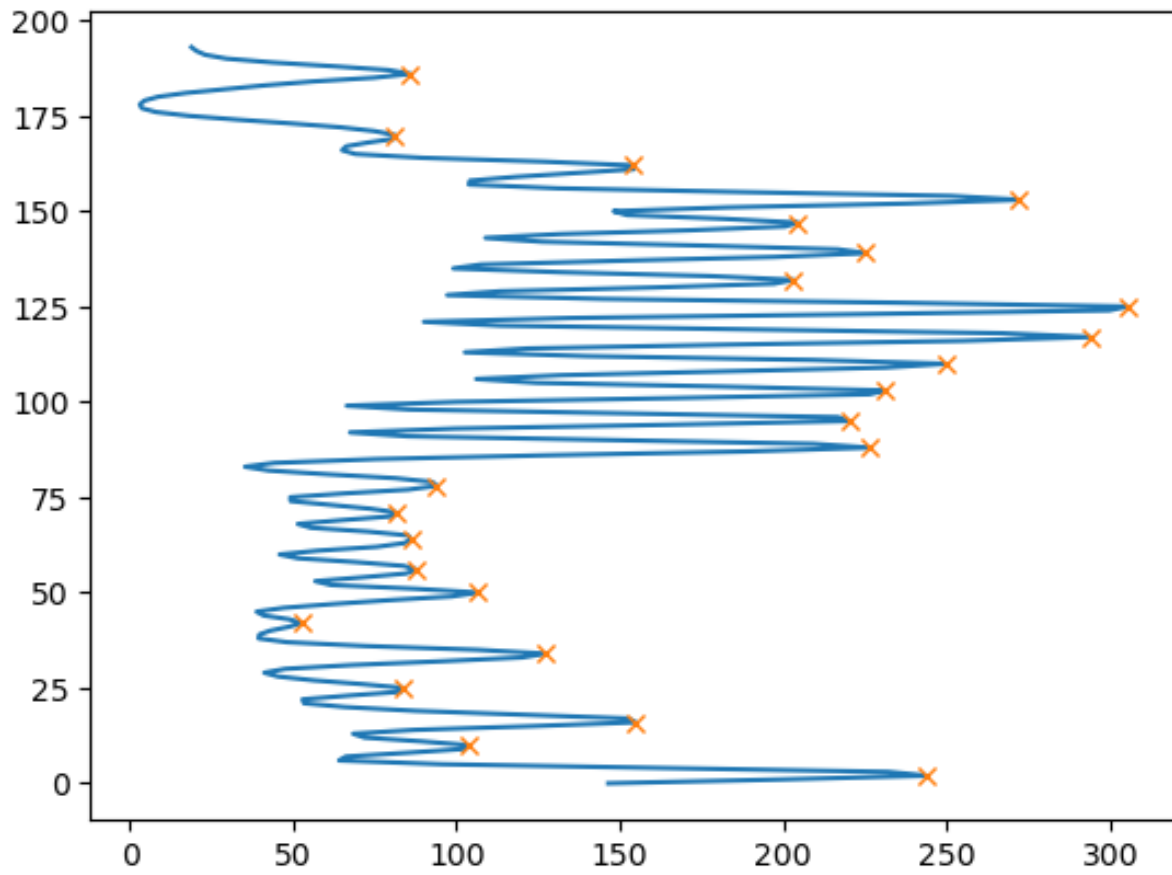


Figura 3.10: Máximos locales encontrados en la función intensidad del índice NDVI

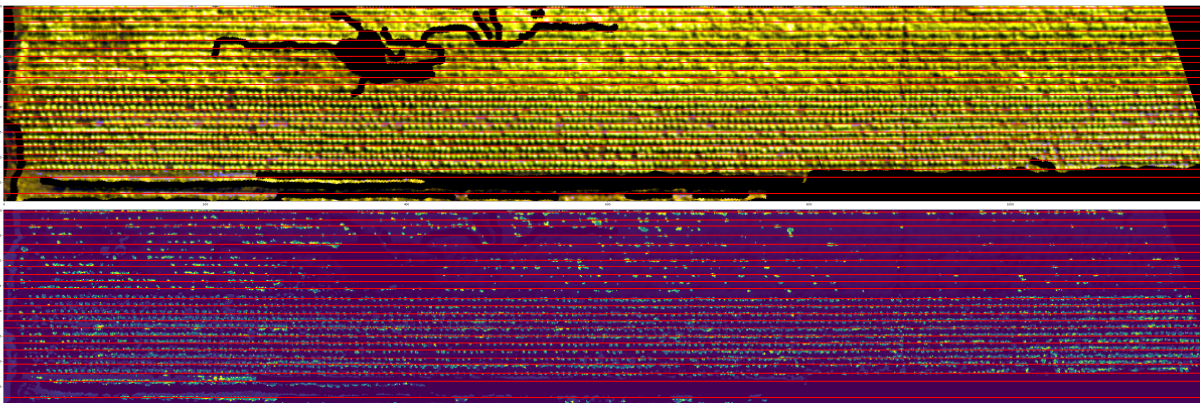


Figura 3.11: Líneas de cultivo detectadas

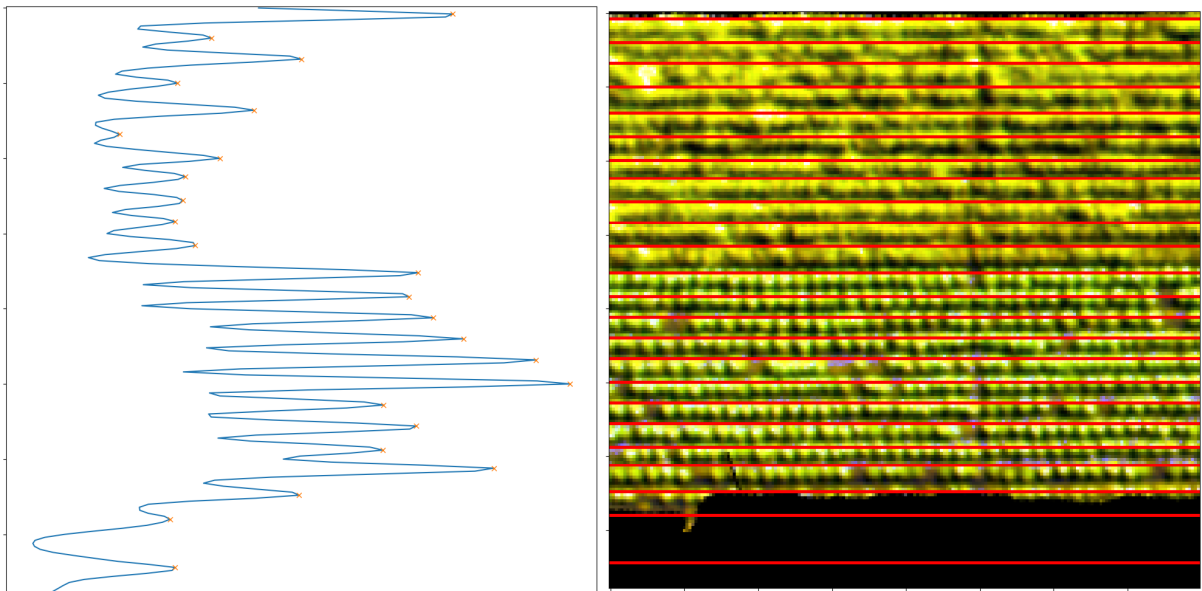


Figura 3.12: Correlación entre máximos locales detectados y las líneas de cultivo



# Capítulo 4

## Experimentos

Para la etapa de segmentación, la cuál es la más sensible a cambios de parámetros, se realizó una búsqueda en grilla para determinar cuál combinación de estos parámetros es óptima. Para ello, es necesario primero, definir una métrica para poder comparar numéricamente la calidad de cada segmentación.

### 4.1. Métricas

Se consideraron dos métricas posibles para el experimento, principalmente debido a su naturaleza no supervisada. Son la **Suma de Espíndola** [7] y **F de Johnson** [10]. Ambas métricas operan sobre la **varianza ponderada (vp)** como medida de homogeneidad interna de cada segmento, y la **autocorrelación espacial (ae)** como medida de heterogeneidad entre los distintos segmentos.

La varianza ponderada se calcula con la siguiente formula:

$$vp = \frac{\sum_{i=1}^n a_i v_i}{\sum_{i=1}^n a_i}$$

donde  $n$  es la cantidad total de segmentos y  $a_i, v_i$  son el área y varianza del segmento  $i$  respectivamente.

La autocorrelación espacial se mide mediante el índice  $I$  de Moran [15], que busca la diferencia entre segmentos vecinos, para ello, primero es necesario computar el grafo de adyacencia correspondiente a cada segmentación, luego el índice se calcula de la siguiente manera:

$$I = \frac{n \sum_{i=1}^n \sum_{j=1}^n w_{ij} (y_i - \bar{y})(y_j - \bar{y})}{(\sum_{i=1}^n (y_i - \bar{y}))^2 (\sum_{i \neq j} \sum w_{ij})}$$

donde nuevamente,  $n$  es la cantidad total de segmentos,  $\bar{y}$  es el valor promedio de la imagen,  $y_i$  es el valor promedio del segmento  $i$  y  $w_{ij}$  representa la adyacencia de los segmentos  $j$  e  $i$ , luego es 1 si son adyacentes, y 0 en caso contrario.

La suma de Espíndola busca una combinación de valores bajos de  $vp$  y de  $I$ , y se obtiene de la siguiente manera:

$$F(x) = vp(x) + I(x)$$

Como se hace claro en esta formula, la suma de Espíndola otorga la misma importancia a la homogeneidad intra-segmento que a la heterogeneidad inter-segmento, lo que puede

ser un problema en el caso particular de las imágenes aéreas de cultivos, ya que los segmentos deseados son altamente texturados y por ende variables, lo que podría llevar a una sobre-segmentación no deseada.

En cambio, la F de Johnson cuenta con un parámetro  $\alpha$  que nos permite elegir la importancia que le queremos dar a cada componente, la formula es la siguiente

$$F(x) = (1 + \alpha^2) \frac{vp(x) \times I(x)}{\alpha^2 \times vp(x) \times I(x)}$$

Es por eso que decidió por utilizar la F de Johnson, con un parámetro  $\alpha = 0.5$  como métrica de comparación de segmentaciones, para darle el doble de importancia al índice de Moran por sobre la varianza ponderada.

## 4.2. Búsqueda de parámetros

Para encontrar o acercarse a los parámetros óptimos, se optó por una estrategia de búsqueda conocida como **Grid search** o **Búsqueda en cuadrilla**, en la cuál se determina el espacio total de combinaciones de parámetros posibles y se corre el programa para todas ellas, luego se utiliza alguna métrica numérica que nos permita comparar y ordenar de menor a mayor los resultados con cada combinación de parámetros y se elige la deseada, que usualmente es la que mejor valor de la métrica retornó. Para el caso de este trabajo y como fue descrito en la sección anterior, fue seleccionada la F de Johnson para cumplir este rol.

Los parámetros que se buscaron optimizar fueron los siguientes:

- La combinación de bandas de la imagen a utilizar. A elegir 3 bandas de las 5 posibles
- La cantidad de marcadores de Watershed a utilizar, 50, 100, 200 o 500.
- El valor de Sigma para el algoritmo de Canny. 1, 3, 5 o 10.

Este espacio de parámetros resulta en **200** combinaciones posibles.

Luego de correr el algoritmo de Grid search, se inspeccionaron visualmente los mejores 5 resultados según la F de Johnson y se decidió por la siguiente combinación de parámetros.

- Bandas Red Edge, NIR, y NDVI
- 50 marcadores
- Sigma = 5

# Capítulo 5

## Análisis de resultados

Para el análisis de resultados, se dividirá en 3 etapas: Segmentación, Rotación y recorte y Detección de líneas, cada una correspondiente a la respectiva etapa del procesamiento. La etapa inicial de preparación no será analizada en detalle debido a su simple naturaleza y predecibles resultados.

### 5.1. Segmentación

La etapa de segmentación es quizás la más importante de todas en término de su influencia con el resultado final, y es mucho más sensible a los parámetros de entrada que el resto de procedimientos. En el caso de la imagen usada durante el desarrollo, la segmentación es de alta calidad, tanto por la predisposición de la imagen misma para ser segmentada, como por el largo tratamiento de ajuste de parámetros que se realizó para ella.

La mayoría de regiones corresponden a un lote correctamente segmentado (5.1), sin embargo, una imperfección común de este algoritmo de segmentación es la sobre-segmentación de lotes (5.1), y el caso opuesto, la fusión de lotes claramente independientes en una misma región (5.1).

Otro problema frecuente es la propagación de la región hacia sectores adyacentes al lote, es decir, la región correctamente rodea el lote pero se expande más allá de este, siguiendo caminos, arboledas u otros sectores (5.1). Dependiendo el tamaño y forma de las impurezas, estas pueden o no ser resueltas en la etapa de recorte.

### 5.2. Rotación y recorte

Dado que la rotación se basa completamente en la capacidad de encontrar líneas rectas en la imagen, mientras mejor y mas marcados se vean los surcos, mejor será el resultado. Luego hay dos situaciones que podrían hacer que el algoritmo no funcione correctamente, no encontrar suficientes líneas rectas, que podría deberse a que los surcos no resaltan lo suficiente, o encontrar demasiadas líneas, que ocurre cuando durante la segmentación, la región se expande más allá del lote e incluye áreas cercanas que no tengan la misma

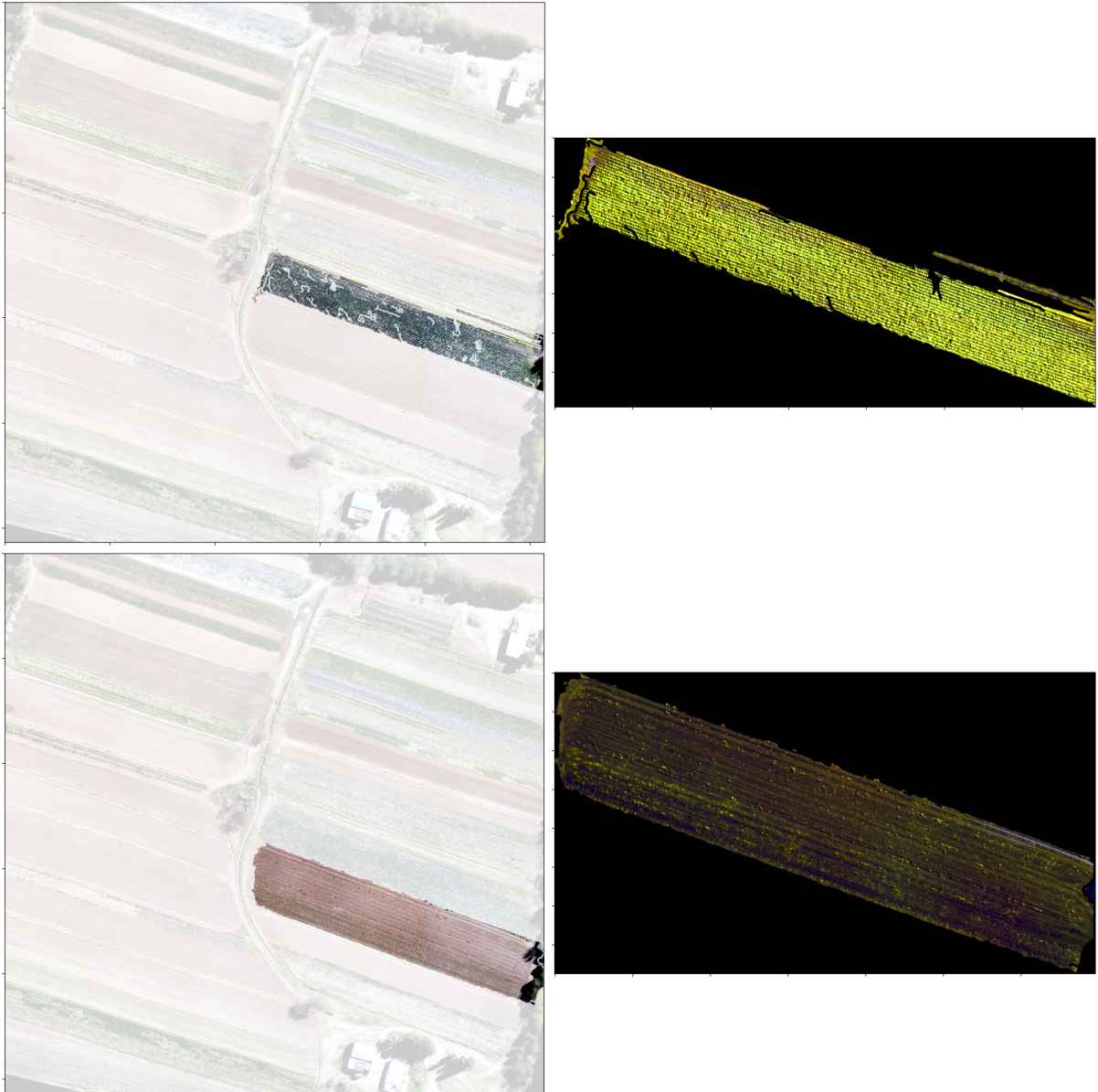


Figura 5.1: Lotes correctamente segmentados



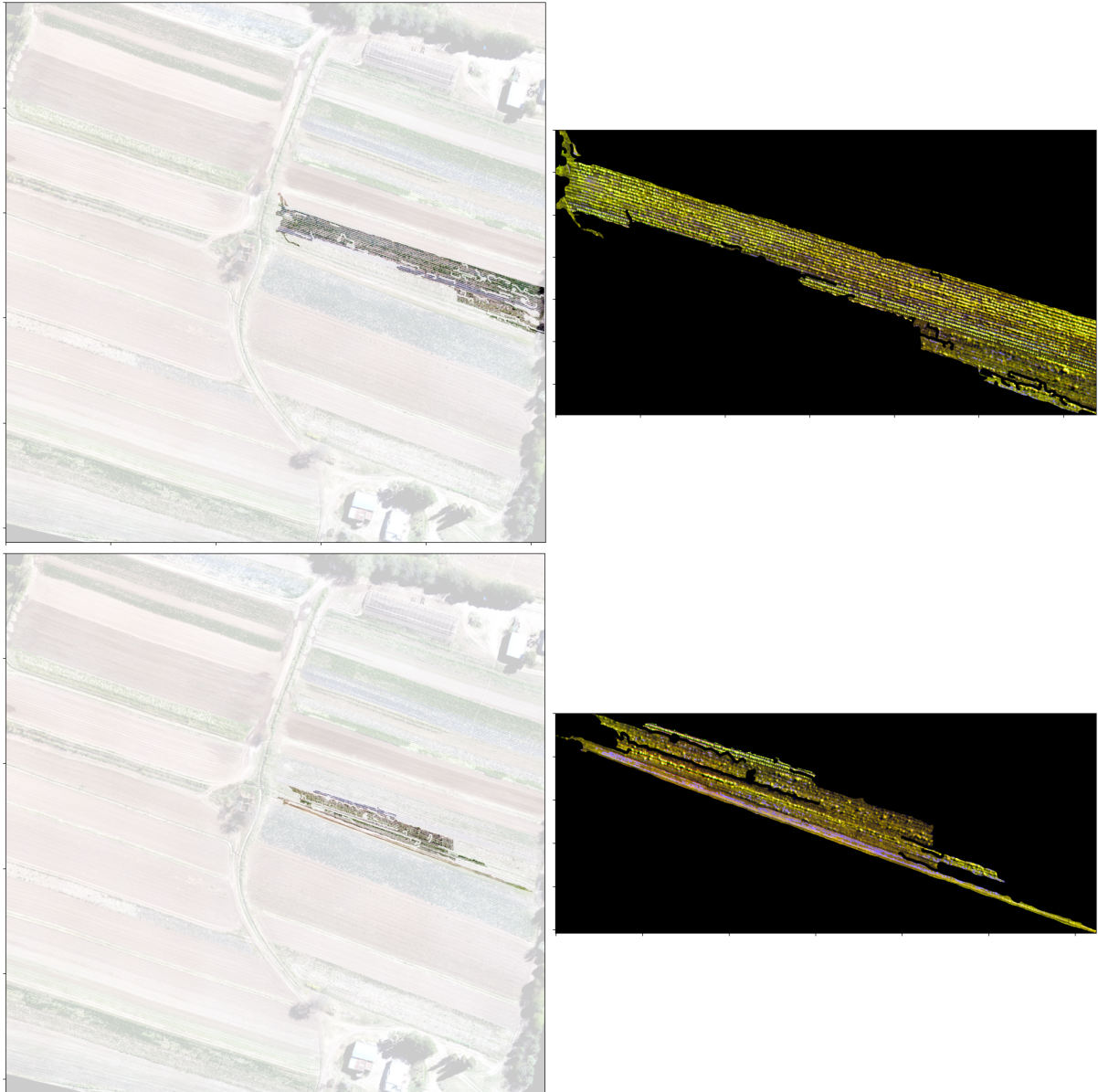


Figura 5.2: Lote sobre-segmentado. Notar que uno es el complemento del otro

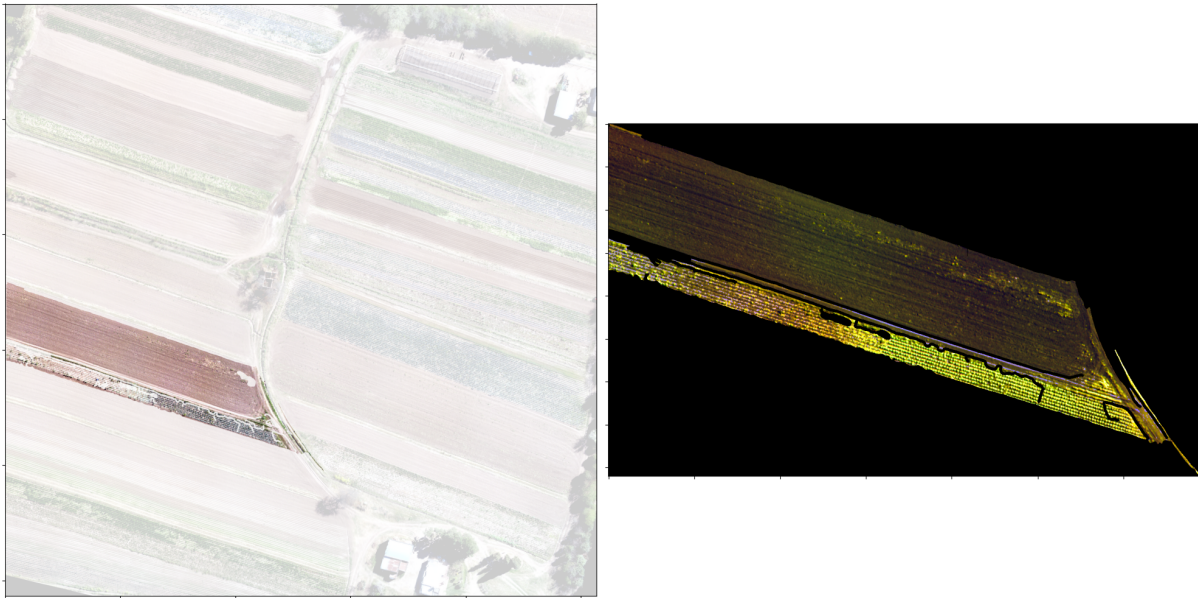


Figura 5.3: Lotes fusionados

orientación. Como regla general, si la segmentación es buena, también lo será la rotación (5.2).

El recorte suele cumplir su función de aislar el área de interés con una consistencia más que aceptable, pero nuevamente, fallas en la etapa de segmentación, pueden generar resultados no deseados.

### 5.3. Detección de líneas de cultivo

Es clave para la detección de líneas de cultivo, que la rotación sea correcta, ya que depende en su totalidad de la capacidad de acumular sobre el eje horizontal el valor NDVI de los píxeles, por lo que si el ángulo de rotación es incorrecto, aunque sea por muy poco, puede resultar en líneas de cultivo totalmente erróneas. Nuevamente, la calidad de la segmentación termina culminando en una buena detección de líneas (5.3). Por otro lado, la dependencia de este proceso del índice NDVI, hace que los resultados sobre lotes donde la semilla aun no ha surgido de la tierra o donde todavía no se ha sembrado son cuestionables (5.3).

En los casos donde en la etapa de segmentación hubo propagación hacia otros sectores, se produce una sobre-detección de líneas de cultivo, donde claramente no las hay (5.3).

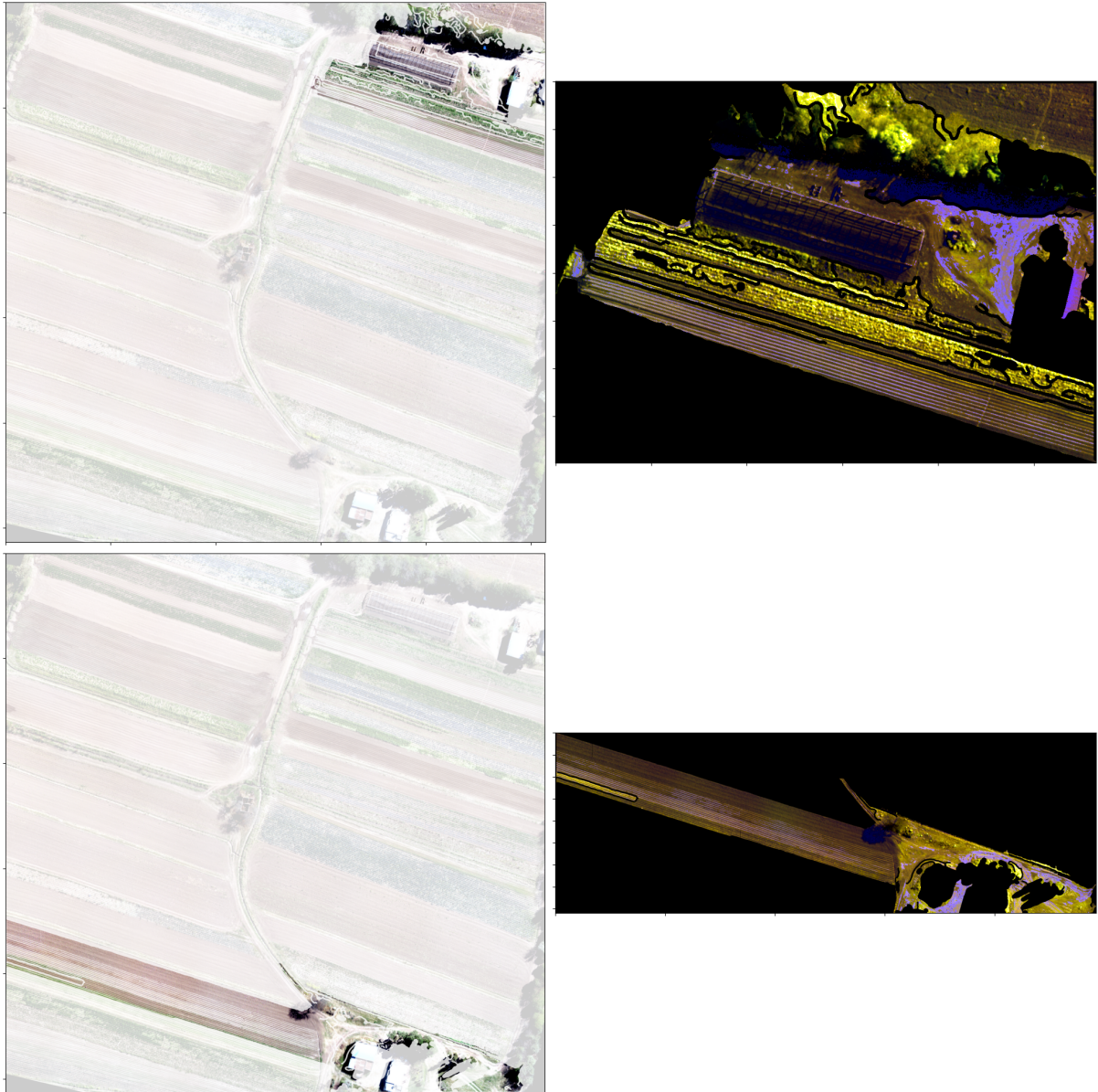


Figura 5.4: Lotes con propagación a otros sectores

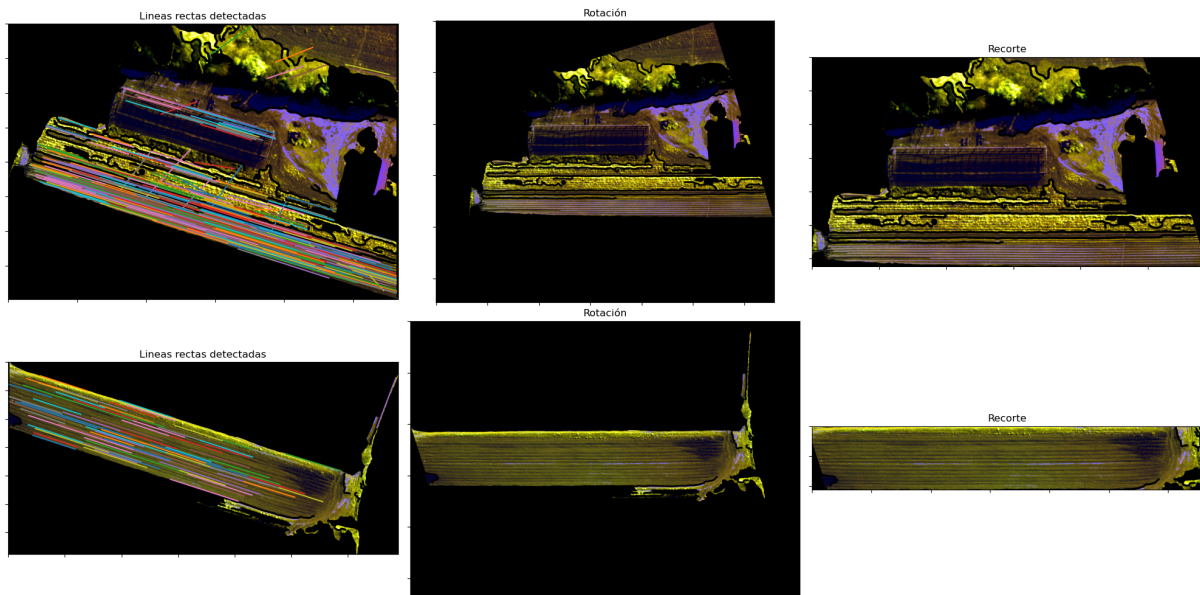


Figura 5.5: Etapa de rotación y recorte

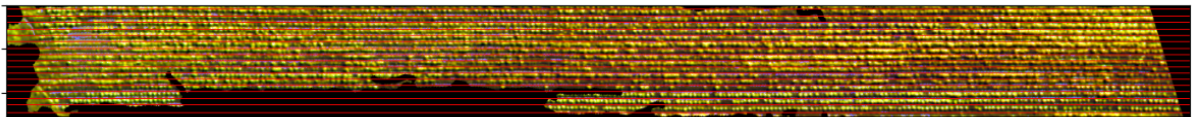


Figura 5.6: Lineas de cultivo detectadas en un lote ya surgido

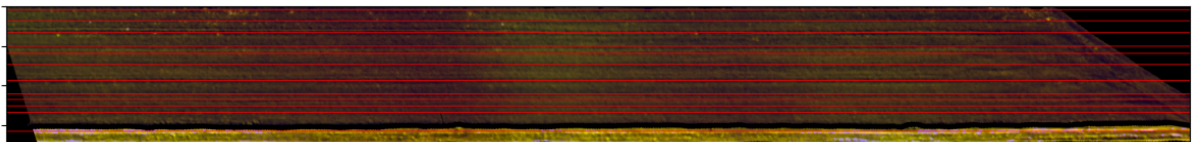


Figura 5.7: Lineas de cultivo en un lote no sembrado

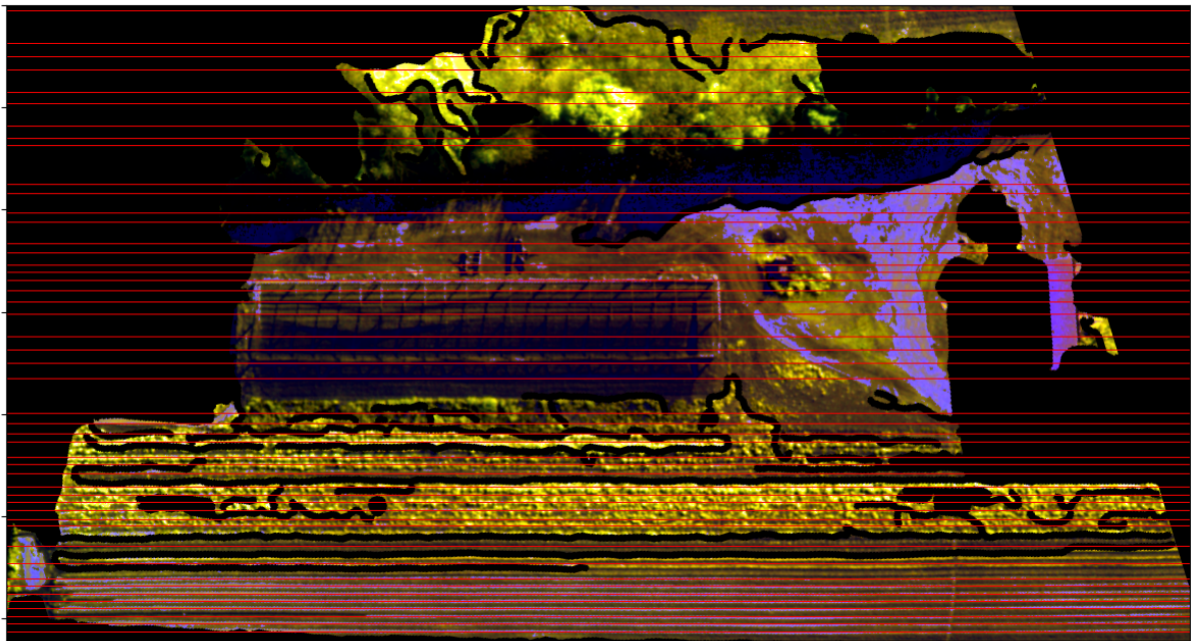


Figura 5.8: Líneas detectadas en un lote con propagación a otro sector



# Capítulo 6

## Conclusión y Trabajo Futuro

### 6.1. Conclusión

Como se vio a lo largo de este trabajo, es muy común la recurrencia a datos etiquetados para este tipo de desarrollos, y junto con ellos, a modelos complejos de Machine Learning o Redes Neuronales, que además de ser computacionalmente exigentes, los conjuntos de datos de los cuales dependen pueden ser extremadamente difíciles y costosos de conseguir, además de estar atados a determinadas condiciones del contexto donde fueron generados, como la ubicación geográfica, estación del año, hora del día, etc. Se demostró en esta tesis que es posible un acercamiento no supervisado a la problemática de la detección de líneas de cultivo, y al trabajo con imágenes de alta resolución sobre campos hortícolas en contexto multiespecies (complejo). Basándose principalmente en el empleo inteligente de técnicas clásicas del procesamiento de imágenes, que prueban ser fácilmente adaptables y transferibles a aplicaciones similares.

### 6.2. Trabajo Futuro

Es evidente que el etapa de **segmentación** es fundamental para los resultados finales, debido a que es el punto de partida para las etapas de rotación, recorte y detección de líneas, por lo que una mejora en el algoritmo de segmentación resultaría en una mejoría directa de la confiabilidad y calidad de los resultados del proceso completo. Una primera instancia de trabajo futuro, es la exploración más profunda de los posibles algoritmos de segmentación y preprocesamiento aplicado a las imágenes que estos usan como input. La librería de **Sklearn-image** (`skimage.segmentation`) ofrece una amplia variedad de algoritmos de segmentación, muchos de ellos los cuales no fueron considerados durante el desarrollo de esta tesis, ni tampoco la posibilidad de combinar varios de estos.

Otro posible desarrollo que probaría ser de utilidad sería el entrenamiento de un clasificador de regiones, ya que luego de la etapa de segmentación, muchas de las regiones resultantes no corresponden a lotes utilizados para la siembra, sino a otros paisajes comunes de los campos donde no corresponde la búsqueda de líneas de cultivo, como pueden ser caminos, galpones, arboledas o casas. Si bien el concepto de no-supervisión es bastante central para esta tesis, ya que al momento de trabajar con imágenes aéreas de campos, el

acceso a grandes cantidades de datos para entrenamiento puede resultar costoso, se cree posible la creación de un buen clasificador sin la necesidad de una gran cantidad de datos etiquetados.

También, queda como un pendiente el estudio y exploración de resultados de aplicar el procesamiento desarrollado durante este trabajo a imágenes de distinta resolución, con el fin de comprobar la generalidad y adaptabilidad de la metodología lograda.



# Bibliografía

- [1] Miriam Alfie Cohen. “Planeación urbana y medio ambiente: los cinturones verdes”. En: *Revista de temas contemporáneos, sobre lugares, política y cultura* (2011).
- [2] BID, Banco Interamericano de Desarrollo. *Programa de desarrollo de tecnologías satelitales (PROSAT II) AR-L1310*. Inf. téc. BID, July 31, 2019. URL: <https://www.iadb.org/es/project/AR-L1310>.
- [3] John Canny. “A Computational Approach to Edge Detection”. En: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (1986), págs. 679-698. DOI: 10.1109/TPAMI.1986.4767851.
- [4] L. Casuccio y A. Kotze. “Corn planting quality assessment in very high-resolution RGB UAV imagery using Yolov5 and Python”. En: *AGILE: GIScience Series 3* (2022), pág. 28. DOI: 10.5194/agile-giss-3-28-2022. URL: <https://agile-giss.copernicus.org/articles/3/28/2022/>.
- [5] H. Digabel y C. Lantuéjoul. “Iterative algorithms”. En: *Actes du Second Symposium Europeen d’Analyse Quantitative des Microstructures en Sciences des Materiaux, Biologie et Medecine*. Ed. por R Verlag. Oct. de 1978, págs. 85-99.
- [6] Richard O. Duda y Peter E. Hart. “Use of the Hough Transformation to Detect Lines and Curves in Pictures”. En: *Commun. ACM* 15.1 (ene. de 1972), págs. 11-15. ISSN: 0001-0782. DOI: 10.1145/361237.361242. URL: <https://doi.org/10.1145/361237.361242>.
- [7] GM Espindola et al. “Parameter selection for region-growing image segmentation algorithms using spatial autocorrelation”. En: *International Journal of Remote Sensing* 27.14 (2006), págs. 3035-3040.
- [8] FAO. *Fruit and vegetables—Your dietary essentials*. 2020.
- [9] Beatriz Liliana Giobellina. *La alimentación de las ciudades. Transformaciones territoriales y cambio climático en el cinturón verde de Córdoba*. Inf. téc. Ediciones INTA, 2018.
- [10] Brian Johnson et al. “Image segmentation parameter optimization considering within- and between-segment heterogeneity at multiple scale levels: Test case for mapping residential areas using Landsat imagery”. En: *ISPRS International Journal of Geo-Information* 4.4 (2015), págs. 2292-2305.
- [11] Maria Victoria Marinelli. “Herramientas analíticas de valoración y cuantificación de la producción hortícola basada en sensores remotos”. Tesis de mtría. 2020.
- [12] Maria Victoria Marinelli et al. “Geoscience and remote sensing on horticulture as support for management and planning”. En: (2019).

- [13] Maria Victoria Marinelli et al. “Sustainable Food Supply by Peri-Urban Diversified Farms of the Agri-Food Region of Central Córdoba, Argentina”. En: *Land* 12.1 (2023), pág. 101.
- [14] Jiri Matas, Charles Galambos y Josef Kittler. “Progressive Probabilistic Hough Transform”. En: *British Machine Vision Conference*. 1998.
- [15] Percy artist Moran. “Notes on continuous stochastic phenomena.” En: *Biometrika* (1950).
- [16] *Numpy python library*. URL: <https://numpy.org/>.
- [17] Diego H. Pons et al. *MONITOREO ESPACIAL DE LA PRODUCCIÓN HORTÍCOLA EN PERIURBANOS DE LA REPÚBLICA ARGENTINA. Informe Hito 3 – Sistema implementado en prueba*. Inf. téc. PROSAT II del BID, CONAE, 2022.
- [18] *Rasterio python docs*. URL: <https://rasterio.readthedocs.io/en/stable/>.
- [19] Jos B T M Roerdink y Arnold Meijster. “The Watershed Transform: Definitions, Algorithms and Parallelization Strategies”. En: *Fundamenta Informaticae* 41 (2000), págs. 187-228.
- [20] J\_W Rouse Jr et al. “Monitoring vegetation systems in the Great Plains with ERTS”. En: (1974).
- [21] *Scikit-image python library*. URL: <https://scikit-image.org/>.
- [22] *Scipy python library*. URL: <https://scipy.org/>.
- [23] Alimohammad Shirzadifar et al. “Mapping crop stand count and planting uniformity using high resolution imagery in a maize crop”. En: *Biosystems Engineering* 200 (2020), págs. 377-390. ISSN: 1537-5110. DOI: <https://doi.org/10.1016/j.biosystemseng.2020.10.013>. URL: <https://www.sciencedirect.com/science/article/pii/S1537511020302919>.
- [24] Irwin Sobel. “An Isotropic 3x3 Image Gradient Operator”. En: *Presentation at Stanford A.I. Project 1968* (feb. de 2014).
- [25] *Wikipedia - Kernel (image processing)*. URL: [https://en.wikipedia.org/wiki/Kernel\\_\(image\\_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing)).
- [26] Ruoyu Yang et al. “Detecting functional field units from satellite images in smallholder farming systems using a deep learning based computer vision approach: A case study from Bangladesh”. En: *Remote Sensing Applications: Society and Environment* 20 (2020), pág. 100413. ISSN: 2352-9385. DOI: <https://doi.org/10.1016/j.rsase.2020.100413>. URL: <https://www.sciencedirect.com/science/article/pii/S2352938519301831>.