

# **Uso de Aprendizaje Automático para la corrección de Modelos Digitales de Elevación Radar con imágenes Landsat**

Tesis para optar al título  
Ingeniería De Sistemas



Facultad de Ciencias Exactas  
UNCPBA  
Argentina

**Alumna:** Carolina Bruscantini - bruscantinic@gmail.com

**Directores:** Dr Lucas Lo Vercio y Dr Pablo Rinaldi

# **Uso de Aprendizaje Automático para la corrección de Modelos Digitales de Elevación Radar con imágenes Landsat**

## **Resumen**

Los Modelos Digitales de Elevación (DEMs) representan el relieve de la superficie terrestre mediante un conjunto de puntos distribuidos de forma homogénea, generalmente en formato de grilla o ráster, que contienen una cota de altura en referencia al nivel del mar.

Existen diferentes métodos para la obtención de un DEM desde el relevamiento terrestre con instrumental específico hasta la reconstrucción a partir de imágenes aéreas o satelitales con diferentes tipos de sensores.

A nivel global, los DEMs más ampliamente difundidos son los de origen radar producidos por la NASA, denominados SRTM, por su probada calidad, su cobertura cercana al 90 % del planeta y disponibilidad gratuita en versión de 30 metros de resolución.

Estos modelos representan la elevación natural del terreno y todo lo que esté sobre éste, sea natural o artificial como árboles o edificaciones. Para ciertas aplicaciones como la simulación de escurrimiento superficial esto representa un inconveniente. En particular, cuando los árboles se encuentran sembrados en conjunto pueden generar grandes diferencias entre la superficie del DEM y la superficie real por donde escurre el agua. Este es el caso de las zonas rurales de la provincia de Buenos Aires donde los árboles son generalmente plantados por el hombre en hileras que se transforman en muros a la hora de simular inundaciones sobre un DEM SRTM.

En el presente trabajo se propone una herramienta que permite la detección automática de la vegetación arbórea en DEMs de origen radar con el fin de corregir estas diferencias y mejorar su eficiencia en el modelado de inundaciones.

Para la detección automática se generaron diferentes clasificadores supervisados (Random Forest) a partir de distintas zonas que fueron evaluados con un conjunto de datos de testeo obteniendo resultados de hasta un F1 Score de 0,8 en los casos que incluyeron indicadores relacionados con las texturas del terreno. Este proceso genera como resultado máscaras de vegetación arbórea que luego son utilizadas para corregir los DEMs a través de un método de interpolación. Los DEMs corregidos fueron comparados utilizando DEMs de alta resolución obtenidos a través de la tecnología LiDAR, los cuales representan un modelo de tierra desnuda.

Todo el proceso se automatizó mediante la implementación de un plug-in para la herramienta GIS de uso libre QGis.

# Índice general

<b>1. Introducción</b>	<b>5</b>
1.1. Motivación . . . . .	5
1.2. Objetivos . . . . .	7
1.3. Estado del arte . . . . .	7
<b>2. Imágenes Satelitales</b>	<b>9</b>
2.1. GIS y Herramientas para Procesamiento de Imágenes . . . . .	9
2.2. Sistema de Referencia de Coordenadas . . . . .	10
2.3. Proyecciones . . . . .	11
2.4. Imágenes Landsat . . . . .	12
2.4.1. Pre-procesamiento . . . . .	13
2.4.2. Landsat Surface Reflectance Data . . . . .	14
2.5. DEM, DSM y DTM . . . . .	14
2.6. DEMs derivados de SRTM . . . . .	15
2.6.1. Generación de Mosaicos y Recorte de Rasters . . . . .	16
2.6.2. Alineación de Rasters . . . . .	16
2.7. Imágenes LiDAR . . . . .	17
2.8. Imágenes Vectoriales . . . . .	19
<b>3. Método Propuesto</b>	<b>20</b>
3.1. Detección de Árboles . . . . .	22
3.1.1. Random Forest . . . . .	22
3.1.1.1. Importancia de los Indicadores . . . . .	23
3.1.2. Métricas . . . . .	24
3.1.3. Indicadores . . . . .	25
3.1.3.1. Texturas de Haralick: Re-escalado vs Recodificado .	27
3.2. Corrección de Árboles . . . . .	32
3.3. QGIS Python Plugin: Lumberjack . . . . .	33
3.3.1. La Interfaz Gráfica . . . . .	33
3.3.2. Utilización de la Herramienta . . . . .	36
<b>4. Resultados</b>	<b>38</b>
4.1. Zonas de Estudio . . . . .	38
4.2. Resultados de la clasificación: Métricas y Visuales . . . . .	41
4.2.1. Resultados para una Misma Región . . . . .	41
4.2.2. Validación Utilizando una Región Distinta . . . . .	44
4.2.3. Caso Completo . . . . .	46
4.3. Estudio del Efecto Estacional . . . . .	49
4.4. Corrección de Árboles . . . . .	51

4.5. Validación con LiDAR . . . . .	53
<b>5. Conclusión</b>	<b>56</b>
5.1. Contribuciones . . . . .	57
5.2. Trabajos Futuros . . . . .	57
<b>A. Apéndice</b>	<b>59</b>
A.1. Herramientas . . . . .	59
A.1.1. OSGeo4W . . . . .	59
A.1.1.1. OSGeo4W Shell . . . . .	59
A.1.1.2. QGIS . . . . .	59
A.1.1.3. GRASS . . . . .	60
A.1.1.4. QGIS Processing Toolbox . . . . .	61
A.1.1.5. GDAL . . . . .	61
A.1.2. Python y Scikit-learn . . . . .	62
A.1.3. Otros Plugins en QGIS . . . . .	63
A.1.3.1. Plugin Builder y Plugin Reloader . . . . .	63
A.1.3.2. Profile Tool . . . . .	63
A.1.4. ArcGIS World Imagery Map Server . . . . .	63
A.2. Cálculo de extensión y recorte de imágenes . . . . .	64
A.3. GDAL, Python y NumPy para calcular NDVI . . . . .	64
A.4. Cálculo de texturas con GRASS Python Scripting Library . . . . .	65
A.5. Re-escalado vs Recodificado . . . . .	71

# Capítulo 1

## Introducción

Se denomina Modelo Digital de Elevación (MDE o DEM por sus siglas en inglés) a un modelo simbólico de estructura numérica y digital [Fel94] que pretende representar la distribución espacial de la elevación del terreno, siendo la altura una variable escalar (sobre un nivel de referencia) que se distribuye en un espacio bidimensional. Generalmente se presentan en formato de grilla (raster) formando una matriz bidimensional de filas y columnas equidistantes con una cota de valor de altura de terreno en cada celda.

Los DEMs pueden ser derivados de estudios de campo, de mapas de curvas de nivel, o de imágenes aéreas o satelitales de origen radar, óptico o laser. Los DEMs obtenidos a partir de imágenes de Radar con antena de Apertura Sintética (SAR) son ampliamente utilizados, ya que es muy sensible a las variaciones del terreno. La imagen radar es producida por un sensor activo que emite microondas, recibiendo y registrando los ecos reflejados por el objetivo. La ventaja sobre sistemas infrarrojos y ópticos, es que las imágenes radar siempre son nítidas independientemente de que se capten de día, de noche, o bajo cualquier condición climática. Los DEMs radar más utilizados son los SRTM (Shuttle Radar Topography Mission) de la NASA, por su probada calidad y gratuidad.

### 1.1. Motivación

Uno de los inconvenientes con los DEMs radar es que, por la frecuencia que utilizan, la señal no penetra en la vegetación dando cotas incorrectas de nivel de terreno en zonas con vegetación alta, en especial árboles. En estos modelos los datos de elevación para zonas forestadas indican valores mayores a las elevaciones reales del suelo. Esto representa un inconveniente para algunos usos, como la simulación de escurrimiento superficial en el modelado de inundaciones.

En zonas de llanuras productivas como es la Provincia de Buenos Aires, las pendientes son suaves y los árboles son plantados por el hombre, generalmente en formas de hileras. Estas líneas de árboles, forman cotas más altas en los DEMs que hacen un efecto de dique en las simulaciones de inundación. Contrariamente, lo que sucede en la realidad, es que el agua escurre por debajo de las copas de los árboles.

Es por ello que las cotas de DEMs en zonas arboladas deben ser corregidas, primero identificando la ubicación de los mismos y luego obteniendo una cota de altura más adecuada para el terreno en base al terreno circundante.

Para la identificación de zonas arboladas, las imágenes multi-espectrales del tipo

Landsat suelen ser habitualmente empleadas. Estas, están también disponibles de forma gratuita al igual que los DEMs de SRTM. Las imágenes Landsat se componen en capas del reflejo de longitudes de onda visibles y no visibles, como aquellas relacionadas a los niveles de clorofila [Lei+12].

Si bien existen algunos indicadores tradicionales de vegetación como el índice de vegetación de diferencias normalizadas (NDVI) de amplia utilización en agronomía de precisión, no están orientados a diferenciar únicamente los árboles de otros tipos de vegetación como cultivos. Trabajos previos en esta línea como el de Thomas, D'Amato y Rinaldi [TDR15] realizan otra combinación de capas Landsat para la identificación, pero los valores de umbrales obtenidos manualmente muchas veces no producen el resultado esperado.

Sacando provecho de las múltiples longitudes de onda contenidas en las imágenes Landsat y los patrones de distribución de los árboles implantados, se propone la investigación de un detector de presencia de árboles. Debido al número de bandas espectrales de las imágenes Landsat, sus combinaciones posibles y los patrones texturales que pueden derivarse, se propone la utilización de métodos de aprendizaje automático [Mel+15]. Estos métodos permiten de forma automática, sin importar la cantidad de características o indicadores, identificar cuáles son las más relevantes o las que mejor separan las distintas clases y la separación a realizar; tarea que sería prácticamente inviable de forma manual.

Para la construcción del clasificador automático basado en clasificación supervisada, se debe contar con un etiquetado sobre las imágenes Landsat realizado por un observador humano. Como el área de estudio es la región pampeana, se utilizarán imágenes correspondientes a la zona.

Entre los clasificadores supervisados se encuentra el Árbol Aleatorio o Random Forest (RF). Este clasificador está basado en el ensamble de un gran número de árboles binarios de clasificación. El RF es un clasificador robusto para espacios de indicadores no linealmente separables, e incluso para los que se desconoce la distribución de estos [Bre01].

El algoritmo para generar un RF busca automáticamente la mejor combinación de indicadores espectrales y texturales, junto a sus umbrales para separar las clases. Esta característica permite derivar del clasificador información sobre cuáles fueron las características más relevantes que utilizó para la clasificación, y los valores que separan una clase de otra [LDL16].

Con el clasificador automático generado, se podrá detectar vegetación arbórea en imágenes de otras regiones. Luego, donde el clasificador detecte árboles, se procederá a la corrección del DEM en los sectores detectados por interpolación de valores de píxeles no arbolados circundantes [TDR15].

Se espera que la aplicación resultante esté integrada a un sistema de información geográfico (GIS), para aprovechar sus librerías para manejo de información geoespacial y su visualización. Entre los GIS extensibles más populares se encuentra QGIS. Este GIS ha ganado popularidad debido a que es extensible mediante el lenguaje de programación Python, que cada vez es más utilizado para software de procesamiento científico, con una gran cantidad librerías disponibles [Che+10].

## 1.2. Objetivos

- Determinación de un conjunto de imágenes satelitales etiquetadas con líneas de vegetación arbórea, según las estaciones del año.
- Detección automática de vegetación arbórea mediante la combinación de imágenes satelitales, el cálculo de indicadores sobre estas y la utilización de un clasificador.
- Corrección automática de modelos digitales de elevación correspondientes a la región pampeana de Argentina en base a la detección de la vegetación arbórea previa.
- Evaluación del resultado obtenido mediante la comparación de la elevación del suelo obtenida de otras formas.

## 1.3. Estado del arte

Entre las investigaciones pertinentes, se encuentra el trabajo final de grado de Amalia L. Thomas [TDR15] dirigida por Pablo Rinaldi y codirigida por Juan D'Amato: “Corrección de Modelos Digitales de Elevación Radar en base a imágenes satelitales multiespectrales mediante procesamiento paralelo”. Amalia, en su trabajo final, detecta vegetación arbórea definiendo umbrales para las reflectancias de las bandas verde, roja, infrarroja cercana e infrarroja de onda corta. El cálculo de la reflectancia representada por cada pixel de las imágenes lo realizó a partir de la información leída del archivo de metadatos del producto Landsat y los encabezados de cada banda [TDR15]. Luego, los umbrales de las bandas los definió ajustando funciones senoidales a una serie de umbrales elegidos manualmente para una zona en diversos momentos del año [TDR15]. Si bien el método propuesto funciona, requiere un análisis manual y no tiene en consideración texturas u otro tipo de información derivada. A su vez, puede presentar una reducción en la calidad de los resultados a medida que aumentan la extensión de las disposiciones de árboles y las pendientes del suelo o la irregularidad de las elevaciones [TDR15].

Algunos autores como Barret Lowe y Arun Kulkarni [LK15], evaluaron el desempeño de métodos de clasificación de píxeles en imágenes multiespectrales usando las primeras siete bandas de imágenes Landsat-8 y clasificadores de máxima verosimilitud, redes neuronales, support vector machines (SVM) y random forest (RF). Por último, compararon sus resultados, obteniendo en promedio la mejor precisión para el RF dada las distintas clases (bosque, agua, suelo, vegetación).

Por otro lado, Özlem Akar y Oğuz Güngör [AG12] también evaluaron el desempeño de RF comparando los resultados obtenidos a partir de Gentle AdaBoost, SVM y clasificación de máxima verosimilitud. Particularmente, utilizaron dos datasets distintos generados a partir de imágenes con cuatro bandas, utilizando aproximadamente misma cantidad de píxeles de cada clase para el entrenamiento. Los mejores resultados los obtuvieron con RF. Entre sus conclusiones mencionan que como investigación futura, se centrarán en integrar la texturas, pendiente y otra información no espectral al método de RF para mejorar aún más su rendimiento en la clasificación de imágenes de satélite.

Dada la disponibilidad de DEMs de SRTM, su uso se ha ido popularizando a través del tiempo. Köthe y Bock [KB09] exponen en su artículo “Preprocesamiento

de Modelos Digitales de Elevación” las consecuencias y los problemas que surgen en cuanto al análisis de terreno; en particular nombran (1) la vegetación al momento de estudiar parámetros morfométricos en la superficie de la tierra, (2) la distinción entre superficie natural y artificial (hecha por el hombre) que para ciertos análisis puede ser de importancia distinguir y (3) la identificación de cuerpos de agua en un modelo digital del terreno. Explican que la estimación de la elevación de la vegetación arbórea necesita la información de la localización (no altura) de ésta y que es posible estimarla a partir de los bordes de los bosques realizando una interpolación; mientras más plano el terreno mejor.

Entre otros autores que tratan el procesamiento de Modelos Digitales de Elevación, se encuentran Gallant [GRD12], quien define el proceso de remover vegetación arbórea como un paso crucial para obtener un modelo de elevación de la tierra desnuda y como uno de los pasos mas técnicamente desafiantes en el pre-procesamiento de datos SRTM. A su vez nombra problemáticas como ríos que son bordeados por distintas arboledas; los ríos aparecen en el SRTM como crestas elevadas en lugar de canales evitando el uso efectivo del DEM para determinados análisis. Gallant propone un método de interpolación basado en mínimos cuadrados para estimar la altura real de la superficie. Algunos artefactos y errores se presentaron en máscaras donde la vegetación no era correcta o los bordes de las zonas no estaban muy bien definidos, o existían cambios abruptos del terreno.

Otros autores, como Reuter [RNJ07] evaluaron distintos métodos de interpolación para el relleno de vacíos o huecos en datos de SRTM. Reuter reconoce que los DEMs derivados de SRTM presentan “vacíos de datos” y otro tipo de anomalías que impiden el uso inmediato en muchas aplicaciones. Afirma a su vez, que existe poca orientación sobre que método es más apropiado dentro la variedad de algoritmos de interpolación existentes. Establece entonces, que la elección del método es dependiente tanto del tamaño como del tipo de terreno correspondiente al vacío. Generaliza en su artículo que método utilizar en qué caso:

- Kriging o Inverse Distance Weighting (IDW) para vacíos pequeños o medianos en áreas planas bajas.
- Interpolación de splines para vacíos pequeños y medianos en terrenos disecionados de gran altitud (terrenos que se han visto gravemente erosionados de modo que su relieve es abrupto).
- Red irregular de triángulos o IDW para grandes huecos en áreas muy planas.
- ANUDEM (un método avanzado de splines) para vacíos extensos en otro tipo de terrenos.

# Capítulo 2

## Imágenes Satelitales

Al momento de trabajar con distintas imágenes sobre la superficie de la Tierra, existen algunos desafíos y consideraciones a tener en cuenta. Los datos geoespaciales pueden presentarse en distintos tipos como por ejemplo rasters, vectores o nubes de puntos. A su vez, estos tipos de datos pueden presentarse en distintos formatos como archivos con distintas extensiones. Las imágenes pertenecientes a una misma zona pueden utilizar un sistema de coordenadas distinto o utilizar distintas proyecciones e incluso pueden tener distintas resoluciones o pueden no estar alineadas entre ellas. Por ejemplo, en el caso de las imágenes Landsat, cada vez más utilizadas para monitoreo ecológico, a menudo se requiere un procesamiento previo al análisis. Es importante tener en cuenta que realizar procesamiento adicional a ciertas imágenes puede introducir ruido y efectos no deseados, sin ganar ningún valor [Ken+09].

A continuación se da una introducción a los distintos tipos de imágenes utilizadas, a ciertos conceptos necesarios para trabajar con estas imágenes y formas de resolver algunos de los inconvenientes que pueden surgir.

### 2.1. GIS y Herramientas para Procesamiento de Imágenes

Un Sistema de Información Geográfica (o GIS por sus siglas en Inglés) es un sistema diseñado para capturar, guardar, manipular, analizar, manejar y presentar datos espaciales o geográficos. Algunas de las consideraciones a tener en cuenta a la hora de elegir un GIS son si:

- Es Open Source y/o de uso gratuito
- Es fácil de extender la funcionalidad
- Tiene una comunidad activa de personas que usen el software
- Es fácil de usar/user-friendly
- Está disponible en múltiples plataformas
- Es capaz de abordar una amplia variedad de problemas
- Es posible la integración con otras herramientas

Mientras ArcGIS ha sido probablemente de los GIS más populares y dominantes [Tia16], existe una alternativa gratuita, Open Surce, multiplataforma y extensible, que es QGIS [GIS15].

A continuación se provee una lista de las herramientas utilizadas y el motivo por el cual fueron elegidas:

- QGIS y GRASS GIS, porque integran herramientas que facilitan el manejo y análisis de los datos y permiten organizar la información y visualizarla de manera sencilla.
- GDAL por permitir manipular los datos geoespaciales y operar con ellos.
- Scikit-learn por ser una librería de Python que contiene algoritmos para realizar clasificación en imágenes.
- OSGeo4W Shell por presentar una forma de configurar el entorno y otorgar la oportunidad de utilizar Python y GDAL directamente desde ahí.
- QGIS Processing Toolbox por incorporar algoritmos geoespaciales complementarios a QGIS a través de distintos proveedores como SAGA o GRASS GIS.
- Plugin Builder y Plugin Reloader, por facilitar la tarea a la hora de desarrollar un plugin/complemento para QGIS.
- Profile Tool, por generar perfiles del terreno y permitir el análisis de las imágenes.
- World Imagery, por proveer imágenes de alta resolución dentro del canvas en QGIS evitando la descarga de otro tipo de imágenes o el uso de otras herramientas aparte como Google Earth.

En el apéndice (Sección A.1), se encuentra una breve introducción sobre las funcionalidades básicas de las herramientas, relaciones entre ellas, ventajas o desventajas y aquello que se encontró más relevante/interesante en cuanto a cada una.

## 2.2. Sistema de Referencia de Coordenadas

Para definir la posición de objetos en la Tierra, es necesario un sistema de coordenadas que se adapte a la forma de la Tierra. Un Sistema de Referencia de Coordenadas (CRS por sus siglas en inglés) permite la construcción de un espacio bidimensional (como un mapa) a partir de un espacio tridimensional (la tierra). Un CRS geográfico se compone de:

- Sistema de Coordenadas (SC): Una secuencia de ejes de coordenadas con unidades de medida específicas [Oil12]; es un sistema que utiliza uno o más números (o coordenadas) para determinar la posición de puntos u otros elementos geométricos en un espacio. Un SC es un concepto matemático abstracto sin ninguna relación definida con la tierra. Los sistemas de coordenadas generalmente no se han descrito explícitamente en la literatura geodésica, y rara vez tienen nombres bien conocidos por los cuales se identifican [Oil12].
- Datum: La Tierra tiene una forma aproximada de elipsoide. Un datum consiste en una serie de números que definen la forma y el tamaño del elipsoide y su orientación en el espacio. Se elige un datum para dar el mejor ajuste posible a la verdadera forma de la Tierra. Algunos datums son precisos solo en ciertos lugares de la Tierra, ya que el elipsoide puede ser muy parecido a la superficie

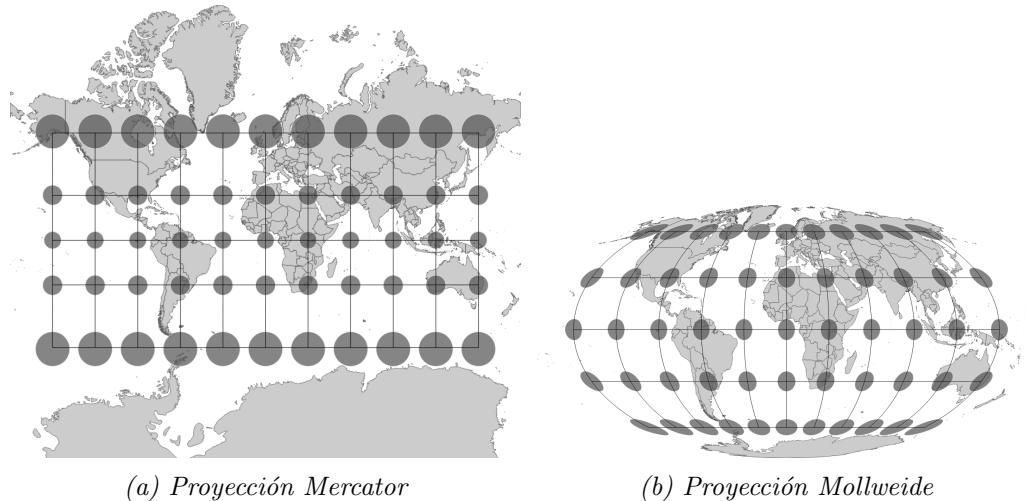
real allí, pero puede estar muy desfasado para otros lugares. Algunos ejemplos son “New Zealand Geodetic Datum 1949” utilizado en Nueva Zelanda o “WGS-84” utilizado a nivel global.

## 2.3. Proyecciones

Un sistema de referencia de coordenadas que se basa en un CRS geográfico usa una proyección para convertir las coordenadas en un plano. La distorsión que es inherente al proceso es cuidadosamente controlada y conocida. Un CRS geográfico puede servir como base para muchos CRS proyectados.

Existen distintos tipos de distorsiones en base a la representación de las formas, el área, las distancias y las direcciones. Hay muchos tipos de proyecciones, y algunos obtienen mejores resultados para ciertas distorsiones.

Por ejemplo, en la proyección Mercator se conservan las formas (Figura 2.1a), pero el área crece a medida que uno se acerca a los polos. Puede observarse que Groenlandia parece ser del mismo tamaño que África, mientras que en realidad África es unas 14 veces más grande. En la proyección Mollweide (Figura 2.1b), el área se conserva pero las formas están distorsionadas.



*Figura 2.1: Diferencias entre proyecciones. Sobre los mapas puede verse la Indicatriz de Tissot, la cual se usa para ilustrar gráficamente las distorsiones lineales, angulares y de área. En QGIS puede generarse con el plugin Indicatrix Mapper. Mercator conserva formas mientras Mollweide conserva el área.*

Por estas razones, al momento de trabajar con distintos datos georeferenciados, es importante comprender que existen múltiples CRSs y que si se tienen datos de la misma ubicación en diferentes CRSs, los datos no estarán alineados. Por lo tanto, al realizar análisis y procesamiento, se querrá tener los datos en la misma proyección. Algunos GIS, como en el caso de QGIS, tienen una funcionalidad llamada “On The Fly Projection” o “Proyección Sobre la Marcha”. Esto significa que se puede definir una cierta proyección cuando se inicia el GIS y todos los capas que son añadidas luego, sin importar qué sistema de referencia de coordenadas tengan, se mostrarán automáticamente en la proyección que se definió. Esta funcionalidad permite superponer capas dentro de la vista del mapa, aunque puedan estar en diferentes sistemas de referencia.

GRASS GIS tiene un enfoque distinto. De acuerdo con lo establecido en su wiki [Wik17], “*Por razones de calidad, GRASS GIS maneja una proyección por ubicación. Se puede reproyectar entre ubicaciones y reproyectar durante la importación de datos si es necesario*”. La idea es que el proceso no se realice ciegamente sin una elección apropiada de los métodos.

## 2.4. Imágenes Landsat

Las misiones Landsat son una serie de satélites construidos y puestos en órbita por EEUU para la observación en alta resolución de la superficie terrestre. El Landsat 1 fue lanzado en 1972 y el más reciente, el Landsat 8, en el 2013. Este último tiene un ciclo de repetición de 16 días. Los dos sensores que transporta completan 11 bandas espectrales que corresponden a distintas longitudes de onda (Cuadro 2.1).

Banda	Descripción	Rango de Onda	Resolución
Banda 1	Coastal / Aerosol	(0.43 - 0.45 µm)	30 m
Banda 2	Visible Azul	(0.45 - 0.51 µm)	30 m
Banda 3	Visible Verde	(0.53 - 0.59 µm)	30 m
Banda 4	Visible Rojo	(0.64 - 0.67 µm)	30 m
Banda 5	Near-Infrared	(0.85 - 0.88 µm)	30 m
Banda 6	SWIR 1	(1.57 - 1.65 µm)	30 m
Banda 7	SWIR 2	(2.11 - 2.29 µm)	30 m
Banda 8	Panchromatic (PAN)	(0.50 - 0.68 µm)	15 m
Banda 9	Cirrus	(1.36 - 1.38 µm)	30 m
Banda 10	TIRS 1	(10.6 - 11.19 µm)	100 m
Banda 11	TIRS 2	(11.5 - 12.51 µm)	100 m

*Cuadro 2.1: Bandas espectrales del Landsat 8.*

Las bandas 2, 3 y 4 conforman el espectro visible por el ojo humano y la combinación de estas bandas generan una imagen de aspecto natural. Sin embargo, diferentes longitudes de onda y diferentes combinaciones de bandas, a menudo pueden ayudar a discernir algunas características mejor que otras. Cada banda detecta mejor:

- Banda 1: Aguas poco profundas, partículas finas de polvo.
- Banda 2: Aguas profundas, atmósfera.
- Banda 3: Vegetación.
- Banda 4: Objetos hechos por el hombre, tierra, vegetación.
- Banda 5: Costas, vegetación.
- Banda 6: Penetración de nubes, humedad del suelo y vegetación.
- Banda 7: Mejora la penetración de las nubes, la humedad del suelo y la vegetación.
- Banda 8: Imágenes en blanco y negro, detalles nítidos.
- Banda 9: Nubes cirrus.
- Banda 10: Mapeo térmico, humedad estimada del suelo.

- Banda 11: Mapeo térmico mejorado, humedad estimada del suelo.

Las imágenes son distribuidas como formato GeoTIFF. Un GeoTIFF es un archivo TIFF (archivo que almacena mapas de bits) con una georreferencia embebida, es decir, con un posicionamiento espacial en una localización geográfica única y bien definida por un sistema de coordenadas y datum específicos.

### 2.4.1. Pre-procesamiento

Antes de comenzar ciertos análisis en imágenes Landsat, se debe realizar un pre-procesamiento para tener en cuenta los efectos solares, atmosféricos, topográficos y de los sensores. Como primer paso, es fundamental comprender las unidades asociadas comúnmente a estos datos. Las señales inicialmente tomadas por los sensores son registradas como valores de radiancia. La radiancia es el flujo de energía registrado por el sensor. Los valores entre sensores pueden variar en base a la degradación de cada uno. Estos valores se rescalan a “números digitales” (DN) (enteros sin signo de 12 bits para Landsat 8). Los productos Landsat Level-1 se encuentran en DN pero pueden ser convertidos a radiancia o reflectancia; la reflectancia es una medida sin unidades que expresa la relación de radiación reflejada por un objeto en relación con la radiación incidente sobre el objeto [You+17].

Los pasos del pre-procesamiento pueden ser de tipo geométrico, absoluto o relativo; algunos pasos convierten de una unidad a otra mientras que otros tratan otro tipos de correcciones como son las topográficas.

Los productos Landsat Level-1 son geométricamente corregidos a través de procesos que involucran puntos de control en la tierra y modelos de elevación digitales. Los productos Tier 1 son los más adecuados para análisis de series temporales a nivel de píxel [USGb], mientras que los Tier 2 y Real-Time pueden no ser lo suficientemente precisos para algunos tipos de análisis.

Las correcciones absolutas son necesarias para obtener valores comparables. Las imágenes que pasan por las mismas correcciones absolutas pueden llegar a ser comparadas (a través del tiempo, espacio o sensores). Las imágenes en DN no pueden ser comparadas a través del tiempo, dada la degradación de sensores y diferencias entre distinto tipo de sensores. Lo que se hace, es convertir los DN a valores de radiancia mediante un proceso lineal obteniendo las variables necesarias desde los metadatos de las imágenes. Rara vez se utiliza esta conversión aislada; sino que se realiza luego una corrección solar que implica obtener la reflectancia a partir de la radiancia incorporando los efectos del sol, que pueden variar en base a la fecha y lugar donde fue tomada la imagen. La corrección solar, no tiene en cuenta pendientes y elevaciones en el terreno que pueden generar variaciones en la reflectancia, generando la necesidad de realizar correcciones topográficas. A su vez, la absorción y dispersión de la radiación afectan la calidad de las imágenes y son la causa principal para generar correcciones atmosféricas.<sup>1</sup>

Como se puede ver, el pre-procesamiento de imágenes no es un trabajo fácil. Afortunadamente, USGS dispone de “Landsat Surface Reflectance Data”, el cual es un producto que provee un estimado de la reflectancia de la superficie como si fuera medida directamente desde el suelo, sin los efectos de absorción y dispersión provocados en la atmósfera, la iluminación o geometría de observación.

---

<sup>1</sup>Para una introducción un poco más extensa sobre el pre-procesamiento de imágenes Landsat se recomienda consultar [You+17].

### 2.4.2. Landsat Surface Reflectance Data

La United States Geological Survey (USGS) es una agencia científica perteneciente a Estados Unidos, que “proporciona ciencia sobre los peligros naturales que amenazan la vida y los medios de vida, el agua, la energía, los minerales y otros recursos naturales, la salud de los ecosistemas y el medio ambiente, y los impactos del cambio climático y el uso del suelo” [USGa]. Desde su página: <https://earthexplorer.usgs.gov/>, es posible acceder a múltiples datasets. Entre ellos se encuentran el servicio on-demand para Landsat 8 de “Surface Reflectance” que remueve los artefactos no deseados, aumentando la consistencia y comparabilidad de imágenes tomadas en distintos momentos. Muchos productos como índices de vegetación, albedo, Leaf Area Index (LAI), áreas quemadas y otros cambios en la cobertura del suelo, se basan en la reflectancia de la superficie. Este producto se encuentra dentro de los “Level-2 Science Products” y se lo considera un “High-Level product”.

La mayoría de las imágenes Landsat 8 de Level-1 son procesadas a Surface Reflectance, aunque algunas consideraciones a tener en cuenta son [USGd]:

- La reflectancia de la superficie no es procesada para imágenes que poseen un ángulo solar de mas de 76°.
- Se advierte a los usuarios contra el procesamiento de imágenes adquiridas en latitudes altas ( $>65^\circ$ ).
- La eficacia de la reflectancia se verá reducida en áreas donde la corrección atmosférica se vea afectada por condiciones adversas:
  - Regiones hiperáridas o cubiertas de nieve
  - Condiciones con bajo ángulo solar
  - Regiones costeras donde el área de la tierra es pequeña en relación con el agua adyacente.
  - Áreas con gran contaminación de nubes.

## 2.5. DEM, DSM y DTM

Existen algunas diferencias en la bibliografía en cuanto a la definición de estos términos. Según especifica USGS en [Hei18]:

- DEM (Digital Elevation Model): En general, un DEM se compone de valores de altura representando la tierra desnuda, a intervalos regulares de direcciones en x e y. Sin embargo, el espaciado de la grilla, datum, sistema de coordenadas, formatos de los datos y otras características pueden variar ampliamente.

Según USGS, un DEM es una representación cartográfica digital de la elevación de la tierra a intervalos regulares en direcciones de x e y, utilizando valores z de altura. La mayoría de los proveedores de datos utilizan DEM como un término genérico para DSMs y DTMs.

- DSM (Digital Surface Model): Similar a los DEMs excepto que pueden representar elevaciones de los edificios, árboles, torres y otras características sobre la tierra desnuda.

- DTM (Digital Terrain Model): En ciertos lugares, DTMs y DEMs son utilizados como sinónimos, representando la tierra desnuda con valores de altura uniformemente separados, como en un raster.

En los Estados Unidos, en general DTM es un conjunto de datos vectoriales compuestos por líneas y puntos espaciados irregularmente, que caracteriza la forma del terreno. Un DEM puede derivarse de un DTM.

## 2.6. DEMs derivados de SRTM

La Shuttle Radar Topography Mission (SRTM) fue una misión de la NASA realizada en el 2000 para obtener datos de elevación del terreno. El radar utilizado fue volado 11 días a bordo del Transbordador Espacial Endeavour, el cuál ha volado en total en unas 25 misiones desde 1992 al 2011 luego de ser retirado. Los productos SRTM suelen ser unos de los más utilizados para modelos de elevación digital dado que poseen una cobertura casi global (80 % de la superficie terrestre entre 60°N y 56°S) con resolución relativamente alta (1 arc-second o alrededor de 30 metros) [USGc].

Los radares se pueden clasificar en función de la dirección de su mirada: nadir y “side-looking”. Los radares nadir miran directamente hacia abajo; los side-looking miran lateralmente [EC17]. Los radares de apertura sintética son instrumentos de con una mirada lateral y adquieren datos a lo largo de franjas continuas. Debido a esta técnica de radar de vista lateral utilizada por los SRTM, se producen vacíos en regiones montañosas.

USGS dispone de un producto “SRTM 1 Arc-Second Global” con una cobertura global, donde NGA (National Geospatial-Intelligence Agency) llenó los vacíos utilizando algoritmos de interpolación en combinación con otras fuentes de datos de elevación. Las imágenes son distribuidas en tiles y algunos de estos pueden contener huecos todavía, y aquellos ubicados al norte de 50°N y al sur de 50°S tienen una resolución diferente de 2 arc-second por 1 arc-second. Las imágenes pueden ser descargadas en formato GeoTIFF. Cada celda del mapa de bit representa una altura en metros respecto del nivel del mar.

A pesar de la amplia utilización en muchos campos de estudio, los DEM de SRTM pueden sufrir de “striping” o “bandeado” que genera un efecto de rayas. Este artefacto se debe a un error de movimiento no compensado [TS13] y puede propagarse rápidamente a análisis posteriores que pueden afectar significativamente la usabilidad en varias aplicaciones. Un método basado en la transformada de Fourier es el que ha demostrado buen potencial para la resolución del problema. Otros métodos deben incluir cierto conocimiento sobre el ángulo de striping, el radio de las rayas y las distancias entre ellas. Dentro del Processing Toolbox de QGIS, se encuentran los algoritmos de SAGA, el cual es otro GIS como QGIS, gratuito y open-source, que contiene un conjunto de módulos geocientíficos. Entre los algoritmos de SAGA de la Processing Toolbox, se encuentran “Destriping” y “Destriping with mask” que permitirían remover este artefacto de striping. SAGA Gis, de manera independiente, contiene un módulo `saga_cmd contrib_perego 5` para corregir este efecto de striping.

### 2.6.1. Generación de Mosaicos y Recorte de Rasters

Los DEMs suelen distribuirse en recortes o “tiles”, y algunas veces sucede que las zonas de interés se encuentran en los límites de estos. Lo mismo puede llegar a pasar para imágenes Landsat. Para solucionar este problema, se puede utilizar `gdal_merge`, ya sea desde el OSGeo4W Shell, Python o directamente desde la interfaz de QGIS (**Raster >> Miscellaneous >> Merge**). Esta utilidad permite crear un mosaico a partir de un conjunto de imágenes (Figura 2.2). Todas las imágenes deben estar en el mismo sistema de coordenadas y tener un número coincidente de bandas, pero pueden superponerse y tener diferentes resoluciones. En áreas de superposición, la última imagen se copia sobre las anteriores.

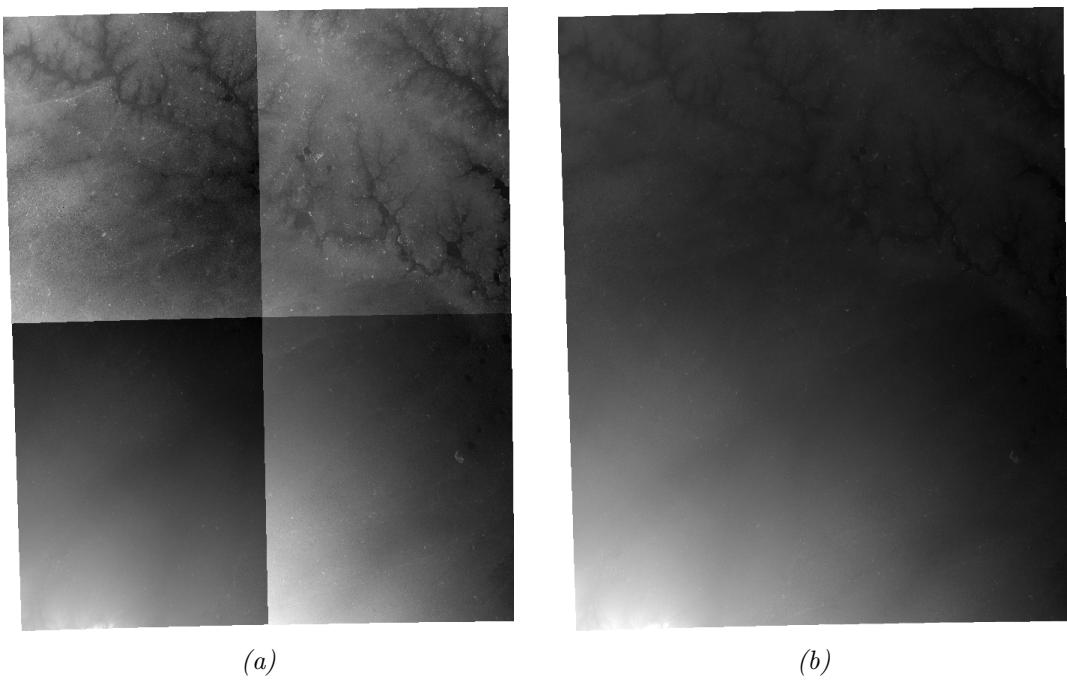


Figura 2.2: Imágenes de elevación luego de aplicar `gdal_merge`. (a) Recortes de DEMs de SRTM. (b) Mosaico generado a partir de recortes de DEMs de SRTM. QGIS automáticamente toma los valores máximos y mínimos de las bandas y escala los colores acorde al rango; es por esto que en la figura (a), los DEMs en realidad tienen valores continuos como muestra (b), pero visualmente no lo pareciera dado el escalado de los colores.

Otras veces, las zonas de interés suelen no ser tan extensas o se espera que la extensión coincida con la de otra imagen. Esto puede lograrse con `gdal_translate` y eligiendo la extensión o tomando la extensión a partir de otra imagen.<sup>2</sup>

### 2.6.2. Alineación de Rasters

Otra consideración a tener en cuenta al combinar el uso de distintos datasets en formato de raster, es si los píxeles se encuentran alineados entre los distintos datasets. Alinear rasters implica:

- Reproyectar los rasters al mismo CRS.

---

<sup>2</sup>En el apéndice (Sección A.2) se incluye un ejemplo.

- Remuestrear al mismo tamaño de píxel/celda considerando los offsets de las grillas.
- Recortar las imágenes a la misma región de interés.

El remuestreo de las imágenes puede lograrse con `gdal_translate`. Como alternativa, QGIS puede realizar el alineado de las imágenes desde **Raster >> Align Raster**, especificando la resolución, extensión y CRS de referencia.

Existen distintos métodos para el remuestreo de las imágenes. Los más comunes según se explican en una guía de ArcGIS [ESRb]:

- Nearest Neighbor: No cambia ningún valor del raster de entrada. Determina el valor a partir de valor de la celda más próxima. Es usado principalmente para datos discretos, como para clasificación de uso del suelo. Es de los métodos más rápidos y el error espacial máximo será la mitad del tamaño de una celda.
- Bilinear: Realiza una interpolación bilineal y determina el nuevo valor de una celda en función del promedio de distancia ponderada de los cuatro centros de celdas de entrada más cercanos. Es útil para datos continuos.
- Cubic: Similar a la interpolación bilineal, pero toma las 16 celdas más cercanas. El tiempo de procesamiento es bastante mayor pero puede ser muy útil para la reducción de ruido.

`gdal_translate` dentro de la opción `-r` cuenta, además de los ya nombrados, con los algoritmos: Cubic Spline, Lanczos, Average y Mode.

En la Figura 2.3 se muestra un ejemplo de alineación de rasters sobre una misma zona. Se superpone una grilla sobre las imágenes para poder notar la diferencia entre la imagen Landsat y el DEM a medida que es reproyectado y remuestreado.

## 2.7. Imágenes LiDAR

Light Detection and Ranging (LiDAR) es una tecnología de teledetección utilizada para obtener DEMs de alta resolución.<sup>3</sup> A través de un láser que emite pulsos de luz y un sensor, se miden las diferencias en los tiempos de retorno y la luz reflejada por los objetos. El tiempo de retorno es traducido en valores de altura. Los pulsos emitidos son reflejados por distintos objetos a medida que viaja hacia el suelo, lo cual genera distintas formas de onda donde es posible identificar la vegetación.

A menudo, las imágenes LiDAR son distribuidas en formato `.las` o `.laz`. Estas imágenes contienen nubes de puntos con coordenadas (X, Y) y un valor de elevación asociada a cada coordenada. Con frecuencia, estas imágenes vienen ya clasificadas con puntos “ground/non-ground” dependiendo de cómo fue reflejada la luz. Otras

---

<sup>3</sup>Una de los aportes más importantes de LiDAR, probablemente sean las contribuciones a la arqueología. Entre los acontecimientos más importantes se encuentra Tikal en Guatemala y Angamuco en México. Tikal son las ruinas de una ciudad Maya (probablemente llamada Yax Mutual) con mas de 60.000 edificaciones, escondida entre la jungla de Guatemala. Con la ayuda de LiDAR se mapearon 2100 km<sup>2</sup> (como referencia, el partido de Tandil tiene 4935 km<sup>2</sup> y General Pueyrredón tiene 1453 km<sup>2</sup>). Se cree que entre el 650 y 800 A.C. vivieron allí entre 10 a 15 millones de personas. Angamuco fue descubierta en 2007 y en el año 2012, utilizando LiDAR, se identificaron más de 39.000 edificios en un territorio de 25 km<sup>2</sup> donde se cree que vivieron más de 100.000 personas entre el año 1000 y 1350.

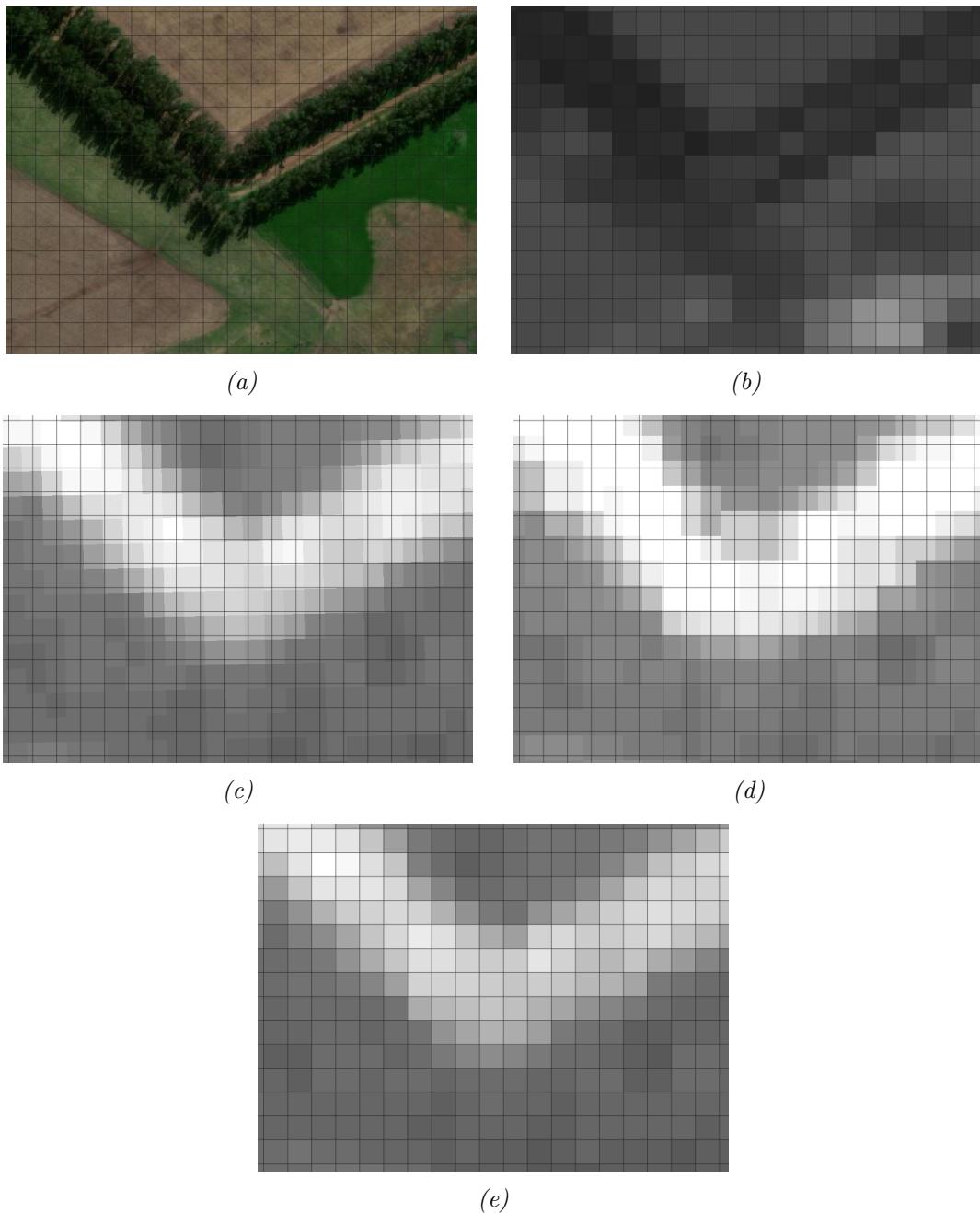


Figura 2.3: Alineado y cambio de resolución. Sobre las imágenes puede verse una grilla alineada respecto de las imágenes Landsat. (a) Imagen de World Imagery de la zona en cuestión. (b) Banda verde de imagen Landsat con CRS en EPSG:32621. (c) DEM con CRS en EPSG:4326. (d) DEM con CRS en EPSG:32621. (e) DEM alineado y con mismo tamaño de píxel respecto de la imagen Landsat.

imágenes pueden estar pre-procesadas y presentar una clasificación más específica identificando distintos tipos de vegetación, cuerpos de agua, edificios, caminos, etc.

Para la visualización de imágenes .las existen algunos visualizadores online como [lidarview.com](http://lidarview.com) y [plas.io](http://plas.io) que funcionan directamente desde el navegador y permiten explorar la nube de puntos. QGIS actualmente no soporta la visualización de .las. De todos modos, dada la inmensa cantidad de puntos que puede tener una de estas imágenes, la nube de puntos suele ser transformada a un formato de raster. Para esto, el manual de entrenamiento de QGIS [QGIb], muestra cómo obtener un DEM a partir de un .las pero la herramienta que utiliza es `las2dem` de “Lastools” y su uso requiere de una licencia. Usar la herramienta sin licencia se puede, pero genera unas líneas diagonales en el DEM de resultado. Como alternativa, GRASS tiene un módulo `r.in.lidar` que rasteriza los puntos.

USGS dispone de imágenes LiDAR a través de [viewer.nationalmap.gov/basic](http://viewer.nationalmap.gov/basic). El “3D Elevation Program” (3DEP), que tiene como objetivo completar la adquisición de datos de elevación de alta resolución en EEUU para 2023, dentro de sus productos dispone de nubes de puntos y DEMs en formato de raster. Los “Source Resolution DEMs” son los DEMs de la tierra desnuda derivados a partir de las nubes. Un detalle a tener en cuenta, es que los DEMs se encuentran en pies. Esto puede convertirse fácilmente a metros con la calculadora de rasters de QGIS (`Raster >> Raster Calculator`) insertando la expresión "`NombreDem@1`" \* 0.3048 donde @1 indica la banda (en este caso los DEMs poseen una sola) y 0.3048 la constante para la conversión.

## 2.8. Imágenes Vectoriales

Frecuentemente se utilizan vectores para almacenar posiciones y atributos asociados a estas. A partir de vértices, se define la forma de los objetos. Existen tres tipos de geometrías: puntos, líneas y polígonos. **Shapefile** es de los formatos más utilizados y consiste de un conjunto de archivos obligatorios que definen las formas y las características de éstas.

Desde QGIS es posible crear capas vectoriales desde `Layer >> Create Layer >> New Shapefile Layer`. Se debe elegir el tipo de geometría, la proyección y se puede definir la tabla de los atributos relacionados a cada forma. Una vez creada la capa y habilitada la edición, se puede agregar las geometrías marcando vértices en el canvas. En el manual de entrenamiento de QGIS se explica con detalle como generar y administrar capas vectoriales [QGIa].

# Capítulo 3

## Método Propuesto

A fin de cumplir con los objetivos propuestos, se diseño un método sistemático que divide el problema en dos tareas principalmente: detección y corrección de árboles. No obstante, antes de realizar estas tareas, es necesario contar con un etiquetado sobre las imágenes Landsat realizado por un observador humano, y preprocesar las imágenes Landsat y los DEMs.

En la Figura 3.1 se observa el diagrama de flujo del método. Los recuadros (a), (b) y (c) representan las tareas previas a la detección de árboles:

- (a) El etiquetado de las imágenes se realizó utilizando imágenes vectoriales en QGIS. Para esto se marcaron polígonos utilizando como referencia las imágenes provistas por World Imagery y seleccionando regiones que presentaban la característica de tener los árboles sembrados de manera contigua.
- (b) Para el preprocesamiento de las imágenes Landsat, dada las regiones seleccionadas, se recortaron las imágenes para hacerlas coincidir con el área de interés y achicar la región computacional.
- (c) Para el preprocesamiento de los DEMs, como estos son descargados en recortes, algunas veces el área de interés puede estar ubicada en los límites de los recortes. En ese caso, es necesario generar un mosaico con los DEMs limítrofes. Una vez que el DEM abarca todo el área de interés, es necesario que el DEM esté alineado con las imágenes Landsat, por las razones mencionadas en la Sección 2.3. Esto implica reproyección, remuestreo y recorte del DEM (Sección 2.6.2). En este caso se utilizó un remuestreo cúbico.

Las tareas de detección y corrección están representadas por los recuadros (d) y (g):

- (d) Dado que distintos objetos poseen distinta firma espectral, es posible identificarlos a través de ellas [LK15]. El RF permite, a través de una serie de características o indicadores, obtener los umbrales que permiten realizar una mejor separación de los datos. Las tareas contenidas dentro del recuadro (d) ((e) y (f)) representan el cálculo de los indicadores de las imágenes Landsat y del DEM. En este capítulo se da una introducción a los RF y se enumeran los indicadores utilizados, al igual que se explica la forma en la que son calculados. Para la clasificación, se utilizaron siempre las bandas espectrales y una combinación de los indicadores, al igual que la máscara generada en (a) que determina que píxeles corresponden a árbol/no-árbol.

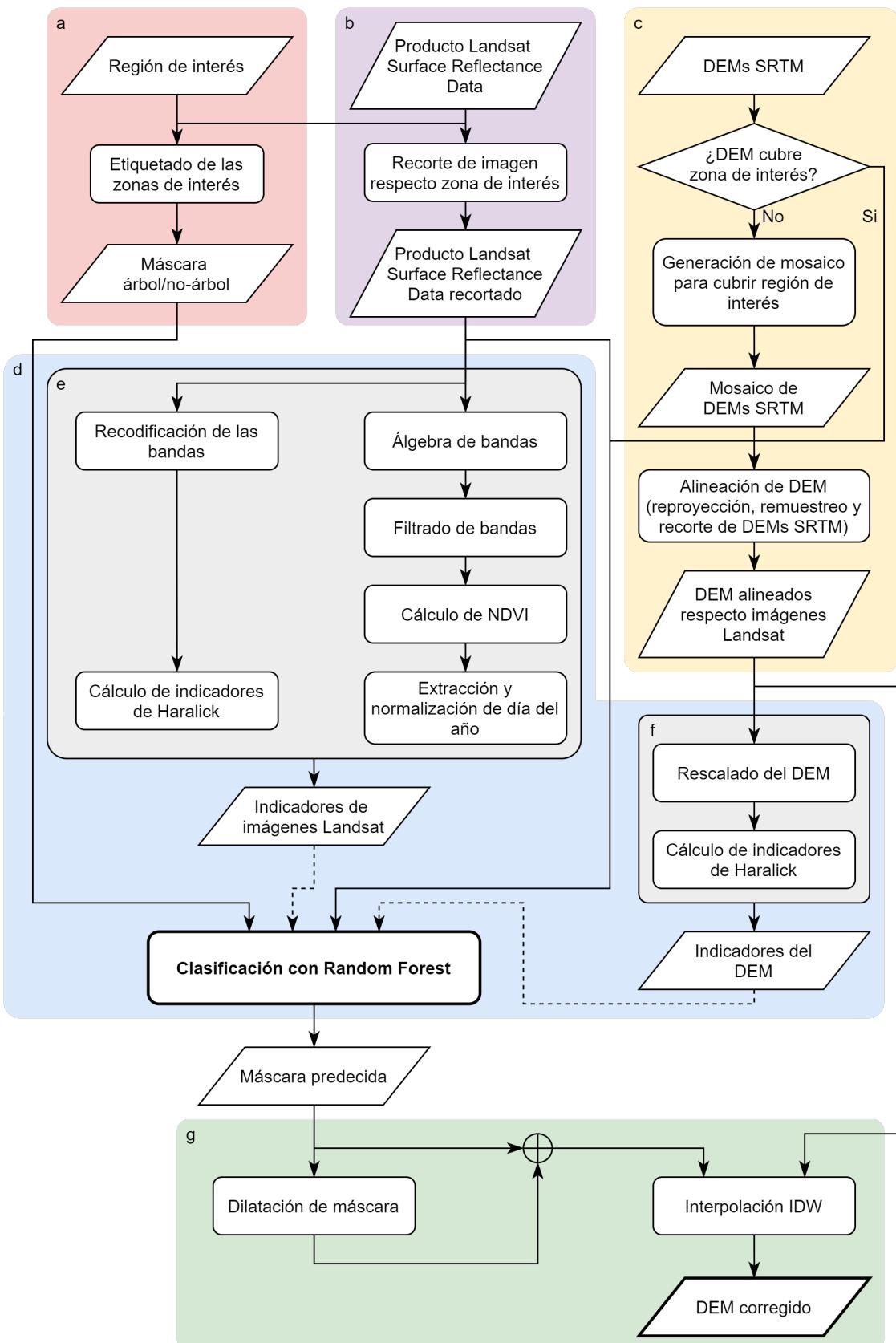


Figura 3.1: Diagrama de flujo del método propuesto. (a) representa el etiquetado manual de los datos, (b) el preprocessamiento de las imágenes Landsat, (c) el preprocessamiento del DEM, (d) la detección de árboles que incluye las tareas (e) y (f) que representan el cálculo de indicadores de las imágenes Landsat y el DEM, y (g) que define la corrección de los árboles en el DEM.

- (g) Para eliminar la altura de los árboles del DEM y así corregirlos, se utiliza la máscara resultante de la clasificación en la tarea anterior. En este caso, se utilizó un método de interpolación denominado IDW (que se menciona en la Sección 1.3 y se explica con más detalle en este capítulo). Como las máscaras obtenidas de la clasificación deben llegar hasta los bordes de los bosques para interpolar a partir de la altura del suelo, se dilataron las máscaras algunos píxeles para que recubran toda la zona de árboles en el DEM.

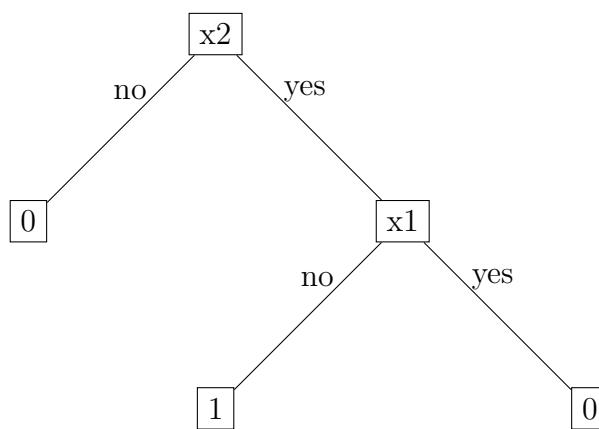
### 3.1. Detección de Árboles

Una parte importante del aprendizaje automático corresponde al problema de la clasificación; al proceso de predecir la clase a la que pertenece una observación. Dentro de los algoritmos para clasificación se encuentran los Random Forest (RF). Algunos autores ([LK15] y [AG12]) estudiaron el análisis de imágenes multiespectrales utilizando RF respecto de otros tipos de clasificación, obteniendo los mejores resultados de accuracy con este método.

Entre las ventajas más significativas de RF se encuentran: precisión sin igual entre los algoritmos tradicionales, implementación eficiente en conjuntos grandes de datos, y una estructura fácil de almacenar para el uso futuro de árboles pre-generados [GPG10]. A su vez, el algoritmo de clasificación RF es superior a muchos algoritmos basados en árboles dado que no es sensible a ruido y no está sujeto a overfitting [WL08].

#### 3.1.1. Random Forest

Un RF se conforma por múltiples árboles de decisión. Un árbol de decisión está formado por nodos que denotan un test o una comprobación sobre un atributo. Cada rama que sale del nodo, representa una salida del test y los nodos hoja representan una clase. Un formato general de un árbol de decisión tiene el siguiente aspecto:



En este caso,  $x1$  y  $x2$  podrían estar representando comprobaciones como: ¿Es el valor de este píxel para la banda verde mayor a 0.8? ¿Corresponde este píxel a una imagen tomada en otoño? Los árboles de decisión también pueden ser n-arios donde cada rama representa un rango de valores. Las hojas definirían si el píxel pertenece a la clase 0 o 1, es decir, si es árbol o no.

Para la construcción de un árbol de decisión, como primer paso, primero se genera un dataset a partir del remuestreo del dataset original. Este proceso se lo denomina como “**bootstrap aggregating**” (bagging) o agregación de bootstrap y consiste en seleccionar muestras de manera aleatoria del dataset original, donde se permite seleccionar a la misma muestra más de una vez. Entonces, si  $A_{m,n}$  define una matriz de  $m$  muestras y  $n$  características, en este paso se obtiene una nueva matriz  $B_{m,n}$  generada a partir de la selección aleatoria de  $m$  filas de  $A_{m,n}$  que pueden ser seleccionadas más de una vez. La idea principal de la agregación bootstrap es mejorar la estabilidad del modelo, reduciendo la varianza y aumentando la capacidad de predicción, evitando el overfitting.

Como segundo paso, una vez generada la re-muestra de bootstrap, se elige un subconjunto aleatorio de las variables o indicadores que definen a las muestras. Por lo general, se consideran  $\sqrt{n}$  indicadores. Es decir, de  $B_{m,n}$  solo se consideraran  $\sqrt{n}$  columnas. Luego, a partir del subconjunto de indicadores elegidos, se toma aquel que mejor separa las muestras y se utiliza como nodo para la construcción del árbol. Para esto se debe medir la impureza del nodo. Un nodo que es totalmente impuro, es un nodo que divide los datos de manera equivalente entre las clases, y es totalmente puro, cuando todos los datos pertenecen a la misma clase. Una alternativa es calcular la “impureza Gini” para determinar el indicador a elegir como nodo (que es la medida utilizada por el RF de Scikit-learn por defecto). Otra alternativa puede ser utilizar la entropía en vez de Gini. En general estas medidas de impureza se usan de manera intercambiable. Según Raileanu y Stoffel en [RS04], solo en el 2 % de los casos importa realmente si se utiliza Gini o entropía y la entropía suele ser más compleja de calcular computacionalmente. La impureza Gini se calcula como:

$$I_G(x) = 1 - \sum_{i=1}^J p_i^2 \quad (3.1)$$

donde  $J$  es la cantidad de clases, y  $p_i$  la fracción de ítems con la clase  $i$  en el set. En la construcción del modelo, la idea es alcanzar una puridad máxima y evitar la impureza; maximizar la ganancia de Gini. El proceso de seleccionar el indicador que mejor divide las muestras se repite nodo a nodo hasta obtener un árbol. Una vez obtenido un árbol, a partir de múltiples remuestreos se genera una variedad de árboles que conformaran el RF.

Para medir el desempeño del RF, se utilizan las muestras que no fueron utilizadas al realizar el bagging; mientras algunas muestras fueron seleccionadas más de una vez, algunas no fueron seleccionadas nunca. Estas muestras se denominan “muestras OOB” (Out of Bag samples) y se evalúan en todos los árboles. Partiendo de la raíz, se avanza por los nodos de decisión hasta alcanzar un nodo hoja que representa una clase. La clase a la que pertenece la muestra queda definida por lo que determina la mayoría de los árboles. La proporción de muestras OOB que son correctamente clasificadas definen así la “accuracy” o el desempeño del RF.

### 3.1.1.1. Importancia de los Indicadores

Como algunos indicadores separan mejor las muestras que otros, existe un método para determinar la importancia de estos. El método consiste en calcular la ganancia de Gini substrayendo la impureza ponderada de las ramas respecto de la impureza del nodo. Luego, la importancia de los indicadores se calcula como la disminución de la impureza del nodo ponderado por la probabilidad de llegar a ese

nodo:

$$\text{Feature Importance} = \frac{N_t}{N} \cdot (I_G - \frac{N_{tR}}{N_t} \cdot I_{GR} - \frac{N_{tL}}{N_t} \cdot I_{GL}) \quad (3.2)$$

donde  $N$  representa el número total de ítems,  $N_t$  representa el número de ítems en el nodo,  $N_{tR}$  y  $N_{tL}$  el número de ítems en el nodo derecho e izquierdo respectivamente.

Resumiendo, se puede decir que en cada división de un nodo en un árbol, la mejora en el criterio de división es la importancia asociada al indicador que realiza la división y en un RF, se acumula sobre todos los árboles para cada variable.

Por último, otra forma para medir la importancia de los indicadores, es realizando permutaciones en un indicador en las muestras OOB y calculando la accuracy nuevamente. Hastie [HTF01] muestra con un ejemplo como los rankings en los indicadores de los dos métodos son similares, pero las importancias en el caso del método de permutación son más uniformes.

### 3.1.2. Métricas

Una tarea fundamental en el proceso de construcción de un modelo de aprendizaje automático, es la evaluación del desempeño o *performance* del modelo. Para poder comparar los distintos modelos existen métricas que cuantifican el desempeño. Para la evaluación de un algoritmo de clasificación supervisado como lo es el RF, se utiliza una matriz de confusión que permite visualizar los errores del modelo.

En este caso en particular, interesa determinar si las máscaras generadas por la clasificación, predicen correctamente el valor real. La matriz de confusión se muestra en el Cuadro 3.1 y facilita ver si el sistema está confundiendo las clases. En las filas se encuentra las instancias de la clase real, mientras que en las columnas, al número de instancias predecidas. Cada matriz está formada por la cantidad de:

- Verdaderos positivos (*TP*): cantidad de instancias que el modelo predice correctamente como positivas.
- Falso negativos (*FN*): cantidad de instancias donde el modelo indica incorrectamente la no-presencia de una condición cuando realmente está presente.
- Falso positivos (*FP*): cantidad de instancias donde el modelo indica la presencia de una condición que realmente no está presente.
- Verdaderos negativos (*TN*): cantidad de instancias que el modelo predice correctamente como negativas.

	Predicción = Árbol	Predicción = No Árbol
Valor Real = Árbol	True Positive (TP)	False Negative (FN)
Valor Real = No Árbol	False Positive (FP)	True Negative (TN)

Cuadro 3.1: Matriz de confusión.

A partir de la matriz de confusión, se derivan las métricas:

- Accuracy: Representa la proporción de clasificaciones correctas. Se calcula como:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.3)$$

- Precision (o Positive Prediction Value): Se interpreta como la capacidad del clasificador de no etiquetar como positiva una muestra que es negativa. Expresa la proporción de puntos que el modelo dice que fueron relevantes. El mejor valor es 1 y el peor valor es 0.

$$Precision = \frac{TP}{TP + FP} \quad (3.4)$$

- Recall (o Sensitivity): Representa la capacidad del clasificador para encontrar todas las muestras positivas o la habilidad de encontrar todas las instancias relevantes. El mejor valor es 1 y el peor valor es 0.

$$Recall = \frac{TP}{TP + FN} \quad (3.5)$$

- F1 Score: Se puede interpretar como un promedio entre precision y recall, pero en vez de utilizar simplemente la media, se utiliza la media armónica para penalizar mejor los valores extremos.  $F1$  le da el mismo peso a ambas métricas. El mejor valor se obtiene cuando  $F1$  se acerca a 1, y el peor, cuando se acerca a 0.

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (3.6)$$

### 3.1.3. Indicadores

Los indicadores tomados en cuenta para la clasificación fueron:

- **Bandas espectrales 1 a 7:** aquellas con resolución de 30m que vienen preprocesadas a Surface Reflectance (pueden verse en el Cuadro 2.1).
- **Álgebra de bandas y fitted line de la firma espectral:** estos indicadores fueron calculados a partir de las bandas de la firma espectral, seleccionando el mismo píxel en cada banda y calculando el promedio, desvío estándar y la regresión lineal (pendiente y ordenada al origen de la fitted line) entre la reflectancia o “tono de gris” del píxel en cada banda. El cálculo se logra leyendo las imágenes como arreglos de números y procesándolos con NumPy.<sup>1</sup>
- **Bandas con filtro de media y filtro gaussiano:** SciPy es una librería de Python que contiene un paquete (`scipy.ndimage`) para procesamiento de imágenes el cual incluye numerosos filtros; entre ellos `scipy.ndimage.gaussian_filter` y `scipy.ndimage.median_filter`. Análogo al caso anterior, las bandas 1 a 7 se leen como arreglos de números y se procesan con estos dos filtros.<sup>1</sup>
- **NDVI:** El Índice de Vegetación de Diferencia Normalizada (o NDVI por sus siglas en inglés) es un indicador estandarizado que es usado comúnmente en teledetección para detectar vegetación sana. Se calcula como:

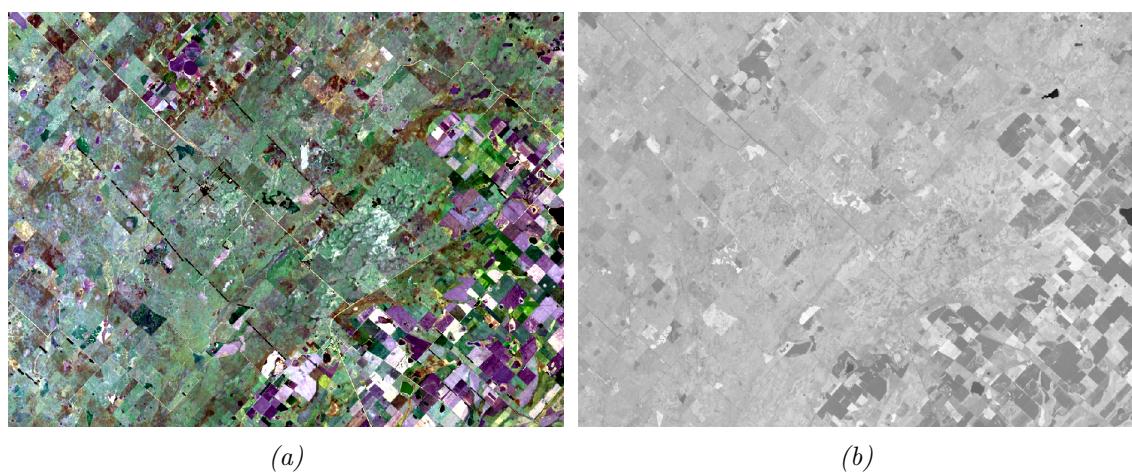
$$NDVI = \frac{NIR - Red}{NIR + Red} \quad (3.7)$$

---

<sup>1</sup>En el apéndice (Sección A.3) se incluye un script en Python de ejemplo que muestra el cálculo del NDVI con GDAL y NumPy. El proceso es análogo para el cálculo de los filtros o para realizar otro tipo de álgebra con las bandas.

y sus valores van de -1 a 1. Los valores negativos de NDVI indican muy probablemente la presencia de cuerpos de agua, mientras que los valores cercanos al límite superior indican alta densidad de hojas verdes, dado que la presencia de clorofila refleja mejor NIR y verdes, y absorbe más rojos y azules respecto de otras longitudes de onda. En la Figura 3.2 puede verse un ejemplo visual del NDVI.

El NDVI puede calcularse fácilmente operando con los arreglos que provee NumPy.<sup>1</sup> La diferencia con utilizar las listas que provee Python, es que el array de NumPy es una “grilla” con valores del mismo tipo, mientras que las listas de Python son redimensionables y pueden contener elementos de diferentes tipos. Mientras las listas pueden ser muy útiles, hay que tener en cuenta que la funcionalidad la tienen a expensas de más espacio y tiempos mayores de procesamiento.



*Figura 3.2: Comparación de imágenes Landsat por la zona de Chaves, provincia de Buenos Aires. (a) Imagen en color natural. (b) NDVI resultante de la zona en cuestión.*

- **Indicadores de texturas de Haralick por banda y sobre el DEM:** Las texturas de Haralick son utilizadas para cuantificar una imagen basada en los patrones y colores que posee. El concepto fundamental involucrado en el cálculo de las texturas, es la matriz de co-ocurrencia (Gray Level Co-occurrence Matrix o GLCM). La idea básica de una GLCM es buscar pares de píxeles adyacentes que ocurren en una imagen y registrarlos en la GLCM. De las matrices de GLCM, pueden calcularse 14 indicadores [HSD73] pero como algunos de ellos pueden inferirse en base a otros, para este caso se utilizaron 5 de los indicadores:

1. Angular Second Moment - Uniformity (ASM): Mide la uniformidad local de los valores de gris. Cuando los píxeles son muy similares, el valor de ASM es alto.
2. Contrast (CON): Es una medida de la intensidad o variación de gris entre el píxel de referencia y sus vecinos.
3. Sum of Squares - Variance (VAR): Medida de la varianza de los niveles de gris dentro de la ventana móvil.

4. Inverse Difference Moment - Homogeneity (IDM): Mide la distribución de los elementos respecto de la diagonal en la GLCM. En general, mientras la homogeneidad incrementa, el contraste decrece.
5. Entropy (ENT): La entropía representa el grado de desorden en la imagen.

A modo de ejemplo, niveles altos de contraste reflejan grandes diferencias de intensidad en la GLCM y se calcula como:

$$Contrast = \sum_i \sum_j (i - j)^2 \cdot p(i, j) \quad (3.8)$$

donde  $p(i, j)$  es el elemento  $i, j$  de la GLCM normalizada.

Para el cálculo de las texturas, GRASS GIS contiene un módulo `r.texture` capaz de generar los indicadores a partir de un raster.<sup>2</sup>

- **Día del año normalizado y día del año transformado:** Las imágenes Landsat vienen acompañadas de un archivo con el sufijo `MTL.txt` donde se encuentran los metadatos. Entre ellos, se puede obtener la fecha en la que fue adquirida una imagen. Dado que los días del año poseen una estructura cíclica, el día es transformado; de esta forma los días de enero se encuentran más próximos a los de diciembre. Los dos indicadores se calcularon como:

- Día del año normalizado:

$$diaN = \frac{dia}{366} \quad (3.9)$$

- Día del año transformado:

$$diaT = \sin \left( \frac{dia}{366} \pi \right) \quad (3.10)$$

### 3.1.3.1. Texturas de Haralick: Re-escalado vs Recodificado

El módulo `r.texture` de GRASS GIS permite generar indicadores de texturas. Para el cálculo se generan las matrices GLCM en cuatro orientaciones distintas y, por defecto, se promedian los resultados. El análisis requiere que las imágenes sean cuantificadas a un número dado de niveles de grises [HSD73]. `r.texture` asume niveles de gris dentro del rango (0, 255) y la entrada es re-escalada automáticamente si su rango está por fuera de los límites. Para reducir el ruido de los datos de entrada (por lo tanto reforzar las características de textura) y acelerar el procesamiento, se recomienda “recodificar” la imagen utilizando cuantificación de igual probabilidad [Ant+]. En la Figura 3.3, se muestra el contraste luego de re-escalar y recodificar la imagen Landsat y se observa cómo se resaltan los bordes.

Por otro lado, en la Figura 3.4a, se muestra el DEM de SRTM de la misma zona que el ejemplo anterior, donde se pueden observar varias hileras de árboles. Hacia el este de la imagen en 3.4a, por la zona de las sierras, se observan ciertos puntos más claros que corresponden a arboledas. Al recodificar la imagen y calcular las texturas, se observa cómo se ve afectado el contraste de forma negativa para las zonas de interés. Mientras que en las imágenes Landsat el recodificado genera un

---

<sup>2</sup>En el apéndice (Sección A.4) se encuentra una versión simplificada del script utilizado para la generación de las texturas.

efecto deseado para las texturas, en el caso del DEM no sucede lo mismo. Se debe tener en cuenta también que mientras la imagen Landsat original tiene un rango de (224, 5281), el DEM tiene un rango de (183, 236).

En la Figura 3.5 se muestra lo que ocurre con los histogramas de la imagen Landsat y el DEM luego de ser re-escalado y recodificado. El histograma de la Figura 3.5b se corresponde con la Figura 3.4a, y el histograma de la Figura 3.5a, con la imagen en 3.3a. Los histogramas fueron generados con el módulo `d.histogram` que posee GRASS.

La zona de Chaves que corresponde a las Figuras 3.3 y 3.4 es ideal para mostrar NDVI y contraste en las imágenes Landsat dada la presencia de cuerpos de agua, distintos tipos de suelos, caminos, entre otros, que resaltan estas características. Dentro de todo, el DEM no se ve tan afectado por el recodificado como en otras regiones. En el apéndice se adjuntan dos zonas más (Figuras A.4 y A.5) donde se observa el efecto aún más amplificado. Particularmente, el DEM de la Figura A.4 sufre del artefacto de striping que se menciona en la Sección 2.6.

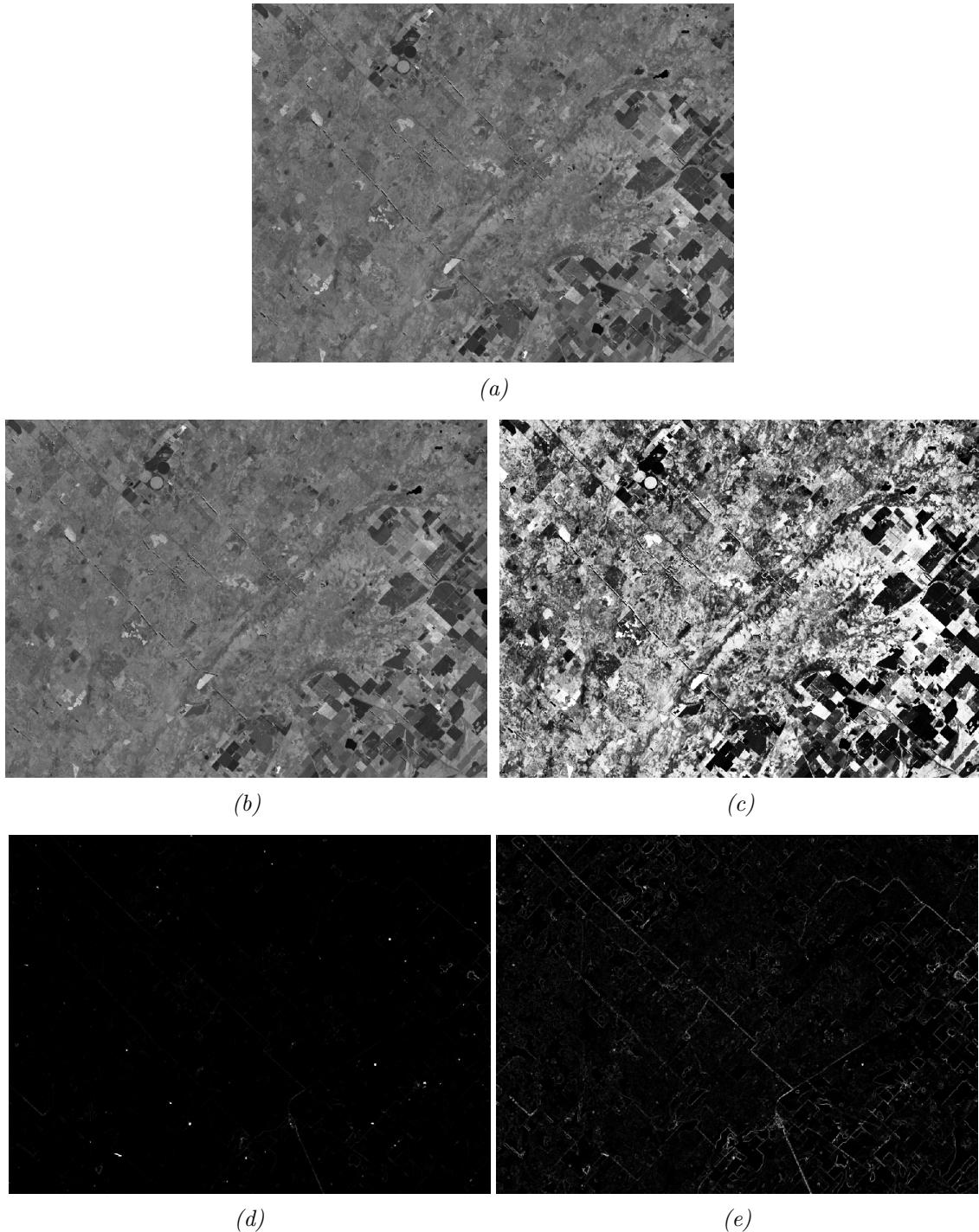


Figura 3.3: Imágenes Landsat correspondientes a la banda 5 (NIR) cerca de Chaves, provincia de Buenos Aires. Se observa cómo recodificar la imagen y calcular las texturas hace resaltar los bordes. (a) Banda 5 original. (b) Banda 5 re-escalada. (c) Banda 5 recodificada. (d) Contraste correspondiente a la imagen re-escalada. (e) Contraste correspondiente a la imagen recodificada.

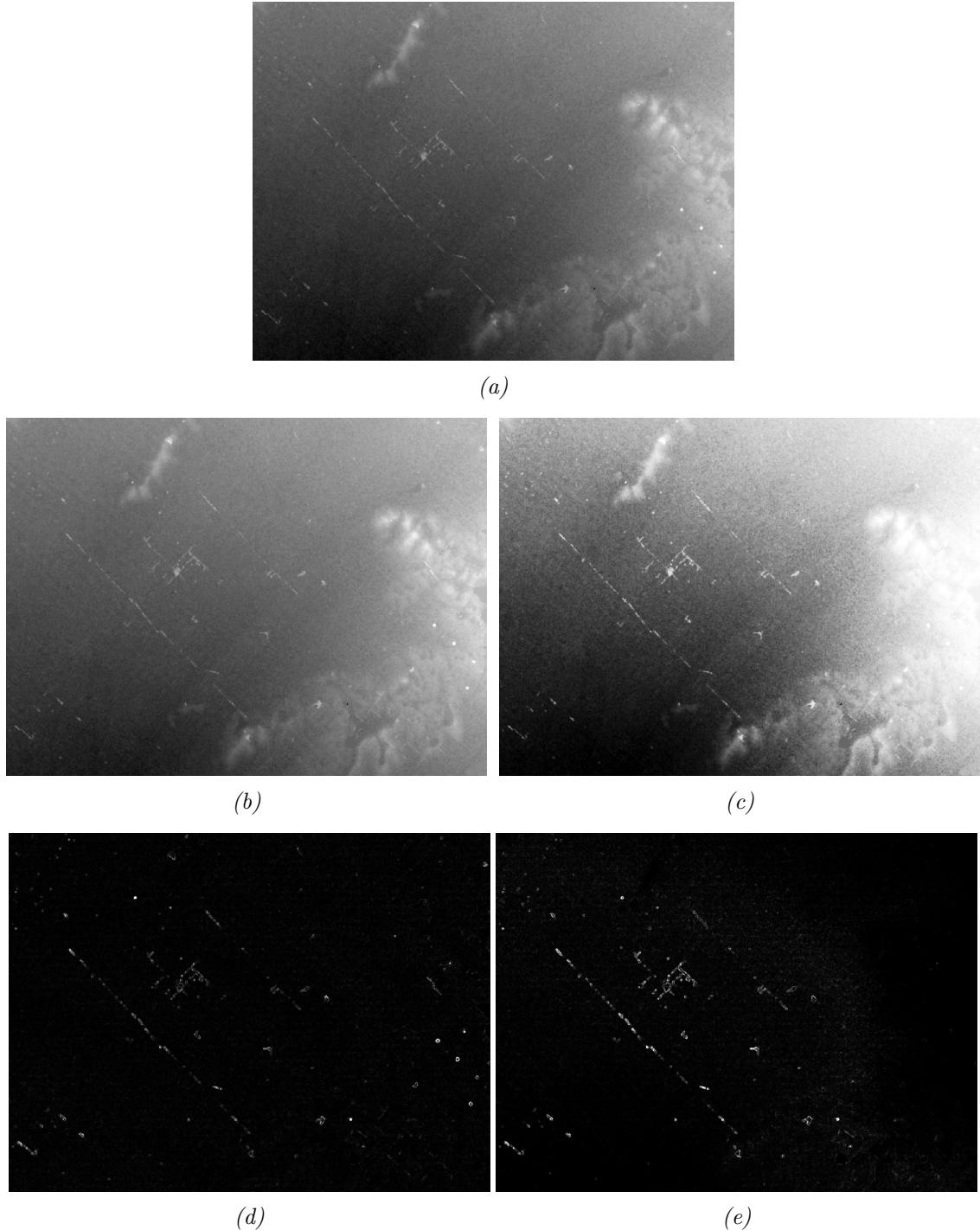
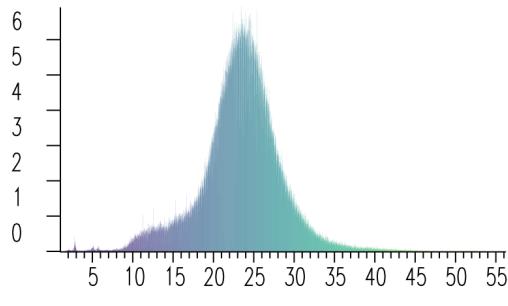


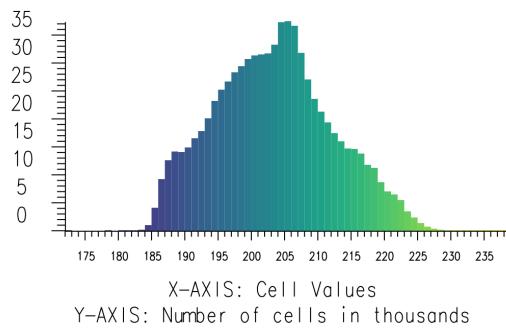
Figura 3.4: DEMs de SRTM correspondientes a una zona cerca de Chaves, provincia de Buenos Aires. Se observa cómo recodificar la imagen y calcular las texturas hace que ciertos bordes correspondientes a zonas de interés no se vean resaltados. (a) DEM original. (b) DEM re-escalado. (c) DEM recodificada. (d) Contraste correspondiente al DEM re-escalado. (e) Contraste correspondiente al DEM recodificado.

.1TP\_225086\_20190608\_20190619\_01\_T1\_sr\_merged.5@PERM



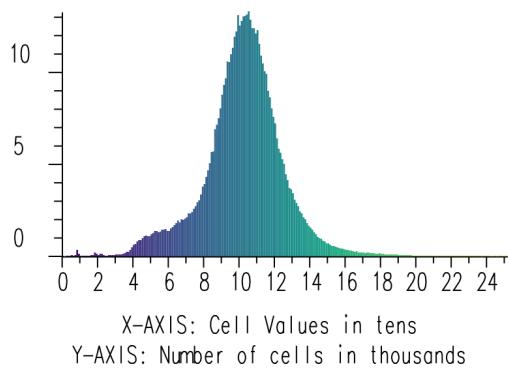
(a)

demChavesB\_dem@PERMANENT



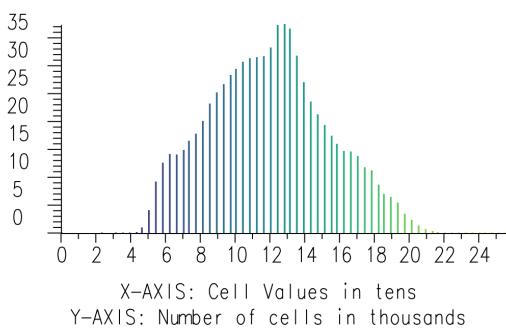
(b)

band5\_rescale@PERMANENT



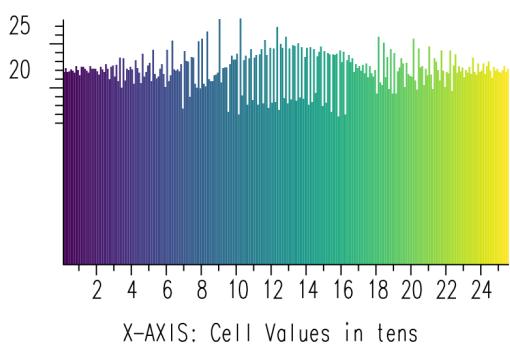
(c)

dem\_chaves\_rescale@PERMANENT



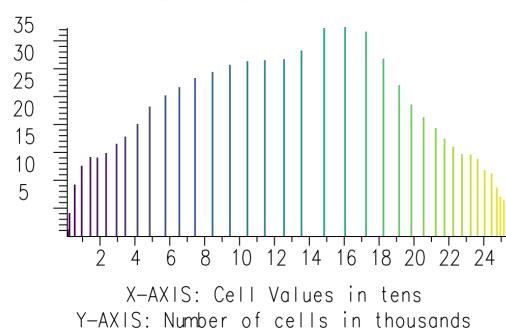
(d)

band5\_recode@PERMANENT



(e)

dem\_chaves\_recode@PERMANENT



(f)

Figura 3.5: Histogramas de imágenes originales, recodificadas y re-escaladas. (a) Banda 5 de imagen Landsat. (b) DEM de SRTM. (c) Banda 5 re-escalada. (d) DEM re-escalado. (e) Banda 5 recodificada. (f) DEM recodificado.

## 3.2. Corrección de Árboles

Como se mencionó en la Sección 2.6, el rellenado de huecos, o la inferencia de un valor a partir de una zona contigua, es algo que puede suceder comúnmente en modelos de elevación. Entre los métodos utilizados de interpolación existe “Inverse Distance Weighting” (IDW). IDW asume que los valores más cercanos al punto que se desea inferir son más relevantes que aquellos que se encuentran más lejos. A su vez, es posible especificar el radio de búsqueda para definir los puntos que se utilizarán al calcular el valor de la interpolación dado que los puntos más distantes tendrán poca o ninguna influencia en la predicción del valor, y acelerar de esta manera los cálculos.

La fórmula general para encontrar el valor interpolado  $u$  dado un punto  $x$  basado en las muestras  $u_i$  para  $i = 1, \dots, N$  es:

$$u(x) = \left( \sum_{i=1}^n \frac{u_i}{d(x, x_i)^p} \right) \Big/ \left( \sum_{i=1}^n \frac{1}{d(x, x_i)^p} \right) \quad (3.11)$$

donde  $d(x, x_i)$  representa la distancia entre el punto conocido  $x_i$  y el punto desconocido  $x$ , y  $p$  representa un número positivo real conocido como el parámetro potencia. Mientras mayores sean los valores de  $p$ , mayor influencia tienen los valores más cercanos al punto interpolado. Generalmente se utilizan valores de potencia mayores o iguales a 1; cuando  $p = 2$ , el método se conoce como “Inverse Distance Squared Weighted Interpolation” aunque no existe una justificación teórica para preferir este valor sobre otros [ESRa].

GDAL cuenta con un algoritmo (`gdal_fillnodata`) para el rellenado de huecos el cual interpola valores para los píxeles sin-dato (aquellos con el valor 0 dada una máscara). Para cada píxel, se buscan los elementos próximos en cuatro direcciones y se realiza la interpolación utilizando IDW. Una vez interpolados los valores, se puede aplicar un suavizado que corresponde a un filtro de media de 3x3 en los píxeles interpolados para mitigar cualquier tipo de artefacto.

Este algoritmo de GDAL generalmente es adecuado para interpolar regiones faltantes de rasters que varían de forma bastante continua (como en el caso de los modelos de elevación). También es adecuado para llenar huecos pequeños y grietas en imágenes más irregulares (como fotos aéreas) [GDA]. Por lo general no es tan bueno para interpolar un raster a partir de datos de puntos dispersos; en ese caso probablemente sea mejor `gdal_grid` [GDA].

### 3.3. QGIS Python Plugin: Lumberjack

La aplicación para realizar la detección y corrección de árboles se realizó como un complemento de QGIS, al cual se le dió el nombre de “Lumberjack”. Se encuentra disponible en GitHub: [https://github.com/carobrus/lumberjack\\_qgis\\_plugin](https://github.com/carobrus/lumberjack_qgis_plugin). Desde allí puede ser descargado para su utilización o puede consultarse el código fuente. A su vez, en el repositorio (Figura 3.6) se encuentra un archivo README.md el cual lista los requisitos, indica cómo instalar las librerías necesarias, cómo agregar el plugin a QGIS y cómo utilizar la herramienta.

**Repository to hold my engineering thesis > Use of Machine Learning to correct Digital Elevation Models with Landsat images**

44 commits | 4 branches | 0 packages | 0 releases | 1 contributor

Branch: master | New pull request | Find file | Clone or download

**carobrus Updates README.md** | Latest commit 59d1a86 7 minutes ago

<a href="#">_pycache_</a>	Adds a few comments and changes line lengths	5 hours ago
<a href="#">grass_script</a>	Modifies GRASS Script	2 days ago
<a href="#">help</a>	First commit	6 months ago
<a href="#">i18n</a>	First commit	6 months ago
<a href="#">images</a>	Updates README.md	7 minutes ago
...		
<a href="#">resources.py</a>	Code refactoring	3 months ago
<a href="#">resources.qrc</a>	First commit	6 months ago

**README.md**

**Lumberjack**

A QGIS plugin to calculate features of images, classify and remove trees out of elevation maps

**Prerequisites**

- QGIS Desktop 3 with GRASS - A Free and Open Source Geographic Information System
- Python 3
- NumPy
- SciPy
- scikit-learn

**Installation of python libraries** ...

Figura 3.6: Repositorio del plugin

Una vez que se agrega el plugin a QGIS, se puede encontrar su ícono en la barra de herramientas junto con los otros plugins (Figura 3.7a) o se puede acceder desde el menú desplegable de los plugins (Figura 3.7b).

#### 3.3.1. La Interfaz Gráfica

La herramienta cuenta con tres vistas. La primera de ellas (Figura 3.8a) permite hacer la clasificación seleccionando qué indicadores se quieren calcular/utilizar, es decir, permite entrenar un clasificador, testearlo y predecir para otras imágenes generando una máscara con las clases. También es posible importar y exportar los

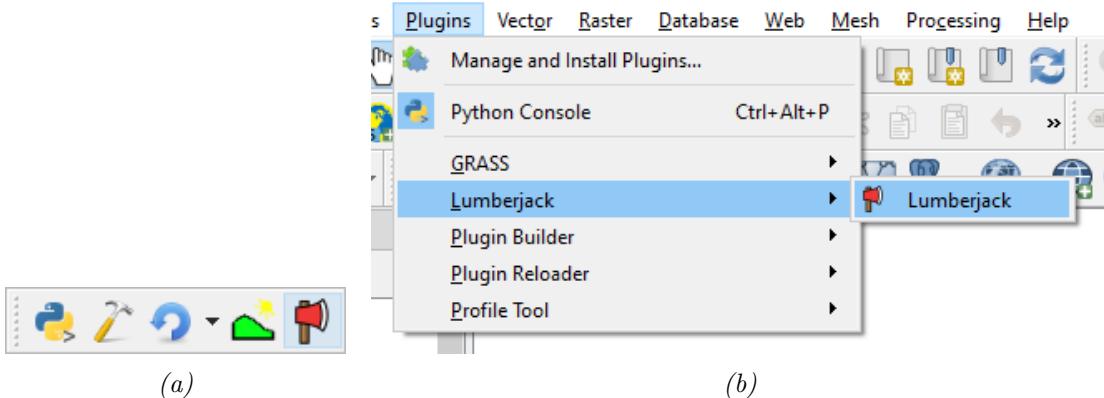


Figura 3.7: Interfaz gráfica de QGIS. (a) Barra de herramientas con plugins: Python Console, Plugin Builder, Plugin Reloader, Profile Tool, Lumberjack. (b) Menú despegable con plugins.

clasificadores, lo cual permite almacenar un clasificador y si se quiere, descartar las imágenes luego de ser utilizadas para el entrenamiento, pudiendo reducir el espacio en disco notablemente.

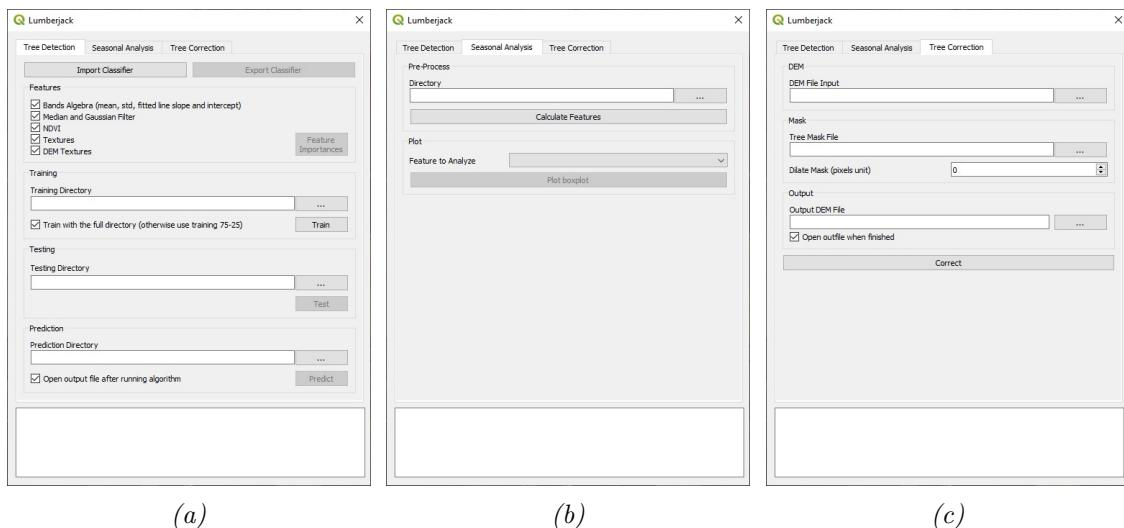


Figura 3.8: Interfaz gráfica del plugin. (a) Clasificación. (b) Estudio del efecto estacional. (c) Corrección de árboles a partir de una máscara.

La segunda de las vistas (Figura 3.8b) permite realizar el estudio del efecto estacional de las imágenes. Dado un indicador, se realiza un gráfico de los valores que definen la máscara en función del tiempo donde cada boxplot está definido por una imagen Landsat. Es posible analizar a través del gráfico los valores de umbrales que definen la máscara de árboles.

La tercera de las vistas, dado un DEM de entrada para ser corregido y dada la máscara que determina donde se encuentran los árboles, permite realizar la corrección de los árboles generando un nuevo archivo como resultado.

Las tres vistas cuentan en la parte inferior con un “plain text edit” de solo lectura que informa las métricas y tiempos de procesamiento.

Por último, en la Figura 3.9 se muestra la importancia de los indicadores que

puede ser ordenada de manera decreciente, o por el orden predeterminado en un principio. Esto se cambia con un botón comutador en la parte superior de la ventana. La Figura 3.10 muestra los boxplots generados que determinan los umbrales que definen la máscara de entrada, dado cierto indicador.

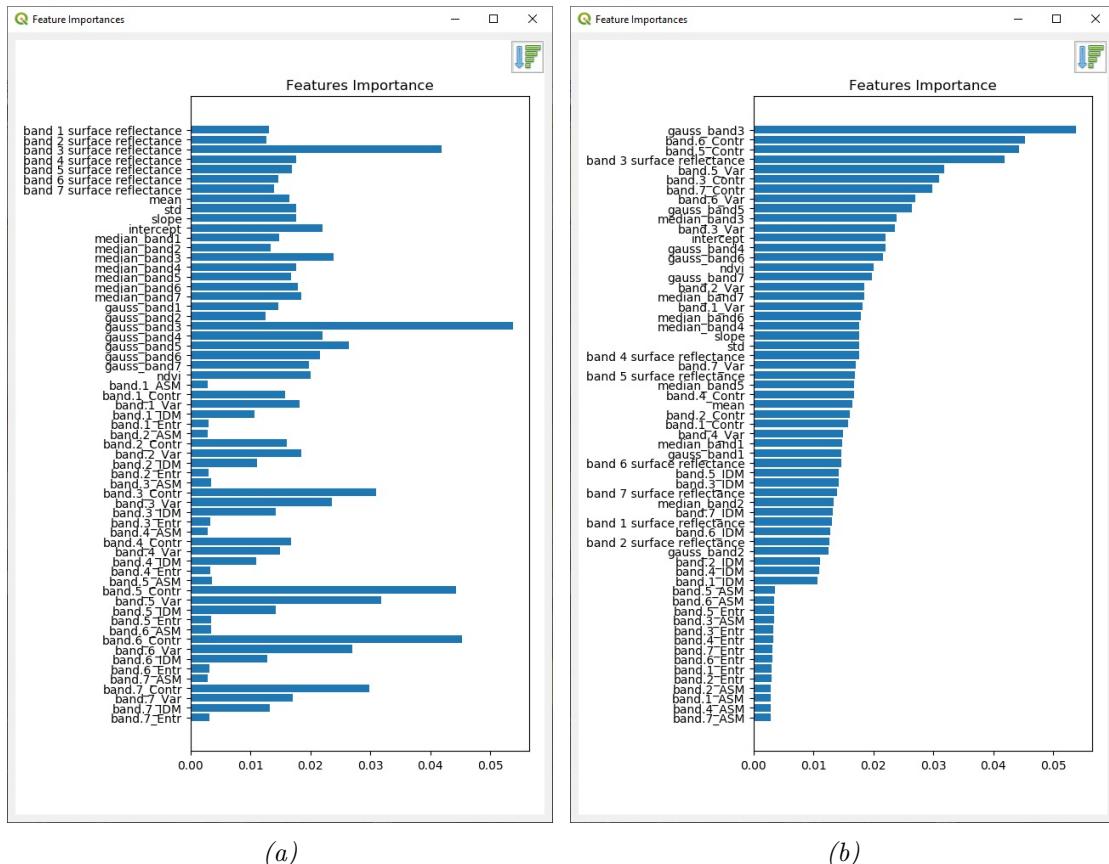


Figura 3.9: Interfaz gráfica del plugin sobre la importancia de los indicadores. (a) Importancia de los indicadores en su orden predeterminado. (b) Importancia de los indicadores ordenadas de manera decreciente.

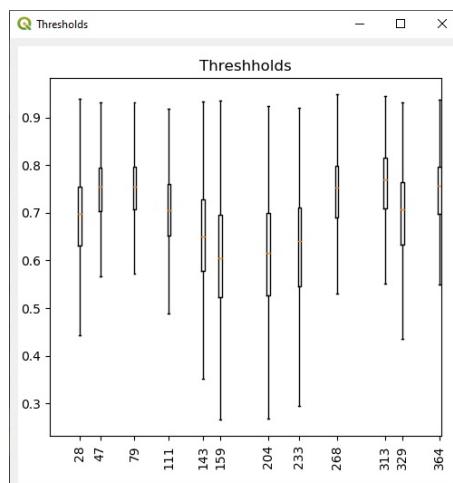
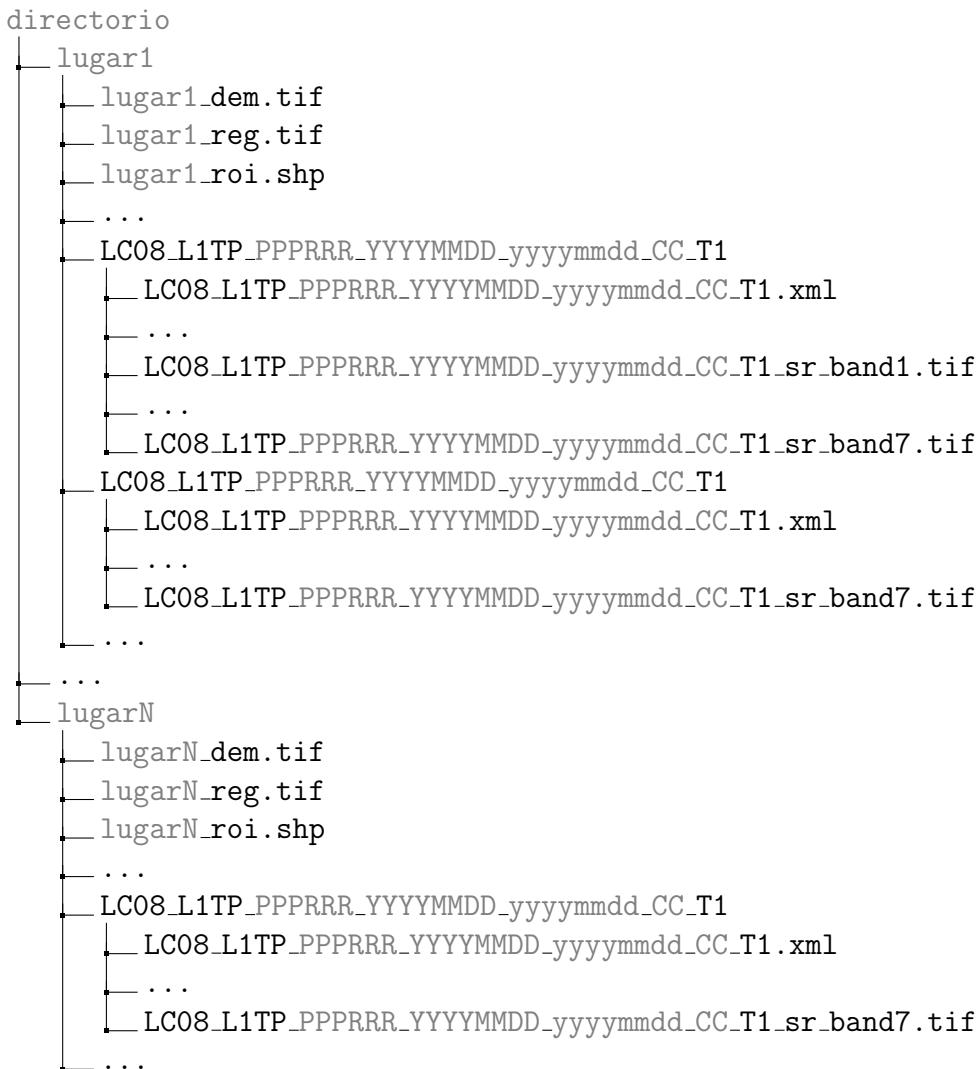


Figura 3.10: Interfaz gráfica del plugin sobre los valores de umbral que definen una máscara

### 3.3.2. Utilización de la Herramienta

Con el fin de automatizar el procesamiento de las imágenes y mantener la interfaz simple, el plugin requiere que se mantenga cierta estructura para el directorio de los archivos. El formato general es el siguiente:



Cada directorio debe estar conformado por lugares. Cada lugar debe contener los archivos que lo definen junto con las imágenes Landsat (una escena por carpeta, con los archivos nombrados al igual que los distribuye USGS). Los archivos son:

- Las bandas (`*_sr_bandn.tif`) y los metadatos (`*_MTL.txt`) de cada escena.
- Un archivo con el sufijo `_reg.tif` el cuál representa la extensión de cada lugar con la cual se quiere trabajar. Puede utilizarse el recorte de una de las bandas y renombrarse para cumplir esta característica.
- Un archivo GeoTIFF con el sufijo `_dem.tif` que contiene el DEM del lugar.
- Un Shapefile con geometría de polígonos con el sufijo `_roi.shp` con un campo `id` donde los polígonos con `id = 1` representan zonas con árboles y los polígonos con `id = 2` representan otro tipo de zonas, como puede ser cuerpos de agua, caminos, distintos tipos de suelo, entre otros.

- Un archivo con el sufijo `.mask.tif` el cual representa la máscara en formato GeoTIFF que indica las presencia o no de árboles en un lugar. El propósito, es utilizar como máscara al archivo resultante de la predicción (es decir, la salida del clasificador), que genera el raster con dos valores; 1 en caso de presencia de árbol y 2 en el caso contrario. Esta misma máscara es la que se puede utilizar como entrada de la corrección de los árboles.

No todas las tareas necesitan de todos los archivos. Los archivos correspondientes a las texturas son solo necesarios en caso de querer utilizar estos indicadores. La corrección de árboles se genera a partir de un DEM a corregir, y una máscara; no necesita de un directorio como el resto de las tareas. A continuación se muestran en una tabla las tareas y los archivos necesarios:

	<code>*_sr_bandn.tif</code>	<code>*_MTL.txt</code>	<code>*_reg.tif</code>	<code>*_dem.tif</code>	<code>*_roi.shp</code>	<code>*_mask.tif</code>	<code>*_text.tif</code>
Cálculo de Texturas Landsat	Si		Si				
Cálculo de Texturas DEM		Si		Si			
Entrenamiento	Si	Si	Si		Si		Si/No
Testeo	Si	Si	Si		Si		Si/No
Predicción	Si	Si	Si				Si/No
Análisis de Umbrales	Si	Si	Si			Si	Si/No

*Cuadro 3.2: Archivos necesarios para cada una de las tareas que tienen como entrada un directorio.*

# Capítulo 4

## Resultados

A continuación se presentan las zonas de estudios utilizadas, los distintos clasificadores entrenados y los resultados obtenidos luego de ser evaluados.

### 4.1. Zonas de Estudio

Las zonas de estudio fueron principalmente tres ubicadas en la provincia de Buenos Aires:

1. Región cerca de General Villegas (34.96 S, 63.14 O): 256 km<sup>2</sup> correspondientes a la Figura 4.1a.
2. Región cerca de Las Flores (36.10 S, 58.86 O): 1126 km<sup>2</sup> correspondientes a la Figura 4.2a.
3. Región cerca de Chaves (37.73 S, 60.57 O): 893 km<sup>2</sup> correspondientes a la Figura 4.2b.

El conjunto de datos utilizado para el entrenamiento fue seleccionado en cada imagen identificando visualmente múltiples polígonos para cada clase. Para cada región se escogió un número aproximadamente equivalente de píxeles árbol y no-árbol, como se muestra en el Cuadro 4.1. Las máscaras con los polígonos corresponden a las Figuras 4.1b, 4.2c, 4.2d. Para los píxeles no-árbol se trató de seleccionar distinta variedad de píxeles: múltiples tipos de suelos, caminos, orillas de cuerpos de agua, y principalmente toda aquella región que se creyó más difícil de identificar para el algoritmo respecto de los árboles.

Región	Cantidad de píxeles árbol	Cantidad de píxeles no-árbol	Total de píxeles
General Villegas	2219	2334	4553
Las Flores	5379	5307	10686
Chaves	1660	1583	3243

*Cuadro 4.1: Proporción de píxeles utilizada para la construcción de los datasets de entrenamiento.*

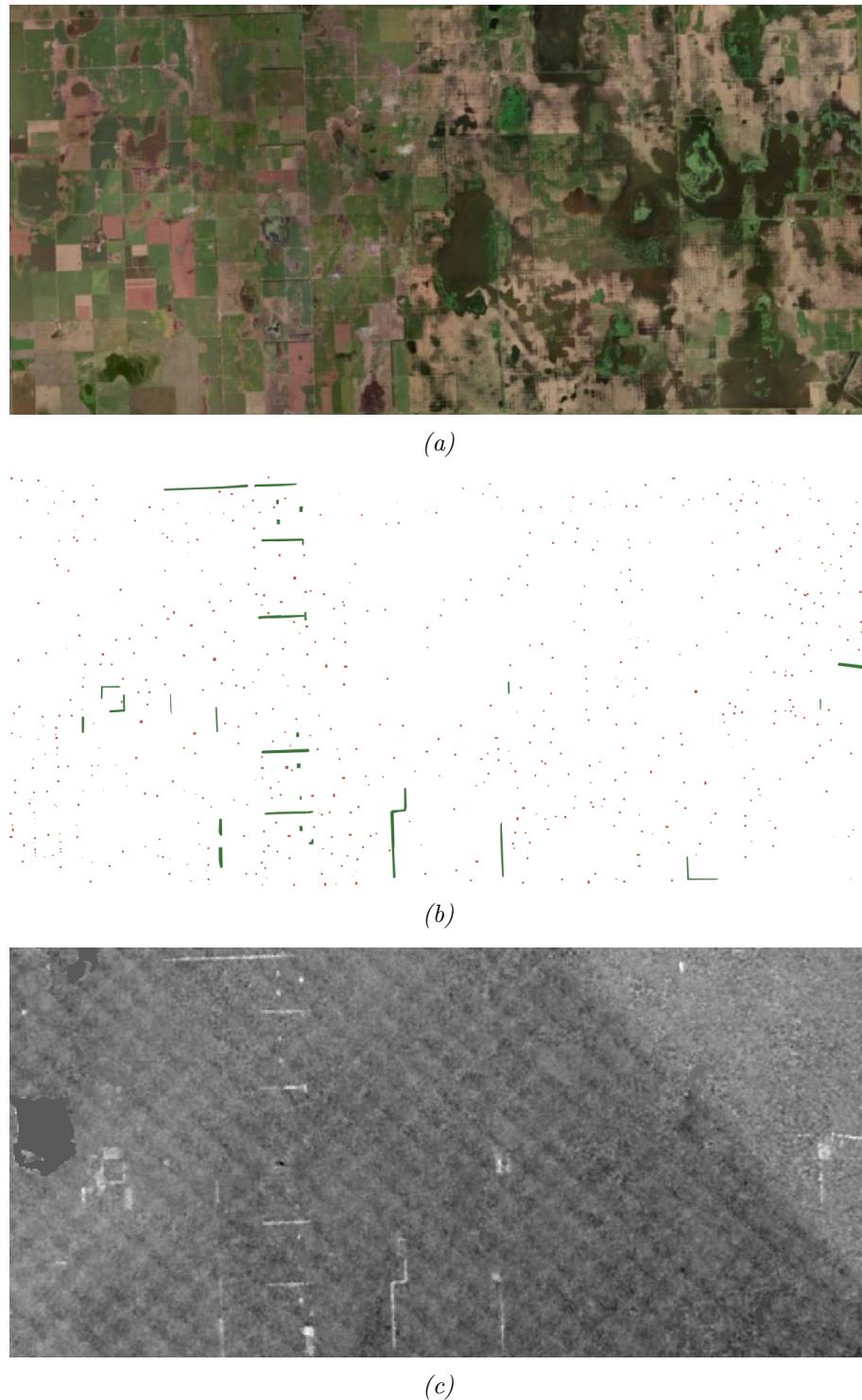


Figura 4.1: Región cerca de General Villegas. (a) Imagen en color natural. (b) Máscara generada manualmente con polígonos indicando presencia de árbol/no árbol. (c) DEM de SRTM de la zona en cuestión.

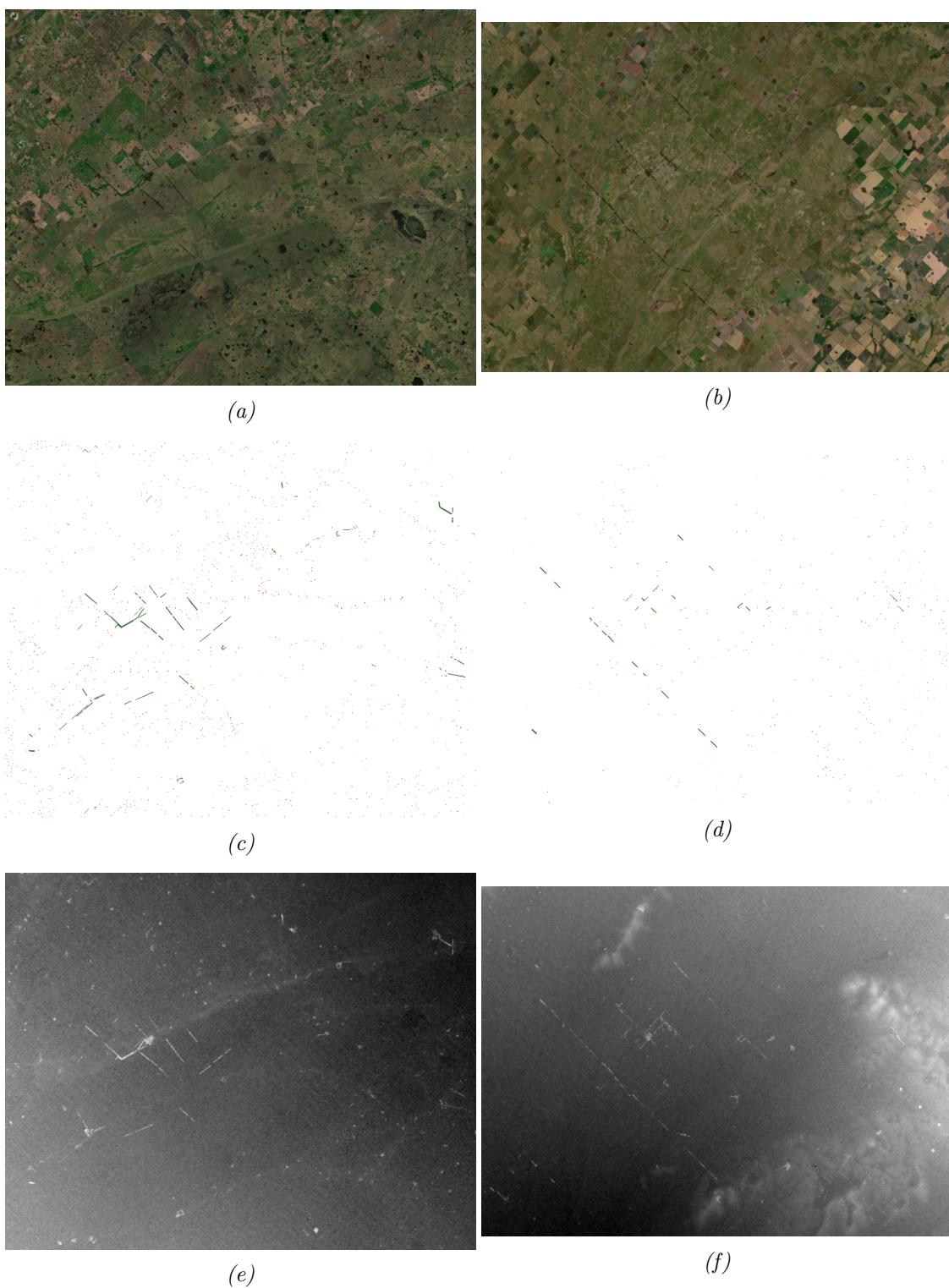


Figura 4.2: Región cerca de Las Flores y Chaves. (a) Imagen en color natural de Las Flores. (b) Imagen en color natural de Chaves. (c) Máscara de Las Flores generada manualmente con polígonos indicando presencia de árbol/no árbol. (d) Máscara de Chaves generada manualmente con polígonos indicando presencia de árbol/no árbol. (e) DEM de SRTM de Las Flores. (f) DEM de SRTM de Chaves.

## 4.2. Resultados de la clasificación: Métricas y Visuales

Para el análisis de la clasificación y de la generación de la máscara de árboles, se propusieron diferentes experimentos donde cada uno de los clasificadores fue evaluado utilizando las métricas mencionadas en la Sección 3.1.2 y la importancia de los indicadores calculada cómo se menciona en la Sección 3.1.1.1. Los experimentos consistieron en generar distintos clasificadores a partir de la combinación de imágenes correspondientes a distintas zonas, épocas del año e indicadores.

A continuación se dividen los experimentos en tres categorías:

1. Resultados para una misma región
  - Utilización de la misma imagen
  - Utilización de misma región y una época similar
  - Utilización de misma región y una época distinta
2. Validación utilizando una región distinta
3. Caso completo: distintas regiones y múltiples épocas

Los experimentos etiquetados con (a) fueron realizados con 61 indicadores (aque-lllos nombrados en la Sección 3.1.3 a excepción de los días y las texturas correspondientes al DEM); aquellos etiquetados con (b) comprendieron la totalidad de los indicadores mencionadas en la Sección 3.1.3.

### 4.2.1. Resultados para una Misma Región

En el Cuadro 4.2 se presentan las métricas que corresponden a los experimentos para una misma región; la generación del clasificador, la predicción y el testeo se realizan con la misma región. La Figura 4.3 contiene las importancias de los indicadores y la Figura 4.4 contiene los resultados visuales de las predicciones para las imágenes de testeo. Como los primeros tres experimentos utilizan exactamente la misma imagen, y los últimos dos incorporan otra imagen de la misma región, las importancia de los indicadores resultó ser la misma o muy similar, por lo que solo se incluyen para el primer experimento.

#### Utilización de la misma imagen

En el primer experimento Exp01, el 75 % de los datos se utilizó para la generación del clasificador y el otro 25 % para la evaluación. En cuanto a la importancia de los indicadores para el caso (a), se observa que las texturas de Angular Second Moment y Entropy fueron las que recibieron menos importancia, es decir, las que probablemente no hacen una buena separación de los datos. La banda 3 (Verde Visible) recibió mucha importancia como era de suponer, pero por otro lado, el NDVI no destacó de otros indicadores siendo este probablemente el índice de vegetación más popular para estimar la cantidad, calidad y desarrollo de la vegetación. Al incorporar los indicadores restantes en el caso (b), se puede observar que la elevación del terreno pasa a tener gran importancia respecto del resto de los otros indicadores.

Exp	Época y Zonas	Métricas	a	b
01	Entrenamiento: 2019/02/16 - Las Flores Testeo: 2019/02/16 - Las Flores	Matriz	[1240 144] [ 146 1142]	[1254 102] [ 48 1268]
		Accuracy	0.891	0.944
		Precision	0.895	0.963
		Recall	0.896	0.925
		F1 Score	0.895	0.944
02	Entrenamiento: 2019/02/16 - Las Flores Testeo: 2019/03/20 - Las Flores	Matriz	[4649 730] [1581 3726]	[4886 493] [ 204 5103]
		Accuracy	0.784	0.935
		Precision	0.746	0.960
		Recall	0.864	0.908
		F1 Score	0.801	0.933
03	Entrenamiento: 2019/02/16 - Las Flores Testeo: 2019/06/08 - Las Flores	Matriz	[2441 2938] [ 236 5071]	
		Accuracy	0.703	
		Precision	0.912	
		Recall	0.454	
		F1 Score	0.606	
04	Entrenamiento: 2019/02/16 - Las Flores 2019/06/08 - Las Flores Testeo: 2019/03/20 - Las Flores	Matriz	[4845 534] [1974 3333]	
		Accuracy	0.765	
		Precision	0.711	
		Recall	0.901	
		F1 Score	0.794	
05	Entrenamiento: 2019/02/16 - Las Flores 2019/06/08 - Las Flores Testeo: 2019/03/20 - Las Flores 2019/05/23 - Las Flores	Matriz	[9898 860] [2841 7773]	
		Accuracy	0.827	
		Precision	0.777	
		Recall	0.920	
		F1 Score	0.842	

Cuadro 4.2: Resultados de las métricas para experimentos que involucran la misma región.

### Utilización de misma región y una época similar

A pesar de que los resultados para la primera experiencia son prometedores dado la baja cantidad de falsos negativos y positivos, la idea es que el clasificador pueda ser aplicado a imágenes de distintas zonas y períodos de tiempo. Como primera variación para analizar lo sucedido al utilizar imágenes distintas del mismo lugar pero dentro de la misma estación del año, se realizó la Exp02. Los resultados visuales no se incluyeron porque resultaron muy parecidos al caso anterior. En cuanto a las métricas, se observa un leve empeoramiento en los resultados, particularmente en las métricas respecto de Exp01. Si se compara el caso (a) de Exp01 con el de Exp02

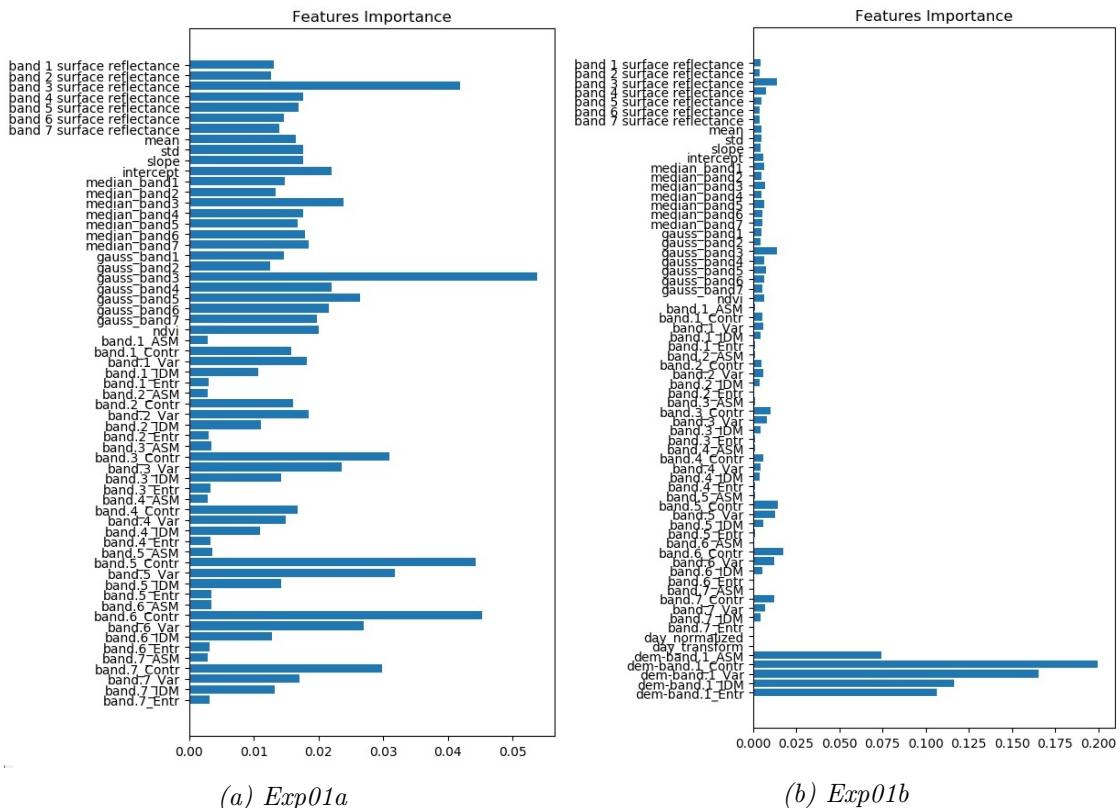


Figura 4.3: Importancia de los indicadores para primer experimento.

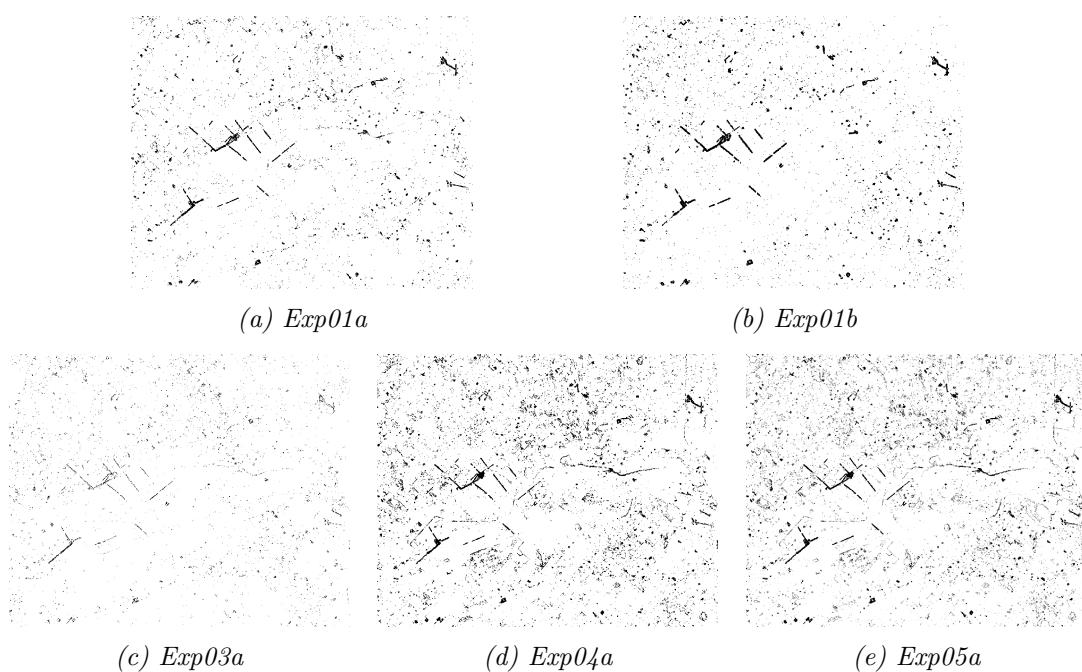


Figura 4.4: Resultados visuales para los experimentos de la misma región.

y el caso (b) de Exp01 con Exp02, se observa que (a) obtiene en relación a (b) un desempeño inferior. Esto tiene sentido dado que, mientras en Exp01 se utiliza la misma imagen Landsat y el mismo DEM, en Exp02 se utiliza distinta imagen Landsat, pero mismo DEM.

### Utilización de misma región y una época distinta

Para el análisis en el cambio de temporadas, se realizaron los experimentos Exp03a, Exp04a y Exp05. El caso (b) no fue considerado por ser la misma zona, y por consecuencia, tener el mismo DEM. Las métricas de estos experimentos reflejan que evidentemente la temporada influye en gran medida a la hora de generar la máscara. Sobre todo, esto último queda claro con Exp03, que genera el clasificador con una imagen tomada en verano y predice para una imagen tomada en invierno.

En cuanto a los resultados visuales, Exp03 es también la que obtiene el peor resultado dado el aspecto. Para Exp05, como la imagen se predijo para múltiples imágenes, los resultados se encuentran superpuestos y con 50 % de opacidad.

Por último, Exp04 respecto de Exp02, incorpora al entrenamiento una imagen de otra época, y se puede observar que las métricas no se ven realmente afectadas por esta incorporación.

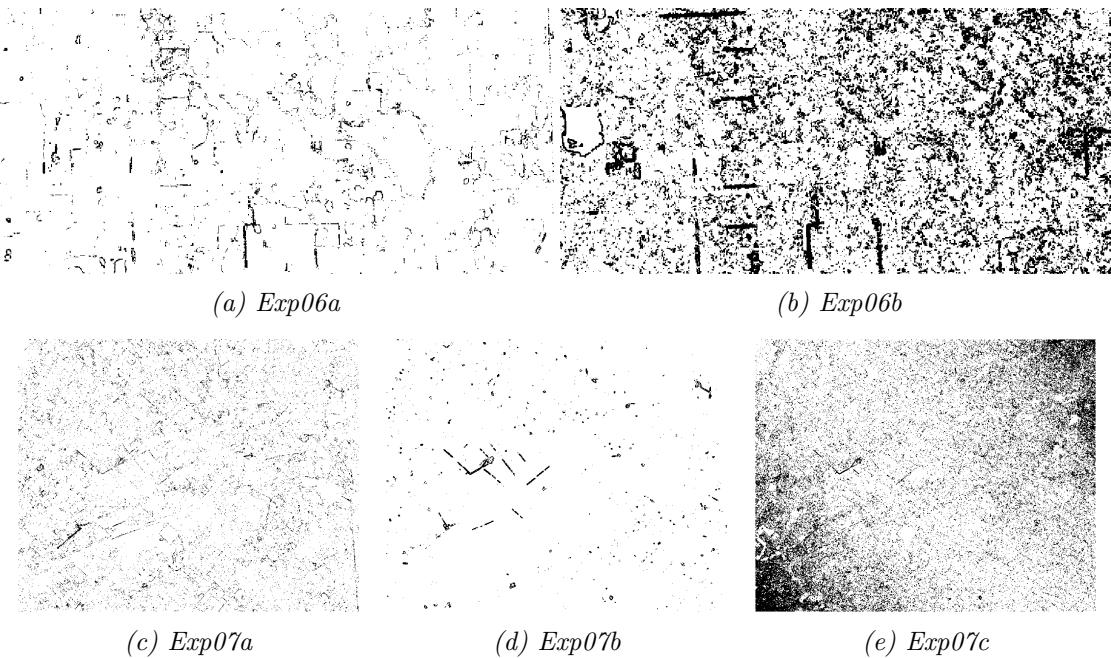
#### 4.2.2. Validación Utilizando una Región Distinta

En Exp06 y Exp07 se propone la utilización de distintas zonas, es decir, se genera el clasificador con imágenes correspondientes a una zona, y se predice el resultado y se testea en otra. En el Cuadro 4.3 se muestran las métricas resultantes y en la Figura 4.5 se muestran los resultados visuales. Para el caso de Exp07, se observa un experimento extra que también incluye todos los indicadores (el (c)), donde el DEM utilizado fue recodificado, en vez de re-escalado (Sección 3.1.3.1). Originalmente, los experimentos habían sido realizados a partir de los DEMs recodificados, pero como se observa en la imagen visual de la predicción, se resalta la elevación del terreno, haciendo que el resultado de la predicción con el DEM re-escalado sea notablemente superior, al igual que el resultado de las métricas. Las experiencias

Exp	Época y Zonas	Métricas	a	b	c
06	Entrenamiento: 2019/02/16 - Las Flores Testeo: 2019/02/05 - Gral Villegas	Matriz	[713 1506] [237 2097]	[2126 93] [ 738 1596]	
		Accuracy	0.617	0.817	
		Precision	0.751	0.742	
		Recall	0.321	0.958	
		F1 Score	0.45	0.837	
07	Entrenamiento: 2019/02/05 - Gral Villegas Testeo: 2019/02/16 - Las Flores	Matriz	[4458 6300] [1656 8958]	[3829 1550] [ 29 5278]	[2804 2575] [1293 4014]
		Accuracy	0.628	0.852	0.638
		Precision	0.729	0.992	0.684
		Recall	0.414	0.721	0.521
		F1 Score	0.528	0.829	0.592

Cuadro 4.3: Experimentos con imágenes de distinto lugar, misma estación.

## CAPÍTULO 4. RESULTADOS



*Figura 4.5: Resultados visuales para las experiencias que utilizan regiones distintas.*

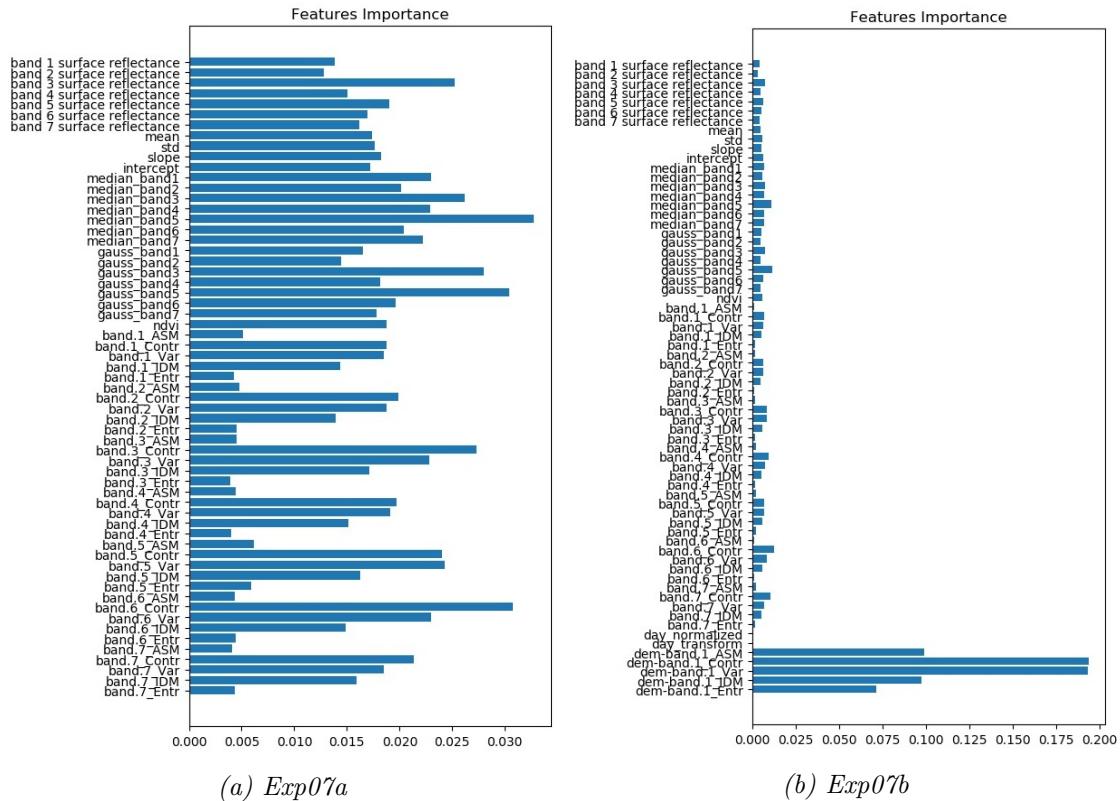


Figura 4.6: Importancia de los indicadores para Exp07.

Exp06a y Exp07a, demuestran que los indicadores sobre las imágenes Landsat son útiles para determinar la precisión, pero no tanto el recall. Es decir, si el clasificador dice que es un árbol, probablemente lo sea, pero puede que no todos los árboles sean detectados. La incorporación de los indicadores de elevación, obtienen más positivos, aumentando consecuentemente el recall.

Los clasificadores que incluyen los indicadores relacionados con el DEM reescalado siguen obteniendo un buen desempeño, con respecto de las experiencias para la misma región.

En cuanto a los resultados visuales de Exp06b y Exp07b (Figuras 4.5b y 4.5d) se observa que Exp07b obtiene resultados más satisfactorios que Exp06b. A partir de estos resultados, se observó que el DEM de SRTM de la zona de entrenamiento de Exp07b (Gral Villegas) sufre del efecto de striping mencionado en la sección 2.6, lo cual genera ruido en la imagen. Los resultados más satisfactorios se obtienen de entrenar con la imagen que posee el mayor error y predecir para la imagen con el menor ruido, que el caso inverso.

La Figura 4.6 contiene la importancia de los indicadores para Exp07 (en Exp06 se utilizó la misma imagen de entrenamiento que los casos que utilizaron la misma región, por lo tanto la importancia es la misma). En cuanto a la importancia de los indicadores, no existen grandes diferencias con los entrenamientos anteriores, sobre todo en lo referente a las texturas; simplemente están un poco más uniformes los valores, y aparece una variación de los indicadores más importantes, como el filtro de media de la banda 5.

#### 4.2.3. Caso Completo

Por último, los últimos tres experimentos representan el caso completo de la manera en la cual se pretende usar la herramienta, es decir, donde el clasificador es generado con una región distinta a la que se utiliza para predecir, y la época puede ser cualquiera en el año.

En el Cuadro 4.4 se observan las métricas de los tres experimentos realizados. Nuevamente, el clasificador generado a partir de la imagen que sufre de striping (Gral Villegas) se ve afectado negativamente respecto del resto. Las otras dos zonas (Las Flores y Chaves) resultan más parecidas entre ellas. Sin embargo, al generar un clasificador entre Gral Villegas y Las Flores, los resultados fueron muy parecidos a los generados por el clasificador construido sólo a partir de Las Flores. En la Exp09, en cuanto a las experiencias de la Sección 4.2.2, ocurre lo mismo con la precisión: los indicadores en las imágenes Landsat permiten no tener una elevada cantidad de falsos positivos. La precisión sube al agregar los indicadores relacionados con la elevación, pero ya es alta y mucho mayor al recall en el escenario a.

Las imágenes visuales de las predicciones están generadas a partir de todas las imágenes de testeо, donde a cada máscara se le asignó un color con un 8 % de transparencia. Los mejores resultados se obtuvieron en Exp09b y Exp10b, correspondiente con los valores de las métricas.

En la Figura 4.8 se muestra la comparación entre dos máscaras de Chaves. La Subfigura 4.8a es la máscara resultante de entrenar el clasificador con 75 % de los datos y predecir para la misma imagen. La Subfigura 4.8b es el resultado de la operación OR de todas las máscaras obtenidas en Exp09b.

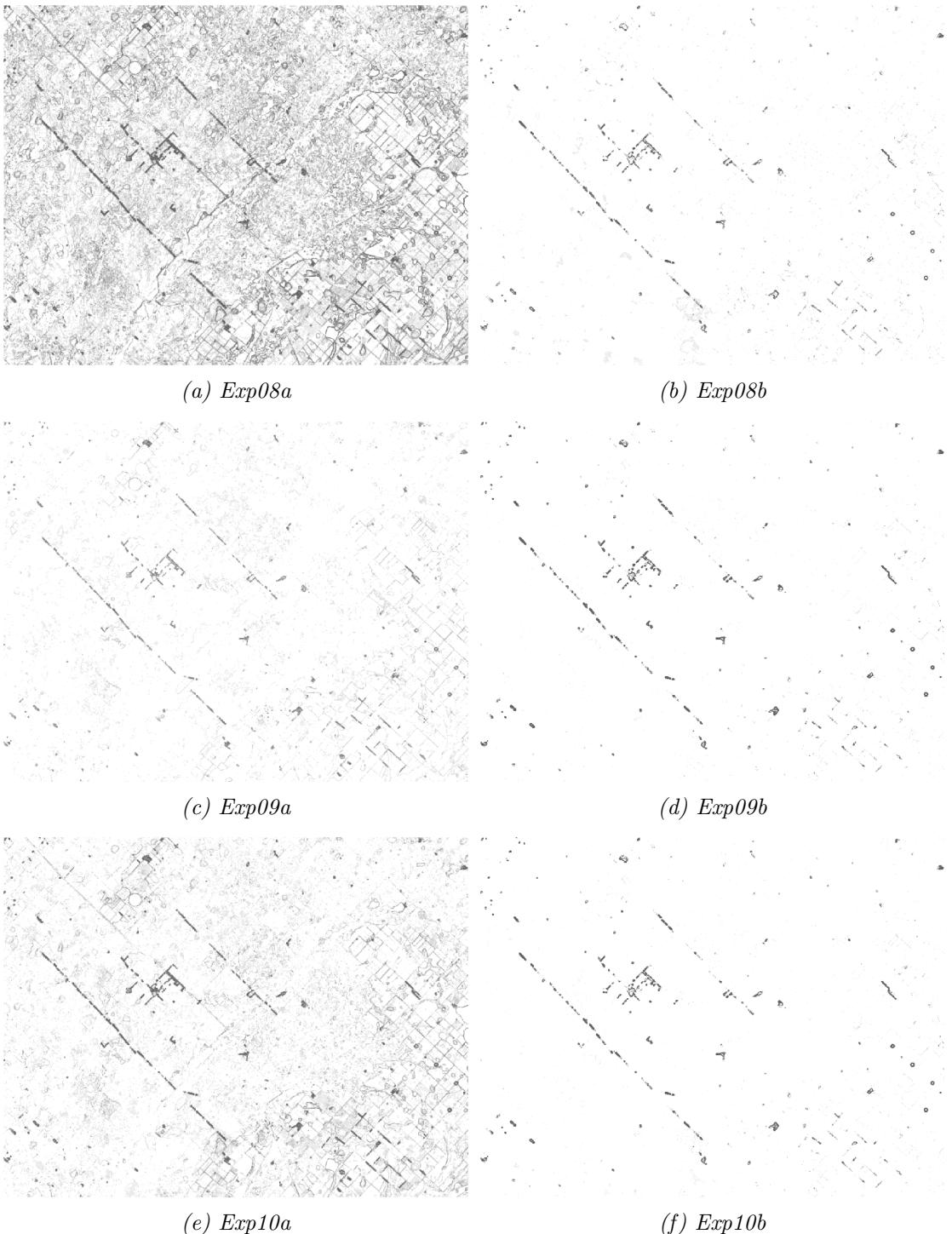


Figura 4.7: Resultados visuales para el caso completo (distintas regiones y múltiples épocas).

Exp	Época y Zonas	Métricas	a	b
08	Entrenamiento: 2019/01/04 - Gral Villegas 2019/02/05 - Gral Villegas 2018/03/06 - Gral Villegas 2018/03/22 - Gral Villegas 2019/05/12 - Gral Villegas 2018/05/25 - Gral Villegas 2019/07/15 - Gral Villegas 2018/08/29 - Gral Villegas 2018/09/14 - Gral Villegas 2018/10/16 - Gral Villegas 2017/11/30 - Gral Villegas 2018/12/03 - Gral Villegas	Testeo: 2018/01/28 - Chaves 2019/02/16 - Chaves 2018/03/01 - Chaves 2019/04/21 - Chaves 2019/05/23 - Chaves 2019/06/08 - Chaves 2018/07/23 - Chaves 2017/08/21 - Chaves 2019/09/12 - Chaves 2017/11/25 - Chaves 2017/12/11 - Chaves 2018/12/30 - Chaves	Matriz	[17279 2641] [ 7504 11492]
			Accuracy	0.739
			Precision	0.697
			Recall	0.867
			F1 Score	0.773
				0.756
09	Entrenamiento: 2018/01/28 - Las Flores 2019/02/16 - Las Flores 2019/03/20 - Las Flores 2019/04/21 - Las Flores 2019/05/23 - Las Flores 2019/06/08 - Las Flores 2018/07/23 - Las Flores 2017/08/21 - Las Flores 2018/09/25 - Las Flores 2017/11/09 - Las Flores 2017/11/25 - Las Flores 2018/12/30 - Las Flores	Testeo: Chaves de Exp08	Matriz	[9562 10358] [1382 17614]
			Accuracy	0.698
			Precision	0.874
			Recall	0.480
			F1 Score	0.620
				0.800
10	Entrenamiento: Gral Villegas de Exp08 y Las Flores de Exp09	Testeo: Chaves de Exp08	Matriz	[15053 4867] [ 3814 15182]
			Accuracy	0.777
			Precision	0.798
			Recall	0.756
			F1 Score	0.776
				0.799

Cuadro 4.4: Experimentos de caso completo (distinto lugar y distintas estaciones).

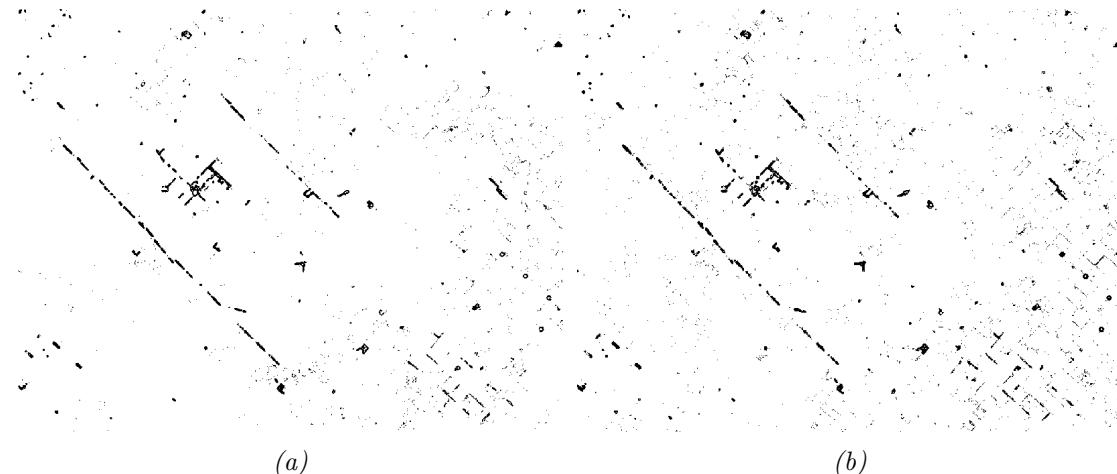


Figura 4.8: Comparación de máscaras para Chaves. (a) Predicción resultante para Chaves entrenando con el 75 % de los datos de la misma imagen. (b) Predicción resultante para Chaves realizando un OR entre las máscaras obtenidas para Exp09b.

### 4.3. Estudio del Efecto Estacional

Mediante el análisis de series temporales de la bandas espectrales es posible la identificación de patrones para entender el comportamiento de los objetos teledetectados. A través de la técnica de Random Forest se detectan automáticamente los umbrales que segmentan la vegetación. A continuación se presentan los resultados obtenidos de los umbrales que definen la máscara de árboles. En algunos casos, la máscara corresponde a la Subfigura 4.8b (generada a partir de la operación **OR** de los resultados de Exp09b); en otros, la máscara corresponde solo a los valores reales, es decir, a los datos etiquetados como se muestra en la Subfigura 4.2d.

Las Subfiguras 4.9a y 4.9b corresponden a los umbrales de la banda verde para la máscara generada a partir del etiquetado manual y para la que fue calculada a partir de Exp09b. Se observa que los umbrales se corresponden aproximadamente entre ellos, pero como era de esperar, en 4.9a los datos se encuentran más concentrados en el centro. Probablemente, esto ocurre dada la presencia de algunos falsos positivos en la máscara de Exp09.

Por otro lado, las Subfiguras 4.9c y 4.9d corresponden a la banda 3 con el filtro gaussiano de la máscara generada a partir del etiquetado manual y de la máscara generada a partir de Exp09b. Entre ellas se observa lo mismo que el caso anterior, pero respecto de las anteriores, el filtro gaussiano acota aun más los boxplots. Esto tiene sentido dado que el filtro produce un suavizado que elimina grandes contrastes haciendo los píxeles más parecidos entre ellos.

En las Subfiguras 4.9e y 4.9f se presentan los umbrales del NDVI tanto de la máscara etiquetada como de la obtenida de Exp09b. En 4.10a y 4.10b se observan los umbrales de las bandas 4 (rojo visible) y 5 (infrarrojo cercano) de Exp09b, las cuales se utilizan para calcular el NDVI. Entre estos umbrales de NDVI, se puede ver que la máscara y la predicción coinciden a grandes rasgos.

Si se comparan los resultados con aquellos reportados por trabajos previos, particularmente por la tesis de grado de Amalia L. Thomas [TDR15], quien analizó los umbrales óptimos para diferenciar la vegetación arbórea, pueden existir algunas diferencias. Thomas, en su tesis, toma las bandas relacionadas con la vegetación (verde, roja, infrarroja cercana e infrarroja de onda corta 1), determinó los umbrales máximos y mínimos para imágenes entre el 2013 y 2015 para Landsat 8 y a partir de ellos determinó una función sinusoidal que se ajuste a los valores de umbrales. Particularmente en este caso, solo se tomó una imagen por mes. Dada la presencia de nubes, el conjunto de imágenes puede ser del 2019, 2018 o 2017 y se observó la distribución de los datos a través de los box-plots. Algunas bandas, como por ejemplo la banda 3, tienen una forma que se asemeja a una función sinusoidal mientras que otras, como la banda 5, no presentan una forma característica.

Por otro lado, los árboles perennes, caracterizados por tener hojas vivas (que no se caen) durante todo el año, mantienen un NDVI más constante durante todo el ciclo [Yan+19]. Si se observa la Subfigura 4.10c generada a partir de Exp01b, se observa que para la máscara de Las Flores, el NDVI varía de forma más característica a un árbol de tipo caducifolio, es decir, que pierde su follaje durante cierta época del año, en general, en las épocas más frías del año.

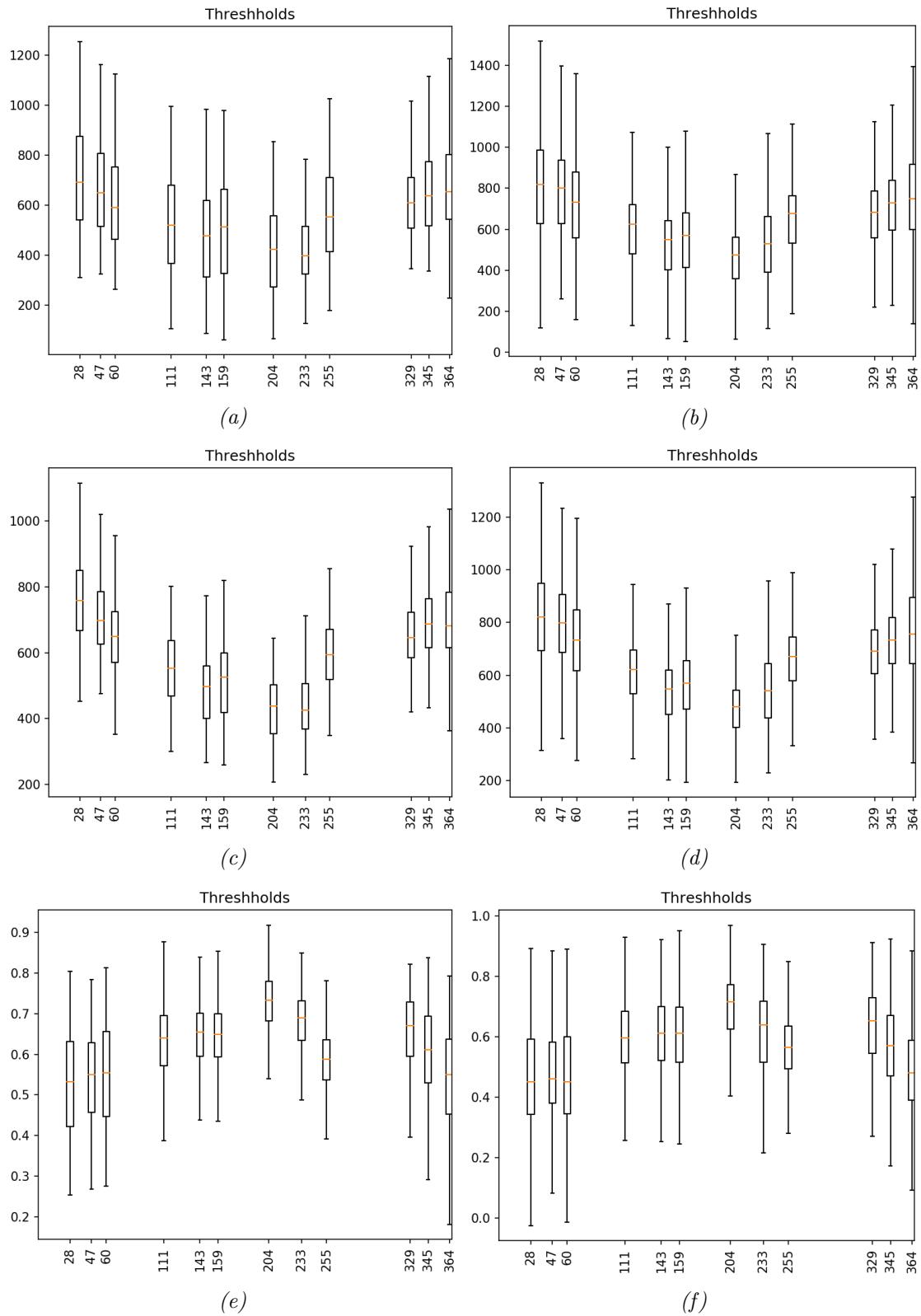


Figura 4.9: Box-plots de los umbrales que definen la máscara de vegetación arbórea. (a) Umbrales de la banda 3 (verde visible) para la máscara etiquetada. (b) Umbrales de la banda 3 (verde visible) para la máscara predecida a partir de los resultados de Exp09b. (c) Umbrales de la banda 3 con el filtro gaussiano para la máscara etiquetada. (d) Umbrales de la banda 3 con el filtro gaussiano para la máscara predecida a partir de los resultados de Exp09b. (e) Umbrales del NDVI para la máscara etiquetada. (f) Umbrales del NDVI para la máscara predecida a partir de los resultados de Exp09b.

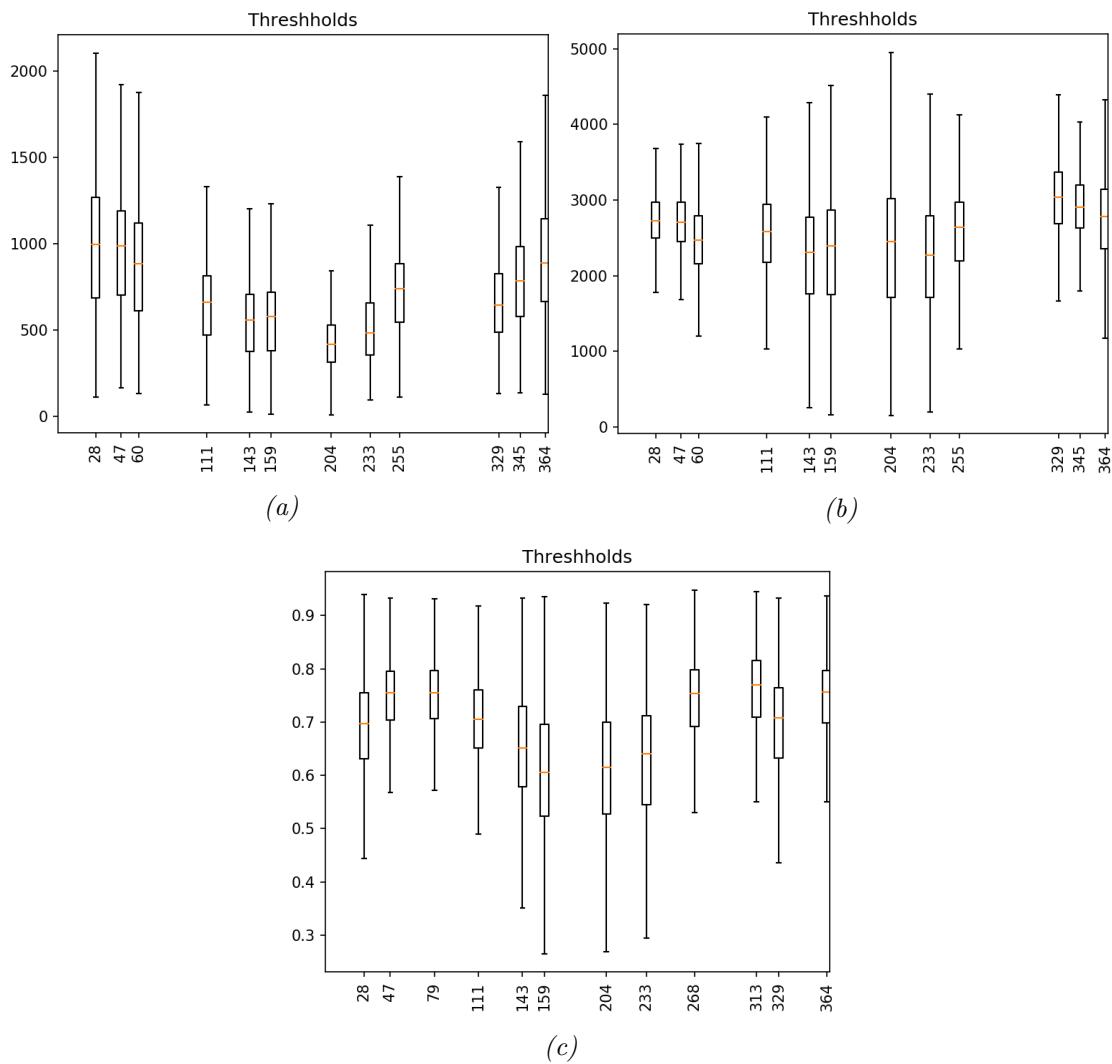


Figura 4.10: Box-plots de los umbrales que definen la máscara de vegetación arbórea. (a) Umbrales de la banda 4 (rojo visible) para la máscara predecida a partir de los resultados de Exp09b. (b) Umbrales de la banda 5 (Near Infrared) para la máscara predecida a partir de los resultados de Exp09b. (c) Umbrales para NDVI de la máscara de Exp01b.

#### 4.4. Corrección de Árboles

En la Figura 4.11a se muestra el resultado de corregir el DEM de la zona de Chaves utilizando el método de interpolación IDW explicado en la sección 3.2.

Como las zonas arboladas presentan bordes un poco difusos, se procede a ensanchar o dilatar la máscara en un píxel, como se muestra en la Subfigura 4.11d y se corrige el DEM (Subfigura 4.11e), obteniendo resultados superiores a los de la corrección anterior. En las Subfiguras 4.12a y 4.12b se muestra el resultado de dilatar la máscara en dos píxeles. Si bien corrige un poco mejor aquellas zonas donde el área de los árboles es mayor, o más parecida a pequeños bosques, posiblemente altere el DEM de manera no deseada.

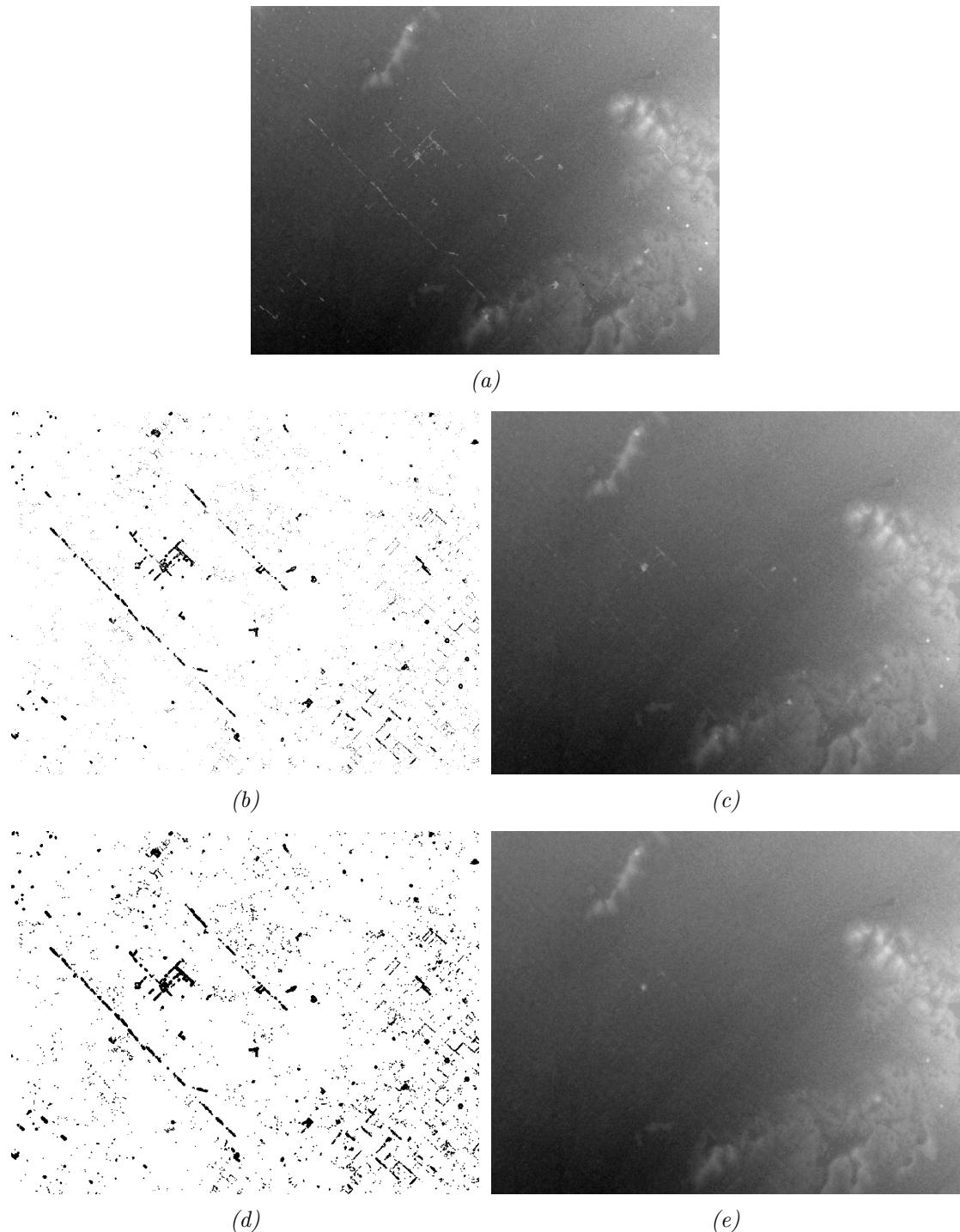


Figura 4.11: Corrección del DEM de Chaves. (a) DEM original. (b) Máscara de árboles sin dilatar. (c) Corrección obtenida del DEM a partir de la máscara en (b). (d) Máscara de árboles dilatada en un pixel. (e) Corrección obtenida del DEM a partir de la máscara en (d).

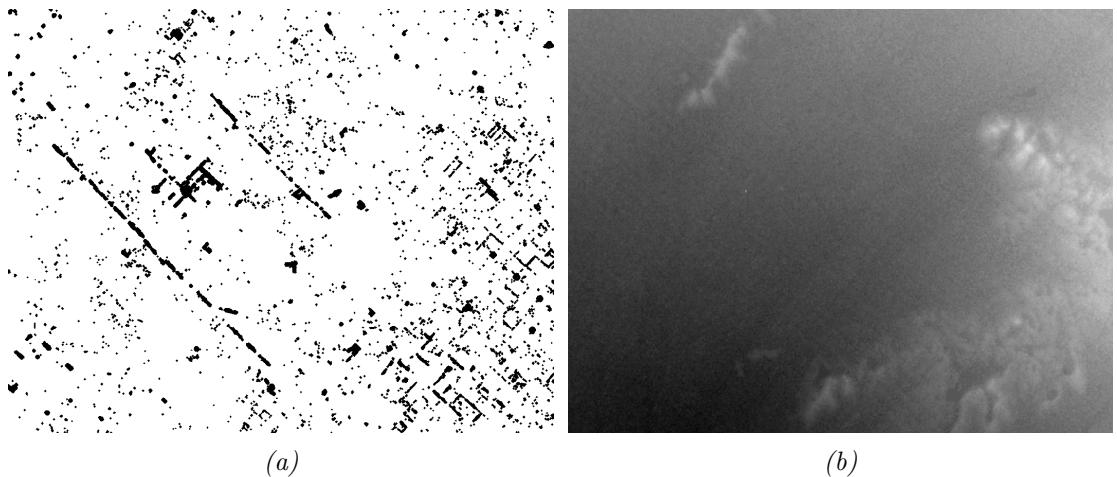


Figura 4.12: Corrección del DEM de Chaves. (a) Máscara de árboles dilatada en dos píxeles. (b) Corrección obtenida del DEM a partir de la máscara en (a).

## 4.5. Validación con LiDAR

Para validar los resultados, se utilizó un DEM obtenido a través de un dispositivo LiDAR de la zona del Condado de Blackford en EEUU (40.47 N, 85.32 O). La máscara se obtuvo a partir del clasificador en Exp09b. En la Subfigura 4.13a se muestra la elevación del terreno a partir de la imagen obtenida con LiDAR; la Subfigura 4.13b muestra el DEM de SRTM. La Figura 4.13 incluye también las imágenes de las máscaras superpuestas sobre el DEM donde se aprecia mejor el recubrimiento de la máscara sobre la elevación del terreno. Las Subfiguras 4.14a y 4.14b muestran el resultado de la corrección hecha a partir de la máscara sin dilatar y otra dilatada en dos píxeles. Se observa que la máscara sin dilatar detecta bastante bien los contornos de las zonas con árboles. Probablemente, el resto de la zona arbolada pueda detectarse también, incorporando al entrenamiento una zona que tenga pequeños bosques de árboles donde se tomen en cuenta mejor otras texturas.

Por último, en la Figura 4.15 se analiza el perfil del terreno con Profile Tool (Sección A.1.3.2). En el gráfico de líneas se observa en rojo, el DEM de SRTM que contiene los árboles, en azul el DEM corregido y en verde, el DEM de LiDAR. El canvas muestra el raster de World Imagery de ArcGIS (Sección A.1.4) con la máscara dilatada en dos píxeles (Subfigura 4.13d) y el recorrido a través del cuál se genera el perfil del terreno. En gran medida, el DEM corregido y el DEM de LiDAR coinciden; como era de esperar, en las zonas grandes de árboles (donde se puede ver el cursor) la máscara no marca los árboles, generando una elevación en el perfil del DEM corregido. A su vez, hay que tener en cuenta que los DEM de SRTM no son lo más preciso que hay; a menudo contienen ruido speckle que puede reducir significativamente la calidad de los análisis geomorfométricos [SSM10].

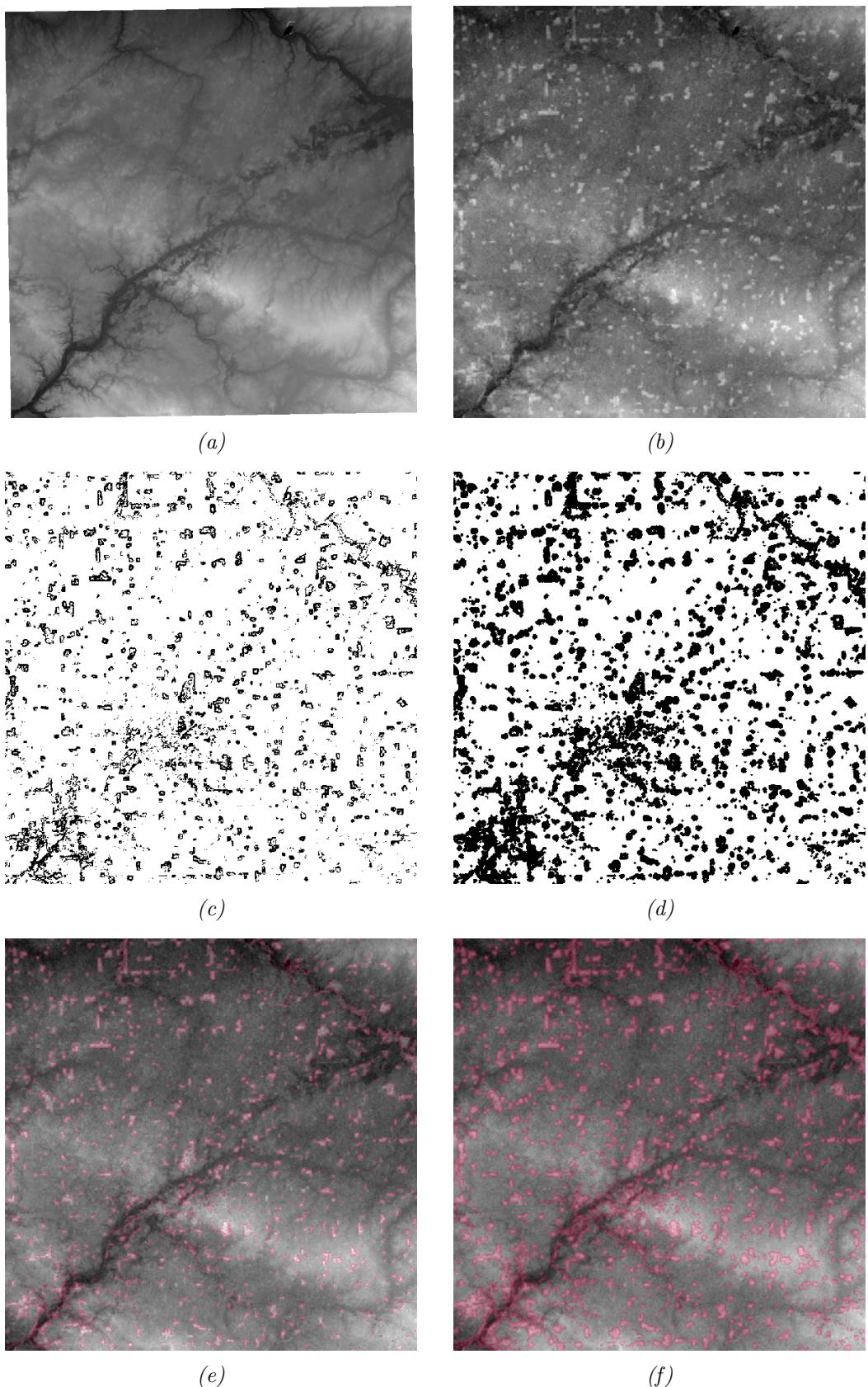
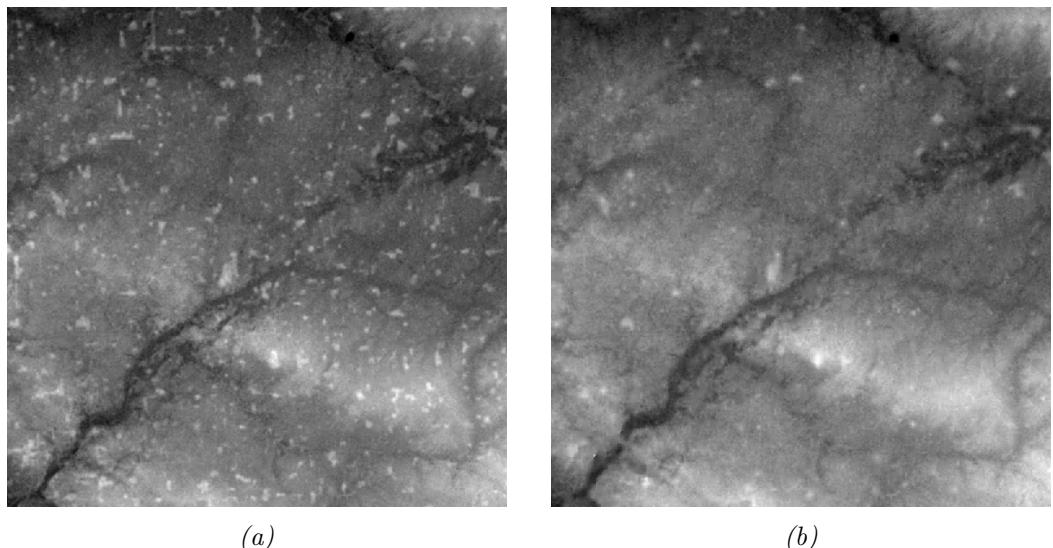


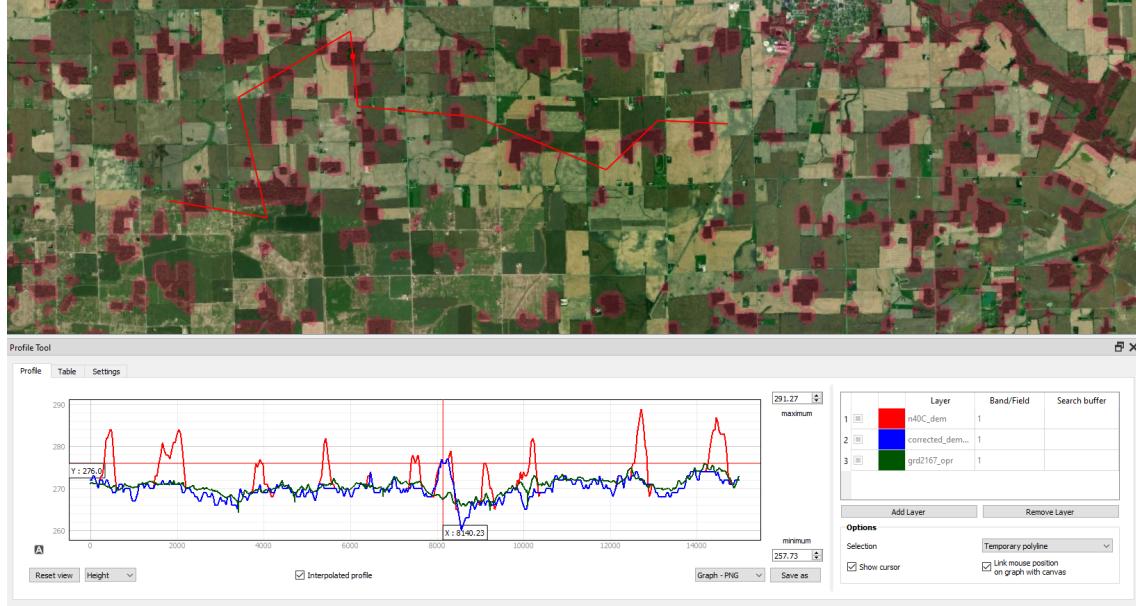
Figura 4.13: Validación con LiDAR. (a) DEM Lidar. (b) DEM de SRTM. (c) Máscara predecida a partir de Exp09b. (d) Máscara obtenida a partir del clasificador de Exp09b dilatada en dos píxeles. (e) Máscara no dilatada sobre DEM de SRTM. (f) Máscara obtenida a partir del clasificador de Exp09b dilatada en dos píxeles sobre DEM de SRTM.



(a)

(b)

Figura 4.14: Validación con LiDAR. (a) DEM de SRTM corregido con máscara sin dilatar. (b) DEM de SRTM corregido dilatando la máscara en dos píxeles.



(a)

Figura 4.15: Análisis del perfil del terreno. En la parte superior se observa una imagen en color natural de la zona, junto con la máscara superpuesta en una tonalidad de rojo con transparencia, y el recorrido a través del cual se genera el perfil del terreno. En la parte inferior, el gráfico muestra efectivamente los perfiles para: en rojo, el DEM de SRTM; en azul, el DEM SRTM corregido y en verde, el DEM de LiDAR.

# Capítulo 5

## Conclusión

La obtención de un modelo del suelo desnudo a partir de la elevación del terreno es muchas veces necesaria para realizar análisis sobre la superficie de la Tierra. En el presente trabajo se realiza un procedimiento para la corrección de Modelos Digitales de Elevación (DEMs) con el objetivo de detectar de forma automática la vegetación arbórea y removerla de la superficie de la Tierra, en particular, para la aplicación de simuladores de inundación sobre el terreno.

A través del análisis de imágenes multiespectrales (particularmente imágenes Landsat 8) y un algoritmo de clasificación (Random Forest o RF), se crearon máscaras con la zona de interés. Los indicadores incorporados incluyeron: bandas espectrales, álgebra de bandas, filtros, NDVI, fechas de adquisición de imágenes, texturas sobre las bandas y sobre la elevación del terreno. Luego, se prosiguió a remover, a través de un método de interpolación, la vegetación arbórea a partir de la máscara obtenida. Como resultados y observaciones más importantes:

- El desempeño de los clasificadores llegó a un F1 Score de hasta 0.8, siendo los clasificadores construidos a partir de una imagen por mes en el año, y siendo evaluados en otra zona geográfica.
- Los indicadores más importantes fueron principalmente las texturas evaluadas en los DEMs, junto con las bandas 3 (verde visible), 5 (near infrared) y 6 (SWIR 1), y filtros relacionados a estas bandas. Por otro lado, el NDVI no destacó dentro de ellas.
- Para el cálculo de las texturas, si bien recodificar las imágenes para el caso de las bandas espectrales demostró resaltar características como los bordes, esto no fue del todo cierto para el cálculo de las texturas en los DEMs. El recodificado del DEM demostró resaltar la pendiente del terreno pero no tanto de los árboles. Al ser considerado el DEM recodificado en la clasificación, esto generó defectos en las máscaras, mientras que los resultados fueron superiores al ser simplemente re-escalado.
- Como los bordes de la vegetación arbórea resultan difusos en los DEMs, dilatar las máscaras demostró ser de gran utilidad a la hora de interpolar los valores, ya que estos deben tomarse desde lo que sería el suelo, y no de una ligera pendiente provocada por el árbol.
- Por último, las máscaras generadas presentaron algunos defectos al ser testeadas en zonas que incluyen más píxeles contiguos con árboles. Esto se debe a

que en un principio la aplicación fue ideada para remover líneas de árboles que generan un efecto de dique (comúnmente en la región pampeana). Por consiguiente, el etiquetado de los datos de entrenamiento fue generado principalmente a partir de esta característica. De todos modos, aun si se detectaran estos arboles, la corrección del DEM es un proceso complicado debido a la ausencia de terreno desnudo próximo a los píxeles a corregir.

Cabe destacar, que si bien las máscaras pueden tener sus desperfectos, la implementación del plugin en QGIS permite la automatización de semejante tarea. En el caso más sencillo de utilización para un usuario no experto, se puede predecir sobre la misma imagen que fue utilizada para entrenar. Esto requiere que el usuario genere manualmente el set de entrenamiento, pero simplemente marcando algunas regiones, puede obtenerse una máscara como las que fueron obtenidas en un principio. Como alternativa, siempre pueden utilizarse los clasificadores ya generados y dilatar un poco más la máscara para que incluya los puntos contiguos a las zonas detectadas como árboles que fueron falsos negativos en la clasificación. La máscara incorporará probablemente algunos falsos positivos también, pero que a la hora de ser interpolados, no presentaran un gran defecto en la elevación del terreno.

## 5.1. Contribuciones

El procesamiento de DEMs puede resultar en una tarea tediosa y complicada, sobre todo cuando se trata de obtener un modelo de tierra desnuda. El presente trabajo aporta:

- La detección automática de vegetación arbórea en imágenes multiespectrales para la generación de máscaras que definen el posicionamiento de los árboles.
- Una manera de substraer de un modelo de representación de la superficie, los objetos marcados a partir de una máscara.
- Una herramienta integrada a un GIS Open Source, que no requiere de conocimientos avanzados para su utilización.
- Una serie de clasificadores listos para utilizar y generar la clasificación de píxeles en árbol/no-árbol.

## 5.2. Trabajos Futuros

Durante el desarrollo del presente trabajo, surgieron otras posibles líneas de continuación dada cuestiones que fueron surgiendo en el proceso, o que excedían el alcance del trabajo y pretenden seguir avanzando o mejorando sobre los objetivos logrados. Entre ellas se destacan:

- La incorporación de otros métodos de clasificación, como podría ser la utilización de redes neuronales.

- La incorporación y el análisis de nuevos indicadores. Por ejemplo, algunos podrían ser: Enhanced Vegetation Index (EVI), Normalized Difference Water Index (NDWI), Modified Soil-Adjusted Vegetation Index (MSAVI2), Soil Adjusted Total Vegetation Index (SATVI), Soil Adjusted Vegetation Index (SAVI), Advanced Vegetation Index (AVI), Shadow Index (SI), Bare Soil Index (BI), Temperature Index (TI), entre otros.
- La comparación de los DEMs obtenidos utilizando otros métodos de interpolación que funcionan relativamente bien para vacíos pequeños.
- El tratamiento de DEMs que sufren del defecto de striping.

# Apéndice A

## Apéndice

### A.1. Herramientas

A continuación se listan distintas herramientas utilizadas, su funcionalidad básica, relaciones entre ellas, ventajas o desventajas y aquello que se encontró más relevante/interesante en cuanto a cada una para proporcionar una breve introducción de ellas.

#### A.1.1. OSGeo4W

Open Source Geospatial Foundation (OSGeo) es una organización sin fines de lucro cuya misión es fomentar la adopción de tecnologías geoespaciales abiertas. Entre algunos de sus proyectos más importantes se encuentran distintas librerías geoespaciales como GDAL/OGR, Orfeo Toolbox, PostGIS, etc., y aplicaciones de escritorio como QGIS y GRASS, entre otras.

El proyecto OSGeo4W es el instalador para Windows que incluye QGIS, GRASS, GDAL/OGR e interpretadores de Python al igual que otros paquetes (más de 150) [OSG].

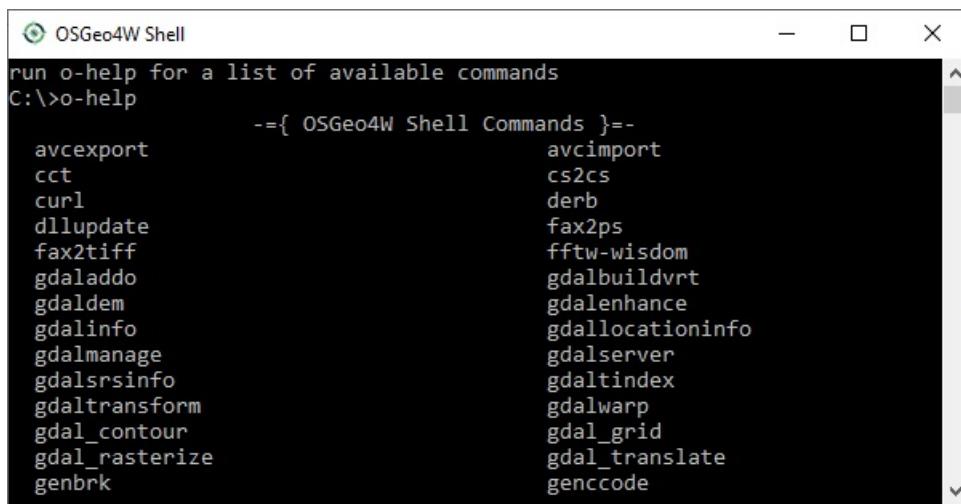
A su vez, OSGeo se encarga de OSGeolive, una máquina virtual basada en Lubuntu que permite probar una amplia variedad de software geoespacial Open Source, sin la necesidad de instalar software adicional.

##### A.1.1.1. OSGeo4W Shell

El OSGeo4W Shell (Figura A.1) es instalado a través de OSGeo4W y permite, a través de líneas de comando, ejecutar distintas aplicaciones como GDAL o scripts de Python. Permite, a su vez, instalar y administrar distintos paquetes y librerías de las distintas versiones de Python que son incluidas con la instalación.

##### A.1.1.2. QGIS

QGIS es una aplicación de escritorio que permite visualizar, editar y realizar análisis de datos geoespaciales. QGIS soporta múltiples formatos de imágenes raster y vectoriales. Se integra con múltiples paquetes GIS, incluidos PostGIS, GRASS GIS y MapServer. Distintos plugins (o complementos) escritos en Python o C++ permite extender la funcionalidad de QGIS.



```
OSGeo4W Shell
run o-help for a list of available commands
C:\>o-help
      ={ OSGeo4W Shell Commands }=-
avcexport           avcimport
cct                 cs2cs
curl                derb
dllupdate           fax2ps
fax2tiff            fftw-wisdom
gdaladdo            gdalbuildvrt
gdaldem              gdalenhance
gdalinfo             gdallocationinfo
gdalmanage           gdalserver
gdalsrsinfo          gdaltindex
gdaltransform         gdalwarp
gdal_contour         gdal_grid
gdal_rasterize       gdal_translate
genbrk               genccode
```

Figura A.1: OSGeo4W Shell luego de listar los comandos con la opción `o-help`

### A.1.1.3. GRASS

GRASS GIS se utiliza para la gestión y análisis de datos geoespaciales, procesamiento de imágenes, generación de gráficos y mapas, modelado espacial y visualización [Tea]. Originalmente fue desarrollado por “U.S. Army Construction Engineering Research Laboratories” y hoy en día cuenta con más de 350 módulos. GRASS proporciona ciertas funcionalidades que brindan soluciones a usuarios, que no están disponibles en otros GIS (al menos no de forma gratuita). Desde teledetección y gestión de imágenes satelitales/LiDAR hasta modelado de incendios forestales e inundaciones, GRASS puede enfrentar una amplia cantidad de problemas. Por otro lado, la UI/UX puede resultar un poco más compleja respecto a QGIS.

Al utilizar la versión de escritorio de GRASS, deben definirse ciertos detalles de antemano:

- Un directorio para una base de datos (GIS DATABASE). Dentro de esta DATABASE, los proyectos están organizados por áreas de proyecto almacenadas en subdirectorios llamados “LOCATIONS”.
- Una LOCATION o un proyecto: Una “localización” queda definida por un sistema de coordenadas, una proyección y límites geográficos.
- Un MAPSET o subproyecto: Cada localización puede tener múltiples MAPSETS. Cada MAPSET es un subdirectorio dentro de una LOCATION.

Una vez definidos estos elementos es posible “Comenzar una Sesión”, la cual abrirá una consola y la GUI con dos ventanas: el “layer manager” y el visualizador de mapas.

Los módulos de GRASS es posible utilizarlos desde la GUI o de la consola a través de shell scripting. A menudo es conveniente la automatización de trabajos repetitivos. El usuario puede controlar GRASS a través de scripts propios para facilitar el trabajo. Como alternativa al shell scripting existe Python. El interprete puede utilizarse desde la consola de GRASS o viene embebido en la GUI junto con un editor de texto. La librería es `grass.script` generalmente utilizada como: `import grass.script as gscript`. Los scripts pueden contener ciertas definiciones que al ser procesadas con `g.parser` definen una interfaz sin la necesidad de escribir código extra, verificando inputs y generando incluso el esqueleto para la página de ayuda.

Por último, los comandos en GRASS respetan cierta nomenclatura. La primera letra indica el formato de los datos o la funcionalidad general, seguida de un punto y una palabra breve que indica la tarea que realiza el comando. Por ejemplo:

- g.\* - Comandos generales de manejo de archivos: `g.list`, `g.copy`.
- i.\* - Comandos de procesamiento de imagen: `i.group`.
- r.\* - Comandos de procesamiento de rasters: `r.mapcalc`.
- v.\* - Comandos de procesamiento de vectores: `v.digit`, `v.to.rast`.
- m.\* - Comandos misceláneos.

#### A.1.1.4. QGIS Processing Toolbox

Una de las ventajas de QGIS, es que GRASS se encuentra integrado en este y puede ser utilizado desde su propio plugin, o a través de la “Processing Toolbox” que agrupa una serie de herramientas. Los algoritmos geoespaciales contenidos entre estas herramientas son instalados con QGIS, y de hecho, algunos de ellos llaman a aplicaciones externas, como SAGA o GRASS GIS, por lo que los algoritmos están organizados por proveedor. Algunas funcionalidades o parámetros pueden no encontrarse respecto de la aplicación original y otros comandos pueden llegar a estar divididos en más de un algoritmo, pero para ciertas funciones, la Toolbox puede resultar muy práctica.

La consola de Python permite a usuarios avanzados aumentar la productividad y realizar operaciones complejas que no se pueden realizar a través de la GUI de la Processing Toolbox [Doc]. A través de línea de comandos, se pueden definir modelos que involucren múltiples algoritmos y operaciones adicionales como bucles y sentencias condicionales para crear flujos de trabajo más flexibles y robustos.

Todos los algoritmos de la Processing Toolbox están disponibles desde la consola de Python incorporada en QGIS. Estos pueden accederse de la forma: `from qgis import processing`. Eso significa que se pueden incorporar estos algoritmos con el resto de las funcionalidades que están disponibles desde allí.

A su vez, los scripts de Python pueden convertirse en nuevos algoritmos que pueden ser llamados luego desde la Toolbox o desde el modelador gráfico de QGIS.

#### A.1.1.5. GDAL

GDAL (Geospatial Data Abstraction Library) es una librería para leer y escribir datos geoespaciales de tipo raster y vectorial en distintos formatos. Contiene a su vez una variedad de utilidades para traducción de datos y procesamiento. GDAL puede utilizarse desde el OSGeo4W Shell con shell scripting. Otra alternativa, es utilizar GDAL a través de Python importandola como: `from osgeo import gdal` o utilizando el módulo `subprocess` como `import subprocess` que permite generar un pipe de procesos que son llamados obteniendo códigos de retorno.

Por ejemplo, `gdal_translate` es utilizado para convertir datos raster entre distintos formatos potencialmente realizando operaciones como remuestreo, re-escalado de píxeles y otros. En la Figura A.1 puede observarse el comando `gdal_translate` luego de utilizar la opción para listar los comandos. En la Figura A.2 puede verse la GUI de QGIS. El diálogo “Clip Raster By Extent” es accedido a través de `Raster`

>> Extraction >> Clip Raster by Extent y contiene un Text Edit con el llamado a consola del comando. Por otro lado, el diálogo “Translate (Convert Format)” es accedido a través de la Processing Toolbox, que se encuentra ubicada por detrás de éste. En la parte inferior de la figura se encuentra el plugin de Python mostrando que es cierto que existe un método `Translate` para `gdal` y mostrando el output de correr la ayuda del comando como un subprocesso utilizando `subprocess`.

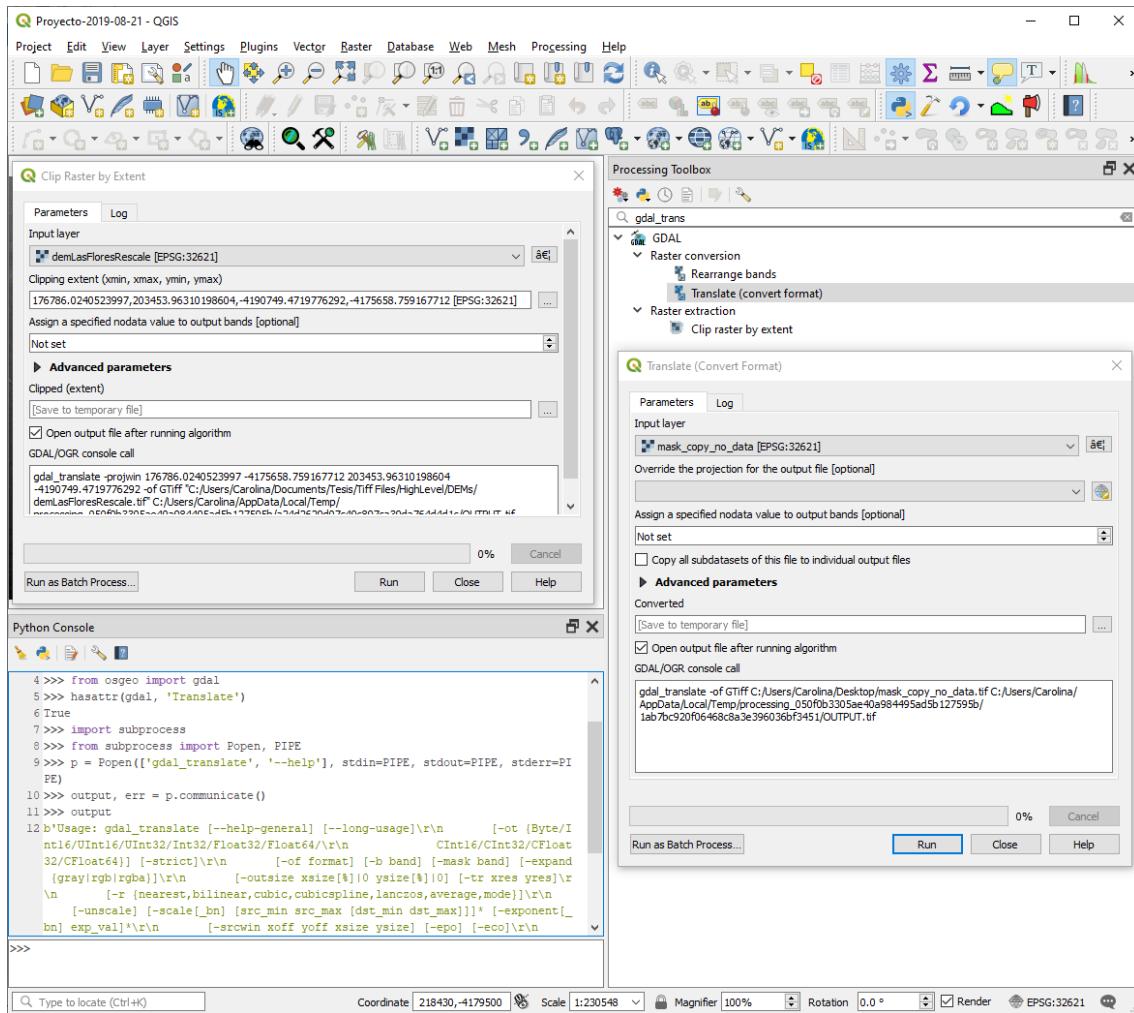


Figura A.2: QGIS GUI mostrando `gdal_translate`

### A.1.2. Python y Scikit-learn

QGIS utiliza una instalación separada de Python distinta a la del sistema operativo donde se encuentra instalado. Los interpretes de Python pueden utilizarse desde el OSGeo4W Shell y desde el plugin de Python en QGIS. Para correr Python 3 desde el OSGeo4W Shell, primero es necesario establecer el `PYTHONHOME` y el `PATH` correcto. Esto puede lograrse corriendo un archivo batch, que viene incluido con OSGeo4W, ejecutando `call py3_env.bat` en la consola.

Scikit-learn es una librería de machine learning para Python la cual cuenta con varios algoritmos de clasificación, regresión y agrupación incluyendo support vector machines, random forests, gradient boosting, k-means y DBSCAN. La librería es Open Source y tiene una licencia BSD. Algunas librerías de Python pueden instalarse

de manera bastante intuitiva a través del OSGeo4W Setup, el cual contiene una lista de librerías. La otra forma de instalar una librería es a través del comando `pip`, el cual permite instalar paquetes en Python. Para esto se debe setear las variables de entorno con el archivo batch y ejecutar las líneas `pip install nombre_libreria` o `python -m pip install nombre_libreria`. Una vez instaladas las librerías de esta forma, se podrán usar desde el plugin de Python en QGIS.

### A.1.3. Otros Plugins en QGIS

#### A.1.3.1. Plugin Builder y Plugin Loader

Para el desarrollo del plugin, se utilizó Plugin Builder. Este complemento permite crear un template para utilizar como punto de partida en el desarrollo. Dispone de tres tipos distintos de templates. Un tutorial paso a paso de un plugin que guarda atributos de una imagen vectorial en un `.csv` puede encontrarte en [Gan].

Por otro lado, Plugin Loader permite cambiar el código del plugin en construcción y que los cambios se vean reflejados en QGIS sin necesidad de reiniciar QGIS o activar/desactivar el plugin desde el administrador de estos.

#### A.1.3.2. Profile Tool

Profile Tool es una herramienta que permite generar perfiles de terreno o topográficos. Estos perfiles son una representación del relieve del terreno que se obtiene cortando transversalmente una imagen de mapa de bits o raster. A partir de las imágenes seleccionadas en QGIS, se pueden dibujar los cortes dentro de la vista del canvas lo cual genera un gráfico con la altura de cada píxel por el cual pasa el corte. Los valores pueden ser copiados incluso de forma sencilla como una tabla para ser analizados con otro software.

### A.1.4. ArcGIS World Imagery Map Server

Existe un servicio de Esri (Environmental Systems Research Institute) que provee imágenes satelitales de áreas sobre distintas partes del mundo. Este servicio es World Imagery y puede tener resoluciones de 0.3m hasta 1m para la mayoría de las zonas.

Es posible añadir estas imágenes en QGIS como una imagen o capa. La conexión con el servidor se hace desde `Layer >> Add Layer >> Add ArcGIS MapServer Layer`.

World Imagery resulta útil para obtener mejor resolución de ciertas zonas sin necesidad de descargar explícitamente imágenes con altas resoluciones que pueden ocupar mucho espacio. A su vez, se puede aplicar transparencias al layer como a cualquier otro layer en QGIS permitiendo ver con facilidad las diferencias con otras imágenes. QGIS permite también generar mapas visuales exportando las vistas del canvas como mapas en distintos formatos de imágenes, lo cual en conjunto con World Imagery puede resultar conveniente sobre la utilización de otros servicios como Google Earth.

## A.2. Cálculo de extensión y recorte de imágenes

En el siguiente script se calcula una extensión a partir de una imagen y se recortan las bandas de una imagen Landsat para poder trabajar solo con la zona de interés.

- **extension\_file\_path**: Contiene la dirección donde se encuentra la imagen que define la extensión del área de interés.
- **file\_name\_band**: Contiene la dirección y el formato del nombre de la imagen Landsat según la nomenclatura propuesta por USGS.
- **file\_name\_crop**: Contiene la dirección y el nuevo formato del nombre así se puede utilizar esta variable para generar los archivos resultantes recortados.

```

1 import subprocess
2 from osgeo import gdal
3
4 extension_dataset = gdal.Open(extension_file_path, gdal.GA_ReadOnly)
5 geoTransform = extension_dataset.GetGeoTransform()
6 minx = geoTransform[0]
7 maxy = geoTransform[3]
8 maxx = minx + geoTransform[1] * extension_dataset.RasterXSize
9 miny = maxy + geoTransform[5] * extension_dataset.RasterYSize
10
11 # Crops all images according to the extension.
12 for i in range(1, BAND_TOTAL+1):
13     command_translate =
14         "gdal_translate -projwin {} {} {} {} -ot Int16 -of GTiff \"{}\""
15         → \"{}\""
16     subprocess.call(command_translate.format(
17         minx, maxy, maxx, miny, file_name_band.format(i),
18         file_name_crop.format(i)), stdout=open(os.devnull, 'wb'),
19         shell=True)

```

## A.3. GDAL, Python y NumPy para calcular NDVI

A continuación se muestra un ejemplo sobre como leer las bandas de un archivo GeoTIFF, operar con ellas y generar un archivo nuevo a partir del resultado. En este caso se calcula el NDVI, pero podría fácilmente calcularse un filtro con SciPy o realizarse cualquier otro tipo de álgebra entre las bandas.

- **stacked\_image\_path**: Contiene la dirección donde se encuentra la imagen Landsat con las 7 bandas apiladas.
- **output\_file\_path**: Contiene la dirección de la imagen resultante a ser creada con el NDVI calculado.

```

1 from osgeo import gdal
2 from osgeo import gdal_array
3
4 # Se abre el dataset
5 # Si el dataset retorna null, es porque el abrir falló
6 dataset = gdal.Open(stacked_image_path, gdal.GA_ReadOnly)

```

```

7   # Se obtiene el tipo de dato de la imagen a partir de la primera banda
8   image_datatype = dataset.GetRasterBand(1).DataType
9
10  # Se asigna un espacio en memoria para un arreglo de NumPy, del
11  # tamaño del ráster, con dos bandas y el tipo de dato leido
12  image = np.zeros(
13      (dataset.RasterYSize, dataset.RasterXSize, 2),
14      dtype=gdal_array.GDALTypeCodeToNumericTypeCode(image_datatype))
15
16
17  # La banda roja es la 4ta
18  # Se recupera del dataset y se escribe en el arreglo
19  band = dataset.GetRasterBand(4)
20  b_red = 0
21  image[:, :, b_red] = band.ReadAsArray()
22
23  # Lo mismo para la 5ta banda (NIR)
24  band = dataset.GetRasterBand(5)
25  b_nir = 1
26  image[:, :, b_nir] = band.ReadAsArray()
27
28  # Se realiza el cálculo entre las dos bandas
29  ndvi = ((image[:, :, b_nir] - image[:, :, b_red]) /
30          (image[:, :, b_red] + image[:, :, b_nir]))
31
32  # Para generar un nuevo archivo, se define el path, tamaño del ráster,
33  # cantidad de bandas y tipo de dato de la imagen
34  out_raster_ds = gdal.GetDriverByName('GTiff').Create(
35      output_file_path, dataset.RasterXSize, dataset.RasterYSize,
36      1, gdal.GDT_Float32)
37
38  # Se setea la proyección y la geotransformación que define la posición
39  # del pixel superior izquierdo junto con la resolución y la orientación
40  out_raster_ds.SetProjection(dataset.GetProjectionRef())
41  out_raster_ds.SetGeoTransform(dataset.GetGeoTransform())
42
43  # Se recupera la primera banda del nuevo dataset y se escribe el NDVI
44  # Por último se desasigna el espacio de memoria
45  outband = out_raster_ds.GetRasterBand(1)
46  outband.WriteArray(ndvi)
47  outband.FlushCache()
48  out_raster_ds = None

```

## A.4. Cálculo de texturas con GRASS Python Scripting Library

El cálculo de texturas se realizó con un módulo de GRASS. A continuación se muestra un ejemplo reducido que da una idea de cómo se llevó a cabo esto y de cómo utilizar Python y GRASS.

Para la generación del módulo se utiliza un mecanismo de parseo de GRASS el cual provee un parser para los parámetros de los scripts, genera la ayuda y crea opcionalmente una GUI auto-generada. La función `parser()` de `grass.script` parsea los comentarios especiales que comienzan con `#%` escritos al principio del script y

procesa las entradas. Como requisito mínimo para generar un módulo, los comentarios que definen la interfaz deben contener una descripción del módulo y algunas palabras claves; luego se pueden definir las opciones. En este caso, en la linea 8 se define una opción G\_OPT\_F\_BIN\_INPUT, la cual representa un nombre para un archivo de entrada. La lista de opciones estandarizadas se puede encontrar en la documentación. Luego, se definen tres opciones como enteros y un flag. En la Figura A.3 se muestra como queda la GUI auto-generada con estos parámetros.

```

1  #>%module
2  #>% description: Compute asm, contrast, var, idm and entr for a TIFF file
3  #>% keyword: raster
4  #>% keyword: haralick
5  #>% keyword: features
6  #>%end
7
8  #>%option G_OPT_F_BIN_INPUT
9  #>% key: input
10 #>% type: string
11 #>% required : yes
12 #>%end
13
14 #>%option
15 #>% key: size
16 #>% type: integer
17 #>% required: yes
18 #>% multiple: no
19 #>% key_desc: value
20 #>% description: The size of moving window (odd and >= 3)
21 #>% answer: 3
22 #>%end
23
24 #>%option
25 #>% key: distance
26 #>% type: integer
27 #>% required: yes
28 #>% multiple: no
29 #>% key_desc: value
30 #>% label: The distance between two samples (>= 1)
31 #>% description: The distance must be smaller than the size of the moving
   ↳ window
32 #>% answer: 1
33 #>%end
34
35 #>%option
36 #>% key: categories
37 #>% type: integer
38 #>% description: Number of categories to rescale/recode the image
39 #>% answer: 255
40 #>% required : yes
41 #>%end
42
43 #>%flag
44 #>% key: r
45 #>% description: Rescales instead of recoding the image
46 #>%end

```

Luego se importan las librerías necesarias y se define una función `main()` que

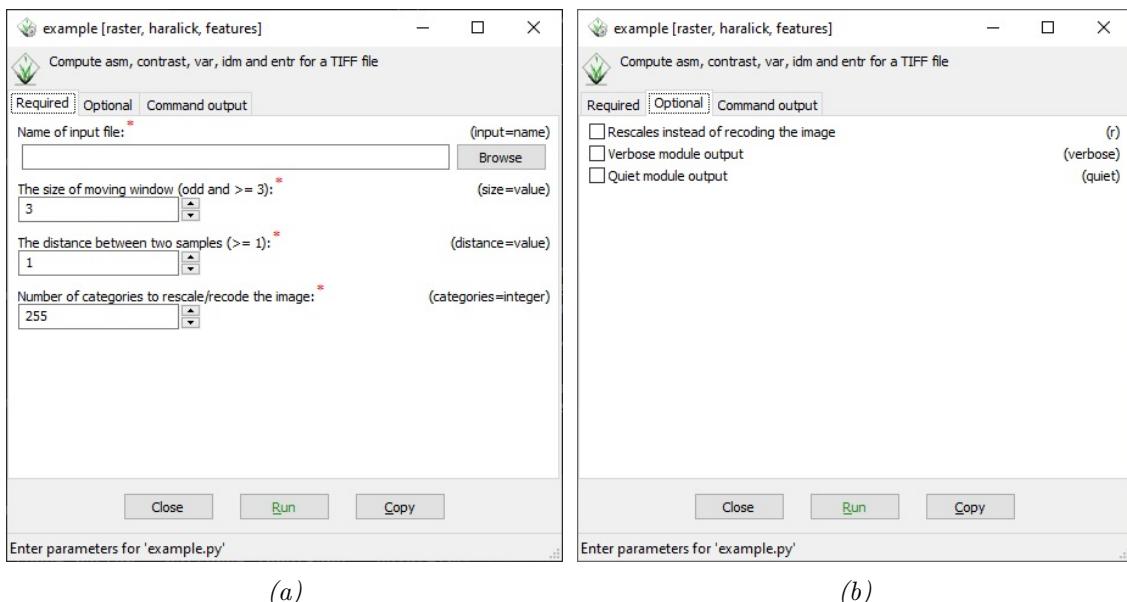


Figura A.3: Interfaz del módulo de GRASS generado

recupera las entradas y verifica si son valores dentro del rango. En caso de que no se encuentren en el rango, se imprime un mensaje sobre los valores posibles y en el caso contrario, se llama a una función `perform_task()` donde se realiza el procesamiento de la imagen.

```

47 import sys
48 import grass.script as gscript
49 from osgeo import gdal
50 import os
51 import time
52 import atexit
53
54
55 def main():
56     options,flags = gscript.parser()
57     # Se accede a los valores a través de la clave definida en las opciones
58     input_file = options['input']
59     size_moving_window = options['size']
60     distance = options['distance']
61     num_categories = options['categories']
62     recode = flags['r']
63
64     try:
65         # Validación de parámetros
66         if (int(size_moving_window) % 2 == 0) or
67             (int(size_moving_window) < 2) or
68             (int(distance) >= int(size_moving_window)) or
69             (int(num_categories) > 255):
70             raise
71     except:
72         print "Size of moving windows must be odd and >=3."
73         print "The distance must be smaller than the size of the moving
74             ↴ window."
75         print "The raster map cannot have more than 255 categories."
76     else:

```

```

76     perform_task(options, flags, input_file, size_moving_window,
77                   distance, num_categories, recode)

```

Para procesar una imagen, primero debe tenerse en cuenta la proyección de la misma. En este caso con `g.proj` se establece la proyección a partir de la imagen de entrada. La proyección debe actualizarse dentro del mapset que GRASS crea por defecto (`PERMANENT`).

Con `r.in.gdal` se importan los rásters de las imágenes. En este caso se considera que la imagen tiene una sola banda ya que USGS separa cada banda en archivos distintos. Si la imagen posee mas de una banda, los ráster se agregan a la base de datos de GRASS respetando el nombre del output y agregando como sufijo `.n` donde `n` depende del número de la banda.

El comando `g.region` permite establecer la región computacional que define la extensión en la cual se quiere trabajar. En este caso se toma toda la imagen de entrada, pero las imágenes descargadas de USGS tienen una extensión de aproximadamente  $40.000\text{km}^2$  donde es posible que la región de interés sea mucho más chica y se quiera obtener a partir de otra imagen.

En base a la opción `-r` se determina si se debe recodificar o re-escalar la imagen según la cantidad de categorías deseadas. La cuantificación se realiza con `r.recode` o `r.rescale`. Para el recodificado es necesario especificar las reglas de recodificación que definen los intervalos de los niveles de gris y para eso se utiliza `r.quantile`. Este último modulo genera el output como salida de texto, y normalmente en bash se utilizaría con un pipe. En este caso, en vez de `run_command()`, el concepto del pipe se genera con `read_command()` y `write_command()`.

Al ráster ya cuantificado se le calculan las texturas con `r.texture` y luego se genera un grupo con la salida para poder generar un archivo multibanda a través de GDAL.

```

78     def perform_task(options, flags, input_file, size_moving_window, distance,
79                       num_categories, recode):
80
81         gscript.run_command('g.mapset', mapset='PERMANENT')
82
83         gscript.run_command('g.proj', flags='c', georef=input_file)
84
85         # Importa datos de tipo ráster en GRASS usando GDAL
86         gscript.run_command('r.in.gdal', flags='k', input=input_file,
87                           output='inputBand', overwrite=True)
88
89         # Determina los límites de la región geográfica a utilizar
90         gscript.run_command('g.region', raster='inputBand')
91
92         # Se determina si recodificar o re-escalar la imagen
93         if recode == False:
94             rules = gscript.read_command(
95                 'r.quantile', flags='r', input='inputBand',
96                 quantiles=num_categories, overwrite=True, quiet=True)
97             gscript.write_command(
98                 'r.recode', input='inputBand', output='bandRescale',
99                 rules='-', overwrite=True, stdin=rules)
100        else:
101            gscript.run_command(
102                'r.rescale', input='inputBand', output='bandRescale',

```

```

103         to='1','+num_categories, overwrite=True)
104
105     # Se calculan las texturas de la imagen cuantificada
106     gscript.run_command(
107         'r.texture', flags='n', input='bandRescale', output='band',
108         method='asm,contrast,var,idm,entr', size=size_moving_window,
109         distance=distance, overwrite=True)
110
111     # Se crea un grupo para exportar la imagen tiff
112     group_input = 'band_ASM,band_Contr,band_Var,band_IDM,band_Eentr'
113     gscript.run_command(
114         'i.group', group='outFileBands', input=group_input, quiet=True)
115
116     # Se exporta la imagen tiff a partir del grupo creado
117     output_file = input_file[:-4] + "_text.tif"
118     gscript.run_command(
119         'r.out.gdal', flags='cm', input='outFileBands',
120         output=output_file, format='GTiff', type='Float32',
121         overwrite=True, verbose=True)

```

Por último se define una función `cleanup()` la cual elimina los grupos y rásters importados para no dejar ningún tipo de dato temporal en la base de datos de GRASS. Si la función `cleanup()` es llamada normalmente al terminar la ejecución, como cualquier otra función, se limpian los datos generados. Si el script falla a mitad de ejecución, la función no es llamada y pueden quedar elementos que no fueron eliminados. Por este motivo, se utiliza la función `atexit()` la cual garantiza que siempre se llame la función `cleanup()` al terminar el script.

Por otro lado, `g.remove`, dado un tipo de dato y un patrón del nombre, elimina de la base de datos todos los elementos que coinciden con esas características.

```

122 def cleanup():
123     # Remueve los rásters de un grupo
124     cf = gscript.find_file(name='outFileBands', element='group')
125     if not cf['fullname'] == '':
126         gscript.run_command('g.remove', flags='f', type='group',
127             name='outFileBands', quiet=True)
128
129     # Remover los rásters
130     cf = gscript.find_file(name='band_ASM', element='cell')
131     if not cf['fullname'] == '':
132         gscript.run_command('g.remove', flags='f', type='raster',
133             pattern='band_*', quiet=True)
134
135     cf = gscript.find_file(name='bandRescale', element='cell')
136     if not cf['fullname'] == '':
137         gscript.run_command('g.remove', flags='f', type='raster',
138             pattern='bandRescale', quiet=True)
139
140     cf = gscript.find_file(name='inputBand', element='cell')
141     if not cf['fullname'] == '':
142         gscript.run_command('g.remove', flags='f', type='raster',
143             pattern='inputBand', quiet=True)
144
145
146 if __name__ == '__main__':
147     atexit.register(cleanup)

```

148

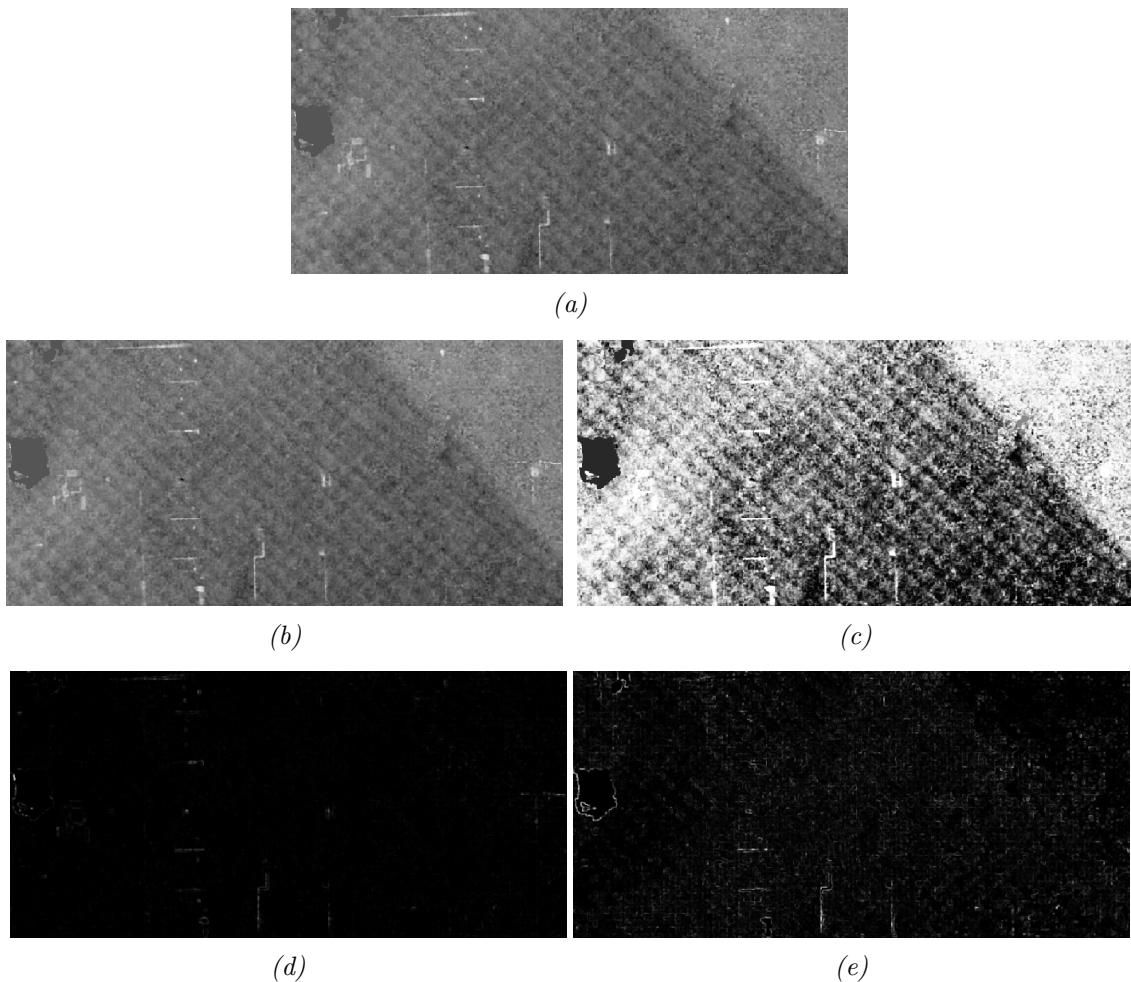
`main()`

En el script original, realmente se trabaja con todas las bandas de una misma imagen Landsat y todas las texturas son apiladas para luego poder ser utilizadas como indicadores. Existe una clase llamada `ParallelModuleQueue` provista por `grass.pygrass.modules` que permite paralelizar el cálculo de las texturas. Los módulos se agregan a una cola de procesamiento y se van eliminando a medida que terminan su ejecución. En la práctica, comparando los tiempos de ejecución de la versión secuencial y paralela, se obtuvo una aceleración de 2.24 para la porción del script que calcula las texturas.

## A.5. Re-escalado vs Recodificado

En las Figuras A.4 y A.5 se observa la diferencia entre re-escalcar y recodificar un DEM; se nota claramente como el constaste calculado a partir de la recodificación resalta desperfectos en las imágenes y no necesariamente la vegetación arbórea. Particularmente, en A.4e se resalta aún más la diferencia de la elevación del terreno respecto de A.5e y el contraste deja de resaltar las líneas de árboles.

A su vez, el DEM de la Figura A.4 sufre del artefacto de striping que se menciona en la Sección 2.6, y el contraste calculado a partir de la imagen recodificada resalta esta alteración no deseada de manera más intensa respecto al re-escalado.



*Figura A.4: DEMs de SRTM correspondientes a una zona cerca de General Villegas, provincia de Buenos Aires. Se observa como recodificar la imagen intensifica el ruido y es trasladado al cálculo de las texturas. (a) DEM original. (b) DEM re-escalado. (c) DEM recodificado. (d) Contraste correspondiente al DEM re-escalado. (e) Contraste correspondiente al DEM recodificado.*

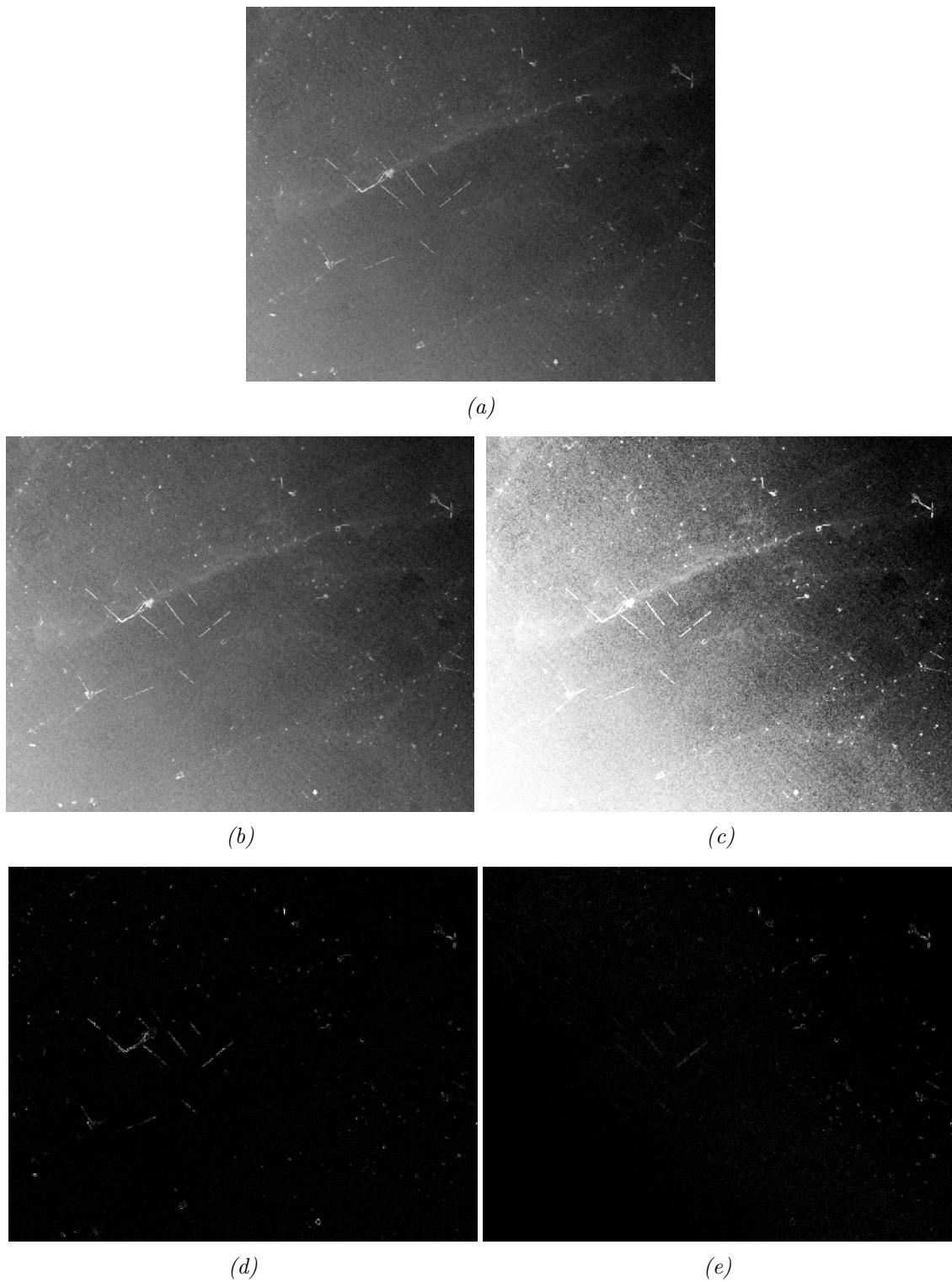


Figura A.5: DEMs de SRTM correspondientes a una zona cerca de Las Flores, provincia de Buenos Aires. Se observa como recodificar la imagen resalta la pendiente en el DEM y los bordes se ven afectados negativamente. (a) DEM original. (b) DEM re-escalado. (c) DEM recodificado. (d) Contraste correspondiente al DEM re-escalado. (e) Contraste correspondiente al DEM recodificado.

# Bibliografía

- [AG12] Özlem Akar y Oguz Gungor. «Classification of Multispectral Images Using Random Forest Algorithm». En: *Journal of Geodesy and Geoinformation* 1 (ene. de 2012), págs. 105-112. DOI: 10.9733/jgg.241212.1.
- [Ant+] G. Antoniol et al. *r.texture - GRASS GIS manual*. URL: <https://grass.osgeo.org/grass78/manuals/r.texture.html>.
- [Bre01] Leo Breiman. «Random Forests». En: *Machine Learning* 45.1 (2001), págs. 5-32. DOI: <http://dx.doi.org/10.1023/A:1010933404324>.
- [Che+10] Daoyi Chen et al. «Assessment of Open Source GIS Software for Water Resources Management in Developing Countries». En: *Journal of Hydro-environment Research* 4 (oct. de 2010). DOI: 10.1016/j.jher.2010.04.017.
- [Doc] QGIS Documentation. *Using processing algorithms from the console*. URL: [https://docs.qgis.org/testing/en/docs/user\\_manual/processing/console.html](https://docs.qgis.org/testing/en/docs/user_manual/processing/console.html).
- [EC17] William Emery y Adriano Camps. *Chapter 5 - Radar*. Ed. por William Emery y Adriano Camps. Elsevier, 2017, págs. 291-453. ISBN: 978-0-12-809254-5. DOI: <https://doi.org/10.1016/B978-0-12-809254-5.00005-1>.
- [ESRa] ESRI. *How inverse distance weighted interpolation works*. URL: <https://pro.arcgis.com/en/pro-app/help/analysis/geostatistical-analyst/how-inverse-distance-weighted-interpolation-works.htm>.
- [ESRb] ESRI. *Resample*. URL: <https://pro.arcgis.com/en/pro-app/tool-reference/data-management/resample.htm>.
- [Fel94] Ángel Manuel Felicísimo. *Modelos digitales del terreno*. Oviedo, España: Pentalfa, 1994.
- [Gan] Ujal Gandhi. *Building a Python Plugin (QGIS3)*. URL: [https://www.qgistutorials.com/en/docs/3/building\\_a\\_python\\_plugin.html](https://www.qgistutorials.com/en/docs/3/building_a_python_plugin.html).
- [GDA] GDAL. *GDAL Algorithms C API*. URL: [https://gdal.org/api/gdal\\_alg.html](https://gdal.org/api/gdal_alg.html).
- [GIS15] GISGeography. *27 Differences Between ArcGIS and QGIS – The Most Epic GIS Software Battle in GIS History*. 2015. URL: <https://gisgeography.com/qgis-arcgis-differences/>.

- [GPG10] Mrinal Kanti Ghose, Ratika Pradhan y Sucheta Sushan Ghose. «Decision Tree Classification of Remotely Sensed Satellite Data using Spectral Separability Matrix». En: (2010).
- [GRD12] John Gallant, A. Read y T. Dowling. «REMOVAL OF TREE OFF-SETS FROM SRTM AND OTHER DIGITAL SURFACE MODELS». En: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XXXIX-B4 (jul. de 2012), págs. 275-280. DOI: 10.5194/isprarchives-XXXIX-B4-275-2012.
- [Hei18] Hans Karl Heidemann. *Lidar Base Specification*. Reston, VA, EEUU: U.S. Geological Survey, 2018. DOI: <https://doi.org/10.3133/tm11b4>.
- [HSD73] R. M. Haralick, K. Shanmugam e I. Dinstein. «Textural Features for Image Classification». En: *IEEE Transactions on Systems, Man, and Cybernetics* SMC-3.6 (1973), págs. 610-621. ISSN: 2168-2909. DOI: 10.1109/TSMC.1973.4309314.
- [HTF01] Trevor Hastie, Robert Tibshirani y Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., 2001.
- [KB09] R Köthe y Michael Bock. «Preprocessing of Digital Elevation Models – derived from Laser Scanning and Radar Interferometry – for Terrain Analysis in Geosciences». En: (ene. de 2009).
- [Ken+09] R. E. Kenedy et al. «Remote sensing change detection tools for natural resource managers: understanding concepts and tradeoffs in the design of landscape monitoring projects.» En: *Remote Sensing of Environment* 113.7 (2009), págs. 1382-1396. DOI: <https://doi.org/10.1016/j.rse.2008.07.018>.
- [LDL16] L. Lo Vercio, M. Del Fresno e I. Larrabide. «Detection of morphological structures for vessel wall segmentation in IVUS using random forests». En: *12th International Symposium on Medical Information Processing and Analysis* 10160 (2016). DOI: <https://doi.org/10.1117/12.2255748>.
- [Lei+12] Ira Leifer et al. «State of the art satellite and airborne marine oil spill remote sensing: Application to the BP Deepwater Horizon oil spill». En: *Remote Sensing of Environment* 124 (2012), págs. 185-209. DOI: <http://dx.doi.org/10.1016/j.rse.2012.03.024>.
- [LK15] Barrett Lowe y Arun Kulkarni. «Multispectral Image Analysis Using Random Forest». En: *International Journal on Soft Computing* 6 (feb. de 2015), págs. 1-14. DOI: 10.5121/ijsc.2015.6101.
- [Mel+15] A. Mellor et al. «Exploring issues of training data imbalance and mislabelling on random forest performance for large area land cover classification using the ensemble margin». En: *ISPRS Journal of Photogrammetry and Remote Sensing* 105 (2015), págs. 155-168. DOI: <https://doi.org/10.1016/j.isprsjprs.2015.03.014>.
- [Oil12] International Association of Oil & Gas Producers. «Using the EPSG geodetic parameter dataset». En: (2012), págs. 10-14.

- [OSG] OSGeo. *OSGeo4W*. URL: <https://trac.osgeo.org/osgeo4w/>.
- [QGIa] QGIS. *Creating a New Vector Dataset*. URL: [https://docs.qgis.org/3.4/en/docs/training\\_manual/create\\_vector\\_data/create\\_new\\_vector.html](https://docs.qgis.org/3.4/en/docs/training_manual/create_vector_data/create_new_vector.html).
- [QGIb] QGIS. *DEM from LiDAR Data*. URL: [https://docs.qgis.org/3.4/en/docs/training\\_manual/forestry/basic\\_lidar.html](https://docs.qgis.org/3.4/en/docs/training_manual/forestry/basic_lidar.html).
- [RNJ07] Hannes Reuter, Andy Nelson y Andrew Jarvis. «An Evaluation of Void-Filling Interpolation Methods for SRTM Data». En: *International Journal of Geographical Information Science* 21 (oct. de 2007), págs. 983-1008. DOI: [10.1080/13658810601169899](https://doi.org/10.1080/13658810601169899).
- [RS04] Laura Raileanu y Kilian Stoffel. «Theoretical Comparison between the Gini Index and Information Gain Criteria». En: *Annals of Mathematics and Artificial Intelligence* 41 (mayo de 2004), págs. 77-93. DOI: [10.1023/B:AMAI.0000018580.96245.c6](https://doi.org/10.1023/B:AMAI.0000018580.96245.c6).
- [SSM10] John Stevenson, Xianfang Sun y Neil Mitchell. «Despeckling SRTM and other topographic data with a denoising algorithm». En: *Geomorphology* 114 (ene. de 2010), págs. 238-252. DOI: [10.1016/j.geomorph.2009.07.006](https://doi.org/10.1016/j.geomorph.2009.07.006).
- [TDR15] A. Thomas, J.P. D'Amato y P. Rinaldi. «Corrección de Modelos Digitales de Elevación con Imágenes Multi-Espectrales». En: *XXI Congreso Argentino de Ciencias de la Computación* (2015).
- [Tea] GRASS Development Team. *GRASS GIS*. URL: <https://grass.osgeo.org/>.
- [Tia16] B. Tian. *GIS Technology Applications in Environmental and Earth Sciences*. CRC Press, 2016, pág. 24. ISBN: 9781498776059. URL: <https://books.google.com.ar/books?id=QKqKDQAAQBAJ>.
- [TS13] Tesfaye H. Tarekegn y Takahiro Sayama. «Correction of SRTM DEM artefacts by Fourier Transform for Flood Inundation Modeling». En: *Journal of Japan Society of Civil Engineers, Ser. B1 (Hydraulic Engineering)* 69.4 (2013), I\_193-I\_198. DOI: [https://doi.org/10.2208/jscejhe.69.I\\_193](https://doi.org/10.2208/jscejhe.69.I_193).
- [USGa] USGS. *About Us*. URL: <https://www.usgs.gov/about/about-us>.
- [USGb] USGS. *Landsat Collections*. URL: [https://www.usgs.gov/land-resources/nli/landsat/landsat-collection-1?qt-science\\_support\\_page\\_related\\_con=1#qt-science\\_support\\_page\\_related\\_con](https://www.usgs.gov/land-resources/nli/landsat/landsat-collection-1?qt-science_support_page_related_con=1#qt-science_support_page_related_con).
- [USGc] USGS. *USGS EROS Archive - Digital Elevation - Shuttle Radar Topography Mission (SRTM) 1 Arc-Second Global*. URL: [https://www.usgs.gov/centers/eros/science/usgs-eros-archive-digital-elevation-shuttle-radar-topography-mission-srtm-1-arc?qt-science\\_center\\_objects=0#qt-science\\_center\\_objects](https://www.usgs.gov/centers/eros/science/usgs-eros-archive-digital-elevation-shuttle-radar-topography-mission-srtm-1-arc?qt-science_center_objects=0#qt-science_center_objects).

- [USGd] USGS. *USGS EROS Archive - Landsat Archives - Landsat 8 OLI/TIRS Level-2 Data Products - Surface Reflectance*. URL: [https://www.usgs.gov/centers/eros/science/usgs-eros-archive-landsat-archives-landsat-8-olitirs-level-2-data-products?qt-science\\_center\\_objects=0#qt-science\\_center\\_objects](https://www.usgs.gov/centers/eros/science/usgs-eros-archive-landsat-archives-landsat-8-olitirs-level-2-data-products?qt-science_center_objects=0#qt-science_center_objects).
- [Wik17] GRASS Wiki. *Map Reprojection*. 2017. URL: [https://grasswiki.osgeo.org/wiki/Map\\_Reprojection](https://grasswiki.osgeo.org/wiki/Map_Reprojection).
- [WL08] Jennifer Watts y R Lawrence. «Merging random forest classification with an object-oriented approach for analysis of agricultural lands». En: *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 37 (ene. de 2008).
- [Yan+19] Yingying Yang et al. «The NDVI-CV Method for Mapping Evergreen Trees in Complex Urban Areas Using Reconstructed Landsat 8 Time-Series Data». En: *Forests* 10 (feb. de 2019), pág. 139. DOI: 10.3390/f10020139.
- [You+17] Nicholas E. Young et al. «A survival guide to Landsat preprocessing». En: *Ecology: Ecological Society of America* 98.4 (2017), págs. 920-932. DOI: <https://doi.org/10.1002/ecy.1730>.